Scilab Textbook Companion for Stoichiometry by B. I. Bhatt And S. B. Thakore[1]

Created by
Sumit Kumar Singh
B.Tech
Chemical Engineering
NIT Rourkela, Odhisa
College Teacher
Hara Mohan Jena
Cross-Checked by
Ganesh

May 24, 2016

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

Book Description

Title: Stoichiometry

Author: B. I. Bhatt And S. B. Thakore

Publisher: Tata McGraw - Hill Education

Edition: 5

Year: 2010

ISBN: 9780070681149

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 1

Dimensions and Units

Scilab code Exa 1.1 Mass flow rate

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 1
6  // Dimensions and Units
7
8
9  // Example 1.1
10 // Page 12
11 printf("Example 1.1, Page 12 \n \n");
12
13 // solution
14
15 // Using conversion factors from table 1.3 (Pg 9)
16 q1 = 75 // [gallon/min] (volumetric flow rate)
17 q2 = 75/(60*.219969) // [dm^3/s]
18 row = 0.8 // [kg/dm^3]
19 q3 = q2*row // [kg/s] (mass flow rate)
20 printf("mass flow rate = "+string(q3)+" [kg/s] \n")
```

Scilab code Exa 1.2 steam velocity in pipeline

```
clear;
clc;

// Stoichiometry
// Chapter 1
// Dimensions and Units


// Example 1.2
// Page 12
printf("Example 1.2, Page 12 \n \n");

// solution

qm = 2000 // [kg/h] (mass flow rate)
d1 = 3.068 // [in] (internal dia of pipe)
// Using conversion factors from table 1.3 (Pg 9)
d2 = 3.068/.0393701 // [mm]
A = ((%pi/4)*d2^2)/10^6 // [m^2] (cross section area
    )
// Using steam tables; Appendix IV.3
v = 0.46166 // [m^3/kg] (sp. vol. of steam at 440
    kPa)
qv = (qm*v)/3600 // [m^3/s]
vs = qv/A // [m/s]
printf("velocity of the steam in the pipeline is "+
    string(vs)+" m/s")
```

Scilab code Exa 1.3 conversion of TR

```
clear;
clc;

// Stoichiometry
// Chapter 1
// Dimensions and Units


// Example 1.3
// Page 13
printf("Example 1.3, Page 13 \n \n");

// solution

m = 2000 // [lb] (mass flow rate)
t = 24 //[hr]
lf = 144 // [Btu/lb] (latent heat of fusion)
// Using conversion factors from table 1.3 (Pg 9)
TR = (m*lf*.251996*4.184)/(3600*24)
printf("1 TR = "+string(TR)+" kW")
```

Scilab code Exa 1.4 conversion of equation into SI units

```
clear;
clc;

// Stoichiometry
// Chapter 1
// Dimensions and Units


// Example 1.4
// Page 13
printf("Example 1.4, Page 13 \n \n");

```

```
13 // solution
14
15 // C = 89.2*A*(T/M^).5    [ft^3/s]
16 k = 89.2 //
17 C1 = 1 // [ft^3/s]
18 // Using conversion factors from table 1.3 (Pg 9)
19 C2 = 35.31467*C1
20 T1 = 1 //[dgree R]
21 T2 = 1.8*T1 // [K]
22 A1 = 1 // [ft^2]
23 A2 = 10.76391
24 k2 = (k*A2*(1.8)^.5)/35.34167
25 printf("eq in SI becomes \n C = "+string(k2)+"*(T/M)
    ^.5   [m^3/s]")
```

Chapter 2

Basic Chemical Calculations

Scilab code Exa 2.1 gm of NH4Cl

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.1
10 // Page 17
11 printf("Example 2.1, Page 17 \n \n");
12
13 // solution
14
15 // NH4Cl
16 M = 14+4+35.5 // [g] (molar mass of NH4Cl)
17 n=5 // [mol]
18 m = M*n // [g]
19 printf("5 mol of NH4Cl = "+string(m)+" [g]")
```

Scilab code Exa 2.2 equivalent moles of CuSO4

```
clear;
clc;

// Stoichiometry
// Chapter 2
// Basic Chemical Calculations


// Example 2.2
// Page 17
printf("Example 2.2, Page 17 \n \n");

// solution

// CuSO4.5H2O
M1 = 159.5 // [g] (molar mass of CuSO4)
M2 = 159.5+5*(2+16) // (molar mass of CuSO4.5H2O)
m = 499
n = m/M2 //[mol]
printf("In the formula CuSO4.5H2O, the moles of
   CuSO4 is one hence, \n the equivalent moles of
   CuSO4 in the crystal is "+string(n)+".")
```

Scilab code Exa 2.3 moles of K2CO3

```
clear;
clc;

// Stoichiometry
// Chapter 2
```

```
6  // Basic Chemical Calculations
7
8
9  // Example 2.3
10 // Page 17
11 printf("Example 2.3, Page 17 \n \n");
12
13 // solution
14
15 // K2CO3
16 m = 117 // [kg] (wt of K)
17 Mk = 39 // [g] (at wt of K)
18 a = m/Mk // [kg atoms]
19 // 1 mol of K2CO3 contains 2 atoms of K
20 n = a/2 // [kmol] (moles of K2CO3)
21 printf(" "+string(n)+" kmol of K2CO3 contains 117 kg
       of K.")
```

Scilab code Exa 2.4 no of atoms of BaCl2

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.4
10 // Page 18
11 printf("Example 2.4, Page 18 \n \n");
12
13 // solution
14
15 // BaCl2
```

```
16 M = 137.3+2*35.5 // [g] (molar mass of BaCl2)
17 m = 416.6 // [g]
18 n = m/M // [mol]
19 N = n*6.022*10^23 // (no. of atoms)
20 printf("Atoms present in 416.6 g BaCl2 = "+string(N)
   +"")
```

Scilab code Exa 2.5 equivalent mass

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.5
10 // Page 19
11 printf("Example 2.5, Page 19 \n \n");
12
13 // solution
14
15 printf("(a) \n \n")
16 //PO4 radical
17 M = 31+4*16 //[g]
18 V = 3 // (valence of PO4)
19 eqm = M/V
20 printf("eq. mass of PO4 is "+string(eqm)+" [g] \n \n
       \n")
21 printf("(b) \n \n")
22 //Na3PO4
23 M = 3*23+95 //[g]
24 V = 3
25 eqm = M/V
```

```
26 printf("eq. mass of Na3PO4 is "+string(eqm)+" [g] \n
    \n")
```

Scilab code Exa 2.6 equivalents

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.6
10 // Page 19
11 printf("Example 2.6, Page 19 \n \n");
12
13 // solution
14
15 // AlCl3
16 v = 3 // valency of Al ion
17 eq = 3*3 // [mol]
18 printf("no. of equivalents in 3 kmol of AlCl3 is "+
   string(eq)+" keq.")
```

Scilab code Exa 2.7 composition of mixture

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
```

```
7
8
9  // Example 2.7
10 // Page 20
11 printf("Example 2.7, Page 20 \n \n");
12
13 // solution
14
15 // (a)
16 printf("(a) \n \n")
17 // mass %
18 m1 = 600 //[kg] (NaCl)
19 m2 = 200 //[kg] (KCl)
20 m = m1+m2 // total mass
21 Wa = (m1/m)*100
22 Wb = (m2/m)*100
23 printf("mass percentage of NaCl is "+string(Wa)+" \
   nmass percentage of KCl is "+string(Wb)+" \n \n \
   n")
24 // (b)
25 printf("(b) \n \n")
26 //mol %
27 M1 = 23+35.5 // molar mass of NaCl
28 n1 = m1/M1 // no. of moles of NaCl
29 M2 = 39+35.5 // molar mass of KCl
30 n2 = m2/M2 // no. of moles of KCl
31 n = n1+n2
32 N1 = (n1/n)*100
33 N2 = (n2/n)*100
34 printf("mol percentage of NaCl is "+string(N1)+" \
   nmol percentage of KCl is "+string(N2)+" \n")
```

Scilab code Exa 2.8 composition and molar mass

```
1 clear;
```

```
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.8
10 // Page 21
11 printf("Example 2.8, Page 21 \n \n");
12
13 // solution
14 // CH.35O.35S.14
15 // mass %
16 C = 12.0107 //[kg]
17 H = 1.00794*.35 // [kg]
18 O = 15.9994*.35// [kg]
19 S = 32.065*.14 //[kg]
20 m = C+H+O+S
21 m1 = (C/m)*100
22 m2 = (H/m)*100
23 m3 = (O/m)*100
24 m4 = (S/m)*100
25 printf("mass percentage of C is "+string(m1)+" \
      nmass percentage of H is "+string(m2)+" \nmass
      percentage of O is "+string(m3)+" \nmass
      percentage of S is "+string(m4)+" \n \n")
26 M = m/(1+.35+.35+.14)
27 printf("molar mass = "+string(M)+" kg/kmol.")
```

Scilab code Exa 2.9 actual urea content

```
1  clear;
2  clc;
3
```

```
4 // Stoichiometry
5 // Chapter 2
6 // Basic Chemical Calculations
7
8
9 // Example 2.9
10 // Page 22
11 printf("Example 2.9, Page 22 \n \n");
12
13 // solution
14
15 // basis 100kg urea
16 m1 = 45 //[kg] (mass of N present)
17 Mu = 60 // (molar mass of urea)
18 m2 = 14*2 //[kg] (mass of N in 1 kmol of urea)
19 m = (Mu/m2)*m1
20 printf("The sample contains "+string(m)+" percent
   urea.")
```

Scilab code Exa 2.10 mass percent

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 2
6 // Basic Chemical Calculations
7
8
9 // Example 2.10
10 // Page 22
11 printf("Example 2.10, Page 22 \n \n");
12
13 // solution
14
```

```
15 // NaOH
16
17 Impurity = 60 // [ppm] SiO2
18 m = (60/1000000)*100
19 printf("Mass percent of SiO2 is "+string(m)+".")
```

Scilab code Exa 2.11 no of ions

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.11
10 // Page 22
11 printf("Example 2.11, Page 22 \n \n");
12
13 // solution
14
15 Ca = 40.078 // at. wt of Ca
16 F = 18.9984032 // at wt of F
17 M1 = 3*Ca +2*(30.97762+(4*15.9994)) // molar mass of
       Ca3PO4
18 M2 = Ca +12.0107+3*15.9994 // molar mass of CaCO3
19 M3 = Ca+2*F // molar mass of CaF2
20 m1 = 800 //[mg] Ca3PO4
21 m2 = 200 //[mg] CaCO3
22 m3 = 5 //[mg] CaF2
23 n1 = ((3*Ca)/M1)*m1+(Ca/M2)*m2+(Ca/M3)*m3 // [mg]
      total Ca ions
24 n2 = (F/M3)*2*5 //[mg] total F ions
25 printf("Total no. of Ca+ ions is "+string(n1)+" and
```

```
\ntotal no. of F- ions is "+string(n2)+".")
```

Scilab code Exa 2.12 composition of solution

```
clear;
clc;

// Stoichiometry
// Chapter 2
// Basic Chemical Calculations


// Example 2.12
// Page 23
printf("Example 2.12, Page 23 \n \n");

// solution

// (a)
printf("(a) \n \n")
// mass %
m1 = 100 //[kg] methanol (basis)
m2 = 64 //[kg] salicylic acid
m = m1+m2 // [kg] mass of solution
w1 = m2/m*100
w2 = 100-w1
printf("mass percent of salicylic acid is "+string(
   w1)+" and \nmass percent of methanol is "+string(
   w2)+". \n \n \n")

//(b)
printf("(b) \n \n")
//mole %
M1 = 32 // molar mass of methanol
M2 = 138 //molar mass of salicylic acid
```

```
30 n1 = m1/M1 //[kmol] methanol
31 n2 = m2/M2 //[kmol] salicylic acid
32 n = n1+n2
33 N1 = n1/n*100
34 N2 = n2/n*100
35 printf("Mole percent of methanol is "+string(N1)+"
   and \nMole percent of salicylic acid is "+string(
   N2)+".")
```

Scilab code Exa 2.13 composition of solution

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.13
10 // Page 24
11 printf("Example 2.13, Page 24 \n \n");
12
13 // solution
14
15
16 //mass %
17 m1 = 13.70 // HCl
18 m2 = 8.67 // NaCl
19 m3 = 100 // H2O
20 m = m1+m2+m3 // mass of solution
21 w1 = m1/m*100
22 w2 = m2/m*100
23 w3 = m3/m*100
24
```

```
25 printf("mass percent of HCl is "+string(w1)+" , \
     nmass percent of NaCl is "+string(w2)+"and \nmass
      percent of H2O is "+string(w3)+". \n \n \n")
26 M1=36.4609 //HCl
27 M2=58.4428 //NaCl
28 M3=18.0153 //H2O
29 n1=m1/M1 //HCl
30 n2=m2/M2 //NaCl
31 n3=m3/M3 //H2O
32 n=n1+n2+n3
33 N1=n1/n*100
34 N2=n2/n*100
35 N3=n3/n*100
36 printf("Mole percent of HCl is "+string(N1)+" , \
     nMole percent of NaCl is "+string(N2)+"and \nMole
      percent of H2O is "+string(N3)+".")
```

Scilab code Exa 2.14 Na2O percentage

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.14
10 // Page 24
11 printf("Example 2.14, Page 24 \n \n");
12
13 // solution
14
15 m = 100 //[kg] Lye (basis)
16 m1 = 73 //[kg] NaOH
```

```
17 M1 = 40 // NaOH
18 M2 = 62 // Na2O
19 p = (M2*m1)/(2*M1)
20 printf("percentage of Na2O in the solution is "+
      string(p)+".")
```

Scilab code Exa 2.15 TOC and ThOD

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.15
10 // Page 25
11 printf("Example 2.15, Page 25 \n \n");
12
13 // solution
14
15 //(CH2OH)3
16 M = 92 // molar mass of glycerin
17 C = 600 //[mg/l] glycerin conc.
18 TOC = (3*12/92)*600 //[mg/l]
19 // by combustion reaction we see 3.5 O2 is required
      for 1 mol of (CH2OH)3
20 ThOD = (3.5*32*600)/92 //[mg/l]
21 printf("TOC = "+string(TOC)+" mg/l\nThOD = "+string(
      ThOD)+" mg/l")
```

Scilab code Exa 2.16 conc of salts

```
clear;
clc;

// Stoichiometry
// Chapter 2
// Basic Chemical Calculations


// Example 2.16
// Page 25
printf("Example 2.16, Page 25 \n \n");

// solution

M1 = 100 // CaCO3
v1 = 2 // valence of CaCO3
eqm1 = M1/v1 // equivalent mass of CaCO3
M2 = 162 // Ca(HCO3)2
v2 = 2
eqm2 = M2/v2
m = 500 // [mg/l] CaCO3
C1 = (eqm2/eqm1)*m*.6 // [mg/l] conc. of Ca(HCO3)2
M3 = 146.3 // Mg(HCO3)2
v3 = 2
eqm3 = M3/v3
C2 = (eqm3/eqm1)*m*.4 //[mg/l] conc. of Mg(HCO3)2
printf("Actual concentration of Ca(HCO3)2 in the
   sample water is "+string(C1)+" mg/l and of Mg(
   HCO3)2 is "+string(C2)+" mg/l.")
```

Scilab code Exa 2.17 ppm unit

```
clear;
clc;

```

```
4 // Stoichiometry
5 // Chapter 2
6 // Basic Chemical Calculations
7
8
9 // Example 2.17
10 // Page 26
11 printf("Example 2.17, Page 26 \n \n");
12
13 // solution
14
15 S = .68 // sulphur content by mass
16 d = .85 // kg/l
17 s = (S*d*10^6)/100 // [mg/l] or [ppm]
18 printf("Sulphur content in LDO is "+string(s)+" ppm.
   ")
```

Scilab code Exa 2.18 molarity normality and molality

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 2
6 // Basic Chemical Calculations
7
8 // Example 2.18
9
10 // Page 26
11 printf("Example 2.18, Page 26 \n \n");
12
13 // solution
14
15 m1 = 100 //[kg] solution (basis)
16 m2 = 20 //[kg] NaCl
```

```
17 d = 1.127 //[kg/l]
18 V = m1/d // volume of 100 kg sol.
19 n = (m2/58.5)*100 // [mol] NaCl
20 M = n/V //[M]
21 v = 1 // valence of NaCl so,
22 N = M
23 m = n/(m1-m2) //[mol/kg]
24 printf("Molarity = "+string(M)+" M \nNormality = "+
   string(N)+" N \nMolality = "+string(m)+" mol/kg."
   )
```

Scilab code Exa 2.19 molarity of solution

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8  // Example 2.19
9
10 // Page 27
11 printf("Example 2.19, Page 27 \n \n");
12
13 // solution
14
15 m1 = 100 //[kg] TEA solution (basis)
16 m2 = 50 //[kg] TEA
17 M1 = 149 // molar mass of TEA
18 d = 1.05 //[kg/l]
19 V = m1/d // volume of 100 kg sol.
20 n = (m2/M1)*100 // [mol] NaCl
21 M = n/V //[M]
22 printf("Molarity of solution = "+string(M)+" M.")
```

Scilab code Exa 2.20 conc of CO2

```
clear;
clc;

// Stoichiometry
// Chapter 2
// Basic Chemical Calculations

// Example 2.20

// Page 27
printf("Example 2.20, Page 27 \n \n");

// solution

m1 = 100 //[kg] MEA solution (basis)
m2 = 20 //[kg] MEA
M1 = 61 // molar mass of MEA
n1 = m2/M1 // [kmol]
C = .206
n2 = C*n1 //[kmol] dissolved CO2
m3 = n2*44 // [kg] mass of CO2
n3 = (m1-m2-m3)/18 //[kmol] water
n = (n2/(n1+n2+n3))*100
m = (m3/100)*100
printf("Mass percent of CO2 = "+string(m)+" and Mol
    percent = "+string(n)+".")
```

Scilab code Exa 2.21 pH of HOCl

```
clear;
clc;

// Stoichiometry
// Chapter 2
// Basic Chemical Calculations

// Example 2.21

// Page 27
printf("Example 2.21, Page 29 \n \n");

// solution

//HOCl
Ma = .1 //molarity
Ka = 9.6*10^-7
C = (Ma*Ka)^.5 // conc. of H+ ions
pH = -log10(C)
printf("pH of the sol is "+string(pH)+".")
```

Scilab code Exa 2.22 Mavg and composition of air

```
clear;
clc;

