

Scilab Textbook Companion for
Process Heat Transfer
by D. Q. Kern¹

Created by
Byragoni Spandana
Physics
Electrical Engineering
IIT-B
College Teacher
None
Cross-Checked by
Lavitha

June 11, 2015

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: Process Heat Transfer

Author: D. Q. Kern

Publisher: Tata McGraw-Hill, NY

Edition: 1

Year: 1950

ISBN: 0-07-085353-3

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes	5
2 Conduction	9
4 Radiation	14
5 Temperature	17
6 Counterflow Double pipe exchangers	23
7 Parallel Counterflow Shell and Tube Exchangers	33
8 Flow Arrangements for increased heat recovery	53
9 Gases	62
10 Streamline flow and steam convection	69
11 Calculations for process conditions	80
12 Condensation of single vapor	104
13 Condensation of Mixed vapors	132
14 Evaporation	164
15 Vaporizers Evaporators and Reboilers	185
16 Extended Surfaces	213

17 Cooling Towers	228
18 Batch and unsteady process	240
19 Furnace Calculations	250
20 Additional applications	257

List of Scilab Codes

Exa 2.1	Value of Q	9
Exa 2.2	Temperature increase	9
Exa 2.3	Value of Q	11
Exa 2.4	Heat flow through pipe wall	12
Exa 2.5	Temperature	12
Exa 4.1	Radiation between large 2 planes	14
Exa 4.2	Radiation between Planes with Different Emissivities	14
Exa 4.3	Calculation of Radiation from a Pipe	15
Exa 4.4	Radiation from a Pipe to a Duct	16
Exa 5.1	Calculation of the LMTD	17
Exa 5.2	Calculation of the LMTD with Equal Outlet Temperatures	18
Exa 5.3	Calculation of the LMTD	18
Exa 5.4	Calculation of the LMTD with One Isothermal Fluid	19
Exa 5.5	Calculation of point	20
Exa 5.6	Calculation of the Caloric Temperature	21
Exa 6.1	Double Pipe Benzene Toluene Exchanger	23
Exa 6.2	Calculation of the True Temperature Difference	27
Exa 6.3	Double Pipe Lube Oil Crude Oil Exchanger	28
Exa 7.1	shell side equivalent	33
Exa 7.2	Calculation of FT for Fluids with Equal Ranges	33
Exa 7.3	Calculation of a Kerosene Crude Oil Exchanger	35
Exa 7.4	Calculation of a Distilled water Raw water Exchanger	39
Exa 7.5	Calculation of the Optimum Outlet water Temperature	43
Exa 7.6	Calculation of a Phosphate Solution Cooler	44
Exa 7.7	The Optimum Use of Exhaust and Process Steam	47
Exa 7.8	Calculation of a Sugar solution Heater without Baffles	48
Exa 7.9	Outlet Temperatures	52

Exa 8.1	Calculation of a Oil Cooler	53
Exa 8.2	calculation of an Acetone Acetic Acid Exchanger	57
Exa 9.1	Calculation of an Ammonia Compressor Aftercooler	62
Exa 9.2	Calculation of the Heat Load for an Air Intercooler	66
Exa 9.3	Calculation of the Dew Point after Compression	68
Exa 10.1	Crude Oil Heater	69
Exa 10.2	Kerosene Heater	71
Exa 10.3	Gas Oil Heater Using Cores	74
Exa 10.4	Calculation of a Heating Bundle for an Aniline Storage Tank	78
Exa 11.1	Calculation of a Straw Oil Naphtha Exchanger	80
Exa 11.2	Calculation of a Flue Gas Cooler	84
Exa 11.3	Calculation of a Caustic Solution Cooler	89
Exa 11.4	Calculation of an Alcohol Heater	94
Exa 11.5	Calculation of a Flue Gas Cooler	98
Exa 12.1	Calculation of a Horizontal n Propanol Condenser	104
Exa 12.2	Design of a Vertical n Propanol Condenser	108
Exa 12.3	Calculation of a Butane Desuperheater condenser	111
Exa 12.4	Calculation of a Vertical Condenser subcooler	117
Exa 12.5	Calculation of a Horizontal Condenser subcooler	122
Exa 12.6	Calculation of Vertical Reflux type CSo Condenser	126
Exa 12.7	Calculation of a Surface Condenser	130
Exa 13.1	Calculation of the Bubble Point	132
Exa 13.2	Calculation of the Bubble Point and Vapor Composition by Relative	138
Exa 13.3	Condenser Calculations for a Multicomponent Mixture	140
Exa 13.4	Vapor	147
Exa 13.5	Calculation of a Steam Carbon Dioxide Condenser	147
Exa 13.6	Calculation of a Condenser	151
Exa 14.1	Calculation of Evaporator Surface	164
Exa 14.2	Calculation of a Triple effect Forward feed Evaporator	165
Exa 14.3	Backward feed Multiple effect Evaporator	167
Exa 14.4	evoparator installer	169
Exa 14.5	unit calculation	174
Exa 14.6	Evoparator specification	178
Exa 14.7	heat and steam	183
Exa 15.1	Calculation of the Average Specific Volume	185

Exa 15.2	Vaporizer or Pump through Reboiler with Isothermal Boiling	186
Exa 15.3	Calculation of a Kettle Reboiler	191
Exa 15.4	Calculation of a Once through Horizontal Thermosyphon Reboiler	193
Exa 15.5	Calculation of a Vertical Thermosyphon Reboiler	197
Exa 15.6	Calculation of the Reboiler Duty	201
Exa 15.7	Distillation of a Binary Mixture	202
Exa 15.8	The Reboiler Duty for a Multicomponent Mixture	206
Exa 16.1	Calculation of the Fin Efficiency and a Weighted Efficiency Curve	213
Exa 16.2	Calculation of a Heat transfer Curve from Experimental Data	214
Exa 16.3	Calculation of a Double Pipe Extended surface Gas Oil Cooler	215
Exa 16.4	Calculation of a Longitudinal Fin Shell and tube Exchanger	220
Exa 16.5	Calculation of a Transverse fin Air Cooler	223
Exa 17.1	Calculation of the Enthalpy of Saturated Air	228
Exa 17.2	Calculation of the Number of Diffusion Units	228
Exa 17.3	Calculation of the Required Height of Fill	230
Exa 17.4	Determination of a Cooling tower Guarantee	230
Exa 17.5	The Recalculation of Cooling tower Performance	231
Exa 17.6	Calculation of a Direct contact Gas Cooler	235
Exa 17.7	Approximate Calculation of a Gas Cooler	238
Exa 18.1	Calculation of Batch Heating	240
Exa 18.2	Heat Flow through a Wall	242
Exa 18.3	Center line Temperature of a Shaft	243
Exa 18.4	The Schack Chart	243
Exa 18.5	The Gurney Lurie Chart	245
Exa 18.6	The Application of Newmans Method to Heating a Brick	245
Exa 18.7	The Graphical Determination of the Time Temperature Distribution	247
Exa 18.8	Calculations for a Wall with Periodic Temperature Variation	247
Exa 18.9	Calculation of the Length of a Bed	248
Exa 19.1	Lobo and Evans	250

Exa 19.2	Calculation of a Furnace by the Method of Wilson and Lobo and Hottel	253
Exa 19.3	Calculation of Performance by the Orrok Hudson Equation	253
Exa 19.4	Calculation of the Equivalent Radiant Coefficient	254
Exa 19.5	Furnace calculation	255
Exa 20.1	Calculation of a Jacketed Vessel	257
Exa 20.2	Calculation of a Tube Coil	258
Exa 20.3	Calculation of a Submerged pipe Coil Slurry Cooler	259
Exa 20.4	Calculation of a Trombone 2	261
Exa 20.5	Calculation of an Atmospheric Jacket Water Cooler	264
Exa 20.6	Calculation of the True Temperature Difference	266
Exa 20.7	calculation of Sand Cooling with Negllgible Resistance	267
Exa 20.8.1	Immersion Water Heater	268
Exa 20.8.2	Strip Heater for Air Heating	269
Exa 20.8.3	Finned strip Heater	270
Exa 20.8.4	Clamp on Plastic Heating	270

Chapter 2

Conduction

Scilab code Exa 2.1 Value of Q

```
1  clc
2  //page 13
3  printf("\t Example 2.1 \n");
4  printf("\t approximate values are mentioned in the
   book \n");
5  Tavg=900; // average temperature of the wall ,F
6  k=0.15; // Thermal conductivity at 932 F,Btu/(hr)(ft
   ^2)(F/ft)
7  T1=1500; // hot side temperature ,F
8  T2=300; // cold side temperature ,F
9  A=192; // surface area ,ft^2
10 L=0.5; // thickness ,ft
11 Q=(k)*(A)*(T1-T2)/L; // formula for heat ,Btu/hr
12 printf("\t heat is : %.2e Btu/hr \n",Q);
13 //end
```

Scilab code Exa 2.2 Temperature increase

```

1 //Page 14
2 clc
3 printf("\t Example 2.2 \n");
4 printf("\t approximate values are mentioned in the
   book \n");
5 La=8/12; // Thickness of firebrick wall,ft
6 Lb=4/12; // Thickness of insulating brick wall,ft
7 Lc=6/12; // Thickness of building brick wall,ft
8 Ka=0.68; // themal conductivity of firebrick ,Btu/(hr
   )*(ft ^2)*(F/ft)
9 Kb=0.15; // themal conductivity of insulating brick ,
   Btu/(hr)*(ft ^2)*(F/ft)
10 Kc=0.40; // themal conductivity of building brick ,
   Btu/(hr)*(ft ^2)*(F/ft)
11 A=1; // surface area ,ft^2
12 Ta=1600; // temperature of inner wall,F
13 Tb=125; // temperature of outer wall.F
14 Ra=La/(Ka)*(A); // formula for resistance ,(hr)*(F)/
   Btu
15 printf("\t resistance offered by firebrick : %.2f (
   hr)*(F)/Btu \n",Ra);
16 Rb=Lb/(Kb)*(A); // formula for resistance ,(hr)*(F)/
   Btu
17 printf("\t resistance offered by insulating brick :
   %.2f (hr)*(F)/Btu \n",Rb);
18 Rc=Lc/(Kc)*(A); // formula for resistance ,(hr)*(F)/
   Btu
19 printf("\t resistance offered by buildingbrick : %.2
   f (hr)*(F)/Btu \n",Rc);
20 R=Ra+Rb+Rc; // total resistance offered by three
   walls ,(hr)*(F)/Btu
21 printf("\t total resistance offered by three walls :
   %.2f (hr)*(F)/Btu \n",R);
22 Q=(1600-125)/4.45; // using formula for heat loss/ft
   ^2,Btu/hr
23 printf("\t heat loss/ft^2 : %.0f Btu/hr \n",Q);
24 // T1,T2 are temperatures at interface of firebrick
   and insulating brick , and insulating brick and

```

```

    building brick respectively ,F
25 delta=(Q)*(Ra); // formula for temperature
    difference ,F
26 printf("\t delta is : %.0f F \n",delta);
27 T1=Ta-((Q)*(Ra)); // temperature at interface of
    firebrick and insulating brick ,F
28 printf("\t temperature at interface of firebrick and
    insulating brick :%.0f F \n",T1);
29 deltb=Q*(Rb);
30 printf("\t deltb is : %.0f F \n",deltb);
31 T2=T1-((Q)*(Rb)); //temperature at interface of
    insulating brick and building brick ,F
32 printf("\t temperature at interface of insulating
    brick and building brick :%.0f F \n",T2);
33 //end

```

Scilab code Exa 2.3 Value of Q

```

1 clc
2 //page 15
3 printf("\t example 2.3 \n");
4 printf("\t approximate values are mentioned in the
    book \n");
5 Lair=0.25/12; // thickness of air film ,ft
6 Kair=0.0265; // thermal conductivity of air at 572F,
    Btu/(hr)*(ft^2)(F/ft)
7 A=1; // surface area ,ft^2
8 Rair=Lair/(Kair*(A)); // resistance offered by air
    film , (hr)(F)/Btu
9 printf("\t resistance offered by air film %.2f (hr)(
    F)/Btu \n",Rair);
10 R=4.45; // resistance from previous example 2.2 ,(hr)
    (F)/Btu
11 Rt=(R)+Rair; // total resistance ,(hr)(F)/Btu
12 printf("\t total resistance %.2f (hr)(F)/Btu \n",Rt)

```

```

;
13 Ta=1600; // temperature of inner wall ,F
14 Tb=125; // temperature of outer wall ,F
15 Q=(1600-125)/Rt; // heat loss , Btu/hr
16 printf("\t heat loss %.2f Btu/hr \n",Q);

```

Scilab code Exa 2.4 Heat flow through pipe wall

```

1 //page 16
2 clc
3 printf("\t example 2.4 \n");
4 printf("\t approximate values are mentioned in the
   book \n");
5 k=0.63; // thermal conductivity of pipe , Btu/(hr)*(
   ft ^2)*(F/ft)
6 Do=6; // in
7 Di=5; // in
8 Ti=200; // inner side temperature ,F
9 To=175; // outer side temperature ,F
10 q=(2*(3.14)*(k)*(Ti-To))/(2.3*log10(Do/Di)); //
   formula for heat flow ,Btu/(hr)*(ft)
11 printf("\t heat flow is : %.0f Btu/(hr)*(ft) \n",q);
12 // caculation mistake in book
13 // end

```

Scilab code Exa 2.5 Temperature

```

1 clc
2 //page 19
3 printf("\t example 2.5 \n");
4 printf("\t approximate values are mentioned in the
   book \n");

```

```

5 t1=150; // assume temperature of outer surface of
   rockwool ,F
6 ta=70; // temperature of surrounding air ,F
7 ha=2.23; // surface coefficient ,Btu/(hr)*(ft ^2)*(F)
8 q=(3.14)*(300-70)/(((2.3/(2*0.033))*log10
   (3.375/2.375))+1/(((2.23)*(3.375/12))))); // using
   formula for heat loss ,Btu/(hr)*(lin ft),
   calculation mistake
9 printf("\t heat loss for linear foot is : %.1f Btu/(
   hr)*(lin ft) \n",q);
10 printf("\t Check between ts and t1, since deltc/R =
   deltc/Rc \n");
11 t1=300-(((104.8)*((2.3)*(log10(3.375/2.375)))))/((2)
   *(3.14)*(.033)); // using eq 2.31,F
12 printf("\t t1 is : %.1f F \n",t1);
13 t1=125; // assume temperature of outer surface of
   rockwool ,F
14 ha=2.10; // surface coefficient ,Btu/(hr)*(ft ^2)*(F)
15 q=((3.14)*(300-70))/(((2.3/(2*0.033))*log10
   (3.375/2.375))+1/(((2.10)*(3.375/12))))); // using
   formula for heat loss ,Btu/(hr)*(lin ft)
16 printf("\t heat loss for linear foot is : %.1f Btu/(
   hr)*(lin ft) \n",q);
17 printf("\t Check between ts and t1, since deltc/R =
   deltc/Rc \n");
18 t1=300-(((103)*((2.3)*(log10(3.375/2.375)))))/((2)
   *(3.14)*(.033)); // using eq 2.31,F
19 printf("\t t1 is : %.1f F \n",t1);
20 // end

```

Chapter 4

Radiation

Scilab code Exa 4.1 Radiation between large 2 planes

```
1 clc
2 //page 75
3 printf("\\t example 4.1 \\n");
4 printf("\\t approximate values are mentioned in the
   book \\n");
5 T1=1000+460; // R
6 T2=800+460; // R
7 Q=((0.173)*((14.6)^4-(12.6)^4)); // using eq.4.24,
   Btu/(hr)*(ft^2)
8 printf("\\t heat removed from colder wall per unit
   area is : %.0f Btu/(hr)*(ft^2) \\n",Q);
9 // end
```

Scilab code Exa 4.2 Radiation between Planes with Different Emissivities

```
1 printf("\\t example 4.2 \\n");
2 printf("\\t approximate values are mentioned in the
   book \\n");
```

```

3 T1=1000+460; // R
4 T2=800+460; // R
5 e1=0.6; // emissivity of hotter wall
6 e2=0.8; // emissivity of colder wall
7 Q=((0.173)/((1/0.6)+(1/0.8)-1))*((14.6)^4-(12.6)^4)
   ); // using eq.4.26,heat loss per unit area ,Btu/(
   hr)*(ft^2)
8 printf("\t heat removed from colder wall per unit
   area is : %.0f Btu/(hr)*(ft^2) \n",Q);
9 printf("\t For perfect black bodies the value was
   3500 Btu/(hr)(ft^2) \n");
10 // end

```

Scilab code Exa 4.3 Calculation of Radiation from a Pipe

```

1 clc
2 //page 78
3 printf("\t example 4.3 \n");
4 printf("\t approximate values are mentioned in the
   book \n");
5 T1=125+460; // R
6 T2=70+460; // R
7 e=0.9; // emissivity ,using table 4.1B
8 A=(%pi)*(3.375/12)*(1); // area ,ft^2/lin ft
9 printf("\t area is : %.2f ft^2/lin ft \n",A);
10 Q=(0.9)*(0.88)*(0.173)*((T1/100)^4-(T2/100)^4); //
   heat loss using eq.4.32,Btu/(hr)*(lin ft)
11 printf("\t heat loss is : %.1f Btu/(hr)*(lin ft) \n
   ",Q);
12 hr=(Q)/((A)*(T1-T2)); // fictitious film coefficient
   , using eq 4.33,Btu/(hr)(ft^2)(F)
13 printf("\t fictitious film coefficient is : %.2f Btu
   /(hr)(ft^2)(F) \n",hr);
14 //end

```

Scilab code Exa 4.4 Radiation from a Pipe to a Duct

```
1  clc
2  //page 82
3  printf("\t example 4.4 \n");
4  printf("\t approximate values are mentioned in the
      book \n");
5  T1=300+460; // R
6  T2=75+460; //R
7  A1=0.622; // area from table 11 in the appendix A,ft
      ^2/lin ft
8  A2=4*(1*1); // surface area of duct,ft^2/lin ft
9  e1=0.79; // emissivity of oxidized steel from table
      4.1
10 e2=0.276; // emissivity of oxidized zinc from table
      4.1
11 printf("\t surface area of pipe is : %.3f ft^2/lin
      ft \n",A1);
12 printf("\t surface area of duct is : %.0f ft^2/lin
      ft \n",A2);
13 printf("\t The surface of the pipe is not negligible
      by comparison with that of the duct, and(f) of
      Table 4.2 applies most nearly \n");
14 Fa=1; // from table 4.2
15 Fe=((1)/((1/e1)+((A1/A2)*((1/e2)-1))))); // from
      table 4.2
16 printf("\t Fe is : %.2f \n",Fe);
17 Q=(0.173*10^-8)*(Fa)*(Fe)*(A1)*((T1)^4-(T2)^4); //
      heat loss due to radiation ,Btu/(hr)*(lin ft)
18 printf("\t heat loss due to radiation is : %.0f Btu
      /(hr)*(lin ft) \n",Q);
19 // end
```

Chapter 5

Temperature

Scilab code Exa 5.1 Calculation of the LMTD

```
1 printf("\t example 5.1 \n");
2 T1=300; // hot fluid inlet temperature ,F
3 T2=200; // hot fluid outlet temperature ,F
4 t1=100; // cold fluid inlet temperature ,F
5 t2=150; // cold fluid outlet temperature ,F
6 printf("\t for counter current flow \n");
7 delT1=T1-t2; //F
8 delT2=T2-t1; // F
9 printf("\t delT1 is : %.0f F \n",delT1);
10 printf("\t delT2 is : %.0f F \n",delT2);
11 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
12 printf("\t LMTD is :%.1f F \n",LMTD);
13 printf("\t for parallel flow \n");
14 delT1=T1-t1; // F
15 delT2=T2-t2; // F
16 printf("\t delT1 is : %.0f F \n",delT1);
17 printf("\t delT2 is : %.0f F \n",delT2);
18 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
19 printf("\t LMTD is :%.0f F \n",LMTD);
20 //end
```

Scilab code Exa 5.2 Calculation of the LMTD with Equal Outlet Temperatures

```
1 printf("\t example 5.2 \n");
2 T1=300; // hot fluid inlet temperature ,F
3 T2=200; // hot fluid outlet temperature ,F
4 t1=150; // cold fluid inlet temperature ,F
5 t2=200; // cold fluid outlet temperature ,F
6 printf("\t for counter current flow \n");
7 delT1=T1-t2; //F
8 delT2=T2-t1; // F
9 printf("\t delT1 is : %.0f F \n",delT1);
10 printf("\t delT2 is : %.0f F \n",delT2);
11 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
12 printf("\t LMTD is :%.0f F \n",LMTD);
13 printf("\t for parallel flow \n");
14 delT1=T1-t1; // F
15 delT2=T2-t2; // F
16 printf("\t delT1 is : %.0f F \n",delT1);
17 printf("\t delT2 is : %.0f F \n",delT2);
18 if(delT2==0);
19     printf("\t denominator becomes infinity so LMTD
20         becomes Zero \n");
21     printf("\t LMTD is Zero \n");
21 else
22     LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1)))
23         );
23 printf("\t LMTD is :%.0f F \n",LMTD);
24     end
25 //end
```

Scilab code Exa 5.3 Calculation of the LMTD

```

1 printf("\t example 5.3 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=300; // hot fluid inlet temperature ,F
4 T2=200; // hot fluid outlet temperature ,F
5 t1=100; // cold fluid inlet temperature ,F
6 t2=275; // cold fluid outlet temperature ,F
7 printf("\t for counter current flow \n");
8 deltc=T2-t1; //F
9 delth=T1-t2; // F
10 printf("\t delth is : %.0f F \n",delth);
11 printf("\t deltc is : %.0f F \n",deltc);
12 LMTD=((delth-deltc)/((2.3)*(log10(delth/deltc))));
13 printf("\t LMTD is :%.1f F \n",LMTD);
14 //end

```

Scilab code Exa 5.4 Calculation of the LMTD with One Isothermal Fluid

```

1 printf("\t example 5.4 \n");
2 printf("\t process is isothermal with hot fluid so
   temperature of hot fluid remains constant \n");
3 T1=300; // hot fluid inlet temperature ,F
4 T2=300; // hot fluid outlet temperature ,F
5 t1=100; // cold fluid inlet temperature ,F
6 t2=275; // cold fluid outlet temperature ,F
7 printf("\t for counter current flow \n");
8 delt1=T1-t2; //F
9 delt2=T2-t1; // F
10 printf("\t delt1 is : %.0f F \n",delt1);
11 printf("\t delt2 is : %.0f F \n",delt2);
12 LMTD=((delt2-delt1)/((2.3)*(log10(delt2/delt1))));
13 printf("\t LMTD is :%.0f F \n",LMTD);
14 printf("\t for parallel flow \n");
15 delt1=T1-t1; // F
16 delt2=T2-t2; // F

```

```

17 printf("\t delt1 is : %.0f F \n",delt1);
18 printf("\t delt2 is : %.0f F \n",delt2);
19 if(delt2==0);
20     printf("\t denominator becomes infinity so LMTD
           becomes Zero \n");
21     printf("\t LMTD is Zero \n");
22 else
23     LMTD=((delt2-delt1)/((2.3)*(log10(delt2/delt1)))
           );
24 printf("\t LMTD is :%.0f F \n",LMTD);
25     end
26 printf("\t these are identical \n");
27 //end

```

Scilab code Exa 5.5 Calculation of point

```

1 printf("\t example 5.5 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 printf("\t for inlet \n");
4 t1=99.1; // temperature of inlet ,F
5 t2=129.2; // temperature of outlet ,F
6 c=.478; // Btu/(hr)*(ft)*(F/ft)
7 mu=2.95*2.42; // lb/(ft)(hr)
8 k=0.078; // Btu/(hr)*(ft)*(F/ft)
9 G=854000; // mass velocity ,lb/(ft^2)(hr)
10 D=0.622/12; // diameter ,ft
11 Re=((D)*((G)/(mu)))^(0.9);
12 printf("\t Re is : %.2e \n",Re);
13 Pr=((c)*(mu)/k)^(1/3); // prandtl number raised to
   power 1/3
14 printf("\t Pr is : %.2f \n",Pr);
15 Nu=0.0115*(Re)*(Pr); // formula for nusselt number
16 printf("\t nusselt number is : %.0f \n",Nu);
17 hi=((k)*(Nu)/(D)); // heat transfer coefficient

```

```

18 printf("\t heat transfer coefficient is : %.0f \n",
    hi); // caculation mistake in book
19 printf("\t for outlet \n");
20 c=.495; // Btu/(hr)*(ft)*(F/ft)
21 mu=2.20*2.42; // lb/(ft)(hr)
22 k=0.078; // Btu/(hr)*(ft)*(F/ft)
23 G=854000; // mass velocity ,lb/(ft^2)(hr)
24 D=0.622/12; // diameter ,ft
25 Re=((D)*((G)/(mu)))^(.9); // reynolds number raised
    to poer 0.9, calculation mistake in book
26 printf("\t Re is : %.2e \n",Re);
27 Pr=((c)*(mu)/k)^(1/3); // prandtl number raised to
    power 1/3
28 printf("\t Pr is : %.2f \n",Pr);
29 Nu=0.0115*(Re)*(Pr); // formula for nusselt number
30 printf("\t nusselt number is : %.0f \n",Nu);
31 hi=((k)*(Nu)/(D)); // heat transfer coefficient
32 printf("\t heat transfer coefficient is : %.0f \n",
    hi); // caculation mistake in book
33 //end

```

Scilab code Exa 5.6 Calculation of the Caloric Temperature

```

1 printf("\t example 5.6 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=300; // hot fluid inlet temperature ,F
4 T2=200; // hot fluid outlet temperature ,F
5 t1=80; // cold fluid inlet temperature ,F
6 t2=120; // cold fluid outlet temperature ,F
7 printf("\t for counter current flow \n");
8 delT=T1-T2; // temperature difference for crude oil ,
    F
9 printf("\t temperature difference for crude oil is :
    %.0f F \n",delT);

```

```

10 Kc=0.68; // from fig.17
11 delT=t2-t1; // temperature difference for gasoline ,F
12 printf("\t temperature difference for gasoline is :
    %.0f F \n",delT);
13 Kc<=0.10; // from fig.17
14 printf("\t The larger value of K. correspQnds to the
    controlling heat transfer coefficient which is
    assumed to establish the variation of U with
    temperature \n");
15 delTc=T2-t1; //F
16 delTh=T1-t2; // F
17 printf("\t delTc is : %.0f F \n",delTc);
18 printf("\t delTh is : %.0f F \n",delTh);
19 A=((delTc)/(delTh));
20 printf("\t ratio of two local temperature difference
    is : %.3f \n",A);
21 Fc=0.425; // from fig.17
22 Tc=((T2)+((Fc)*(T1-T2))); // caloric temperature of
    hot fluid ,F
23 printf("\t caloric temperature of hot fluid is : %.1
    f F \n",Tc);
24 tc=((t1)+((Fc)*(t2-t1))); // caloric temperature of
    cold fluid ,F
25 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
26 // end

```

Chapter 6

Counterflow Double pipe exchangers

Scilab code Exa 6.1 Double Pipe Benzene Toluene Exchanger

```
1 printf("\t example 6.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=160; // inlet hot fluid ,F
4 T2=100; // outlet hot fluid ,F
5 t1=80; // inlet cold fluid ,F
6 t2=120; // outlet cold fluid ,F
7 w=9820; // lb/hr
8 printf("\t 1.for heat balance \n");
9 printf("\t for benzene \n");
10 tav=((t1+t2)/2); // F
11 printf("\t average temperature of benzene is : %.0f
   F \n",tav);
12 c=0.425; // Btu/(lb)*(F)
13 Q=((w)*(c)*(t2-t1)); // Btu/hr
14 printf("\t total heat required for benzene is : %.2e
   Btu/hr \n",Q);
15 printf("\t for toulene \n");
16 Tav=((T1+T2)/2); //F
```



```

17 printf("\t average temperature of toluene is : %.0f
    F \n",Tav);
18 c=0.44; // Btu/(lb)*(F)
19 W=((Q)/((c)*(T1-T2))); // lb/hr
20 printf("\t W is :%.2e lb/hr \n",W);
21 printf("\t 2.LMTD \n");
22 printf("\t for counter current flow \n");
23 deltt1=T2-t1; //F
24 deltt2=T1-t2; // F
25 printf("\t deltt1 is : %.0f F \n",deltt1);
26 printf("\t deltt2 is : %.0f F \n",deltt2);
27 LMTD=((deltt2-deltt1)/((2.3)*(log10(deltt2/deltt1))));
28 printf("\t LMTD is :%.1f F \n",LMTD);
29 printf("\t 3.caloric temperatures \n");
30 printf("\t both streams will show that neither is
    viscous at the cold terminal (the viscosities
    less than 1 centipoise) and the temperature
    ranges and temperature difference are moderate.
    The coefficients may accordingly be evaluated
    from properties at the arithmetic mean, and the
    value of (mu/muw)^0.14 may be assumed equal to
    1.0 \n");
31 tav=((t1+t2)/2); // F
32 printf("\t average temperature of benzene is : %.0f
    F \n",tav);
33 Tav=((T1+T2)/2); //F
34 printf("\t average temperature of toluene is : %.0f
    F \n",Tav);
35 printf("\t hot fluid:annulus ,toluene \n");
36 D1=0.138; // ft
37 D2=0.1725; // ft
38 aa=((%pi)*(D2^2-D1^2)/4); // flow area ,ft^2
39 printf("\t flow area is : %.5f ft^2 \n",aa);
40 De=(D2^2-D1^2)/D1; // equiv diameter ,ft
41 printf("\t equiv diameter is : %.4f ft \n",De);
42 Ga=(W/aa); // mass velocity ,lb/(hr)*(ft^2)
43 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Ga);

```

```

44 mu1=0.41*2.42; // at 130 F, lb/(ft)*(hr)
45 Rea=((De)*(Ga)/mu1); // reynolds number
46 printf("\t reynolds number is : %.1e \n",Rea);
47 jH=167; // from fig.24
48 c=0.44; // Btu/(lb)*(F), at 130F
49 k=0.085; // Btu/(hr)*(ft^2)*(F/ft), from table 4
50 Pr=((c)*(mu1)/k)^(1/3); // prandelt number raised to
    power 1/3
51 printf("\t Pr is : %.3f \n",Pr);
52 ho=((jH)*(k/De)*(Pr)*(1^0.14)); // using eq.6.15b,
    Btu/(hr)*(ft^2)*(F)
53 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
54 printf("\t cold fluid:inner pipe,benzene \n");
55 D=0.115; // ft
56 ap=((%pi)*(D^2)/4); // flow area , ft^2
57 printf("\t flow area is : %.4f ft^2 \n",ap);
58 Gp=(w/ap); // mass velocity ,lb/(hr)*(ft^2)
59 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gp);
60 mu2=0.5*2.42; // at 130 F, lb/(ft)*(hr)
61 Rep=((D)*(Gp)/mu2); // reynolds number
62 printf("\t reynolds number is : %.2e \n",Rep);
63 jH=236; // from fig.24
64 c=0.425; // Btu/(lb)*(F), at 130F
65 k=0.091; // Btu/(hr)*(ft^2)*(F/ft), from table 4
66 Pr=((c)*(mu2)/k)^(1/3); // prandelt number raised to
    power 1/3
67 printf("\t Pr is : %.3f \n",Pr);
68 hi=((jH)*(k/D)*(Pr)*(1^0.14)); // using eq.6.15a, Btu
    /(hr)*(ft^2)*(F)
69 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
70 ID=1.38; // ft
71 OD=1.66; //ft
72 hio=((hi)*(ID/OD)); // using eq.6.5
73 printf("\t Correct hi to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);

```

```

74 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
75 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
76 Rd=0.002; // required by problem ,(hr)*(ft^2)*(F)/Btu
77 UD=((Uc)/((1)+(Uc*Rd))); // design overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
78 printf("\t design overall coefficient is : %.0f Btu
    /(hr)*(ft^2)*(F) \n",UD);
79 A=((Q)/((UD)*(LMTD))); // required surface ,ft^2
80 printf("\t required surface is : %.1f ft^2 \n",A);
81 A1=0.435; // From Table 11 for 1(1/4)in IPS standard
    pipe there are 0.435 ft2 of external surface per
    foot length ,ft^2
82 L=(A/A1); // required length;lin ft
83 printf("\t required length is : %.0f lin ft \n",L);
84 printf("\t This may be fulfilled by connecting three
    20-ft hairpins in series \n");
85 A2=120*0.435; // actual surface supplied ,ft^2
86 printf("\t actual surface supplied is : %.1f ft^2 \n
    ",A2);
87 UD=((Q)/((A2)*(LMTD)));
88 printf("\t actual design overall coefficient is : %
    .0f Btu/(hr)*(ft^2)*(F) \n",UD);
89 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
90 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
91 printf("\t pressure drop for annulus \n");
92 De1=(D2-D1); //ft
93 printf("\t De1 is : %.4f ft \n",De1);
94 Rea1=((De1)*(Ga)/mu1); // reynolds number
95 printf("\t reynolds number is : %.2e \n",Rea1);
96 f=(0.0035)+((0.264)/(Rea1^0.42)); // friction factor
    , using eq.3.47b
97 printf("\t friction factor is : %.4f \n",f);
98 s=0.87;
99 row=62.5*0.87; // from table 6
100 delFa=((4*f*(Ga^2)*L)/(2*4.18*(10^8)*(row^2)*(De1)))

```

```

    ; // ft
101 printf("\t delFa is : %.1f ft \n",delFa);
102 V=((Ga)/(3600*row)); //fps
103 printf("\t V is : %.2f fps \n",V);
104 F1=((3*(V^2))/(2*32.2)); //ft
105 printf("\t F1 is : %.1f ft \n",F1);
106 delPa=((delFa+F1)*(row)/144); // psi
107 printf("\t delPa is : %.1f psi \n",delPa);
108 printf("\t allowable delPa is 10 psi \n");
109 printf("\t pressure drop for inner pipe \n");
110 f=(0.0035)+((0.264)/(Rep^0.42)); // friction factor ,
    using eq.3.47b
111 printf("\t friction factor is : %.4f \n",f);
112 s=0.88;
113 row=62.5*0.88; // from table 6
114 delFp=((4*f*(Gp^2)*L)/(2*4.18*(10^8)*(row^2)*(D)));
    // ft
115 printf("\t delFp is : %.1f ft \n",delFp);
116 delPp=((delFp)*(row)/144); // psi
117 printf("\t delPp is : %.1f psi \n",delPp);
118 printf("\t allowable delPp is 10 psi \n");
119 //end

```

Scilab code Exa 6.2 Calculation of the True Temperature Difference

```

1 printf("\t example 6.2 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=300; // inlet hot fluid ,F
4 T2=200; // outlet hot fluid ,F
5 t1=190; // inlet cold fluid ,F
6 t2=220; // outlet cold fluid ,F
7 n=6; // number of parallel streams
8 P=((T2-t1)/(T1-t1));
9 printf("\t P is : %.3f \n",P);

```

```

10 R=((T1-T2)/((n)*(t2-t1)));
11 printf("\t R is : %.3f \n",R);
12 gama=((1-P)/((2.3)*((n*R)/(R-1))*log10(((R-1)/R)*(1/
    P)^(1/n)+(1/R)))); // using eq.6.35a
13 printf("\t gama is : %.3f \n",gama);
14 delT=(gama*(T1-t1)); // true temperature difference ,
    F
15 printf("\ true temperature difference is : %.1f F \n
    ",delT);
16 //end

```

Scilab code Exa 6.3 Double Pipe Lube Oil Crude Oil Exchanger

```

1 printf("\t example 6.3 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=450; // inlet hot fluid ,F
4 T2=350; // outlet hot fluid ,F
5 t1=300; // inlet cold fluid ,F
6 t2=310; // outlet cold fluid ,F
7 W=6900; // lb/hr
8 w=72500; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for lube oil \n");
11 c=0.62; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for lube oil is : %.2
    e Btu/hr \n",Q);
14 printf("\t for crude oil \n");
15 c=0.585; // Btu/(lb)*(F)
16 Q1=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for crude oil is : %
    .2e Btu/hr \n",Q1); // calculation mistake in
    book
18 delT1=T2-t1; //F

```

```

19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 A=((delT1)/(delT2));
25 printf("\t ratio of two local temperature difference
      is : %.3f \n",A);
26 Fc=0.395; // from fig.17
27 Tc=((T2)+((Fc)*(T1-T2))); // caloric temperature of
      hot fluid ,F
28 printf("\t caloric temperature of hot fluid is : %.1
      f F \n",Tc);
29 tc=((t1)+((Fc)*(t2-t1))); // caloric temperature of
      cold fluid ,F
30 printf("\t caloric temperature of cold fluid is : %
      .0f F \n",tc);
31 printf("\t hot fluid:annulus,lube oil \n");
32 D1=0.199; // ft
33 D2=0.256; // ft
34 aa=((%pi)*(D2^2-D1^2)/4); // flow area ,ft^2
35 printf("\t flow area is : %.4f ft^2 \n",aa);
36 De=(D2^2-D1^2)/D1; // equiv diameter ,ft
37 printf("\t equiv diameter is : %.2f ft \n",De);
38 Ga=(W/aa); // mass velocity ,lb/(hr)*(ft^2)
39 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
      ",Ga);
40 mu1=3*2.42; // at 389.5F,lb/(ft)*(hr), from fig.14
41 Rea=((De)*(Ga)/mu1); // reynolds number
42 printf("\t reynolds number is : %.0e \n",Rea);
43 jH=20.5; // from fig.24
44 c=0.615; // Btu/(lb)*(F),at 130F
45 k=0.067; // Btu/(hr)*(ft^2)*(F/ft), from table 4
46 Pr=((c)*(mu1)/k)^(1/3); // prandelt number raised to
      power 1/3
47 printf("\t Pr is : %.3f \n",Pr);
48 Ho=((jH)*(k/De)*(Pr)); // H0=(h0/phyA),using eq
      .6.15 ,Btu/(hr)*(ft^2)*(F)

```

```

49 printf("\t individual heat transfer coefficient is :
      %.1f Btu/(hr)*(ft^2)*(F) \n",Ho);
50 printf("\t cold fluid:inner pipe,crude oil \n");
51 D=0.172; // ft
52 ap=((%pi)*(D^2)/4); // flow area, ft^2
53 printf("\t flow area is : %.4f ft^2 \n",ap);
54 Gp=(w/(2*ap)); // mass velocity, lb/(hr)*(ft^2)
55 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
      ",Gp);
56 mu2=0.83*2.42; // at 304 F, lb/(ft)*(hr)
57 Rep=((D)*(Gp)/mu2); // reynolds number
58 printf("\t reynolds number is : %.2e \n",Rep);
59 jH=320; // from fig.24
60 c=0.585; // Btu/(lb)*(F), at 304F, from fig.4
61 k=0.073; // Btu/(hr)*(ft^2)*(F/ft), from fig.1
62 Pr=((c)*(mu2)/k)^(1/3); // prandelt number raised to
      power 1/3
63 printf("\t Pr is : %.3f \n",Pr);
64 Hi=((jH)*(k/D)*(Pr)*(1^0.14)); //Hi=(hi/phyph), using
      eq.6.15a, Btu/(hr)*(ft^2)*(F)
65 printf("\t Hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",Hi);
66 ID=2.067; // ft
67 OD=2.38; //ft
68 Hio=((Hi)*(ID/OD)); //Hio=(hio/phyph), using eq.6.5
69 printf("\t Correct Hi0 to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",Hio);
70 muw=0.77*2.42; // lb/(ft)*(hr), from fig.14
71 phyp=(mu2/muw)^0.14;
72 printf("\t phyp is : %.0f \n",phyp); // from fig.24
73 hio=(Hio)*(1); // from eq.6.37
74 printf("\t Correct hio to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
75 tw=(tc)+(((Ho)/(Hio+Ho))*(Tc-tc)); // from eq.5.31
76 printf("\t tw is : %.0f F \n",tw);
77 muw=6.6*2.42; // lb/(ft)*(hr), from fig.14
78 phya=(mu1/muw)^0.14;
79 printf("\t phya is : %.1f \n",phya); // from fig.24
80 ho=(Ho)*(phya); // from eq.6.36

```

```

81 printf("\t Correct h0 to the surface at the OD is :
      %.1f Btu/(hr)*(ft^2)*(F) \n",ho);
82 Uc=((hio)*(ho)/(hio+ho)); // clean overall
      coefficient ,Btu/(hr)*(ft^2)*(F)
83 printf("\t clean overall coefficient is : %.0f Btu/(
      hr)*(ft^2)*(F) \n",Uc);
84 Rd=0.006; // required by problem ,(hr)*(ft^2)*(F)/Btu
85 UD=((Uc)/((1)+(Uc*Rd))); // design overall
      coefficient ,Btu/(hr)*(ft^2)*(F)
86 printf("\t design overall coefficient is : %.1f Btu
      /(hr)*(ft^2)*(F) \n",UD);
87 A=((Q)/((UD)*(LMTD))); // required surface ,ft^2
88 printf("\t required surface is : %.0f ft^2 \n",A);
89 A1=0.622; // From Table 11,ft^2
90 Lr=(A/A1); // required length;lin ft
91 printf("\t required length is : %.0f lin ft \n",Lr);
92 printf("\t Since two parallel streams are employed,
      use eight 20 ft hairpins or 320 lin. feet \n");
93 L=320;
94 A2=320*0.622; // actual surface supplied ,ft^2
95 printf("\t actual surface supplied is : %.1f ft^2 \n
      ",A2);
96 UD=((Q)/((A2)*(LMTD)));
97 printf("\t actual design overall coefficient is : %
      .1f Btu/(hr)*(ft^2)*(F) \n",UD);
98 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
99 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
      n",Rd);
100 printf("\t pressure drop for annulus \n");
101 De1=.058; //ft
102 printf("\t De1 is : %.3f ft \n",De1);
103 Rea1=((De1)*(Ga)/7.25); // reynolds number
104 printf("\t reynolds number is : %.2e \n",Rea1);
105 f=(0.0035)+((0.264)/(2680^0.42)); // friction factor
      , using eq.3.47b
106 printf("\t friction factor is : %.4f \n",f);
107 s=0.775;
108 row=62.5*0.775; // from fig 6

```



```

109 delFa=((4*f*(Ga^2)*L)/(2*4.18*(10^8)*(row^2)*(De1)))
    ; // ft
110 printf("\t delFa is : %.1f ft \n",delFa);
111 V=((Ga)/(3600*row)); //fps
112 printf("\t V is : %.1f fps \n",V);
113 delF1=((8*(V^2))/(2*32.2)); //ft
114 printf("\t delF1 is : %.2f ft \n",delF1);
115 delPa=((delFa+delF1)*(row)/144); // psi
116 printf("\t delPa is : %.1f psi \n",delPa);
117 printf("\t allowable delPa is 10 psi \n");
118 printf("\t pressure drop for inner pipe \n");
119 f=(0.0035)+((0.264)/(Rep^0.42)); // friction factor ,
    using eq.3.47b
120 printf("\t friction factor is : %.5f \n",f);
121 s=0.76;
122 row=62.5*0.76; // from table 6
123 Lp=160;
124 delFp=((4*f*(Gp^2)*Lp)/(2*4.18*(10^8)*(row^2)*(D)));
    // ft
125 printf("\t delFp is : %.1f ft \n",delFp);
126 delPp=((delFp)*(row)/144); // psi
127 printf("\t delPp is : %.1f psi \n",delPp);
128 printf("\t allowable delPp is 10 psi \n");
129 // end

```

Chapter 7

Parallel Counterflow Shell and Tube Excahngers

Scilab code Exa 7.1 shell side equivalent

```
1 printf("\t example 7.1 \n");
2 PT=1; // square pitch ,in
3 do=0.75; // outer diameter ,in
4 de=((4*(PT^2-(3.14*do^2/4)))/(3.14*do));
5 printf("\t equivalent diameter is : %.2f in \n",de);
6 De=(de/12); // ft
7 printf("\t De is : %.3f in \n",De);
8 //end
```

Scilab code Exa 7.2 Calculation of FT for Fluids with Equal Ranges

```
1 printf("\t example 7.2 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 printf("\t considering 50F approach \n");
4 T1=350; //F
```

```

5 T2=250; //F
6 t2=T2-50; // formula for approach ,f
7 printf("\t t2 is : %.0f F \n",t2);
8 printf("\t fluids are with equal ranges ,so \n");
9 t1=t2-(T1-T2); // F
10 printf("\t t1 is : %.0f F \n",t1);
11 R=((T1-T2)/(t2-t1));
12 printf("\t R is : %.0f \n",R);
13 S=((t2-t1)/(T1-t1));
14 printf("\t S is : %.2f \n",S);
15 printf("\t FT is 0.925 \n"); // from fig 18
16 printf("\t considering 0F approach \n");
17 T1=300; //F
18 T2=200; //F
19 t2=T2-0; // formula for approach ,f
20 printf("\t t2 is : %.0f F \n",t2);
21 printf("\t fluids are with equal ranges ,so \n");
22 t1=t2-(T1-T2); // F
23 printf("\t t1 is : %.0f F \n",t1);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.0f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.2f \n",S);
28 printf("\t FT is 0.80 \n"); // from fig 18
29 printf("\t considering 20F cross \n");
30 T1=280; //F
31 T2=180; //F
32 t2=T2+20; // formula for approach ,f
33 printf("\t t2 is : %.0f F \n",t2);
34 printf("\t fluids are with equal ranges ,so \n");
35 t1=t2-(T1-T2); // F
36 printf("\t t1 is : %.0f F \n",t1);
37 R=((T1-T2)/(t2-t1));
38 printf("\t R is : %.0f \n",R);
39 S=((t2-t1)/(T1-t1));
40 printf("\t S is : %.3f \n",S);
41 printf("\t FT is 0.64 \n"); // from fig 18
42 //end

```

Scilab code Exa 7.3 Calculation of a Kerosene Crude Oil Exchanger

```
1 printf("\t example 7.3 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=390; // inlet hot fluid ,F
4 T2=200; // outlet hot fluid ,F
5 t1=100; // inlet cold fluid ,F
6 t2=170; // outlet cold fluid ,F
7 W=43800; // lb/hr
8 w=149000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for kerosene \n");
11 c=0.605; // Btu/(lb)*(F)
12 Q1=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for kerosene is : %.1
   e Btu/hr \n",Q1); // calculation mistake in
   problem
14 printf("\t for crude oil \n");
15 c=0.49; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for mid continent
   crude is : %.1e Btu/hr \n",Q); // calculation
   mistake in problem
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.2f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.3f \n",S);
```

```

28 printf("\t FT is 0.905 \n"); // from fig 18
29 delT=(0.905*LMTD); // F
30 printf("\t delT is : %.0f F \n",delT);
31 X=((delT1)/(delT2));
32 printf("\t ratio of two local temperature difference
    is : %.3f \n",X);
33 Fc=0.42; // from fig.17
34 Kc=0.20; // crude oil controlling
35 Tc=((T2)+((Fc)*(T1-T2))); // caloric temperature of
    hot fluid ,F
36 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
37 tc=((t1)+((Fc)*(t2-t1))); // caloric temperature of
    cold fluid ,F
38 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
39 printf("\t hot fluid:shell side ,kerosene \n");
40 ID=21.25; // in
41 C=0.25; // clearance
42 B=5; // baffle spacing ,in
43 PT=1.25;
44 as=((ID*C*B)/(144*PT)); // flow area ,ft^2
45 printf("\t flow area is : %.4f ft^2 \n",as);
46 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2)
47 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
48 mu1=0.40*2.42; // at 280F,lb/(ft)*(hr) , from fig.14
49 De=0.99/12; // from fig.28 ,ft
50 Res=((De)*(Gs)/mu1); // reynolds number
51 printf("\t reynolds number is : %.2e \n",Res);
52 jH=93; // from fig.28
53 c=0.59; // Btu/(lb)*(F) ,at 280F,from fig.4
54 k=0.0765; // Btu/(hr)*(ft^2)*(F/ft) , from fig.1
55 Pr=((c)*(mu1)/k)^(1/3); // prandelt number raised to
    power 1/3
56 printf("\t Pr is : %.3f \n",Pr);
57 Ho=((jH)*(k/De)*(Pr)); // H0=(h0/phyA) ,using eq
    .6.15 ,Btu/(hr)*(ft^2)*(F)

```

```

58 printf("\t individual heat transfer coefficient is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",Ho);
59 printf("\t cold fluid:inner tube side ,crude oil \n")
    ;
60 D=0.0675; // ft
61 Nt=158;
62 n=4; // number of passes
63 L=16; //ft
64 at1=0.515; // flow area , in^2
65 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
66 printf("\t flow area is : %.3f ft^2 \n",at);
67 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
68 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
69 mu2=3.6*2.42; // at 129F,lb/(ft)*(hr)
70 Ret=((D)*(Gt)/mu2); // reynolds number
71 printf("\t reynolds number is : %.2e \n",Ret);
72 jH=31; // from fig.24
73 c=0.49; // Btu/(lb)*(F),at 304F,from fig.4
74 k=0.077; // Btu/(hr)*(ft^2)*(F/ft), from fig.1
75 Pr=((c)*(mu2)/k)^(1/3); // prandelt number raised to
    power 1/3
76 printf("\t Pr is : %.3f \n",Pr);
77 Hi=((jH)*(k/D)*(Pr)*(1^0.14)); //Hi=(hi/phyt), using
    eq.6.15a,Btu/(hr)*(ft^2)*(F)
78 printf("\t Hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",Hi);
79 ID=0.81; // ft
80 OD=1; //ft
81 Hio=((Hi)*(ID/OD)); //Hio=(hio/phyt), using eq.6.5
82 printf("\t Correct Hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Hio);
83 muw=1.5*2.42; // lb/(ft)*(hr), from fig.14
84 phyt=(mu2/muw)^0.14;
85 printf("\t phyt is : %.2f \n",phyt); // from fig.24
86 hio=(Hio)*(phyt); // from eq.6.37
87 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);

```

```

88 tw=(tc)+(((Ho)/(Hio+Ho))*(Tc-tc)); // from eq.5.31
89 printf("\t tw is : %.0f F \n",tw);
90 muw=0.56*2.42; // lb/(ft)*(hr), from fig.14
91 phys=(mu1/muw)^0.14;
92 printf("\t phys is : %.2f \n",phys); // from fig.24
93 ho=(Ho)*(phys); // from eq.6.36
94 printf("\t Correct h0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
95 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
96 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
97 A2=0.2618; // actual surface supplied for each tube,
    ft^2,from table 10
98 A=(Nt*L*A2); // ft^2
99 printf("\t total surface area is : %.0f ft^2 \n",A);
100 UD=((Q)/((A)*(delt)));
101 printf("\t actual design overall coefficient is : %
    .1f Btu/(hr)*(ft^2)*(F) \n",UD);
102 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
103 printf("\t actual Rd is : %.5f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
104 printf("\t pressure drop for annulus \n");
105 f=0.00175; // friction factor for reynolds number
    25300, using fig.29
106 s=0.73; // for reynolds number 25300,using fig.6
107 Ds=21.25/12; // ft
108 N=(12*L/B); // number of crosses ,using eq.7.43
109 printf("\t number of crosses are : %.0f \n",N);
110 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys))); // using eq.7.44 ,psi
111 printf("\t delPs is : %.1f psi \n",delPs);
112 printf("\t allowable delPa is 10 psi \n");
113 printf("\t pressure drop for inner pipe \n");
114 f=0.000285; // friction factor for reynolds number
    8220, using fig.26
115 s=0.83;
116 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(

```

```

        phyt))); // using eq.7.45 , psi
117 printf("\t delPt is : %.1f psi \n",delPt);
118 X1=0.15; // X1=((V^2)/(2*g)), for Gt 1060000, using
        fig.27
119 delPr=((4*n*X1)/(s)); // using eq.7.46 , psi
120 printf("\t delPr is : %.1f psi \n",delPr);
121 delPT=delPt+delPr; // using eq.7.47 , psi
122 printf("\t delPT is : %.1f psi \n",delPT);
123 printf("\t allowable delPs is 10 psi \n");
124 //end

```

Scilab code Exa 7.4 Calculation of a Distilled water Raw water Exchanger

```

1 printf("\t example 7.4 \n");
2 printf("\t approximate values are mentioned in the
        book \n");
3 T1=93; // inlet hot fluid ,F
4 T2=85; // outlet hot fluid ,F
5 t1=75; // inlet cold fluid ,F
6 t2=80; // outlet cold fluid ,F
7 W=175000; // lb/hr
8 w=280000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for distilled water \n");
11 c=1; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for distilled water
        is : %.1e Btu/hr \n",Q);
14 printf("\t for raw water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for raw water is : %
        .1e Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F

```



```

20 printf("\t delt1 is : %.0f F \n",delt1);
21 printf("\t delt2 is : %.0f F \n",delt2);
22 LMTD=((delt2-delt1)/((2.3)*(log10(delt2/delt1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.2f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.3f \n",S);
28 printf("\t FT is 0.945 \n"); // from fig 18
29 delt=(0.945*LMTD); // F
30 printf("\t delt is : %.2f F \n",delt);
31 X=((delt1)/(delt2));
32 printf("\t ratio of two local temperature difference
      is : %.3f \n",X);
33 Fc=0.42; // from fig.17
34 Kc=0.20; // crude oil controlling
35 Tc=((T2)+(T1))/2; // caloric temperature of hot
      fluid ,F
36 printf("\t caloric temperature of hot fluid is : %.0
      f F \n",Tc);
37 tc=((t1)+(t2))/2; // caloric temperature of cold
      fluid ,F
38 printf("\t caloric temperature of cold fluid is : %
      .1f F \n",tc);
39 printf("\t hot fluid:shell side ,distilled water \n")
      ;
40 ID=15.25; // in
41 C=0.1875; // clearance
42 B=12; // baffle spacing ,in
43 PT=0.9375;
44 as=((ID*C*B)/(144*PT)); // flow area ,ft ^2,using eq
      .7.1
45 printf("\t flow area is : %.3f ft ^2 \n",as);
46 Gs=(W/as); // mass velocity ,lb/(hr)*(ft ^2),using eq
      .7.2
47 printf("\t mass velocity is : %.1e lb/(hr)*(ft ^2) \n
      ",Gs);
48 mu1=0.81*2.42; // at 89F,lb/(ft)*(hr), from fig.14

```

```

49 De=0.55/12; // from fig.28,ft
50 Res=((De)*(Gs)/mu1); // reynolds number
51 printf("\t reynolds number is : %.2e \n",Res);
52 jH=73; // from fig.28
53 c=1; // Btu/(lb)*(F),at 89F,from fig.table 4
54 k=0.36; // Btu/(hr)*(ft^2)*(F/ft), from table 4
55 Pr=((c)*(mu1)/k)^(1/3); // prandelt number raised to
    power 1/3
56 printf("\t Pr is : %.3f \n",Pr);
57 ho=((jH)*(k/De)*(Pr)); // using eq.6.15 ,Btu/(hr)*(ft
    ^2)*(F)
58 printf("\t individual heat transfer coefficient is :
    %.2e Btu/(hr)*(ft^2)*(F) \n",ho);
59 printf("\t cold fluid:inner tube side ,raw water \n")
    ;
60 Nt=160;
61 n=2; // number of passes
62 L=16; //ft
63 at1=0.334; // flow area , in^2,from table 10
64 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
65 printf("\t flow area is : %.3f ft^2 \n",at);
66 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
67 printf("\t mass velocity is : %.3e lb/(hr)*(ft^2) \n
    ",Gt);
68 V=(Gt/(3600*62.5));
69 printf("\t V is %.1f fps \n",V);
70 mu2=0.92*2.42; // at 77.5F,lb/(ft)*(hr)
71 D=0.65/12; //ft
72 Ret=((D)*(Gt)/mu2); // reynolds number
73 printf("\t reynolds number is : %.2e \n",Ret);
74 hi=1350*0.99; //using fig.25 ,Btu/(hr)*(ft^2)*(F)
75 ID=0.65; // ft
76 OD=0.75; //ft
77 hio=((hi)*(ID/OD)); // using eq.6.5
78 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
79 Uc=((hio)*(ho)/(hio+ho)); // clean overall

```

```

    coefficient ,Btu/(hr)*(ft ^2)*(F)
80 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft ^2)*(F) \n",Uc);
81 printf("\t when both. film coefficients are high
    the thermal resistance of the tube metal is not
    necessarily insignificant as assumed in the
    derivation of Eq. (6.38). For a steel 1.8 BWG
    tube Rm= 0.00017 and for copper Rm= 0.000017 \n")
    ;
82 A2=0.1963; // actual surface supplied for each tube,
    ft ^2,from table 10
83 A=(Nt*L*A2); // ft ^2
84 printf("\t total surface area is : %.0f ft ^2 \n",A);
85 UD=((Q)/((A)*(delt)));
86 printf("\t actual design overall coefficient is : %
    .0f Btu/(hr)*(ft ^2)*(F) \n",UD);
87 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft ^2)*(F)/Btu
88 printf("\t actual Rd is : %.4f (hr)*(ft ^2)*(F)/Btu \
    n",Rd);
89 printf("\t pressure drop for annulus \n");
90 f=0.0019; // friction factor for reynolds number
    16200, using fig.29
91 s=1; // for reynolds number 25300,using fig.6
92 Ds=15.25/12; // ft
93 phys=1;
94 N=(12*L/B); // number of crosses ,using eq.7.43
95 printf("\t number of crosses are : %.0f \n",N);
96 delPs=((f*(Gs ^2)*(Ds)*(N))/(5.22*(10 ^10)*(De)*(s)*(
    phys))); // using eq.7.44 ,psi
97 printf("\t delPs is : %.1f psi \n",delPs);
98 printf("\t allowable delPs is 10 psi \n");
99 printf("\t pressure drop for inner pipe \n");
100 f=0.00019; // friction factor for reynolds number
    36400, using fig.26
101 s=1;
102 phyt=1;
103 D=0.054; // ft
104 delPt=((f*(Gt ^2)*(L)*(n))/(5.22*(10 ^10)*(D)*(s)*(

```

```

        phyt))); // using eq.7.45 , psi
105 printf("\t delPt is : %.1f psi \n",delPt);
106 X1=0.33; // X1=((V^2)/(2*g)), for Gt 1060000, using
        fig.27
107 delPr=((4*n*X1)/(s)); // using eq.7.46 , psi
108 printf("\t delPr is : %.1f psi \n",delPr);
109 delPT=delPt+delPr; // using eq.7.47 , psi
110 printf("\t delPT is : %.1f psi \n",delPT);
111 printf("\t allowable delPT is 10 psi \n");
112 //end

```

Scilab code Exa 7.5 Calculation of the Optimum Outlet water Temperature

```

1 printf("\t example 7.5 \n");
2 printf("\t approximate values are mentioned in the
        book \n");
3 T1=175; // inlet hot fluid ,F
4 T2=150; // outlet hot fluid ,F
5 t1=85; // inlet cold fluid ,F
6 delT1=T2-t1; //F
7 printf("\t delT1 is : %.0f F \n",delT1);
8 U=15; // assumption ,Btu/(hr)*(ft^2)*(F)
9 theta=8000; // operating hours ,hr
10 CW=(0.01/8300); // water cost ,$/lb
11 printf("\t For annual charges assume 20 per cent
        repair and maintenanc.e and 10 per cent
        depreciation \n");
12 CF=(0.3*4); // annual fixed charges/ft^2
13 c=1; // Btu/(lb)*(F)
14 X=((U)*(theta)*(CW)/(CF*c));
15 printf("\t X is : %.4f \n",X);
16 Y=((T1-T2)/delT1);
17 printf("\t Y is : %.2f \n",Y);
18 A=0.96; // A=(delT2/delT1), from fig 7.24

```

```

19 delT2=0.96*delT1;
20 printf("\t delT2 is : %.1f F \n",delT2);
21 t2=T1-delT2; // F
22 printf("\t t2 is : %.1f F \n",t2);
23 //end

```

Scilab code Exa 7.6 Calculation of a Phosphate Solution Cooler

```

1 printf("\t example 7.6 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=150; // inlet hot fluid ,F
4 T2=90; // outlet hot fluid ,F
5 t1=68; // inlet cold fluid ,F
6 t2=90; // outlet cold fluid ,F
7 W=20160; // lb/hr
8 w=41600; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for solution \n");
11 c=(0.3*0.19)+(0.7*1); // Btu/(lb)*(F), bcoz of 30
   percent of solution
12 Q1=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for solution is : %.2
   e Btu/hr \n",Q1);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.2e
   Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);

```

```

24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.2f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.3f \n",S);
28 printf("\t FT is 0.81 \n"); // from fig 18
29 delT=(0.81*LMTD); // F
30 printf("\t delT is : %.1f F \n",delT);
31 Tc=((T2)+(T1))/2; // caloric temperature of hot
    fluid ,F
32 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
33 tc=((t1)+(t2))/2; // caloric temperature of cold
    fluid ,F
34 printf("\t caloric temperature of cold fluid is : %
    .1f F \n",tc);
35 printf("\t hot fluid:shell side ,phosphate solution \
    n");
36 ID=10.02; // in
37 C=0.25; // clearance
38 B=2; // baffle spacing ,in
39 PT=1;
40 as=((ID*C*B)/(144*PT)); // flow area ,ft^2,using eq
    .7.1
41 printf("\t flow area is : %.4f ft^2 \n",as);
42 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2),using eq
    .7.2
43 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
44 mu1=1.20*2.42; // at 120F,lb/(ft)*(hr), from fig.14
45 De=0.95/12; // from fig.28 ,ft
46 Res=((De)*(Gs)/mu1); // reynolds number
47 printf("\t reynolds number is : %.3e \n",Res);
48 jH=71; // from fig.28
49 c=1; // Btu/(lb)*(F), at 120F,from fig.table 4
50 k=0.33; // Btu/(hr)*(ft^2)*(F/ft), from table 4
51 Pr=((0.757)*(mu1)/k)^(1/3); // prandelt number
    raised to power 1/3
52 printf("\t Pr is : %.3f \n",Pr);

```

```

53 ho=((jH)*(k/De)*(Pr)); // using eq.6.15 ,Btu/(hr)*(ft
    ^2)*(F)
54 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
55 printf("\t cold fluid:inner tube side ,raw water \n")
    ;
56 Nt=52;
57 n=2; // number of passes
58 L=16; //ft
59 at1=0.302; // flow area , in^2,from table 10
60 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
61 printf("\t flow area is : %.4f ft^2 \n",at);
62 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
63 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
64 V=(Gt/(3600*62.5));
65 printf("\t V is %.1f fps \n",V);
66 mu2=0.91*2.42; // at 79F,lb/(ft)*(hr),from table 14
67 D=(0.62/12); // from table 10
68 Ret=((D)*(Gt)/mu2); // reynolds number
69 printf("\t reynolds number is : %.2e \n",Ret);
70 hi=800*1; //using fig.25 ,Btu/(hr)*(ft^2)*(F)
71 ID=0.62; // ft
72 OD=0.75; //ft
73 hio=((hi)*(ID/OD)); // using eq.6.5
74 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
75 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
76 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
77 A2=0.1963; // actual surface supplied for each tube ,
    ft^2,from table 10
78 A=(Nt*L*A2); // ft^2
79 printf("\t total surface area is : %.0f ft^2 \n",A);
80 UD=((Q)/((A)*(delt)));
81 printf("\t actual design overall coefficient is : %"

```

```

    .0 f Btu/(hr)*(ft ^2)*(F) \n",UD);
82 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft ^2)*(F)/Btu
83 printf("\t actual Rd is : %.5f (hr)*(ft ^2)*(F)/Btu \
    n",Rd);
84 printf("\t pressure drop for annulus \n");
85 f=0.0019; // friction factor for reynolds number
    15750, using fig.29
86 s=1.3; // for reynolds number 25300,using fig.6
87 Ds=10.02/12; // ft
88 phys=1;
89 N=(12*L/B); // number of crosses ,using eq.7.43
90 printf("\t number of crosses are : %.0f \n",N);
91 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys))); // using eq.7.44 ,psi
92 printf("\t delPs is : %.1f psi \n",delPs);
93 printf("\t allowable delPs is 10 psi \n");
94 printf("\t pressure drop for inner pipe \n");
95 f=0.00023; // friction factor for reynolds number
    17900, using fig.26
96 s=1;
97 phyt=1;
98 D=0.0517; // ft
99 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45 ,psi
100 printf("\t delPt is : %.1f psi \n",delPt);
101 X1=0.08; // X1=((V^2)/(2*g)), for Gt 1060000,using
    fig.27
102 delPr=((4*n*X1)/(s)); // using eq.7.46 ,psi
103 printf("\t delPr is : %.1f psi \n",delPr);
104 delPT=delPt+delPr; // using eq.7.47 ,psi
105 printf("\t delPT is : %.1f psi \n",delPT);
106 printf("\t allowable delPT is 10 psi \n");
107 //end

```

Scilab code Exa 7.7 The Optimum Use of Exhaust and Process Steam


```

1 printf("\t example 7.7 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 U=50; // Btu/(hr)*(ft ^2)*(F)
4 TP=328; // F
5 TE=228; // F
6 CP=(0.30/(888.8*1000));
7 CE=(0.05/(960*1000));
8 CF=1.20;
9 theta=8000; // annual hours
10 X=((CF*(TP-TE))/((CP-CE)*U*theta)); // from eq 7.53
11 printf("\t X is : %.9f \n",X);
12 a=(1); // coefficient of t^2
13 b=(-556); // coefficient of t
14 c=(74784-X); // constant
15 printf("\t coefficient of t^2 is : %.2f \n",a);
16 printf("\t coefficient of t is : %.2f \n",b);
17 printf("\t constant term is : %.9f \n",c);
18 P=poly([c b a], 't', 'c');
19 t=roots(P);
20 printf("\t t is :%.0f \n",t);
21 printf("\t t cannot be greater than 328F \n \t t is
   218F \n");
22 //end

```

Scilab code Exa 7.8 Calculation of a Sugar solution Heater without Baffles

```

1 printf("\t example 7.8 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=228; // inlet hot fluid ,F
4 T2=228; // outlet hot fluid ,F
5 t1=100; // inlet cold fluid ,F
6 t2=122; // outlet cold fluid ,F
7 W=200000; // lb/hr

```

```

8 w=3950; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for solution \n");
11 c=(0.2*0.30)+(0.8*1); // bcoz of 20 percent solution
    ,Btu/(lb)*(F)
12 Q1=((W)*(c)*(t2-t1)); // Btu/hr
13 printf("\t total heat required for solution is : %.2
    e Btu/hr \n",Q1);
14 printf("\t for steam \n");
15 l=960.1; // latent heat of condensation ,Btu/(lb)
16 Q=((w)*(l)); // Btu/hr
17 printf("\t total heat required for steam is : %.2e
    Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.0f \n",R);
26 delT=(LMTD); // when R=0,F
27 printf("\t delT is : %.1f F \n",delT);
28 printf("\t The steam coefficient will be very great
    compared with that for the sugar solution , and
    the tube wall will be considerably nearer 228 F
    than the caloric temperature of the fluid . Obtain
    Fc from U1 and U0 Failure to correct for wall
    effects , however, will keep the heater
    calculation on the safe side.\n");
29 ta=111; //F
30 Ta=228; //f
31 printf("\t hot fluid:tube side ,steam \n");
32 Nt=76;
33 n=2; // number of passes
34 L=16; //ft
35 at1=0.302; // flow area , in^2
36 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq

```

```

    .7.48
37 printf("\t flow area is : %.4f ft^2 \n",at);
38 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
39 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n"
    ,Gt);
40 mu2=0.0128*2.42; // at 228F,lb/(ft)*(hr)
41 D=(0.62/12); // from table 10,ft
42 Ret=((D)*(Gt)/mu2); // reynolds number
43 printf("\t reynolds number is : %.2e \n",Ret);
44 hio=1500; // for condensation of steam
45 printf("\t Correct hio to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
46 printf("\t cold fluid:shell side,sugar solution \n")
    ;
47 ID=12; // in
48 d=0.75/12; // diameter of tube,ft
49 Nt=76; // number of tubes
50 as=((3.14*(12^2)/4)-(76*3.14*(0.75^2)/4))/144; //
    flow area ,ft^2
51 printf("\t flow area is : %.2f ft^2 \n",as);
52 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2)
53 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
54 mu1=1.30*2.42; // at 111F,lb/(ft)*(hr), from fig.14
55 De=((4*as)/(Nt*3.14*d)); // from eq.6.3 ,ft
56 printf("\t De is : %.3f ft \n",De);
57 Res=((De)*(Gs)/mu1); // reynolds number
58 printf("\t reynolds number is : %.2e \n",Res);
59 jH=61.5; // from fig.24, tube side data
60 c=0.86; // Btu/(lb)*(F),at 111F,from fig.4
61 k=0.333; // Btu/(hr)*(ft^2)*(F/ft)
62 Pr=((c)*(mu1)/k)^(1/3); // prandelt number raised to
    power 1/3
63 printf("\t Pr is : %.0f \n",Pr);
64 Ho=((jH)*(k/De)*(Pr)); // H0=(h0/phy),using eq
    .6.15 ,Btu/(hr)*(ft^2)*(F)
65 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Ho);

```

```

66 muw=0.51*2.42; // at 210F,lb/(ft)*(hr), from fig.14
67 phys=(mu1/muw)^0.14;
68 printf("\t phys is : %.2f \n",phys); // from fig.24
69 ho=(Ho)*(phys); // from eq.6.36
70 printf("\t Correct h0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
71 tw=(ta)+(((hio)/(hio+Ho))*(Ta-ta)); // from eq.5.31
72 printf("\t tw is : %.0f F \n",tw);
73 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
74 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
75 A2=0.1963; // actual surface supplied for each tube,
    ft^2,from table 10
76 A=(Nt*L*A2); // ft^2
77 printf("\t total surface area is : %.0f ft^2 \n",A);
78 UD=((Q)/((A)*(LMTD)));
79 printf("\t actual design overall coefficient is : %
    .0f Btu/(hr)*(ft^2)*(F) \n",UD);
80 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
81 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
82 printf("\t pressure drop for inner pipe \n");
83 f=0.000155; // friction factor for reynolds number
    82500, using fig.26
84 s=0.0008;
85 phyt=1;
86 D=0.0517;
87 delPt=((f*(Gt^2)*(L)*(2))/(5.22*(10^10)*(D)*(s)*(
    phyt)))/2; // using eq.7.45, psi
88 printf("\t delPt is : %.1f psi \n",delPt);
89 printf("\t pressure drop for annulus \n");
90 De1=((4*as)/((Nt*3.14*d)+(3.14*1))); // from eq.6.4,
    ft
91 printf("\t De1 is : %.3f ft \n",De1);
92 Res1=(De1*Gs/mu1); // from eq 7.3
93 printf("\t Res1 is : %.2e \n",Res1);
94 f=0.00025; // friction factor, using fig.26

```

```

95 s=1.08; // for reynolds number 25300,using fig.6
96 delPs=((f*(Gs^2)*(L)*(1))/(5.22*(10^10)*(De1)*(s)*(
    phys))); // using eq.7.44 ,psi
97 printf("\t delPs is : %.2f psi \n",delPs);
98 //end

```

Scilab code Exa 7.9 Outlet Temperatures

```

1 printf("\t example 7.9 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=390; // F
4 t1=100; // F
5 U=69.3; // Btu/(hr)*(ft^2)*(F)
6 A=662; // ft^2
7 W=43800; // lb/hr
8 w=149000; // lb/hr
9 C=0.60; // Btu/(lb)*(F)
10 c=0.49; // Btu/(lb)*(F)
11 X=((U*A)/(w*c));
12 printf("\t X is : %.2f \n",X);
13 R=((w*c)/(W*C));
14 printf("\t R is : %.2f \n",R);
15 S=0.265; // from fig 7.25, by comparing X an R
16 t2=(t1)+((0.265)*(T1-t1)); // S=((t2-t1)/(T1-t1))
17 printf("\t t2 is : %.0f F \n",t2);
18 T2=((T1)-((R)*(t2-t1))); // R=((T1-T2)/(t2-t1))
19 printf("\t T2 is : %.0f F \n",T2);
20 // end

```

Chapter 8

Flow Arrangements for increased heat recovery

Scilab code Exa 8.1 Calculation of a Oil Cooler

```
1 printf("\t example 8.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=358; // inlet hot fluid ,F
4 T2=100; // outlet hot fluid ,F
5 t1=90; // inlet cold fluid ,F
6 t2=120; // outlet cold fluid ,F
7 W=49600; // lb/hr
8 w=233000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for oil \n");
11 c=0.545; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for oil is : %.2e Btu
   /hr \n",Q);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.2e
```

```

    Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.0f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.1f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.3f \n",S);
28 printf("\t FT is 0.93 \n"); // from fig 19 for 2-4
    exchanger
29 delT=(0.93*LMTD); // F
30 printf("\t delT is : %.1f F \n",delT);
31 X=((delT1)/(delT2));
32 printf("\t ratio of two local temperature difference
    is : %.3f \n",X);
33 Fc=0.25; // from fig.17
34 Kc=0.47; // crude oil controlling
35 Tc=((T2)+((Fc)*(T1-T2))); // caloric temperature of
    hot fluid ,F
36 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
37 tc=((t1)+((Fc)*(t2-t1))); // caloric temperature of
    cold fluid ,F
38 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
39 printf("\t hot fluid:shell side ,oil \n");
40 ID=35; // in
41 C=0.25; // clearance
42 B=7; // baffle spacing ,in
43 PT=1.25;
44 as=((ID*C*B)/(144*PT))/2; // flow area ,ft^2 ,from eq
    7.1
45 printf("\t flow area is : %.2f ft^2 \n",as);
46 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2) ,from eq
    7.2

```

```

47 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
48 mu1=1.12*2.42; // at 165F,lb/(ft)*(hr), from fig.14
49 De=0.99/12; // from fig.28,ft
50 Res=((De)*(Gs)/mu1); // reynolds number
51 printf("\t reynolds number is : %.1e \n",Res);
52 jH=52.5; // from fig.28
53 Z=0.2; // Z=(k)*(Pr*(1/3)) prandelt number
54 Ho=((jH)*(1/De)*(Z)); // H0=(h0/phys),using eq.6.15,
    Btu/(hr)*(ft^2)*(F)
55 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Ho);
56 printf("\t cold fluid:inner tube side,water \n");
57 Nt=454;
58 n=6; // number of passes
59 L=12; //ft
60 at1=0.455; // flow area, in^2
61 at=((Nt*at1)/(144*n)); // total area,ft^2,from eq
    .7.48
62 printf("\t flow area is : %.3f ft^2 \n",at);
63 Gt=(w/(at)); // mass velocity,lb/(hr)*(ft^2)
64 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
65 V=(Gt/(3600*62.5)); // fps
66 printf("\t V is : %.2f fps \n",V);
67 mu2=0.73*2.42; // at 98F,lb/(ft)*(hr),from fig 14
68 D=(0.76/12); // ft,from table 10
69 Ret=((D)*(Gt)/mu2); // reynolds number
70 printf("\t reynolds number is : %.2e \n",Ret);
71 hi=1010*0.96; // using fig 25,Btu/(hr)*(ft^2)*(F)
72 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
73 ID=0.76; // ft
74 OD=1; //ft
75 hio=((hi)*(ID/OD)); // using eq.6.5
76 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
77 tw=(tc)+(((Ho)/(hio+Ho))*(Tc-tc)); // from eq.5.31
78 printf("\t tw is : %.0f F \n",tw);

```



```

79 muw=1.95*2.42; // lb/(ft)*(hr), from fig.14
80 phys=(mu1/muw)^0.14;
81 printf("\t phys is : %.2f \n",phys); // from fig.24
82 ho=(Ho)*(phys); // from eq.6.36
83 printf("\t Correct h0 to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
84 Uc=((hio)*(ho)/(hio+ho)); // clean overall
      coefficient ,Btu/(hr)*(ft^2)*(F)
85 printf("\t clean overall coefficient is : %.0f Btu/(
      hr)*(ft^2)*(F) \n",Uc);
86 A2=0.2618; // actual surface supplied for each tube,
      ft^2,from table 10
87 A=(Nt*L*A2); // ft^2
88 printf("\t total surface area is : %.0f ft^2 \n",A);
89 Q=6980000; // taking rounded value ,Btu/hr
90 UD=((Q)/((A)*(delt)));
91 printf("\t actual design overall coefficient is : %
      .1f Btu/(hr)*(ft^2)*(F) \n",UD);
92 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
93 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
      n",Rd);
94 printf("\t pressure drop for annulus \n");
95 f=0.00215; // friction factor for reynolds number
      8900, using fig.29
96 s=0.82; // for reynolds number 25300,using fig.6
97 Ds=35/12; // ft
98 N=(12*L/B); // number of crosses ,using eq.7.43
99 printf("\t number of crosses are : %.0f \n",N);
100 delPs=((f*(Gs^2)*(Ds)*(2*N))/(5.22*(10^10)*(De)*(s)
      *(phys))); // using eq.7.44,psi
101 printf("\t delPs is : %.0f psi \n",delPs);
102 printf("\t allowable delPs is 10 psi \n");
103 printf("\t pressure drop for inner pipe \n");
104 f=0.000195; // friction factor for reynolds number
      34900, using fig.26
105 s=1;
106 D=0.0633; //ft
107 phyt=1;

```

```

108 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45 , psi
109 printf("\t delPt is : %.1f psi \n",delPt);
110 X1=0.13; // X1=((V^2)/(2*g)), for Gt 1060000, using
    fig.27
111 delPr=((4*n*X1)/(s)); // using eq.7.46 , psi
112 printf("\t delPr is : %.1f psi \n",delPr);
113 delPT=delPt+delPr; // using eq.7.47 , psi
114 printf("\t delPT is : %.1f psi \n",delPT);
115 printf("\t allowable delPT is 10 psi \n");
116 //end

```

Scilab code Exa 8.2 calculation of an Acetone Acetic Acid Exchanger

```

1 printf("\t example 8.2 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=250; // inlet hot fluid ,F
4 T2=100; // outlet hot fluid ,F
5 t1=90; // inlet cold fluid ,F
6 t2=150; // outlet cold fluid ,F
7 W=60000; // lb/hr
8 w=168000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for acetone \n");
11 c=0.57; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for acetone is : %.2e
    Btu/hr \n",Q); // calculation mistake in problem
14 printf("\t for acetic acid \n");
15 c=0.51; // Btu/(lb)*(F)
16 Q1=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for acetic acid is :
    %.2e Btu/hr \n",Q1); // calculation mistake in
    problem

```

```

18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.1f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.3f \n",S);
28 printf("\t FT is 0.88 \n"); // from fig 20,for 3-6
    exchanger
29 delT=(0.88*LMTD); // F
30 printf("\t delT is : %.1f F \n",delT);
31 Tc=((T2)+(T1))/2; // caloric temperature of hot
    fluid ,F
32 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
33 tc=((t1)+(t2))/2; // caloric temperature of cold
    fluid ,F
34 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
35 printf("\t hot fluid:shell side ,acetone \n");
36 ID=21.25; // in
37 C=0.25; // clearance
38 B=5; // baffle spacing ,in
39 PT=1;
40 as=((ID*C*B)/(144*PT)); // flow area ,ft^2
41 printf("\t flow area is : %.3f ft^2 \n",as);
42 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2)
43 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
44 mu1=0.20*2.42; // at 175F,lb/(ft)*(hr), from fig.14
45 De=0.95/12; // from fig.28 ,ft
46 Res=((De)*(Gs)/mu1); // reynolds number
47 printf("\t reynolds number is : %.2e \n",Res);
48 phys=1;
49 jH=137; // from fig.28

```

```

50 c=0.63; // Btu/(lb)*(F),at 175F,from fig.2
51 k=0.095; // Btu/(hr)*(ft^2)*(F/ft), from table 4
52 Pr=((c)*(mu1)/k)^(1/3); // prandelt number raised to
    power 1/3
53 printf("\t Pr is : %.3f \n",Pr);
54 ho=((jH)*(k/De)*(Pr)*1); // using eq.6.15b,Btu/(hr)
    *(ft^2)*(F)
55 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
56 printf("\t cold fluid:inner tube side,acetic acid \n
    ");
57 Nt=270;
58 n=2; // number of passes
59 L=16; //ft
60 at1=0.268; // flow area, in^2,from table 10
61 at=((Nt*at1)/(144*n)); // total area,ft^2,from eq
    .7.48
62 printf("\t flow area is : %.3f ft^2 \n",at);
63 Gt=(w/(at)); // mass velocity,lb/(hr)*(ft^2)
64 printf("\t mass velocity is : %.1e lb/(hr)*(ft^2) \n
    ",Gt);
65 mu2=0.85*2.42; // at 120F,lb/(ft)*(hr)
66 D=(0.584/12); // ft
67 Ret=((D)*(Gt)/mu2); // reynolds number
68 printf("\t reynolds number is : %.2e \n",Ret);
69 jH=56; // from fig.24
70 c=0.51; // Btu/(lb)*(F),at 120F,from fig.2
71 k=0.098; // Btu/(hr)*(ft^2)*(F/ft), from table 4
72 phyt=1;
73 Pr=((c)*(mu2)/k)^(1/3); // prandelt number raised to
    power 1/3
74 printf("\t Pr is : %.3f \n",Pr);
75 hi=((jH)*(k/D)*(Pr)*(1^0.14)); // using eq.6.15a,Btu
    /(hr)*(ft^2)*(F)
76 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
77 ID=0.584; // ft
78 OD=0.75; //ft
79 hio=((hi)*(ID/OD)); //Hio=(hio/phyp), using eq.6.5

```

```

80 printf("\t Correct hi0 to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
81 Uc=((hio)*(ho)/(hio+ho)); // clean overall
      coefficient ,Btu/(hr)*(ft^2)*(F)
82 printf("\t clean overall coefficient is : %.1f Btu/(
      hr)*(ft^2)*(F) \n",Uc);
83 A2=0.1963; // actual surface supplied for each tube,
      ft^2,from table 10
84 A=3*(Nt*L*A2); // ft^2
85 printf("\t total surface area is : %.2e ft^2 \n",A);
86 UD=((Q)/((A)*(delt)));
87 printf("\t actual design overall coefficient is : %
      .1f Btu/(hr)*(ft^2)*(F) \n",UD);
88 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
89 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
      n",Rd);
90 printf("\t pressure drop for annulus \n");
91 f=0.00155; // friction factor for reynolds number
      52200, using fig.29
92 s=0.79; // for reynolds number 25300,using table.6
93 Ds=21.25/12; // ft
94 N=(12*L/B)+1; // number of crosses ,using eq.7.43
95 printf("\t number of crosses are : %.0f \n",N);
96 delPs=((f*(Gs^2)*(Ds)*(3*N))/(5.22*(10^10)*(De)*(s)
      *(phys))); // using eq.7.44,psi
97 printf("\t delPs is : %.1f psi \n",delPs);
98 printf("\t allowable delPs is 10 psi \n");
99 printf("\t pressure drop for inner pipe \n");
100 f=0.00024; // friction factor for reynolds number
      158000, using fig.26
101 s=1.07;
102 D=0.0487; // ft
103 delPt=((f*(Gt^2)*(L)*(n*3))/(5.22*(10^10)*(D)*(s)*(
      phyt))); // using eq.7.45 ,psi
104 printf("\t delPt is : %.1f psi \n",delPt);
105 X1=0.063; // X1=((V^2)/(2*g)), for Gt 1060000,using
      fig.27
106 delPr=(3)*((4*n*X1)/(s)); // using eq.7.46 ,psi

```

```
107 printf("\t delPr is : %.1f psi \n",delPr);
108 delPT=delPt+delPr; // using eq.7.47, psi
109 printf("\t delPT is : %.1f psi \n",delPT);
110 printf("\t allowable delPT is 10 psi \n");
111 //end
```

Chapter 9

Gases

Scilab code Exa 9.1 Calculation of an Ammonia Compressor Aftercooler

```
1 printf("\t example 9.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=245; // inlet hot fluid ,F
4 T2=95; // outlet hot fluid ,F
5 t1=85; // inlet cold fluid ,F
6 t2=95; // outlet cold fluid ,F
7 W=9872; // lb/hr
8 w=78500; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for ammonia gas \n");
11 c=0.53; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for ammonia gas is :
   %.2e Btu/hr \n",Q);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.2f
   Btu/hr \n",Q);
18 delT1=T2-t1; //F
```

```

19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.0f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.4f \n",S);
28 printf("\t FT is 0.837 \n"); // from fig 18
29 delT=(0.837*LMTD); // F
30 printf("\t delT is : %.1f F \n",delT);
31 Tc=((T2)+(T1))/2; // caloric temperature of hot
    fluid ,F
32 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
33 tc=((t1)+(t2))/2; // caloric temperature of cold
    fluid ,F
34 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
35 printf("\t hot fluid:shell side ,ammonia at 83psia \n
    ");
36 ID=23.25; // in
37 C=0.1875; // clearance
38 B=12; // baffle spacing ,in
39 PT=0.937;
40 as=((ID*C*B)/(144*PT)); // flow area ,ft ^2,from eq
    7.1
41 printf("\t flow area is : %.3f ft ^2 \n",as);
42 Gs=(W/as); // mass velocity ,lb/(hr)*(ft ^2),from eq
    7.2
43 printf("\t mass velocity is : %.2e lb/(hr)*(ft ^2) \n
    ",Gs);
44 mu1=0.012*2.42; // at 170F,lb/(ft)*(hr) , from fig.15
45 De=0.55/12; // from fig.28,ft
46 Res=((De)*(Gs)/mu1); // reynolds number
47 printf("\t reynolds number is : %.2e \n",Res);
48 jH=118; // from fig.28

```



```

49 k=0.017; // Btu/(hr)*(ft^2)*(F/ft),from table 5
50 Z=0.97; // Z=(Pr*(1/3)) prandelt number
51 ho=((jH)*(k/De)*(Z)*1); // using eq.6.15 ,Btu/(hr)*(
    ft^2)*(F)
52 printf("\t individual heat transfer coefficient is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",ho);
53 printf("\t cold fluid:inner tube side ,water \n");
54 Nt=364;
55 n=8; // number of passes
56 L=8; //ft
57 at1=0.302; // flow area , in^2,from table 10
58 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
59 printf("\t flow area is : %.4f ft^2 \n",at);
60 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
61 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
62 V=(Gt/(3600*62.5)); // fps
63 printf("\t V is : %.2f fps \n",V);
64 mu2=0.82*2.42; // at 90F,lb/(ft)*(hr),from fig 14
65 D=(0.62/12); // ft ,from table 10
66 Ret=((D)*(Gt)/mu2); // reynolds number
67 printf("\t reynolds number is : %.2e \n",Ret);
68 hi=900; // using fig 25,Btu/(hr)*(ft^2)*(F)
69 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
70 ID=0.62; // ft
71 OD=0.75; //ft
72 hio=((hi)*(ID/OD)); // using eq.6.5
73 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
74 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
75 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
76 A2=0.1963; // actual surface supplied for each tube ,
    ft^2,from table 10
77 A=(Nt*L*A2); // ft^2
78 printf("\t total surface area is : %.0f ft^2 \n",A);

```

```

79 UD=((Q)/((A)*(delt)));
80 printf("\t actual design overall coefficient is : %
      .1f Btu/(hr)*(ft^2)*(F) \n",UD);
81 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
82 printf("\t actual Rd is : %.3f (hr)*(ft^2)*(F)/Btu \
      n",Rd);
83 printf("\t pressure drop for annulus \n");
84 f=0.00162; // friction factor for reynolds number
      40200, using fig.29
85 Ds=23.25/12; // ft
86 phys=1;
87 N=(12*L/B); // number of crosses ,using eq.7.43
88 printf("\t number of crosses are : %.0f \n",N);
89 rowgas=0.209;
90 printf("\t rowgas is %.3f lb/ft^3 \n",rowgas);
91 s=rowgas/62.5;
92 printf("\t s is %.5f \n",s);
93 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
      phys))); // using eq.7.44 , psi
94 printf("\t delPs is : %.0f psi \n",delPs);
95 printf("\t allowable delPs is 2 psi \n");
96 printf("\t pressure drop for inner pipe \n");
97 f=0.000225; // friction factor for reynolds number
      21400, using fig.26
98 s=1;
99 D=0.0517; //ft
100 phyt=1;
101 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
      phyt))); // using eq.7.45 , psi
102 printf("\t delPt is : %.1f psi \n",delPt);
103 X1=0.090; // X1=((V^2)/(2*g)), for Gt 1060000, using
      fig.27
104 delPr=((4*n*X1)/(s)); // using eq.7.46 , psi
105 printf("\t delPr is : %.1f psi \n",delPr);
106 delPT=delPt+delPr; // using eq.7.47 , psi
107 printf("\t delPT is : %.1f psi \n",delPT);
108 printf("\t allowable delPT is 10 psi \n");
109 //end

```

Scilab code Exa 9.2 Calculation of the Heat Load for an Air Intercooler

```
1 printf("\t example 9.2 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 V1=4670; // inlet air volume,cfm
4 Pp=0.8153; // Saturation partial pressure of water
   at 95F,psi,from table 7
5 Ps=404.3; // Saturation specific volume of water at
   95F,ft^3/lb, from table 7
6 printf("\t The air and water both occupy the same
   volume at their respective partial pressures \n")
   ;
7 Vw1=(V1*60/Ps); // water entering per hr,lb
8 printf("\t volume of water entering is : %.0f lb \n"
   ,Vw1);
9 printf("\t for first stage \n");
10 c=2.33; // compression ratio
11 P1=14.7; // psi
12 P2=(P1*c); // (c=(P2/P1)),psi
13 printf("\t P2 is : %.1f psi \n",P2);
14 gama=1.4; // for air
15 T1abs=95; // F
16 T2absr=((T1abs+460)*(P2/P1)^((gama-1)/gama));
17 printf("\t T2absr is : %.0f R \n",T2absr);
18 T2abs=(T2absr-459.67); // F
19 printf("\t T2abs is : %.0f F \n",T2abs);
20 printf("\t for intercooler \n");
21 V2=(V1*60*P1/P2); // ft^3/hr
22 printf("\t final gas volume is : %.1e ft^3/hr \n",V2
   );
23 Vw2=(V2/Ps); // water remaining in air , lb/hr
24 printf("\t water remaining in air is : %.0f lb/hr \n"
   ",Vw2);
```

```

25 C=(Vw1-Vw2); // condensation in inter cooler , lb/hr
26 printf("\t condensation in inter cooler is : %.0f lb
/hr \n",C);
27 Vs=14.8; // Specific volume of atmospheric air ,ft^3/
lb
28 printf("\t Specific volume of atmospheric air is : %
.1f ft^3/lb \n",Vs);
29 Va=(V1*60/Vs); // air in inlet gas , lb/hr
30 printf("\t air in inlet gas is : %.2e lb/hr\n",Va);
31 printf("\t heat load(245 to 95F) \n");
32 printf("\t sensible heat \n");
33 Qair=((Va)*(0.25)*(245-T1abs)); // Btu/hr
34 printf("\t Qair is : %.2e Btu/hr \n",Qair);
35 Qwaters=(Vw1*0.45*(245-T1abs)); // Btu/hr
36 printf("\t Qwaters is : %.2e Btu/hr \n",Qwaters);
37 printf("\t latent heat \n");
38 l=1040.1; // latent heat
39 Qwaterl=(C*l); // Btu/hr
40 printf("\t Qwaterl is : %.2e Btu/hr \n",Qwaterl);
41 Qt1=Qair+Qwaters+Qwaterl;
42 printf("\t total heat is : %.3e Btu/hr \n",Qt1);
43 printf("\t for second stage \n");
44 c=2.33; // compression ratio
45 P3=(P2*c); // (c=(P3/P1)),psi
46 printf("\t P3 is : %.1f psi \n",P3);
47 V3=(V1*60*P1/P3); // ft^3/hr
48 printf("\t final gas volume is : %.2e ft^3/hr \n",V3
);
49 Vw3=(V3/Ps); // water remaining in air , lb/hr
50 printf("\t water remaining in air is : %.1f lb/hr \n
",Vw3);
51 C1=(297-Vw3); // condensation in inter cooler , lb/hr
52 printf("\t condensation in inter cooler is : %.1f lb
/hr \n",C1);
53 printf("\t heat load(245 to 95F) \n");
54 printf("\t sensible heat \n");
55 Qair=(Va*0.25*(245-T1abs)); // Btu/hr
56 printf("\t Qair is : %.2e Btu/hr \n",Qair);

```

```

57 Qwaters=(Vw2*0.44*(245-T1abs)); // Btu/hr
58 printf("\t Qwater is : %.2e Btu/hr \n",Qwaters);
59 printf("\t latent heat \n");
60 l=1040.1; // latent heat
61 Qwaterl=(C1*1); // Btu/hr, calculation mistake in
    book
62 printf("\t Qwater is : %.2e Btu/hr \n",Qwaterl);
63 Qt1=Qair+Qwaters+Qwaterl;
64 printf("\t total heat is : %.3e Btu/hr \n",Qt1);
65 // end

```

Scilab code Exa 9.3 Calculation of the Dew Point after Compression

```

1 printf("\t example 9.3 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 Va=18900; // air in inlet gas
4 Vw1=692; // water entering
5 Ma=(Va/29); // moles
6 Mw=(Vw1/18); // moles
7 M=(Ma+Mw); // moles
8 printf("\t total number of moles re : %.1f \n",M);
9 printf("\t Moles of air is : %.0f \n",Ma);
10 printf("\t Moles of water is : %.1f \n",Mw);
11 printf("\t after compression \n");
12 P=34.2; // pressure ,psi
13 pw=(Mw/M)*(P); // partial pressure
14 printf("\t partial pressure is :%.1f psi \n",pw);
15 Td=124; // F, table table 7
16 printf("\t dew point is : %.0f F \n",Td);
17 // end

```

Chapter 10

Streamlineflow and steam convection

Scilab code Exa 10.1 Crude Oil Heater

```
1 printf("\t example 10.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=250; // inlet hot fluid ,F
4 T2=250; // outlet hot fluid ,F
5 t1=95; // inlet cold fluid ,F
6 t2=145; // outlet cold fluid ,F
7 W=16000; // lb/hr
8 w=410; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for crude \n");
11 c=0.485; // Btu/(lb)*(F)
12 Q=((W)*(c)*(t2-t1)); // Btu/hr
13 printf("\t total heat required for crude is : %.2e
   Btu/hr \n",Q);
14 printf("\t for steam \n");
15 l=945.5; // Btu/(lb)
16 Q=((w)*(l)); // Btu/hr
17 printf("\t total heat required for steam is : %.2e
```

```

    Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.0f F \n",LMTD);
24 printf("\t On the assumption that the fluids are
    mixed between passes, each pass must be solved
    independently. Since only two passes are present
    in this exchanger, it is simply a matter of
    assuming the temperature at the end of the first
    pass. More than half the heat load must be
    transferred in the first pass; therefore assume
    ti at the end of the first pass is 125 F \n");
25 ti=125; // F
26 tc=((t1)+(ti))/2; // caloric temperature of cold
    fluid ,F
27 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
28 printf("\t hot fluid:shell side ,steam \n");
29 ho=(1500); // condensation of steam Btu/(hr)*(ft^2)
    *(F)
30 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
31 printf("\t cold fluid:inner tube side ,crude \n");
32 Nt=86;
33 n=2; // number of passes
34 L=12; //ft
35 at1=0.594; // flow area, in^2,from table 10
36 at=((Nt*at1)/(144*n)); // total area,ft^2,from eq
    .7.48
37 printf("\t flow area is : %.3f ft^2 \n",at);
38 Gt=(W/(.177)); // mass velocity ,lb/(hr)*(ft^2)
39 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
40 mu2=2.95*2.42; // at 145F,lb/(ft)*(hr)
41 D=(0.87/12); // ft

```

```

42 Ret1=((D)*(Gt)/mu2); // reynolds number
43 printf("\t reynolds number is : %.0f \n",Ret1);
44 mu3=4.8*2.42; // at 110F,lb/(ft)*(hr)
45 D=(0.87/12); // ft
46 Ret2=((D)*(Gt)/mu3); // reynolds number
47 printf("\t reynolds number is : %.0f \n",Ret2);
48 c=0.485; // Btu/(lb)*(F),at 120F,from fig.2
49 k=0.0775; // Btu/(hr)*(ft^2)*(F/ft), from table 4
50 Pr=((c)*(mu3)/k); // prandelt number
51 printf("\t prandelt number is : %.1f \n",Pr);
52 Hi=((1.86)*(k/D)*((Ret2*(D/L)*Pr)^(1/3))); // using
    eq.6.1 ,Btu/(hr)*(ft^2)*(F)
53 printf("\t Hi is : %.1f Btu/(hr)*(ft^2)*(F) \n",Hi);
54 muw=1.2*2.42; // lb/(ft)*(hr),at 249F from fig.14
55 phyt=(mu3/muw)^0.14;
56 printf("\t phyt is : %.1f \n",phyt); // from fig.24
57 hi=(Hi)*(phyt); // from eq.6.37
58 printf("\t Correct hi to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
59 tp=(tc)+(((ho)/(hi+ho))*(T1-tc)); // from eq.5.31
60 printf("\t tp is : %.0f F \n",tp);
61 delt=tp-tc; //F
62 printf("\t delt is : %.0f F \n",delt);
63 Ai1=0.228 // internal surface per foot of length ,ft
64 Ai=(Nt*L*Ai1/2); // ft^2
65 printf("\t total surface area is : %.1f ft^2 \n",Ai)
    ;
66 delt3=((hi*Ai*delt)/(W*c)); // delt3=ti-t1, F
67 printf("\t delt3 is : %.1f F \n",delt3);
68 ti=t1+delt3; // F
69 printf("\t ti is : %.1f F \n",ti);
70 printf("\t The oil now enters the second pass at
    126.9 F \n");
71 // end