// Stoichiometry
// Chapter 2
// Basic Chemical Calculations

// Example 2.22

// Page 39
printf("Example 2.22, Page 39 \n \n");

```

```
13 // solution
14
15 n = 100 // [mol] air (basis)
16 n1 = 21 //[mol] O2
17 n2 = 78 //[mol] N2
18 n3 = 1 //[mol] Ar
19 M1 = 31.9988 // O2
20 M2 = 28.0134 // N2
21 M3 = 39.948 // Ar
22 m1 = n1*M1
23 m2 = n2*M2
24 m3 = n3*M3
25 Ma = (m1+m2+m3)/n
26 printf("average molar mass of air is "+string(Ma)+"
   g.")
```

Scilab code Exa 2.23 Composition and specific gravity

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 2
6 // Basic Chemical Calculations
7
8 // Example 2.23
9
10 // Page 39
11 printf("Example 2.23, Page 39 \n \n");
12
13 // solution
14
15 // (a)
16 printf("(a) \n \n")
17 n = 100 // [kmol] cracked gas (basis)
```

```
18 n1 = 45 // methane
19 n2 = 10 // ethane
20 n3 = 25 // ethylene
21 n4 = 7 // propane
22 n5 = 8 // propylene
23 n6 = 5 // n-butane
24 M1 = 16
25 M2 = 30
26 M3 = 28
27 M4 = 44
28 M5 = 42
29 M6 = 58
30 m1 = n1*M1
31 m2 = n2*M2
32 m3 = n3*M3
33 m4 = n4*M4
34 m5 = n5*M5
35 m6 = n6*M6
36 m = m1+m2+m3+m4+m5+m6
37 M = m/n
38 printf("Average molar mass of gas is "+string(M)+" g
      .")
39 //(b)
40 printf("(b) \n \n")
41 // composition
42 p1 = (m1/m)*100
43 p2 = m2*100/m
44 p3 = m3*100/m
45 p4 = m4*100/m
46 p5 = m5*100/m
47 p6 = m6*100/m
48 printf("   GAS                Mass Percent \n Methane
                    "+string(p1)+" \n Ethane
                    "+string(p2)+" \n Ethylene
                    "+string(p3)+" \n Propane
                    "+string(p4)+" \n Propylene
                    "+string(p5)+" \n n-Butane
                    "+string(p6)+" \n \n \n")
```

```
49 // (c)
50 printf("(c) \n \n")
51 // specific gravity
52 g = M/28.97
53 printf("Specific gravity is "+string(g)+".")
```

Scilab code Exa 2.24 percentage error

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 2
6 // Basic Chemical Calculations
7
8
9 // Example 2.24
10 // Page 40
11 printf("Example 2.24, Page 40 \n \n");
12
13 // solution
14
15 p = 100 //[bar]
16 T = 623.15 //[K]
17 R = .083145
18 V = R*T/p // [l/mol] molar volume
19 v = V/18.0153 //
20 printf("Specific volume = "+string(v)+" m^3/kg.")
```

Scilab code Exa 2.25 molar volume

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

```
3
4 // Stoichiometry
5 // Chapter 2
6 // Basic Chemical Calculations
7
8
9 // Example 2.25
10 // Page 40
11 printf("Example 2.25, Page 40 \n \n");
12
13 // solution
14
15 p = 4 //[bar]
16 T = 773.15 //[K]
17 R = .083145
18 V = R*T/p // [l/mol] molar volume
19 printf("Molar volume = "+string(V)+" l/mol.\n \n \n"
     )
20  // using appendix III
21  // calculating Tc and Pc of different gases
       according to their mass fractions
22 Tc1 = .352*32.20 // H2
23 Tc2 = .148*190.56 // methane
24 Tc3 = .128*282.34 //ethylene
25 Tc4 = .339*132.91 // CO
26 Tc5 = .015*304.10 // CO2
27 Tc6 = .018*126.09 // N2
28 Tc = Tc1+Tc2+Tc3+Tc4+Tc5+Tc6 // Tc of gas
29  // similarly finding Pc
30 Pc1=.352*12.97
31 Pc2=.148*45.99
32 Pc3=.128*50.41
33 Pc4=.339*34.99
34 Pc5=.015*73.75
35 Pc6=.018*33.94
36 Pc=Pc1+Pc2+Pc3+Pc4+Pc5+Pc6 // Pc of gas
37 a = (27*R^2*Tc^2)/(64*Pc) // [bar/mol^2]
38 b = (R*Tc)/(8*Pc)  // l/mol
```

```
39 // substituting these values in vanderwall eq and
   solving by Newton Rhapson method we get
40 V = 15.74 // [l/mol]
41 printf("by Vanderwall eq molar volume = "+string(V)+
   " l/mol")
```

Scilab code Exa 2.26 ternary mix analysis

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.26
10 // Page 43
11 printf("Example 2.26, Page 43 \n \n");
12
13 // solution
14 m = 6.5065 //[g] mixture (basis)
15 Pv = 2.175 //[kPa] V.P. of water over KOH
16 Pa = 102.5-2.175 //[kPa] Partial P of n-butane and 1
   butene
17 V = 415.1*10^-3 //[l]
18 R = 8.314472
19 T = 296.4 //[K]
20 n = (Pa*V)/R*T // moles of butene and butane
21 n1 = n*.431 // n-butane
22 m1 = n1*58 // [g]
23 n2 = n-n1 // 1 butene
24 m2 = n2*56 //[g]
25 m3 = m-m1 // [g] furfural
26 n3 = m3/96
```

```
27 printf("component          mol percent          mass
   percent \nn-Butane          "+string(n1/n*100)+"
                "+string(m1/m*100)+" \n1-Butene
          "+string(n2/n*100)+"                "+
   string(m2/m*100)+" \nFurfural          "+string(n3/
   n*100)+"          "+string(m1/m*100)+"")
```

Scilab code Exa 2.27 vapour mix composition

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 2
6  // Basic Chemical Calculations
7
8
9  // Example 2.27
10 // Page 44
11 printf("Example 2.27, Page 44 \n \n");
12
13 // solution
14
15 P = 5.7+1.01 //[bar] absolute total P
16 // using Roult's law
17 vp = 3.293*.7737 //[kPa] vap P of furfural
18 // using Dalton's law of partial P
19 n1 = vp/(P*100) // mol fraction of furfural
20 n2 = 1-n1 // mol fraction of 1-butene
21 printf("mol fraction of Furfural is "+string(n1)+"\
   nmol fraction of 1-Butene is "+string(n2)+".")
```

Scilab code Exa 2.28 absolute humidity

```
clear;
clc;

// Stoichiometry
// Chapter 2
// Basic Chemical Calculations


// Example 2.28
// Page 44
printf("Example 2.28, Page 44 \n \n");

// solution

P = 100 //[kPa] total P
Pw = 2.5326 //[kPa] V.P> of water at dew point
//absolute humity = mass of water vapour/ mass of
   dry air
H = (Pw/(P-Pw))*(18.0153/28.9697) // absolute
   humidity
printf("absolute humidity = "+string(H)+".")
```

Scilab code Exa 2.29 nozzle outlet T

```
clear;
clc;

// Stoichiometry
// Chapter 2
// Basic Chemical Calculations


// Example 2.29
// Page 45
printf("Example 2.29, Page 45 \n \n");
```

```
12
13 // solution
14
15 //Ti-Tf = mu*(Pi-Pf)
16 Pi = 20.7 //[bar]
17 Pf = 8.7 // [bar]
18 mu = 1.616 //[K/bar]
19 Ti = 355.15 //[K]
20 Tf = Ti-mu*(Pi-Pf)
21 printf("Outlet temperature is "+string(Tf)+" K")
```

Chapter 3

Material Balances without Chemical Reaction

Scilab code Exa 3.1 Lancashire boiler

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 3
6  // Material Balances Without Chemical Reaction
7
8
9  // Example 3.1
10 // Page 60
11 printf("Example 3.1, Page 60 \n \n");
12
13 // solution
14
15 m = 1 //[kg] feed water
16 m1 = 1200 //[mg] dissolved solids in 1 kg feed water
17 m2 = 3500 //[mg] max dissolved solid content
18 x = (m*m1)/m2 // [kg] blown down water
19 printf("Percentage of feed water to be blown down is
```

```
    "+string(x)+".")
```

Scilab code Exa 3.2 Textile mill

```
clear;
clc;

// Stoichiometry
// Chapter 3
// Material Balances Without Chemical Reaction


// Example 3.2
// Page 61
printf("Example 3.2, Page 61 \n \n");

// solution
m = 100 //[kg] weak liquor (feed)
m1 = 4 //[kg] NaOH
p = .25
x = 4/p // water left
y = 100-16 // [kg] evaporated water
printf("Amount of water that evaporated is "+string(
    y)+" kg.")
```

Scilab code Exa 3.3 recovered tannin

```
clear;
clc;

// Stoichiometry
// Chapter 3
// Material Balances Without Chemical Reaction
```

```
7
8
9  // Example 3.3
10 // Page 61
11 printf("Example 3.3, Page 61 \n \n");
12
13 // solution
14
15 m = 100 //[kg] babul bark (basis)
16 m1 = 5.8 //[kg] moisture
17 m2 = 12.6 //[kg] Tannin
18 m3 = 8.3 //[kg] soluble non tannin organic material
19 m4 = m-m1-m2-m3 // [kg] Lignin
20 // lignin content remains unaffected during leaching
21 m5 = 100-.92-.65 // [kg lignin/kg dry residue]
22 x = (m4*100)/m5 // [kg]
23 T1 = x*.0092 //[kg] Tannin present in residue
24 T2 = m2 - T1 // [kg] Tannin recovered
25 T = (T2/m2)*100
26 printf("Percentage of Tannin recovered during
      leaching is "+string(T)+".")
```

Scilab code Exa 3.4 Extraction of dry neem leaves

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 3
6  // Material Balances Without Chemical Reaction
7
8
9  // Example 3.4
10 // Page 62
11 printf("Example 3.4, Page 62 \n \n");
```

```
12
13 // solution
14
15 m = 1 //[kg] dry neem leaves (basis)
16 m1 = .01/100 //[kg] beta cartene content of leaves
17 Ex = (m1*100)/.41 //[kg] extract quantity
18 Tc1 = Ex*.155 //[kg] Alpha Tocopherol in the extract
19 Tc2 = .46/100 //[kg] Alpha Tocopherol in the neem
   leaves
20 R = (Tc1*100)/Tc2 // recovery of Alpha Tocopherol
21 printf("(a) \n \nmass of extract phase per kg of dry
    leaves is "+string(Ex)+" kg \n \n \n(b) \n \
   npercent recovery of Alpha Tocopherol is "+string
   (R)+".")
```

Scilab code Exa 3.5 Extraction of mix of Acetone and Chloroform

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
9 // Example 3.5
10 // Page 62
11 printf("Example 3.5, Page 62 \n \n");
12
13 // solution
14
15 m= 100 //[kg] original mixture (basis)
16 A = 27.8 //[kg]
17 B = 72.2 //[kg]
18 // let x and y be uper and lower layer amounts
```

```
19 // total mixture = (x+y) kg
20 // balancing A and B
21 X = [.075 .203;.035 .673]
22 d = [27.8;72.2]
23 x = X\d
24 M = X(1,1)+X(2,1) // [kg] total mixture
25 Ms = M - m //[kg] mixed solvent
26 Mr = Ms/m // mixed solvent/original mixture
27 S1 = x(1,1)*.574+x(2,1)*.028 //[kg] water balance
28 S2 = x(1,1)*.316+x(2,1)*.096 //[kg] acetic acid
   balance
29 Qs = S1+S2
30 pS1 = (S1*100)/Qs
31 pS2 = 100-pS1
32 printf("(a) \n \nUpper layer = "+string(x(1,1))+" kg
    and Lower layer = "+string(x(2,1))+"\n \n \n(b)
   \n \nmass ratio of the mixed solvent to the
   original mixture is "+string(Mr)+" \n \n \n (c) \
   n \nwater mass percent = "+string(pS1)+" and
   acetic acid mass percent = "+string(pS2)+".")
```

Scilab code Exa 3.6 Pressure Swing Adsorption

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
9 // Example 3.6
10 // Page 63
11 printf("Example 3.6, Page 63 \n \n");
12
```

```
13 // solution
14
15 m = 170 //[Nm^3/h] air (basis)
16 m1 = 50*.99 //[Nm^3/h] N2 content of the stream
17 m2 = 50*.01 //[Nm^3/h]
18 N = m*.79-m1 // [Nm^3/h] N2
19 O = m*.21-m2 // [Nm^3/h] O2
20 V1 = N*100/(N+O)
21 V2 = O*100/(N+O)
22 printf("Vol percent of N2 is "+string(V1)+" and Vol
   percent of O2 is "+string(V2)+".")
```

Scilab code Exa 3.7 Required Oleum strength

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
9 // Example 3.7
10 // Page 64
11 printf("Example 3.7, Page 64 \n \n");
12
13 // solution
14
15 m = 100 //[kg] SO3 free mixed acid (basis)
16 m1 = 55 //[kg] HNO3
17 m2 = 45 //[kg] H2SO4
18 // SO3 + H2O --> H2SO4
19 m3 = (80/18)*3 //[kg] SO3 equivalent to 3 kg of
   water
20 Q = m2+m3 //[kg] oleum to be mixed
```

```
21 S = (m3/Q)*100 // strength of oleum
22 R = m1/Q
23 printf("Strength of Oleum required is "+string(S)+"
   \nHNO3 and Oleum are required to be mixed in the
   proportion of "+string(R)+":1.")
```

Scilab code Exa 3.8 Mixed acid formation

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 3
6  // Material Balances Without Chemical Reaction
7
8
9  // Example 3.8
10 // Page 64
11 printf("Example 3.8, Page 64 \n \n");
12
13 // solution
14
15 m = 1000 //[kg] mixed acid (basis)
16 // doing overall mass balance, H2SO4 balance and
   HNO3 balance
17 A = [1 1 1;.444 0 .98;.113 .9 0]
18 d = [1000;600;320]
19 x = A\d
20 printf("quantities of acids required are :\n  Spent
   = "+string(x(1,1))+"kg \n  HNO3 = "+string(x(2,1)
   )+" kg\n  H2SO4 = "+string(x(3,1))+" kg.")
```

Scilab code Exa 3.9 Actual analysis of borewell water

```
clear;
clc;

// Stoichiometry
// Chapter 3
// Material Balances Without Chemical Reaction


// Example 3.9
// Page 65
printf("Example 3.9, Page 65 \n \n");

// solution

l = 1 //[litre] water (basis)
Cl = 475.6 //[mg]
m1 = (58.5/35.5)*Cl //[mg] NaCl present in water
SO4 = 102.9 //[mg] // SO4
m3 = (142/96)*SO4 //[mg] Na2SO4 present in water
// carbonates are present due to Na2CO3
// eq mass of CaCO3 = 50
// eq mass of Na2CO3 = 53
m4 = (53/50)*65.9 // [mg] Na2CO3 present in water
// NaHCO3 in water = bicarbonates - temporary
   hardness
m5 = 390.6-384 // [mg] NaHCO3 present as CaCO3
m6 = (84/50)*m5 // [mg] NaHCO3 present in water
// equivalent mass of Mg(HCO3)2 = 73.15
m7 = (m6/50)*225
m8  = 384-225 //[mg] CaCO3 from Ca(HCO3)2
// equivalent mass of Ca(HCO3)2 is 81
m9 = (m8/50)*159 //[mg] Ca(HCO3)2 present in water
printf("Component analysis of raw water: \n \n \
   nCompound              mg/l \n \nCa(HCO3)2          "+
   string(m9)+" \nMg(HCO3)2          "+string(m7)+" \
   nNaHCO3              "+string(m6)+" \nNa2CO3
              "+string(m4)+" \nNaCl                "+
   string(m1)+" \nNa2SO4              "+string(m3)+"")
```

Scilab code Exa 3.10 Matrix use

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
9 // Example 3.10
10 // Page 67
11 printf("Example 3.10, Page 67 \n \n");
12
13 // solution
14
15 // see examples 3.5 and 3.8
```

Scilab code Exa 3.11 Flowrate calculation

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
9 // Example 3.11
10 // Page 68
11 printf("Example 3.11, Page 68 \n \n");
```

```
12
13 // solution
14
15 // basis : 1000 kg/h of feed
16 // balancing H2SO4, HNO3 and H2O in all the three
      product streams
17 M = [1 0 0 1 0 0 1 0 0;0 1 0 0 1 0 0 1 0;0 0 1 0 0 1
       0 0 1;1 0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0 0;0 0 1
      0 0 0 0 0 0;0 0 0 1 0 0 0 0 0;0 0 0 0 1 0 0 0 0;0
       0 0 0 0 1 0 0 0]
18 v = [400;100;500;4;94;60;16;6;400]
19 s = M\v
20 A = s(1)+s(2)+s(3)
21 B = s(4)+s(5)+s(6)
22 C = s(7)+s(8)+s(9)
23 printf("Flowrates are :\n A = "+string(A)+" kg/h \n
      B = "+string(B)+" kg/h\n C = "+string(C)+" kg/h")
```

Scilab code Exa 3.12 solving eqs with graphical plot

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
9 // Example 3.12
10 // Page 70
11 printf("Example 3.12, Page 70 \n \n");
12
13 // solution
14 m = 100  // kg
15 x = linspace(70,110,5);
```

```
16 y = linspace(100,115,4);
17 y1 = 27.8/.203 - .075*x/.203
18 y2 = 72.2/.673 - .035*x/.673
19 x = linspace(70,110,5);
20 plot(x,y1,style=4)
21 plot(x,y2,style=8)
22 // from graph its clear x = 93.4 kg and y = 102.4 kg
     .
23 x = 93.4;
24 y = 102.4;
25 M = x+y // [kg] total mixture
26 Ms = M - m //[kg] mixed solvent
27 Mr = Ms/m // mixed solvent/original mixture
28 S1 = x*.574+y*.028 //[kg] water balance
29 S2 = x*.316+y*.096 //[kg] acetic acid balance
30 Qs = S1+S2
31 pS1 = (S1*100)/Qs
32 pS2 = 100-pS1
33 printf("(a) \n \nUpper layer = "+string(x)+" kg and
     Lower layer = "+string(y)+"\n \n \n(b) \n \nmass
     ratio of the mixed solvent to the original
     mixture is "+string(Mr)+" \n \n \n (c) \n \nwater
      mass percent = "+string(pS1)+" and acetic acid
     mass percent = "+string(pS2)+".")
```

Scilab code Exa 3.14 ion exclusion process

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
```

```
// Example 3.14
// Page 73
printf("Example 3.14, Page 73 \n \n");

// solution

//using table 2.7 on page no 75
Rg = 8124*100/9448 // recovery of glycerine
Lg = (16+83)*100/9448 // loss of glycerine in waste
Reg = 100-Rg-Lg // recycle of glycerine
m1 = 238/8124 // NaCl in product
m2 = Rg*12/100 // glycerine in product
m3 = m1+m2 // total solute
n = m1*100/m3 // NaCl percent in total solute
printf("(a) \n \nrecovery percent of glycerine is "+
   string(Rg)+" \n \n \n(b) \n \npercent loss of
   glycerinr is "+string(Lg)+" \n \n \n(c) \n \
   nproduct contamination with respect to salt NaCl
   is "+string(n)+".")
```

Scilab code Exa 3.15 Air Conditioning plant

```
clear;
clc;

// Stoichiometry
// Chapter 3
// Material Balances Without Chemical Reaction


// Example 3.15
// Page 76
printf("Example 3.15, Page 76 \n \n");