```

Scilab code Exa 10.2 Kerosene Heater

```
1 printf("\t example 10.2 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=250; // inlet hot fluid ,F
4 T2=250; // outlet hot fluid ,F
5 t1=95; // inlet cold fluid ,F
6 t2=145; // outlet cold fluid ,F
7 W=16000; // lb/hr
8 w=423; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for kerosene \n");
11 c=0.5; // Btu/(lb)*(F)
12 Q=((W)*(c)*(t2-t1)); // Btu/hr
13 printf("\t total heat required for kerosene is : %.0
   f Btu/hr \n",Q);
14 printf("\t for steam \n");
15 l=945.5; // Btu/(lb)
16 Q=((w)*(l)); // Btu/hr
17 printf("\t total heat required for steam is : %.2e
   Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.0f F \n",LMTD);
24 tc=((t1)+(t2))/2; // caloric temperature of cold
   fluid ,F
25 printf("\t caloric temperature of cold fluid is : %
   .0f F \n",tc);
26 printf("\t hot fluid:shell side ,steam \n");
27 ho=(1500); // condensation of steam Btu/(hr)*(ft^2)
   *(F)
28 printf("\t individual heat transfer coefficient is :
   %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
29 printf("\t cold fluid:inner tube side ,kerosene \n");
```

```

30 Nt=86;
31 n=2; // number of passes
32 L=12; //ft
33 at1=0.594; // flow area, in^2,from table 10
34 at=((Nt*at1)/(144*n)); // total area,ft^2,from eq
    .7.48
35 printf("\t flow area is : %.3f ft^2 \n",at);
36 Gt=(W/(.177)); // mass velocity,lb/(hr)*(ft^2)
37 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
38 mu2=1.36*2.42; // at 145F,lb/(ft)*(hr)
39 D=(0.87/12); // ft
40 Ret1=((D)*(Gt)/mu2); // reynolds number
41 printf("\t reynolds number is : %.0f \n",Ret1);
42 mu3=1.75*2.42; // at 120F,lb/(ft)*(hr)
43 D=(0.87/12); // ft
44 Ret2=((D)*(Gt)/mu3); // reynolds number
45 printf("\t reynolds number is : %.1e \n",Ret2);
46 Z1=331; // Z1=(L*n/D)
47 jH=3.1; // from fig 24
48 mu4=1.75; // cp and 40 API
49 Z2=0.24; // Z2=((k)*(c*mu4/k)^(1/3)), from fig 16
50 Hi=((jH)*(1/D)*(Z2)); // using eq.6.15a,Btu/(hr)*(ft
    ^2)*(F)
51 printf("\t Hi is : %.2f Btu/(hr)*(ft^2)*(F) \n",Hi);
52 ID=0.87; // ft
53 OD=1; //ft
54 Hio=(Hi*(ID/OD)); //Btu/(hr)*(ft^2)*(F), from eq.6.5
55 printf("\t Hio is : %.2f Btu/(hr)*(ft^2)*(F) \n",Hio
    );
56 tw=(tc)+(((ho)/(Hio+ho))*(T1-tc)); // from eq.5.31
57 printf("\t tw is : %.0f F \n",tw);
58 muw=1.45; // lb/(ft)*(hr),at 249F from fig.14
59 phyt=(mu3/muw)^0.14;
60 printf("\t phyt is : %.1f \n",phyt); // from fig.24
61 hio=(Hio)*(phyt); // from eq.6.37
62 printf("\t Correct hio to the surface at the OD is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",hio);

```

```

63 delt=tw-tc; //F
64 printf("\t delt is : %.0f F \n",delt);
65 printf("\t Since the kerosene has a viscosity of
    only 1.75 cp at the caloric temperature and delt
    =129F, free convection should be investigated. \n
    ");
66 s=0.8;
67 row=50; // lb/ft^3, from fig 6
68 s1=0.810; // at 95F
69 s2=0.792; // at 145F
70 bita=((s1^2-s2^2)/(2*(t2-t1)*s1*s2)); // /F
71 printf("\t beta is : %.6f /F \n",bita);
72 G=((D^3)*(row^2)*(bita)*(delt)*(4.18*10^8)/(mu3^2));
73 printf("\t G is : %.1e \n",G);
74 psy=((2.25)*(1+(0.01*G^(1/3)))/(log10(Ret2)));
75 printf("\t psy is : %.2f \n",psy);
76 hio1=(hio*psy);
77 printf("\t corrected hio1 is : %.1f Btu/(hr)*(ft^2)
    *(F) \n",hio1);
78 Uc=((hio1)*(ho)/(hio1+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
79 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
80 A2=0.2618; // actual surface supplied for each tube,
    ft^2,from table 10
81 A=(Nt*L*A2); // ft^2
82 printf("\t total surface area is : %.0f ft^2 \n",A);
83 UD=((Q)/((A)*(delt)));
84 printf("\t actual design overall coefficient is : %
    .1f Btu/(hr)*(ft^2)*(F) \n",UD);
85 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
86 printf("\t actual Rd is : %.2f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
87 // end

```

Scilab code Exa 10.3 Gas Oil Heater Using Cores

```
1 printf("\t example 10.3 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=250; // inlet hot fluid ,F
4 T2=250; // outlet hot fluid ,F
5 t1=105; // inlet cold fluid ,F
6 t2=130; // outlet cold fluid ,F
7 w=50000; // lb/hr
8 W=622; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for gas oil \n");
11 c=0.47; // Btu/(lb)*(F)
12 Q=((w)*(c)*(t2-t1)); // Btu/hr
13 printf("\t total heat required for gas oil is : %.2e
   Btu/hr \n",Q);
14 printf("\t for steam \n");
15 l=945.5; // Btu/(lb)
16 Q=((W)*(l)); // Btu/hr
17 printf("\t total heat required for steam is : %.2e
   Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 tc=((t1)+(t2))/2; // caloric temperature of cold
   fluid ,F
25 printf("\t caloric temperature of cold fluid is : %
   .1f F \n",tc);
26 printf("\t hot fluid:shell side ,steam \n");
27 ID=15.25; // in
28 C=0.25; // clearance
29 B=15; // baffle spacing ,in
30 PT=1.25;
31 as=((ID*C*B)/(144*PT)); // flow area ,ft ^2, eq 7.1
```

```

32 printf("\t flow area is : %.3f ft^2 \n",as);
33 Gs=(6220/as); // mass velocity ,lb/(hr)*(ft^2),
    calculation mistake
34 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
35 mu1=0.0314; // at 250F,lb/(ft)*(hr), from fig.15
36 De=0.060; // from fig.29,ft
37 Res=((De)*(Gs)/mu1); // reynolds number, calculation
    mistake
38 printf("\t reynolds number is : %.2e \n",Res);
39 ho=1500; //Btu/(hr)*(ft^2)*(F)
40 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
41 printf("\t cold fluid:inner tube side ,crude oil \n")
    ;
42 d1=0.5; // in
43 d2=0.87; // in
44 at1=((3.14*(d2^2-d1^2))/4);
45 printf("\t at1 is : %.1f in^2 \n",at1);
46 Nt=86;
47 n=2; // number of passes
48 L=12; //ft
49 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
50 printf("\t flow area is : %.3f ft^2 \n",at);
51 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
52 printf("\t mass velocity is : %.1e lb/(hr)*(ft^2) \n
    ",Gt);
53 De=(d2^2-d1^2)/(12*d2);
54 printf("\t De is : %.4f ft \n",De);
55 mu2=16.7; // at 117F,lb/(ft)*(hr)
56 Ret=((De)*(Gt)/mu2); // reynolds number
57 printf("\t reynolds number is : %.2e \n",Ret);
58 jH=3.1; // from fig.24
59 Z=0.35; // Z=(K*(c*mu3/k)^(1/3)),Btu/(hr)(ft^2)(F/ft
    ), at mu3=6.9cp and 28 API
60 Hi=((jH)*(1/De)*(Z)); //Hi=(hi/phyp), using eq.6.15a,
    Btu/(hr)*(ft^2)*(F)

```

```

61 printf("\t Hi is : %.1f Btu/(hr)*(ft^2)*(F) \n",Hi);
62 ID=0.87; // ft
63 OD=1; //ft
64 Hio=((Hi)*(ID/OD)); //Hio=(hio/phyt), using eq.6.5
65 printf("\t Correct Hi0 to the surface at the OD is :
        %.1f Btu/(hr)*(ft^2)*(F) \n",Hio);
66 muw=4.84; // lb/(ft)*(hr), from fig.14
67 phyt=(mu2/muw)^0.14;
68 printf("\t phyt is : %.2f \n",phyt); // from fig.24
69 hio=(Hio)*(phyt); // from eq.6.37
70 printf("\t Correct hi0 to the surface at the OD is :
        %.1f Btu/(hr)*(ft^2)*(F) \n",hio);
71 tw=(tc)+(((ho)/(hio+ho))*(T1-tc)); // from eq.5.31
72 printf("\t tw is : %.0f F \n",tw);
73 Uc=((hio)*(ho)/(hio+ho)); // clean overall
        coefficient ,Btu/(hr)*(ft^2)*(F)
74 printf("\t clean overall coefficient is : %.1f Btu/(
        hr)*(ft^2)*(F) \n",Uc);
75 A=270; // ft^2
76 printf("\t total surface area is : %.0f ft^2 \n",A);
77 UD=((Q)/((A)*(LMTD)));
78 printf("\t actual design overall coefficient is : %
        .1f Btu/(hr)*(ft^2)*(F) \n",UD);
79 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
80 printf("\t actual Rd is : %.3f (hr)*(ft^2)*(F)/Btu \
        n",Rd);
81 printf("\t pressure drop for annulus \n");
82 f=0.0016; // friction factor for reynolds number
        25300, using fig.29
83 s=0.00116; // for reynolds number 25300,using fig.6
84 Ds=15.25/12; // ft
85 phys=1;
86 N=(12*L/B); // number of crosses ,using eq.7.43
87 printf("\t number of crosses are : %.0f \n",N);
88 delPs=((f*(19600^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)
        *(phys)))/(2); // using eq.7.44 ,psi
89 printf("\t delPs is : %.1f psi \n",delPs);
90 printf("\t pressure drop for inner pipe \n");

```

```

91 dt=(d2-d1)/(12); // ft
92 printf("\t dt is : %.4f ft \n",dt);
93 Ret2=(dt*Gt/mu2);
94 printf("\t Ret2 is : %.0f \n",Ret2);
95 f=0.00066; // friction factor for reynolds number
    8220, using fig.26
96 phyt=1.35; // fig 6
97 printf("\t phyt is : %.2f \n",phyt);
98 s=0.85;
99 delPt=((f*(420000^2)*(L)*(n))/(5.22*(10^10)*(0.0309)
    *(s)*(phyt))); // using eq.7.45, psi
100 printf("\t delPt is : %.1f psi \n",delPt);
101 printf("\t delPr is negligible \n");
102 //end

```

Scilab code Exa 10.4 Calculation of a Heating Bundle for an Aniline Storage Tank

```

1 printf("\t example 10.4 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 t1=100; // F
4 t2=0; // F
5 T1abs=100+460; // R
6 T2abs=460; //R
7 delT=t1-t2;
8 printf("\t delT is : %.f F \n",delT);
9 hc=0.3*(delT^0.25); // convection loss , Btu/(hr)*(ft
    ^2)*( F )
10 printf("\t convection loss is : %.2f Btu/(hr)(ft ^2)(
    F) \n",hc);
11 e=0.8; // emissivity
12 hr=((0.173*e*((T1abs/100)^4-(T2abs/100)^4))/(T1abs-
    T2abs)); // radiation rate , from 4.32, Btu/(hr)(
    ft ^2)(F)

```

```

13 printf("\t radiation loss is : %.2f Btu/(hr)(ft ^2)(F
    ) \n",hr);
14 hl=hc+hr; // combined loss , Btu/(hr)(ft ^2)(F)
15 printf("\t combined loss is : %.1f Btu/(hr)(ft ^2)(F
    \n",hl);
16 D=5; // ft
17 L=12; // ft
18 A1=((2*3.14*D^2)/(4))+(3.14*D*L); // total tank area
19 printf("\t total tank area is : %.1f ft ^2 \n",A1);
20 Q=(hl*A1*delt); // total heat loss
21 printf("\t total heat loss : %.2e Btu/hr \n",Q);
22 printf("\t This heat must be supplied by the pipe
    bundle ,Assuming exhaust steam to be at 212 F \n"
    );
23 d0=1.32;
24 X=(delt/d0);
25 tf=((t1+212)/2); // F
26 printf("\t X is : %.0f \n",X);
27 printf("\t tf is : %.0f F \n",tf);
28 hio=48; // from fig 10.4, Btu/(hr)(ft ^2)(F)
29 ho=1500; // condensation of steam ,Btu/(hr)(ft ^2)(F)
30 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft ^2)*(F)
31 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft ^2)*(F) \n",Uc);
32 Rd=0.02; // dirt factor , (hr)(ft ^2)(F)/Btu
33 UD=((Uc)/((1)+(Uc*Rd))); // design overall
    coefficient ,Btu/(hr)*(ft ^2)*(F)
34 printf("\t design overall coefficient is : %.1f Btu
    /(hr)*(ft ^2)*(F) \n",UD);
35 A2=((Q)/((UD)*(212-100))); // total surface ,ft ^2
36 printf("\t total surface is : %.1f ft ^2 \n",A2);
37 A3=2.06; // area/pipe
38 N=(A2/A3);
39 printf("\t number of pipes are : %.0f \n",N);
40 //end

```

Chapter 11

Calculations for process conditions

Scilab code Exa 11.1 Calculation of a Straw Oil Naphtha Exchanger

```
1 printf("\t example 11.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=340; // inlet hot fluid ,F
4 T2=240; // outlet hot fluid ,F
5 t1=200; // inlet cold fluid ,F
6 t2=230; // outlet cold fluid ,F
7 W=29800; // lb/hr
8 w=103000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for straw oil \n");
11 c=0.58; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for straw oil is : %
   .2e Btu/hr \n",Q);
14 printf("\t for naphtha \n");
15 c=0.56; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for naphtha is : %.2e
```

```

        Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.1f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.3f \n",S);
28 printf("\t FT is 0.885 \n"); // from fig 18
29 delT=(0.885*LMTD); // F
30 printf("\t delT is : %.1f F \n",delT);
31 X=((delT1)/(delT2));
32 printf("\t ratio of two local temperature difference
        is : %.3f \n",X);
33 L=16;
34 Fc=0.405; // from fig.17
35 Kc=0.23; // crude oil controlling
36 Tc=((T2)+((Fc)*(T1-T2))); // caloric temperature of
        hot fluid ,F
37 printf("\t caloric temperature of hot fluid is : %.1
        f F \n",Tc);
38 tc=((t1)+((Fc)*(t2-t1))); // caloric temperature of
        cold fluid ,F
39 printf("\t caloric temperature of cold fluid is : %
        .0f F \n",tc);
40 UD1=70; // assume, from table 8a
41 A1=((Q)/((UD1)*(delT)));
42 printf("\t A1 is : %.0f ft^2 \n",A1);
43 a1=0.1963; // ft^2/lin ft
44 N1=(A1/(16*a1));
45 printf("\t number of tubes are : %.0f \n",N1);
46 N2=124; // assuming two tube passes, from table 9
47 A2=(N2*L*a1); // ft^2
48 printf("\t total surface area is : %.1e ft^2 \n",A2)
        ;

```

```

49 UD=((Q)/((A2)*(delt)));
50 printf("\t correct design overall coefficient is : %
    .1f Btu/(hr)*(ft^2)*(F) \n",UD);
51 printf("\t hot fluid:shell side ,straw oil \n");
52 ID=15.25; // in
53 C=0.25; // clearance
54 B=3.5; // minimum baffle spacing ,from eq 11.4 ,in
55 PT=1;
56 as=((ID*C*B)/(144*PT)); // flow area ,from eq 7.1 ,ft
    ^2
57 printf("\t flow area is : %.4f ft^2 \n",as);
58 Gs=(W/as); // mass velocity ,from eq 7.2 ,lb/(hr)*(ft
    ^2)
59 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
60 mu1=3.63; // at 280.5F ,lb/(ft)*(hr) , from fig.14
61 De=0.95/12; // from fig.28 ,ft
62 Res=((De)*(Gs)/mu1); // reynolds number
63 printf("\t reynolds number is : %.0e \n",Res);
64 jH=46; // from fig.28
65 Z=0.224; // Z=(K*(c*mu3/k)^(1/3)) ,Btu/(hr)(ft^2)(F/
    ft) , at mu3=1.5cp and 35 API
66 Ho=((jH)*(1/De)*(Z)); // H0=(h0/physa) ,using eq.6.15 ,
    Btu/(hr)*(ft^2)*(F)
67 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Ho);
68 phys=1;
69 ho=(Ho)*(phys); // from eq.6.36
70 printf("\t Correct h0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
71 printf("\t cold fluid:inner tube side ,naphtha \n");
72 Nt=124;
73 n=2; // number of passes
74 L=16; //ft
75 at1=0.302; // flow area , in^2
76 at=((Nt*at1)/(144*n)); // total area ,ft^2 ,from eq
    .7.48
77 printf("\t flow area is : %.3f ft^2 \n",at);

```

```

78 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
79 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
      ",Gt);
80 mu2=1.31; // at 212F,lb/(ft)*(hr)
81 D=0.0517; // ft
82 Ret=((D)*(Gt)/mu2); // reynolds number
83 printf("\t reynolds number is : %.2e \n",Ret);
84 jH=102; // from fig.24
85 Z=0.167; // Z=(K*(c*mu3/k)^(1/3)),Btu/(hr)(ft^2)(F/
      ft), at mu4=0.54cp and 48 API
86 Hi=((jH)*(1/D)*(Z)); //Hi=(hi/phyt), using eq.6.15a,
      Btu/(hr)*(ft^2)*(F)
87 printf("\t Hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",Hi);
88 ID=0.62; // ft
89 OD=0.75; //ft
90 Hio=((Hi)*(ID/OD)); //Hio=(hio/phyt), using eq.6.5
91 printf("\t Correct Hi0 to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",Hio);
92 phyt=1;
93 hio=(Hio)*(phyt); // from eq.6.37
94 printf("\t Correct hi0 to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
95 printf("\t pressure drop for annulus \n");
96 f=0.00225; // friction factor for reynolds number
      7000, using fig.29
97 s=0.76; // for reynolds number 7000,using fig.6
98 Ds=15.25/12; // ft
99 N=(12*L/B); // number of crosses ,using eq.7.43
100 printf("\t number of crosses are : %.0f \n",N);
101 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
      phys))); // using eq.7.44 ,psi
102 printf("\t delPs is : %.1f psi \n",delPs);
103 printf("\t pressure drop for inner pipe \n");
104 f=0.0002; // friction factor for reynolds number
      31300, using fig.26
105 s=0.72;
106 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
      phyt))); // using eq.7.45 ,psi

```

```

107 printf("\t delPt is : %.1f psi \n",delPt);
108 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft ^2)*(F)
109 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft ^2)*(F) \n",Uc);
110 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft ^2)*(F)/Btu
111 printf("\t actual Rd is : %.4f (hr)*(ft ^2)*(F)/Btu \
    n",Rd);
112 printf("\t The first trial is disqualified because
    of failure to meet the required dirt factor \n");
113 printf("\t Proceeding as above and carrying the
    viscosity correction and pressure drops to
    completion the new summary is given using a 17.25
    in. ID shell with 166 tubes on two passes and a
    3.5in. baffle space \n");
114 UD1=60; // assumption for 2 tube passes ,3.5 baffle
    spacing and 17.25in ID
115 UC1=74.8;
116 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft ^2)*(F) \n",UC1);
117 UD2=54.2;
118 printf("\t correct design overall coefficient is : %
    .1f Btu/(hr)*(ft ^2)*(F) \n",UD2);
119 Rd1=0.005;
120 printf("\t actual Rd is : %.3f (hr)*(ft ^2)*(F)/Btu \
    n",Rd1);
121 delPs1=4.7;
122 printf("\t delPs is : %.1f psi \n",delPs1);
123 delPt1=2.1;
124 printf("\t delPt is : %.1f psi \n",delPt1);
125 //end

```

Scilab code Exa 11.2 Calculation of a Flue Gas Cooler

```

1 printf("\t example 11.2 \n");

```

```

2  printf("\t approximate values are mentioned in the
   book \n");
3  T1=350; // inlet hot fluid ,F
4  T2=160; // outlet hot fluid ,F
5  t1=100; // inlet cold fluid ,F
6  t2=295; // outlet cold fluid ,F
7  W=84438; // lb/hr
8  w=86357; // lb/hr
9  printf("\t 1.for heat balance \n");
10 printf("\t for lean oil \n");
11 c=0.56; // Btu/(lb)*(F)
12 Qh=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for lean oil is : %.2
   e Btu/hr \n",Qh);
14 printf("\t for rich oil \n");
15 c=0.53; // Btu/(lb)*(F)
16 Qc=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for rich oil is : %.2
   e Btu/hr \n",Qc);
18 Q=(Qh+Qc)/(2);
19 printf("\t Q is : %.2e V \n",Q);
20 delT1=T2-t1; //F
21 delT2=T1-t2; // F
22 printf("\t delT1 is : %.0f F \n",delT1);
23 printf("\t delT2 is : %.0f F \n",delT2);
24 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
25 printf("\t LMTD is :%.1f F \n",LMTD);
26 R=((T1-T2)/(t2-t1));
27 printf("\t R is : %.3f \n",R);
28 S=((t2-t1)/(T1-t1));
29 printf("\t S is : %.2f \n",S);
30 printf("\t FT is 0.875 \n"); // for 4-8 exchanger ,
   from fig 21
31 delT=(0.875*LMTD); // F
32 printf("\t delT is : %.1f F \n",delT);
33 X=((delT1)/(delT2));
34 printf("\t ratio of two local temperature difference
   is : %.2f \n",X);

```

```

35 Fc=0.48; // from fig.17
36 Kc=0.32; // crude oil controlling
37 Tc=((T2)+((Fc)*(T1-T2))); // caloric temperature of
    hot fluid ,F
38 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
39 tc=((t1)+((Fc)*(t2-t1))); // caloric temperature of
    cold fluid ,F
40 printf("\t caloric temperature of cold fluid is : %
    .1 f \n",tc);
41 UD1=50; // assume, from table 8a
42 A1=((Q)/((UD1)*(delt)));
43 printf("\t A1 is : %.2e ft^2 \n",A1);
44 a1=0.1963; // ft^2/lin ft
45 N1=(A1/(16*a1*2)); // 2-4 exchanger in series
46 printf("\t number of tubes are : %.0f \n",N1);
47 N2=580; // assuming six tube passes,31in ID, from
    table 9
48 A2=(N2*16*a1*2); // ft^2
49 printf("\t total surface area is : %.2e ft^2 \n",A2)
    ;
50 UD=((Q)/((A2)*(delt)));
51 printf("\t correct design overall coefficient is : %
    .0 f Btu/(hr)*(ft^2)*(F) \n",UD);
52 printf("\t hot fluid:inner tube side,lean oil \n");
53 Nt=580;
54 n=6; // number of passes
55 L=16; //ft
56 at1=0.302; // flow area , in^2
57 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
58 printf("\t flow area is : %.3f ft^2 \n",at);
59 Gt=(W/(at)); // mass velocity ,lb/(hr)*(ft^2)
60 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
61 mu2=2.13; // at 212F,lb/(ft)*(hr)
62 D=0.0517; // ft
63 Ret=((D)*(Gt)/mu2); // reynolds number

```

```

64 printf("\t reynolds number is : %.2e \n",Ret);
65 jH=36.5; // from fig.24
66 Z=0.185; // Z=(K*(c*mu3/k)^(1/3)),Btu/(hr)(ft^2)(F/
    ft), at mu4=0.88cp and 35 API
67 Hi=((jH)*(1/D)*(Z)); //Hi=(hi/phypp), using eq.6.15a,
    Btu/(hr)*(ft^2)*(F)
68 printf("\t Hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",Hi);
69 ID=0.62; // ft
70 OD=0.75; //ft
71 Hio=((Hi)*(ID/OD)); //Hio=(hio/phypp), using eq.6.5
72 printf("\t Correct Hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Hio);
73 phyt=1;
74 hio=(Hio)*(phyt); // from eq.6.37
75 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
76 printf("\t cold fluid:shell side,rich oil \n");
77 ID=31; // in
78 C=0.25; // clearance
79 B=12; // minimum baffle spacing ,from eq 11.4,in
80 PT=1;
81 as=((ID*C*B)/(144*PT))/(2); // flow area ,from eq
    7.1,ft^2
82 printf("\t flow area is : %.3f ft^2 \n",as);
83 Gs=(w/as); // mass velocity ,from eq 7.2,lb/(hr)*(ft
    ^2)
84 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
85 mu1=3.15; // at 193.5F,lb/(ft)*(hr), from fig.14
86 De=0.95/12; // from fig.28,ft
87 Res=((De)*(Gs)/mu1); // reynolds number
88 printf("\t reynolds number is : %.2e \n",Res);
89 jH=45; // from fig.28
90 Z=0.213; // Z=(K*(c*mu3/k)^(1/3)),Btu/(hr)(ft^2)(F/
    ft), at mu3=1.3cp and 35 API
91 Ho=((jH)*(1/De)*(Z)); // H0=(h0/phyya), using eq.6.15,
    Btu/(hr)*(ft^2)*(F)
92 printf("\t individual heat transfer coefficient is :

```



```

        %.0f Btu/(hr)*(ft^2)*(F) \n",Ho);
93 phys=1;
94 ho=(Ho)*(phys); // from eq.6.36
95 printf("\t Correct h0 to the surface at the OD is :
        %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
96 printf("\t pressure drop for inner pipe \n");
97 f=0.00027; // friction factor for reynolds number
        10100, using fig.26
98 s=0.77;
99 delPt=((2*f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
        phys))); // using eq.7.45, psi
100 printf("\t delPt is : %.1f psi \n",delPt);
101 X1=0.024; // X1=((V^2)/(2*g)), for Gt 1060000,using
        fig.27
102 delPr=((4*2*n*X1)/(s)); // using eq.7.46, psi
103 printf("\t delPr is : %.1f psi \n",delPr);
104 delPT=delPt+delPr; // using eq.7.47, psi
105 printf("\t delPT is : %.1f psi \n",delPT);
106 printf("\t allowable delPT is 10 psi \n");
107 printf("\t pressure drop for annulus \n");
108 f=0.0023; // friction factor for reynolds number
        6720, using fig.29
109 s=0.79; // for reynolds number 6720,using fig.6
110 Ds=31/12; // ft
111 De=0.0792;
112 N=(4*12*L/B); // number of crosses, using eq.7.43
113 printf("\t number of crosses are : %.0f \n",N);
114 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
        phys))); // using eq.7.44, psi
115 printf("\t delPs is : %.1f psi \n",delPs);
116 printf("\t allowable delPa is 10 psi \n");
117 Uc=((hio)*(ho)/(hio+ho)); // clean overall
        coefficient ,Btu/(hr)*(ft^2)*(F)
118 printf("\t clean overall coefficient is : %.1f Btu/(
        hr)*(ft^2)*(F) \n",Uc);
119 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
120 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
        n",Rd);

```

```

121 printf("\t The initial assumptions have provided an
    exchanger which very nearly meets all the
    requirements. Eight-pass units would meet the
    heat-transfer requirement but would give a tube-
    side pressure drop of 14 psi. The trial exchanger
    will be somewhat less suitable when the value of
    Q, is also taken into account. If the minimum
    dirt factor of 0.0040 is to be taken literally ,
    it will be necessary to try the next size shell \
n");
122 printf("\t Assume a 33 in. ID shell with six1 tube
    passes and baffies spaced 12-in. apart , since the
    pressure drop increases with the diameter of the
    shell for a given mass velocity. \n");
123 UC1=52.3;
124 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft ^2)*(F) \n",UC1);
125 UD2=42;
126 printf("\t correct design overall coefficient is : %
    .1f Btu/(hr)*(ft ^2)*(F) \n",UD2);
127 Rd1=0.0047;
128 printf("\t calculated Rd is : %.4f (hr)*(ft ^2)*(F)/
    Btu \n",Rd1);
129 Rd2=0.004;
130 printf("\t actual Rd is : %.3f (hr)*(ft ^2)*(F)/Btu \
n",Rd2);
131 delPs1=4.4;
132 printf("\t delPs is : %.1f psi \n",delPs1);
133 delPT1=7.9;
134 printf("\t delPt is : %.1f psi \n",delPT1);
135 //end

```

Scilab code Exa 11.3 Calculation of a Caustic Solution Cooler

```

1 printf("\t example 11.3 \n");

```

```

2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=190; // inlet hot fluid ,F
4 T2=120; // outlet hot fluid ,F
5 t1=80; // inlet cold fluid ,F
6 t2=120; // outlet cold fluid ,F
7 W=100000; // lb/hr
8 w=154000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for caustic \n");
11 c=0.88; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for caustic is : %.2e
   Btu/hr \n",Q);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.2e
   Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.2f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.3f \n",S);
28 printf("\t FT is 0.815 \n"); // for 4-8 exchanger ,
   from fig 21
29 delT=(0.815*LMTD); // F
30 printf("\t delT is : %.1f F \n",delT);
31 Tc=((T2)+(T1))/(2); // caloric temperature of hot
   fluid ,F
32 printf("\t caloric temperature of hot fluid is : %.0
   f F \n",Tc);
33 tc=((t1)+(t2))/(2); // caloric temperature of cold

```

```

fluid ,F
34 printf("\t caloric temperature of cold fluid is : %
    .1f \n",tc);
35 UD1=250; // assume, from table 8
36 A1=((Q)/((UD1)*(delt)));
37 printf("\t A1 is : %.0f ft^2 \n",A1);
38 a1=0.2618; // ft^2/lin ft
39 L=16;
40 N1=(A1/(16*a1));
41 printf("\t number of tubes are : %.0f \n",N1);
42 N2=140; // assuming four tube passes ,19.25in ID,
    from table 9
43 A2=(N2*L*a1); // ft^2
44 printf("\t total surface area is : %.0f ft^2 \n",A2)
    ;
45 UD=((Q)/((A2)*(delt)));
46 printf("\t correct design overall coefficient is : %
    .0f Btu/(hr)*(ft^2)*(F) \n",UD);
47 printf("\t hot fluid:shell side ,caustic \n");
48 ID=19.25; // in
49 C=0.25; // clearance
50 B=7; // minimum baffle spacing ,from eq 11.4 ,in
51 PT=1.25;
52 as=((ID*C*B)/(144*PT)); // flow area ,from eq 7.1 ,ft
    ^2
53 printf("\t flow area is : %.4f ft^2 \n",as);
54 Gs=(W/as); // mass velocity ,from eq 7.2 ,lb/(hr)*(ft
    ^2)
55 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
56 mu1=1.84; // at 155F,lb/(ft)*(hr), from fig.14
57 De=0.72/12; // from fig.28 ,ft
58 Res=((De)*(Gs)/mu1); // reynolds number
59 printf("\t reynolds number is : %.2e \n",Res);
60 jH=75; // from fig.28
61 Z=0.575; // Z=(K*(c*mu3/k)^(1/3)),Btu/(hr)(ft^2)(F/
    ft)
62 Ho=((jH)*(1/De)*(Z)); // H0=(h0/phy), using eq.6.15 ,

```

```

        Btu/(hr)*(ft^2)*(F)
63 printf("\t individual heat transfer coefficient is :
        %.0f Btu/(hr)*(ft^2)*(F) \n",Ho);
64 phys=1; // low viscosity
65 ho=(Ho)*(phys); // from eq.6.36
66 printf("\t Correct h0 to the surface at the OD is :
        %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
67 printf("\t cold fluid:inner tube side ,water \n");
68 Nt=140;
69 n=4; // number of passes
70 L=16; //ft
71 at1=0.546; // flow area , in^2
72 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
        .7.48
73 printf("\t flow area is : %.3f ft^2 \n",at);
74 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
75 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
        ",Gt);
76 V=Gt/(3600*62.5);
77 printf("\t V is %.2f fps \n",V);
78 mu2=1.74; // at 100F,lb/(ft)*(hr)
79 D=0.0695; // ft
80 Ret=((D)*(Gt)/mu2); // reynolds number
81 printf("\t reynolds number is : %.2e \n",Ret);
82 hi=1240*0.94; // from fig 25
83 printf("\t Hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
84 ID=0.834; // ft
85 OD=1; //ft
86 hio=((hi)*(ID/OD)); //Hio=(hio/phys), using eq.6.5
87 printf("\t Correct hi0 to the surface at the OD is :
        %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
88 printf("\t pressure drop for annulus \n");
89 f=0.0019; // friction factor for reynolds number
        17400, using fig.29
90 s=1.115; // for reynolds number 17400,using fig.6
91 Ds=19.25/12; // ft
92 De=0.06;
93 N=(12*L/B)+1; // number of crosses ,using eq.7.43

```

```

94 printf("\t number of crosses are : %.0f \n",N);
95 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys))); // using eq.7.44 , psi
96 printf("\t delPs is : %.0f psi \n",delPs);
97 printf("\t allowable delPa is 10 psi \n");
98 printf("\t pressure drop for inner pipe \n");
99 f=0.00018; // friction factor for reynolds number
    46300, using fig.26
100 s=1;
101 phyt=1;
102 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45 , psi
103 printf("\t delPt is : %.1f psi \n",delPt);
104 X1=0.18; // X1=((V^2)/(2*g)), for Gt 1060000, using
    fig.27
105 delPr=((4*n*X1)/(s)); // using eq.7.46 , psi
106 printf("\t delPr is : %.1f psi \n",delPr);
107 delPT=delPt+delPr; // using eq.7.47 , psi
108 printf("\t delPT is : %.1f psi \n",delPT);
109 printf("\t allowable delPa is 10 psi \n");
110 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
111 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
112 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
113 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
114 printf("\t Adjustment of the baffie space to use the
    full 10 psi will still not permit the exchanger
    to make the 0.002 dirt factor. The value of UD
    has been assumed too high \n");
115 printf("\t Try a 21.25in.ID shell with four tube
    passes and a 6in. baffie space. This corresponds
    to 170 tubes \n");
116 UC1=390;
117 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft^2)*(F) \n",UC1);
118 UD2=200;

```

```

119 printf("\t correct design overall coefficient is : %
    .1 f Btu/(hr)*(ft ^2)*(F) \n",UD2);
120 Rd1=0.0024;
121 printf("\t calculated Rd is : %.4 f (hr)*(ft ^2)*(F)/
    Btu \n",Rd1);
122 Rd2=0.002;
123 printf("\t actual Rd is : %.3 f (hr)*(ft ^2)*(F)/Btu \
    n",Rd2);
124 delPs1=9.8;
125 printf("\t delPs is : %.1 f psi \n",delPs1);
126 delPT1=4.9;
127 printf("\t delPt is : %.1 f psi \n",delPT1);
128 //end

```

Scilab code Exa 11.4 Calculation of an Alcohol Heater

```

1 printf("\t example 11.4 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=225; // inlet hot fluid ,F
4 T2=225; // outlet hot fluid ,F
5 t1=80; // inlet cold fluid ,F
6 t2=200; // outlet cold fluid ,F
7 W=10350; // lb/hr
8 w=115000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for steam \n");
11 l=962; // Btu/(lb)
12 Qh=((W)*(l)); // Btu/hr
13 printf("\t total heat required for steam is : %.2e
    Btu/hr \n",Qh);
14 printf("\t for alcohol \n");
15 c=0.72; // Btu/(lb)*(F)
16 Qc=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for alcohol is : %.2e

```

```

        Btu/hr \n",Qc);Q=(Qh+Qc)/(2);
18 Q=(Qh+Qc)/(2);
19 printf("\t Q is : %.2e V \n",Q);
20 delT1=T2-t1; //F
21 delT2=T1-t2; // F
22 printf("\t delT1 is : %.0f F \n",delT1);
23 printf("\t delT2 is : %.0f F \n",delT2);
24 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
25 printf("\t LMTD is :%.1f F \n",LMTD);
26 Tc=((T2)+(T1)); // caloric temperature of hot fluid ,
    F
27 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
28 tc=((t1)+(t2)); // caloric temperature of cold fluid
    ,F
29 printf("\t caloric temperature of cold fluid is : %
    .1f \n",tc);
30 L=12;
31 UD1=200; // assume, from table 8
32 A1=((Q)/((UD1)*(LMTD)));
33 printf("\t A1 is : %.0f ft^2 \n",A1);
34 a1=0.2618; // ft^2/lin ft
35 N1=(A1/(12*a1));
36 printf("\t number of tubes are : %.0f \n",N1);
37 N2=232; // assuming two tube passes,23.25in ID, from
    table 9
38 A2=(N2*L*a1); // ft^2
39 printf("\t total surface area is : %.0f ft^2 \n",A2)
    ;
40 UD=((Q)/((A2)*(LMTD)));
41 printf("\t correct design overall coefficient is : %
    .0f Btu/(hr)*(ft^2)*(F) \n",UD);
42 printf("\t hot fluid:inner tube side ,steam \n");
43 Nt=232;
44 n=2; // number of passes
45 L=12; //ft
46 at1=0.546; // flow area, in^2
47 at=((Nt*at1)/(144*n)); // total area,ft^2,from eq

```



```

    .7.48
48 printf("\t flow area is : %.3f ft^2 \n",at);
49 Gt=(W/(at)); // mass velocity ,lb/(hr)*(ft^2)
50 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
51 mu2=0.0314; // at 225F,lb/(ft)*(hr)
52 D=0.0695; // ft
53 Ret=((D)*(Gt)/mu2); // reynolds number
54 printf("\t reynolds number is : %.1e \n",Ret);
55 hio=1500; // condensation of steam
56 printf("\t Correct hio to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
57 printf("\t cold fluid:shell side ,alcohol \n");
58 ID=23.25; // in
59 C=0.25; // clearance
60 B=7; // minimum baffle spacing ,from eq 11.4,in
61 PT=1.25;
62 as=((ID*C*B)/(144*PT)); // flow area ,from eq 7.1,ft
    ^2
63 printf("\t flow area is : %.3f ft^2 \n",as);
64 Gs=(w/as); // mass velocity ,from eq 7.2,lb/(hr)*(ft
    ^2)
65 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
66 mu1=1.45; // at 193.5F,lb/(ft)*(hr) , from fig.14
67 De=0.72/12; // from fig.28,ft
68 Res=((De)*(Gs)/mu1); // reynolds number
69 printf("\t reynolds number is : %.1e \n",Res);
70 jH=83; // from fig.28
71 Z=0.195; // Z=(K*(c*mu3/k)^(1/3)) ,Btu/(hr)(ft^2)(F/
    ft)
72 Ho=((jH)*(1/De)*(Z)); // H0=(h0/physa) ,using eq.6.15,
    Btu/(hr)*(ft^2)*(F)
73 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Ho);
74 phys=1;
75 ho=(Ho)*(phys); // from eq.6.36
76 printf("\t Correct h0 to the surface at the OD is :

```

```

    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
77 printf("\t pressure drop for inner pipe \n");
78 f=0.000175; // friction factor for reynolds number
    52000, using fig.26
79 s=0.00076;
80 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(1))
    )/(2); // using eq.7.45,psi
81 printf("\t delPt is : %.2f psi \n",delPt);
82 printf("\t delPr is negligible \n");
83 printf("\t allowable delPa is negligible \n");
84 printf("\t pressure drop for annulus \n");
85 f=0.0018; // friction factor for reynolds number
    21000, using fig.29
86 s=0.78; // for reynolds number 21000,using fig.6
87 Ds=1.94; // ft
88 De=0.06;
89 N=(12*L/B); // number of crosses ,using eq.7.43
90 printf("\t number of crosses are : %.0f \n",N);
91 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys))); // using eq.7.44,psi
92 printf("\t delPs is : %.1f psi \n",delPs);
93 printf("\t allowable delPa is 10 psi \n");
94 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
95 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
96 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
97 printf("\t actual Rd is : %.6f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
98 printf("\t This is clearly an instance in which UD
    was assumed too high.It is now a question of how
    much too high. With the aid of the summary it is
    apparent thatin a larger shell a clean overall
    coefficient of about 200 may be expected \n");
99 printf("\t Assume a 27in. ID shell with 2 tube
    passes ,334 tubes and baffies spaced 7in. apart ,
    since the pressure drop increases with the
    diameter of the shell for a given mass velocity.