// solution
```

```
14
15 f1 = 1.25 //[m^3/s] fresh ambient air as feed (basis
   )
16 f2 = 5.806 //[m^/s] air entering auditorium
17 v1 = 8.314*290/101.3 //[m^3/kmol] sp. vol. of moist
   air at 101.3 kPa and 290 K
18 na1 = f2*1000/v1 // [mol/s] molar flow rate of air
   entering auditorium
19 nw1 = 243.95*.0163/1.0163 // [mol/s]
20 na2 = 243.95 - nw1 //[mol/s] dry air flow
21 nw2 = 240.04*.0225 //[mol/s] moisture enterin air
   conditioning plant
22 // using table 3.8
23 m1 = (nw2-nw1) //[kg/h] moisture removed in a c
   plant
24 m2 = na2-.0181 //[mol/s] moisture in air leaving
   auditorium
25 m3 = (m2-nw1)*18 // [kg/h] moisture added in
   auditorium
26 Vm2 = 8.314*308/101.3 // [m^3/kmol]
27 na3 = (f1/25.28)*1000 //[mol/s]
28 n4 = 5.40-1.925 //[mol/s] moisture in recycle stream
29 mr = 240.04-47.525 //[mol/s] molar flow rate of wet
   recycle stream
30 R = mr/na3
31 printf("(a) \n \nmoisture removed in AC plant = "+
   string(m1)+"\n \n \n(b) \n \nmoisture added in
   auditorium = "+string(m3)+" \n \n \n(c) \n \
   nrecycle ratio of moles of air recycled per mole
   mole of fresh ambient air input = "+string(R)+"."
   )
```

Scilab code Exa 3.16 Overall efficiency of Pulp Mill

```
1 clear;
```

```
2 clc;
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
9 // Example 3.16
10 // Page 78
11 printf("Example 3.16 Page 78 \n \n");
12
13 // solution
14
15 // screen 1
16 // feed = N kg
17 // Oversize particle = NE1 kg
18 // Undersize particle = N-NE1
19
20 //screen 2
21 //feed = NE1+X kg
22 // Oversize particle = (NE1+X)*E2 kg
23 // Undersize particle = (NE1+X)(1-E2) kg
24
25
26 //screen 3
27 // feed = (NE1+X)*E2 kg
28 // Oversize particle = (NE1+X)*E2*E3 kg
29 // Undersize particle = (NE1+X)*E2*(1-E3) kg
30 printf("Overall Efficiency = (E1 E2 E3)*100/[(1-E1)
   (1-E2)+E2 E3].")
```

Scilab code Exa 3.17 2 stage membrane CO separation

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

```
3
4 // Stoichiometry
5 // Chapter 3
6 // Material Balances Without Chemical Reaction
7
8
9 // Example 3.17
10 // Page 79
11 printf("Example 3.17, Page 79 \n \n");
12
13 // solution
14
15 printf("(a) \n \n")
16 F = 5000 //[kmol/h] feed (basis)
17 m1 = F*.47 //[kmol/h] CO in F
18 m2 = F-m1 //[kmol/h] H2 in F
19 m3 = m1*.932 // CO in product stream
20 n2 = m3/.98 //[kmol/h]
21 printf("Flow rate of product stream is "+string(n2)+
   " kmol/h. \n \n \n(b) \n \n")
22 n2 = n2-m3 //[kmol/h] H2 in CO stream
23 printf("    Product H2 stream : \n  H2 = "+string(m2
   -n2)+" kmol/h \n  CO = "+string(m1-m3)+" kmol/h \
   n \n \n(c) \n \n")
24 nH2 = 2697.39 //[kmol/h]
25 nCO = 3000-nH2 // [kmol/h]
26 n4 = m2+nH2
27 n5 = m1+nCO
28 n6 = n4+n5
29
30 printf("    Composition of Mixed feed : \n  H2 = "+
   string(n4*100/n6)+" \n  CO = "+string(n5*100/n6)+
   "")
```

Scilab code Exa 3.18 2 stage reverse osmosis

```
clear;
clc;

// Stoichiometry
// Chapter 3
// Material Balances Without Chemical Reaction


// Example 3.18
// Page 79
printf("Example 3.10, Page 79 \n \n");

// solution

// Overall balance
// F=R1+P2
// Balance across Module I
// F+R2 = R1+P1 ==> R1+P2+R2 = R1+P1
// balance across module II
// P1 = P2+R2
P2 = 5 //[m^3/h]
P1 = P2/.8 //[m^3/h]
R2 = P1-P2 //[m^3/h]
F = P1/.66 - R2//[m^3/h]
R1 = F-P2 //[m^3/h]

// Overall balance of DS in water
xR1 = (F*4200-P2*5)/R1 //[mg/l]
xP1 = (P2*5)/(.015*P1) // [mg/l]
xR2 = (P1*xP1-P2*5)/R2 //[mg/l]
m1 = F*4200+R2*xR2 //[g] DS mixeed in MF
C1 = m1/(F+R2) // [mg/l]
m2 = R1*xR1 //[g] DS in R1
r = m2*100/m1 // rejection in module in I
m3 = m1-m2 //[g] DS in P1
C2 = m3/P1 // [mg/l]
R = R2/F
R1 = P2*100/F
```

```
39 printf("F = "+string(F)+" m^3/h \nR1 = "+string(R1)+
   " m^3/h \nP = "+string(P1+P2)+" m^3/h \nR2 = "+
   string(R2)+" m^3/h \nrecycle ratio = "+string(R)+
   " \nrejection percentage of salt in module I = "+
   string(r)+"")
```

Scilab code Exa 3.20 Purging by atmospheric pressure method

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 3
6  // Material Balances Without Chemical Reaction
7
8
9  // Example 3.20
10 // Page 86
11 printf("Example 3.20, Page 86 \n \n");
12
13 // solution
14
15 // concetration of the component after n times
      introduction of v volume of inert gas :
16 // Cn = Co/(1+1/n)^n
17 // we know limn-->infinity (1+1/n)^n = e
18 // therefore Cv = Co/e
```

Chapter 4

Material Balances Involving Chemical Reactions

Scilab code Exa 4.1 Manufacture of MCA

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 4
6  // Material Balances involving Chemical Reaction
7
8
9  // Example 4.1
10 // Page 116
11 printf("Example 4.1, Page 116 \n \n");
12
13 // solution
14
15 // basis one day operation
16 // Cl2 is the limiting component
17 n1 = 4536/71 //[kmol] Cl2 charged
18 // 1mol MCA requires 1 mol Cl2, so
19 n2 = 5000/94.5 // [kmol] Cl2 used for MCA production
```

```
20 // 1 mol DCA requires 2 mol of Cl2
21 n3 = 263*2/129 //[kmol] Cl2 used for DCA production
22 n4 = n2+n3 // total Cl2 used
23 a = n4*100/n1 // conversion %age
24 b = n2*100/n4 // yield % of MCA
25 s = n2/n3
26 printf("Percentage conversion = "+string(a)+" \n \
   nPercentage yield of MCA = "+string(b)+" \n \
   nselectivity of MCA = "+string(s)+".")
```

Scilab code Exa 4.2 Bechamp Process

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 4
6 // Material Balances involving Chemical Reaction
7
8
9 // Example 4.2
10 // Page 117
11 printf("Example 4.2, Page 117 \n \n");
12
13 // solution
14
15 m = 700 //[kg] ONT charged to reactor (basis)
16 m1 = 505*.99 // [kg] OT produced
17 m2 = (4*137*500)/(4*107)  //[kg] ONT required
18 m3 = m*.98 // [kg] ONT reacted
19 n1 = m1*100/m3 // yield of OT
20 m4 = (9*56*m)/(4*137) // [kg] theoretical iron
   reqiurement
21 m5 = 800*.9 //[kg] iron charged
22 E = (m5-m4)*100/m4 // excess iron
```

```
23 printf("(a) \n \nYield of OT = "+string(n1)+" \n \n
   \n(b) \n \nExcess quantity of iron powder = "+
   string(E)+".")
```

Scilab code Exa 4.3 Pilot Plant Calculations

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 4
6  // Material Balances involving Chemical Reaction
7
8
9  // Example 4.3
10 // Page 118
11 printf("Example 4.3, Page 118 \n \n");
12
13 // solution
14
15 printf("(a) \n \n")
16 m = 100 //kg chlorobenzene (basis)
17 m1 = 106.5*.655 //kg HNO3
18 m2 = 108*.936 // kg H2SO4
19 m3 = 106.5*.345 +108*.064 //kg water
20 M = m1+m2+m3
21 printf(" Analysis of charge: \n Component
      mass percent \n Chlorobenzene          "+string(m
      *100/M)+"\n HNO3                    "+string(m1
      *100/M)+" \n H2SO4                   "+string(m2
      *100/M)+" \n H2O                     "+string(m3
      *100/M)+" \n \n \n(b) \n \n")
22 // (b)
23 // total charge mass is constant
24 m4 = 314.5*.02 //[kg] unreacted CB in the product
```

```
m5 = 100-m4 // [kg] CB that reacted
c = m5*100/100 // conversion of CB
printf("Percent conversion of Chloro benzene is "+
    string(c)+" \n \n \n(c) \n \n")
// (c)
m6 = 63*c/112.5 //[kg] HNO3 consumed
m7 = m1-m6 // unreacted HNO3
m7 = 157.5*c/112.5 // [kg] total NCB produced
m8 = m7*.66 //[kg] p-NCB
m9 = m7*.34 //[kg] o-NCB
m10 = 18*c/112.5 //[kg] water produced
m11 = m10+m3 // total water in product
m12 = m4+m8+m9+m7+m2+m11
printf(" Composition of product stream : \n
    Component              mass percent \n CB
                          "+string(m4*100/m12)+" \n p-
    NCB                    "+string(m8*100/m12)+" \n o-
    NCB                    "+string(m9*100/m12)+" \n
    HNO3                     "+string(m7*100/m12)+" \n
    H2SO4                    "+string(m2*100/m12)+" \n
    H2O                      "+string(m11*100/m12)+"")
```

Scilab code Exa 4.4 Manufacturing of Acetaldehyde

```
clear;
clc;

// Stoichiometry
// Chapter 4
// Material Balances involving Chemical Reaction


// Example 4.4
// Page 119
printf("Example 4.4, Page 119 \n \n");
```

```
12
13 // solution
14
15 n=100 //[kmol] outgoing gas from 2nd scrubber
16 n1=.852*n //[kmol] N2
17 n2=21*n1/79 //[kmol] O2
18 n3=n2-2.1 // [kmol] reacted O2
19 // O2 balance
20 // O2 consumed in rxn (ii),(iii),(v) - O2 produced
     by rxn (iv) = 20.55 kmol
21 // let a,b,c be ethanol reacted (ii),(iii),(iv) and
     d be H2 reacted in (v)
22
23 // CO balance
24 a=2.3/2 //kmol
25
26 //CO2 balance
27 b = .7/2
28
29 //CH4 balance
30 c=2.6/2
31
32 //O2 balance
33 d = 41.1-a-3*b+c
34
35 //H2 balance
36 e = 7.1 +c+d //kmol (total H2 produced)
37 f = e-(3*b + 3*a) //kmol  (H2 produced in (i) =
     ethanol reacted in (i))
38 g = f+a+b+c // total ethanol reacted
39 h = 2*(n1+n2) // total ethanol entering
40 c1 = g*100/h
41 printf("(a) \n \n Conversion percent of ethanol = "+
     string(c1)+" \n \n \n")
42 y = f*100/g
43 printf("(b) \n \n Yield of acetaldehyde = "+string(y
     )+".")
```

Scilab code Exa 4.5 Lime Soda process

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 4
6  // Material Balances involving Chemical Reaction
7
8
9  // Example 4.5
10 // Page 121
11 printf("Example 4.5, Page 121 \n \n");
12
13 // solution
14
15 v = 1 //[l] water (basis)
16 // 1 mol (100mg) CaCO3 gives 1 mol (56) Cao
17 // use table 3.3 and eg 3.9
18 x = 56*390.6/100 //[mg/l] lime produced
19 printf("Amount of lime required = "+string(x)+" mg/l
   .")
```

Scilab code Exa 4.6 Manufacture of Ammonia by Fertilizer plant

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 4
6  // Material Balances involving Chemical Reaction
7
```

```
8
9  // Example 4.5
10 // Page 121
11 printf("Example 4.5, Page 121 \n \n");
12
13 // solution
14
15 m=100 //[kmol] (basis) dry mixed gas
16 // x = kmol of water gas
17 // y =kmol of producer gas
18 // overall material balance :
19 // x+y = 100   (i)
20
21 //r2 =.43x+.25y // H2 formed by shift rxn
22 //r2=.51x+.25y // H2 entering with water and
      producer gas
23 //r = r1+r2 // taoal H2
24 ///n =.02x+.63y // N2 entering
25 //N2:H2=1:3
26 // ==> x-1.807y = 0(ii)
27 //solving (i) and (ii)
28 A = [1 1;1 -1.807]
29 d = [100;0]
30 x = A\d
31 s = .43*x(1)+.25*x(2) // steam req.
32 printf("x = "+string(x(1))+" and y = "+string(x(2))+
      "\nAmount of steam required = "+string(s)+" kmol"
      )
```

Scilab code Exa 4.7 Saponification of Tallow

```
1 clear;
2 clc;
3
4 // Stoichiometry
```

```
// Chapter 4
// Material Balances involving Chemical Reaction


// Example 4.7
// Page 123
printf("Example 4.7, Page 123 \n \n");

// solution

m = 100//[kg] Tallow
m1 = 3*403*m/890 // [kg]
m2 = 92*m/890
printf("(a) \n \n NaOH required = "+string(m1)+" kg
   \n \n \n(b) \n \n amount of glycerine liberated =
    "+string(m2)+" kg.")
```

Scilab code Exa 4.8 Sulphur Burner

```
clear;
clc;

// Stoichiometry
// Chapter 4
// Material Balances involving Chemical Reaction


// Example 4.8
// Page 124
printf("Example 4.8, Page 124 \n \n");

// solution

n = 100//[kmol] SO3 free gas basis
n1 = 16.5 //[kmol] SO2
```

```
17 n2 = 3 //[kmol] O2
18 n3 = 80.5 //[kmol] N2
19 // S + O2 = SO2
20 // S + 3/2 O2 = SO3
21 n4 = (21/79)*80.5 //[kmol] O2 supplied
22 n5 = n4-n1-n2 // [kmol] Unaccounted O2
23 // O2 used in 2nd eq is m5
24 n6 = (2/3)*n5 //[kmol] SO3 produced
25 n7 = n1+n6 // sulphur burnt
26 m7 = n7*32 //[kg]
27 f1 = n6/n7 // fraction of SO3 burnt
28 // O2 req. for complete combustion of S = n7
29 n8 = n4-n7 //[kmol] excess O2
30 p1 = n8*100/n7 // %age of excess air
31 n9 = n4+n3 //[kmol/s] air supplied
32 F1 = n9*.3/n7 // air supply rate
33 v = 22.414*(303.15/273.15)*(101.325/100) //[m^3/kmol
   ] sp. vol of air
34 V1 = F1*v //[m^3/s] flow rate of fresh air
35 n10 = n+n7 //[kmol] total gas from burner
36 n11 = n10*.3/m7 // [kmol/s] gas req. for .3 kg/s S
37 V2 = 220414*1073.15*n11/273.15 // flowrate of burner
    gases
38 printf("(a) \n \n The fraction of S burnt = "+string
   (f1)+" \n \n \n(b) \n \n percentage of excess air
    over the amount req. for S oxidising to SO2 = "+
   string(p1)+" \n \n \n(c) \n \n volume of dry air
   = "+string(V1)+" m^3/s \n \n \n(d) \n \n volume
   of burner gases = "+string(V2)+" m^3/s.")
```

Scilab code Exa 4.9 Hydrogenation of Refined Soybean oil

```
1 clear;
2 clc;
3
```

```
4 // Stoichiometry
5 // Chapter 4
6 // Material Balances involving Chemical Reaction
7
8
9 // Example 4.9
10 // Page 125
11 printf("Example 4.9, Page 125 \n \n");
12
13 // solution
14
15 m = 100 //[kg] soya fatty acid (basis)
16 // use table 4.6
17 M1 = m/.3597 // M(avg) of soya fatty acid
18 //3 mol of fatty acid + 1 mol of glycerol = 1 mol
   triglyceride + 3 mol of water
19 M2 = M1*3+92.09-3*18.02 // Mavg of soyabean oil
20 q1 = M2*m/(M1*3) // soyabean oil per 100kg fatty
   acid
21 // based on reactions occuring
22 q2 = .0967+.1822*2+.0241*3 // kmol H2 req. per 100
   kg soya fatty acid
23 q3 = .5101 // kmol H2 req. per 100 kg soyabean oil
24 q4 = 11.434 // Nmˆ3/100kg soyabean oil
25 // x = linoleic acid converted to oleic acid
26 // y = oleic acid converted to stearic acid
27 q5 = 282.46*6.7/278.43 //
28 //q6 = 282.46*x/280.15 = 1.00717x [kg] oleic acid by
    linoleic acid
29 //q7 = 284.48*y/282.46 = 1.00715y [kg] stearic acid
   by oleic acid
30 //q8 = 100.097 + .00717x + .00715y  total fatty acid
31 //stearic balance : -.00105x + 1.00611y = 10.8142
      (i)
32 //linoleic balance : 1.0019x + .00019y = 48.4975
       (ii)
33 // solving (i) and (ii) we get
34 x = 48.5 //kg
```

```
35 y = 10.8 //kg
36 M3 = 100.52/.3596 // Mavg of fatty acid
37 H2req1 = .5334-.2864 // per 100kg fatty acid
38 H2req = 52.95 //Nmˆ3/t
39 I2s = 129.5 //kg I2 per 100 kg soyabean oil // for
      soyabean oil
40 I2h = 69.2 //kg I2 per 100 kg of fat
41 printf("(a) \n \n theoretical H2 required = "+string
      (q4)+" Nmˆ3/100kg soyabean oil \n \n \n(b) \n \n
      actual H2 required = "+string(H2req)+" \n \n \n(c
      ) \n \n Iodine value for soyabean oil = "+string(
      I2s)+". \n \n \n(d) \n \n Iodine value of
      hardened fat = "+string(I2h)+".")
```

Scilab code Exa 4.10 Material Balance in Formox Process

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 4
6  // Material Balances involving Chemical Reaction
7
8
9  // Example 4.10
10 // Page 128
11 printf("Example 4.10, Page 128 \n \n");
12
13 // solution
14
15 F1 = 4000 //kg/h methanol (basis)
16 F2 = F1/32 //kmol/h
17 F3 = F2/.084 //kmol/h gaseous mix flowrate
18 F4 = F2-F3 //kmol/h flow of wet air
19 n1 = .011*29/18 // kmol/kmol dry air
```

```
F5 = F4/(1+n1) // kmol/h dry air flowrate
O2 = F5*.21 //kmol/h
N2 = F5-O2 //kmol/h
Mreacted1 = F2*.99 //kmol/h
Munreacted1 = F2-Mreacted1 //kmol/h
// reaction (i)
Mreacted2 = Mreacted1*.9 //kmol/h
HCHOproduced1 = 111.375
O2consumed1 = 111.375/2
H2Oproduced1 = 111.375
// for rxn ii to iv
Mconsumed = Mreacted1*.1
//rxn (ii)
CH3OHreacted1 = Mconsumed*.71
O2consumed2 = 8.786*1.5
CO2produced = 8.786
H2Oproduced2 = 8.786*2
//rxn(iii)
CH3OHreacted2 = 12.375*.08
COproduced = .99
H2produced = 2*.99
//rxn(iv)
CH3OHreacted3 = 12.375*.05
CH4produced = .619
O2produced = .619/2
//rxn(v)
CH3OHreacted4 = 12.375-CH3OHreacted1-CH3OHreacted2-
   CH3OHreacted3
DMEproduced = 1.98/2
H2Oproduced3 = 1.98/2
O2 = 281.27-O2consumed1-O2consumed2+O2produced
H2O = 23.73+H2Oproduced1+H2Oproduced2+H2Oproduced3
printf("Composition of exit gas stream : \n \n CH3OH
    = "+string(Munreacted1)+" \n HCHO = "+string(
   HCHOproduced1)+" \n CO2 = "+string(CO2produced)+"
    \n CO = "+string(COproduced)+" \n H2 = "+string(
   H2produced)+" \n CH4 = "+string(CH4produced)+" \n
    (CH3)2O = "+string(DMEproduced)+" \n O2 = "+
```

```
    string(O2)+" \n N2 = "+string(N2)+" \n H2O = "+
    string(H2O)+".")
```

Scilab code Exa 4.11 Pyrites fines roasting

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 4
6  // Material Balances involving Chemical Reaction
7
8
9  // Example 4.11
10 // Page 132
11 printf("Example 4.11, Page 132 \n \n");
12
13 // solution
14
15 m = 100 //kg pyrites (basis)
16 //(a)
17 printf("(a) \n \n")
18 S1 = 42 //kg
19 i1 = 58 //kg  inerts
20 // 8 moll S = 3 mol O2 in Fe2O3
21 m1 = 3*32*42/8*32 //kg O2 converted to Fe2O3
22 m2 = i1+m1 // mass of SO3 free cinder
23 //2.3 kg S is in 100kg cinder
24 m3 = 100-(2.3*80/32)
25 m4 = (100/m3)*m2
26 m5 = m4*.023 //kg  S in cinder
27 p1 = 1.8*100/42
28 printf("percentage of cinder remained in cinder = "+
    string(p1)+". \n \n \n(b) \n \n")
29 //(b)
```

```
30 m6 = 100 //kmol SO3 free roaster gas (basis)
31 m7 = 7.12 //kmol O2 as SO2
32 m8 = 10.6 //O2
33 m9 = 100-m8-m7//N2
34 m10 = (21/79)*m9 // O2 entering roaster along N2
35 m11 = m7+m8+(3*7.12/8) // accounted O2
36 m12 = m10-m11 // unaccounted O2
37 m13 = (8/15)*m12 // SO3 formed
38 m14 = m13+m7 // S burnt
39 p2 = (m13/m14)*100
40 printf("percentage of S burnt to form SO3 = "+string
     (p2)+" \n \n \n(c) \n \n")
41 // (c)
42 // basis 100kg pyrite
43 m15 = 37.81/32 // SO2 formed
44 m16 = (m9+m10)*1.181/m7 // air supplied
45 // 4 kg pyrite is roasted
46 m17 = m16*4/100 //kmol/s  total air supplied
47 v1 = m17*24.957
48 printf("volumetric flow rate of air = "+string(v1)+"
      m^3/s \n \n \n(d) \n \n")
49 // (d)
50 m18 = (100.455*m17)/(m9+m10) // roaster gases
51 v2 = m18*66.386
52 printf("volumetric flow rate of roaster gases = "+
     string(v2)+" m^3/s \n\n\n(f) \n\n")
53 //(f)
54 m19 = 4.838*10^-2*.98 // SO3 absorbed in absorber
55 // SO3 + H2O = H2SO4
56 m20 = (m19*98*24*3600)/(.98*1000) //[t/d]
57 printf("Amount of 98 percent acid strength produced
     = "+string(m20)+" t/d.")
```

Scilab code Exa 4.12 Burning of Pyrites and ZnS

```
clear;
clc;

// Stoichiometry
// Chapter 4
// Material Balances involving Chemical Reaction


// Example 4.12
// Page 136
printf("Example 4.12, Page 136 \n \n");

// solution

// basis 100kg mixed charge = 75 kg pyrite + 25kg
   ZnS
// pyrites
m1 = 75*.92 //[kg]  FeS2
G1 = 75-m1 // gangue
// 4FeS2 + 11O2 = 2Fe2O3 + 8SO2
// 4FeS2 + 15O2 = 2Fe2O3 + 8SO3
//Zn ore
m2 = 25*.68 // ZnS
I1 = 25-m2 // inerts
// 2ZnS + 3 O2 = 2 ZnO + 2 SO2
I2 = I1+6 // total inerts
// new basis : 100kg cinder
m3 = 3.5*.7 // S as SO3
m4 = 3.5-m3 // S as FeS2
m5 = 100-m3-m4 // S free cinder
m6 = (81.4/97.4)*17 // ZnO
// FeS2 reacted = x
// (FeS2 in cinder/S free cinder) = (69-x)
   /(28.2+.667x) = 1.969/91.906
// solving this we get
x = 67.43 //kg
m7 = m6 + .667*x + 14 // S free cinder
m8 = 69-x // FeS2 in cinder
```

```
37 m9 = 6.125*m7/m5 // SO3
38 m10 = .667*x // Fe2O3
39 m11 = m6+m10+m8+m9+I2
40 printf("(a) \n \n Total cinder produced = "+string(
   m11)+"kg \n Composition of cinder : \n ZnO = "+
   string(m6)+"kg \n Fe2O3 = "+string(m10)+"kg \n S
   as FeS2 = "+string(m8)+"kg \n S as SO3 = "+string
   (m9)+"kg \n inerts = "+string(I2)+"kg \n \n \n(b)
    \n \n")
41 S1 = (64/120)*69 + (32/97.4)*17 //[kg]  S charged to
    burner
42 S2 = .035*79.63 // S in cinder
43 p = S2*100/S1
44 printf("percentage of S left in cinder = "+string(p)
   +"")
```

Scilab code Exa 4.13 Raising pH with NaOH

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 4
6  // Material Balances involving Chemical Reaction
7
8
9  // Example 4.13
10 // Page 138
11 printf("Example 4.13, Page 138 \n \n");
12
13 // solution
14
15 m1 = 1200*1.2 //[kg] mass of reactants
16 pOH1 = 14-6 //pOH of reactants
17 pOH = 14-9 //pOH of final mass
```

```
18 // ROWs = 1/sigma(Wi/ROWsi)
19 //Ms = mass of .5% NaOH required
20 //ROWs = density of final solution
21
22 //ROWs = 1/{((m1*10^3*1)/(((m1*10^3+Ms)*1.2)+(Ms/((
      m1*10^3+Ms)*1.005))}     (i)
23 //balance of OH- ions
24 //1200*10^-8 +Ms*10^-1.15/(1.005*10^-5) =
      (1200*1.2*10^3+Ms)*10^-5/ROWs*10^-5       (ii)
25 //solving (i) and (ii)
26 Ms = 170.21 //g
27 ROWs = 1.2016 //[kg/l]
28 printf("Mass of 0.5 percnt NaOH required to be added
      to raise the pH = "+string(Ms)+"g.")
```

Scilab code Exa 4.14 Solving eg 10 with Linear Model Method

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 4
6 // Material Balances involving Chemical Reaction
7
8
9 // Example 4.14
10 // Page 140
11 printf("Example 4.14, Page 140 \n \n");
12
13 // solution
14
15 // using equations of example 4.10
16
17 // soving 4.10 by linear model method
18 M = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0 0
```

```
    0 1 1 1 1 2;0 0 1 0 0 0 0 0 0 0 .5 1.5 0 -.5 0;0
    0 0 1 0 0 0 0 0 0 -1 -2 0 0 -1;0 0 0 0 1 0 0 0 0
    0 -1 0 0 0 0;0 0 0 0 0 1 0 0 0 0 0 -1 0 0 0;0 0
   0 0 0 0 1 0 0 0 0 0 -1 0 0;0 0 0 0 0 0 0 1 0 0 0
   0 -2 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0;0 0 0 0 0
    0 0 0 0 1 0 0 0 0 -1;0 0 0 0 0 0 0 0 0 0 1 1 1 1
    2;0 0 0 0 0 0 0 0 0 0 .5 1.5 0 -.5 0;0 0 0 0 0 0
    0 0 0 0 -1 -2 0 0 -1;0 0 0 0 0 0 0 0 0 0 0 0 0 1
    0;0 0 0 0 0 0 0 0 0 0 0 0 0 0 2]
V =
   [1058.1;125;281.27;23.73;0;0;0;0;0;.99*125;.2437*281.27;-5.4756
X = M\V
disp(X)
```

Scilab code Exa 4.15 Electrochemical cell

```
clear;
clc;

// Stoichiometry
// Chapter 4
// Material Balances involving Chemical Reaction


// Example 4.15
// Page 143
printf("Example 4.15, Page 143 \n \n");

// solution

// basis = 1.12 M63 O2 at NTP
m1 = 1.12*1000*32/22.4 //[g] O2
m2 = m1/8 // g eq  O2
//at cathode : Cu++ +2e = Cu
```

```
19 //at anode : SO4-- - 2e = SO4
20 eqwtCu = 63.5/2
21 depositedCu = eqwtCu*m2
22 E = (1130*18000)/96485//faradays  Total energy
      passed to cell
23 libCu = (1130*18000*eqwtCu)/96485 //[g]  theoritical
      liberation of Cu
24 eff = (depositedCu/libCu)*100 // current efficiency
25 printf("(a) \n \n Amount of Cu lberated = "+string(
      libCu)+" \n \n \n(b) \n \n Current efficiency of
      the cell = "+string(eff)+" percent.")
```

Scilab code Exa 4.16 Hooker type Diaphragm Cell

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 4
6  // Material Balances involving Chemical Reaction
7
8
9  // Example 4.16
10 // Page 144
11 printf("Example 4.16, Page 144 \n \n");
12
13 // solution
14
15 // basis : 1day operation
16 // NaCl = Na+ + Cl-
17 //H2O = H+ + OH-
18 //Na+ + OH- = NaOH
19 //H+ + e = (1/2)H2
20 //Cl- - e = (1/2)Cl2
21 E = (15000*3600*24)/96485 // faraday/day   Total
```

```
    energy passed thrrough cell
22 NaOH = (15000*3600*24*40)/(96485*1000) //[kg/day]
    theoretical NaOH
23 eff = (514.1/NaOH)*100 // current efficiency
24 Cl2 = (35.5/40)*514.1
25 H2 = (456.3*2)/(35.5*2)
26 //40 g NaOH = 58.5g NaCl
27 consNaCl = (58.5/40)*514.1 // NaCl consumed
28 Tliquor = 514.1/.11 //[kg/day]  total cell liquor
29 remNaCl = 514.1*1.4
30 totalNaCl = consNaCl+remNaCl
31 Fbrine = totalNaCl/.266 //feed rate of brine
32 consH2O = (18/40)*514.1
33 lossH2O = Fbrine-Tliquor-consH2O
34 printf("(a) \n \n Current efficiency of the cell = "
    +string(eff)+" percent. \n \n \n(b) \n \n Cl2
    produced = "+string(Cl2)+" kg/day \n H2 produced
    = "+string(H2)+" kg/day\n \n \n(c) \n \n loss of
    water = "+string(lossH2O)+" kg/day")
```

Scilab code Exa 4.17 Naptha Reforming to Ammonia

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 4
6 // Material Balances involving Chemical Reaction
7
8
9 // Example 4.17
10 // Page 146
11 printf("Example 4.17, Page 146 \n \n");
12
13 // solution
```

```
14
15 //M = mix feed rate, F = fresh feed rate , R =
      recycle stream
16 // using fig 4.3
17 // N2 balance
18 // a = 24.75M/(.25M+7.5M)      (i)
19 // P = (4.15M + 17.75a)/M      (ii)
20 // .585M -1.775a +(4.15M+17.75a)/M = 100   (iii)
21 //solving (i,) (ii), (iii)
22 M = 438.589 //[kmol/s]
23 a = (24.75*M)/((.25*M)+7.5) //kmol/s
24 P = (4.15*438.589+17.75*92.662)/M //kmol/s
25 R = M-100 // kmol/s
26 r = R/100 // recycle ratio
27 NH3 = (.585*M-2.275*a)*17.0305 //kg/s
28 printf("(a) \n \n recycle feed rate = "+string(R)+"
      kmol/s \n \n \n(b) \n \n purge gas rate = "+
      string(P)+" kmol/s \n \n \n(c) \n \n mass rate of
       NH3 = "+string(NH3)+" kg/s")
```

Scilab code Exa 4.18 Additional membrane separator in eg 17

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 4
6 // Material Balances involving Chemical Reaction
7
8
9 // Example 4.18
10 // Page 149
11 printf("Example 4.18, Page 149 \n \n");
12
13 // solution
```

```
14
15 // given
16
17 // (.1*M*R1)/(.415M+1.775a) + (.1125a*P)/(.415M +
   1.775a) + 1 = .1M
18 // R1*(.315M-1.225a)/(.415M + 1.775a) = .9M-4a
19 // M = 100 + R1 + (2.25a*p)/(.415M + 1.775a)
20 // .1M*P/(.415M + 1.775a) - (.1125a*P)/(.415M1.775a)
21
22 //solving them
23 M = 457.011 // kmol/s
24 R1 = 350.771 // kmol/s
25 P = 10.368 // kmol/s
26 a = 96.608 // kmol/s
27 R2 = 2.25*96.608*10.369/(.415*457.011 +
   1.775*96.608) // kmol/s
28 F = M -R1 - R2
29 printf("Mixed feed rate = "+string(M)+" kmol/s \
   nRecycle stream = "+string(R1)+" kmol/s \
   nRecovered H2 stream = "+string(R2)+" kmol/s \
   nFresh feed rate = "+string(F)+" kmol/s \nRecycle
    ratio = "+string((R1+R2)/F)+" kmol/kmol of fresh
    feed.")
```

Scilab code Exa 4.19 Partial Demineralisation Plant

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 4
6 // Material Balances involving Chemical Reaction
7
8
9 // Example 4.19
```

```
10 // Page 153
11 printf("Example 4.19, Page 153 \n \n");
12
13 // solution
14
15 m1 = (50/35.5)*312 //[mg/l]  Cl2 expressed as
      equivalent CaCO3
16 m2 = (50/48)*43.2 //[mg/l]  Sulphates as equivalent
      CaCO3
17 A = m1+m2 //[mg/l as CaCO3]    EMA in raw water
18 M1 = 550 // alkalinity of raw water
19 M2 = 50 // alkalinity of blend water
20 //let 100 l of raw water enters both ion exchangers
21 // balancing neutrilasion
22 x = 100*(M1-M2)/(A+M1) // raw water inlet to H2 ion
      echanger
23 printf(""+string(x)+" percent of total raw water is
      passed through the H ion exchanger.")
```

Scilab code Exa 4.20 Capacity increment by Second Reactor

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 4
6 // Material Balances involving Chemical Reaction
7
8
9 // Example 4.20
10 // Page 155
11 printf("Example 4.20, Page 155 \n \n");
12
13 // solution
14
```

```
15 m1 = 1488.1 //kmol/h gas mix to reactor1 (basis)
16 m2 = m1*.0625 // CH3OH
17 m3 = m1-m2 // ambient air flow
18 m4 = m3/1.01772 // dry air flow rate
19 m5 = m3-m4 // moisture
20 m6 = m2*.99 // CH3OH conversion in R1
21 m7 = m2-m6 // unreacted CH3OH
22 //rxn i
23 m8 = m7*.9 // CH3OH reacted = HCHO produced = H2O
      produced
24 m9 = m8/2 // O2 consumed
25 m10 = m6-m8 // CH3OH reacted in rxn ii to v
26 //rxn ii
27 m11 = m10*.71 // CH3OH reacted = CO2 produced
28 m12 = m11*1.5 // O2 consumed
29 m13 = 2*m11 // H2O produced
30 //rxn iii
31 m14 = m10*.08 // CH3OH reacted = CO produced
32 m15 = 2*m14 // H2 produced
33 //rxn iv
34 m16 = m10*.05 // Ch3OH reacted = CH4 produced
35 m17 = m16/2 // O2 produced
36 //rxn v
37 m18 = m10-m16-m14-m11 // CH3OH reacted
38 m19 = m18/2 // (CH3)2O = H2O produced
39
40 m20 = 287.87-m9-m12+m17 // O2 in R1 exit stream
41 m21 = m5+m8+m13+m19 // H2O in R1
42 m = m7+m8+m11+m14+m15+m16+m19+m20+1082.93+m21
43 // R2
44 // x kmol/h CH3OH is added b/w reactors
45 // (m7+x)/(m+x) = .084 solving it
46 x = 140.548 //[kmol/h]
47 m22 = x+m7 // CH3OH entering R2
48 m23 = m22*.99 //CH3OH reacted
49 m24 = m22-m23 // CH3OH unreacted
50 //rxn i
51 m25 = m23*.9 // CH3OH reacted = HCHO produced = H2O
```

```
    produced
m26 = m25/2 // O2 consumed
m27 = m23 - m25 // CH3OH reacted in rxn ii to v
//rxn ii
m28 = m27*.71 // CH3OH reacted = CO2 produced
m29 = m28*1.5 // O2 consumed
m30 = m28*2 // H2O produced
//rxn iii
m31 = m27*.08 // CH3OH reacted = CO produced
m32 = m31*2 // H2 produced
//rxn iv
m33 = m27*.05 // Ch3OH reacted = CH4 produced
m34 = m33/2 // O2 produced
//rxn v
m35 = m27-m28-m31-m33 // CH3OH reacted
m36 = m35/2 // (CH3)2O = H2O produced

m37 = m20 - m26-m29+m34 // O2 in R2 exit stream
m38 = m21+m25+m36 // H2O in R2
m39 = 92.07+m25 // HCHO in R2
m40 = m24+m39+m28+m31+m32+m33+m36+m37+m38+1082.93

m41 = m39*30 // kg/h   HCHO produced
m42 = m41/.37 // bottom sol floe rate
c = (m42-9030.4)*100/9030.4 // increase in capacity
printf("Increase in capacity = "+string(c)+" percent
    .")
```