```

```

    \n");
100 UC1=214;
101 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft ^2)*(F) \n",UC1);
102 UD2=138.5;
103 printf("\t correct design overall coefficient is : %
    .1f Btu/(hr)*(ft ^2)*(F) \n",UD2);
104 Rd1=0.0025;
105 printf("\t calculated Rd is : %.4f (hr)*(ft ^2)*(F)/
    Btu \n",Rd1);
106 Rd2=0.002;
107 printf("\t actual Rd is : %.3f (hr)*(ft ^2)*(F)/Btu \
    n",Rd2);
108 delPs1=0.23;
109 printf("\t delPs is : %.2f psi \n",delPs1);
110 delPT1=7.1;
111 printf("\t delPt is : %.1f psi \n",delPT1);
112 //end

```

Scilab code Exa 11.5 Calculation of a Flue Gas Cooler

```

1 printf("\t example 11.5 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=250; // inlet hot fluid ,F
4 T2=125; // outlet hot fluid ,F
5 t1=80; // inlet cold fluid ,F
6 t2=100; // outlet cold fluid ,F
7 W=41300; // lb/hr
8 w=64500; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for gas \n");
11 c=0.25; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for gas is : %.2e Btu

```

```

    /hr \n",Q);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.2e
    Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 R=((T1-T2)/(t2-t1));
25 printf("\t R is : %.2f \n",R);
26 S=((t2-t1)/(T1-t1));
27 printf("\t S is : %.3f \n",S);
28 printf("\t FT is 0.935 \n"); // from fig 18
29 delT=(0.935*LMTD); // F
30 printf("\t delT is : %.1f F \n",delT);
31 Tc=((T2)+(T1))/(2); // caloric temperature of hot
    fluid ,F
32 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
33 tc=((t1)+(t2))/(2); // caloric temperature of cold
    fluid ,F
34 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
35 UD1=15; // assume, from table 8
36 A1=((Q)/((UD1)*(delT)));
37 printf("\t A1 is : %.0f ft^2 \n",A1);
38 a1=0.2618; // ft^2/lin ft
39 N1=(A1/(12*a1));
40 printf("\t number of tubes are : %.0f \n",N1);
41 N2=358; // assuming 12 tube passes, from table 9
42 L=12;
43 A2=(N2*L*a1); // ft^2
44 printf("\t total surface area is : %.0f ft^2 \n",A2)
    ;

```

```

45 UD=((Q)/((A2)*(delt)));
46 printf("\t correct design overall coefficient is : %
    .0f Btu/(hr)*(ft ^2)*(F) \n",UD);
47 printf("\t When solved in a manner identical with
    the preceding examples and using the smallest
    integral number of bundle crosses (five)
    corresponding to a 28.8in.spacing \n");
48 UC1=22.7;
49 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft ^2)*(F) \n",UC1);
50 UD2=14;
51 printf("\t correct design overall coefficient is : %
    .1f Btu/(hr)*(ft ^2)*(F) \n",UD2);
52 Rd1=0.027;
53 printf("\t calculated Rd is : %.3f (hr)*(ft ^2)*(F)/
    Btu \n",Rd1);
54 Rd1=0.005;
55 printf("\t required Rd is : %.3f (hr)*(ft ^2)*(F)/Btu
    \n",Rd1);
56 delPs1=5.2;
57 printf("\t delPs is : %.1f psi \n",delPs1);
58 delPt1=1.0;
59 printf("\t delPt is : %.1f psi \n",delPt1);
60 printf("\t The first trial is disqualified because
    of failure to meet the required dirt factor and
    the the pressure drop is five times greater than
    the allowable \n");
61 printf("\t This would be unsatisfactory , since gases
    require large inlet connections and the flow
    distribution on the first and third bundle
    crosses would be poor and the conditions of
    allowable pressure drop would still not be met \n
    ");
62 UD1=15; // assume, from table 8
63 A1=((Q)/((UD1)*(delt)));
64 printf("\t A1 is : %.0f ft ^2 \n",A1);
65 a1=0.2618; // ft ^2/lin ft
66 N1=(A1/(12*a1));

```

```

67 printf("\t number of tubes are : %.0f \n",N1);
68 N2=340; // assuming eight tube passes , from table 9
69 A2=(N2*L*a1); // ft^2
70 printf("\t total surface area is : %.2e ft^2 \n",A2)
    ;
71 UD=((Q)/((A2)*(delt)));
72 printf("\t correct design overall coefficient is : %
    .1f Btu/(hr)*(ft^2)*(F) \n",UD);
73 printf("\t hot fluid:shell side ,gas \n");
74 ID=31; // in
75 C=0.25; // clearance
76 B=24; // baffle spacing ,in
77 PT=1.25;
78 as=((ID*C*B)/(144*PT)); // flow area ,from eq 7.1 ,ft
    ^2
79 printf("\t flow area is : %.2f ft^2 \n",as);
80 Gs=(W/as)/(2); // mass velocity ,from eq 7.2 ,lb/(hr)
    *(ft^2)
81 printf("\t mass velocity is : %.0e lb/(hr)*(ft^2) \n
    ",Gs);
82 mu1=0.050; // at 187.5F,lb/(ft)*(hr) , from fig.15
83 De=0.99/12; // from fig.28 ,ft
84 Res=((De)*(Gs)/mu1); // reynolds number
85 printf("\t reynolds number is : %.1e \n",Res);
86 jH=105; // from fig.28
87 k=0.015; // Btu/(hr)(ft^2)( F/ft)
88 Z=0.94; // Z=((c*mu3/k)^(1/3)) ,Btu/(hr)(ft^2)(F/ft)
89 Ho=((jH)*(k/De)*(Z)); // H0=(h0/phyA) ,using eq.6.15 ,
    Btu/(hr)*(ft^2)*(F)
90 printf("\t individual heat transfer coefficient is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",Ho);
91 phys=1;
92 ho=(Ho)*(phys); // from eq.6.36
93 printf("\t Correct h0 to the surface at the OD is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",ho);
94 printf("\t cold fluid:inner tube side ,crude oil \n")
    ;
95 Nt=340;

```

```

96 n=12; // number of passes
97 L=12; //ft
98 at1=0.546; // flow area , in^2
99 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
100 printf("\t flow area is : %.3f ft^2 \n",at);
101 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
102 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
103 V=(Gt/(3600*62.5));
104 printf("\t V is : %.2f fps \n",V);
105 mu2=1.96; // at 90F,lb/(ft)*(hr)
106 D=0.0695; // ft
107 Ret=((D)*(Gt)/mu2); // reynolds number
108 printf("\t reynolds number is : %.2e \n",Ret);
109 hi=667; //Btu/(hr)*(ft^2)*(F)
110 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
111 ID=0.83; // ft
112 OD=1; //ft
113 hio=((hi)*(ID/OD)); //Hio=(hio/phyp), using eq.6.5
114 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio); //
    calculation mistake
115 phyt=1;
116 printf("\t pressure drop for annulus \n");
117 f=0.0017; // friction factor for reynolds number
    33000, using fig.29
118 s=0.0012; // for reynolds number 33000,using fig.6
119 Ds=31/12; // ft
120 N=(3); // number of crosses ,using eq.7.43
121 printf("\t number of crosses are : %.0f \n",N);
122 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys))); // using eq.7.44 ,psi
123 printf("\t delPs is : %.1f psi \n",delPs);
124 printf("\t pressure drop for inner pipe \n");
125 f=0.00022; // friction factor for reynolds number
    21300, using fig.26
126 s=1;

```

```

127 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45 , psi
128 printf("\t delPt is : %.1f psi \n",delPt);
129 X1=0.052; // X1=((V^2)/(2*g)), for Gt 1060000, using
    fig.27
130 delPr=((4*n*X1)/(s)); // using eq.7.46 , psi
131 printf("\t delPr is : %.1f psi \n",delPr);
132 delPT=delPt+delPr; // using eq.7.47 , psi
133 printf("\t delPT is : %.1f psi \n",delPT);
134 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
135 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
136 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
137 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
138 // end

```

Chapter 12

Condensation of single vapor

Scilab code Exa 12.1 Calculation of a Horizontal n Propanol Condenser

```
1 printf("\t example 12.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=244; // inlet hot fluid ,F
4 T2=244; // outlet hot fluid ,F
5 t1=85; // inlet cold fluid ,F
6 t2=120; // outlet cold fluid ,F
7 W=60000; // lb/hr
8 w=488000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for propanol \n");
11 l=285; // Btu/(lb)
12 Q=((W)*(l)); // Btu/hr
13 printf("\t total heat required for propanol is : %.2
   e Btu/hr \n",Q);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.2e
   Btu/hr \n",Q);
18 deltt1=T2-t1; //F
```

```

19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.0f F \n",LMTD);
24 Tc=((T2)+(T1))/(2); // caloric temperature of hot
    fluid ,F
25 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
26 tc=((t1)+(t2))/(2); // caloric temperature of cold
    fluid ,F
27 printf("\t caloric temperature of cold fluid is : %
    .1f F \n",tc);
28 UD1=100; // assume, from table 8
29 A1=((Q)/((UD1)*(LMTD)));
30 printf("\t A1 is : %.0f ft^2 \n",A1);
31 a1=0.1963; // ft^2/lin ft
32 N1=(A1/(8*a1));
33 printf("\t number of tubes are : %.0f \n",N1);
34 N2=766; // assuming 4 tube passes, from table 9
35 A2=(N2*8*a1); // ft^2
36 printf("\t total surface area is : %.0f ft^2 \n",A2)
    ;
37 UD=((Q)/((A2)*(LMTD)));
38 printf("\t correct design overall coefficient is : %
    .0f Btu/(hr)*(ft^2)*(F) \n",UD);
39 printf("\t hot fluid:shell side ,propanol \n");
40 ID=31; // in
41 C=0.1875; // clearance
42 B=31; // baffle spacing ,in
43 PT=0.937;
44 L=8; // ft
45 as=((ID*C*B)/(144*PT)); // flow area ,from eq 7.1 ,ft
    ^2
46 printf("\t flow area is : %.2f ft^2 \n",as);
47 Gs=(W/as); // mass velocity ,from eq 7.2 ,lb/(hr)*(ft
    ^2)
48 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n

```

```

    ",Gs);
49 G1=(W/(L*N2^(2/3))); // from eq.12.43
50 printf("\t G1 is : %.1f lb/(hr)*(lin ft) \n",G1);
51 printf("\t cold fluid:inner tube side ,water \n");
52 Nt=766;
53 n=4; // number of passes
54 L=8; //ft
55 at1=0.302; // flow area , in^2
56 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
57 printf("\t flow area is : %.3f ft^2 \n",at);
58 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
59 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
60 V=(Gt/(3600*62.5));
61 printf("\t V is : %.2f fps \n",V);
62 mu2=1.74; // at 102.5F,lb/(ft)*(hr)
63 D=0.0517; // ft
64 Ret=((D)*(Gt)/mu2); // reynolds number
65 printf("\t reynolds number is : %.2e \n",Ret);
66 hi=1300; //Btu/(hr)*(ft^2)*(F)
67 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
68 ID=0.62; // ft
69 OD=0.75; //ft
70 hio=((hi)*(ID/OD)); // using eq.6.5
71 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio); //
    calculation mistake
72 ho=200; // assumption
73 tw=(tc)+(((ho)/(hio+ho))*(Tc-tc)); // from eq.5.31
74 printf("\t tw is : %.0f F \n",tw);
75 tf=(Tc+tw)/(2); // from eq 12.19
76 printf("\t tf is : %.1f F \n",tf);
77 kf=0.094; // Btu/(hr)*(ft^2)*(F/ft) , from table 4
78 sf=0.8; // from table 6
79 muf=0.62; // cp , from fig 14
80 ho=172; // Btu/(hr)*(ft^2)*(F) , from fig 12.9
81 printf("\t Correct ho to the surface at the OD is :

```

```

    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
82 printf("\t Based on h=172 instead of the assumed 200
    a new value of tw,and tf could be obtained to
    give a more exact value of h based on fluid
    properties at a value of tf more nearly correct \
n");
83 printf("\t pressure drop for annulus \n");
84 mu1=0.0242; // lb/(ft)*(hr), fir 15
85 De=0.0458; // fig 28
86 Res=((De)*(Gs)/mu1); // reynolds number
87 printf("\t reynolds number is : %.2e \n",Res);
88 f=0.00141; // friction factor for reynolds number
    84600, using fig.29
89 s=0.00381; // for reynolds number 84600,using fig.6
90 Ds=31/12; // ft
91 phys=1;
92 N=(3); // number of crosses ,using eq.7.43
93 printf("\t number of crosses are : %.0f \n",N);
94 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys)))/(2); // using eq.12.47 ,psi
95 printf("\t delPs is : %.1f psi \n",delPs);
96 printf("\t allowable delPa is 2 psi \n");
97 printf("\t pressure drop for inner pipe \n");
98 f=0.00019; // friction factor for reynolds number
    36200, using fig.26
99 s=1;
100 phyt=1;
101 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45 ,psi
102 printf("\t delPt is : %.1f psi \n",delPt);
103 X1=0.2; // X1=((V^2)/(2*g)),using fig.27
104 delPr=((4*n*X1)/(s)); // using eq.7.46 ,psi
105 printf("\t delPr is : %.1f psi \n",delPr);
106 delPT=delPt+delPr; // using eq.7.47 ,psi
107 printf("\t delPT is : %.1f psi \n",delPT);
108 printf("\t allowable delPT is 10 psi \n");
109 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)

```

```

110 printf("\t clean overall coefficient is : %.1f Btu/(
      hr)*(ft ^2)*(F) \n",Uc);
111 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft ^2)*(F)/Btu
112 printf("\t actual Rd is : %.4f (hr)*(ft ^2)*(F)/Btu \
      n",Rd);
113 // end

```

Scilab code Exa 12.2 Design of a Vertical n Propanol Condenser

```

1 printf("\t example 12.2 \n");
2 printf("\t approximate values are mentioned in the
      book \n");
3 T1=244; // inlet hot fluid ,F
4 T2=244; // outlet hot fluid ,F
5 t1=85; // inlet cold fluid ,F
6 t2=120; // outlet cold fluid ,F
7 W=60000; // lb/hr
8 w=488000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for propanol \n");
11 l=285; // Btu/(lb)
12 Q=((W)*(l)); // Btu/hr
13 printf("\t total heat required for propanol is : %.2
      e Btu/hr \n",Q);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.2e
      Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.0f F \n",LMTD);

```

```

24 Tc=((T2)+(T1))/(2); // caloric temperature of hot
    fluid ,F
25 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
26 tc=((t1)+(t2))/(2); // caloric temperature of cold
    fluid ,F
27 printf("\t caloric temperature of cold fluid is : %
    .1f F \n",tc);
28 UD1=70; // assume, from table 8
29 A1=((Q)/((UD1)*(LMTD)));
30 printf("\t A1 is : %.2e ft^2 \n",A1);
31 N2=766; // assuming 4 tube passes, from table 9
32 a1=0.1963; // ft^2/lin ft
33 L=(A1/(N2*a1));
34 printf("\t L is : %.1f ft \n",L);
35 A2=(N2*12*a1); // ft^2
36 printf("\t total surface area is : %.0f ft^2 \n",A2)
    ;
37 UD=((Q)/((A2)*(LMTD)));
38 printf("\t correct design overall coefficient is : %
    .1f Btu/(hr)*(ft^2)*(F) \n",UD);
39 printf("\t hot fluid:shell side ,propanol \n");
40 Do=0.0625; // ft
41 G1=(W/(3.14*N2*Do)); // from eq.12.36
42 printf("\t G1 is : %.0f lb/(hr)*(lin ft) \n",G1);
43 printf("\t cold fluid:inner tube side ,water \n");
44 Nt=766;
45 n=4; // number of passes
46 L=12; //ft
47 at1=0.302; // flow area, in^2
48 at=((Nt*at1)/(144*n)); // total area, ft^2,from eq
    .7.48
49 printf("\t flow area is : %.3f ft^2 \n",at);
50 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
51 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
52 V=(Gt/(3600*62.5));
53 printf("\t V is : %.2f fps \n",V);

```

```

54 mu2=1.74; // at 102.5F, lb/(ft)*(hr)
55 D=0.0517; // ft
56 Ret=((D)*(Gt)/mu2); // reynolds number
57 printf("\t reynolds number is : %.2e \n",Ret);
58 hi=1300; //Btu/(hr)*(ft^2)*(F)
59 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
60 ID=0.62; // ft
61 OD=0.75; //ft
62 hio=((hi)*(ID/OD)); // using eq.6.5
63 printf("\t Correct hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
64 ho=100; // assumption
65 tw=(tc)+(((ho)/(hio+ho))*(Tc-tc)); // from eq.5.31
66 printf("\t tw is : %.1f F \n",tw);
67 tf=(Tc+tw)/(2); // from eq 12.19
68 printf("\t tf is : %.0f F \n",tf);
69 kf=0.0945; // Btu/(hr)*(ft^2)*(F/ft), from table 4
70 sf=0.76; // from table 6
71 muf=0.65; // cp, from fig 14
72 ho=102; // Btu/(hr)*(ft^2)*(F), from fig 12.9
73 printf("\t Correct ho to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
74 printf("\t pressure drop for annulus \n");
75 ID=31; // in
76 C=0.1875; // clearance
77 B=29; // baffle spacing,in
78 PT=0.937;
79 as=((ID*C*B)/(144*PT)); // flow area ,from eq 7.1,ft
    ^2
80 printf("\t flow area is : %.2f ft^2 \n",as);
81 Gs=(W/as); // mass velocity ,from eq 7.2,lb/(hr)*(ft
    ^2)
82 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
83 mu1=0.0242; // lb/(ft)*(hr), fig 15
84 De=0.0458; // fig 28
85 Res=((De)*(Gs)/mu1); // reynolds number
86 printf("\t reynolds number is : %.1e \n",Res);

```

```

87 f=0.0014; // friction factor for reynolds number
    91000, using fig.29
88 s=0.00381; // for reynolds number 91000,using fig.6
89 Ds=31/12; // ft
90 phys=1;
91 N=(5); // number of crosses ,using eq.7.43
92 printf("\t number of crosses are : %.0f \n",N);
93 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys)))/(2); // using eq.12.47, psi
94 printf("\t delPs is : %.1f psi \n",delPs);
95 printf("\t allowable delPa is 2 psi \n");
96 printf("\t pressure drop for inner pipe \n");
97 f=0.00019; // friction factor for reynolds number
    36200, using fig.26
98 s=1;
99 phyt=1;
100 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45, psi
101 printf("\t delPt is : %.1f psi \n",delPt);
102 X1=0.2; // X1=((V^2)/(2*g)), using fig.27
103 delPr=((4*n*X1)/(s)); // using eq.7.46, psi
104 printf("\t delPr is : %.1f psi \n",delPr);
105 delPT=delPt+delPr; // using eq.7.47, psi
106 printf("\t delPT is : %.1f psi \n",delPT);
107 printf("\t allowable delPT is 10 psi \n");
108 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,eq 6.38, Btu/(hr)*(ft^2)*(F)
109 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
110 Rd=((Uc-UD)/((UD)*(Uc))); // eq 6.13, (hr)*(ft^2)*(F)
    /Btu
111 printf("\t actual Rd is : %.5f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
112 // end

```

Scilab code Exa 12.3 Calculation of a Butane Desuperheater condenser

```
1 printf("\t example 12.3 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=200; // inlet hot fluid ,F
4 T2=130; // outlet hot fluid ,F
5 T3=125; // after condensation
6 t1=65; // inlet cold fluid ,F
7 t3=100; // outlet cold fluid ,F
8 W=27958; // lb/hr
9 w=135500; // lb/hr
10 printf("\t 1.for heat balance \n");
11 printf("\t for butane \n");
12 c=0.44; // Btu/(lb)(F)
13 qd=((W)*(c)*(T1-T2)); // Btu/hr
14 printf("\t total heat required for desuperheating of
   butane is : %.1e Btu/hr \n",qd);
15 HT2=309; // enthalpy at T2, Btu/lb
16 HT3=170; // enthalpy at T3, Btu/lb
17 qc=(W*(HT2-HT3)); // for condensation
18 printf("\t total heat required for condensing of
   butane is : %.2e Btu/hr \n",qc);
19 Q=qd+qc;
20 printf("\t total heat required for butane is : %.2e
   Btu/hr \n",Q);
21 printf("\t for water \n");
22 c=1; // Btu/(lb)*(F)
23 Q=((w)*(c)*(t3-t1)); // Btu/hr
24 printf("\t total heat required for water is : %.2e
   Btu/hr \n",Q);
25 deltw=(qc/w);
26 printf("\t deltw is : %.1f F \n",deltw);
27 t2=t1+deltw;
28 printf("\t t2 is : %.1f F \n",t2)
29 printf("\t for desuperheating \n");
30 deltt1=T2-t2; //F
31 deltt2=T1-t3; // F
```

```

32 printf("\t delt1 is : %.0f F \n",delt1);
33 printf("\t delt2 is : %.0f F \n",delt2);
34 LMTDd=((delt2-delt1)/((2.3)*(log10(delt2/delt1))));
35 printf("\t LMTD is :%.0f F \n",LMTDd);
36 w1=(qd/LMTDd);
37 printf("\t w1 is : %.3e lb/hr \n",w1);
38 printf("\t for condensing \n");
39 delt3=T3-t1; //F
40 delt4=T2-t2; // F
41 printf("\t delt1 is : %.0f F \n",delt3);
42 printf("\t delt2 is : %.0f F \n",delt4);
43 LMTDc=((delt4-delt3)/((2.3)*(log10(delt4/delt3))));
44 printf("\t LMTD is :%.0f F \n",LMTDc);
45 w2=(qc/LMTDc);
46 printf("\t w1 is : %.2e lb/hr \n",w2);
47 delt=(Q/(w1+w2));
48 printf("\t delt is : % .1f F \n",delt);
49 Tc=((T3)+(T2))/(2); // caloric temperature of hot
    fluid ,F
50 printf("\t caloric temperature of hot fluid is : %.1
    f F \n",Tc);
51 tc=((t1)+(t3))/(2); // caloric temperature of cold
    fluid ,F
52 printf("\t caloric temperature of cold fluid is : %
    .1f F \n",tc);
53 printf("\t hot fluid:shell side ,butane \n");
54 ID=23.25; // in
55 C=0.25; // clearance
56 B=12; // baffle spacing ,in
57 PT=1;
58 as=((ID*C*B)/(144*PT)); // flow area ,ft^2
59 printf("\t flow area is : %.3f ft^2 \n",as);
60 printf("\t desuperheating \n");
61 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2)
62 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
63 mu1=0.0242; // at 165F,lb/(ft)*(hr), from fig.15
64 De=0.73/12; // from fig.28,ft

```

```

65 Res=((De)*(Gs)/mu1); // reynolds number
66 printf("\t reynolds number is : %.2e \n",Res);
67 jH=239; // from fig.28
68 k=0.012; // Btu/(hr)*(ft^2)*(F/ft), from table 5
69 Z=0.96; // Z=((c)*(mu1)/k)^(1/3)
70 ho=((jH)*(k/De)*(Z)); // H0=(h0/phyA), using eq.6.15b
    ,Btu/(hr)*(ft^2)*(F)
71 printf("\t individual heat transfer coefficient is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",ho);
72 printf("\t cold fluid:inner tube side ,water \n");
73 Nt=352;
74 n=4; // number of passes
75 L=16; //ft
76 at1=0.302; // flow area ,table 10, in^2
77 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
78 printf("\t flow area is : %.3f ft^2 \n",at);
79 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
80 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
81 V=(Gt/(3600*62.5));
82 printf("\t V is : %.2f fps \n",V);
83 mu2=2.11; // at 82.5F, fig 14,lb/(ft)*(hr)
84 D=0.0517; // ft
85 Ret=((D)*(Gt)/mu2); // reynolds number
86 printf("\t reynolds number is : %.2e \n",Ret);
87 hi=800; // fig 25,Btu/(hr)*(ft^2)*(F)
88 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
89 ID=0.62; // ft
90 OD=0.75; //ft
91 hio=((hi)*(ID/OD)); // using eq.6.5
92 printf("\t Correct hio to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
93 Ud=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,eq 6.38,Btu/(hr)*(ft^2)*(F)
94 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft^2)*(F) \n",Ud);
95 Ad=(qd/(Ud*LMTDd));

```

```

96 printf("\t clean surface required for desuperheating
      : %.0f ft^2 \n",Ad);
97 printf("\t for condensaton \n");
98 Lc=16*0.6; // condensation occurs 60% of the tube
      length
99 G1=(W/(Lc*Nt^(2/3))); // from eq.12.43
100 printf("\t G1 is : %.1f lb/(hr)*(lin ft) \n",G1);
101 ho=200; // assumption
102 tw=(tc)+(((ho)/(hio+ho))*(Tc-tc)); // from eq.5.31
103 printf("\t tw is : %.0f F \n",tw);
104 tf=(Tc+tw)/(2); // from eq 12.19
105 printf("\t tf is : %.0f F \n",tf);
106 kf=0.075; // Btu/(hr)*(ft^2)*(F/ft)
107 sf=0.55; // from table 6
108 muf=0.14; // cp, from fig 14
109 ho=207; // Btu/(hr)*(ft^2)*(F), from fig 12.9
110 printf("\t Correct ho to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
111 Uc=((hio)*(ho)/(hio+ho)); // clean overall
      coefficient ,Btu/(hr)*(ft^2)*(F)
112 printf("\t clean overall coefficient is : %.0f Btu/(
      hr)*(ft^2)*(F) \n",Uc);
113 Ac=(qc/(Uc*LMTDc));
114 printf("\t clean surface required for desuperheating
      : %.0f ft^2 \n",Ac);
115 AC=Ad+Ac;
116 printf("\t total clean surface : %.0f ft^2 \n",AC);
117 lc=(Ac/(Ac+Ad));
118 printf("\t assumed condensing length percentage : %
      .2f \n",lc);
119 UC=((Ud*Ad)+(Uc*Ac))/(AC);
120 printf("\t weighted clean overall coefficient : %.0f
      Btu/(hr)*(ft^2)*(F) \n",UC);
121 A2=0.1963; // actual surface supplied for each tube,
      ft^2,from table 10
122 A=(Nt*L*A2); // ft^2
123 printf("\t total surface area is : %.0f ft^2 \n",A);
124 UD=((Q)/((A)*(delt)));

```

```

125 printf("\t actual design overall coefficient is : %
      .0f Btu/(hr)*(ft^2)*(F) \n",UD);
126 Rd=((UC-UD)/((UD)*(UC))); // (hr)*(ft^2)*(F)/Btu
127 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
      n",Rd);
128 printf("\t pressure drop for annulus \n");
129 printf("\t desuperheating \n");
130 Ld=6.4; //ft
131 De=0.0608; // fig 28
132 f=0.0013; // friction factor for reynolds number
      145000, using fig.29
133 Ds=1.94; // ft
134 phys=1;
135 N=(12*Ld/B); // number of crosses ,using eq.7.43
136 printf("\t number of crosses are : %.0f \n",N);
137 row=(58.1/((359)*(625/492)*(14.7/99.7)));
138 printf("\t row is %.3f lb/ft^3 \n",row);
139 s=(row/62.5);
140 printf("\t s is %.4f \n",s);
141 delPsd=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
      phys))); // using eq.7.44 ,psi
142 printf("\t delPs is : %.1f psi \n",delPsd);
143 printf("\t condensation \n");
144 N=(12*Lc/B); // number of crosses ,using eq.7.43
145 printf("\t number of crosses are : %.0f \n",N);
146 delPsc=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
      phys)))/(2); // using eq 12.47 ,psi
147 printf("\t delPsc is : %.1f psi \n",delPsc);
148 delPS=delPsd+delPsc;
149 printf("\t delPS is : %.0f psi \n",delPS);
150 printf("\t allowable delPa is 2 psi \n");
151 printf("\t pressure drop for inner pipe \n");
152 f=0.00023; // friction factor for reynolds number
      17900, using fig.26
153 s=1;
154 phyt=1;
155 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
      phyt))); // using eq.7.45 ,psi

```

```

156 printf("\t delPt is : %.0f psi \n",delPt);
157 X1=0.075; // X1=((V^2)/(2*g)), using fig.27
158 delPr=((4*n*X1)/(s)); // using eq.7.46, psi
159 printf("\t delPr is : %.1f psi \n",delPr);
160 delPT=delPt+delPr; // using eq.7.47, psi
161 printf("\t delPT is : %.1f psi \n",delPT);
162 printf("\t allowable delPa is 10 psi \n");
163 //end

```

Scilab code Exa 12.4 Calculation of a Vertical Condenser subcooler

```

1 printf("\t example 12.4 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=130; // inlet hot fluid ,F
4 T2=125; // outlet hot fluid ,F
5 T3=100; // after sucooling
6 t1=80; // inlet cold fluid ,F
7 t3=100; // outlet cold fluid ,F
8 W=21000; // lb/hr
9 w=167000; // lb/hr
10 printf("\t 1.for heat balance \n");
11 printf("\t for pentane \n");
12 HT1=315; // enthalpy at T1, Btu/lb
13 HT2=170; // enthalpy at T2, Btu/lb
14 qc=(W*(HT1-HT2)); // for condensation
15 printf("\t total heat required for condensing of
   pentane is : %.2e Btu/hr \n",qc);
16 c=0.57; // Btu/(lb)(F)
17 qs=((W)*(c)*(T2-T3)); // Btu/hr
18 printf("\t total heat required for subcooling of
   pentane is : %.0e Btu/hr \n",qs);
19 Q=qs+qc;
20 printf("\t total heat required for pentane is : %.2
   e Btu/hr \n",Q);

```

```

21 printf("\t for water \n");
22 c=1; // Btu/(lb)*(F)
23 Q=((w)*(c)*(t3-t1)); // Btu/hr
24 printf("\t total heat required for water is : %.2e
      Btu/hr \n",Q);
25 deltw=(qc/w);
26 printf("\t deltw is : %.1f F \n",deltw);
27 t2=t3-deltw;
28 printf("\t t2 is : %.1f F \n",t2)
29 printf("\t for condensing \n");
30 deltt1=T2-t2; //F
31 deltt2=T1-t3; // F
32 printf("\t deltt1 is : %.0f F \n",deltt1);
33 printf("\t deltt2 is : %.0f F \n",deltt2);
34 LMTDc=((deltt2-deltt1)/((2.3)*(log10(deltt2/deltt1))));
35 printf("\t LMTD is :%.1f F \n",LMTDc);
36 w1=(qc/LMTDc);
37 printf("\t w1 is : %.2e lb/hr \n",w1);
38 printf("\t subcooling \n");
39 deltt3=T3-t1; //F
40 deltt4=T2-t2; // F
41 printf("\t deltt1 is : %.0f F \n",deltt3);
42 printf("\t deltt2 is : %.0f F \n",deltt4);
43 LMTDs=((deltt4-deltt3)/((2.3)*(log10(deltt4/deltt3))));
44 printf("\t LMTD is :%.1f F \n",LMTDs);
45 w2=(qs/LMTDs);
46 printf("\t w1 is : %.2e lb/hr \n",w2);
47 deltt=(Q/(w1+w2));
48 printf("\t deltt is : % .1f F \n",deltt);
49 Tc=((T1)+(T2))/(2); // caloric temperature of hot
      fluid ,F
50 printf("\t caloric temperature of hot fluid is : %.1
      f F \n",Tc);
51 tc=((t1)+(t3))/(2); // caloric temperature of cold
      fluid ,F
52 printf("\t caloric temperature of cold fluid is : %
      .0f F \n",tc);
53 printf("\t hot fluid:shell side ,pentane \n");

```

```

54 printf("\t for condensaton \n");
55 Do=0.0625; // ft
56 Nt=370; // number of tubes
57 G1=(W/(3.14*Nt*Do)); // from eq.12.42
58 printf("\t G1 is : %.1e lb/(hr)*(lin ft) \n",G1);
59 printf("\t cold fluid:inner tube side,water \n");
60 n=4; // number of passes
61 L=16; //ft
62 at1=0.302; // flow area, in^2
63 at=((Nt*at1)/(144*n)); // total area,ft^2,from eq
    .7.48
64 printf("\t flow area is : %.3f ft^2 \n",at);
65 Gt=(w/(at)); // mass velocity,lb/(hr)*(ft^2)
66 printf("\t mass velocity is : %.1e lb/(hr)*(ft^2) \n
    ",Gt);
67 V=(Gt/(3600*62.5));
68 printf("\t V is : %.2f fps \n",V);
69 mu2=1.98; // at 90F,lb/(ft)*(hr)
70 D=0.0517; // ft
71 Ret=((D)*(Gt)/mu2); // reynolds number
72 printf("\t reynolds number is : %.2e \n",Ret);
73 hi=940; //Btu/(hr)*(ft^2)*(F)
74 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
75 ID=0.62; // ft
76 OD=0.75; //ft
77 hio=((hi)*(ID/OD)); // using eq.6.5
78 printf("\t Correct hio to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
79 ho=125; // assumption
80 tw=(tc)+(((ho)/(hio+ho))*(Tc-tc)); // from eq.5.31
81 printf("\t tw is : %.0f F \n",tw);
82 tf=(Tc+tw)/(2); // from eq 12.19
83 printf("\t tf is : %.0f F \n",tf);
84 kf=0.077; // Btu/(hr)*(ft^2)*(F/ft), table 4
85 sf=0.6; // from table 6
86 muf=0.19; // cp, from fig 14
87 ho=120; // Btu/(hr)*(ft^2)*(F), from fig 12.9
88 printf("\t Correct ho to the surface at the OD is :

```



```

    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
89 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
90 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
91 Ac=(3040000/(104*36.4));
92 printf("\t clean surface required for dcondensation
    : %.0f ft^2 \n",Ac);
93 printf("\t subcooling \n");
94 ID=25; // in
95 C=0.25; // clearance
96 B=12; // baffle spacing ,in
97 PT=1;
98 as=((ID*C*B)/(144*PT)); // flow area ,ft^2
99 printf("\t flow area is : %.3f ft^2 \n",as);
100 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2)
101 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
102 mu1=0.46; // at 112.5F,lb/(ft)*(hr), from fig.14
103 De=0.95/12; // from fig.28,ft
104 Res=((De)*(Gs)/mu1); // reynolds number
105 printf("\t reynolds number is : %.2e \n",Res);
106 jH=46.5; // from fig.28
107 k=0.077; // Btu/(hr)*(ft^2)*(F/ft), from table 4
108 Z=1.51; // Z=((c)*(mu1)/k)^(1/3)
109 ho=((jH)*(k/De)*(Z)); // using eq.6.15b,Btu/(hr)*(ft
    ^2)*(F)
110 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
111 Us=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
112 printf("\t clean overall coefficient is : %.1f Btu/(
    hr)*(ft^2)*(F) \n",Us);
113 As=(qs/(Us*LMTDs));
114 printf("\t clean surface required for desuperheating
    : %.1f ft^2 \n",As);
115 AC=As+Ac;
116 printf("\t total clean surface : %.0f ft^2 \n",AC);

```

```

117 UC=((Us*As)+(Uc*Ac))/(AC);
118 printf("\t weighted clean overall coefficient : %.1f
      Btu/(hr)*(ft^2)*(F) \n",UC);
119 A2=0.1963; // actual surface supplied for each tube,
      ft^2,from table 10
120 A=(Nt*L*A2); // ft^2
121 printf("\t total surface area is : %.0f ft^2 \n",A);
122 UD=((Q)/((A)*(delt)));
123 printf("\t actual design overall coefficient is : %
      .1f Btu/(hr)*(ft^2)*(F) \n",UD);
124 Rd=((UC-UD)/((UD)*(UC))); // (hr)*(ft^2)*(F)/Btu
125 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
      n",Rd);
126 printf("\t pressure drop for annulus \n");
127 printf("\t condensation \n");
128 Lc=13.4; //ft
129 De=0.0792; // fig 28
130 f=0.0012; // friction factor for reynolds number
      193000, using fig.29
131 mu3=0.0165; // at 127.5F
132 Ds=2.08; // ft
133 phys=1;
134 Res1=(De*Gs/mu3);
135 printf("\t reynolds number is %.2e \n",Res1);
136 rowvap=(72.2/((359)*(590/492)*(14.7/25)));
137 printf("\t rowvapour is %.3f ld/ft^3 \n",rowvap);
138 s=(rowvap/62.5);
139 printf("\t s is %.5f \n",s);
140 N=(12*Lc/B)+(1); // number of crosses,using eq.7.43
141 printf("\t number of crosses are : %.0f \n",N);
142 delPsc=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
      phys)))/(2); // using eq.12.47, psi
143 printf("\t delPsc is : %.1f psi \n",delPsc);
144 printf("\t delPss is negligible \n");
145 printf("\t allowable delPa is 2 psi \n");
146 printf("\t pressure drop for inner pipe \n");
147 f=0.00022; // friction factor for reynolds number
      22500, using fig.26

```

```

148 s=1;
149 phyt=1;
150 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45 , psi
151 printf("\t delPt is : %.1f psi \n",delPt);
152 X1=0.1; // X1=((V^2)/(2*g)), using fig.27
153 delPr=((4*n*X1)/(s)); // using eq.7.46 , psi
154 printf("\t delPr is : %.1f psi \n",delPr);
155 delPT=delPt+delPr; // using eq.7.47 , psi
156 printf("\t delPT is : %.1f psi \n",delPT);
157 printf("\t allowable delPT is 10 psi \n");
158 //end

```

Scilab code Exa 12.5 Calculation of a Horizontal Condenser subcooler

```

1 printf("\t example 12.5 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=130; // inlet hot fluid ,F
4 T2=125; // outlet hot fluid ,F
5 T3=100; // after subcooling
6 t1=80; // inlet cold fluid ,F
7 t3=100; // outlet cold fluid ,F
8 W=21000; // lb/hr
9 w=167000; // lb/hr
10 printf("\t 1.for heat balance \n");
11 printf("\t for pentane \n");
12 c=0.57; // Btu/(lb)(F)
13 qs=((W)*(c)*(T2-T3)); // Btu/hr
14 printf("\t total heat required for subcooling of
    pentane is : %.0e Btu/hr \n",qs);
15 HT1=315; // enthalpy at T1, Btu/lb
16 HT2=170; // enthalpy at T2, Btu/lb
17 qc=(W*(HT1-HT2)); // for condensation
18 printf("\t total heat required for condensing of

```

```

    pentane is : %.2e Btu/hr \n",qc);
19 Q=qs+qc;
20 printf("\t total heat required for pentane is : %.2
    e Btu/hr \n",Q);
21 printf("\t for water \n");
22 c=1; // Btu/(lb)*(F)
23 Q=((w)*(c)*(t3-t1)); // Btu/hr
24 printf("\t total heat required for water is : %.2e
    Btu/hr \n",Q);
25 deltw=18.2;
26 printf("\t deltw is : %.1f F \n",deltw);
27 t2=t3-deltw;
28 printf("\t t2 is : %.1f F \n",t2)
29 printf("\t for condensing \n");
30 deltt1=T2-t2; //F
31 deltt2=T1-t3; // F
32 printf("\t deltt1 is : %.0f F \n",deltt1);
33 printf("\t deltt2 is : %.0f F \n",deltt2);
34 LMTDc=((deltt2-deltt1)/((2.3)*(log10(deltt2/deltt1))));
35 printf("\t LMTD is :%.1f F \n",LMTDc);
36 w1=(qc/LMTDc);
37 printf("\t w1 is : %.2e lb/hr \n",w1);
38 printf("\t subcooling \n");
39 deltt3=T3-t1; //F
40 deltt4=T2-t2; // F
41 printf("\t deltt1 is : %.0f F \n",deltt3);
42 printf("\t deltt2 is : %.0f F \n",deltt4);
43 LMTDs=((deltt4-deltt3)/((2.3)*(log10(deltt4/deltt3))));
44 printf("\t LMTD is :%.1f F \n",LMTDs);
45 w2=(qs/LMTDs);
46 printf("\t w1 is : %.2e lb/hr \n",w2);
47 deltt=(Q/(w1+w2));
48 printf("\t deltt is : % .1f F \n",deltt);
49 Tc=((T1)+(T2))/(2); // caloric temperature of hot
    fluid ,F
50 printf("\t caloric temperature of hot fluid is : %.1
    f F \n",Tc);
51 tc=((t1)+(t3))/(2); // caloric temperature of cold

```

```

fluid ,F
52 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
53 printf("\t hot fluid:shell side ,pentane \n");
54 C1=0.198; // for 0.3Ds
55 Ds=25; // in
56 L=16; // ft
57 N=370
58 a=(C1*Ds^2);
59 printf("\t a is : %.0f in^2 \n",a);
60 N1=((N*a*4)/(3.14*Ds^2));
61 printf("\t number of submerged tubes are : %.0f \n",
    N1);
62 Nt=N-N1;
63 printf("\t number of tubes for condensation are : %
    .0f \n",Nt);
64 Af=(N1/N);
65 printf("\t flooded surface : %.2f \n",Af);
66 printf("\t for condensaton \n");
67 G1=(W/(L*Nt^(2/3))); // from eq.12.43
68 printf("\t G1 is : %.1f lb/(hr)*(lin ft) \n",G1);
69 printf("\t cold fluid:inner tube side ,water \n");
70 n=4; // number of passes
71 L=16; //ft
72 at1=0.302; // flow area , in^2
73 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
74 printf("\t flow area is : %.3f ft^2 \n",at);
75 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
76 printf("\t mass velocity is : %.1e lb/(hr)*(ft^2) \n
    ",Gt);
77 V=(Gt/(3600*62.5));
78 printf("\t V is : %.2f fps \n",V);
79 mu2=1.98; // lb/(ft)*(hr)
80 D=0.0517; // ft
81 Ret=((D)*(Gt)/mu2); // reynolds number
82 printf("\t reynolds number is : %.2e \n",Ret);
83 hi=940; //Btu/(hr)*(ft^2)*(F)

```

```

84 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
85 ID=0.62; // ft
86 OD=0.75; //ft
87 hio=((hi)*(ID/OD)); // using eq.6.5
88 printf("\t Correct hio to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",hio);
89 ho=251; // Btu/(hr)*(ft^2)*(F), from fig 12.9
90 printf("\t Correct ho to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
91 Uc=((hio)*(ho)/(hio+ho)); // clean overall
      coefficient ,Btu/(hr)*(ft^2)*(F)
92 printf("\t clean overall coefficient is : %.0f Btu/(
      hr)*(ft^2)*(F) \n",Uc);
93 Ac=(qc/(Uc*LMTDc));
94 printf("\t clean surface required for dcondensation
      : %.0f ft^2 \n",Ac);
95 printf("\t subcooling \n");
96 ho=50; // Btu/(hr)*(ft^2)*(F)
97 printf("\t individual heat transfer coefficient is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
98 Us=((hio)*(ho)/(hio+ho)); // clean overall
      coefficient ,Btu/(hr)*(ft^2)*(F)
99 printf("\t clean overall coefficient is : %.0f Btu/(
      hr)*(ft^2)*(F) \n",Us);
100 As=(qs/(Us*LMTDs));
101 printf("\t clean surface required for desuperheating
      : %.0f ft^2 \n",As);
102 AC=As+Ac;
103 printf("\t total clean surface : %.0f ft^2 \n",AC);
104 UC=((Us*As)+(Uc*Ac))/(AC);
105 printf("\t weighted clean overall coefficient : %.0f
      Btu/(hr)*(ft^2)*(F) \n",UC);
106 A=1160; // ft^2
107 printf("\t total surface area is : %.0f ft^2 \n",A);
108 UD=((Q)/((A)*(delt)));
109 printf("\t actual design overall coefficient is : %
      .1f Btu/(hr)*(ft^2)*(F) \n",UD);
110 Rd=((UC-UD)/((UD)*(UC))); // (hr)*(ft^2)*(F)/Btu

```

```

111 printf("\t actual Rd is : %.4f (hr)*(ft ^2)*(F)/Btu \
    n",Rd);
112 printf("\t pressure drop for annulus \n");
113 printf("\t condensation \n");
114 printf("\t It will be necessary to spread the batHes
    to a spacing of 18in.to compensate for the
    reduction in crossfiow area due to the flooded
    subcooling zone. The tube-side pressure drop will
    be the same as before. Assume bundle flooded to
    0.3Ds.\n");
115 As=0.547; // ft^2
116 Gs=(W/(As)); // mass velocity ,lb/(hr)*(ft ^2)
117 printf("\t mass velocity is : %.1e lb/(hr)*(ft ^2) \n
    ",Gs);
118 De=0.0792; // fig 28
119 Res=((De)*(Gs)/0.0165); // reynolds number
120 printf("\t reynolds number is : %.2e \n",Res);
121 f=0.00121; // friction factor for reynolds number
    193000, using fig.29
122 s=0.00454; // for reynolds number 193000,using fig.6
123 Ds=2.08; // ft
124 B=18
125 phys=1;
126 N=(12*L/B); // number of crosses ,using eq.7.43
127 printf("\t number of crosses are : %.0f \n",N);
128 delPsc=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys)))/(2); // using eq.12.47 ,psi
129 printf("\t delPsc is : %.1f psi \n",delPsc);
130 printf("\t delPss is negligible \n");
131 printf("\t allowable delPa is 2 psi \n");
132 //end

```

Scilab code Exa 12.6 Calculation of Vertical Reflux type CSo Condenser

```

1 printf("\t example 12.6 \n");

```

```

2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=176; // inlet hot fluid ,F
4 T2=176; // outlet hot fluid ,F
5 t1=85; // inlet cold fluid ,F
6 t2=120; // outlet cold fluid ,F
7 W=30000; // lb/hr
8 w=120000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for carbon disulfide \n");
11 l=140; // Btu/(lb)
12 Q=((W)*l); // Btu/hr
13 printf("\t total heat required for carbon disulfide
   is : %.1e Btu/hr \n",Q);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.0f
   Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.1f F \n",LMTD);
24 Tc=((T2)+T1)/2; // caloric temperature of hot fluid ,
   F
25 printf("\t caloric temperature of hot fluid is : %.0
   f F \n",Tc);
26 tc=((t1)+(t2))/2; // caloric temperature of cold
   fluid ,F
27 printf("\t caloric temperature of cold fluid is : %
   .1f F \n",tc);
28 printf("\t hot fluid:inner tube side ,carbon
   disulfide \n");
29 hio=300; // Btu/(hr)*(ft^2)*(F)
30 printf("\t cold fluid:shell side ,water \n");
31 ID=17.25; // in

```



```

32 C=0.25; // clearance
33 B=6; // baffle spacing,in
34 PT=1;
35 as=((ID*C*B)/(144*PT)); // flow area,ft^2
36 printf("\t flow area is : %.2f ft^2 \n",as);
37 Gs=(w/as); // mass velocity,lb/(hr)*(ft^2)
38 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
39 mu1=1.7; // at 280F,lb/(ft)*(hr), from fig.14
40 De=0.0792; // from fig.28,ft
41 Res=((De)*(Gs)/mu1); // reynolds number
42 printf("\t reynolds number is : %.1e \n",Res);
43 jH=103; // from fig.28
44 k=0.36; // Btu/(hr)*(ft^2)*(F/ft), from fig.1
45 Z=1.68; // Z=((c)*(mu1)/k)^(1/3); // prandelt number
46 ho=((jH)*(k/De)*(Z)); // using eq.6.15,Btu/(hr)*(ft
    ^2)*(F)
47 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
48 tw=(tc)+(((hio)/(hio+ho))*(Tc-tc)); // from eq.5.31
49 printf("\t tw is : %.1f F \n",tw);
50 tf=(Tc+tw)/(2); // from eq 12.19
51 printf("\t tf is : %.1f F \n",tf);
52 printf("\t hot fluid:inner tube side ,carbon
    disulfide \n");
53 kf=0.09; // Btu/(hr)*(ft^2)*(F/ft), from fig 14
54 sf=1.26; // from table 6
55 rowf=78.8; // lb/ft^3
56 muf=0.68; // cp, from fig 24
57 Nt=177;
58 D=0.0517; // ft
59 G1=(W/(3.14*Nt*D));
60 printf("\t G1 is : %.f lb/(hr)*(lin ft) \n",G1);
61 Ret=((4)*(G1)/muf); // reynolds number
62 printf("\t reynolds number is : %.0f \n",Ret);
63 hi=(0.251*(((kf^3)*(rowf^2)*(4.17*10^8))/(muf^2))
    ^((1/3))); // hi*(((kf^3)*(rowf^2)*(4.17*10^8))/(
    muf^2))^(-1)=0.251, from fig 12.12

```

```

64 printf("\t hi is : %.0e Btu/(hr)*(ft^2)*(F) \n",hi);
65 ID=0.62; // ft
66 OD=.75; //ft
67 hio1=((hi)*(ID/OD)); //Hio=(hio/phy), using eq.6.5
68 printf("\t Correct hio1 to the surface at the OD is
        : %.0f Btu/(hr)*(ft^2)*(F) \n",hio1);
69 Uc=((hio1)*(ho)/(hio1+ho)); // clean overall
        coefficient ,Btu/(hr)*(ft^2)*(F)
70 printf("\t clean overall coefficient is : %.0f Btu/(
        hr)*(ft^2)*(F) \n",Uc);
71 A2=0.1963; // actual surface supplied for each tube,
        ft^2,from table 10
72 L=16;
73 A=(Nt*L*A2); // ft^2
74 printf("\t total surface area is : %.0f ft^2 \n",A);
75 UD=((Q)/((A)*(LMTD)));
76 printf("\t actual design overall coefficient is : %
        .0f Btu/(hr)*(ft^2)*(F) \n",UD);
77 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
78 printf("\t actual Rd is : %.5f (hr)*(ft^2)*(F)/Btu \
        n",Rd);
79 printf("\t pressure drop for inner pipe \n");
80 n=1; // number of passes
81 at1=0.302; // flow area, in^2
82 at=((Nt*at1)/(144*n)); // total area,ft^2,from eq
        .7.48
83 printf("\t flow area is : %.3f ft^2 \n",at);
84 Gt=(30000/(0.372)); // mass velocity ,lb/(hr)*(ft^2)
85 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
        ",Gt);
86 mu2=0.029; // at inlet ,lb/(ft)*(hr)
87 Ret=((D)*(Gt)/mu2); // reynolds number
88 printf("\t reynolds number is : %.2e \n",Ret);
89 row=(76.1/((359)*(636/492)*(14.7/39.7)));
90 printf("\t row is %.3f ld/ft^3 \n",row);
91 s=(row/62.5);
92 printf("\t s is %.4f \n",s);
93 f=0.000138; // friction factor for reynolds number

```

```

143000, using fig.26
94 delPt=((f*(Gt^2)*(16)*(1))/(5.22*(10^10)*(0.0517)*(s
    )))/(2); // using eq.7.45, psi
95 printf("\t delPt is : %.1f psi \n",delPt);
96 printf("\t allowable delPa is negligible psi \n");
97 printf("\t pressure drop for annulus \n");
98 f=0.0017; // friction factor for reynolds number
    31000, using fig.29
99 s=1; // for reynolds number 31000,using fig.6
100 Ds=17.25/12; // ft
101 B=6;
102 N=(12*L/B); // number of crosses ,using eq.7.43
103 printf("\t number of crosses are : %.0f \n",N);
104 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)));
    // using eq.7.44, psi
105 printf("\t delPs is : %.1f psi \n",delPs);
106 printf("\t allowable delPT is 10 psi \n");
107 //end

```

Scilab code Exa 12.7 Calculation of a Surface Condenser

```

1 printf("\t example 12.7 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 V=7.5; // fps
4 W=250000;
5 CCl=0.85;
6 CT=1;
7 CL=1;
8 Ct=263;
9 UD=(CCl*CT*CL*Ct*(V^(1/2)));
10 printf("\t design overall coefficient is : %.0f Btu
    /(hr)*(ft^2)*(F) \n",UD);
11 A=(W/8);
12 printf("\t area is : %.0f ft^2 \n",A);

```

```

13 a1=0.229; // ft^2/ft, table 10
14 at=0.475; // in^2, table 10
15 t1=70;
16 Ts=91.72; //F
17 n=2;
18 L=26;
19 t2=(Ts)-((Ts-t1)/((10)^(0.000279*UD*L*n*a1/(V*at))))
    ;
20 printf("\t t2 is : %.1f F \n",t2); // calculation
    mistake in book
21 Go=(W*950)/((t2-t1)*500);
22 printf("\t circulation rate is : %.0f gpm \n",Go);
23 // end

```

Chapter 13

Condensation of Mixed vapors

Scilab code Exa 13.1 Calculation of the Bubble Point

```
1 printf("\t example 13.1 \n");
2 // at atmospheric pressure ,Pt=760 mm Hg
3 printf("\t approximate values are mentioned in the
   book \n");
4 x(1)=0.077; // mole fraction of C4
5 x(2)=0.613; // mole fraction of C5
6 x(3)=0.310; // mole fraction of C6
7 printf("\t for T 100 F \n");
8 Pp(1)=3170; // vapour pressure of C4, from fig 13.3
9 Pp(2)=790; // vapour pressure of C5, from fig 13.3
10 Pp(3)=250; // vapour pressure of C6, from fig 13.3
11 i=1;
12 while(i<4)
13     p(i)=(Pp(i)*x(i));
14     printf(" \n x(i)      Pp(i)      p(i) \n "+
           string(x(i))+
           " "+string(Pp(i))+
           " "+string(p(i))+
           "\n");
15     i=i+1;
16 end
17 pt=p(1)+p(2)+p(3);
18 printf("\t total pressure is : %.1f mm Hg \n",pt);
```

```

19 printf("\t pressure is too high \n");
20 printf("\t for T 96 F \n");
21 Pp(1)=2990; // vapour pressure of C4, from fig 13.3
22 Pp(2)=725; // vapour pressure of C5,from fig 13.3
23 Pp(3)=229; // vapour pressure of C6,from fig 13.3
24 i=1;
25 while(i<4)
26     p(i)=(Pp(i)*x(i));
27     printf(" \n x(i)          Pp(i)          p(i) \n " +
            string(x(i))+"          "+string(Pp(i))+"          "+
            string(p(i))+" \n");
28 i=i+1;
29 end
30 pt=p(1)+p(2)+p(3);
31 printf("\t total pressure is : %.1f mm Hg \n",pt);
32 printf("\t pressure is too low \n");
33 printf("\t for T 97 F \n");
34 Pp(1)=3040; // vapour pressure of C4, from fig 13.3
35 Pp(2)=740; // vapour pressure of C5,from fig 13.3
36 Pp(3)=234; // vapour pressure of C6,from fig 13.3
37 i=1;
38 while(i<4)
39     p(i)=(Pp(i)*x(i));
40     printf(" \n x(i)          Pp(i)          p(i) \n " +
            string(x(i))+"          "+string(Pp(i))+"          "+
            string(p(i))+" \n");
41 i=i+1;
42 end
43 pt=p(1)+p(2)+p(3);
44 printf("\t total pressure is : %.1f mm Hg \n",pt);
45 i=1;
46 while(i<4)
47     y(i)=(Pp(i)*x(i)/pt);
48     printf("\n x(i)          y(i) \n "+string(x(i))+"
            "+string(y(i))+" \n");
49     i=i+1;
50 end
51 printf("\t solution for b \n");