Scilab code Exa 4.21 Blast Furnace Calculations

```
clear;
clc;

// Stoichiometry
// Chapter 4
```

```
6  // Material Balances involving Chemical Reaction
7
8
9  // Example 4.21
10 // Page 159
11 printf("Example 4.21, Page 159 \n \n");
12
13 // solution
14
15 // basis 1 tonne of pig iron
16 coke = 1000 //kg
17 flux = 400 //kg
18 Fe1 = 1000*.95 // Fe in pig iron
19 Fe2 = (112/160)*.8 // Fe available per kg of ore
20 ore = Fe1/Fe2 // kg
21 Si = .014*1000 //kg  //Si in pig iron
22 si1 = (60/28)*14 // silica present in pig iron
23 si2 = ore*.12 // silica in ore
24 si3 = .1*coke // silica in coke
25 si4 = si2+si3-si1 // silica in slag
26 alumina = ore*.08 // Al2O3 in ore = Al2O3 in slag
27 CaO = flux*(56/100)
28 slag = si4+alumina+CaO
29 printf("(a) \n \n Mass of slag made = "+string(slag)
      +" kg. \n \n \n(b) \n \n Mass of ore required = "
      +string(ore)+" kg. \n \n \n(c) \n \n Composition
      of slag : \n SiO2 = "+string(si4)+" kg \n Al2O3 =
       "+string(alumina)+" kg \n CaO = "+string(CaO)+"
      kg.  \n \n \n(d) \n \n")
30 C = .9*coke+(12/100)*flux-36 // total C available
31 // CO:CO2 = 2:1
32 C1 = C/3 // C converted to CO2
33 C2 = 2*C/3 // C converted to CO
34 O21 = C1*(32/12)+C2*(16/12) // O2 required for CO
      and CO2 formation
35 O22 = (32/28)*Si // O2 from SiO2
36 O23 = ore*(.8*48/160) // O2 from Fe2O3
37 O24 = flux*(32/100) // O2 from CaCO3
```

```
38 O25 = O21-O22-O23-O24 //kg O2 to be supplied
39 O26 = O25/32 //kmol
40 air = O26/.21 //kmol
41 V = air*22.414 //m^3
42 printf(" Volume of air to be supplied = "+string(V)+
     " m^3.")
```

Chapter 5

Energy Balances

Scilab code Exa 5.1 Pumping of water

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.1
10 // Page 186
11 printf("Example 5.1, Page 186 \n \n");
12
13 // solution
14
15 // basis pumping of 1 l/s of water
16 Hadd = 52  // kW
17 Hlost = 21  // kW
18 fi = Hadd - Hlost  // kW
19 p1 = 101325  // Pa
20 p2 = p1
21 Z1 = -50  // m
```

```
Z2 = 10  // m
g = 9.80665  // m/s sq
gc = 1  // kg.m/(N.s sq)
row = 1  // kg/l
W = 1.5*.55  // kW
// energy balance b/w A and B
// dE = E2-E1 = W + Q + (Z1-Z2)*(g/gc)*qm
dE = 31.237 // kW
printf("Increase in internal energy between the
   storage tank and the bottom of the well = "+
   string(dE)+" kW.")
```

Scilab code Exa 5.2 Heating of CH4

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.2
// Page 197
printf("Example 5.2, Page 197 \n \n");

// solution

// using table 5.1
// basis 1 kmol of methane
T1 = 303.15  // K
T2 = 523.15  // K
// using eq 5.17
H = 19.2494*(T2-T1) + 52.1135*10^-3*(T2^2-T1^2)/2 +
   11.973*10^-6*(T2^3-T1^3)/3 - 11.3173*(T2^4-T1^4)
```

```
    *10^-9/4   // kJ
printf(" Heat added = "+string(H)+" kJ/kmol methane.
    ")
```

Scilab code Exa 5.3 Calculation of heat added

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.3
// Page 198
printf("Example 5.3, Page 198 \n \n");

// solution

// basis 1 kmol methane at 25 bar
Pc = 46.04   // bar
Tc = 190.5   // K
Pr = 25/Pc
// H-Ho = intgr(from303.15 to 523.15){CmpR dT}
// solving it by simpson's rule
HE = 255.2   // kJ/kmol
H = 9175.1+HE
printf(" Heat added = "+string(H)+" kJ/kmol of
    methane.")
```

Scilab code Exa 5.4 Heating of Toulene

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.4
// Page 206
printf("Example 5.4, Page 206 \n \n");

// solution

// using table 5.3
// .25 kg/s toulene heated from 290.15K to 350.15K
qm = .25/92  // kmol/s
// reference 7
fi = 2.717*10^-3*[1.8083*(350.15-290.15) +
   812.223*10^-3*(350.15^2-290.15^2)/2 -
   1512.67*10^-6*(350.15^3-290.15^3)/3 +
   1630.01*10^-9*(350.15^4-290.15^4)/4]
printf(" Heat required to be added to toulene = "+
   string(fi)+" kW.")
```

Scilab code Exa 5.5 Aq caustic soda heating

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


```

```
// Example 5.5
// Page 206
printf("Example 5.5, Page 206 \n \n");

// solution

// basis 1kg of 20% NaOH sol
// referring to fig 5.4
C11 = 3.56  // kJ/kg.K   at 280.15K
C12 = 3.71  // kJ/kg.K   at 360.15K
C1m = (C11+C12)/2
H = 1*C1m*(360.15-280.15)  // kJ
printf(" Heat required to be added = "+string(H)+"kJ
   .")
```

Scilab code Exa 5.6 Heating Chlorinated diphenyl

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.6
// Page 207
printf("Example 5.6, Page 207 \n \n");

// solution

// basis 1kg Diphyl A-30
Q = .7511*(553.15-313.15) +
   1.465*10^-3*(553.15^2-313.15^2)/2   // kJ/kg
fi = Q*4000  // kJ/h    for mass flowrate 4000 kg/h
```

```
Clm = (1.1807+1.5198)/2
fi1 = Clm*(553.15-313.15)*4000/3600  // kJ/h
err = (fi1-Q)*100/Q
printf(" Heat to be supplied = "+string(fi1)+" kW \n
    Percent error = "+string(err)+".")
```

Scilab code Exa 5.7 Roasting of pyrites fine

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.7
// Page 208
printf("Example 5.7, Page 208 \n \n");

// solution

T1 = 298.15 // K
T2 = 775.15  //K
// using eq 5.17
Q = 28.839*(T2-T1)+2.0395*10^-3*(T2^2-T1^2)/2 +
   6.9907*10^-6*(T2^3-T1^3)/3 - 3.2304*10^-9*(T2^4-
   T1^4)/4  // kJ/kmol
printf(" Heat content of 1 kmol of gas mixture at
   298K = "+string(Q)+" kJ/kmol.")
```

Scilab code Exa 5.8 Anniline and water mix subcooled

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.8
// Page 210
printf("Example 5.8, Page 210 \n \n");

// solution

// basis 8000 kg/h mixture is to be cooled
qn1m = .118*8000     // kg/h
qn1 = qn1m/93.1242   // kmol/h
qn2m = 8000-qn1m      // kg/h
qn2 = qn2m/18         // kmol/h
T1 = 373.15    //K
T2 = 313.15    //K
fi = qn1*[206.27*(T1-T2)-211.5065*10^-3*(T1^2-T2^2)
   /2+564.2902*10^-6*(T1^3-T2^3)/3] + qn2*[50.845*(
   T1-T2)+213.08*10^-3*(T1^2-T2^2)/2-631.398*10^-6*(
   T1^3-T2^3)/3+648.746*10^-9*(T1^4-T2^4)/4]   // kJ/
   h
printf(" Heat removal rate of subcooling zone of the
    condenser = "+string(fi)+" kJ/h.")
```

Scilab code Exa 5.9 Vapor Pressure calculations

```
clear;
clc;

// Stoichiometry
```

```
// Chapter 5
// Energy Balances


// Example 5.9
// Page 220
printf("Example 5.9, Page 220 \n \n");

// solution

// (a)
T = 305.15 //K
Pv1 = 10^(4.0026-(1171.530/(305.15-48.784)))  // bar
// (b)
T = 395.15
Pv2 = 10^(3.559-(643.748/(395.15-198.043)))   // bar
printf(" (a) \n \n V.P. of n-hexane at 305.15K = "+
   string(Pv1)+" bar. \n \n \n (b) \n \n V.P. of
   water at 395.15K = "+string(Pv2)+" bar.")
```

Scilab code Exa 5.10 Calculations on O zylene

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.10
// Page 225
printf("Example 5.10, Page 225 \n \n");

// solution
```

```
14
15 // (a)
16 Pc = 3732  // kPa
17 Tc = 630.3  // K
18 Tb = 417.6  //K
19 TBr = Tb/Tc
20 lambdav = 8.314472*417.6*(1.092*(log(3732)-5.6182)
      /(.930-.6625))
21 // (b)
22 T1 = 298.15  //K
23 lambdav1 = 36240*[(630.3-298.15)/(630.3-417.6)]^.38
24 printf(" (a) \n \n Latent heat of vaporization at Tb
       using Riedel eq is "+string(lambdav)+" kJ/kmol.
      \n \n \n (b) \n \n Latent heat of vaporizaation
      at 298.15 K using Watson eq is "+string(lambdav1)
      +" kJ/kmol.")
```

Scilab code Exa 5.11 latent heat of vaporization of ethanol

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.11
10 // Page 225
11 printf("Example 5.11, Page 225 \n \n");
12
13 // solution
14
15 // (a)
16 Pc = 61.37 // bar
```

```
17 Tc = 514  //K
18 Tb = 351.4
19 P = 1 // atm
20 TBr = Tb/Tc
21 // Riedel eq
22 lambdav1 = 8.314472*Tb*1.092*(log(6137)-5.6182)
      /(.930-TBr)
23 // NIST eq
24 lambdav2 = 50430*exp(-(-.4475*TBr))*(1-TBr)^.4989
25 // (b)
26 T1 = 298.15
27 TBr1 = T1/Tc
28 // Watson eq
29 lambdav21 = 38563*[(514-298.15)/(514-351.4)]^.38
30 // NIST eq
31 lambdav22 = 50430*exp(-(-.4475*TBr1))*(1-TBr1)^.4969
32 printf(" (a) \n \n Latent heat of vaporization at Tb
       using \n Riedel eq is "+string(lambdav1)+" kJ/
      kmol \n NIST eq is "+string(lambdav2)+" kJ/kmol \
      n \n \n (b) \n \n Latent heat of vaporization at
      298.15 K using \n Watson eq is "+string(lambdav21
      )+" kJ/kmol \n NIST eq is "+string(lambdav22)+"
      kJ/kmol")
```

Scilab code Exa 5.12 Saturation P of steam

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.12
```

```
10 // Page 227
11 printf("Example 5.12, Page 227 \n \n");
12
13 // solution
14
15 // using Appendix IV.2
16 Ps1 = 75
17 Ps2 = 80
18 T1 = 563.65
19 T2 = 568.12
20 T = 565.15
21 Ps = 75*exp((T2*(T-T1)*log(80/75)/(T*(T2-T1))))
22 printf(" Saturation Pressure of steam at 565.15K is
   "+string(Ps)+" bar.")
```

Scilab code Exa 5.13 Bubble and Dew pt calculations

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.13
10 // Page 236
11 printf("Example 5.13, Page 236 \n \n");
12
13 // solution
14
15 // basis  1 kmol equimolar mix
16 npent = .5 // kmol
17 nhex = .5 // kmol
18 P = 101.325  // kPa
```

```
19 x1 = .5
20 x2 = x1
21 Ts1 = 309.2 // K
22 Ts2 = 341.9 // K
23 T1 = (Ts1+Ts2)/2
24 // using these data, we get table 5.10 and 5.11
25 Tbb = 321.6  //K
26 Tdp = 329.9  //K
27 printf(" Bubble point = "+string(Tbb)+" K and \n Dew
       point = "+string(Tdp)+" K.")
```

Scilab code Exa 5.14 Hot air drying machine

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.14
10 // Page 237
11 printf("Example 5.14, Page 237 \n \n");
12
13 // solution
14
15 // basis 1000 kg/h of condensate at the saturation
      temperature corresponding to 8 bar a
16 // using Appendix IV.2
17 H = 720.94 // kJ/kg
18 Hm = 419.06 // kJ/kg
19 x = poly(0,'x')
20 condensate = 1000-x
21 Hcondensate1 = 1000*H
```

```
22 Hcondensate2 = condensate*419.06
23 Ht = x*2676
24 p = Hcondensate2+Ht-Hcondensate1
25 printf(" The quqntity of flash steam produced = "+
     string(roots(p))+" kg/h.")
```

Scilab code Exa 5.15 Flow of saturated vapors of R134

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.15
10 // Page 238
11 printf("Example 5.15, Page 238 \n \n");
12
13 // solution
14
15 qv1 = 50  // l/s
16 qm = qv1*1.08  // kg/s
17 fi = qm*3.08*(263.15-258.15) // kW
18 lv = 384.19-168.7  // kJ/kg
19 qm2 = fi/lv
20 H = 256.35  // kJ/kg
21 x = poly(0, 'x')
22 p = H*(qm2+x) - 168.7*qm2-x*384.19
23 a = qm2+roots(p)
24 printf(" Flow of vapor from he chiller = "+string(a)
     +" kg/s.")
```

Scilab code Exa 5.16 Liquifaction of Cl2

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.16
// Page 238
printf("Example 5.16, Page 238 \n \n");

// solution

// basis liquifaction capacity = 0.116 kg/s
p1 = 101 // kPa
Ts1 = 239.15
lv1 = 288.13 // kJ/kg
p2 = 530 // kPa
Ts2 = 290.75 // K
lv2 = 252.93 // kJ/kg
// referring to table 5.3 and using eq 5.21
H1 = -39.246*(Ts2-Ts1)+1401.223*10^-3*(Ts2^2-Ts1^2)
   /2-6047.226*10^-6*(Ts2^3-Ts1^3)/3+8591.4*10^-9*(
   Ts2^4-Ts1^4)/4  // kJ/kmol
T3 = 313.15
H2 = [28.5463*(T3-Ts1)+23.8795*10^-3*(T3^2-Ts1^2)
   /2-21.3631*10^-6*(T3^3-Ts1^3)/3+6.4726*10^-9*(T3
   ^4-Ts1^4)/4]/70.903  // kJ/kg
fi2 = .116*H2
Cl2evp = fi2/lv1  // kg/s
Cl2recy = Cl2evp/(1-.185)
```

```
29 R = Cl2recy/.116 // kg/kg fresh feed
30 // T4/T1 = (p2/p1)^[(gamma-1)/gamma]
31 gm = 1.355
32 p22 = 326.3
33 p21 = 101
34 T4 = Ts1*(p2/p1)^[(gm-1)/gm]
35 T5 = 313.15
36 fi3 = 1.88*10^-3*(343.1+91.6-26.2+2.5)  // kW
37 Fwater1 = fi3/(8*4.1868)  // kg/s
38 // similarly
39 T6 = 379.9
40 fi4 = 1.88*10^-3*[28.5463*(T6-T5)+23.8795*10^-3*(T6
      ^2-T5^2)/2-21.3631*10^-6*(T6^3-T5^3)
      /3+6.4726*10^-9*(T6^4-T5^4)/4]  // kW
41 Fwater2 = fi4/(8*4.1868)  // kg/s
42 Wreq = Fwater1+Fwater2
43 fi5 = 1.88*10^-3*[28.5463*(T5-Ts2)+23.8795*10^-3*(T5
      ^2-Ts2^2)/2-21.3631*10^-6*(T5^3-Ts2^3)
      /3+6.4726*10^-9*(T5^4-Ts2^4)/4] +.1333*252.93 //
      kW
44 printf(" (a) \n \n Recycle ratio = "+string(R)+" kg
      Cl2/kg fresh feed \n \n \n (b) \n \n Cooling
      water required at \n interface = "+string(Fwater1
      )+" kg/s \n after cooler = "+string(Wreq)+" kg/s
      \n \n \n (c) \n \n Refrigiration load of chiller
      = "+string(fi5)+" kW.")
```

Scilab code Exa 5.17 Melting of Tin

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
```

```
7
8
9  // Example 5.17
10 // Page 242
11 printf("Example 5.17, Page 242 \n \n");
12
13 // solution
14
15 // basis 100 kg of tin
16 T1 = 303.15
17 T2 = 505.15
18 n = 100/118.7  // kmol
19 // Q1 = n*[intgr from T1 to T2 (Cms dT)]
20 Q1 = 4973.3 // kJ
21 lf = 7201
22 Q2 = n*lf  // kJ
23 Q = Q1+Q2
24 lv = 278 // kJ/kg
25 vp = Q/lv // kg
26 printf(" Quantity of eutectic mixture condensed = "+
     string(vp)+" kg per 100 kg of tin melted at its
     melting point.")
```

Scilab code Exa 5.18 steam fluctuation calculations

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.18
10 // Page 243
```

```
11 printf("Example 5.18, Page 243 \n \n");
12
13 // solution
14
15 Ts1 = (438.2+436)/2
16 Ta = 300
17 fi1 = .045*(Ts1-Ta)*3600
18 theta1 = 307293/fi1  //h
19 Ts2 = (436+434)/2
20 fi2 = .045*(Ts2-Ta)*3600
21 theta2 = 302415/fi2
22 Ts3 = (434+432.1)/2
23 fi3 = .045*(Ts3-Ta)*3600
24 theta3 = 313859/fi3
25 theta = theta1+theta2+theta3
26 printf(" total time required = "+string(theta)+" hrs
   .")
```

Scilab code Exa 5.19 Manufacture of dry ice

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.19
10 // Page 245
11 printf("Example 5.19, Page 245 \n \n");
12
13 // solution
14
15 H1 = 482.9  // kJ/kg
```

```
16 H2 = 273.4
17 fi1 = 100*(H1-H2) // kJ/h
18 T1 = 313.15
19 T2 = 403.15
20 fi11 = 21.3655*(T2-T1)+64.2841*10^-3*(T2^2-T1^2)
      /2-41.0506*10^-6*(T2^3-T1^3)/3+9.7999*10^-9*(T2
      ^4-T1^4)/4  // kJ/h
21 // at 20 MPa
22 h1 = 211.1
23 Ts = 277.6
24 H11 = 427.8
25 x = poly(0, 'x')
26 p = x*h1+(100-x)*H11-100*H2
27 a = roots(p)
28 fi2 = (100-a)*(H11-h1)  // kJ/h
29 h2 = -148.39
30 H3 = 422.61
31 y = poly(0, 'y')
32 p1 = 100*176.18-(100-y)*H3+h2*y
33 b = roots(p1)
34 fi3 = 100*(h1-176.8)
35 H = fi3+24021
36 H4 = H/(100-43.16)
37 // from ref 23
38 T = 262.15
39 printf(" (a) \n \n Yield of dry ice = "+string(b)+"
      kg. \n \n \n (b) \n \n Percent liquifaction = "+
      string(a)+". \n \n \n (c) \n \n Temp of vented
      gas = "+string(T)+" K.")
```

Scilab code Exa 5.20 Steam produced in S burner

```
1 clear;
2 clc;
3
```

```
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.20
10 // Page 247
11 printf("Example 5.20, Page 247 \n \n");
12
13 // solution
14
15 // basis 200 kg/h of Sulphur firing
16 F = 200/32  // kmol/h
17 O2req = 6.25*1.1
18 airin = O2req/.21
19 N2in = airin-O2req
20 T1 = 1144.15
21 T2 = 463.15
22 fi = 788852.2 // kJ/h
23 H = 15*4.1868+1945.2
24 qm = fi*.9/2008  // kg/h
25 printf(" Amount of steam produced = "+string(qm)+"
   kg/h.")
```