```

```

52 // Similarly at what temperature will the mixture
    start to boil if the system is under a pressure
    of 35 psia
53 printf("\t for T 150 F \n");
54 Pp(1)=6100; // vapour pressure of C4, from fig 13.3
55 Pp(2)=1880; // vapour pressure of C5,from fig 13.3
56 Pp(3)=680; // vapour pressure of C6,from fig 13.3
57 i=1;
58 while(i<4)
59     p(i)=(Pp(i)*x(i));
60     printf("\n x(i)      Pp(i)      p(i) \n "+
        string(x(i))+
        "+string(Pp(i))+
        string(p(i))+
        "\n");
61 i=i+1;
62 end
63 pt=p(1)+p(2)+p(3);
64 printf("\t total pressure is : %.0f mm Hg \n",pt);
65 printf("\t pressure is too high \n");
66 printf("\t for T 149F \n");
67 Pp(1)=6050; // vapour pressure of C4, from fig 13.3
68 Pp(2)=1850; // vapour pressure of C5,from fig 13.3
69 Pp(3)=670; // vapour pressure of C6,from fig 13.3
70 i=1;
71 while(i<4)
72     p(i)=(Pp(i)*x(i));
73     printf("\n x(i)      Pp(i)      p(i) \n "+
        string(x(i))+
        "+string(Pp(i))+
        string(p(i))+
        "\n");
74 i=i+1;
75 end
76 pt=p(1)+p(2)+p(3);
77 printf("\t total pressure is : %.0f mm Hg \n",pt);
78 i=1;
79 while(i<4)
80     y(i)=(Pp(i)*x(i)/pt);
81     printf("\n x(i)      y(i) \n "+string(x(i))+
        "+string(y(i))+
        "\n");
82     i=i+1;

```

```

83 end
84 printf("\t solution for c \n");
85 printf("\t for T 95F \n");
86 K(1)=3.13; // fig 7
87 K(2)=0.92; // fig 7
88 K(3)=0.30; // fig 7
89 i=1;
90 while(i<4)
91     y(i)=(K(i)*x(i));
92     printf("\n x(i)          K(i)          y(i) \n "+string(
          x(i))+          "+string(K(i))+          "+string(y
          (i))+          "\n");
93     i=i+1;
94 end
95 yt=y(1)+y(2)+y(3);
96 printf("\t yt is : %.3f \n",yt);
97 printf("\t yt is too low \n");
98 printf("\t for T 100F \n");
99 K(1)=3.35; // fig 7
100 K(2)=1; // fig 7
101 K(3)=0.335; // fig 7
102 i=1;
103 while(i<4)
104     y(i)=(K(i)*x(i));
105     printf("\n x(i)          K(i)          y(i) \n "+string(
          x(i))+          "+string(K(i))+          "+string(y
          (i))+          "\n");
106     i=i+1;
107 end
108 yt=y(1)+y(2)+y(3);
109 printf("\t yt is : %.3f \n",yt);
110 printf("\t yt is too low \n");
111 printf("\t for T 102F \n");
112 K(1)=3.45; // fig 7
113 K(2)=1.02; // fig 7
114 K(3)=0.35; // fig 7
115 i=1;
116 while(i<4)

```



```

117     y(i)=(K(i)*x(i));
118     printf("\n x(i)          K(i)          y(i) \n "+string(
        x(i))+          "+string(K(i))+          "+string(y
        (i))+          \n");
119     i=i+1;
120 end
121 yt=y(1)+y(2)+y(3);
122 printf("\t yt is : %.3f \n",yt);
123 printf("\t solution for d \n");
124 // The use of K values gives y, directly and permits
        use of the total mol fraction of yt = 1.00 as
        the criterion for equilibrium
125 printf("\t for T 150F \n");
126 K(1)=2.8; // fig 7
127 K(2)=1.01; // fig 7
128 K(3)=0.4; // fig 7
129 i=1;
130 while(i<4)
131     y(i)=(K(i)*x(i));
132     printf("\n x(i)          K(i)          y(i) \n "+string(
        x(i))+          "+string(K(i))+          "+string(y
        (i))+          \n");
133     i=i+1;
134 end
135 yt=y(1)+y(2)+y(3);
136 printf("\t yt is : %.3f \n",yt);
137 printf("\t yt is too low \n");
138 printf("\t for T 153F \n");
139 K(1)=2.90; // fig 7
140 K(2)=1.06; // fig 7
141 K(3)=0.415; // fig 7
142 i=1;
143 while(i<4)
144     y(i)=(K(i)*x(i));
145     printf("\n x(i)          K(i)          y(i) \n "+string(
        x(i))+          "+string(K(i))+          "+string(y
        (i))+          \n");
146     i=i+1;

```

```

147 end
148 yt=y(1)+y(2)+y(3);
149 printf("\t yt is : %.3f \n",yt);
150 printf("\t solution for e at pt=760mm Hg \n");
151 y(1)=0.077; // mole fraction of C4
152 y(2)=0.613; // mole fraction of C5
153 y(3)=0.310; // mole fraction of C6
154 printf("\t for T 130F \n");
155 K(1)=5; // fig 7
156 K(2)=1.65; // fig 7
157 K(3)=0.62; // fig 7
158 i=1;
159 while(i<4)
160     x(i)=(y(i)/K(i));
161     printf("\n y(i)          K(i)          x(i) \n "+string(
        y(i))+          "+string(K(i))+          "+string(x
        (i))+          \n");
162     i=i+1;
163 end
164 xt=x(1)+x(2)+x(3);
165 printf("\t xt is : %.3f \n",xt);
166 printf("\t xt is too low \n");
167 printf("\t for T 120F \n");
168 K(1)=4.4; // fig 7
169 K(2)=1.4; // fig 7
170 K(3)=0.51; // fig 7
171 i=1;
172 while(i<4)
173     x(i)=(y(i)/K(i));
174     printf("\n y(i)          K(i)          x(i) \n "+string(
        y(i))+          "+string(K(i))+          "+string(x
        (i))+          \n");
175     i=i+1;
176 end
177 xt=x(1)+x(2)+x(3);
178 printf("\t xt is : %.3f \n",xt);
179 printf("\t xt is high \n");
180 printf("\t for T 123F \n");

```

```

181 K(1)=4.6; // fig 7
182 K(2)=1.49; // fig 7
183 K(3)=0.545; // fig 7
184 i=1;
185 while(i<4)
186     x(i)=(y(i)/K(i));
187     printf("\n y(i)          K(i)          x(i) \n "+string(
        y(i))+          "+string(K(i))+          "+string(x
        (i))+          "\n");
188     i=i+1;
189 end
190 xt=x(1)+x(2)+x(3);
191 printf("\t xt is : %.3f \n",xt);
192 printf("\t dew point at 760mm is 123F \n");
193 printf("\t dew point at 35psia \n");
194 printf("\t for T 174F \n");
195 K(1)=3.7; // fig 7
196 K(2)=1.38; // fig 7
197 K(3)=0.58; // fig 7
198 i=1;
199 while(i<4)
200     x(i)=(y(i)/K(i));
201     printf("\n y(i)          K(i)          x(i) \n "+string(
        y(i))+          "+string(K(i))+          "+string(x
        (i))+          "\n");
202     i=i+1;
203 end
204 xt=x(1)+x(2)+x(3);
205 printf("\t xt is : %.3f \n",xt);
206 printf("\t dew point is 174F \n");
207 // end

```

Scilab code Exa 13.2 Calculation of the Bubble Point and Vapor Composition by Relative

```

1 printf("\t example 13.2 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 printf("\t bubble point at 95F and 14.7 psia \n");
4 x(1)=0.077; // mole fraction of C4
5 x(2)=0.613; // mole fraction of C5
6 x(3)=0.310; // mole fraction of C6
7 K(1)=3.13; // fig 7
8 K(2)=0.92; // fig 7
9 K(3)=0.3; // fig 7
10 a(1)=3.4; // a= alpha
11 a(2)=1;
12 a(3)=0.326;
13 i=1;
14 while(i<4)
15     Z(i)=(a(i)*x(i));
16     i=i+1;
17 end
18 Zt=Z(1)+Z(2)+Z(3);
19 printf("\t Zt is : %.3f \n",Zt);
20 i=1;
21 while(i<4)
22     y(i)=(a(i)*x(i)/(Zt));
23     printf(" \n x(i)      K(i)      a(i)      Z(i)
           y(i) \n "+string(x(i))+      "+string(K
           (i))+      "+string(a(i))+      "+string(Z(i)
           ))+"      "+string(y(i))+ \n");
24     i=i+1;
25 end
26 yt=y(1)+y(2)+y(3);
27 printf("\t yt is : %.3f \n",yt);
28 K2=(y(2)/x(2));
29 printf("\t K2 is : %.3f \n",K2);
30 printf("\t bubble point is 102 \n"); // from fig 7 ,
   comparing K2 value
31 printf("\t dew point at 130F and 14.7 psia \n");
32 y(1)=0.077; // mole fraction of C4
33 y(2)=0.613; // mole fraction of C5

```

```

34 y(3)=0.310; // mole fraction of C6
35 K(1)=5; // fig 7
36 K(2)=1.65; // fig 7
37 K(3)=0.62; // fig 7
38 a(1)=3.03; // a= alpha
39 a(2)=1;
40 a(3)=0.376;
41 i=1;
42 while(i<4)
43     Z(i)=(y(i)/a(i));
44     i=i+1;
45 end
46 Zt=Z(1)+Z(2)+Z(3);
47 printf("\t Zt is : %.3f \n",Zt);
48 i=1;
49 while(i<4)
50     x(i)=(Z(i)/Zt);
51     printf("\n y(i)      K(i)      a(i)      Z(i)
           x(i) \n "+string(y(i))+      "+string(K
           (i))+      "+string(a(i))+      "+string(Z(i)
           )+      "+string(x(i))+      \n");
52     i=i+1;
53 end
54 xt=x(1)+x(2)+x(3);
55 printf("\t xt is : %.0f \n",xt);
56 K2=(y(2)/x(2));
57 printf("\t K2 is : %.2f \n",K2);
58 printf("\t dew point is 122F \n"); // from fig 7,
    comparing K2 value
59 // end

```

Scilab code Exa 13.3 Condenser Calculations for a Multicomponent Mixture

```

1 printf("\t example 13.3 \n");

```

```

2  printf("\t approximate values are mentioned in the
   book \n");
3  printf("\t for condensing range \n");
4  V(1)=170.5; // volume of C3,Mol/hr
5  V(2)=284; // volume of C4,Mol/hr
6  V(3)=56.8; // volume of C6,Mol/hr
7  V(4)=341.1; // volume of C7,Mol/hr
8  V(5)=284; // volume of C8,Mol/hr
9  Tw=283; // dew point assumption
10 Tb=120; // bubble point assumption
11 K(1)=13.75 // at 283F
12 K(2)=6.18 // at 283F
13 K(3)=1.60 // at 283F
14 K(4)=0.825 // at 283F
15 K(5)=0.452 // at 283F
16 i=1;
17 while(i<6)
18     Z(i)=(V(i)/K(i));
19     i=i+1;
20 end
21 Vt=V(1)+V(2)+V(3)+V(4)+V(5);
22 Zt=Z(1)+Z(2)+Z(3)+Z(4)+Z(5);
23 L(1)=170.5; // volume of C3,Mol/hr
24 L(2)=284; // volume of C4,Mol/hr
25 L(3)=56.8; // volume of C6,Mol/hr
26 L(4)=341.1; // volume of C7,Mol/hr
27 L(5)=284; // volume of C8,Mol/hr
28 K1(1)=4.1 // at 283F
29 K1(2)=1.39 // at 283F
30 K1(3)=0.17 // at 283F
31 K1(4)=0.06 // at 283F
32 K1(5)=0.023 // at 283F
33 i=1;
34 while(i<6)
35     Z1(i)=(L(i)*K1(i));
36     printf(" \n V(i)      K(i)      Z(i)      L(i)
           Kl(i)      Z1(i) \n "+string(V(i))+
           "+string(K(i))+      "+string(Z(i))+

```

```

        "+string(L(i))+"          "+string(K1(i))+"
        "+string(Z1(i))+" \n");
37     i=i+1;
38 end
39 Lt=L(1)+L(2)+L(3)+L(4)+L(5);
40 Z1t=Z1(1)+Z1(2)+Z1(3)+Z1(4)+Z1(5);
41 printf("\t total volume in vapour phase : %.1f \n",
        Vt);
42 printf("\t total Zt in vapour phase : %.1f \n",Zt);
43 printf("\t total volume in liquid phase : %.1f \n",
        Lt);
44 printf("\t total Z1t in liquid phase : %.1f \n",Z1t)
    ;
45 // Range: 283 to 270 F
46 // Trial: Assume V /L = 4.00.
47 R=4; // R=(V/L), assumption
48 K(1)=12.75 // at 270F
49 K(2)=5.61 // at 270F
50 K(3)=1.40 // at 270F
51 K(4)=0.705 // at 270F
52 K(5)=0.375 // at 270F
53 i=1;
54 Y(i)=V(i);
55 while(i<6)
56     P(i)=(K(i)*R);
57     L1(i)=(V(i)/(1+P(i))); // V(i)=Y(i)
58     printf(" \n Y(i)      K(i)      P(i)      L1(i)
        \n "+string(V(i))+"          "+string(K(i))+"
        "+string(P(i))+"          "+string(L1(i))+"
        \n");
59     i=i+1;
60 end
61 L1t=L1(1)+L1(2)+L1(3)+L1(4)+L1(5);
62 V1t=(Vt-L1t);
63 R1=(V1t/L1t);
64 printf("\t total liquid at 270F : %.0f \n",L1t);
65 printf("\t total vapour at 270F : %.0f \n",V1t);
66 printf("\t R1 is : %.0f \n",R1);

```

```

67 // If the assumed and calculated values of V /L had
    not checked , a new value would have been assumed.
68 printf("\t for condensing curve \n");
69 R270=4; // V/L at 270, from table 13.2
70 R270=1.567; // V/L at 250, from table 13.2
71 R270=0.916; // V/L at 230, from table 13.2
72 R270=0.520; // V/L at 200, from table 13.2
73 R270=0.226; // V/L at 160, from table 13.2
74 H270=30835500; // 4th table in solution ,enthalpies
    calculated from fig 10
75 printf("\t heat load at 270F is : %.0f Btu/hr \n",
    H270);
76 H250=27042400; // 5th table in solution ,enthalpies
    calculated from fig 10
77 printf("\t heat load at 250F is : %.0f Btu/hr \n",
    H250);
78 Q=H270-H250;
79 printf("\t heat load for interval 270-250F : %.0f
    Btu/hr \n",Q);
80 qt=21203000; // 6th table in solution , calculated
    from fig 10
81 printf("\t heat load for entire range is : %.0f Btu/
    hr \n",qt);
82 M=210410; // M=sum(U*A) , 6th table in solution ,
    calculated from fig 10
83 w=(qt/(120-80));
84 printf("\t water flow rate : %.1e lb/hr \n",w);
85 W=95450; // flow rate of feed ,lb/hr
86 delT=(qt/M);
87 printf("\t weighted delT is : %.1f F \n",delT);
88 q1=[0 3.4765 7.2696 10.109 13.468 17.399 21.203];
89 T1=[283 270 250 230 200 160 120];
90 plot2d(q1,T1,style=3,rect=[0,0,25,300]);
91 q2=[0 21.203];
92 T2=[283 120];
93 plot2d(q2,T2,style=5,rect=[0,0,25,300]);
94 xtitle("condensing curve","heat load ,Btu/hr","
    temperature ,F");

```



```

95 legend("green-differential vapour","red-vapour");
96 printf("\t calculation of the exchanger \n");
97 T1=283; // inlet hot fluid ,F
98 T2=120; // outlet hot fluid ,F
99 t1=80; // inlet cold fluid ,F
100 t2=120; // outlet cold fluid ,F
101 L=16;
102 Nt=774;
103 n=4;
104 row=62.5;
105 Qs=21203000; // Btu/hr
106 Qw=(w*1*(120-80));
107 printf("\t heat absorbed by water : %.4e Btu/hr \n",
    Qw);
108 Mavg=84; // This corresponds very closely to hexane
    (mol. Wt. = 86.2) whose properties will be used
    throughout.
109 Qc=W*(0.6/2)*(283-120);
110 printf("\t condensate sensible heat load: %.2e Btu/
    hr \n",Qc);
111 S=(Qc*(100/Qs));
112 printf("\t submergence : %.0f \n",S);
113 Tc=((T1+T2)/2); // caloric temperature of hot fluid ,
    F
114 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
115 tc=((t1+t2)/2); // caloric temperature of cold fluid
    ,F
116 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
117 printf("\t hot fluid:shellside ,vapour \n");
118 Nts=(774*(1-.22)); // as submergence is 22%
119 printf("\t unmerged tubes : %.0f \n",Nts);
120 Gs=(W/(L*(Nts^(2/3)))); // eq 12.43
121 printf("\t Gs is : %.1f \n",Gs);
122 Ho=200; // assumption
123 printf("\t cold fluid:inner tube side ,water \n");
124 at1=0.302; // flow area , in^2

```

```

125 at=((Nt*at1)/(144*n)); // total area ,ft ^2,from eq
    .7.48
126 printf("\t flow area is : %.3f ft^2 \n",at);
127 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
128 printf("\t mass velocity is : %.1e lb/(hr)*(ft^2) \n
    ",Gt);
129 V=(Gt/(3600*row));
130 printf("\t V is : %.2f fps \n",V);
131 hi=1355; // fig 25
132 ID=0.62;
133 OD=0.75;
134 hio=((hi)*(ID/OD)); //Hio=(hio/phyp), using eq.6.5
135 printf("\t Correct hio to the surface at the OD is :
    %.2e Btu/(hr)*(ft^2)*(F) \n",hio);
136 tw=(tc)+(((Ho)/(hio+Ho))*(Tc-tc)); // from eq.5.31
137 printf("\t tw is : %.0f F \n",tw);
138 tf=(Tc+tw)/(2); // from eq 12.19
139 printf("\t tf is : %.0f F \n",tf);
140 kf=0.077; //table 4, Btu/(hr)*(ft^2)*(F/ft)
141 sf=0.60; // from table 6
142 muf=0.21; // cp, from fig 14
143 ho=206; // Btu/(hr)*(ft^2)*(F), from fig 12.9
144 printf("\t Correct ho to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
145 Uc=((hio)*(ho)/(hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
146 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
147 Ac=(Qw/(174*delt));
148 printf("\t clean surface required for condensation :
    %.2e ft^2 \n",Ac);
149 As=1210*0.22;
150 printf("\t clean surface required for subcooling : %
    .0f ft^2 \n",As);
151 AG=As+Ac;
152 printf("\t total clean surface : %.0f ft^2 \n",AG);
153 UC=(Qw/(AG*delt));
154 printf("\t weighted clean overall coefficient : %.0f

```

```

        Btu/(hr)*(ft^2)*(F) \n",UC);
155 A2=0.1963; // actual surface supplied for each tube,
        ft^2,from table 10
156 A=(Nt*L*A2); // ft^2
157 printf("\t total surface area is : %.2e ft^2 \n",A);
158 UD=((Qw)/((A)*(delt)));
159 printf("\t actual design overall coefficient is : %
        .1f Btu/(hr)*(ft^2)*(F) \n",UD);
160 Rd=((UC-UD)/((UD)*(UC))); // (hr)*(ft^2)*(F)/Btu
161 printf("\t actual Rd is : %.5f (hr)*(ft^2)*(F)/Btu \
        n",Rd);
162 printf("\t pressure drop for annulus \n");
163 B=30;
164 as=33*0.25*(30/144)*1; // eq 7.1
165 printf("\t as is : %.2f ft^2 \n",as);
166 Gs=(W/as);
167 printf("\t Gs is : %.2e lb/(hr)*(ft^2) \n",Gs); //
        eq 7.2
168 mu1=0.0218; // at 283F
169 De=0.0608; // ft, from fig 15
170 Res=(De*Gs)/(mu1);
171 printf("\t reynolds number is : %.2e \n",Res);
172 f=0.00125; // fig 29
173 N=(12*L/B); // eq 7.43
174 printf("\t number crosses : %.0f \n",N);
175 row1=0.527; //lb/ft^3
176 s=0.00844;
177 Ds=2.75; // ft
178 delPs=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)
        *(1)))/(2); // using eq 12.47,psi
179 printf("\t delPs is : %.1f psi \n",delPs);
180 printf("\t pressure drop for inner pipe \n");
181 mu2=1.74; // fig 14
182 D=0.0517; // ft
183 s=1;
184 Ret=(D*Gt/mu2);
185 printf("\t reynolds number : %.2e \n",Ret);
186 f=0.00019; // ft^2/in^2

```

```

187 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(1)*(1))
    ); // using eq.7.45, psi
188 printf("\t delPt is : %.1f psi \n",delPt);
189 X1=0.23; // X1=((V^2)/(2*g)), using fig.27
190 delPr=((4*n*X1)/(s)); // using eq.7.46, psi
191 printf("\t delPr is : %.1f psi \n",delPr);
192 delPT=delPt+delPr; // using eq.7.47, psi
193 printf("\t delPT is : %.1f psi \n",delPT);
194 printf("\t allowable delPa is 10 psi \n");
195 // end

```

Scilab code Exa 13.4 Vapor

```

1 printf("\t example 13.4 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 vA=2*3.7+(7.4); // for steam
4 vB=14.8+(2*7.4); // for CO2
5 MA=18;
6 MB=44;
7 T=403; // K
8 Pt=3.04; // atm
9 kd=(0.0166)*(((403^(3/2))/(3.04*(14.8^(1/3)
    +29.6^(1/3))^2))*((1/18)+(1/44))^(1/2)); // eq
    13.31
10 printf("\t diffusivity is : %.2f ft^2/hr \n",kd);
11 // end

```

Scilab code Exa 13.5 Calculation of a Steam Carbon Dioxide Condenser

```

1 printf("\t example 13.5 \n");
2 // for a Basis of one Hour

```

```

3 printf("\t approximate values are mentioned in the
   book \n");
4 c(1)=1544; // Flow rate of CO2, Lb/hr
5 h(1)=4500; // Flow rate of H2O, Lb/hr
6
7 c(2)=35; //Flow rate of CO2, Mol/hr
8 h(2)=250; //Flow rate of H2O, Mol/hr
9
10 t(1)=c(1)+h(1); //Total flow rate , Lb/hr
11 t(2)=c(2)+h(2); //Total flow rate , Mol/hr
12
13 Pt = (30+14.7)/(14.7); //Total Pressure in atm
14 printf("\t Pt is %.2f\n",Pt);
15 Pw = ( h(2)/t(2) )*Pt; //Partial pressure of Water
   in atm
16
17 printf("\t Partial Pressure of Water: %.2f atm \n",
   Pw);
18
19 Tw = 267; // from table 7 at 2.68atm
20 Mm = (t(1)/t(2));
21
22 printf("\t mean molecular weight : %.1f \n",Mm);
23 // weighted temperature difference
24 // overall balance
25 //for Inlet
26 Pv=2.68; // water vapour pressure , atm
27 Pg=Pt-Pv; // Inert pressure
28 //for Exit
29 Pw1 = 0.1152 // Partial pressure of water at 120 F
30 Pv1 = 0.115; // Water vapor pressure
31 Pg1 = 2.935; // Inert pressure
32
33 w1 = 250; //Pound mols steam inlet
34 w2 = c(2)*(Pv1/Pg1);
35 printf("\tPound mols steam exit: %.2f\n",w2);
36 w3 = w1 - w2;
37 printf("\tPound mols steam condensed: %.2f\n",w3);

```

```

38 //Assume points at 267, 262, 255,225,150,120 deg F
39 //For the interval from 267 to 262 F
40
41 Pv2 = 2.49; // From table 7 at 262 F
42 Pg2 = Pt - Pv2; //Inert pressure
43 printf("\tPg is %.2 f",Pg2);
44
45 w4 = c(2) * (Pv2/Pg2); //Mol steam remaining
46 w5 = h(2) - w4; //Mol steam condensed
47
48 printf("\tMol steam remaining:%.0 f\n",w4);
49 printf("\tMol steam condensed:%.0 f\n",w5);
50
51 h1 = (w5*18*937.3) + (0.46*(267-262) * w5 * 18); //
    Heat of condensation
52 h2 = (w4 * 18 * 0.46*(267-262)); //Heat from
    uncondensed steam
53 h3 = c(1)*0.22*5.0; //Heat from noncondensable
54
55 printf("\tHeat of condensation:%.2 e\n",h1);
56 printf("\tHeat from uncondensed steam:%.2 e\n",h2);
57 printf("\tHeat from noncondensable:%.1 e\n",h3);
58
59 ht = h1+h2+h3;//Total heat
60 printf("\tTotal heat:%.0 f\n",ht);
61
62 //Similarly calculating the Heat balance for other
    intervals
63 printf("\tInterval ,F\tTotal Heat\n\t267-262\t1
    ,598,000\n\t262-255\t1,104,000\n\t255-225\t1
    ,172,000\n\t225-150\t751,000\n\t150-120\t177,000\
    n\tTotal\t4,802,000\n");
64
65 w=4802000/(115-80); //Total water
66 printf("\tTotal water: %.2 e\n",w);
67 //Water coefficient
68 Nt = 246;
69 at1 = 0.302;

```

```

70 n = 4;
71
72 at = Nt * (at1/(144*n)); // From eq 7.48
73 printf("\tat is %.3f ft^2\n",at);
74 Gt = w/at;
75 printf("\tGt is %.2e lb/(hr)(ft^2)\n",Gt);
76 ro = 62.5;
77 V = Gt/(3600*ro);
78 printf("\tV is %.2f fps\n",V);
79 hi = 1120; // From fig. 25
80 ID = 0.62;
81 OD = 0.75;
82 hi0= hi *(ID/OD); //From eq 6.5
83 printf("\thi0 is %.0f\n",hi0);
84 //Mean properties at 267 F
85 c = ((c(1)*0.22)+(h(1)*0.46))/t(1); // Calculation
      mistake in Book
86 printf("\tMean c:%.3f Btu/(lb)(F)\n",c);
87
88 k = ((c(1)*0.0128)+(h(1)*0.015))/t(1); //
      Calculation mistake in Book
89 printf("\tMean k:%.4f Btu/(hr)(ft^2)(F/ft)\n",k);
90
91 mu = (((c(1)*0.019)+(h(1)*0.0136))/t(1))* 2.42; //
      Calculation mistake in Book
92 printf("\tMean mu:%.4f lb/(hr)(ft)\n",mu);
93
94 ID1 = 21.25;
95 C = 0.25;
96 B = 12;
97 PT = 1.0;
98
99 as = ID1 * C * (B/(144*PT)); //From eq 7.1
100 printf("\tas is %.3f ft^2\n",as);
101 Gs = t(1)/as //From eq 7.2
102 printf("\tGs is %.3e lb/(hr)(ft^2)\n",Gs);
103 Ds = 0.0792; // From Fig 28
104 Res = Ds * (Gs/0.0363); // From eq 7.3

```

```

105 printf("\tRes is %.2e\n",Res);
106 jH = 102; // From Fig 28
107 x = ((c*mu)/k)^(1/3);
108 printf("\t(c.mu/k)^1/3 is %.0f\n",x);
109 h0 = jH * 0.0146 * (x/Ds); //From eq 6.15b
110 printf("\th0 is %.0f\n",h0);
111 y = 0.62 // y = (mu/ro * kd)^(2/3)
112 z = 1.01; // z = ((c*mu)/k)^(2/3)
113
114 K = (h0*z)/(0.407*Mm*y); //KG = K/p0f
115 printf("\tK is %.2f\n",K);
116 //at point 1
117 Tg = 244; // F
118 tW = 115;
119 deltt=(Tg-tW);
120 printf("\tdelt is %.0f F \n",delt);

```

Scilab code Exa 13.6 Calculation of a Condenser

```

1 printf("\t example 13.6a \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3
4 ds=[0 10 20 30 40 50 60 70 80 90 100];
5 tmp=[90 145 180 208 234 260 286 312 338 367 400];
6 clf();
7 subplot(3,2,1);
8 plot2d(ds,tmp,style=2,rect=[0,80,100,400]);
9 xtitle("Plot of ASTM curve",boxed=1);
10 xlabel("Per cent distilled off");
11 ylabel("Temperature F");
12
13 //From the plotted ASTM curve and reference line
14 s = (312-145)/60; // (70% - 10%)/60%
15 printf("\tSlope of ASTm = %.2f F \n",s);

```



```

16 ap = (180+260+338)/3; // (20% +50% +80%)/3
17 printf("\tAverage 50prcnt point = %.1f  F\n",ap);
18
19 fc = 38; // F , from Fig.13.8
20 printf("\t50prcnt point ASTM = 50prcnt point flash
    curve = %.0f  F\n",fc);
21 fc1 = ap - fc; // F , fixing first point on EFC
22 printf("\t50prcnt on EFC = %.0f  F\n",fc1);
23
24 s1 = 1.65; // ( F /%) from fig 13.10, upper curve
25 ten = 221 - 40*s1; //
26 printf("\t10prcnt on EFC = 50prcnt - 40prcnt = %.0f
    F\n",ten);
27 sty = 221 + 20*s1; //
28 printf("\t70prcnt on EFC = 50prcnt + 20prcnt %.0f
    F\n",sty);
29
30 //Draw this line as a reference through the 50%
    point. Calculate the flash curve for different
    percentages off
31
32 //0% off
33 printf("\n\t0 prcnt off:\n");
34 dela = 90 - 117; // Step (8)
35 printf("\t\tDelT ASTM = %.0f  F\n",dela);
36 delE = dela * 0.50; // Step (9)
37 printf("\t\tDelT EFC = %.1f  F\n",delE);
38 FE = 139 - delE; // Step (10)
39 printf("\t\t F EFC = %.1f\n",FE);
40 //end
41 ov=13300; //lb/hr
42 ng=90; //lb/hr
43 mng=50; // mol. wt
44 st=370; //lb/hr
45 avG=50; // F API
46 //For 80%
47 ouc=ov*0.80; //lb/hr
48 printf("\toil uncondensed = %.0f lb/hr\n",ouc);

```



```

        t\t\tpsia\t\toil , psia\t\tNC gas , psia\n");
109 mo(1)=107.5;
110 mo(2)=94.3;
111 mo(3)=77.7;
112 mo(4)=57.4;
113 mo(5)=31.8;
114 mo(6)=17.1;
115 mo(7)=8.9;
116 i=1;
117 while(i<8)
118     mt(i)=mo(i)+mg1+ms1;
119     ppo(i)=(mo(i)/mt(i))*tp;
120     ppg(i)=(mg1/mt(i))*tp;
121     i=i+1;
122 end
123 printf("\t
n");
124 printf("\t100\t13330\t\t300\t\t317\t\t124\t\t%.1f\t\t1.8\t\t20.6\t\t%.1f\t\t19.7\t\t%.1f\t\t\t%.3f\t\t\t305\n",mo(1),mt(1),ppo(1),ppg(1));
125 printf("\t80\t10664\t\t269\t\t286\t\t113\t\t%.1f\t\t1.8\t\t20.6\t\t%.1f\t\t19.7\t\t%.1f\t\t\t%.3f\t\t\t277\n",mo(2),mt(2),ppo(2),ppg(2));
126 printf("\t60\t7998\t\t239\t\t256\t\t103\t\t%.1f\t\t1.8\t\t20.6\t\t%.1f\t\t19.7\t\t%.1f\t\t\t%.3f\t\t\t240\n",mo(3),mt(3),ppo(3),ppg(3));
127 printf("\t40\t5332\t\t207\t\t224\t\t93\t\t%.1f\t\t1.8\t\t20.6\t\t%.1f\t\t19.7\t\t%.1f\t\t\t%.3f\t\t\t205\n",mo(4),mt(4),ppo(4),ppg(4));
128 printf("\t20\t2666\t\t178\t\t195\t\t84\t\t%.1f\t\t1.8\t\t20.6\t\t%.1f\t\t19.7\t\t%.1f\t\t\t%.3f\t\t\t163\n",mo(5),mt(5),ppo(5),ppg(5));
129 printf("\t10\t1333\t\t155\t\t172\t\t78\t\t%.1f\t\t1.8\t\t20.6\t\t%.1f\t\t19.7\t\t%.1f\t\t\t%.3f\t\t\t127\n",mo(6),mt(6),ppo(6),ppg(6));
130 printf("\t5\t667\t\t141\t\t158\t\t75\t\t%.1f\t\t1.8\t\t20.6\t\t%.1f\t\t19.7\t\t%.1f\t\t\t%.3f\t\t\t95

```

```

        \n" , mo (7) , mt (7) , ppo (7) , ppg (7) ) ;
131
132 //Trail 1:
133 m=78; //50   API mol. wt. for condesables 1333
134 vap=(ov*0.10)/78; //Mol/hr
135 printf("\n\t\t\t\tMol/hr\n\tOil vapor\t\t%.1f\n\tNC
        gas\t\t\t%.1f\n\tSteam\t\t\t\tX\n\tTotal\t\t\t\t18.9+
        X\n" , vap , mg1) ;
136 vap1=vap+mg1; //Mol/hr
137 psteam=5.09; //psia , For 163 F
138 x1=(psteam*vap1)/(tp-psteam); //mols steam
139 printf("\tX = %.2f mols steam\n" , x1) ;
140 tv=vap1+x1;
141 printf("\n\t\t\t\tMol/hr\tmf\tmf*pt = p-partial\n" ) ;
142 mf1=vap/(tv) ;
143 ppar1=mf1*tp;
144 printf("\tOil vapor\t%.1f\t%.3f\t%.2f\n" , vap , mf1 ,
        ppar1) ;
145 mf2=mg1/tv;
146 ppar2=mf2*tp;
147 printf("\tNC gas\t\t%.1f\t%.3f\t%.2f\n" , mg1 , mf2 ,
        ppar2) ;
148 mf3=x1/tv;
149 ppar3=mf3*tp;
150 printf("\tSteam\t\t\t%.2f\t%.3f\t%.2f\n" , x1 , mf3 , ppar3)
        ;
151 tot1=vap+mg1+x1;
152 tot2=mf1+mf2+mf3;
153 tot3=ppar1+ppar2+ppar3;
154 printf("\tTotal\t\t\t%.2f\t%.3f\t%.2f\n" , tot1 , tot2 ,
        tot3) ;
155 //Error was found. So trail 2 is done in a similar
        way
156 printf("\n\tSimilarly , \n\tT , F \tOil cond , prent\
        tOil cond , lb\tSteam cond , lb\n" ) ;
157 printf("\t173\t74\t\t9863\t\t0\n\t163\t85\t\t11350\t
        \t204\n\t127\t97.5\t\t13000\t\t357\n\t95\t100\t\t
        t13330\t\t370\n" ) ;

```



```

184 hld(3)=1.2; //million Btu
185 hld(4)=1.75; //million Btu
186 hld(5)=2.3; //million Btu
187 hld(6)=2.73; //million Btu
188 hld(7)=3.3; //million Btu
189 hld(8)=3.66; //million Btu
190 subplot(2,2,3);
191 plot2d(hld,ttp,style=6,rect=[0,60,3.8,320]);
192 xtitle("Condensation of mixed hydrocarbons with gas
        and steam",boxed=1);
193 xlabel("Heat load, million Btu");
194 ylabel("Temperature F");
195 //summary
196 dp=3042800; //Btu/hr
197 ttt=3638400; //Btu/hr
198 i2s=thh-dp; //Btu/hr
199 printf("\tInlet to steam dew point = %.4eBtu/hr\n",
        i2s);
200 so=dp-1735900; //Btu/hr
201 printf("\tSteam dew point to outlet = %.4e Btu/hr\n"
        ,so);
202 totl=i2s+so; //Btu/hr
203 printf("\tTotal\t\t\t= %.4e Btu/hr\n",totl);
204 twa=ttt/(120-85);
205 printf("\tTotal water = %.2e lb/hr\n",twa);
206 wt=85+((1306900/ttt)*35); // F
207 printf("\tWater temperature at dew point of steam =
        %.0 f F\n",wt);
208 //Weighted true temperature difference, delT:
209 //Inlet to dew point of steam:
210 delq=2331500;
211 delT1=122.2;
212 UA1=delq/delT1;
213 printf("\tUA = %.0 f\n",UA1);
214 printf("\n\tDew point of steam to outlet\n");
215 printf("\tq\t delq\t Tc\t tw\t delTav\t (delq/delTav) =
        UA\n");
216 printf("\t

```

```

    n");
217 q(1)=2331500;
218 q(2)=2500000;
219 q(3)=2750000;
220 q(4)=3000000;
221 q(5)=3250000;
222 q(6)=3500000;
223 q(7)=3638000;
224 i=1;
225 while(i<7)
226     dq(i)=q(i+1)-q(i);
227     i=i+1;
228 end
229 dpt(1)=173;
230 dpt(2)=169;
231 dpt(3)=161;
232 dpt(4)=149;
233 dpt(5)=134;
234 dpt(6)=112;
235 dpt(7)=95;
236 dtw(1)=97.5;
237 dtw(2)=96;
238 dtw(3)=93;
239 dtw(4)=91;
240 dtw(5)=89;
241 dtw(6)=86;
242 dtw(7)=85;
243 i=1;
244 tua=0;
245 while(i<7)
246     dpdelt(i)=((dpt(i+1)-dtw(i+1))+(dpt(i)-dtw(i)))
                /2;
247     UA(i)=dq(i)/dpdelt(i);
248     tua=tua+UA(i);
249     i=i+1;
250 end
251 printf("\t2331500\t.....\t173\t173\t97.5\n");

```



```

252 i=1;
253 while(i<7)
254     printf("\t"+string(q(i+1))+"\t"+string(dq(i))+"\t"
            +string(dpt(i+1))+"\t"+string(dtw(i+1))+"\t"
            +string(dpdelat(i))+"\t%.0f\n",UA(i)); //from
            Fig. 13.16
255 i=i+1;
256 end
257
258 printf("\t\t\t\t\t\t\t%.0f\tUA = sigma{delq/delt}\n",
        tua);
259 wdt=1306900/tua; // F
260 printf("\tWeighted delat = %.1f F\n",wdt);
261 owdt=ttt/(tua+UA1); // F
262 printf("\tOverall weighted temperature difference =
        %.1f F\n",owdt);
263 printf("\tThe uncorrected LMID is 60.1 F\n");
264 //end
265
266
267 printf("\t example 13.6b \n");
268 printf("\t approximate values are mentioned in the
        book \n");
269
270 // EXCHANGER
271 //Shell side
272 Id = 27; // inches
273 Bs = 16; // inches
274 Ps = 1; // passes
275
276 //Tube side
277 N = 286; // number
278 l = 12; // inches
279 Od = 1; // inch
280 BWG = 14; // BWG
281 Ptc = 1.25; //inches
282 Ps1 = 8; // passes
283

```

```

284 //Clesan surface requirements
285
286 //Head load inlet to dew point of steam
287 st = 2331500; // Btu/hr
288 delT = 122.2 // F
289 hio = 700; // Btu/((hr)(ft^2)( F )) for water
290
291 //From table 13.4 at inlet
292 NC = 1.8; //NC gas , mol/hr
293 sm = 20.6; // steam , mol/hr
294 tt = NC + sm; // mol/hr
295 printf("\tNC gas + steam is %.1f mol/hr\n",tt);
296 pN = tt/129.9; // mol/hr
297 printf("\tpercentage NC gas is %.4f\n",pN);
298
299 //From Fig 13.17
300 hn = 205; //Btu/((hr)(ft^2)( F ))
301 //At dew point of steam
302 No=40.75; // Mol/hr
303 t1 = tt + No; // Mol/hr , total
304 pN1 = tt/t1; // Mol/hr , %NC
305 printf("\tpercentage NC is %.3f\n",pN1);
306
307 //From fig 13.7
308 hn1 = 140; //Btu/((hr)(ft^2)( F ))
309 lm = 136.5; //Btu/((hr)(ft^2)( F ))
310 delT = 122.2; // F
311 Ac1 = st/(lm * delT); // ft^2
312 printf("\tAc1 = Q/(U * delT) is %.1f ft^2\n",Ac1);
313
314 //At dew point of steam to outlet
315 sm1 = 20.64; // Mol/hr , Steam
316 t2 = NC + sm1; // total , Mol/hr
317 printf("\tNC gas + steam is %.1f mol/hr\n",t2);
318 pN1 = NC/t2; // % NC gas
319 printf("\tpercentage NC gas is %.3f \n",pN1);
320
321 Uc = 212; // From Fig 13.17, weighted for oil and

```

```

        steam
322
323 //At outlet , steam = negligible
324
325 Uc = 15; //From Fig 13.17
326
327 //Log mean overall coefficient
328 lm = 74.5; // Btu/((hr)(ft^2)( F )) , From Fig 13.17
329 delT = 44.8; // F
330 Ac2 = 1306900/(lm * delT);
331 printf("\tAc2 is %.0f ft^2\n",Ac2);
332
333 h1 = 770000; // Btu/hr
334 printf("\tHeat of Liquid(50 API) is %.1ef\n",h1);
335 wr = (h1/3638400)*35; // F
336 printf("\tWater rise = %.1f F\n",wr);
337
338 LMTD = 66.3; // F
339 U1=50 //for free convection
340 As = h1/(U1*LMTD); // ft^2
341 printf("\tAs = %.1f ft^2\n",As);
342 Ac = Ac1 + Ac2 + As; //ft^2
343 printf("\tTotal clean surface %.0f ft^2\n",Ac);
344
345 Uc = 3638400/(Ac * 75.5); // Btu/((hr)(ft^2)( F ))
346 printf("\tClean overall coefficient Uc = %.1f Btu/((
        hr)(ft^2)( F ))\n",Uc);
347
348 x = 0.2618; // ft , from table 10
349 A = N * l * x; //ft^2
350 Ud = 3638400/(A * 75.5);
351 printf("\tDesign coefficient Ud is %.1f\n",Ud);
352 Rd =(Uc - Ud)/(Uc * Ud); // ((hr)(ft^2)( F ))/Btu
353 printf("\tDirt factor Rd is %.4f ((hr)(ft^2)( F ))/
        Btu\n",Rd);
354
355 yo = (As/Ac)*A; // ft^2
356 printf("\tSubmerge = %.0f ft^2 of surface\n",yo);

```

357 //end

Chapter 14

Evopartion

Scilab code Exa 14.1 Calculation of Evaporator Surface

```
1 printf("\t example 14.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3
4 t1 = 300; // F
5 t2 = 226; // F
6 bs = 700; // Btu/((hr)(ft ^2)( F ))
7 //Heat Balance
8 Qv = 10000 * 961; // Btu/hr
9 printf("\tQevap is %.2e Btu/hr\n",Qv);
10 Q3 = 10550 * 910; //Btu/hr
11 printf("\tQ300 F is %.2e Btu/hr\n",Q3);
12
13 delT = t1-t2; // F
14 printf("\tTemperature head = %.0f F \n",delT);
15 Ud = bs * 0.865;
16 printf("\tOverall coefficient %.0f\n",Ud);
17 A = Qv/(Ud * delT); //ft^2
18 printf("\tSurface required is %.0f ft ^2\n",A); //
   Wrong calculation in book
19 //end
```

Scilab code Exa 14.2 Calculation of a Triple effect Forward feed Evaporator

```
1 printf("\t example 14.2 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3
4 wf = 50000; // lb/hr
5 sf = wf * 0.10; // lb/hr
6 tp = sf/0.50; // lb/hr
7 printf("\tTotal product is %.0f lb/hr\n",tp);
8 te = wf - tp;
9 printf("\tTotal evaporation is %.0f lb/hr\n",te);
10 cf = 1.0;
11 tF = 100; // F
12 T1 = 244; // F
13 T2 = 125; // F
14 U1=600; // Btu/((hr)*(ft ^2)*( F ))
15 U2=250; // Btu/((hr)*(ft ^2)*( F ))
16 U3=125; // Btu/((hr)*(ft ^2)*( F ))
17
18 T = T1-T2;
19 printf("\tTotal temperature difference is delT%.0f
   F \n",T);
20 df = (26.70- 1.95)/3; // psi/effect
21 printf("\tAverage pressure difference is delP%.2f
   psi/effect \n",df);
22
23 printf("\n\t\t\t\t\tPressure , psia\t\t delP , psi \t
   Steam or vapor , F \t lambda, Btu/lb\n\tSteam
   chest , 1st effect \t 26.70 \t\t\t ... \t\t Ts =
   244 \t\t ls = 949 \n\tSteam chest , 2nd effect \t
   18.45 \t\t\t 8.25 \t\t t1 = 224 \t\t l1 = 961 \n\t
   tSteam chest , 3rd effect \t 10.20(20.7 in. Hg) \t
```

```

    8.25 \t\t t2 = 194 \t\t l1 = 981 \n\tVapor to
condenser \t\t 1.95(26 in. Hg) \t 8.25 \t\t t2 =
125 \t\t l1 = 1022 \n");
24
25 printf("\t949*Ws + 50000*(100-224) = 961*w1\n\t961*
    w1 + (50000 - w1)*(224-194) = 981 * w2\n\t981*w2
    + (50000-w1-w2)(194-125) = 1022 * w2\n\tw1+w2+w3
    = 40000\n");
26 printf("\tSolving simultaneously\n");
27 w1=12400;
28 printf("\tw1 = %.2e \n",w1);
29 w2=13300;
30 printf("\tw2 = %.2e \n",w2);
31 w3=14300;
32 printf("\tw3 = %.2e \n",w3);
33
34 Wt = w1+w2+w3;
35 printf("\tW1-3 is %.0e \n",Wt);
36 Ws = 19100;
37 lms = 949;
38 lm1 = 961;
39 lm2 = 981;
40 lm3 = 1022;
41 Ts = 244;
42 t1 = 224;
43 t2 = 194;
44 t3 = 125;
45
46 A1 = (Ws * lms)/(U1*(Ts-t1)); //ft^2
47 printf("\tA1 is %.0f ft^2 \n",A1);
48 A2 = (w1*lm1)/(U2*(t1-t2)); //ft^2
49 printf("\tA2 is %.0f ft^2 \n",A2);
50 A3 = (w2 * lm2)/(U3*(t2-t3)); //ft^2
51 printf("\tA3 is %.0f ft^2 \n",A3);
52
53 hc = w3 * lm3; // Btu/hr , WRONG CALCULATION IN TEXT
    BOOK
54 printf("\tHeat to condenser is %.3e Btu/hr\n",hc);

```

```

55 wr = hc/(120-85); //lb/hr
56 printf("\tWater requirement is %.1e lb/hr\n",wr);
57 wr1 = wr/500;
58 printf("\t= %.0f gpm \n",wr1);
59 //end

```

Scilab code Exa 14.3 Backward feed Multiple effect Evaporator

```

1  printf("\t example 14.3 \n");
2  printf("\t approximate values are mentioned in the
   book \n");
3  //Same conditions as example 14.2
4  U1 = 400; //Btu/((hr)*(ft ^2)*( F ))
5  U2 = 250; //Btu/((hr)*(ft ^2)*( F ))
6  U3 = 175; //Btu/((hr)*(ft ^2)*( F ))
7
8  w1 = 50000; // lb/hr      From example 14.2
9  wt = 40000; // lb/hr      From example 14.2
10 cf = 1; // From example 14.2
11
12 printf("\t981*w2 + 50000*(100-125) = 1022*w3\n\t961*
   w1 + (50000 - w3)*(125-194) = 981 * w2\n\t949*Ws
   + (50000-w3-w2)(194-224) = 961 * w1\n\tw1+w2+w3 =
   40000\n");
13 printf("\tSolving simultaneously\n");
14 w1 = 15950;
15 w2 = 12900;
16 w3 = 11150;
17 lms = 949;
18 lm1 = 961;
19 lm2 = 981;
20 lm3 = 1022;
21
22 wt = w1+w2+w3;
23 printf("\tw1-3 = %.0f \n",wt);

```



```

54 printf("\tw2 is %.0f\n",w2);
55 w3 = 11350;
56 printf("\tw3 is %.0f\n",w3);
57 Ws = 16850;
58 printf("\tWs is %.0f\n",Ws);
59 Hc = w3 * 1022;
60 printf("\tHeat to condenser is %.2e Btu/hr\n",Hc);
61 wr = Hc/(120-85); //lb/hr
62 printf("\tWater requirement %.2e lb/hr\n",wr);
63 wr1 = wr/500;
64 printf("\t\t\t= %.0fgpm\n",wr1);
65 ec = wt/Ws;
66 printf("\tEconomy, lb evaporation/lb steam %.2f\n",
    ec);
67
68 //comparision of forward and backward feed
69 printf("\t\t\t\t\tForward\t\t\tBackward\n\tTotal steam,
    lb/hr\t19100\t\t16850\n\tCooling water, gpm\t840\t
    \t\t664\n\tTotal surface, ft^2\t4800\t\t4500");

```

Scilab code Exa 14.4 evoparator installer

```

1 printf("\texample 14.4 \n");
2 printf("\tapproximate values are mentioned in the
    book \n");
3 //Assumed that 37500 lb/hr of 15 psig vapor is bled
    from the first effect for use in thevacuum pans
4 printf("\n\tAVERAGE EVAPORATION PER SQUARE FOOT
    HEATING SURFACE FOR SUGAR EVAPORATORS\n");
5 printf("\tEffects\t\tWater evaporated(lb/(hr)*(ft^2)
    )\n");
6 printf("\t1\t\t14-16\n\t2\t\t6-8\n\t3\t\t5-6\n\t4\t\t
    4-5\n\t5\t\t3-4\n");
7 printf("\n\tEVAPORATOR SUMMARY\n");
8 printf("\t

```



```

18 t2 = 147; // F
19 t = t1-t2; // F
20 printf("\tTotal temperature difference in the
    evaporator system = %.0f F\n",t);
21 bpr1 = 1; // F
22 bpr2 = 2; // F
23 bpr3 = 2; // F
24 bpr4 = 4; // F
25 bpr5 = 7; // F
26 bpr = bpr1 + bpr2 + bpr3 + bpr4 + bpr5; // F
27 printf("\tThe sum of all the BPR(from effect 1B to
    the fifth effect inclusive) = %.0f F\n",bpr);
28 tf = t-bpr; // F
29 printf("\tTotal EFFECTIVE temperature difference =
    %.0f F\n",tf);
30 lbh = 229000; //lb/hr
31 tp1=212; // F
32 tp2=184; // F
33 tp3=144; // F
34 tp4=82; // F
35 tj1=243; // F
36 tj2=220; // F
37 tj3=200; // F
38 Ud1=231;
39 Ud2=243;
40 Ud3=230;
41 Ud4=214;
42 Ud5=217;
43 printf("\n\t\t\t\tSUGAR-JUICE HEATERS\n");
44 printf("\tRaw-juice heaters\t\t\t\tClear=juice
    heaters\n\t
    n");
45 rj1=lbh*(tp1-tp2)*(0.91); //Btu/hr
46 printf("\t1.%.0f(%.0f-%.0f)(0.91) = %.2e Btu/hr",lbh
    ,tp1,tp2,rj1);
47 rj2=lbh*(tj1-tj2)*(0.91); //Btu/hr
48 printf("\t1.%.0f(%.0f-%.0f)(0.91) = %.1e Btu/hr\n",

```

```

    lbh , tj1 , tj2 , rj2 );
49 printf("\tVapor temp. = 227 F \tdelT=26.6 F \t\
    tVapor temp. = 250 F \tdelT=15.8 F \n");
50 printf("\tUD=%0.0 f\t\t\t\t\t\t\tUD=%0.0 f\n", Ud1 , Ud2 );
51 A1=rj1/(26.6*Ud1); // ft ^2
52 A2=rj2/(15.8*Ud2); // ft ^2
53 printf("\tSurface ,A=%0.0 f ft ^2\t\t\t\t\tSurface ,A=%0.0 f
    ft ^2\n\n", A1 , A2);
54
55 rj3=lbh*(tp2-tp3)*(0.90); //Btu/hr
56 printf("\t2.%.0 f(%.0 f-%.0 f) (0.91) = %.2 e Btu/hr", lbh
    , tp2 , tp3 , rj3);
57 rj4=lbh*(tj2-tj3)*(0.90); //Btu/hr
58 printf("\t2.%.0 f(%.0 f-%.0 f) (0.91) = %.2 e Btu/hr\n",
    lbh , tj2 , tj3 , rj4);
59 printf("\tVapor temp. = 205 F \tdelT=37.6 F \t\
    tVapor temp. = 227 F \tdelT=14.8 F \n");
60 printf("\tUD=%0.0 f\t\t\t\t\t\t\tUD=%0.0 f\n", Ud3 , Ud4 );
61 A3=rj3/(37.6*Ud3); // ft ^2
62 A4=rj4/(14.8*Ud4); // ft ^2
63 printf("\tSurface ,A=%0.0 f ft ^2\t\t\t\t\tSurface ,A=%0.0 f
    ft ^2\n\n", A3 , A4);
64
65 rj5=lbh*(tp3-tp4)*(0.90); //Btu/hr
66 printf("\t2.%.0 f(%.0 f-%.0 f) (0.91) = %.2 e Btu/hr", lbh
    , tp3 , tp4 , rj4);
67 printf("\t(Use 2 heaters at 1300 ft ^2 each plus 1\n\
    t\t\t\t\t\t\t\ttheater at 1300 ft ^2 as spare)\n");
68 A5=rj5/(62.2*Ud5); // ft ^2
69 printf("\tVapor temp. = 181 F \tdelT=62.2 F \n\
    tSurface ,A=%0.0 f\n", A5);
70 printf("\t(Use 3 heaters at 100 ft ^2\n\teach plus 1
    heater as spare)\n\n");
71
72 v1=42600; //lb/hr
73 tt1=251; // F
74 printf("\t\t\t\t\tHEAT BALANCE\n");
75 printf("\tEffect\t\t\t\tBtu/hr\t\tEvaporation , l/hr\n")

```



```

        \n\t      1st effect used for vaccum pans
        .....%.0f\n",bb);
106 printf("\t(c) Equivalent evaporation from 1st effect
        \n\t      vapors used for clarified-juice heaters
        .....%.0f\n",cc);
107 printf("\t(d) Equivalent evaporation from 2d effect
        \n\t      vapors used for clarified-and raw-juice
        heaters.....%.0f\n",dd);
108 printf("\t(e) Equivalent evaporation from 3d effect
        \n\t      vapors used for raw-juice heaters
        .....%.0f\n",ee)
109 printf("\t(f) Equivalent evaporation from 4th effect
        \n\t      vapors used for raw-juice heaters
        .....%.0f\n",ff);
110 printf("\t
        -----\n")
111 printf("\t      Extrapolated evaporation
        .....%.0f\n",tto);
112 esq=tto/5;//lb/hr
113 printf("\t\tEstimated steam quantity = %.0f lb/hr\n"
        ,esq);
114 aesq=80600;//lb/hr
115 err = esq-aesq;//lb/hr
116 printf("\t\tActual steam required from final heat
        balance = %.0f lb/hr\n",aesq);
117 printf("\t\t\t\t\tError = %.0f lb/hr\n",err);
118 ta=15;
119 Q=14575000; //Btu/hr Total hourly evaporation
120 Gpm=Q/(500*(t2-tp4-ta));//From equation 14.4
121 printf("\tGallons per minute of Water required = %.0
        f gpm",Gpm);

```

Scilab code Exa 14.5 unit calculation

```

1 printf("\texample 14.5\n");
2 printf("\tapproximate values are mentioned in the
   book \n");
3 st1=280; // F
4 vt6=125; // F
5 odT=st1-vt6; // F
6 printf("\tOverall temperature difference = %.0f F \
   n",odT); //corresponding to 35 psig and 26 in. Hg
7 bpr(1)=10; // F
8 bpr(2)=8; // F
9 bpr(3)=7; // F
10 bpr(4)=6; // F
11 bpr(5)=5; // F
12 bpr(6)=5; // F
13 i=1;
14 tbpr=0;
15 while(i<7)
16     tbpr=tbpr+bpr(i);
17     i=i+1;
18 end
19 printf("\tThe estimated total BPR = %.0f F\n",tbpr
   ); //from fig. 14.36a
20 edT=odT-tbpr;
21 printf("\tEffective temperature difference = %.0f
   F\n",edT);
22 printf("\n\t\t\t\t\tEVAPORATOR SUMMARY\n\tAll bodies
   will consist of 300 2 in. OD, 10 BWG tubes 24
   long\n");
23 printf("\t
   -----
   n");
24 printf("\tItem\t\t\t\t\tEffects\n\t\t\t\t\t
   -----
   n\t\t\t\t\t1A\t\t1B\t\t2\t\t3\t\t4\t\t5\n");
25 printf("\t
   -----
   n");
26 printf("\t1.Steam flow , lb/hr\t\t20000\n\t2.Steam

```



```

pressure , psi/in.Hg\t35\t\t14.5\t\t4\t\t7\t\t16
.5\t\t22\n\t3.Steam temp, F\t\t\t280\t\t249\t\t
t224\t\t\t199\t\t\t174\t\t\t151\n\t4.delT, F\t\t\t21\t\t
\t17\t\t\t18\t\t\t19\t\t\t18\t\t\t21\n\t5.Liquor temp,
F\t\t\t259\t\t\t232\t\t\t206\t\t\t180\t\t\t156\t\t\t130\n\t
t6.BPR, F\t\t\t10\t\t\t8\t\t\t7\t\t\t6\t\t\t5\t\t\t5\n\t7
.Vapor temp, F\t\t\t259\t\t\t232\t\t\t206\t\t\t180\t\t\t
t156\t\t\t130\n\t8.Vapor pressure , pis/in.Hg\t14.5\t
\t\t4\t\t\t7\t\t\t6\t\t\t5\t\t\t5\n\t9.Lambda, Btu/lb\t\t
t946\t\t\t962\t\t\t978\t\t\t994\t\t\t1008\t\t\t1022\n\t10.
Liquor in , lb/hr\t\t\t73400\t\t\t88300\t\t\t101000\t\t\t
t113000\t\t\t72000\t\t\t72000\n\t11.Liquor out , lb/hr
\t\t\t56200\t\t\t73400\t\t\t88300\t\t\t101100\t\t\t58300\t\t\t
t54700\n\t12.Evaporation ,lb/hr\t\t\t17200\t\t\t14900\t\t\t
t\t\t12800\t\t\t11900\t\t\t13700\t\t\t17300\n\t13.Total
solids , \t\t\t38.9\t\t\t29.8\t\t\t24.7\t\t\t21.6\t\t\t18.7\t\t\t
t\t\t20.0\n\t14.A, ft^2\t\t\t\t3250\t\t\t3250\t\t\t3250\t\t\t
t3250\t\t\t3250\t\t\t3250\n\t15.UD,Btu/(hr)*(ft^2)*(
F)\t262\t\t\t295\t\t\t252\t\t\t251\t\t\t221\t\t\t221\n\t
t16.UD delT,Btu/(hr)*(ft^2)\t5510\t\t\t5000\t\t\t
t4530\t\t\t4770\t\t\t3980\t\t\t4650\n");//BPR values
from fig 14.36a
27 //Specific-heat data are given in Fig. 14.36b
28 ev(1)=17200; //lb/hr
29 ev(2)=14900; //lb/hr
30 ev(3)=12800; //lb/hr
31 ev(4)=11900; //lb/hr
32 ev(5)=13700; //lb/hr
33 ev(6)=17300; //lb/hr
34 i=1;
35 tev =0;
36 while(i<7)
37     tev = tev+ev(i);
38     i=i+1;
39 end
40 printf("\n\tTotal amount of water evaporated = %.0f
    lb/hr\n",tev);
41 ttev=tev/6;//lb/hr

```

```

42 printf("\tTheoretical amount of steam for a six-
    effect evaporator = %.0f lb/hr\n",ttev);
43 tev2=tev/(6*0.75); //lb/hr . order of 75 percent of
    theoretical
44 printf("\tSteam used for trail balance = %.0f lb/hr\
    n",tev2);
45 lq=(tev/6);
46 lq=lq+(lq*0.15);
47 printf("\tEstimate of the amount of evaporation in
    the first effect = %.0f lb/hr\n",lq);
48 lout6=54000; //lb/hr
49 lq2=lout6+lq+2200; //lb/hr
50 printf("\tEstimated discharge from second effect = %
    .0f lb/hr\n",lq2);
51 printf("\n\t\t\t\t\tHEAT BALANCE\n");
52 cw = 17750000/(500*(125-15-60)); //gpm, values from
    table 14.6
53 printf("\t\t\tCooling water at 60 F = %.0f gpm\n",cw
    );
54 printf("\t
    -----\
    n");
55 printf("\tEffect\t\t\t\t\tBtu/hr\t\t\tEvaporation , l/hr\n")
    ;
56 printf("\t
    -----\
    n");
57 sf=20000; //lb/hr
58 lqi=73400; //lb/hr
59 lqi2=88300
60 lt1=259; // F
61 lt2=232; // F
62 lt3=206; // F
63 ev=17200; //lb/hr
64 his=sf*924*0.97; //Btu/hr
65 printf("\t1.a.Heat in steam \t%.2e\n",his);
66 hl=lqi*(lt1-lt2)*0.82; //Btu/hr
67 printf("\t b.Heating liquor \t%.2e\n",hl);

```

```

68 hh=his-h1;
69 ev1=(hh)/946; //lb/hr
70 printf("\t c. Evaporation\t\t\t\t%.0f\n", ev1);
71 dif=lqi-ev1;
72 tft=(dif)*(lt1-209)*0.78;
73 printf("\t d. To flash tank\t%.1e", tft);
74 ev2=tft/978; //lb/hr
75 printf("\t\t\t%.0f\n", ev2);
76 printf("\t e. Flashed vapor=%.0f\n", ev2);
77 p=dif-ev2;
78 printf("\t f. product %.1e\n", p);
79 printf("\n\t2. a. Heat in 1st vapors\t%.3e\n", hh);
80 hl2=lqi2*(lt2-lt3)*0.85;
81 printf("\t b. Heating liquour\t%.2e\n", hl2);
82 ev3=(hh-hl2)/962;
83 printf("\t c. Evaporation=%.0f", ev3);
84
85 printf("\t\t\t\t%.0f\n", ev3);
86 lto1=lqi2-ev3;
87 printf("\t d. Liquor to 1b=%.0f\n", lto1);
88 //end

```

Scilab code Exa 14.6 Evoparator specification

```

1 printf("\texample 14.6\n");
2 printf("\tapproximate values are mentioned in the
   book \n");
3 st1=274; // F
4 vt6=115; // F
5 odT=st1-vt6; // F
6 printf("\tTotal temperature difference = %.0f F\n",
   ,odT); //corresponding to 35 psig
7 eb1=77; // F , From fig.14.38
8 eb2=26; // F , From fig.14.38
9 etd=odT-(eb1+eb2); // F

```

```

10 printf("\tThe effective temperature difference is %
      .0f F\n",etd);
11 printf("\n\t\t\tCAUSTIC EVAPORATOR MATERIAL BALANCE\
      n");
12 //Basis: 1 ton/hr NaOH
13 printf("\tCell liquour at 120 F \t\tWash at 80 F\n
      ");
14 printf("\t
      -----\n")
      ;
15 l1=2000; //Lb
16 l2=3800; //Lb
17 l3=17050; //Lb
18 lq=l1+l2+l3; //Lb
19 w1=340; //Lb
20 w2=1020; //Lb
21 w=w1+w2; //Lb
22 printf("\t8.75 prcent NaOH = %.0f\n\t16.6 prcent NaCl
      = %.0f\t\t25 prcent NaCl = %.0f\n",l1,l2,w1);
23 printf("\t74.65 prcent H2O = %.0f\t\t75 prcent H2O = %
      .0f\n",l3,w2);
24 printf("\tTotal cell liquor = %.0f\tTotoal wash = %
      .0f\n",lq,w);
25 printf("\n\t
      -----
      n");
26 printf("\t\t\t\tNaOH\t\tNaCl\t\tH2O ,Lb\tTotal ,Lb\n\t
      \t\t\t\tprcent\tLb\tprcent\tLb\n");
27 printf("\t
      -----
      n");
28 printf("\tOverall operation:\n\t Cell liquor
      ..... 8.75\t"+string(l1)+"\t16.60\t"+string(
      l2)+"\t"+string(l3)+"\t"+string(lq)+"\n");
29 printf("\t Wash ..... \t.... \t25.00\
      t"+string(w1)+"\t"+string(w2)+"\t"+string(w)+"\n"
      );
30 w11=l2+w1; //Lb

```



```

63 printf("\t\t\t\t(Basis = 1ton/hr NaOH)\n");
64 printf("\t
    ");
65 printf("\t\tEFFECT\t\t\tBtu/hr\t\tEvaopration , lb/hr
    \n");
66 hi=10500*930*0.974; //Btu/hr
67 hl=18230*(246-150)*0.83; //Btu/hr
68 rh=hi-hl; //Btu/hr
69 hc=300000; //Btu/hr
70 hv=rh-hc; //Btu/hr
71 evv=hv/997; //lb/hr
72 printf("\t1.a.Heat in steam\t\t%.1e\n\t b.Heating
    liquor\t\t%.2e\n\t c.Resultant heat\t\t%.2ef\n\t
    d.Heat of concentrate\t\t%.0e\n\t e.Heat of
    vapors\t\t%.2e\t%.0f\n",hi,hl,rh,hc,hv,evv);
73 s1=1.35;
74 G1=V*s1*62.5*3600; //lb/((hr)*(ft^2))
75 printf("\n\tG = V(s*62.5*3600) = %.2e lb/((hr)*(ft
    ^2))\n",G1);
76 UD1=700; //Btu/((hr)*(ft^2)*( F ))
77 //Using thermal characteristics for this solution
78 printf("\tUD = %.0f Btu/((hr)*(ft^2)*( F ))\n",UD1);
79 //As for effect I:
80 printf("\n\t
    ");
81 printf("\n\ttx , F \tw, lb/hr\t\t\tdelT\tUC\tA, ft^2\tat ,
    flow area\tGcalc\t\tUcalc\n\t\t\t\t\t\t\t\tper pass
    , ft^2\n");
82 printf("\t
    ");
83 printf("\t146\t2400000\t\t25.4\t700\t620\t0.80\t\t
    2790000\t\t700\n\t146.5\t2160000\t\t25.2\t700\t
    683\t0.89\t\t2430000\n");
84 //end

```



```

        tEvaporation , lb/hr\t\t20600\t\t22500\t\t23500\t\t
        t30000\t\t17600\t\t19100\n\t Brix (out)\t\t\t\t\t\t\t
        \t\t\t\t\t\t\t\t\t\t30\n\tCondenser water , gpm\t\t\t\t\t
        t455\t\t\t\t\t\t\t\t\t365\n");
11 printf("\n\t\t\t\t\tHEAT BALANCE-STRAIGHT TRIPLE
        EFFECT\n\t\t\t\t\tCondenser water = 455 gpm\n");
12 printf("\t
        _____\
        n");
13 printf("\tEffect\t\t\t\t\tBtu/hr\t\tEvaporation , l/hr\n")
        ;
14 printf("\t
        _____\
        n");
15 sf=22400; //lb/hr
16 lc=100000; //lb/hr
17 t1=238; // F
18 t2=230; // F
19 his=sf*940*0.97; //Btu/hr
20 hlq=lc*(t1-t2)*0.92; //Btu/hr
21 hd=his-hlq; //Btu/hr
22 eva=(hd)/954; //lb/hr
23 l2d=lc-eva;
24 printf("\t1.a.Heat in steam\t%.2e\n\t b.Heating
        liquor\t%.2e\n\t c.Evaporation\t\t%.4e/954\t%.0f
        \n\t d.Liquor to 2d = %.0f",his,hlq,hd,eva,l2d);
25 //end
        _____

```

Chapter 15

Vaporizers Evaporators and Reboilers

Scilab code Exa 15.1 Calculation of the Average Specific Volume

```
1 printf("\t example 15.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 ts=250;
4 T1=400;
5 T2=300;
6 w=10000; // lb/hr
7 W=150000; // lb/hr
8 l=945.3; // Btu/(lb) , table 7
9 Q=((w)*(l)); // Btu/hr
10 printf("\t total heat required for steam is : %.2e
   Btu/hr \n",Q);
11 C=0.63; // Btu/(lb)*(F)
12 Q=((W)*(C)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for kerosene is : %.2
   e Btu/hr \n",Q);
14 delT1=T2-ts; //F
15 delT2=T1-ts; // F
16 printf("\t delT1 is : %.0f F \n",delT1);
```

```

17 printf("\t delt2 is : %.0f F \n",delt2);
18 LMTD=((delt2-delt1)/((2.3)*(log10(delt2/delt1))));
19 printf("\t LMTD is :%.0f F \n",LMTD);
20 UD=100;
21 A=(Q/(UD*LMTD));
22 printf("\t A : %.2e ft ^2 \n",A);
23 WC=94500; // Btu/F
24 v1=0.017; // ft^3/lb, from table 7
25 vv=13.75; // ft^3/lb, from table 7
26 printf("\t By the law of mixtures \n");
27 // Assume 80 per cent of the outlet fluid is vapor
28 v2=(0.8*vv)+(.2*v1);
29 printf("\t v2 : %.0f ft ^3/lb \n",v2);
30 vav=(WC*(v2-v1)/(UD*A))-((WC*(T2-ts)/(1*w))*(vv-v1))
    +v1;
31 printf("\t vav : %.2f ft ^3/lb \n",vav);
32 printf("\t By the approximate method \n");
33 vav1=(v1+v2)/(2);
34 printf("\t vav : %.2f ft ^3/lb \n",vav1);
35 row=62.5;
36 rowac=(1/vav);
37 s=(rowac/row);
38 printf("\t actual density : %.3f lb/ft ^3 \n",rowac);
39 printf("\t s : %.4f \n",s);
40 rowap=(1/vav1);
41 s=(rowap/row);
42 printf("\t approximate density : %.3f lb/ft ^3 \n",
    rowac);
43 printf("\t s : %.4f \n",s);
44 // end

```

Scilab code Exa 15.2 Vaporizer or Pump through Reboiler with Isothermal Boiling

```

1 printf("\t example 15.2 \n");

```

```

2  printf("\t approximate values are mentioned in the
      book \n");
3  t1=108; // inlet cold fluid ,F
4  t2=235; // outlet cold fluid ,F
5  Ts=338;
6  Wp=24700; // lb/hr
7  Wv=19750; // lb/hr
8  w=4880; // lb/hr
9  printf("\t 1.for heat balance \n");
10 Ht1=162; // enthalpy at t1, Btu/lb, fig 9
11 Ht2=248; // enthalpy at t2, Btu/lb, fig 9
12 qp=(Wp*(Ht2-Ht1)); // for preheat
13 printf("\t total heat required for preheat of butane
      is : %.2e Btu/hr \n",qp);
14 Ht3=358; // enthalpy of vapour at t2, Btu/lb, fig 9
15 qv=Wv*(Ht3-Ht2);
16 printf("\t total heat required for vapourisation of
      butane is : %.2e Btu/hr \n",qv);
17 Q=qp+qv;
18 printf("\t total heat required for butane is : %.2e
      Btu/hr \n",Q);
19 printf("\t for steam \n");
20 l=880.6; // Btu/(lb), table 7
21 Q=((w)*(l)); // Btu/hr
22 printf("\t total heat required for steam is : %.2e
      Btu/hr \n",Q);
23 deltp=158.5; // F, from eq 5.14
24 deltv=103; // F eq 5.14
25 Wp1=(qp/deltp);
26 printf("\t Wp1 is : %.2e lb/hr \n",Wp1);
27 Wv1=(qv/deltv);
28 printf("\t Wv1 is : %.2e lb/hr \n",Wv1);
29 W=(Wp1+Wv1);
30 printf("\t W is : %.2e lb/hr \n",W);
31 deltt=(Q/W);
32 printf("\t weighted deltt is : % .1f F \n",deltt);
33 Tc=((Ts)+(Ts))/(2); // caloric temperature of hot
      fluid ,F

```

```

34 printf("\t caloric temperature of hot fluid is : %.1
    f F \n",Tc);
35 tc=((t1)+(t2))/(2); // caloric temperature of cold
    fluid ,F
36 printf("\t caloric temperature of cold fluid is : %
    .1 f F \n",tc);
37 printf("\t hot fluid:inner tube side ,steam \n");
38 Nt=76;
39 n=2; // number of passes
40 L=16; //ft
41 at1=0.594; // flow area ,table 10, in^2
42 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
43 printf("\t flow area is : %.3 f ft^2 \n",at);
44 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
45 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
46 mu1=0.0363; // at 338F, fig 15,lb/(ft)*(hr)
47 D=0.0725; // ft
48 Ret=((D)*(Gt)/mu1); // reynolds number
49 printf("\t reynolds number is : %.1e \n",Ret);
50 hio=1500; // condensing steam ,Btu/(hr)*(ft^2)*(F)
51 printf("\t hio is : %.0f Btu/(hr)*(ft^2)*(F) \n",hio
    );
52 printf("\t cold fluid:shell side ,butane \n");
53 printf("\t preheating \n");
54 ID=15.25; // in
55 C=0.25; // clearance
56 B=5; // baffle spacing ,in
57 PT=1.25;
58 as=((ID*C*B)/(144*PT)); // flow area ,ft^2
59 printf("\t flow area is : %.3 f ft^2 \n",as);
60 Gs=(Wp/as); // mass velocity ,lb/(hr)*(ft^2)
61 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
62 mu2=0.278; // at 172F,lb/(ft)*(hr), from fig.14
63 De=0.0825; // from fig.28,ft
64 Res=((De)*(Gs)/mu2); // reynolds number

```

```

65 printf("\t reynolds number is : %.2e \n",Res);
66 jH=159; // from fig.28
67 Z=0.12; // Z=k*((c)*(mu1)/k)^(1/3), fig 16
68 hop=((jH)*(1/De)*(Z)); //using eq.6.15b,Btu/(hr)*(ft
^2)*(F)
69 printf("\t individual heat transfer coefficient is :
%.0f Btu/(hr)*(ft^2)*(F) \n",hop);
70 Up=((hio)*(hop)/(hio+hop)); // clean overall
coefficient ,eq 6.38,Btu/(hr)*(ft^2)*(F)
71 printf("\t clean overall coefficient for preheating
: %.0e Btu/(hr)*(ft^2)*(F) \n",Up);
72 Ap=(qp/(Up*deltP));
73 printf("\t clean surface required for preheating : %
.0f ft^2 \n",Ap);
74 printf("\t for vapourisation \n");
75 mu2=0.242; // at 172F,lb/(ft)*(hr), from fig.14
76 Res=((De)*(Gs)/mu2); // reynolds number
77 printf("\t reynolds number is : %.2e \n",Res);
78 jH=170; // from fig.28
79 Z=0.115; // Z=k*((c)*(mu1)/k)^(1/3), fig 16
80 hov=((jH)*(1/De)*(Z)); //using eq.6.15b,Btu/(hr)*(ft
^2)*(F)
81 printf("\t individual heat transfer coefficient is :
%.0f Btu/(hr)*(ft^2)*(F) \n",hov);
82 Uv=((hio)*(hov)/(hio+hov)); // clean overall
coefficient ,eq 6.38,Btu/(hr)*(ft^2)*(F)
83 printf("\t clean overall coefficient for
vapourisation : %.0f Btu/(hr)*(ft^2)*(F) \n",Uv);
84 Av=(qv/(Uv*deltv));
85 printf("\t clean surface required for vapourisation
: %.0f ft^2 \n",Av);
86 Ac=Ap+Av;
87 printf("\t total clean surface : %.1e ft^2 \n",Ac);
88 UC=((Up*Ap)+(Uv*Av))/(Ac);
89 printf("\t weighted clean overall coefficient : %.0f
Btu/(hr)*(ft^2)*(F) \n",UC);
90 A2=0.2618; // actual surface supplied for each tube,
ft^2,from table 10

```

```

91 A=(Nt*L*A2); // ft^2
92 printf("\t total surface area is : %.0f ft^2 \n",A);
93 UD=((Q)/((A)*(delt)));
94 printf("\t actual design overall coefficient is : %
    .1f Btu/(hr)*(ft^2)*(F) \n",UD);
95 // A total of 170 ft2 are required of which 103 are
    to be used for vaporization. For the total
    surface required 318 ft2 will be provided. It can
    be assumed, then, that the surface provided for
    vaporization is 193ft^2
96 // then flux is Q/A=10700, which is with in
    satisfactory levels.
97 Rd=((UC-UD)/((UD)*(UC))); // (hr)*(ft^2)*(F)/Btu
98 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
99 printf("\t pressure drop for inner pipe \n");
100 f=0.000165; // friction factor for reynolds number
    62000, using fig.26
101 s=0.00413;
102 phyt=1;
103 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt)))/(2); // using eq.7.45, psi
104 printf("\t delPt is : %.2f psi \n",delPt);
105 printf("\t allowable delPa is negligible \n");
106 printf("\t pressure drop for annulus \n");
107 printf("\t preheating \n");
108 f=0.00145; // friction factor for reynolds number
    69200, using fig.29
109 Lp=(L*Ap/Ac); //ft
110 printf("\t length of preheat zone : %.1f ft \n",Lp);
111 N=(12*Lp/B); // number of crosses ,using eq.7.43
112 printf("\t number of crosses are : %.0f \n",N);
113 s=0.5; // for reynolds number 69200,using fig.6
114 Ds=1.27; // fig 28
115 phys=1;
116 delPsp=((f*(Gs^2)*(Ds)*(N))/(5.22*(10^10)*(De)*(s)*(
    phys))); // using eq.7.44, psi
117 printf("\t delPsp is : %.1f psi \n",delPsp);

```

```

118 printf("\t vapourisation \n");
119 f=0.00142;
120 Lv=9.7; // Lv=L-Lp
121 Nv=(12*Lv/B); // number of crosses, using eq.7.43
122 printf("\t number of crosses are : %.0f \n",Nv);
123 s=0.28;
124 delPsv=((f*(Gs^2)*(Ds)*(Nv))/(5.22*(10^10)*(De)*(s)
        *(1))); // using eq 12.47, psi
125 printf("\t delPsv is : %.1f psi \n",delPsv);
126 delPS=delPsp+delPsv;
127 printf("\t delPS is : %.1f psi \n",delPS);
128 printf("\t allowable delPa is 5 psi \n");
129 //end

```

Scilab code Exa 15.3 Calculation of a Kettle Reboiler

```

1 printf("\t example 15.3 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 ts=400;
4 T1=575;
5 T2=475;
6 W=28100; // lb/hr
7 w=34700; // lb/hr
8 printf("\t 1.for heat balance \n");
9 HT1=290; // enthalpy at T1, Btu/lb, fig 11
10 HT2=385; // enthalpy at T2, Btu/lb, fig 11
11 Q=(W*(HT2-HT1)); // for preheat
12 printf("\t total heat required for gasoline is : %.2
    e Btu/hr \n",Q);
13 c=0.77; // Btu/(lb), table 7
14 Q=((w)*(c)*(T1-T2)); // Btu/hr
15 printf("\t total heat required for gasoil is : %.2e
    Btu/hr \n",Q);
16 delT=118; // F eq 5.14

```



```

17 S=((T2-ts)/(T1-ts));
18 printf("\t S is : %.3f \n",S);
19 Kc=0.37; // fig 17
20 Fc=0.42;
21 Tc=(T2+(0.42*(T1-T2)));
22 printf("\t Tc is : %.0f F \n",Tc);
23 printf("\t hot fluid:inner tube side ,gasoil \n");
24 Nt=68;
25 n=6; // number of passes
26 L=12; //ft
27 at1=0.546; // flow area ,table 10, in^2
28 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
    .7.48
29 printf("\t flow area is : %.3f ft^2 \n",at);
30 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
31 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
32 mu1=0.65; // at 517F, fig 14,lb/(ft)*(hr)
33 D=0.0694; // ft
34 Ret=((D)*(Gt)/mu1); // reynolds number
35 printf("\t reynolds number is : %.2e \n",Ret);
36 jH=220; // from fig.24
37 Z=0.118; // Z=k*((c)*(mu1)/k)^(1/3), fig 16
38 Hi=((jH)*(1/D)*(Z)); //hi/phyt, Hi=( )using eq.6.15d,
    Btu/(hr)*(ft^2)*(F)
39 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Hi);
40 Hio=((Hi)*(0.834/1)); //Hio=(hio/phyt), using eq.6.9
41 printf("\t Correct Hi0 to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Hio);
42 // (mul/muw)^(0.14) is negligible
43 printf("\t cold fluid:shell side ,gasoline \n");
44 ho=300; // assumption
45 tw=(ts)+(((Hio)/(Hio+ho))*(Tc-ts)); // from eq.5.31
46 printf("\t tw is : %.0f F \n",tw);
47 deltw=(tw-ts);
48 printf("\t deltw : %.0f F \n",deltw);
49 // from fig 15.11, ho>300

```

```

50 Uc=((Hio)*(ho)/(Hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft ^2)*(F)
51 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft ^2)*(F) \n",Uc);
52 A2=0.2618; // actual surface supplied for each tube,
    ft ^2,from table 10
53 A=(Nt*L*A2); // ft ^2
54 printf("\t total surface area is : %.0f ft ^2 \n",A);
55 UD=((Q)/((A)*(delt)));
56 printf("\t actual design overall coefficient is : %
    .1f Btu/(hr)*(ft ^2)*(F) \n",UD);
57 // check for max. flux=Q/A=12500.(satisfactory)
58 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft ^2)*(F)/Btu
59 printf("\t actual Rd is : %.4f (hr)*(ft ^2)*(F)/Btu \
    n",Rd);
60 printf("\t pressure drop for inner pipe \n");
61 f=0.00015; // friction factor for reynolds number
    85700, using fig.26
62 s=0.71;
63 phyt=1;
64 delPt=((f*(Gt ^2)*(L)*(n))/(5.22*(10 ^10)*(D)*(s)*(
    phyt))); // using eq.7.45 ,psi
65 printf("\t delPt is : %.1f psi \n",delPt);
66 X1=0.09; // X1=((V ^2)/(2*g)), for Gt 1060000, using
    fig.27
67 delPr=((4*n*X1)/(s)); // using eq.7.46 ,psi
68 printf("\t delPr is : %.1f psi \n",delPr);
69 delPT=delPt+delPr; // using eq.7.47 ,psi
70 printf("\t delPT is : %.1f psi \n",delPT);
71 printf("\t allowable delPa is 10psi \n");
72 printf("\t delPs is negligible \n");
73 //end

```

Scilab code Exa 15.4 Calculation of a Once through Horizontal Thermosyphon Reboiler

```

1 printf("\t example 15.4\n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 t1=315; // inlet cold fluid ,F
4 t2=335; // outlet cold fluid ,F
5 T1=525;
6 T2=400;
7 Wv=29000; // lb/hr
8 Ws=38500; // lb/hr
9 w=51000; // lb/hr
10 printf("\t 1.for heat balance \n");
11 Ht1=238; // enthalpy at t1, Btu/lb, fig 9
12 Ht2=252; // enthalpy at t2, Btu/lb, fig 9
13 Ht3=378; // enthalpy of vapour at t2
14 qv=(Wv*(Ht3-Ht2)); // for preheat
15 printf("\t qv is : %.2e Btu/hr \n",qv);
16 qs=Ws*(Ht2-Ht1);
17 printf("\t qs is : %.2e Btu/hr \n",qs);
18 Q=qs+qv;
19 printf("\t total heat required for naphtha is : %.2e
   Btu/hr \n",Q);
20 c=0.66; // Btu/(lb)(F)
21 Q=((w)*(c)*(T1-T2)); // Btu/hr
22 printf("\t total heat required for gasoil is : %.2e
   Btu/hr \n",Q);
23 delT1=T2-t1; //F
24 delT2=T1-t2; // F
25 printf("\t delT1 is : %.0f F \n",delT1);
26 printf("\t delT2 is : %.0f F \n",delT2);
27 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
28 printf("\t LMTD is :%.0f F \n",LMTD);
29 R=((T1-T2)/(t2-t1));
30 printf("\t R is : %.2f \n",R);
31 S=((t2-t1)/(T1-t1));
32 printf("\t S is : %.3f \n",S);
33 printf("\t FT is 0.97 \n"); // from fig 18
34 delT=(0.97*LMTD); // F
35 printf("\t delT is : %.0f F \n",delT);

```

```

36 X=((delt1)/(delt2)); // fig 17
37 printf("\t ratio of two local temperature difference
      is : %.3f \n",X);
38 Fc=0.41; // from fig.17
39 Kc=0.42;
40 Tc=((T2)+((Fc)*(T1-T2))); // caloric temperature of
      hot fluid ,F
41 printf("\t caloric temperature of hot fluid is : %.0
      f F \n",Tc);
42 tc=((t1)+((Fc)*(t2-t1))); // caloric temperature of
      cold fluid ,F
43 printf("\t caloric temperature of cold fluid is : %
      .0f F \n",tc);
44 printf("\t hot fluid:inner tube side,steam \n");
45 Nt=116;
46 n=8; // number of passes
47 L=12; //ft
48 at1=0.546; // flow area ,table 10, in^2
49 at=((Nt*at1)/(144*n)); // total area ,ft^2,from eq
      .7.48
50 printf("\t flow area is : %.3f ft^2 \n",at);
51 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
52 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
      ",Gt);
53 mu1=1.09; // at 451F, fig 14,lb/(ft)*(hr)
54 D=0.0695; // ft
55 Ret=((D)*(Gt)/mu1); // reynolds number
56 printf("\t reynolds number is : %.2e \n",Ret);
57 jH=168; // from fig.24
58 Z=0.142; // Z=k*((c)*(mu1)/k)^(1/3), fig 16
59 Hi=((jH)*(1/D)*(Z)); // , Hi=(hi/phyt) using eq.6.15d,
      Btu/(hr)*(ft^2)*(F)
60 printf("\t individual heat transfer coefficient is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",Hi);
61 Hio=((Hi)*(0.834/1)); //Hio=(hio/phyt), using eq.6.9
62 printf("\t Correct Hio to the surface at the OD is :
      %.0f Btu/(hr)*(ft^2)*(F) \n",Hio);
63 printf("\t cold fluid:shell side,naphtha \n");

```

```

64 ho1=200; // assumption
65 tw=(tc)+(((Hio)/(Hio+ho1))*(Tc-tc)); // from eq
    .5.31, calculation mistake
66 printf("\t tw is : %.0f F \n",tw);
67 deltw=(tw-tc);
68 printf("\t deltw : %.0f F \n",deltw);
69 // from fig 15.11, hv>300, hs=60
70 Av=(qv/300);
71 As=qs/60;
72 printf("\t qv/hv : %.3e \n",Av);
73 printf("\t qs/hs : %.0e \n",As);
74 A1=As+Av;
75 printf("\t A : %.3e \n",A1);
76 ho=(Q/A1);
77 printf("\t ho : %.0f \n",ho);
78 Uc=(((Hio)*(ho)/(Hio+ho))); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
79 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
80 A2=0.2618; // actual surface supplied for each tube,
    ft^2,from table 10
81 A=(Nt*L*A2); // ft^2
82 printf("\t total surface area is : %.0f ft^2 \n",A);
83 UD=(((Q)/((A)*(delt)));
84 printf("\t actual design overall coefficient is : %
    .1f Btu/(hr)*(ft^2)*(F) \n",UD);
85 // check for max. flux=Q/A=11500.(satisfactory)
86 Rd=(((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
87 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
88 printf("\t pressure drop for inner pipe \n");
89 f=0.000168; // friction factor for reynolds number
    59200, using fig.26
90 s=0.73;
91 phyt=1;
92 delPt=(((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45, psi
93 printf("\t delPt is : %.1f psi \n",delPt);

```

```

94 X1=0.11; // X1=((V^2)/(2*g)), for Gt 1060000, using
    fig.27
95 delPr=((4*n*X1)/(s)); // using eq.7.46, psi
96 printf("\t delPr is : %.1f psi \n",delPr);
97 delPT=delPt+delPr; // using eq.7.47, psi
98 printf("\t delPT is : %.1f psi \n",delPT);
99 printf("\t allowable delPa is negligible \n");
100 printf("\t pressure drop for annulus \n");
101 Af=(3.14*(21.25^2-(116))/8);
102 printf("\t flow area : %.0f in^2 \n",Af);
103 as=0.917; // ft^2
104 p=(3.14*21.25/2)+(3.14*1*116/2)+(21.25);
105 printf("\t wetted perimeter : %.1f in \n",p);
106 De=0.186; // ft
107 Gs=(Ws/(2*as)); // mass velocity, lb/(hr)*(ft^2)
108 printf("\t mass velocity is : %.1e lb/(hr)*(ft^2) \n
    ",Gs);
109 mu2=0.435; // at 315F, fig 14, lb/(ft)*(hr)
110 Res=((De)*(Gs)/mu2); // reynolds number
111 printf("\t reynolds number is : %.2e \n",Ret);
112 f=0.00028; // using fig.26
113 row=0.337; // fig 13.14
114 // soutlet max=0.071,
115 s=0.35; // using fig.6
116 phys=1;
117 delPs=((f*(Gs^2)*(L))/(5.22*(10^10)*(De)*(s)*(phys))
    ); // using eq.7.44, psi
118 printf("\t delPs is : %.4f psi \n",delPs);
119 printf("\t allowable delPa is .25 psi \n");
120 //end

```

Scilab code Exa 15.5 Calculation of a Vertical Thermosyphon Reboiler

```

1 printf("\t example 15.5\n");
2 printf("\t approximate values are mentioned in the

```

```

    book \n");
3 W=40800; // lb/hr
4 w=4570; // lb/hr
5 printf("\t 1.for heat balance \n");
6 Ht1=241; // enthalpy of liquid at 228F, Btu/lb, fig
    9
7 Ht2=338; // enthalpy of vapour at 228F, Btu/lb, fig 9
8 Q=(W*(Ht2-Ht1));
9 printf("\t total heat required for butane is : %.2e
    Btu/hr \n",Q);
10 l=868; // Btu/(lb), table 7
11 Q=((w)*(l)); // Btu/hr
12 printf("\t total heat required for steam is : %.2e
    Btu/hr \n",Q);
13 delT=125; // delT=LMTD, isothermal boiling, eq 5.14
14 // Tc and tc: Both streams are isothermal
15 printf("\t trail 1 \n");
16 A1=((Q)/((12000))); // Q/A1 =12000, first trial
    should always be taken for the maximum allowable
    flux
17 printf("\t A1 is : %.1e ft^2 \n",A1);
18 a1=0.1963; // ft^2/lin ft
19 L=16;
20 N1=(A1/(L*a1)); // table 10
21 printf("\t number of tubes are : %.0f \n",N1);
22 N2=109; // assuming one tube passes, 13.25-in ID,
    from table 9
23 A2=(N2*L*a1); // ft^2
24 printf("\t total surface area is : %.0f ft^2 \n",A2)
    ;
25 UD=((Q)/((A2)*(delT)));
26 printf("\t correct design overall coefficient is : %
    .1f Btu/(hr)*(ft^2)*(F) \n",UD);
27 // Assume 4: 1 recirculation ratio
28 rowv=(58/(359*(688/492)*(14.7/290))); // eq 15.18
29 printf("\t vapour density : %.2f lb/ft^3 \n",rowv);
30 Vv=0.44;
31 V1=0.0372; // fig 6

```

```

32 W1=4*W;
33 printf("\t weight flow of recirculated liquid : %.3e
        lb/hr \n",W1);
34 VL=W1*Vl;
35 VV=W*Vv;
36 printf("\t volume of liquid : %.2e ft^3 \n",VL);
37 printf("\t volume of vapour : %.3e ft^3 \n",VV);
38 V=VL+VV;
39 printf("\t total volume out of reboiler : %.3e ft^3
        \n",V);
40 vo=(V/(W1+W));
41 printf("\t vo is : %.4f ft^3/lb \n",vo);
42 Pl=((2.3*16)/(144*(vo-Vl)))*(log10(vo/Vl));
43 printf("\t pressure leg : %.1f psi \n",Pl);
44 printf("\t frictional resistance \n");
45 Nt=109;
46 n=1; // number of passes
47 at1=0.302; // flow area, table 10, in^2
48 at=((Nt*at1)/(144*n)); // total area, ft^2, from eq
        .7.48
49 printf("\t flow area is : %.3f ft^2 \n",at);
50 Gt=((W1+W)/(at)); // mass velocity, lb/(hr)*(ft^2)
51 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
        ",Gt);
52 mu1=0.242; // at 228F, fig 14, lb/(ft)*(hr)
53 D=0.0517; // ft
54 Ret=((D)*(Gt)/mu1); // reynolds number
55 printf("\t reynolds number is : %.1e \n",Ret);
56 f=0.000127; // using fig.26
57 s=0.285;
58 phyt=1;
59 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(D)*(s)*(
        phyt))); // using eq.7.45, psi
60 printf("\t delPt is : %.2f psi \n",delPt);
61 P=Pl+delPt;
62 printf("\t total resistance : %.2f psi \n",P);
63 F=(16*0.43*62.5/144);
64 printf("\t driving force : %.2f psi \n",F);

```



```

65 // The resistances are greater than the hydrostatic
    head can provide; hence the recirculation ratio
    will be less than 4: 1
66 printf("\t trial 2 \n"); // Assume 12'0" tubes and
    4:1 recirculation ratio
67 A1=((Q)/((12000))); // Q/A1 =12000, first trial
    should always be taken for the maximum allowable
    flux
68 printf("\t A1 is : %.1e ft^2 \n",A1);
69 a1=0.1963; // ft^2/lin ft
70 L=12;
71 N1=(A1/(L*a1)); // table 10
72 printf("\t number of tubes are : %.0f \n",N1);
73 N2=151; // assuming one tube passes, 15.25-in ID,
    from table 9
74 A2=(N2*L*a1); // ft^2
75 printf("\t total surface area is : %.0f ft^2 \n",A2)
    ;
76 UD=((Q)/((A2)*(delt)));
77 printf("\t correct design overall coefficient is : %
    .0f Btu/(hr)*(ft^2)*(F) \n",UD);
78 P1=((2.3*12)/(144*(vo-V1)))*(log10(vo/V1));
79 printf("\t pressure leg : %.1f psi \n",P1);
80 printf("\t frictional resistance \n");
81 Nt=151;
82 n=1; // number of passes
83 at1=0.302; // flow area, table 10, in^2
84 at=((Nt*at1)/(144*n)); // total area, ft^2, from eq
    .7.48
85 printf("\t flow area is : %.3f ft^2 \n",at);
86 Gt=((W1+W)/(at)); // mass velocity, lb/(hr)*(ft^2)
87 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
88 mu1=0.242; // at 228F, fig 14, lb/(ft)*(hr)
89 D=0.0517; // ft
90 Ret=((D)*(Gt)/mu1); // reynolds number
91 printf("\t reynolds number is : %.2e \n",Ret);
92 f=0.000135; // using fig.26

```

```

93 s=0.285;
94 phyt=1;
95 delPt=((f*(Gt^2)*(12)*(n))/(5.22*(10^10)*(D)*(s)*(
    phyt))); // using eq.7.45 , psi
96 printf("\t delPt is : %.2f psi \n",delPt);
97 P=P1+delPt;
98 printf("\t total resisitance : %.2f psi \n",P);
99 F=(12*0.43*62.5/144);
100 printf("\t driving force : %.2f psi \n",F);
101 // Since the driving force is slightly greater than
    the resistances , a recirculation ratio better
    than 4:1 is assured.
102 printf("\t hot fluid : shell side ,steam \n");
103 ho=1500; // condensing steam
104 printf("\t cold fluid:inner tube side , butane \n");
105 jH=330; // from fig.24
106 Z=0.115; // Z=k*((c)*(mul)/k)^(1/3) , fig 16
107 Hi=((jH)*(1/D)*(Z)); // , Hi=(hi/phyt) using eq.6.15d,
    Btu/(hr)*(ft^2)*(F)
108 printf("\t individual heat transfer coefficient is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Hi);
109 Hio=((300)*(0.62/0.75)); //Hio=(hio/phyp) , using eq
    .6.9
110 printf("\t Correct Hio to the surface at the OD is :
    %.0f Btu/(hr)*(ft^2)*(F) \n",Hio);
111 Uc=((Hio)*(ho)/(Hio+ho)); // clean overall
    coefficient ,Btu/(hr)*(ft^2)*(F)
112 printf("\t clean overall coefficient is : %.0f Btu/(
    hr)*(ft^2)*(F) \n",Uc);
113 UD=89;
114 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
115 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
    n",Rd);
116 // end

```

Scilab code Exa 15.6 Calculation of the Reboiler Duty

```
1 printf("\t example 15.6\n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 //20000=WD+WB;
4 //0.99*WD+(0.05*WB)=(20000*.5);
5 // solving above two eq. we get WD and WB
6 WD=9570; // lb/hr
7 WB=10430; // lb/hr
8 HB1=108; // fig 3 and 12
9 HD1=85.8; //fig 3 and 12
10 HDv=253.8; // fig 3 and 12
11 HF1=92; // fig 3 and 12
12 l=153; // fig 3 and 12
13 QR=((2.54+1)*WD*(HDv))-(2.54*WD*HD1)+(WB*HB1)
   -(20000*HF1);
14 printf("\t total heat duty : %.1e Btu/hr \n",QR);
15 Q=QR/153;
16 printf("\t total heat duty : %.2e lb/hr \n",Q);
17 // end
```