Scilab code Exa 5.21 Equimoar pentane and hexane mix

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.21
```

```
10 // Page 248
11 printf("Example 5.21, Page 248 \n \n");
12
13 // solution
14
15 // enthalpy at Tbb
16 Tbb = 321.6
17 T1 = 298.15
18 H1 = 65.4961*(Tbb-T1)+628.628*10^-3*(Tbb^2-T1^2)
   /2-1898.8*10^-6*(Tbb^3-T1^3)/3+3186.51*10^-9*(Tbb
   ^4-T1^4)/4  // kJ/kmol
19 H2 = 31.421*(Tbb-T1)+976.058*10^-3*(Tbb^2-T1^2)
   /2-2353.68*10^-6*(Tbb^3-T1^3)/3+3092.73*10^-9*(
   Tbb^4-T1^4)/4  // kJ/kmol
20 Hsol = (H1+H2)/2  // kJ/kmol
21 // enthalpy at Tdp
22 lv1 = 25790*((469.7-329.9)/(469.7-309.2))^.38
23 lv2 = 28850*((507.6-329.9)/(507.6-341.9))^.38
24 Tdp = 329.9
25 H21ig = 65.4961*(Tdp-T1)+628.628*10^-3*(Tdp^2-T1^2)
   /2-1898.8*10^-6*(Tdp^3-T1^3)/3+3186.51*10^-9*(Tdp
   ^4-T1^4)/4 + lv1 // kJ/kmol
26 H22ig = 31.421*(Tdp-T1)+976.058*10^-3*(Tdp^2-T1^2)
   /2-2353.68*10^-6*(Tdp^3-T1^3)/3+3092.73*10^-9*(
   Tdp^4-T1^4)/4 +lv2 // kJ/kmol
27 Hmixig = (H21ig+H22ig)/2
28 printf(" (a) \n \n H = "+string(Hsol)+" kJ/kmol \n \
   n \n (b) \n \n H = "+string(Hmixig)+" kJ/kmol")
```

Scilab code Exa 5.22 Flashing of saturated liq mix

```
1 clear;
2 clc;
3
4 // Stoichiometry
```

```
// Chapter 5
// Energy Balances


// Example 5.22
// Page 252
printf("Example 5.22, Page 252 \n \n");

// solution

H1 = 23549  //kJ/kmol
H2 = 16325
H3 = 28332
H4 = .4*H2+.6*H3
printf("Enthalpy of vapor-liquid mixture after
   flashing = "+string(H4)+" kJ/mol.")
```

Scilab code Exa 5.23 H2 recovery from Refinery off gases

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.23
// Page 253
printf("Example 5.23, Page 253 \n \n");

// solution

// basis feed gas = 12000 Nm^3 = 535.4 kmol/h
T1 = 147.65  // K
```

```
17 n1 = 535.4*.3156  // kmol/h  HP tail gas stream
18 T = 118.5 // K
19 n2 = (535.4-n1)*.0602 // kmol/h    LP tail stream
20 n3 = 535.4-n2-n1 // kmol/h  product H2 stream
21 p = 315.35*100/n3
22 printf(" Purity of product H2 stream = "+string(p)+"
       percent.")
```

Scilab code Exa 5.24 Refrigiration calculations

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.24
10 // Page 256
11 printf("Example 5.24, Page 256 \n \n");
12
13 // solution
14
15 // fi1 = integr (from 304.15 to 313.15)
       {11831.6+24997.4*10T^-3-5979.8*10^-6T
       ^2-31.7*10^-9T3}dt
16 fi1 = 170787.7 // kJ/h
17 fi2 = 535.4*12086 -
       [344.36*8743.2+168.97*18036+22.07*15892]  // kJ/h
18 printf(" (a) \n \n Refrigiration requirement = "+
       string(fi1)+" kJ/h \n \n \n (b) \n \n
       Refrigiration requirement based on real
       enthalpies = "+string(fi2)+" kJ/h.")
```

Scilab code Exa 5.25 Chlorination of benzene

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.25
// Page 257
printf("Example 5.25, Page 257 \n \n");

// solution

// basis 100 kmol/h of benzene feed rate
Cl2 = .4*100
HClp = 40
Benzenecon = 37
MCBp = 100*.37*.9189
DCBp = Benzenecon-MCBp
unreactBenzene = 100-Benzenecon
Nt = HClp + MCBp + DCBp + unreactBenzene
// using eq      xi = Ni/(L(1-K1)+NtKi)  and sigma
   xi = 1
L = 89.669  // kmol/h
V = Nt - L
printf(" Liquid product stream = "+string(L)+" kmol/
   h \n Vapor product stream = "+string(V)+" kmol/h"
   )
```

Scilab code Exa 5.26 Heat of formation of ethylene

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.26
// Page 260
printf("Example 5.26, Page 260 \n \n");

// solution

// 2C + 2O2 = 2CO2                    A
// 2H2 + O2 = 2H2O                    B
// C2H4 + 3O2 = 2CO2 + 2H2O           C
// A+B-C  gives
// 2C(g) + 2H2 = C2H4(g)              D
H = -2*393.51-2*241.82+1323.1  // kJ/mol
printf(" Heat of formation of Ethylene is "+string(H
    )+" kJ/mol.")
```

Scilab code Exa 5.27 Heat of combustion of ethyl mercaptan

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


```

```
// Example 5.27
// Page 260
printf("Example 5.27, Page 260 \n \n");

// solution

Hc = 2*(-393.51)-887.811+2*(-285.83)-(-73.6+0)  //kJ
   /mol
printf(" Heat of combustion of ethyl mercaptan = "+
   string(Hc)+" kJ/mol.")
```

Scilab code Exa 5.28 Std heat of formation of gaseous di ethyl ether

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.28
// Page 261
printf("Example 5.28, Page 261 \n \n");

// solution

lv1 = 26694  // kj/kmol
Tc = 466.74
lv2 = lv1*((Tc-298.15)/(Tc-307.7))^.38/1000  // kJ/
   mol
Hf = -252    // kJ/mol
Hf1 = Hf-lv2  // kJ/kmol
printf("Heat of formation of liquid di ethyl ether =
    "+string(Hf1)+" kJ/mol.")
```

Scilab code Exa 5.29 Heat of formation of motor spirit

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.29
// Page 261
printf("Example 5.29, Page 261 \n \n");

// solution

// basis 1 kg motor spirit
G = 141.5/(131.5+64)
// r = C/H
r = (74+15*G)/(26-15*G)
C = r/6.605 // C content of motor spirit
H2 = 1-C
O2req = C+H2
Hf = 44050-27829-18306 // kJ/kg
printf(" Heat of formation of motor spirit = "+
   string(Hf)+" kJ/kg.")
```

Scilab code Exa 5.30 Mean heat capacity

```
clear;
clc;
```

```
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.30
10 // Page 267
11 printf("Example 5.30, Page 267 \n \n");
12
13 // solution
14
15 // basis 1 kmol of styrene
16 dH = 241749-189398  // kJ/mol
17 Cmpn = dH/(600-298.15)  // kJ/kmol K
18 printf(" Mean heat capacity between 600K and 298.15
     K is "+string(Cmpn)+" kJ/kmol K.")
```

Scilab code Exa 5.31 Heat of reaction

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.31
10 // Page 269
11 printf("Example 5.31, Page 269 \n \n");
12
13 // solution
14
15 // basis 1 mol of SiO2 reacted
```

```
16 Hf = [-2879+3*(-296.81)+3*0/2]-[3*(-1432.7)
   +1*(-903.5)]  // kJ/mol SiO2
17 printf(" Heat of reaction = "+string(Hf)+" kJ/mol
   SiO2.")
```

Scilab code Exa 5.32 Std heat of reaction

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.32
10 // Page 269
11 printf("Example 5.32, Page 269 \n \n");
12
13 // solution
14
15 // basis 100 kg of 2% ammonia solution
16 NH3 = 2 // kg
17 H2O = 98 // kg
18 Hr = -361.2-(-45.94-285.83)  // kJ/mol NH3 dissolved
19 Hd = -(Hr*2*1000/17.0305)  // kJ/100 kg sol.
20 printf(" heat of reaction = "+string(Hd)+" kJ/100 kg
    solution.")
```

Scilab code Exa 5.33 Burning of SO2

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

```
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.33
10 // Page 272
11 printf("Example 5.33, Page 272 \n \n");
12
13 // solution
14
15 // basis 1 kmol of SO2 reacted
16 a = 22.036-24.771-.5*(26.026)
17 b = (121.624-62.948-.5*11.755)
18 c = (-91.876+44.258-.5*(-2.343))
19 d = (24.369-11.122-.5*(-.562))
20 Hr = -395720+296810  // kJ/kmol
21 Hro = Hr-a*298.15-b*10^-3*298.15^2/2-c
   *10^-6*298.15^3/3-d*10^-9*298.15^4/4
22 T = 778.15
23 Hrt = -Hro-15.748*T+26.4*10^-3*T^2-15.48*10^-6*T
   ^3+3.382*10^-9*T^4
24 printf(" Heat of reaction at 775K is "+string(Hrt)+"
    kJ/kmol.")
```

Scilab code Exa 5.34 Esterification of acetic acid

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
```

```
8
9  // Example 5.34
10 // Page 272
11 printf("Example 5.34, Page 272 \n \n");
12
13 // solution
14
15 Hr = -480-285.83+277.2+484.2  // kJ/mol
16 Hrt1 = Hr*1000 + [146.89+75.76-119.55-129.70]*75  //
       kJ/kmol
17 a = 4.2905+50.845-100.92-155.48
18 b = 934.378+213.08+111.8386+326.5951
19 c = -2640-631.398-498.54-744.199
20 d = 3342.58+648.746
21 Hro = Hr*1000+a*(-298.15)+b*10^-3*(-298.15^2)/2+c
      *10^-6*(-298.15^3)/3+d*10^-9*(-298.15^4)/4
22 T = 373.15
23 Hrt = Hro+a*T+792.949*10^-3*T^2-1504.712*10^-6*T
      ^3+997.832*10^-9*T^4
24 printf(" Heat of reaction at 373 K is "+string(Hrt)+
      " kJ/kmol reactant.")
```

Scilab code Exa 5.35 Heat transfer in intercoolers

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.35
10 // Page 273
11 printf("Example 5.35, Page 273 \n \n");
```

```
12
13 // solution
14
15 T2 = 800
16 T1 = 298.15
17 fi1 = 3614.577*(T2-T1)+305.561*10^-3*(T2^2-T2^2)
      /2+836.881*10^-6*(T2^3-T1^3)/3-393.707*10^-9*(T2
      ^4-T1^4)/4   // kW
18 T3 = 875
19 fi2 = 3480.737*(T3-T1)+754.347*10^-3*(T3^2-T2^2)
      /2+442.159*10^-6*(T3^3-T1^3)/3-278.735*10^-9*(T3
      ^4-T1^4)/4   // kW
20 Hr = -98910  // kJ/kmol SO2 reacted      by eg 5.33
21 fi3 = (8.8511-.351)*Hr/3600  // kW
22 dH = fi2/3600+fi3-fi1/3600
23 printf(" Net enthalpy change = "+string(dH)+" kW.")
```

Scilab code Exa 5.36 Enthalpy balance in the reactor

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.36
10 // Page 275
11 printf("Example 5.36, Page 275 \n \n");
12
13 // solution
14
15 // basis 100 kmol outgoing gas mixture from scrubber
16 moistin = 3127.7*.015/18  // kmol
```

```
17 waterin = 40.2+moistin  // kmol
18 // using tables 5.29 and 5.30
19 Hr = -27002658-(-26853359)
20 Hr1 = Hr/246.4493  // kJ/kmol total reactants
21 printf(" Heat of reaction = "+string(Hr1)+" kJ/kmol
      total reactants.")
```

Scilab code Exa 5.37 Calculation of circulation rate

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.37
10 // Page 276
11 printf("Example 5.37, Page 276 \n \n");
12
13 // solution
14
15 fi3 = 15505407  // kJ/h
16 lv = 296.2  // from table 5.6
17 Ht = 17131551  // kJ/h
18 r = Ht/lv  // kg/h
19 printf(" Downtherm circulation rate = "+string(r)+"
      kg/h.")
```

Scilab code Exa 5.38 Loop reactor for EDC manufacture

```
1 clear;
```

```
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.38
// Page 279
printf("Example 5.38, Page 279 \n \n");

// solution

F = 100  // kmol/h    feed rate of ethylene
Econ = .99*F
Econ1 = Econ*.998
Econ2 = Econ-Econ1
Cl2con = Econ1+2*Econ2
Cl2in = F*1.1
Cl2s3 = Cl2in-Cl2con
HCls3 = Econ2
TCEp = Econ2
EDCp = Econ1
nC2H4 = 1
T = 328.15
pv1 = exp(4.58518-1521.789/(T-24.67))  // bar
pv2 = exp(4.06974-1310.297/(T-64.41))  // bar
xEDC = Econ1/(Econ1+Econ2)
xTEC = 1-xEDC
pEDC = 37.2*xEDC
pTEC = 12.64*xTEC
pCl2HClC2H4 = 1.6*100-pEDC-pTEC
yEDC = pEDC/160
yTEC = pTEC/160
nt = (Cl2s3+Econ2+1)*160/pCl2HClC2H4
nEDC = yEDC*nt
nTEC = yTEC*nt
printf(" Compositions of gas streams : \n \n
```

```
    Component              Stream 3              Stream 5
              Stream 4              Stream 6 \n Cl2
                    "+string(Cl2s3)+"                "+
    string(Cl2s3)+" \n HCl                    "+string(
    HCls3)+"                "+string(HCls3)+" \n C2H4
                   "+string(nC2H4)+"
    "+string(nC2H4)+" \n EDC                    "+string
    (nEDC)+"           0.2355              3.3947
                98.5665 \n TEC                    "+
    string(nTEC)+"             Nil                    "+string
    (nTEC)+"           "+string(TCEp)+" \n \n ")
fi1 =
    (10.802*33.9+.198*29.1+1*43.6+3.6302*17.4+.0025*85.3)
    *(328.15-273.15)
fi2 = 35.053*1000*3.3947+39.58*1000*.0025
fi3 = (3.3947*129.4+.0025*144.4)*55/2
fi = fi1+ fi2+ fi3  // kJ/h
printf(" Heavy duty of Overhead condenser = "+string
    (fi)+" kJ/h. \n \n ")
fi5 = (100*43.6+110*33.9)*(328.15-273.15)
fi6 = 3.6302*1000*33.6+.0025*1000*38.166
fi7 = (98.5665*129.4+.1988*144.4)*(328.15-273.15)
fi8 = 216845.5*98.802+392394.5*.198
ficol = fi5+fi8-fi1-fi6-fi7
printf(" Heavy duty of external cooler = "+string(
    ficol)+" kJ/h.")
```

Scilab code Exa 5.39 Calculations in adiabatic converter

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances
```

```
7
8
9  // Example 5.39
10 // Page 284
11 printf("Example 5.39, Page 284 \n \n");
12
13 // solution
14
15 To = 298.15
16 T1 = 483.15
17 // fi1 = intgr(from To to T1){12199.5+2241.4*10^-3*T
   +1557.7*10^-6*T^2-671.3*10^-9*T^3}dT
18 fi1 = 2455874.6  // kJ/h
19 dHr = 2*(-45.94)  // kJ/mol N2 reacted
20 fi2 = 91.88*1000*23.168
21 fi3 = fi1+fi2
22 // fi3 = intgr(from To to T2){10713.9+3841*10^-3*T
   +1278.8*10^-6*T^2-752.6*10^-9*T^3}dT
23 // solving it
24 T2 = 657.41  // K
25 printf("Temperature of the gas mixture leaving the
   reactor = "+string(T2)+" K.")
```

Scilab code Exa 5.40 Burning of HCl

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.40
10 // Page 292
```

```
11 printf("Example 5.40, Page 292 \n \n");
12
13 // solution
14
15 // basis 4 kmol of HCl gas
16 O2req = 1  // kmol
17 O2spply = 1.35*1
18 N2 = 1.35*79/21
19 air = O2spply+N2
20 HClbrnt = .8*4
21 HCl = 4-HClbrnt
22 O2 = O2spply-.8
23 Cl2 = .8*2
24 H2O = .8*2
25 printf(" (a) \n \n Composition of dry product gas
   stream : \n Component              Dry product gas
   stream,kmol \n HCl                    "+string(HCl)+
   " \n O2                    "+string(O2)+" \n Cl2
                    "+string(Cl2)+" \n H2O
                    "+string(H2O)+" \n N2
                     "+string(N2)+" \n \n \n (b) \n \
   n ")
26 H2 = 114.4*1000*.8
27 // H2 = intgr(from 298.15 to T)
   {286.554+12.596*10^-3*T+63.246*10^-6*T
   ^2-25.933*10^-9*T^3}dT
28 // solving it
29 T = 599.5  // K
30 printf(" Adiabatic reaction temperature of product
   gas stream = "+string(T)+" K.")
```

Scilab code Exa 5.41 Dehydrogenation of EB

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

```
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.41
10 // Page 294
11 printf("Example 5.41, Page 294 \n \n");
12
13 // solution
14
15 // 1 kmol of EB vapors entering the reactor at
      811.15 K
16 // (from 811.15 to T1)intgr{-36.72+671.12*10^-3*T
      -422.02*10^-6*T^2+101.15*10^-9*T^3}dT = (from T1
      to 978.15)intgr{487.38+1.19*10^-3*T+198.16*10^-6*
      T^2-68.21*10^-9*T^3}dT
17 // we get
18 T1 = 929.72  // K
19 To = 298.15
20 H1 = 493405  // kJ
21 EBr = .35
22 Styrenep = EBr*.9
23 Benzeneb = EBr*.03
24 Ethyleneb = Benzeneb
25 Cb = EBr*.01
26 Toulened = EBr*.06
27 Hr1 = 147.36-29.92  // kJ/mol EB
28 Hr2 = 82.93+52.5-29.92
29 Hr3 = -29.92
30 Hr4 = 50.17-74.52-147.36  // kJ/mol styrene
31 dHr = 1000*(Hr1*(Styrenep+Toulened)+Hr2*Benzeneb+Hr3
      *Cb+Hr4*Toulened)
32 H2 = H1-dHr
33 // H2 = (from To t0 T2)intgr{Comp2dT
34 // we get
35 T2 = 798.79  // K
```

```
36 printf(" Adiabatic reaction T at the outlet of the
     reactor is "+string(T2)+" K.")
```

Scilab code Exa 5.42 Heat of crystallization

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.42
10 // Page 297
11 printf("Example 5.42, Page 297 \n \n");
12
13 // solution
14
15 Hsol = 62.86  // kJ/mol solute
16 Mcrystal = 286.1414
17 Hcry = Hsol*1000/Mcrystal  // kJ/kg solute
18 printf(" Heat of crystallization of 1 kg crystal is
     "+string(Hcry)+" kJ.")
```

Scilab code Exa 5.43 Heat of crystallization

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
```

```
7
8
9  // Example 5.43
10 // Page 297
11 printf("Example 5.43, Page 297 \n \n");
12
13 // solution
14
15 Hf = -285.82  // kJ/mol     of H2O
16 Hcryst = -4327.26-(-1387.08+10*Hf)
17 printf(" Heat of crystallization = "+string(Hcryst)+
       " kJ/mol.")
```