Scilab code Exa 15.7 Distillation of a Binary Mixture

```
1 printf("\t example 15.7 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3
4 //Basis: One hour
5 //20000=WD+WB , material balance
6 //0.99*WD+(0.05*WB)=(20000*0.5) , Benzene balance
7 // solving above two eq. we get WD and WB
8 WD=9570; // lb/hr
9 WB=10430; // lb/hr
10
```

```

11 //Compositions and Boiling Points
12 //Feed
13 l1 = 10000; //Lb/hr , C6H4
14 l2 = 10000; //Lb/hr , C7H8
15 lb = l1+l2; //Lb/hr
16 printf("\ttotal Lb/hr is %.0f\n",lb);
17 mo1 = 78.1; //Mol. wt. , C6H6
18 mo2 = 93.1; //Mol. wt , C7H8
19 mh1 = 128.0; //Mol/hr , C6H6
20 mh2 = 107.5; //Mol/hr , C7H8
21 mh = mh1 + mh2; // Mol/hr
22 printf("\ttotal Mol/hr is %.1f\n",mh);
23 x1 = mh1/mh;
24 printf("\tx1 of C6H6 is %.3f\n",x1);
25 x2 = mh2/mh;
26 printf("\tx1 of C7H8 is %.3f\n",x2);
27 x = x1+x2;
28 printf("\tTotal x1 is %.3f\n",x);
29 Pp1= 1380; // 214 F
30 Pp2=575; // 214 F
31 xp1 = x1*Pp1;
32 printf("\tx1Pp1 of C6H6 is %.0f\n",xp1);
33 xp2 = x2*Pp2;
34 printf("\tx1Pp1 of C7H8 is %.0f\n",xp2);
35 sxp = xp1 + xp2;
36 printf("\tTotal x1Pp1 is %.0f\n",sxp);
37 y1 = xp1/sxp;
38 printf("\ty1 of C6H6 is %.3f\n",y1);
39 y2 = xp2/sxp;
40 printf("\ty1 of C7H8 is %.3f\n",y2);
41 y = y1+y2;
42 printf("\tTotal y1 is %.3f\n",y);
43
44
45 w1 = 0.558; //from eq 15.42
46 printf("\t(WR' /V =((xD - yF) /.(xD - xF))) = %.3fmol/
mol\n",w1);
47 wD=1;

```

```

48 xD = 0.992;
49 //V = WR' + WD
50 // WR'/V = 0.558
51 //Solving , WR' = (WR' * 0.558) + (0.558 * WD)
52 Wr = 1.27; // mol reflux/mol distillate
53 printf("\tWR' = %.2f (mol reflux)/(mol distillate)\n
",Wr);
54 Wr1 = Wr * 2; // mol/ mol distillate
55 printf("\tAssumed 200 percent of the theoretical
minimum reflux as economic\n\tWR = %.2f(mol)/(mil
distillate)\n",Wr1);
56 in = (wD * xD)/(Wr1 + 1); //intercept for the upper
operating line
57 printf("\tThe intercept for the upper operating line
= %.3f\n",in);
58 p = 13; // From fig. 15.23, connecting the
corresponding lines
59 printf("\tConnecting the corresponding line in Fig.
15.23, plates required: %.0f\n",p);
60 fp = 7; // From fig. 15.23, connecting the
corresponding lines
61 printf("\tFeed plate is %.0fth(from top)\n",fp);
62 d=122.5;
63 tf = Wr1 * d;
64 printf("\tTotal reflux is %.1f\n",tf);
65 printf("\t\t\t\t\tHeat balances");
66
67 //Heat Balances
68 l1 = 33900;
69 l2 = 9570;
70 l3 = 24330;
71 b1 = 253.8;
72 b2 = 85.8;
73 b3 = 85.8;
74 bt1 = b1*l1;
75 bt2 = b2*l2;
76 bt3 = b3*l3;
77 bt4 = 5688000;

```



```

27
28 bc(1)=4.1; //Mol/hr
29 bc(2)=49.3; //Mol/hr
30 bc(3)=71.9; //Mol/hr
31 bc(4)=52.5; //Mol/hr
32 bc(5)=54.7; //Mol/hr
33 bc(6)=82.5; //Mol/hr
34 bc(7)=76.6; //Mol/hr
35 bc(8)=22.4; //Mol/hr
36 tbc = 0;
37 i=1;
38 while(i<9)
39     tbc = tbc+bc(i);
40     i=i+1;
41 end
42
43 bK(1)=5.8; //at 330 F , 40 psia
44 bK(2)=3.0; //at 330 F , 40 psia
45 bK(3)=1.68; //at 330 F , 40 psia
46 bK(4)=0.98; //at 330 F , 40 psia
47 bK(5)=0.57; //at 330 F , 40 psia
48 bK(6)=0.35; //at 330 F , 40 psia
49 bK(7)=0.21; //at 330 F , 40 psia
50 bK(8)=0.13; //at 330 F , 40 psia
51
52 KL(1)=23.8;
53 KL(2)=148.0;
54 KL(3)=120.8;
55 KL(4)=51.4;
56 KL(5)=31.2;
57 KL(6)=28.9;
58 KL(7)=16.1;
59 KL(8)=2.9;
60 tk =0;
61 i=1;
62 while(i<9)
63     tk = tk + KL(i);
64     i=i+1;

```


Chapter 16

Extended Surfaces

Scilab code Exa 16.1 Calculation of the Fin Efficiency and a Weighted Efficiency Curve

```
1 printf("\t example 16.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 Af=(20*0.75*12*2)/(144);
4 Ao=((3.14*1.25)-(20*0.035))*(12/144);
5 printf("\t fin surface is : %.1f ft^2/lin ft \n",Af)
   ;
6 printf("\t bare tube surface is : %.3f ft^2/lin ft \
   n",Ao);
7 A=(Af+Ao);
8 printf("\t total outside surface : %.2f ft^2/lin ft
   \n",A);
9 Ai=(3.14*1.06*12)/(144);
10 printf("\t total inside surface : %.3f ft^2/lin ft \
   n",Ai);
11 printf("\t fin efficiencies \n");
12 b=0.0625; // ft
13 hf=4; // from table in solution
14 m=(5.24*(hf^(1/2))); // m=((hf*P)/(Kax))^(1/2), eq
   16.8
```

```

15 n=(tanh(m*b))/(m*b); // efficiency , eq 16.26
16 printf("\n hf      m      n \n "+string(hf)+"      "
    +string(m)+"      "+string(n)+" \n");
17 // similarly efficiencies values are calculated at
    different hf values
18 printf("\t weighted efficiency curve \n");
19 hfi=((n*Af)+(Ao))*(hf/Ai); // eq 16.34
20 printf("\n hf      hfi \n "+string(hf)+"      "+
    string(hfi)+" \n");
21 // similarly efficiencies values are calculated at
    different hf values
22 hf=[4 16 36 100 400 625 900]; // from 2nd table in
    the solution
23 hfi=[35.4 110.8 193.5 370 935 1295 1700]; // from 2
    nd table in the solution
24 plot2d("oll",hf,hfi);
25 xtitle("weighted fin efficiency curve","heat
    transfer coefficient to fin ,Btu/(ft^2)*(hr)","
    coefficient hf referred to the tube ID");
26 //end

```

Scilab code Exa 16.2 Calculation of a Heat transfer Curve from Experimental Data

```

1 printf("\t example 16.2 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 Ts=302; // F
4 t1=151;
5 t2=185;
6 w=15200; // lb/hr
7 // The dropwise condensation of steam was promoted
    with oil.
8 aa=(3.14*(3.068^2-1.25^2))/(4*144)-((20*0.035*0.75)
    /(144));

```

```

 9 printf("\t annulus flow area : %.4f ft^2 \n",aa);
10 p=(3.14*(1.25/12))-(20*0.035/12)+(20*0.75*2/12);
11 printf("\t wetted perimeter : %.2f ft \n",p);
12 De=(4*aa/p);
13 printf("\t equivalent diameter : %.3f ft \n",De);
14 Q=w*0.523*(t2-t1);
15 printf("\t heat load : %.2e Btu/hr \n",Q);
16 delT1=Ts-t1; //F
17 delT2=Ts-t2; // F
18 printf("\t delT1 is : %.0f F \n",delT1);
19 printf("\t delT2 is : %.0f F \n",delT2);
20 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
21 printf("\t LMTD is :%.0f F \n",LMTD);
22 Ai=0.277; // ft^2/ft
23 n=20; // number of fins
24 Ui=(Q/(Ai*n*LMTD));
25 printf("\t Ui : %.0f Btu/(hr)*(ft^2)*(F) \n",Ui);
26 hi=3000; // assumed value for dropwise condensation
    of steam
27 hfi=(Ui*hi)/(hi-Ui);
28 printf("\t hfi : %.0f Btu/(hr)*(ft^2)*(F) \n",hfi);
29 hf=120; // from fig 16.7 for hfi=418
30 mu=1.94; // lb/(ft*hr)
31 k=0.079;
32 Z=2.34; // Z=((c*mu)/k)^(1/3)
33 jf=(hf*De/(Z*k)); // eq 16.36
34 printf("\t jf : %.0f \n",jf);
35 Ga=(w/aa);
36 printf("\t Ga : %.2e lb/(hr)*(ft^2) \n",Ga);
37 Rea=(De*Ga/mu);
38 printf("\t Rea : %.2e \n",Rea);
39 // end

```

Scilab code Exa 16.3 Calculation of a Double Pipe Extended surface Gas Oil Cooler


```

1 printf("\t example 16.3 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=250; // inlet hot fluid ,F
4 T2=200; // outlet hot fluid ,F
5 t1=80; // inlet cold fluid ,F
6 t2=120; // outlet cold fluid ,F
7 W=18000; // lb/hr
8 w=11950; // lb/hr
9 printf("\t 1.for heat balance \n")
10 C=0.53; // Btu/(lb)*(F)
11 Q=((W)*(C)*(T1-T2)); // Btu/hr
12 printf("\t total heat required for gas oil is : %.2e
   Btu/hr \n",Q);
13 c=1; // Btu/(lb)*(F)
14 Q=((w)*(c)*(t2-t1)); // Btu/hr
15 printf("\t total heat required for water is : %.2e
   Btu/hr \n",Q);
16 delT1=T2-t1; //F
17 delT2=T1-t2; // F
18 printf("\t delT1 is : %.0f F \n",delT1);
19 printf("\t delT2 is : %.0f F \n",delT2);
20 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
21 printf("\t LMTD is :%.0f F \n",LMTD);
22 X=((delT1)/(delT2));
23 printf("\t ratio of two local temperature difference
   is : %.2f \n",X);
24 Fc=0.47; // from fig.17
25 Kc=0.27;
26 Tc=((T2)+((Fc)*(T1-T2))); // caloric temperature of
   hot fluid ,F
27 printf("\t caloric temperature of hot fluid is : %.0
   f F \n",Tc);
28 tc=((t1)+((Fc)*(t2-t1))); // caloric temperature of
   cold fluid ,F
29 printf("\t caloric temperature of cold fluid is : %
   .0f F \n",tc);
30 printf("\t hot fluid:shell side ,gas oil \n");

```

```

31 ID=3.068; // in , table 11
32 OD=1.9; // in , table 11
33 af=0.0175; // fin cross section ,table 10
34 aa=((3.14*ID^2/(4))-(3.14*OD^2/(4))-(24*af))/(144);
35 printf("\t flow area is : %.4f ft^2 \n",aa);
36 p=(3.14*(OD))-(24*0.035)+(24*0.5*2);
37 printf("\t wetted perimeter : %.2f in \n",p);
38 De=(4*aa*12/(p));
39 printf("\t De : %.4f ft \n",De);
40 Ga=(W/aa); // mass velocity ,lb/(hr)*(ft^2)
41 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Ga);
42 mu1=2.5*2.42; // at 224F,lb/(ft)*(hr), from fig.14
43 Rea=((De)*(Ga)/mu1); // reynolds number
44 printf("\t reynolds number is : %.2e \n",Rea);
45 jf=18.4; // from fig.16.10
46 Z=0.25; // Z=k*((c)*(mu1)/k)^(1/3), fig 16
47 Hf=((jf)*(1/De)*(Z)); // Hf=(hf/phyA), using eq.6.15,
    Btu/(hr)*(ft^2)*(F)
48 printf("\t individual heat transfer coefficient is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",Hf);
49 printf("\t cold fluid:inner tube side ,water \n");
50 D=0.134; // ft
51 row=62.5;
52 at=(3.14*D^2/(4));
53 printf("\t flow area is : %.4f ft^2 \n",at);
54 Gt=(w/(at)); // mass velocity ,lb/(hr)*(ft^2)
55 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
56 V=(Gt/(3600*row));
57 printf("\t V is : %.2f fps \n",V);
58 mu2=0.72*2.42; // at 99F,lb/(ft)*(hr)
59 Ret=((D)*(Gt)/mu2); // reynolds number
60 printf("\t reynolds number is : %.1e \n",Ret);
61 hi=(970*0.82); // fig 25
62 printf("\t hi : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
63 printf("\t calculation of tfw \n");
64 // Tc-tfw=40F assumption from fig 14

```

```

65 tfw=184;
66 mufw=3.5; // cp, at 184F
67 phya=(2.5/mufw)^0.14;
68 printf("\t phya is : %.2f \n",phya); // from fig.24
69 hf=(Hf)*(phya); // from eq.6.36
70 printf("\t Correct hf to the surface at the OD is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",hf);
71 Rdo=0.002;
72 Rf=(1/hf);
73 printf("\t Rf : %.4f \n",Rf);
74 hf1=(1/(Rdo+Rf)); // eq 16.37
75 printf("\t hf1 : %.1f \n",hf1);
76 hfi1=255; // fig 16.9
77 hfi2=(hf1*5.76); // eq 16.38 and fig 16.9,((Af+Ao)/(
    Ai))=5.76 from previous prblm
78 printf("\t hfi2 : %.0f \n",hfi2);
79 Rmetal=(hfi2-hfi1)/(hfi2*hfi1); // eq 16.39
80 printf("\t Rmetal : %.5f \n",Rmetal);
81 phyt=1; // for cooling water
82 Rdi=0.003;
83 Ri=(1/hi);
84 printf("\t Ri : %.5f \n",Ri);
85 hi1=(1/(Rdi+Ri)); // eq 16.40
86 printf("\t hi1 : %.1f \n",hi1);
87 UDi=(hi1*hfi1)/(hi1+hfi1); // eq 16.41
88 printf("\t UDi : %.0f \n",UDi);
89 // To obtain the true flux the heat load must be
    divided by the actual heat-transfer surface.For a
    1}2-in. IPS pipe there are 0.422 ft2/lin foot ,
    from table 11
90 // trial
91 Ai=(Q/(UDi*LMTD)); // LMTD=delt
92 printf("\t Ai : %.1f ft^2 \n",Ai);
93 L=(Ai/0.422);
94 printf("\t length of pipe required : %.1f lin ft \n"
    ,L);
95 // Use two 20-ft hairpins = 80 lin ft
96 Ai1=(80*0.422); // ft^2

```

```

97 r=(Q/Ai1);
98 printf("\t Q/Ai1 : %.2e Btu/(hr)*(ft ^2) \n",r);
99 deltf=(r/hfi2);
100 deltdo=(r*Rdo/5.76);
101 printf("\t annulus film : %.1f \n",deltf);
102 printf("\t annulus dirt : %.1f \n",deltdo);
103 d=deltf+deltdo; // d=Tc-tfw
104 deltmetal=(r*Rmetal);
105 deltdi=(r*Rdi);
106 delti=(r/hi);
107 printf("\t Tc-tfw : %.1f \n",d);
108 printf("\t fin and tube metal : %.1f \n",deltmetal);
109 printf("\t tube side dirt : %.1f \n",deltdi);
110 printf("\t tubeside film : %.1f \n",delti);
111 Td=deltf+deltdo+deltmetal+deltdi+delti;
112 printf("\t total temperature drop : %.1f F \n",Td);
113 printf("\t pressure drop for annulus \n");
114 De1=0.0359; // ft
115 Rea1=(De1*Ga/mu1);
116 printf("\t reynolds number : %.2e \n",Rea1);
117 f=0.00036; // fig 16.10
118 s=0.82; //using fig.6
119 delPs=((f*(Ga^2)*(80))/(5.22*(10^10)*(De1)*(s)*(phya
    ))); // using eq.7.44, psi
120 printf("\t delPs is : %.1f psi \n",delPs);
121 printf("\t allowable delPa is 10 psi \n");
122 printf("\t pressure drop for inner pipe \n");
123 f=0.000192; // friction factor for reynolds number
    65000, using fig.26
124 s=1;
125 delPt=((f*(Gt^2)*(80))/(5.22*(10^10)*(0.134)*(s)*(1)
    )); // using eq.7.45, psi
126 printf("\t delPt is : %.1f psi \n",delPt);
127 printf("\t allowable delPa is 10 psi \n");
128 //end

```

Scilab code Exa 16.4 Calculation of a Longitudinal Fin Shell and tube Exchanger

```
1 printf("\t example 16.4 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=250; // inlet hot fluid ,F
4 T2=100; // outlet hot fluid ,F
5 t1=80; // inlet cold fluid ,F
6 t2=100; // outlet cold fluid ,F
7 W=30000; // lb/hr
8 w=50500; // lb/hr
9 printf("\t 1.for heat balance \n")
10 C=0.225; // Btu/(lb)*(F)
11 Q=((W)*(C)*(T1-T2)); // Btu/hr
12 printf("\t total heat required for oxygwn is : %.2e
   Btu/hr \n",Q);
13 c=1; // Btu/(lb)*(F)
14 Q=((w)*(c)*(t2-t1)); // Btu/hr
15 printf("\t total heat required for water is : %.2e
   Btu/hr \n",Q);
16 delT1=T2-t1; //F
17 delT2=T1-t2; // F
18 printf("\t delT1 is : %.0f F \n",delT1);
19 printf("\t delT2 is : %.0f F \n",delT2);
20 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
21 printf("\t LMTD is :%.1f F \n",LMTD);
22 R=((T1-T2)/(t2-t1));
23 printf("\t R is : %.1f \n",R);
24 S=((t2-t1)/(T1-t1));
25 printf("\t S is : %.4f \n",S);
26 printf("\t FT is 0.87 \n"); // from fig 18
27 delT=(0.87*LMTD); // F
28 printf("\t delT is : %.1f F \n",delT);
```

```

29 Tc=(T2+T1)/(2); // caloric temperature of hot fluid ,
    F
30 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
31 tc=((t1)+(t2))/(2); // caloric temperature of cold
    fluid ,F
32 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
33 printf("\t hot fluid:shell side ,oxygen \n");
34 ID=19.25; // in , table 11
35 OD=1; // in , table 11
36 as=((3.14*ID^2/(4))-(70*3.14*OD^2/(4))
    -(70*20*0.035*0.5))/(144);
37 printf("\t flow area is : %.2f ft^2 \n",as);
38 p=(70*3.14*(OD))-(70*20*0.035)+(70*20*0.5*2);
39 printf("\t wetted perimeter : %.2e in \n",p);
40 De=(4*as*12/(p));
41 printf("\t De : %.3f ft \n",De);
42 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2)
43 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
44 mu1=0.0545; // at 175F,lb/(ft)*(hr) , from fig.15
45 Res=((De)*(Gs)/mu1); // reynolds number
46 printf("\t reynolds number is : %.3e \n",Res);
47 jH=59.5; // from fig.16.10a
48 k=0.0175;
49 Z=0.89; // Z=((c)*(mu1)/k)^(1/3) , fig
50 hf=((jH)*(k/De)*(Z)); //using eq.6.15 ,Btu/(hr)*(ft
    ^2)*(F)
51 printf("\t individual heat transfer coefficient is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",hf);
52 Rdo=0.003;
53 hdo=(1/Rdo);
54 hf1=(hdo*hf)/(hdo+hf); // eq 16.37
55 printf("\t hf1 : %.1f \n",hf1);
56 hfi1=142; // fig 16.9
57 printf("\t cold fluid:inner tube side ,water \n");
58 at1=0.479; // table 10

```

```

59 L=16;
60 Nt=70;
61 n=4;
62 at=((Nt*at1)/(144*n)); // total area, ft^2, from eq
    .7.48
63 printf("\t flow area is : %.4f ft^2 \n",at);
64 D=0.0652; // ft
65 row=62.5;
66 Gt=(w/(at)); // mass velocity, lb/(hr)*(ft^2)
67 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
68 V=(Gt/(3600*row));
69 printf("\t V is : %.2f fps \n",V);
70 mu2=1.94; // at 90F, lb/(ft)*(hr)
71 Ret=((D)*(Gt)/mu2); // reynolds number
72 printf("\t reynolds number is : %.2e \n",Ret);
73 hi=(940*0.96); // fig 25
74 printf("\t hi : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
75 Rdi=0.003;
76 hdi=(1/Rdi);
77 hi1=(hdi*hi)/(hdi+hi);
78 printf("\t hi1 : %.0f Btu/(hr)*(ft^2)*(F) \n",hi1);
79 UDi=((hfi1)*(hi1)/(hi1+hfi1)); // eq 16.41, Btu/(hr)
    *(ft^2)*(F)
80 printf("\t overall coefficient is : %.1f Btu/(hr)*(
    ft^2)*(F) \n",UDi);
81 A2=0.2048; // actual surface supplied for each tube,
    ft^2, from table 10
82 A=(Nt*L*A2); // ft^2
83 printf("\t total surface area is : %.0f ft^2 \n",A);
84 UDi1=((Q)/((A)*(delt)));
85 printf("\t design overall coefficient is : %.1f Btu
    /(hr)*(ft^2)*(F) \n",UDi1);
86 Re=(1/UDi1)-(1/UDi);
87 printf("\t excess fouling factor : %.5f \n",Re);
88 Ro=9.27; //Adding to the outside fouling factor
89 Rdo1=Rdo+(Re*Ro);
90 printf("\t Rdo : %.4f \n",Rdo1);

```

```

91 hf2=(hf/(1+(hf*Rdo1)));
92 printf("\t hf2 : %.1f \n",hf2);
93 hfi2=113;
94 UDi2=((hfi2)*(hi1)/(hi1+hfi2)); // eq 16.41 ,Btu/(hr)
    *(ft^2)*(F)
95 printf("\t overall coefficient is : %.1f Btu/(hr)*(
    ft^2)*(F) \n",UDi2);
96 printf("\t pressure drop for annulus \n");
97 De1=0.0433; // ft
98 Res1=(De1*Gs/mu1);
99 printf("\t reynolds number : %.2e \n",Res1);
100 f=0.00025; // fig 16.10
101 s=0.00133;
102 delPs=((f*(Gs^2)*(L))/(5.22*(10^10)*(De1)*(s)*(1)));
    // using eq.7.44 ,psi
103 printf("\t delPs is : %.1f psi \n",delPs);
104 printf("\t allowable delPa is 2 psi \n");
105 printf("\t pressure drop for inner pipe \n");
106 f=0.00021; // friction factor for reynolds number
    29100, using fig.26
107 s=1;
108 delPt=((f*(Gt^2)*(L)*(n))/(5.22*(10^10)*(0.0625)*(s)
    *(1))); // using eq.7.45 ,psi
109 printf("\t delPt is : %.0f psi \n",delPt);
110 printf("\t allowable delPa is 10 psi \n");
111 //end

```

Scilab code Exa 16.5 Calculation of a Transverse fin Air Cooler

```

1 printf("\t example 16.5 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=250; // inlet hot fluid ,F
4 T2=200; // outlet hot fluid ,F
5 t1=150; // inlet cold fluid ,F

```



```

6  t2=190; // outlet cold fluid ,F
7  W=100000; // lb/hr
8  w=31200; // lb/hr
9  printf("\t 1.for heat balance \n")
10 C=0.25; // Btu/(lb)*(F)
11 Q=((W)*(C)*(T1-T2)); // Btu/hr
12 printf("\t total heat required for air is : %.2e Btu
    /hr \n",Q);
13 c=1; // Btu/(lb)*(F)
14 Q=((w)*(c)*(t2-t1)); // Btu/hr
15 printf("\t total heat required for water is : %.2e
    Btu/hr \n",Q);
16 delT1=T2-t1; //F
17 delT2=T1-t2; // F
18 printf("\t delT1 is : %.0f F \n",delT1);
19 printf("\t delT2 is : %.0f F \n",delT2);
20 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
21 printf("\t LMTD is :%.1f F \n",LMTD);
22 R=((T1-T2)/(t2-t1));
23 printf("\t R is : %.1f \n",R);
24 S=((t2-t1)/(T1-t1));
25 printf("\t S is : %.4f \n",S);
26 printf("\t FT is 0.985 \n"); // from fig 18
27 delT=(0.985*LMTD); // F
28 printf("\t delT is : %.1f F \n",delT);
29 Tc=(T2+T1)/(2); // caloric temperature of hot fluid ,
    F
30 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
31 tc=((t1)+(t2))/(2); // caloric temperature of cold
    fluid ,F
32 printf("\t caloric temperature of cold fluid is : %
    .0f F \n",tc);
33 Af=(3.14*2*8*12*(1.75^2-1^2))/(4);
34 Ao=((3.14*1*12) - (3.14*1*8*0.035*12));
35 printf("\t fin surface is : %.0f in^2/lin ft \n",Af)
    ;
36 printf("\t bare tube surface is : %.1f in^2/lin ft \

```

```

    n",Ao);
37 A=(Af+Ao);
38 printf("\t total outside surface : %.1f ft^2/lin ft
    \n",A);
39 p=(2*3*2*8*12/8)+(((12)-(8*0.035*12))*(2));
40 printf("\t projected perimeter : %.1f in/ft \n",p);
41 De=(2*A/(3.14*p*12)); // eq 16.104
42 printf("\t De : %.3f ft \n",De);
43 // 21 tubes may be fit in one :vertical bank (Fig.
    16.19b) ,20 tubes in alternating banks for
    triangular pitch
44 as=((4^2*12^2)-(21*1*48)-((21)*(2*0.035*3*8*48/8)))
    /(144); // fig 16.19
45 printf("\t flow area : %.1f ft^2 \n",as);
46 printf("\t hot fluid:shell side ,oxygen \n");
47 Gs=(W/as); // mass velocity ,lb/(hr)*(ft^2)
48 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gs);
49 mu1=0.052; // at 225F,lb/(ft)*(hr), from fig.15
50 Res=((De)*(Gs)/mu1); // reynolds number
51 printf("\t reynolds number is : %.2e \n",Res);
52 jf=157; // from fig.16.18a
53 k=0.0183;
54 Z=0.89; // Z=((c)*(mu1)/k)^(1/3), fig
55 phys=1;
56 hf=((jf)*(k/De)*(Z)); //using eq.6.15 ,Btu/(hr)*(ft
    ^2)*(F)
57 printf("\t individual heat transfer coefficient is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",hf);
58 Rdo=0.003;
59 hdo=(1/Rdo);
60 hf1=(hdo*hf)/(hdo+hf); // eq 16.37
61 printf("\t hf1 : %.1f \n",hf1);
62 hfi1=142; // fig 16.9
63 printf("\t cold fluid:inner tube side ,water \n");
64 at1=0.546; // table 10
65 L=4;
66 Nt=21;

```

```

67 n=1;
68 at=((Nt*at1)/(144*n)); // total area, ft^2, from eq
    .7.48
69 printf("\t flow area is : %.4f ft^2 \n",at);
70 D=0.0695; // ft
71 row=62.5;
72 Gt=(w/(at)); // mass velocity, lb/(hr)*(ft^2)
73 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
74 V=(Gt/(3600*row));
75 printf("\t V is : %.2f fps \n",V);
76 mu2=0.895; // at 170F, lb/(ft)*(hr)
77 Ret=((D)*(Gt)/mu2); // reynolds number
78 printf("\t reynolds number is : %.2e \n",Ret);
79 hi=(710*0.94); // fig 25
80 printf("\t hi : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
81 Rdi=0.003;
82 hdi=(1/Rdi);
83 hi1=(hdi*hi)/(hdi+hi); // 16.40
84 printf("\t hi1 : %.0f Btu/(hr)*(ft^2)*(F) \n",hi1);
85 k1=60; // table 3 , for brass
86 // yb=0.00146 ft
87 X=((0.875-0.5)/12)*(21.5/(60*0.00146))^(1/2);
88 printf("\t X :%.2f \n",X);
89 nf=0.91; // from fig 16.13a , by comparing X value
90 Ai=0.218; // ft^2/ft
91 hfi2=((nf*Af/144)+(Ao/144))*(hf1/Ai); // eq 16.34
92 printf("\t hfi2 : %.0f \n",hfi2);
93 UDi=((hfi2)*(hi1)/(hi1+hfi2)); // eq 16.41, Btu/(hr)
    *(ft^2)*(F)
94 printf("\t overall coefficient is : %.0f Btu/(hr)*(
    ft^2)*(F) \n",UDi);
95 A=(21*4*Ai); // ft^2
96 printf("\t inside surface per bank is : %.1f ft^2 \n
    ",A);
97 Ai1=(Q/(UDi*delt));
98 printf("\t Ai1 : %.0f ft^2 \n",Ai1);
99 Nb=(Ai1/A);

```

```

100 printf("\t number of banks : %.0f \n",Nb);
101 Vn=(4*4*1.95/12) - (41*3.14*1*4/(2*4*144))
    - ((41*3.14*0.035*8*4/(144*2*4))*(1.75^2-1^2)); //
    fig 16.19b
102 printf("\t net free volume : %.2f ft^3 \n",Vn);
103 Af1=(41*2.34*4/2);
104 printf("\t frictional surface : %.0f ft^2 \n",Af1);
105 printf("\t pressure drop for annulus \n");
106 De1=(4*Vn/Af1); // ft
107 printf("\t De1 : %.2f ft \n",De1);
108 Res1=(De1*Gs/mu1);
109 printf("\t reynolds number : %.2e \n",Res1);
110 f=0.0024; // fig 16.18b
111 s=0.000928;
112 Lp=1.95;
113 R1=0.538; // R1=(De1/ST)^(0.4)
114 R2=1; // R2=(SL/ST)^0.6
115 delPs=((f*(Gs^2)*(Lp)*(R1)*(R2))/(5.22*(10^10)*(De1)
    *(s)*(1)));
116 printf("\t delPs is : %.2f psi \n",delPs);
117 printf("\t pressure drop for inner pipe \n");
118 f=0.0002; // friction factor for reynolds number
    30400, using fig.26
119 s=1;
120 delPt=((f*(Gt^2)*(L)*(Nb))/(5.22*(10^10)*(0.0695)*(s)
    *(1))); // using eq.7.45, psi
121 printf("\t delPt is : %.2f psi \n",delPt);
122 //end

```

Chapter 17

Cooling Towers

Scilab code Exa 17.1 Calculation of the Enthalpy of Saturated Air

```
1 printf("\t example 17.1 \n");
2 pw=0.4298; // psia , at 75F, table 7
3 pt=14.696; // psia
4 t=75;
5 Mw=18;
6 Ma=29;
7 X=(pw/(pt-pw))*(Mw/Ma);
8 printf("\t humidity is : %.4f lb water/lb air \n",X)
   ;
9 H=(X*t)+(1051.5*X)+(0.24*t); // eq 17.54
10 printf("\t enthalpy at 75F is : %.1f Btu/lb dry air
   \n",H);
11 // end
```

Scilab code Exa 17.2 Calculation of the Number of Diffusion Units

```
1 printf("\t example 17.2 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
```

```

3 printf("\t by numerical integration \n");
4 T1=85;
5 T2=120;
6 A=576; // ground area, from fig 17.12
7 L=1500*(500/576);
8 G=1400;
9 R=(L/G);
10 printf("\t R is : %.2f \n",R);
11 H1=39.1; // fig 17.12
12 H2=H1+(R*(T2-T1));
13 printf("\t H2 is : %.1f Btu \n",H2);
14 // The area between the saturation line and the
    operating line represents the potential for heat
    transfer
15 // at T=85F
16 Hs=50; // fig 17.12
17 d1=(Hs-H1);
18 printf("\t difference is : %.1f \n",d1);
19 //at t=90
20 Hs=56.7; // fig 17.12
21 H=43.7; // fig 17.12
22 d2=Hs-H;
23 printf("\t difference is : %.1f \n",d2);
24 d=(d1+d2)/(2);
25 printf("\t average of difference is : %.1f \n",d);
26 dT=5; // F
27 nd1=(dT/d);
28 printf("\t nd1 is : %.3f \n",nd1);
29 // similarly calculating nd at each temperature and
    adding them will give you total nd value
30 nd=1.70;
31 printf("\t number of diffusing units : %.2f \n",nd);
32 printf("\t log mean enthalpy difference \n");
33 dt=49.9; // diff. of enthalpies at top of the tower,
    from table in solution
34 db=10.9; // diff of enthalpies at bottom of the
    tower,from table in solution
35 LME=(dt-db)/(2.3*log10(dt/db));

```

```

36 printf("\t log mean of enthalpy : %.1f Btu/lb \n",
    LME);
37 nd=(T2-T1)/(LME);
38 printf("\t number of diffusing units are : %.2f \n",
    nd);
39 // The error is naturally larger the greater the
    range
40 //end

```

Scilab code Exa 17.3 Calculation of the Required Height of Fill

```

1 printf("\t example 17.3 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 // Since the loading is based on 1 ft2 of ground
    area
4 nd=1.7;
5 L=1302;
6 Kxa=115;
7 Z=(nd*L)/(Kxa);
8 printf("\t Z is : %.1f ft \n",Z);
9 HDU=(Z/nd);
10 printf("\t height of diffusion unit : %.1ff ft \n",
    HDU);
11 // end

```

Scilab code Exa 17.4 Determination of a Cooling tower Guarantee

```

1 printf("\t example 17.4 \n");
2 printf("\t approximate values are mentioned in the
    book \n");

```

```

3 // The area between the saturation line and the
   operating line represents the potential for heat
   transfer
4 // at T=79.3F
5 Hs=43.4; // fig 17.12
6 H=30.4; // fig 17.12
7 d1=(Hs-H);
8 printf("\t difference is : %.1f \n",d1);
9 //at t=85
10 Hs=50; // fig 17.12
11 H=35.7; // fig 17.12
12 d2=Hs-H;
13 printf("\t difference is : %.1f \n",d2);
14 d=(d1+d2)/(2);
15 printf("\t average of difference is : %.2f \n",d);
16 dT=(85-79.3); // F
17 nd1=(dT/d);
18 printf("\t nd1 is : %.3f \n",nd1);
19 // similarly calculating nd at each temperature and
   adding them will give you total nd value
20 nd=1.72;
21 printf("\t number of diffusing units : %.2f \n",nd);
22 // end

```

Scilab code Exa 17.5 The Recalculation of Cooling tower Performance

```

1 printf("\t example 17.5 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=85;
4 T2=120;
5 R=0.93; // R=(L/G), for 1500 gpm
6 printf("\t for 120 percent of design \n");
7 R1=1.2*R;
8 printf("\t R is : %.3f \n",R1);

```



```

 9 H1=39.1; // at 87.2F
10 H2=H1+(R1*(T2-T1));
11 printf("\t H2 is : %.1f Btu \n",H2);
12 // The area between the saturation line and the
    operating line represents the potential for heat
    transfer
13 // at T=87.2F
14 Hs=53.1; // from table in the solution
15 d1=(Hs-H1);
16 printf("\t difference is : %.1f \n",d1);
17 //at t=90
18 Hs=56.7; // fig 17.12
19 H=42; // fig 17.12
20 d2=Hs-H;
21 printf("\t difference is : %.1f \n",d2);
22 d=(d1+d2)/(2);
23 printf("\t average of difference is : %.1f \n",d);
24 dT=(90-87.2); // F
25 nd1=(dT/d);
26 printf("\t nd1 is : %.3f \n",nd1);
27 // similarly calculating nd at each temperature and
    adding them will give you total nd value
28 nd=1.53;
29 printf("\t number of diffusing units : %.2f \n",nd);
30 printf("\t for 80 percent of design \n");
31 R2=0.8*R;
32 printf("\t R is : %.3f \n",R2);
33 H1=39.1; // at 87.2F
34 H2=H1+(R2*(T2-T1));
35 printf("\t H2 is : %.0f Btu \n",H2);
36 // The area between the saturation line and the
    operating line represents the potential for heat
    transfer
37 // at T=82.5F
38 Hs=47.2; // from table in the solution
39 d1=(Hs-H1);
40 printf("\t difference is : %.1f \n",d1);
41 //at t=85

```

```

42 Hs=50; // fig 17.12
43 H=40.8; // fig 17.12
44 d2=Hs-H;
45 printf("\t difference is : %.1f \n",d2);
46 d=(d1+d2)/(2);
47 printf("\t average of difference is : %.1f \n",d);
48 dT=(85-82.5); // F
49 nd1=(dT/d);
50 printf("\t nd1 is : %.3f \n",nd1);
51 // similarly calculating nd at each temperature and
    adding them will give you total nd value
52 nd=1.92;
53 printf("\t number of diffusing units : %.2f \n",nd);
54 X=[1.115 0.93 0.74];
55 Y=[1.53 1.70 1.92];
56 plot2d(X,Y,style=3,rect=[0.7,1.4,1.3,2]);
57 xtitle("KxaV/L vs L/G","L/G","nd");
58 printf("\t trial 1 \n");
59 R3=1.1;
60 printf("\t R is : %.3f \n",R3);
61 H1=34.5; // at 87.2F
62 H2=H1+(R3*(T2-T1));
63 printf("\t H2 is : %.0f Btu \n",H2);
64 // The area between the saturation line and the
    operating line represents the potential for heat
    transfer
65 // at T=85F
66 Hs=50; // from table in the solution
67 d1=(Hs-H1);
68 printf("\t difference is : %.1f \n",d1);
69 //at t=90
70 Hs=56.7; // fig 17.12
71 H=40; // fig 17.12
72 d2=Hs-H;
73 printf("\t difference is : %.1f \n",d2);
74 d=(d1+d2)/(2);
75 printf("\t average of difference is : %.1f \n",d);
76 dT=(90-85); // F

```

```

77 nd1=(dT/d);
78 printf("\t nd1 is : %.3f \n",nd1);
79 // similarly calculating nd at each temperature and
    adding them will give you total nd value
80 nd=1.48;
81 printf("\t number of diffusing units : %.2f \n",nd);
82 R3=1.19; // from fig 17.14
83 printf("\t L/G is : %.2f \n",R3);
84 printf("\t trial 2 \n");
85 R4=1.2;
86 printf("\t R4 is : %.3f \n",R4);
87 H1=34.5; // at 87.2F
88 H2=H1+(R4*(T2-T1));
89 printf("\t H2 is : %.1f Btu \n",H2);
90 // The area between the saturation line and the
    operating line represents the potential for heat
    transfer
91 // at T=85F
92 Hs=50; // from table in the solution
93 d1=(Hs-H1);
94 printf("\t difference is : %.1f \n",d1);
95 //at t=90
96 Hs=56.7; // fig 17.12
97 H=40.5; // fig 17.12
98 d2=Hs-H;
99 printf("\t difference is : %.1f \n",d2);
100 d=(d1+d2)/(2);
101 printf("\t average of difference is : %.1f \n",d);
102 dT=(90-85); // F
103 nd1=(dT/d);
104 printf("\t nd1 is : %.3f \n",nd1);
105 // similarly calculating nd at each temperature and
    adding them will give you total nd value
106 nd=1.56;
107 printf("\t number of diffusing units : %.2f \n",nd);
108 R3=1.08; // from fig 17.14
109 printf("\t L/G is : %.2f \n",R3);
110 // end

```

Scilab code Exa 17.6 Calculation of a Direct contact Gas Cooler

```
1 printf("\t example 17.6 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 // basis 1ft^2 ground area
4 //Assumption: 20 per cent of the initial vapor
   content of the gas enters the water body
5 X1=(1.69/(14.7-1.69))*(18/29);
6 printf("\t X1 : %.4f lb/lb \n",X1);
7 G=1500;
8 w1=G*X1;
9 printf("\t total water in inlet gas : %.2f lb/hr \n"
   ,w1);
10 // The inlet gas is at 300F and a 120F dew point.
   Use 0.25 Btu/(lb)( F) for the specific heat of
   nitrogen
11 H1=(0.0807*120)+(0.0807*1025.8)
   +(0.45*0.0807*(300-120))+(0.25*300); // eq 17.55
12 printf("\t H1 : %.0f Btu/lb dry air \n",H1);
13 X2=(w1*(1-.2)/G);
14 printf("\t outlet gas humidity : %.5f lb/lb \n",X2);
15 pw=(X2*29*14.7/18)/(1+(X2*29/18));
16 printf("\t pw : %.3f psia \n",pw);
17 Tw=112.9; // F, from table 7 for above pw
18 // The outlet gas has a temperature of 200 F and a
   112.9 F dew point
19 H2=(X2*Tw)+(X2*1029.8)+(X2*0.45*(200-Tw))+(0.25*200)
   ; // eq 17.55
20 printf("\t H2 : %.1f Btu/lb dry air \n",H2);
21 q=G*(H1-H2);
22 printf("\t total heat load : %.2e Btu/hr \n",q);
23 w2=q/(120-85);
24 printf("\t water loading : %.2e lb/hr \n",w2);
```

```

25 printf("\t interval 1 \n");
26 // (Kxa*delV/L)= 0 to 0.05
27 nd=0.05; // nd=Kxa*V/L
28 Le=0.93; // fig 17.4 at 300F
29 C=(0.25)+(0.45*X1);
30 printf("\t C : %.3f Btu/(lb)*(F) \n",C);
31 haV=(nd*w2*Le*C);
32 printf("\t haV : %.1f Btu/(hr)*(F) \n",haV);
33 qc=(haV*(300-120));
34 printf("\t qc : %.2e Btu/hr \n",qc);
35 delT=(qc/(C*G));
36 printf("\t delT : %.1f F \n",delT);
37 T1=(300-delT);
38 printf("\t T(0.05) : %.1f F \n",T1);
39 delT=(qc/w2);
40 printf("\t delT : %.2f F \n",delT);
41 t1=(120-delT);
42 printf("\t t(0.05) : %.1f F \n",t1);
43 printf("\t interval 2 \n");
44 // (Kxa*delV/L)= 0.05 to 0.15
45 nd1=0.1;
46 haV1=(nd1*w2*Le*C);
47 printf("\t haV1 : %.1f Btu/(hr)*(F) \n",haV1);
48 qc1=(haV1*(T1-t1));
49 printf("\t qc1 : %.1e Btu/hr \n",qc1);
50 delT1=(qc1/(C*G));
51 printf("\t delT1 : %.1f F \n",delT1);
52 T2=(T1-delT1);
53 printf("\t T(0.15) : %.2f F \n",T2);
54 X3=0.0748; // at 117.6F
55 w3=(nd1*w2*(0.0807-X3));
56 printf("\t water diffused during interval : %.3f lb/
    hr \n",w3);
57 w4=(w1-w3);
58 printf("\t water remaining : %.2f lb/hr \n",w4);
59 l1=1027; // Btu/lb, l1= lamda at 117.6F
60 qd=(w3*l1);
61 printf("\t qd : %.0f Btu/hr \n",qd);

```

```

62 q1=(qd+qc1);
63 printf("\t q1 : %.0f Btu/hr \n",q1);
64 delT1=(q1/w2);
65 printf("\t delT1 : %.2f F \n",delT1);
66 t2=(t1-delT1);
67 printf("\t t(0.15) : %.1f F \n",t2);
68 X4=0.0640; // at 112.5
69 X5=(w4/G);
70 printf("\t X(112.5F) : %.4f lb/lb \n",X5);
71 printf("\t interval 3 \n");
72 // (Kxa*delV/L)= 0.15 to 0.25
73 nd1=0.1;
74 haV1=(nd1*w2*Le*C);
75 printf("\t haV1 : %.1f Btu/(hr)*(F) \n",haV1);
76 qc2=(haV1*(T2-t2));
77 printf("\t qc2 : %.2e Btu/hr \n",qc2);
78 delT2=(qc2/(C*G));
79 printf("\t delT2 : %.1f F \n",delT2);
80 T3=(T2-delT2);
81 printf("\t T(0.25) : %.1f F \n",T3);
82 w5=(nd1*w2*(X5-X4));
83 printf("\t water diffused during interval : %.3f lb/
      hr \n",w5);
84 w6=(w4-w5);
85 printf("\t water remaining : %.2f lb/hr \n",w6);
86 l2=1030; // Btu/lb, l1= lamda at 112.5F
87 qd1=(w5*l2);
88 printf("\t qd1 : %.2e Btu/hr \n",qd1);
89 q2=(qd1+qc2);
90 printf("\t q2 : %.3e Btu/hr \n",q2);
91 delT2=(q2/w2);
92 printf("\t delT2 : %.2f F \n",delT2);
93 t3=(t2-delT2);
94 printf("\t t(0.25) : %.1f F \n",t3);
95 X6=0.0533; // at 106.5
96 X7=(w6/G);
97 printf("\t X(106.5F) : %.4f lb/lb \n",X7);
98 // The calculations of the remaining intervals until

```

```

    a. gas temperature of 200 F is reached are
    shown in Fig. 17.17
99 w7=21.92; // total water diffused from table in
    solution
100 d=(w7/w1)*100;
101 printf("\t calculated diffusion : %.0f \n",d);
102 printf("\t Using some standard low-pressure-drop
    data \n");
103 // For G = 1500, extrapolate to L = 2040 on
    logarithmic coordinates. Kxa = 510.
104 ndt=.54; // from 1st table in solution
105 Kxa=510; // from 2nd table in solution
106 Z=(ndt*w2/Kxa);
107 printf("\t tower height : %.2f ft \n",Z);
108 A=(50000/G);
109 printf("\t cross section : %.1f ft^2 \n",A);
110 // end

```

Scilab code Exa 17.7 Approximate Calculation of a Gas Cooler

```

1 printf("\t example 17.7 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 C=0.28; // assumption
4 w=50000; // lb/hr
5 G=1500;
6 Qs=(w*C*(500-200));
7 Qd=(w/G)*(22685); // qd=22685, from previous prblm
8 printf("\t sensible heat : %.1e Btu/hr \n",Qs);
9 printf("\t approximate diffusion : %.2e Btu/hr \n",
    Qd);
10 Q=(Qs+Qd);
11 printf("\t total heat : %.3e Btu/hr \n",Q);
12 // an allowance as high as 30 per cent of the
    sensible load can be made and the excess water

```

```

        compensated for by throttling when the tower is
        in operation
13 w1=(Q/(120-85));
14 printf("\t total water quantity : %.2e lb/hr \n",w1)
    ;
15 // If the maximum liquid loading is taken as 2040 lb
    /(hr)(ft '!), the required tower cross section
16 A=(w1/2040);
17 printf("\t tower cross section : %.1f ft^2 \n",A);
18 w3=(w/A);
19 printf("\t new gas rate : %.0f lb/(hr)(ft^2) \n",w3)
    ;
20 // The two terminal temperature differences are (200
    - 85) and (500 - 120).
21 LMTD=((500-120)-(200-85))/(log((500-120)/(200-85)));
22 printf("\t LMTD : %.0f \n",LMTD);
23 dt=35;
24 N=(dt/LMTD); // eq 17.88
25 printf("\t haV/L : %.2f \n",N);
26 Le=0.93;
27 nd=(N/(C*Le));
28 printf("\t number diffusion units : %.2f \n",nd);
29 // By extrapolation for G = 718 and L = 2040,Kxa=215
30 L=2040;
31 Kxa=215;
32 Z=(nd*L/Kxa); // calculation mistake
33 printf("\t height of tower : %.1f ft \n",Z);
34 di=(A)^(1/2);
35 printf(" ground dimensions : %.1f ft \n",di);
36 // ground dimensions are 5.8*8.3*8.3 ft
37 // end