Scilab code Exa 5.44 Heat of sol of Boric acid

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.44
10 // Page 297
11 printf("Example 5.44, Page 297 \n \n");
12
13 // solution
14
15 Hfs = -1094.33
16 Hfao = -1072.32
17 Hsol = Hfao-Hfs
18 printf(" Heat of solution of Boric acid = "+string(
       Hsol)+" kJ/mol.")
```

Scilab code Exa 5.45 Heat of dissolution

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.45
// Page 297
printf("Example 5.45, Page 297 \n \n");

// solution

// (a)
Hf = -982.8
Hfcryst = -1053.904
Hdis = Hfcryst-Hf
// (b)
Hfcr = -3077.75
Hsol = Hfcryst+7*(-285.83)-(-3077.75)
printf(" (a) \n \n Hdissolulition = "+string(Hdis)+"
    kJ/mol ZnSO4. \n \n \n (b) \n \n Hsolution = "+
   string(Hsol)+" kJ/kmol.")
```

Scilab code Exa 5.46 T change in dissolution

```
clear;
clc;

```

```
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.46
10 // Page 300
11 printf("Example 5.46, Page 300 \n \n");
12
13 // solution
14
15 // using chart 5.16 we get
16 T = 329.5  // K
17 printf(" T = "+string(T)+" K.")
```

Scilab code Exa 5.47 Using std heat of formations

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.47
10 // Page 300
11 printf("Example 5.47, Page 300 \n \n");
12
13 // solution
14 // basis 100(ml) kg 46% sol
15 NaOH = 46  // kg
16 H2O = 54  // kg
17 m2 = NaOH/.25
18 NaOHo = 25  // kg
```

```
19 H2Oo = 75  // kg
20 Hf1 = -453.138  // kJ/mol
21 Hf2 = -467.678  // kJ/mol
22 Hs = Hf2-Hf1
23 Hg = -Hs*1000*1.501
24 // using Appendix IV.1
25 Hw1 = 146.65
26 Hw2 = 104.9
27 Hadd = 84*(Hw1-Hw2)
28 H = Hg+Hadd
29 C1 = 3.55
30 T2 = 298.15+H/(184*C1)  // K
31 printf(" Final sol T = "+string(T2)+" K.")
```

Scilab code Exa 5.48 Heat effect of the solution

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.48
10 // Page 301
11 printf("Example 5.48, Page 301 \n \n");
12
13 // solution
14 // basis 100 kg of sol with 32% N
15 MNH4NO3 = 80.0434
16 MNH2CONO2 = 60.0553
17 MN2 = 28.0134
18 na = 32/(60.9516)
19 Ureadis = 1.1758*na*MNH2CONO2  // kg
```

```
20 water = 100-(na*MNH4NO3+Ureadis)
21 ndis = 525
22 m = ndis/water
23 HE1 = 40.3044-2.5962*m+.1582*m^2-3.4782*10^-3*m^3
24 HE = HE1*ndis
25 printf("Heat effect of the sol = "+string(HE)+" kJ."
   )
```

Scilab code Exa 5.49 Integral heats of solution

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 5
6  // Energy Balances
7
8
9  // Example 5.49
10 // Page 302
11 printf("Example 5.49, Page 302 \n \n");
12
13 // solution
14 Hmix = 896
15 M1 = 88  // molar mass of n-amyl alcohol
16 M2 = 78  // molar mass of benzene
17 B = .473*M2
18 A = .527*M1
19 Ha = Hmix/A
20 Hb = Hmix/B
21 printf(" Integral heat of sol of n-amyl alcohol = "+
   string(Ha)+" kJ/kg n-amyl alcohol and of benzene
   = "+string(Hb)+" kJ/kg benzene.")
```

Scilab code Exa 5.50 Hx for H2SO4

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.50
// Page 302
printf("Example 5.50, Page 302 \n \n");

// solution
// from fig 5.18
Ta = 379.5 // K
dH = -274-(-106.5) // kJ/kg sol
Cm = 2.05  // kJ/kg K
dHc = Cm*(Ta-298.15)
// basis 100 kg of 93 % acid
// acid balance
x = poly(0, 'x')
p = .93*100+x*.15-(100+x)*.77
y = roots(p)
//from fig
y1 = 25.3
printf(" (a) \n \n Resultant T of 77 percent sol = "
   +string(Ta)+" K. \n \n \n (b) \n \n Heat to be
   removed to cool it to 298.15 K = "+string(dH)+"
   kJ/kg sol \n \n \n (c) \n \n By mean heat
   capacity method : "+string(dHc)+" kJ/kg sol \n \n
    \n (d) \n \n Quantity of 15 percent acid to be
   mixed = "+string(y)+" kg. \n \n \n (e) \n \n from
```

```
    fig : "+string(y1)+" kg.")
```

Scilab code Exa 5.51 Using heat of formations of H2SO4

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.51
// Page 304
printf("Example 5.51, Page 304 \n \n");

// solution
// basis 100 kg of 93% acid and 25.8 kg of 15% acid
Hfp = -814
Hf1 = -830
HE1 = Hf1-Hfp
Hf2 = -886.2
HE2 = Hf2-Hfp
Hf3 = -851
HE3 = Hf3-Hfp
Hsol = .9876*1000*(-37)-[.9482*1000*(-16)
   +.0394*1000*(-72.2)]
Hev = 100*(30-25)*1.6
Hcon = 25.8*25*3.7
netHev = -Hsol-Hcon+Hev
T = 298.15+netHev/(125.8*2.05)
printf(" Temp of sol = "+string(T)+" K.")
```

Scilab code Exa 5.52 Heat to be removed for cooling it to 308K

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.52
// Page 306
printf("Example 5.52, Page 306 \n \n");

// solution

// basis 1000 kg of mixed acid
C11 = 2.45
H1 = -296.7+C11*(308.15-273.15)
C12 = 2.2
H2 = -87.8+C12*(308.15-273.15)
C13 = 1.45
H3 = -35.5+C13*(308.15-273.15)
C14 = 1.8
H4 = -148.9+C14*(308.15-273.15)
Hmix = 1000*H4-[76.3*H1+345.9*H2+577.7*H3]
printf(" Heat of mixing = "+string(Hmix)+" kJ.")
```

Scilab code Exa 5.53 Heat changes in formation of MNB

```
clear;
clc;

// Stoichiometry
// Chapter 5
```

```
// Energy Balances


// Example 5.53
// Page 308
printf("Example 5.53, Page 308 \n \n");

// solution

F = 1135
Benzenef = 400*.993
HNO3con = Benzenef*63/78
H1 = -186.5
C11 = 1.88
H11 = H1+C11*(298.15-273.15)
H2 = -288.9
C12 = 1.96
H22 = H2+C12*(298.15-273.15)
H3 = 0
C13 = 1.98
H33 = C13*(298.15-273.15)
Hr = -285.83+12.5-(-174.1+49.08)
Benzener = Benzenef/78.1118
fi = 903.84*H22+HNO3con*H33-F*H11+Benzener*Hr*1000
   // kJ/h
printf(" Total heat exchanged = "+string(fi)+" kJ/h.
   ")
```

Scilab code Exa 5.54 Final T of solution in absorption of NH3

```
clear;
clc;

// Stoichiometry
// Chapter 5
```

```
6 // Energy Balances
7
8
9 // Example 5.54
10 // Page 311
11 printf("Example 5.54, Page 311 \n \n");
12
13 // solution
14
15 // from ref 24
16 H = 1600.83
17 To = 273.15
18 h = 200
19 Hf1 = -79.3  // table 5.59
20 Hf2 = -46.11
21 Hsol = Hf1-Hf2
22 Hg = Hsol*1000*140/17.0305
23 Raq = 140/.15  // kg/h
24 dT = Hg/(4.145*Raq)
25 T = -dT+303
26 printf(" Temp of resultant sol = "+string(T)+" K.")
```

Scilab code Exa 5.55 Using table 5 60

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.55
10 // Page 311
11 printf("Example 5.55, Page 311 \n \n");
```

```
12
13 // solution
14
15 Hf1 = -80.14
16 Hf2 = -46.11
17 Hsol = Hf1-Hf2
18 Hg = Hsol*1000*2/17.0305
19 printf(" Heat generated for making 2 percent
     solution = "+string(Hg)+" kJ/100 kg sol.")
```

Scilab code Exa 5.56 Heat removed in cooler

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 5
6 // Energy Balances
7
8
9 // Example 5.56
10 // Page 312
11 printf("Example 5.56, Page 312 \n \n");
12
13 // solution
14
15 fi3 = 15505407
16 fi4 = 11395056
17 fi5 = fi3-fi4  // kJ/h
18 fi6 = 111.375*62.75*1000
19 fi7 = 1063379
20 fi8 = 5532.15*4.1868*(303.15-298.15)
21 fi9 = 9030.4*3.45*(323.15-298.15)
22 fi = fi5+fi6+fi8-fi7-fi9
23 printf(" Heat removal in the cooler = "+string(fi)+"
```

kJ/h.")

Scilab code Exa 5.57 Hx vs x1

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.57
// Page 314
printf("Example 5.57, Page 314 \n \n");

// solution

To = 273.15
T1 = 308.15
H1 = 124.8*(T1-To)  // kJ/kmol
H2 = 134.9*(T1-To)  // kJ/kmol
HE1 = .1*.9*[542.4+55.4*(.9-.1)-132.8*(.9-.1)
   ^2-168.9*(.9-.1)^3]  // kJ/kmol of mix
Ha = HE1+H1*.1+H2*.9
HE2 = .2*.8*[542.4+55.4*(.8-.2)-132.8*(.8-.2)
   ^2-168.9*(.8-.2)^3]  // kJ/kmol of mix
Hb = HE2+H1*.2+H2*.8
HE3 = .3*.7*[542.4+55.4*(.7-.3)-132.8*(.7-.3)
   ^2-168.9*(.7-.3)^3]  // kJ/kmol of mix
Hc = HE3+H1*.3+H2*.7
HE4 = .4*.6*[542.4+55.4*(.6-.4)-132.8*(.6-.4)
   ^2-168.9*(.6-.4)^3]  // kJ/kmol of mix
Hd = HE4+H1*.4+H2*.6
HE5 = .5*.5*[542.4+55.4*(.5-.5)-132.8*(.5-.5)
```

```
    ^2-168.9*(.5-.5)^3]  // kJ/kmol of mix
He = HE5+H1*.5+H2*.5
HE6 = .6*.4*[542.4+55.4*(.4-.6)-132.8*(.4-.6)
    ^2-168.9*(.4-.6)^3]  // kJ/kmol of mix
Hf = HE6+H1*.6+H2*.4
HE7 = .7*.3*[542.4+55.4*(.3-.7)-132.8*(.3-.7)
    ^2-168.9*(.3-.7)^3]  // kJ/kmol of mix
Hg = HE7+H1*.7+H2*.3
HE8 = .8*.2*[542.4+55.4*(.2-.8)-132.8*(.2-.8)
    ^2-168.9*(.2-.8)^3]  // kJ/kmol of mix
Hh = HE8+H1*.8+H2*.2
HE9 = .9*.1*[542.4+55.4*(.1-.9)-132.8*(.1-.9)
    ^2-168.9*(.1-.9)^3]  // kJ/kmol of mix
Hi = HE9+H1*.9+H2*.1
HE10 = .0*1.*[542.4+55.4*(.0-1.)-132.8*(.0-1.)
    ^2-168.9*(.0-1.)^3]  // kJ/kmol of mix
Hj = HE10+H1+H2*0
x = linspace(0,1,100)
y = linspace(4300,5000,100)
y = 4721.5-57.4*x+1137.7*x^2-3993.6*x^3+3909.2*x
    ^4-1351.2*x^5
plot(x,y)
title("H vs x1")
xlabel("x1")
ylabel("H (kJ/kg sol.)")
printf("                              Enthalpy, kJ/kmol
    mix \n  x1            HE                    H \n  0
                0                    "+string(H2)+" \n
    0.1           "+string(HE1)+"          "+string(Ha
    )+" \n  0.2           "+string(HE2)+"          "+
    string(Hb)+" \n  0.3           "+string(HE3)+"
              "+string(Hc)+" \n  0.4           "+
    string(HE4)+"          "+string(Hd)+" \n  0.5
              "+string(HE5)+"                "+string(He
    )+" \n  0.6           "+string(HE6)+"          "+
    string(Hf)+" \n  0.7           "+string(HE7)+"
              "+string(Hg)+" \n  0.8           "+string
    (HE8)+"          "+string(Hh)+" \n  0.9
```

```
"+string(HE9)+"            "+string(Hi)+" \n   1.0
          "+string(HE10)+"                      "+
string(Hj)+"")
```

Scilab code Exa 5.58 repat of 5 57 using heat capacities

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.58
// Page 316
printf("Example 5.58, Page 316 \n \n");

// solution

// see eg 5.57
printf(" refer to eg 5.57")
```

Scilab code Exa 5.59 He vs x1 of acetone and ethylacetate

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


```

```
// Example 5.59
// Page 318
printf("Example 5.59, Page 318 \n \n");

// solution

// from graph drawn in 5.57 we can see
H1E1 = 300
H1E2 = 63
H2E1 = 30
H2E2 = 214
printf(" H1 at x1=0.3 is "+string(H1E1)+" kJ/kg sol
   \n H2 at x1=0.3 is "+string(H2E1)+" kJ/kg sol \n
   H1 at x1=0.6 is "+string(H1E2)+" kJ/kg sol \n H2
   at x1=0.6 is "+string(H2E2)+" kJ/kg sol.")
```

Scilab code Exa 5.60 Heat of dilution

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.60
// Page 320
printf("Example 5.60, Page 320 \n \n");

// solution

// basis 100 kg 96.1% H2SO4
// from table 5.64
m1SO3 = 78.4  // kg
```

```
m1H2O = 21.6
n1SO3 = m1SO3/80.063
n1H2O = m1H2O/18.015
// resultant sol has 23.2% H2SO4
m2SO3 = 19
m2H2O = 81
Mrsol = m1SO3*100/m2SO3
Mw = Mrsol-100
w = Mrsol-m1SO3/18.015  // kmol
HEosol = n1SO3*(-56940)+n1H2O*(-32657)  // kJ
HErsol = n1SO3*(-156168)+w-(-335)
HE = HErsol-HEosol  // kJ/kg original acid
C = 3.43  // kJ/kg K
dT = -HE/(Mrsol*C)
T = 291.15+dT  // K
printf(" Heat of dilution = "+string(HE)+" kJ/kg
   original solution \n \n Final T of resultant
   solution = "+string(T)+" K.")
```

Scilab code Exa 5.61 eg 5 60 with use of ice at 273K

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.61
// Page 321
printf("Example 5.61, Page 321 \n \n");

// solution

```

```
// basis 100 kg of original acid
lv = 333.7  // kJ/kg
H = -lv-18*4.1868
HE = (-64277-H*312.63)/100  // kJ/kg
printf(" Heat of dilution = "+string(HE)+" kJ/kg.")
```

Chapter 6

Stoichiometry and Unit Operations

Scilab code Exa 6.1 Overall material and energy balance

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 6
6  // Stoichiometry and Unit Operations
7
8
9  // Example 6.1
10 // Page 346
11 printf("Example 6.1, Page 346 \n \n");
12
13 // solution
14
15 // basis = 100kmol of feed
16 Benzene = 100*.72// kmol
17 Toulene = 100-Benzene //kmol
18 // use fig 6.1
19 // D = distillate, B = bottom
```

```
// F = B + D          (i)    overall material balance
xd = .995
xb = .03
xf = .72
// xd*D + xb*B = F*xf      (ii)     benzene balance
// solving (i) and (ii)
D = 71.5 //kmol
B = 28.5 //kmol
printf("(a) \n \n performing overall material
   balance for 100kmol of feed we get "+string(D)+"
   kmol as distillate and "+string(B)+"kml as bottom
    product. \n \n \n (b) \n \n ")
// enthalpy balance
// use fig 6.2
R = 1.95
v = D*(1+R) //kmol     total overhead vapours
To = 273.15 //K
// using fig 6.2
Ev = 42170 //kJ/kmol     enthalpy of vapours overhead
El = 11370 //kJ/kmol     enthalpy of liquid
E1 = Ev-El // enthalpy removed in condenser
Hc = E1*v  // heat load of condenser
Hd = El*71.5
Hb = 18780*28.5
Hf = 44500*100
Hn = Hd+Hc+Hb-Hf // kJ     heat load of reboiler
printf(" performing overall enthalpy balance we get
   Heat load of condenser = "+string(Hc)+"kJ/kmol
   and Heat load of reboiler = "+string(Hn)+"kJ/kmol
   .")
```

Scilab code Exa 6.2 Cryogenic Separation of Nitrogen

```
clear;
clc;
```

```
3
4 // Stoichiometry
5 // Chapter 6
6 // Stoichiometry and Unit Operations
7
8
9 // Example 6.2
10 // Page 349
11 printf("Example 6.2, Page 349 \n \n");
12
13 // solution
14
15 // basis = 2000kg/h liquid feed rate
16 F = 2000/28.84 //kmol/h
17 //D = distillate, W = residue flow rate
18 //N2 balance
19 // F*.79 = .999D + .422W    (i)
20 // 54.840 = D + .4224W      (ii)
21 // solving it
22 W = 25.118 //kmol/h
23 D = 44.230 //kmol/h
24 //using fig 6.4 and 6.5
25 // trial method is used for flash calculations
26 // Trial I
27 x = .75
28 // from fig 6.4
29 y = .8833
30 // from fig 6.5
31 Hl = 1083.65
32 Hv = 6071.7
33 Hf = .3*Hv+Hv*.7
34 // calculating we get Emix is not close to 2592.2kJ/
     kmol
35 //Trial II
36 x = .71
37 y = .859
38 Hl = 1085.6
39 Hv = 6118.6
```

```
40 Hf = .3*Hv+.7*Hl //kJ/kmol
41 // which is aproox equal to 2595.2kJ/kmol, so
      flashing will occur
42 printf("composition of vapour liquid mix : \n mol
      fraction N2 = "+string(x)+" in liquid phase and "
      +string(y)+" in vapour phase.")
```

Scilab code Exa 6.3 Azeotropic distillation of IPA and water

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 6
6  // Stoichiometry and Unit Operations
7
8
9  // Example 6.3
10 // Page 353
11 printf("Example 6.3, Page 353 \n \n");
12
13 // solution
14 // material balance
15
16 // V2 vapour mix is a ternary azeotrope in which all
       cyclohexane of D1 is recycled
17 // V2 stream
18 // Cyclohexane balance
19 // D1 = (.488/.024)*V2
20 // IPA in V2 = .206V2
21 // water in V2 = (1-.488-.206)*V2
22 // W2 stream
23 // IPA in W2 = (.23D1-.206V2)
24 // water in W2 = (1-.024-.23)*D1-.306V2
25 // W2 stream = 4.471V2 + 14.862V2
```