```

Chapter 18

Batch and unsteady process

Scilab code Exa 18.1 Calculation of Batch Heating

```
1 printf("\t example 18.1 \n");
2 // specific gravity of benzene is 0.88
3 // specific heat of benzene is 0.48 Btu/(lb)*(F)
4 U=50;
5 A=400;
6 T1=400;
7 t1=100;
8 t2=300;
9 c=0.48;
10 w=40000;
11 C=0.60;
12 W=10000;
13 printf("\t values are approximately mentioned in the
        book \n");
14 printf("\t for a \n");
15 M=(7500*8.33*0.88);
16 printf("\t weight of benzene is : %.1e lb \n",M);
17 Q1=(w*c);
18 printf("\t Q1 is : %.2e Btu/(hr)*(F) \n",Q1);
19 Q2=(W*C);
20 printf("\t Q2 is : %.0e Btu/(hr)*(F) \n",Q2);
```

```

21 Ks=((%e)^(U*A*((1/Q1)-(1/Q2)))); // eq 18.16
22 printf("\t Ks is : %.3f \n",Ks);
23 Z=log((T1-t1)/(T1-t2));
24 printf("\t Z is : %.3f \n",Z);
25 theta=((M*(Z)*(Ks*6000-(19200)))/((Ks-1)*40000*6000)
);
26 printf("\t theta is : %.1f hr \n",theta);
27 printf("\t for b \n");
28 R=(Q1/Q2);
29 printf("\t R is : %.1f \n",R);
30 KT=((%e)^(U*(A/Q1)*(1+R^2)^(1/2)));
31 printf("\t KT is : %.0f \n",KT);
32 S=((2*(KT-1))/((KT*(R+1+(1+R^2)^(1/2)))-(R+1-(1+R^2)
^(1/2)))); // eq 18.24
33 printf("\t S is : %.3f \n",S);
34 theta1=((M*Z)/(0.266*40000)); // eq 18.25
35 printf("\t theta1 is : %.2f hr \n",theta1);
36 printf("\t for c \n");
37 U1=100;
38 A1=200;
39 K8=((%e)^(U*(A/(2*Q1))*(1+R^2)^(1/2))); // eq 18.32
40 S1=((2*(K8-1)*(1+((1-0.266)*(1-(3.2*0.266)))^(1/2)))
/(((K8-1)*(3.2+1))+((K8+1)*(1+3.2^2)^(1/2)))); //
eq 18.31
41 printf("\t K8 is : %.2f \n",K8);
42 printf("\t S1 is : %.3f \n",S1);
43 theta2=((M*Z)/(0.282*40000)); // eq 18.25
44 printf("\t theta2 is : %.2f hr \n",theta2);
45 printf("\t for d \n");
46 K9=((%e)^(U*(A/(Q1))*(R-1)));
47 S2=((K9-1)/((K9*R)-1)); // eq 18.36
48 printf("\t K9 is : %.2f \n",K9);
49 printf("\t S2 is : %.2f \n",S2);
50 t=100;
51 t1=t+(S2*(T1-t)); // 18.37
52 printf("\t t1 is : %.0f F \n",t1);
53 t2=t1+(S2*(T1-t1));
54 printf("\t t2 is : %.0f F \n",t2);

```

```

55 t3=t2+(S2*(T1-t2));
56 printf("\t t3 is : %.0f F \n",t3);
57 t4=t3+(S2*(T1-t3));
58 printf("\t t4 is : %.0f F \n",t4);
59 x=0.23;
60 printf("\t fractional circulation is : %.2f \n",x);
61 N=3+x;
62 printf("\t total fractional circulation : %.2f \n",N
    );
63 theta3=(N*(M/w));
64 printf("\t theta3 is : %.2f \n",theta3);
65 // end

```

Scilab code Exa 18.2 Heat Flow through a Wall

```

1  printf("\t example 18.2 \n");
2  tav=500; // F
3  Ts=1000;
4  t0=100;
5  c=0.12; // Btu/(lb)*(F)
6  k=24; // Btu/(hr)*(ft ^2)*(F/ft)
7  row=488; // lb/ft ^3
8  alpha=0.41; // alpha=(k/(c*row)), ft ^2/hr
9  x=0.333; // ft
10 theta=4;
11 printf("\t values are approximately mentioned in the
    book \n");
12 X=(x/(2*(alpha*theta)^(1/2)));
13 printf("\t X is : %.2f \n",X);
14 Y=0.142; // Y=f1(X) from fig 18.7
15 t=Ts+(t0-Ts)*(Y); // eq 18.43
16 printf("\t t si : %.0f F \n",t);
17 q=((k*(Ts-t0))/(3.14*alpha*theta)^(1/2)); // q=(Q/A)
    ,from eq 18.47
18 printf("\t q is : %.0f Btu/(hr)*(ft ^2) \n",q);

```

```

19 q1=(2*k*(Ts-t0)*(theta/(3.14*alpha))^(1/2)); // q=(
    Q1/A). eq 18.49
20 printf("\t The total heat which flowed through a
    square foot of wall in the 4 hr is : %.1e Btu/ft
    ^2 \n",q1);
21 // end

```

Scilab code Exa 18.3 Center line Temperature of a Shaft

```

1 printf("\t example 18.3 \n");
2 Ts=1000;
3 t0=100;
4 alpha=0.41; // alpha=(k/(c*row)), ft^2/hr
5 theta=15/60;
6 l=1; // ft
7 X=(4*alpha*theta)/(l^2);
8 printf("\t X is : %.2f \n",X);
9 Y=0.155; // Y=f3*(X)from fig 18.9 when L=infinity
10 t=Ts+(t0-Ts)*(Y); // eq 18.52
11 printf("\t t si : %.1e F \n",t);
12 // end

```

Scilab code Exa 18.4 The Schack Chart

```

1 printf("\t example 18.4 \n");
2 T1=1100; // F
3 T2=70; // F
4 t1=T1+460; // R
5 t2=T2+460; // R
6 k=27; // from appendix
7 c=0.14; // from appendix
8 row=490; // from appendix
9 alpha=0.394;

```

```

10 theta=4;
11 l=10/12; // ft
12 x=0.173*10(-8); // stefan constant
13 e=0.7; // emmisivity
14 printf("\t values are approximately mentioned in the
      book \n");
15 printf("\t for a \n");
16 // Assume the temperature is 500 F after 4 hr. The
      coefficient from plate to air is the sum of the
      radiation and convection coefficients
17 hri=(e*x*(t14-t24))/(T1-T2);
18 printf("\t radiation coefficient is : %.1f Btu/(hr)
      *(ft2)*(F) \n",hri); // eq 4.32
19 hci=(0.3*(T1-T2)(1/4)); // eq 10.10
20 printf("\t convection coefficient is : %.1f Btu/(hr)
      *(ft2)*(F) \n",hci);
21 hti=hri+hci;
22 printf("\t total intial coefficient is : %.1f Btu/(
      hr)*(ft2)*(F) \n",hti);
23 // For the 4-hr coefficient at 500 F
24 hr=2.2; // Btu/(hr)*(ft2)*(F)
25 hc=1.35; // Btu/(hr)*(ft2)*(F)
26 ht=hr+hc;
27 printf("\t total intial coefficient is : %.1f Btu/(
      hr)*(ft2)*(F) \n",ht);
28 h=(hti+ht)/2;
29 printf("\t mean coefficient is : %.1f Btu/(hr)*(ft
      ^2)*(F) \n",h);
30 X=(4*alpha*theta)/(l2);
31 Y=(h*l)/(2*k);
32 printf("\t X is : %.1f \n",X);
33 printf("\t Y is : %.3f \n",Y);
34 Z=0.42; // Z=f3(X,Y), from fig 18.10
35 t=T2+((T1-T2)*Z); // eq 18.53
36 printf("\t t is : %.0f F \n",t);
37 printf("\t for b \n");
38 Z1=0.43; // Z=f4(X,Y), from fig 18.11
39 t1=T2+((T1-T2)*Z1); // eq 18.53

```

```

40 printf("\t temperature of center plane is : %.0f F \
    n",t1);
41 // end

```

Scilab code Exa 18.5 The Gumey Lurle Chart

```

1  printf("\t example 18.5 \n");
2  Ts=400;
3  t0=200;
4  k=25; // from appendix
5  c=0.12; // from appendix
6  row=490; // from appendix
7  alpha=0.45; // alpha=(k/(c*row))
8  theta=15/60;
9  l=8/12; // ft
10 h=50;
11 X=(4*alpha*theta)/(l^2);
12 Z=(2*k)/(h*l);
13 printf("\t X is : %.2f \n",X);
14 printf("\t Z is : %.1f \n",Z);
15 Y=0.31; // Y=(Ts-t)/(Ts-t0), from fig 18.13
16 t=Ts+(t0-Ts)*(Y); // eq 18.43
17 printf("\t t is : %.0f F \n",t);
18 //end

```

Scilab code Exa 18.6 The Application of Newmans Method to Heating a Brick

```

1  printf("\t example 18.6 \n");
2  Ts=300;
3  t0=70;
4  c=0.25; // Btu/(lb)*(F)
5  k=0.3; // Btu/(hr)*(ft^2)*(F/ft)

```

```

6 row=103; // lb/ft^3
7 alpha=0.01164; // alpha=(k/(c*row)), ft^2/hr
8 theta=1;
9 lx=9/12;
10 ly=4.5/12;
11 lz=2.5/12;
12 h=4.1;
13 printf("\t values are approximately mentioned in the
        book \n")
14 X1=(4*alpha*theta)/(lx^2);
15 Z1=(2*k)/(h*lx);
16 printf("\t X1 is : %.4f \n",X1);
17 printf("\t Z1 is : %.3f \n",Z1);
18 X2=(4*alpha*theta)/(ly^2);
19 Z2=(2*k)/(h*ly);
20 printf("\t X2 is : %.4f \n",X2);
21 printf("\t Z2 is : %.3f \n",Z2);
22 X3=(4*alpha*theta)/(lz^2);
23 Z3=(2*k)/(h*lz);
24 printf("\t X3 is : %.3f \n",X3);
25 printf("\t Z3 is : %.3f \n",Z3);
26 printf("\t at centre (2*x/l) is zero \n");
27 Yx=0.98; // fig 18.12
28 Yy=0.75; // fig 18.12
29 Yz=0.43; // fig 18.12
30 printf("\t at surface (2*x/l) is one \n");
31 Yx1=0.325; // fig 18.12
32 Yy1=0.29; // fig 18.12
33 Yz1=0.245; // fig 18.12
34 printf("\t center of brick \n");
35 t1=Ts-(Yx*Yy*Yz*(Ts-t0));
36 printf("\t t1 is : %.1f F \n",t1);
37 printf("\t corner of brick \n");
38 t2=Ts-(Yx1*Yy1*Yz1*(Ts-t0));
39 printf("\t t2 is : %.1f F \n",t2);
40 printf("\t center of 9 by 4.5in face \n");
41 t3=Ts-(Yx*Yy*Yz1*(Ts-t0));
42 printf("\t t3 is : %.1f F \n",t3);

```

```

43 printf("\t center of 9 by 2.5in face \n");
44 t4=Ts-(Yx*Yy1*Yz*(Ts-t0));
45 printf("\t t4 is : %.0f F \n",t4);
46 printf("\t center of 4.5 by 2.5in face \n");
47 t5=Ts-(Yx1*Yy*Yz*(Ts-t0));
48 printf("\t t5 is : %.1f F \n",t5);
49 printf("\t middle of long edge \n");
50 t6=Ts-(Yx*Yy1*Yz1*(Ts-t0));
51 printf("\t t6 is : %.0f F \n",t6);
52 //end

```

Scilab code Exa 18.7 The Graphical Determination of the Time Temperature Distribution

```

1 printf("\t example 18.7 \n");
2 t=20; // min
3 alpha=0.40; // ft^2/hr
4 delx=0.167; // ft
5 // From the conditions of Eq. (18.61) take time
   increments such that alpha(deltheta/delx^2)=1/2
6 printf("\t approximate values are mentioned in the
   book \n");
7 deltheta=(delx^2/(2*alpha));
8 printf("\t deltheta is : %.3f hr \n",deltheta);
9 N=(t/(deltheta*60));
10 printf("\t number of steps required : %.1f \n",N);
11 // end

```

Scilab code Exa 18.8 Calculations for a Wall with Periodic Temperature Variation

```

1 printf("\t example 18.8 \n");
2 k=0.3;

```



```

3 row=103;
4 c=0.25;
5 alpha=0.01164;
6 f=1/24;
7 t1=120;
8 t2=60;
9 printf("\t approximate values are mentioned in the
    book \n");
10 printf("\t temperature lag 6in below the surface \n"
    );
11 x=6/12;
12 theta=(x/2)*(1/(3.14*f*alpha))^(1/2); // eq 18.65
13 printf("\t theta is : %.2f hr \n",theta);
14 printf("\t amplitude \n");
15 deltom=(t1-t2)/2;
16 printf("\t deltom is : %.0f F \n",deltom);
17 delt=(deltom)*(%e)^(-x*(3.14*f/alpha)^(1/2)); // eq
    18.67
18 printf("\t delt is : %.1f F \n",delt); //
    calculation mistake in book
19 printf("\t temperature deviation after 2 hr \n");
20 theta1=2; // hr
21 deltx=(deltom)*((%e)^(-x*(3.14*f/alpha)^(1/2)))*cos
    ((2*3.14*f*theta1)-(x*(3.14*f/alpha)^(1/2))); //
    eq 18.69
22 printf("\t deltx is : %.1f F \n",deltx);
23 printf("\t heat flow during the half period \n");
24 q=(k*deltom*(2/(3.14*f*alpha))^(1/2)); // eq 18.70
25 printf("\t heat flow is : %.0f Btu/(hr)*(ft ^2) \n",q
    );
26 // end

```

Scilab code Exa 18.9 Calculation of the Length of a Bed

```

1 printf("\t example 18.9 \n");

```

```

2 G=60; // lb/(hr)*(ft ^2)
3 De=1/12; // ft
4 theta=6; // hr
5 cs=41.3; // Btu/(ft ^3)*(F)
6 c=0.0191; // Btu/(ft ^3)*(F)
7 f=0.45; // void fraction
8 T=90;
9 T1=200;
10 t0=50;
11 h=(0.79*(G/De)^0.7); // eq 18.90
12 printf("\t h is : %.1f \n",h);
13 X=(h*theta/(cs*(1-f)));
14 Y=(T-t0)/(T1-t0);
15 printf("\t X is : %.0f \n",X);
16 printf("\t Y is : %.3f \n",Y);
17 row=0.0807; // lb/(ft ^3) air
18 Z=24.5; // Z=(h*x*row/(c*G)), by comparing X an Y in
    fig 18.21
19 x=24.5*(c*G/(h*row));
20 printf("\t x is : %.1f ft \n",x);
21 // end

```

Chapter 19

Furnace Calculations

Scilab code Exa 19.1 Lobo and Evans

```
1 printf("\t example 19.1 \n");
2 // For orientation purposes , one can make an
   estimate of the number of tubes required in the
   radiant section by assuming avg flux is 12000 Btu
   /(hr)*(ft^2)
3 // from Fig.19.14 it can be seen that with a tube
   temperature of 800°F, an exit-gas temperature of
   1730 F will be required to effect such a flux.
4 printf("\t approxiate values are mentioned in the
   book \n");
5 Q=50000000; // Btu/hr
6 QF=(Q/0.75); // efficiency of tank is 75%
7 printf("\t heat liberated by the fuel : %.3e Btu/hr
   \n",QF);
8 w1=(QF/17130); // heating value of fuel is 17130Btu/
   lb
9 printf("\t fuel quantity : %.2e lb/hr \n",w1);
10 w2=(w1*17.44); // lb of fuel fired with 17.44lb of
   air
11 printf("\t air required : %.2e lb/hr \n",w2);
12 w3=(w1*0.3); // 0.3 lb of air is used for atomizing
```

```

    lb of fuel
13 printf("\t steam for atomizing : %.2e lb/hr \n",w3);
14 QA=(w2*82); // heating value at 400F is 82Btu/lb
15 printf("\t QA is : %.2e Btu/hr \n",QA);
16 printf("\t QS is negligible \n");
17 QW=(0.02*QF);
18 printf("\t QW is : %.2e Btu/hr \n",QW);
19 Qnet=(QF+QA-QW);
20 printf("\t Qnet is : %.2e Btu/hr \n",Qnet);
21 //Heat out m gases at 1730 F , 25 per cent excess
    air , 476 Btu/lb of flue gas
22 QG=(476*(w1+w2+w3));
23 printf("\t QG is : %.2e Btu/hr \n",QG);
24 Q1=(Qnet-QG);
25 printf("\t Q1 is : %.2e Btu/hr \n",Q1); //
    calculation mistake in book
26 A=(3.14*38.5*(5/12)); // area of tube
27 printf("\t area of tube is : %.1f ft^2 \n",A);
28 Nt=(Q1/(12000*A)); // 12000 is avg flux
29 printf("\t estimated number of tubes : %.0f \n",Nt);
30 // The layout of the cross section of the furnace
    may be as shown m Fig. 19.16.
31 // center to center distance is 8(1/2)in
32 Acp=(8.5*38.5/12);
33 printf("\t cold plane surface per tube : %.1f ft^2 \
    n",Acp); // calculation mistake in book
34 a=0.937; // a=alpha , from fig 19.11 as Ratio of
    center-to-center/OD is 1.7
35 Acp1=(Acp*a);
36 printf("\t Acp1 is : %.0f ft^2 \n",Acp1);
37 Acpt=(Acp1*Nt);
38 printf("\t total cold plane surface is : %.1e ft^2 \
    n",Acpt);
39 A1=(2*20.46*14.92); // from fig 19.16
40 printf("\t surface of end walls : %.0f ft^2 \n",A1);
41 A2=(38.5*14.92); // from fig 19.16
42 printf("\t surface of side wall : %.0f ft^2 \n",A2);
43 A3=(38.5*9.79); // from fig 19.16

```

```

44 printf("\t surface of bridge walls : %.0f ft^2 \n",
    A3);
45 A4=(2*20.46*38.5); // from fig 19.16
46 printf("\t surface of floor and arch : %.0f ft^2 \n"
    ,A4);
47 AT=(A1+A2+A3+A4);
48 printf("\t AT is : %.0f ft^2 \n",AT);
49 AR=(AT-Acpt);
50 printf("\t AR is : %.0f ft^2 \n",AR);
51 Ar=(AR/Acpt);
52 printf("\t ratio of areas is : %.2f \n",Ar);
53 printf("\t dimension ratio is 3:2:1 \n");
54 L=((2/3)*(38.5*20.46*14.92)^(1/3));
55 printf("\t length is : %.0f ft \n",L);
56 printf("\t gas emissivity \n");
57 // From the analysis of the fuel, the steam quantity
    , and the assumption that the humidity of the air
    is 50 per cent of saturation at 60F, the partial
    pressures of CO2 and H2O in the combustion gases
    with 25 per cent excess air are
58 pCO2=0.1084;
59 pH2O=0.1248
60 pCO2L=1.63; // pCO2L=(pCO2*L)
61 pH2OL=1.87;
62 P=((pCO2)/(pCO2+pH2O));
63 printf("\t percentage correction at P : %.3f \n",P);
64 Pt=pCO2L+pH2OL;
65 printf("\t Pt is : %.2f \n",Pt);
66 // %correction estimated to be 8%
67 eG=(((6500+14500)-(650+1950))/(39000-4400))*((100-8)
    /100); // values from fig 19.12 and 19.13, eq
    19.5
68 printf("\t eG is : %.3f \n",eG);
69 f=0.635; // from fig 19.15 as (AR/Acpt)=1.09 and eG
    =0.496
70 printf("\t overall exchange factor : %.3f \n",f);
71 Z=(Q1/(Acpt*f));
72 printf("\t Z is : %.2e \n",Z);

```

```

73 printf("\t TG required (at Ts = 800F) = 1670F
    compared with 1730 F assumed in heat balance) \n
    ");
74 // end

```

Scilab code Exa 19.2 Calculation of a Furnace by the Method of Wilson and Lobo and Hottel

```

1 printf("\t example 19.2 \n");
2 QF=50000000;
3 G=22.36;
4 Acpt=1500;
5 printf("\t approxiate values are mentioned in the
    book \n");
6 Q=(QF/(1+(G/4200)*(QF/Acpt)^(1/2))); // eq 19.15
7 printf("\t Q is : %.2e Btu/hr \n",Q);
8 printf("\t The radiant-section average rate will be
    8350 Btu/(hr) (ft2), and the exit-flue-gas
    temperature 1540 F by heat balance. \n");
9 // end

```

Scilab code Exa 19.3 Calculation of Performance by the Orrok Hudson Equation

```

1 printf("\t example 19.3 \n");
2 Qr=1.5; // Qr=(QF2/QF1)
3 Cr=1.5; // Cr=(CR2/CR1)
4 Gr=140/125; // Gr=(G2/G1)
5 Qr1=0.38; // Qr1=(Q1/QF1)
6 printf("\t approxiate values are mentioned in the
    book \n");
7 a1=1.63; // a1=(G1*(CR1/27)^(1/2)), from eq 19.17
8 printf("\t a1 is : %.2f \n",a1);

```

```

9 a2=1.37*(a1); // a2=(G2*(CR2/27)^(1/2))
10 printf("\t a2 is : %.2f \n",a2);
11 Qr2=(1/(1+a2)); // Qr2=(Q2/QF2),from eq 19.15
12 printf("\t Qr2 is : %.2f \n",Qr2);
13 Q21=(Qr2/Qr1)*(Qr); // Q21=(Q2/Q1)
14 printf("\t ratio of heats is : %.2f \n",Q21);
15 printf("\t Hence the radiant absorption will be
    increased only 22 per cent for an increase of 50
    per cent in the heat liberated. \n");
16 // end

```

Scilab code Exa 19.4 Calculation of the Equivalent Radiant Coefficient

```

1 printf("\t example 19.4 \n");
2 eS=0.9; // assumed
3 TG=1500;
4 TS=650;
5 pCO2=0.1084;
6 pH2O=0.1248;
7 printf("\t approxiate values are mentioned in the
    book \n");
8 L=(0.4*8.5)-(0.567*5); // table 19.1
9 printf("\t L is : %.3f ft \n",L);
10 pH2OL=0.1248*L;
11 pCO2L=0.1084*L;
12 printf("\t pH2OL is : %.4f atm-ft \n",pH2OL);
13 printf("\t pCO2L is : %.4f atm-ft \n",pCO2L);
14 qH2O=1050; // at TG, from fig 19.12 ana 19.13
15 qCO2=1700; // at TG, from fig 19.12 ana 19.13
16 qTG=(qH2O+qCO2);
17 printf("\t qTG is : %.0f \n",qTG);
18 qsH2O=165; // at TS, from fig 19.12 ana 19.13
19 qsCO2=160; // at TS, from fig 19.12 ana 19.13
20 qTS=(qsH2O+qsCO2);
21 printf("\t qTG is : %.0f \n",qTS);

```

```

22 q=(0.9*(qTG-qTS)); // q=(QRC/A)
23 printf("\t q is : %.1f \n",q);
24 P=((pC02)/(pC02+pH20));
25 printf("\t percentage correction at P : %.3f \n",P);
26 Pt=pC02L+pH20L;
27 printf("\t Pt is : %.4f \n",Pt);
28 // %correction estimated to be 2%
29 q1=(q*0.98); // // q1=(QRC/A)
30 printf("\t q1 is : %.2e \n",q1);
31 hr=(q1/(TG-TS));
32 printf("\t radiation coefficient is : %.2f Btu/(hr)
      *(ft^2)*(F) \n",hr);
33 //end

```

Scilab code Exa 19.5 Furnace calculation

```

1 printf("\t example 19.5 \n");
2 Q=500000;
3 printf("\t approxiate values are mentioned in the
      book \n");
4 a=(3.5+(3.14*4*(120/360)))/(2); // a=(alpha*Acp)
      from fig 19.17
5 AR=(3+3.6+3);
6 printf("\t a is : %.2f ft^2/ft \n",a);
7 printf("\t AR is : %.1f ft^2/ft \n",AR);
8 // Arbitrarily neglecting end walls and also .the
      side wall refractory over 3'0" above the floor
9 R=(AR/a);
10 printf("\t ratio of two areas is : %.2f \n",R);
11 eG=0.265;
12 TG=1174; // F
13 TS=500; // F
14 f=0.56; // from fig 19.15 as (AR/Acpt)=2.49 and eG
      =0.265
15 q=15300; // at TG and TS,q=(Q/(a*f))

```



```

16 // However, the convection coefficient is small, 1.0
    Btu/(hr)(ft2)(°F), and AR/a is not 2.0 as in
    the assumptions for the Lobo and Evans equation.
17 q1=(q)-(7*(TG-TS)); // q1=(Q/(a*f))
18 printf("\t q1 is : %.2e Btu/(hr)*(ft^2) \n",q1);
19 q2=(q1*f); // q2=(Q/(a))
20 printf("\t q2 is : %.2e Btu/(hr)*(ft^2) \n",q2);
21 printf("\t convection rate basis \n");
22 q3=(1*(TG-TS)*(4.2/a)); // q2=(Q/(a))
23 printf("\t q3 is : %.1e Btu/(hr)*(ft^2) \n",q3); //
    calculation mistake in book
24 qt=(q2+q3); // qt=(Q/(a))
25 printf("\t qt is : %.2e Btu/(hr)*(ft^2) \n",qt);
26 ar=(Q/qt);
27 printf("\t required a is : %.0f ft^2 \n",ar);
28 L=(ar/a);
29 printf("\t length required is : %.1f ft \n",L);
30 // end

```

Chapter 20

Additional applications

Scilab code Exa 20.1 Calculation of a Jacketed Vessel

```
1 printf("\t example 20.1 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T=150; // F
4 L=0.6; // ft
5 N=7500; // rev/hr
6 row=62.5; // lb/ft^3
7 mu=1.06; // at 150 F and from fig 14, lb/ft*hr
8 k=0.38; // Btu/(hr)*(ft^2)*(F/ft), from table 4
9 c=1; // Btu/(lb)*(F)
10 Rej=(L^2)*(N)*(row)/(mu);
11 printf("\t Rej is : %.1e \n",Rej);
12 Z=1; // Z=(mu/muw)^(0.14), regarded as 1 for water
13 Dj=1.01; // ft, from table 11
14 j=1100; // fig 20.2
15 hi=((j)*(k/Dj)*((c*mu/k)^(1/3))*(Z)^(0.14));
16 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
17 hoi=1500; // Btu/(hr)*(ft^2)*(F)
18 Uc=((hi*hoi)/(hi+hoi)); // from eq 6.38
19 printf("\t Uc is : %.0f Btu/(hr)*(ft^2)*(F) \n",Uc);
20 Rd=0.005;
```

```

21 hd=(1/Rd);
22 printf("\t hd is : %.0f \n",hd);
23 UD=((Uc*hd)/(Uc+hd));
24 printf("\t UD is : %.0f Btu/(hr)*(ft^2)*(F) \n",UD);
25 A=3.43; // ft^2
26 Q=32600;
27 delat=(Q/(UD*A));
28 printf("\t temperature difference is : %.0f F \n",
    delat);
29 Ts=(T+delat);
30 printf("\t temperature of the steam : %.0f F \n",Ts)
    ;
31 // end

```

Scilab code Exa 20.2 Calculation of a Tube Coil

```

1 printf("\t example 20.2 \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 T1=150; // F
4 T2=220; // F
5 L=0.6; // ft
6 N=7500; // rev/hr
7 row=62.5; // lb/ft^3
8 mu=1.06; // at 150 F and from fig 14, lb/ft*hr
9 k=0.38; // Btu/(hr)*(ft^2)*(F/ft), from table 4
10 c=1; // Btu/(lb)*(F)
11 Rej=(L^2)*(N)*(row)/(mu);
12 printf("\t Rej is : %.1e \n",Rej);
13 Z=1; // Z=(mu/muw)^(0.14), regarded as 1 for water
14 Dj=1.01; // ft, from table 11
15 j=1700; // fig 20.2
16 hi=((j)*(k/Dj)*((c*mu/k)^(1/3))*(Z)^(0.14));
17 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
18 hoi=1500; // Btu/(hr)*(ft^2)*(F)

```

```

19 Uc=((hi*hoi)/(hi+hoi)); // from eq 6.38
20 printf("\t Uc is : %.0f Btu/(hr)*(ft^2)*(F) \n",Uc);
21 Rd=0.005;
22 hd=(1/Rd);
23 printf("\t hd is : %.0f \n",hd);
24 UD=((Uc*hd)/(Uc+hd));
25 printf("\t UD is : %.1f Btu/(hr)*(ft^2)*(F) \n",UD);
26 Q=32600;
27 A=(Q/(UD*(T2-T1)));
28 printf("\t Area is : %.2f ft^2 \n",A);
29 a=0.1309; // ft^2/ft
30 a1=(3.14*0.8*a);
31 printf("\t area per turn is : %.3f ft^2 \n",a1);
32 n=(A/a1);
33 printf("\t number of turns : %.1f \n",n);
34 // end

```

Scilab code Exa 20.3 Calculation of a Submerged pipe Coil Slurry Cooler

```

1 printf("\t example 20.3 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=675; // inlet hot fluid ,F
4 T2=200; // outlet hot fluid ,F
5 t1=120; // inlet cold fluid ,F
6 t2=140; // outlet cold fluid ,F
7 W=33100; // lb/hr
8 w=510000; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for oil \n");
11 c=0.64; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for oil is : %.2e Btu
   /hr \n",Q);
14 printf("\t for water \n");

```

```

15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.2e
    Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=230;
23 printf("\t LMTD is :%.0f F \n",LMTD);
24 Tc=((T2)+(T1))/(2); // caloric temperature of hot
    fluid ,F
25 printf("\t caloric temperature of hot fluid is : %.1
    f F \n",Tc);
26 tc=((t1)+(t2))/(2); // caloric temperature of cold
    fluid ,F
27 printf("\t caloric temperature of cold fluid is : %
    .0f \n",tc);
28 printf("\t hot fluid:inner tube side , oil \n");
29 at=0.0458; // flow area , ft^2 , table 11
30 printf("\t flow area is : %.4f ft^2 \n",at);
31 Gt=(W/(at)); // mass velocity ,lb/(hr)*(ft^2)
32 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
33 mu2=5.56; // at 400F,lb/(ft)*(hr)
34 D=0.242; // ft , table 11
35 Ret=((D)*(Gt)/mu2); // reynolds number
36 printf("\t reynolds number is : %.2e \n",Ret);
37 jH=100; // from fig.24
38 Z=0.245; // Z=(k(c*mu/k)^(1/3)) , Btu/(hr)*(ft)*(F/ft
    ), fig 16
39 hi=((jH)*(Z/D)); //Hi=(hi/phyp) , using eq.6.15 ,Btu/(
    hr)*(ft^2)*(F)
40 printf("\t hi is : %.0f Btu/(hr)*(ft^2)*(F) \n",hi);
41 ID=2.9; // ft
42 OD=3.5; // ft
43 hio=((hi)*(ID/OD)); // using eq.6.5
44 printf("\t Correct hio to the surface at the OD is :

```

```

        %.1f Btu/(hr)*(ft^2)*(F) \n",hio);
45 ho=150; // Btu/(hr)*(ft^2)
46 tw=(tc)+(((hio)/(hio+ho))*(Tc-tc)); // from eq.5.31
47 printf("\t tw is : %.0f F \n",tw);
48 tf=(tw+tc)/2;
49 printf("\t tf is : %.0f F \n",tf);
50 delt=110; // F
51 d0=3.5; // in, fig 10.4
52 Uc=((ho*hio)/(ho+hio)); // from eq 6.38
53 printf("\t Uc is : %.1f Btu/(hr)*(ft^2)*(F) \n",Uc);
54 Rd=0.01;
55 hd=(1/Rd);
56 printf("\t hd is : %.0f \n",hd);
57 UD=((Uc*hd)/(Uc+hd));
58 printf("\t UD is : %.0f Btu/(hr)*(ft^2)*(F) \n",UD);
59 A=(Q/(UD*(LMTD)));
60 printf("\t Area is : %.0f ft^2 \n",A);
61 a=0.917; // ft^2/ft, table 11
62 L=(A/(a*24));
63 printf("\t pipe length : %.0f \n",L);
64 // end

```

Scilab code Exa 20.4 Calculation of a Trombone 2

```

1 printf("\t example 20.4 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=450; // inlet hot fluid ,F
4 T2=150; // outlet hot fluid ,F
5 t1=85; // inlet cold fluid ,F
6 t2=100; // outlet cold fluid ,F
7 W=3360; // lb/hr
8 w=11100; // lb/hr
9 printf("\t 1.for heat balance \n");
10 printf("\t for SO2 \n");

```

```

11 c=0.165; // Btu/(lb)*(F)
12 Q=((W)*(c)*(T1-T2)); // Btu/hr
13 printf("\t total heat required for SO2 is : %.3e Btu
    /hr \n",Q);
14 printf("\t for water \n");
15 c=1; // Btu/(lb)*(F)
16 Q=((w)*(c)*(t2-t1)); // Btu/hr
17 printf("\t total heat required for water is : %.3e
    Btu/hr \n",Q);
18 delT1=T2-t1; //F
19 delT2=T1-t2; // F
20 printf("\t delT1 is : %.0f F \n",delT1);
21 printf("\t delT2 is : %.0f F \n",delT2);
22 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
23 printf("\t LMTD is :%.0f F \n",LMTD);
24 R=20;
25 S=0.0412;
26 FT=0.98; // fig 18
27 delT=(FT*LMTD);
28 printf("\t delT is : %.0f F \n",delT);
29 Tc=((T2)+(T1))/(2); // caloric temperature of hot
    fluid ,F
30 printf("\t caloric temperature of hot fluid is : %.0
    f F \n",Tc);
31 tc=((t1)+(t2))/(2); // caloric temperature of cold
    fluid ,F
32 printf("\t caloric temperature of cold fluid is : %
    .1f \n",tc);
33 printf("\t hot fluid:inner tube side , SO2 \n");
34 at=0.0512; // flow area , ft^2, table 11
35 printf("\t flow area is : %.4f ft^2 \n",at);
36 Gt=(W/(at)); // mass velocity ,lb/(hr)*(ft^2)
37 printf("\t mass velocity is : %.2e lb/(hr)*(ft^2) \n
    ",Gt);
38 mu2=0.041; // at 300F,lb/(ft)*(hr), fig 15
39 D=0.256; // ft , table 11
40 Ret=((D)*(Gt)/mu2); // reynolds number
41 printf("\t reynolds number is : %.1e \n",Ret);

```

```

42 jH=790; // from fig.24
43 Z=0.006831; // Z=(k(c*mu/k)^(1/3)), Btu/(hr)*(ft)*(F
    /ft)
44 hi=((jH)*(Z/D)); //Hi=(hi/phy), using eq.6.15, Btu/(
    hr)*(ft^2)*(F)
45 printf("\t hi is : %.1f Btu/(hr)*(ft^2)*(F) \n",hi);
46 ID=3.068; // ft
47 OD=3.5; // ft
48 hio=((hi)*(ID/OD)); // using eq.6.5
49 printf("\t Correct hio to the surface at the OD is :
    %.1f Btu/(hr)*(ft^2)*(F) \n",hio);
50 printf("\t cold fluid water \n");
51 L=8; // ft
52 G=(w/(2*L));
53 printf("\t G : %.0f lb/(hr)*(ft) \n",G);
54 mu1=1.94; // at 92.5F, lb/(ft)*(hr)
55 Re=(4*G/mu1);
56 printf("\t Re is : %.2e \n",Re);
57 Do=0.292; // ft
58 ho=(65*(G/Do)^(1/3));
59 printf("\t ho is : %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
60 Uc=((ho*hio)/(ho+hio)); // from eq 6.38
61 printf("\t Uc is : %.1f Btu/(hr)*(ft^2)*(F) \n",Uc);
62 Rd=0.01;
63 hd=(1/Rd);
64 printf("\t hd is : %.0f \n",hd);
65 UD=((Uc*hd)/(Uc+hd));
66 printf("\t UD is : %.1f Btu/(hr)*(ft^2)*(F) \n",UD);
67 A=(Q/(UD*(LMTD)));
68 printf("\t Area is : %.1f ft^2 \n",A); //
    calculation mistake in book
69 a=0.917; // ft^2/ft, table 11
70 l=(A/(a*8));
71 printf("\t pipe length : %.2f \n",l);
72 // end

```

Scilab code Exa 20.5 Calculation of an Atmospheric Jacket Water Cooler

```
1 printf("\t example 20.5 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 Nt=25; // number of tubes
4 A=50; // total projected area
5 Tav=100; // F
6 s=28; // assumption spray, lb/(min)*(ft^2)
7 Do=0.0833; // ft
8 PH=0.1562;
9 Y=0.874;
10 Z=0.466;
11 E=(0.171*(Do*Y*Z)^0.1); // (E/(Do*Y*Z)^0.1)=0.171,
   from fig 20.10
12 printf("\t evaporation percentage is : %.2f \n",E);
13 Q=(295*500*(143-130));
14 printf("\t heat load is : %.2e Btu/hr \n",Q);
15 Q1=(Q*(1-0.12));
16 printf("\t sensible heat is : %.2e Btu/hr \n",Q1);
17 t2=(90)+(Q1/(28*60*50));
18 printf("\t final spray temperature is : %.0f F \n",
   t2);
19 w=(s*60*50);
20 printf("\t total spray : %.1e lb/hr \n",w);
21 m=(w/(2*4*12));
22 printf("\t m is : %.0f lb/(hr)*(ft^2) \n",m);
23 mu=1.84; // lb/(ft)*(hr)
24 Z=((m^0.3)*Do*Y*Z/(mu*0.125));
25 printf("\t Z is : %.2f \n",Z);
26 N=3; // assume 3 horizontal rows
27 ho=300*(N^0.05); // (ho/(N^0.05))=300, from fig
   20.11
28 printf("\t ho is : %.0f Btu/(hr)*(ft^2)*(F) \n",ho);
```

```

29 printf("\t tube side coefficient \n");
30 printf("\t assuming even number of passes and tube
    side velocity about 8fps \n");
31 at=0.0775; // ft^2
32 Gt=(295*500/(at)); // mass velocity ,lb/(hr)*(ft^2)
33 printf("\t mass velocity is : %.1e lb/(hr)*(ft^2) \n
    ",Gt);
34 V=(Gt/(3600*62.5));
35 printf("\t velocity is : %.2f fps \n",V);
36 hi=2140; // Btu/(hr)*(ft^2)*(F), fig 25
37 ID=0.87; // ft
38 OD=1; // ft
39 hio=((hi)*(ID/OD)); // using eq.6.5
40 printf("\t Correct hio to the surface at the OD is :
    %.2e Btu/(hr)*(ft^2)*(F) \n",hio);
41 Uc=((ho*hio)/(ho+hio)); // from eq 6.38
42 printf("\t Uc is : %.0f Btu/(hr)*(ft^2)*(F) \n",Uc);
43 a=0.2618; // ft^2, table 11
44 A1=(2*3*25*12*a);
45 printf("\t total surface is : %.0f ft^2 \n",A1);
46 T1=143; // inlet hot fluid ,F
47 T2=130; // outlet hot fluid ,F
48 t1=90; // inlet cold fluid ,F
49 t2=110; // outlet cold fluid ,F
50 delT1=T2-t1; //F
51 delT2=T1-t2; // F
52 printf("\t delT1 is : %.0f F \n",delT1);
53 printf("\t delT2 is : %.0f F \n",delT2);
54 LMTD=((delT2-delT1)/((2.3)*(log10(delT2/delT1))));
55 printf("\t LMTD is :%.1f F \n",LMTD); // calculation
    mistake in book
56 R=0.65;
57 S=0.377;
58 FT=0.97; // fig 18
59 delT=(FT*LMTD);
60 printf("\t delT is : %.1f F \n",delT);
61 UD=(Q/(A1*(delT)));
62 printf("\t UD is : %.0f Btu/(hr)*(ft^2)*(F) \n",UD);

```

```

63 Rd=((Uc-UD)/((UD)*(Uc))); // (hr)*(ft^2)*(F)/Btu
64 printf("\t actual Rd is : %.4f (hr)*(ft^2)*(F)/Btu \
n",Rd);
65 printf("\t The assumption of three horizontal rows
is satisfactory , since a dirt factor of 0.004 was
required \n");
66 // end

```

Scilab code Exa 20.6 Calculation of the True Temperature Difference

```

1 printf("\t example 20.6 \n");
2 printf("\t approximate values are mentioned in the
book \n");
3 T1=200; // inlet hot fluid ,F
4 T2=100; // outlet hot fluid ,F
5 t1=50; // inlet cold fluid ,F
6 t2=100; // outlet cold fluid ,F
7 R=((T1-T2)/(t2-t1));
8 printf("\t R is : %.0f \n",R);
9 V=((T1+T2-t1-t2)/(t2-t1))/(2);
10 printf("\t V is : %.1f \n",V);
11 u=120;
12 U=60;
13 F=((u*1)/(U*2));
14 printf("\t F is : %.0f \n",F);
15 E=1.1; // In Fig.20.18b for R = 2.0 and F = 1.0, the
abscissa and ordinate intersect at E =1.10.
16 Z=(E/V);
17 printf("\t Z is : %.3f \n",Z);
18 deltdD=0.783*V; // deltdD/V=0.783, from fig 20.17
19 printf("\t deltdD is : %.3f \n",deltdD);
20 deltd=(deltdD*(t2-t1));
21 printf("\t deltd is : %.1f \n",deltd);
22 deltd1=T2-t1; //F
23 deltd2=T1-t2; // F

```

```

24 printf("\t delt1 is : %.0f F \n",delt1);
25 printf("\t delt2 is : %.0f F \n",delt2);
26 LMTD=((delt2-delt1)/((2.3)*(log10(delt2/delt1))));
27 printf("\t LMTD is :%.0f F \n",LMTD);
28 // end

```

Scilab code Exa 20.7 calculation of Sand Cooling with Negllgible Resistance

```

1 printf("\t example 20.7 \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 T1=284; // inlet hot fluid ,F
4 T2=104; // outlet hot fluid ,F
5 t1=86; // inlet cold fluid ,F
6 t2=104; // outlet cold fluid ,F
7 W=1000; // lb/hr
8 k=0.15; // thermal conductivity
9 L=10;
10 Beta=((2*k)/(500*(2/12))); // hoi=500Btu/(hr)*(ft ^2)
   *(F) for water
11 printf("\t beta is : %.4f \n",Beta);
12 printf("\t for sand \n");
13 C=0.2; // Btu/(lb)*(F)
14 Q=((W)*(C)*(T1-T2)); // Btu/hr
15 printf("\t total heat required for sand is : %.1e
   Btu/hr \n",Q);
16 c=1;
17 w=(Q/(t2-t1));
18 printf("\t w is : %.0e lb/hr \n",w);
19 R=((W*C)/(w*c));
20 printf("\t R is : %.1f \n",R);
21 S=((T2-T1)/(t1-T1));
22 printf("\t S is : %.2f \n",S);
23 W1=(8.33*(k*L)/C); // ((W1*C)/(k*L))=8.33 from fig

```

```

    20.20b for Beta=0
24 printf("\t rate per tube is : %.1f lb/hr \n",W1);
25 N1=(W/W1);
26 printf("\t number of tubes : %.0f \n",N1);
27 printf("\t for air assume hoi=9 and Beta=0.2 \n");
28 c1=0.25;
29 w1=(Q/(c1*(t2-t1)));
30 printf("\t w1 is : %.0e lb/hr \n",w1);
31 W2=(5.23*(k*L)/C); // ((W1*C)/(k*L))=5.23 from fig
    20.20b for Beta=0.2
32 printf("\t rate per tube is : %.0f lb/hr \n",W2);
33 N2=(W/W2);
34 printf("\t number of tubes : %.0f \n",N2);
35 // end

```

Scilab code Exa 20.8.1 Immersion Water Heater

```

1 printf("\t example 20.8a \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 L=3; // ft
4 B=2; // ft
5 h=18/12; // ft , height of water present in tank
6 printf("\t unsteady state \n");
7 m=(L*B*h*62.5);
8 printf("\t Lb of water is : %.1f lb \n",m);
9 t1=50;
10 t2=150;
11 c=1;
12 Q=(m*c*(t2-t1))/(2*3412); // kwhr
13 printf("\t heat to be supplied : %.2f kwhr \n",Q);
14 printf("\t losses \n");
15 Q1=(L*B*260)/(1000); // from fig 20.25c
16 printf("\t from surface of water : %.2f kwhr \n",Q1)
    ;

```

```

17 Q2=(5.5*((2*B*2)+(2*L*B))/(1000)); // from fig 20.25
    c
18 printf("\t from sides of vessel : %.2f kwhr \n",Q2);
19 printf("\t losses from bottom are negigible \n");
20 Qt=(Q+Q1+Q2);
21 printf("\t total requirement : %.2f kwhr \n",Qt);
22 printf("\t steady state \n");
23 m1=8; // gal/hr
24 Qs=(m1*8.33*c*(t2-t1))/(3412); // kwhr
25 printf("\t heat to be supplied : %.2f kwhr \n",Qs);
26 Qts=(Qs+Q1+Q2);
27 printf("\t total requirement : %.2f kwhr \n",Qts);
28 // end

```

Scilab code Exa 20.8.2 Strip Heater for Air Heating

```

1 printf("\t example 20.8b \n");
2 printf("\t approximate values are mentioned in the
    book \n");
3 m=100; // lb
4 t1=70;
5 t2=370;
6 L=4;
7 B=3;
8 n=4; // number of air changers
9 c1=0.12
10 Q1=(m*c1*(t2-t1));
11 printf("\t heat to steel charge : %.1e Btu \n",Q1);
12 c2=0.25
13 Q2=(n*L*B*2*0.075*c2*(t2-t1));
14 printf("\t heat to air : %.1e Btu \n",Q2);
15 printf("\t From Fig. 20.25a for 52ft^2 of oven
    outside surface and a temperature rise of 300F
    the loss is 5kw for 1 in.thick insulations.For 2
    in.thick insulation the loss is 2.5kw \n");

```

```

16 Qt=((Q1+Q2)/(3412))+(2.5);
17 printf("\t total requirement : %.2f kw \n",Qt);
18 // end

```

Scilab code Exa 20.8.3 Finned strip Heater

```

1 printf("\t example 20.8c \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 m=270; // cfm
4 t1=70;
5 t2=120;
6 L=1.5; // ft
7 B=1.5; // ft
8 c=0.25
9 row=0.075; // lb/ft^3
10 Q=(m*row*60*c*(t2-t1));
11 printf("\t heat : %.2e Btu \n",Q);
12 V=(m/(L*B*60)); // fps
13 printf("\t velocity is : %.0f fps \n",V);
14 printf("\t Refer to Fig.20.22a.The air is capable of
   removing 33watts/in which is the maximum
   dissipation which may be expected. Any group of
   heaters providing 5 kw which do not require a
   dissipation of more than 33 w/in. and which will
   fit into the duct will be satisfactory \n");
15 printf("\t Thus in Table 20.3 elements of 350 watts
   with a total length each of 18 in. each are
   satisfactory \n");
16 // end

```

Scilab code Exa 20.8.4 Clamp on Plastic Heating

```

1 printf("\t example 20.8d \n");
2 printf("\t approximate values are mentioned in the
   book \n");
3 t1=70;
4 t2=300;
5 L=26; // in
6 B=12; // in
7 H=1; // in
8 c1=0.13
9 // specific gravity of cast iron is 7.2
10 printf("\t unsteady state \n");
11 m=(L*B*H*62.5*7.2/1728); // lb
12 printf("\t weight of plate : %.0f lb \n",m);
13 Q1=(m*c1*(t2-t1));
14 printf("\t heat : %.1e Btu \n",Q1);
15 printf("\t From Figure 20.25b for a black body the
   radiation is 1.5w/in^2.The radiation from the top
   is actually 110 per cent of this value , and from
   the bottom of the plate it is 55 per cent for an
   average of 82.5 per cent is taken \n");
16 Q2=(2*26*12*1.5*0.825/1000); // ke
17 printf("\t radiation loss : %.1f kw \n",Q2);
18 Qt=((Q1)/(3412))+(Q2);
19 printf("\t total requirement : %.1f kw \n",Qt);
20 printf("\t staedy state \n");
21 m2=70;
22 c2=0.22;
23 Qs=(m2*c2*(t2-t1));
24 printf("\t heat : %.2e Btu \n",Qs);
25 Q1=0.8; // kw
26 Qts=((Qs)/(3412))+(Q1);
27 printf("\t total requirement : %.2f kw \n",Qts);
28 printf("\t The steady state is controlling.The
   requirements are satisfied , by four 24-in. strip
   heaters , but the sheath temperature must now be
   checked. Since the temperature drop per unit flux
   density is 14 to 19F, assume an average of 16.5
   F. For clamp-on strips 24 in. long the watts

```



```
    per square inch deliverable are 16 \n");
29 delt=(16*16.5);
30 printf("\t delt is : %.0f F \n",delt);
31 printf("\t The sheath temperature is then 300 + 264
    = 564 F , which is satisfactory for steel
    sheathed elements with a 750F maximum. \n");
32 // end
```