```
26 // D3 is an azeotrope containing 67.5 mol% IPA
27 // water in W3 stream = (1-.675)F
28 // basis = 100 kmol/h fresh feed
29 // W1+W3 = 100                    (i)
30 // .998W1 + .001W3 = 67.5        (ii)
31 // solving it
32 W1 = 67.603 //kmol/h
33 W3 = 32.397 //kmol/h
34 IPA1 = W3*.001 // IPA in W3
35 //IPA2 = 4.471*V2 - .032   IPA in D3
36 //C-1 = F+D3 = F1
37 // water in D3 = 6.624V2 - .047-4.471V2+.032
38 // water in W3 = 14.862V2-2.153V2+.015
39 // solving them
40 V2 = 2.624 //kmol/h
41 D3 = 2.153*V2-.015
42 D1 = 20.333*V2
43 F1 = 6.624*V2+99.953
44 R = 1.75*D1 // = V1+V2-D1
45 V1 = 144.1
46 r = D3/100 // recycle ratio
47 printf("After performing overall material balance we
       get Reflux, R = "+string(R)+"kmol/h and \n
     recycle ratio = "+string(r)+" kmol/kmol fresh
     feed.")
```

Scilab code Exa 6.4 CO2 absorption in aq MEA solution

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 6
6 // Stoichiometry and Unit Operations
7
```

```
8
9  // Example 6.4
10 // Page 355
11 printf("Example 6.4, Page 355 \n \n");
12
13 // solution
14
15 // basis 0.625 l/s of MEA solution
16 c = 3.2 //M  conc of MEA
17 M = 61 // molar mass of MEA
18 C = M*c //g/l  conc of MEA in sol
19 MEAin = c*.625*3600/1000 // kmol/h
20 CO2diss = .166*7.2 //kmol/h    CO2 dissolved in lean
       MEA
21 v = 26.107 //m^3  sp. vol of gas at 318K and 101.3
      kPa  (table 7.8)
22 qv = 1000/v //kmol/h
23 CO2in = qv*.104 // moles of CO2 in inlet gas
24 CO2freegas = qv - CO2in
25 //outgoing has 4.5% CO2
26 GASout = CO2freegas/(1-.0455) //kmol/h
27 CO2abs = qv-GASout
28 CO2 = CO2diss + CO2abs
29 CO2conc = CO2/MEAin //kmol/kmol MEA
30 printf("Concentration of dissolved CO2 in the
      solution leaving the tower = "+string(CO2conc)+"
      kmol/kmol of MEA.")
```

Scilab code Exa 6.5 Heat effect of Scrubbing

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 6
```

```
// Stoichiometry and Unit Operations


// Example 6.5
// Page 356
printf("Example 6.5, Page 356 \n \n");

// solution

//(a)
printf("(a) \n \n")
// basis 50000 m^3/h of gas mix at 295.5K 100kPa
v = 24.57 //m^3/kmol sp vol of gas at 295.5K and 100
   kPa
n1 = 50000/v // kmol/h    flow of incoming gas
NO2in = n1*.0546
N2O4in = n1*.0214
N2in = n1-NO2in-N2O4in
//N2 is unaffected
n2 = 1880.34/.95 //kmol/h   outgoing gas flow
// using tables 6.3 and 6.4 on page 357
NO2rem = NO2in - (n2*.0393)
N2O4rem = N2O4in - (n2*.0082)
// rxn (ii)
NaOHreac2 = 2*40*N2O4rem
NaNO2pro2 = 69*N2O4rem
NaNO3pro2 = 85*N2O4rem
H2Opro2 = 18*N2O4rem
// rxn (iii)
NO2reac3 = 3*n2*.0025
NaOHreac3 = 2*4.95*40
NaNO3pro3 = 2*4.95*85
H2Opro3 = 4.95*18
NO2abs2 = 33.33-NO2reac3
NaOHreac1 = 18.48*40
NaNO2pro1 = 69*NO2abs2/2
NaNO3pro1 = 85*NO2abs2/2
H2Opro1 = 18*NO2abs2/2
```

```
43 NaNO2t = NaNO2pro2 + NaNO2pro1
44 NaNO3t = NaNO3pro2+NaNO3pro3
45 H2Ot = H2Opro1+H2Opro2+H2Opro3
46 NaOHt = NaOHreac1+NaOHreac2+NaOHreac3
47 liq = 37500 //kg/h
48 NaOHin = liq*.236
49 NaOHout = NaOHin-NaOHt
50 moist = n2*.045*18
51 water = liq-NaOHin-H2Ot-moist //kg/h
52 printf("Composition of final liquor : \n Component
                mi (kg/h) \n NaOH                    "+
   string(NaOHout)+" \n NaNO2                    "+
   string(NaNO2t)+" \n NaNO3                    "+
   string(NaNO3t)+" \n H2O                    "+
   string(water)+" \n \n \n (b)")
53 //(b)
54 //heat effect of scrubbing
55 //using tables 6.6 and 6.7
56 //fi1 = integ{59865.7+4545.8+10^-3 *T +
   15266.3*10^-6*T^2-705.11*10^-9*T^3}
57 fi1 = -155941.3/3600 //kW
58 //similarly
59 fi2 = 75.778 //kW
60 dH1 = (-346.303-450.1-285.83-(2*(-468.257)+2*33.18))
   /2 //kJ/mol NO2
61 dH2 = -346.303-450.1-285.83-(2*(-468.257)+9.16) //kJ
   /mol N2O4
62 dH3 = (2*(-450.1)-285.83+90.25-(2*(-468.257)
   +3*33.18))/3 // kJ/mol NO2
63 dHdil = -469.837-(-468.257) //kJ/mol NaOH
64 fi3 = (dH1*1000*18.48+dH2*1000*27.32+dH3*1000*14.85+
   dHdil*1000*138.23)/3600 //kW
65 fi4 = -fi1+fi2+fi3
66 printf("Heat efeet of scrubbing system = "+string(
   fi4)+" kW.")
```

Scilab code Exa 6.6 Extraction of Acetic Acid

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.6
// Page 361
printf("Example 6.6, Page 361 \n \n");

// solution

//(a)
// basis 100 kg feed mix
// F = E +R = 100              (i)
xf = .475
xe = .82
xr = .14
//acetic acid balance
// xf*F = xe*E + xr*R        (ii)
//solving (i) & (ii)
E = 49.2 //kg
R = 50.8 //kg
a = R*xr //kg  acetic acid leftover
b= (a/(xf*100))*100
printf(" Acetic acid that remained unextracted = "+
   string(b)+" percent.")
```

Scilab code Exa 6.7 Multiple contact counter current Extractor

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.7
// Page 361
printf("Example 6.7, Page 361 \n \n");

// solution

// referring to fig 6.9
//basis 1000kg/h halibut livers
F = 1000 //kg/h
OILin = F*.257
Sin = F-OILin // solid in the charge
U = .23*Sin
OILu = U*.128
Eu = U-OILu // ether in underflow
R = OILin-OILu //kg/h   recovery of oil
p = R*100/OILin
O = R/.7
Eo = O-R
Et = Eu+Eo
printf(" Flow rate of ether to the system = "+string
   (Et)+" kg/h \n and percentage of recovery oil = "
   +string(p)+".")
```

Scilab code Exa 6.8 Recovery of Acetic Acid by Ethyl Acetate Extraction

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.8
// Page 362
printf("Example 6.8, Page 362 \n \n");

// solution

F = 1000 //kg/h     Basis feed rate
// using fig 6.11
// W/A = 15.77/5.87
// A+F+W = 1000
// solving it
W = 15.77*F/21.64 //kg/h
A = F-W //kg/h
// material balance across C3
// R+R1 = D+W
// W/D = 19.31/1.81
// solving it
D = 1.81*W/19.31  //kg/h
M1 = D+W
// R1/R = 4.63/6.57
R1 = 4.63*793/11.2
R = M1-R1
// material balance across C2
m = .89 // = E1/R1
// E = A+E1+R1 = A+M11
// M11/A = 15.6/3.97
M11 = 15.6*A/3.97
E = M11 + A
E1 = M11 - R1
// material balance across C1
```

```
39 // F+S = M = E+R
40 M = E+R
41 S = D+E1
42 AAloss = W*.4*100/(100*.3)
43 AArec = 100-AAloss
44 printf(" Summary : \n Stream
   Flow rate (kg/h) \n Feed
                                  "+string(F)+" \n Solvent
                                "+string(S)+" \n Extract
                               "+string(E)+" \n Raffinate
                             "+string(R)+" \n Acetic acid
                           "+string(A)+" \n Top layer from
    D1                   "+string(E1)+" \n Bottom layer
   from D1               "+string(R1)+" \n Feed to C3
                           "+string(M1)+" \n Overhead
   from C3                   "+string(D)+" \n Water waste
                           "+string(W)+" \n Stream
                               "+string(M)+"")
```

Scilab code Exa 6.9 Yield of Glauber salt

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 6
6  // Stoichiometry and Unit Operations
7
8
9  // Example 6.9
10 // Page 367
11 printf("Example 6.9, Page 367 \n \n");
12
13 // solution
14
```

```
// basis 100 kg free water
Na2SO4in = 32 //kg
Win = 68 //kg
W1 = (180/142)*32 //kg    water with Na2SO4
Wfree1 = Win-W1
GS1 = ((Na2SO4in+W1)*100)/Wfree1 //kg    glauber
   salt present in 100 kg free water
W2 = (180*19.4)/142 // water associated with Na2SO4
   in final mother liquor
Wfree2 = 100-W2
GS2 = ((19.4+W2)/Wfree2)*100
Y = GS1-GS2 //kg
p = Y*100/GS1
printf("Percent yield of glauber salt = "+string(p)+
   ".")
```

Scilab code Exa 6.10 Cooling in a Crystallizer

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.10
// Page 368
printf("Example 6.10, Page 368 \n \n");

// solution

// basis 100kg free water in original sol
// initial T = 353K
W1 = (126/120.3)*64.2    //kg
```

```
Wfree1 = 100-W1
MS1 = ((64.20+W1)*100)/32.76 // MgSO4.7H2O in 100kg
    free water
// 4% of original sol evaporates
E = (MS1 + 100)*.04
Wfree2 = 100-E  // free water in mother liquor
// at 303.15 K
W2 = (126/120.3)*40.8
Wfree3 = 100-W2
MS2 = (W2+40.80)*Wfree2/Wfree3  // crystals of MgSO4
    .7H2O
y = MS1-MS2 //kg
q = 501.2*1000/284.6 // quantity of original sol to
    be fed
printf(" Quantity if original solution to be fed to
    the crystallizer per 1000kg crystals of MgSO4.7
    H2O = "+string(q)+"kg.")
```

Scilab code Exa 6.11 Recovery of p DCB

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.11
// Page 370
printf("Example 6.11, Page 370 \n \n");

// solution

// (a)
```

```
16 printf("(a) \n \n ")
17 // using fig 6.12
18 // peforming material balance at 290K
19 a1 = 5.76
20 b1 = 4.91
21 DCBs = b1*100/(a1+b1) // % of solid separated p-DCB
22 DCBr1 = DCBs*100/70 // recovery of p-DCB
23 printf("Percentage recovery of p-DCB = "+string(
      DCBr1)+". \n \n \n (b) \n \n ")
24
25 //(b)
26 //at 255K
27 a2 = 5.76
28 b2 = 10.22
29 DCBs = b2*100/(a2+b2)
30 DCBr2 = (DCBs*100)/70
31 Ar = DCBr2-DCBr1
32 printf("Additional recovery of p-DCB = "+string(Ar)+
      ".")
```

Scilab code Exa 6.12 Extractive Crystallization of o and p nitrochlorobenzenes

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 6
6 // Stoichiometry and Unit Operations
7
8
9 // Example 6.12
10 // Page 371
11 printf("Example 6.12, Page 371 \n \n");
12
```

```
// solution
F = 5000 //kg/h  solvent free mix fed to simple
    crystallization unit
B1 = 4000/157.5 // kmol/h   p-NCB in feed
A1 = 1000/157.5 // kmol/h   o-NCB in feed
// after crstallization mother liquor has 33.1 mol %
    B, A doesn't crstallizes
m = A1/(1-.331)   // mother liquor entering
    extractive crytallization unit
B2 = m-A1
// optimizing solid flux
// dCt/dR = 1 - 2/R^3 = 0
R = 2^(1/3)
// referring fig 6.14
// overall material balance
// p-isomer (B)
// .98D + xT = 4000      (i)
// o-isomer (A)
// .02D + (1-.05-x)T = 1000      (ii)
// material balance around solvent recovery unit
// B
// 2.26Tx = .198G = xH          (iii)
// A
// 2.26T(.95-x) = .531G          (iv)
// solving above eq
T = 1337.6 // kg/h
D = 3729 // kg/h
G = 3939 // kg/h
x = .258
//putting these values we get composition of various
    streams
printf(" Composition of various streams : \n
   Component              T kg/h                D kg/h\n
       A                  925.6                 74.6 \n
       B                  345.1                 3654.9 \n
        C                  66.9                  nil \n \
   n")
printf(" Purity of top product = 69.2 percent A \n
```

```
Purity of bottom product = 98.0 percent \n Make-
up solvent = 66.9 kg/h.")
```

Scilab code Exa 6.13 Calculation of Dew Point

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.13
// Page 383
printf("Example 6.13, Page 383 \n \n");

// solution

Pw1 = 12.84 //Pa  v.p. of ice at 233.15K  (table
   6.12)
P1 = 101325 //Pa
Hm = (Pw1/(P1-Pw1))   // kmol/kmol dry air
P2 = 801325 //Pa
Pw2 = P2*.0001267/(1+.0001267)
dp = -20.18 + 273.15 //K     from table 6.12
printf("Dew Point = "+string(dp)+"K.")
```

Scilab code Exa 6.14 Calculations on Ambient Air

```
clear;
clc;

```

```
4 // Stoichiometry
5 // Chapter 6
6 // Stoichiometry and Unit Operations
7
8
9 // Example 6.14
10 // Page 384
11 printf("Example 6.14, Page 383 \n \n");
12
13 // solution
14
15 //Pa = v.p. at DP
16 Pw = 2.0624 //kPa
17 P = 100 //kPa
18 Hm = Pw/(P-Pw)    // kmol water vapour / kmol dry air
19 H = .622*Hm  // kg moisture/kg dry air
20 // at saturation, DB = WB = DP
21 Ps = 4.004 //kPa
22 RH = Pw*100/Ps
23 Hs = (Ps/(P-Ps))*.622
24 s = H*100/Hs
25 Ch = 1.006+1.84*H //kJ/kg dry air K
26 Vh = (.00073+.03448)*22.414*1.1062*1.0133  //m^3/kg
   dry air
27 // using fig 6.15
28 WB = 294.55 //K
29 ias = 62.3 // kJ/kg dry air
30 d = -.28 // kJ/kg dry air
31 ia = ias + d
32 printf("The absolute molar humidity = "+string(Hm)+"
    kmol water vapour/kg dry air \nAbsolute humidity
    = "+string(H)+" kg moisture/kg dry air \npercent
    RH = "+string(RH)+" \npercent saturation = "+
   string(s)+" \nHumid heat = "+string(Ch)+" kJ/kg
   dry air K\nHumid volume = "+string(Vh)+" m^3/kg
   dry air.")
```

Scilab code Exa 6.15 Humidification of Air in a Textile Industry

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.15
// Page 385
printf("Example 6.15, Page 385 \n \n");

// solution

//basis 1kg of dry air entering the air washer
//from fig 6.15
H1 = 11.8  //g/kg dry air
H2 = 17.76  //g/kg dry air
H = H2-H1  // moisture added during saturation
DB = 300.95 //K
WB = 298.15 //K
DP = 297.15 //K
Ch = 1.006+1.84*.01776 //kJ/kg dry air K
dT = DB-DP
Hs = Ch*3.8
A = 25000 //m^3/h    actual air at 41 and 24 degree
   celcius
// again from fig 6.15
Vh = .9067  //m^3/kg dry air
qm = A/Vh  //kg dry air/h
fi = qm*Hs //kJ/h
P = 300 //kPa
```

```
32 lamda= 2163.2 //kJ/kg          by appendix IV.2
33 SC = fi/lamda  //kg/h      steam consumption at the
      heater
34 printf(" the moisture added to the air = "+string(H)
      +" g/kg dry air \n DB temp of final air = "+
      string(DB)+"K \n WB temp of final air = "+string(
      WB)+"K \n The heating load of the steam coil per
      kg dry air = "+string(fi)+" kJ/h \n Steam
      consumption = "+string(SC)+" kg/h.")
```

Scilab code Exa 6.16 Induced draft cooling tower

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 6
6  // Stoichiometry and Unit Operations
7
8
9  // Example 6.16
10 // Page 387
11 printf("Example 6.16, Page 387 \n \n");
12
13 // solution
14
15 // M = E+B+W
16 Tav = (45+32)/2 +273.15 //K   avg cooling water T
17 // using steam tables (Appendix A IV.1)
18 lamda = 2410.5  //kJ/kg
19 E = 530/lamda  //kg/s
20 Cl = 4.1868
21 Ti = 45+273.15 //K
22 To = 32+273.15 //K
23 fi = 530 // = mc*Cl*(Ti-To)
```

```
mc = 530/(Cl*(Ti-To))     //kg/s
W = .3*mc/100  //kg/s
// dissolved solid balance
// M*xm = (B+W)*xc
// 500*10^-6*M = (B+.0292)*2000*10^-6
// solving above eqs
B = .0441 //kg/s
M = .2932 //kg/s
//energy balance on cooling tower
// fi = ma*(i2-i1)
// i2-i1 = 11.042 kJ/kg dry air
// moisture balance
//E = ma(H2-H1)
H2 = .2199/48 + .0196
iws = 2546.2 // Appendix IV
Ch1 = 1.006+1.84*.0196
i1 = 1.006*(297.45-273.15)+.0196*iws
   +1.042*(308.15-297.5) // kJ/kg dry air
i2 = i1 + 11.04
Tdb = ((i2 - 1.006*(301.25-273.15)-iws*H2)/1.05)
   +301.25 // K
printf("Air leaves th induced draft fan at "+string(
   Tdb)+" K.")
```

Scilab code Exa 6.17 Waste Heat recovery unit

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.17
```

```
10 // Page 389
11 printf("Example 6.17, Page 389 \n \n");
12
13 // solution
14
15 // basis 1 kg dry air fed to tower
16 // from fig 6.16 we get
17 // at WB=330 K and DB=393 K
18 H1 = .0972 // kg/kg dry air
19 DP = 325.15 //K
20 // at 313 K
21 H2 = .0492 // kg/kg dry air
22 H = H1-H2 // moisture condensed in tower
23 Ch1 = 1.006 + 1.84*H1 // kJ/kg dry air
24 Ch2 = 1.006 + 1.84*H2
25 ia1 = 1.006*(325-273) + H1*2596 + 1.185*(393-325) //
     enthalpy of entering air
26 ia2 = 1.006*(313-273) + H2*2574.4 // enthalpy of
    outgoing air
27 i = ia1-ia2
28 qm = 2000/(1+H1)
29 fi1 = qm*i // heat loss rate
30 fi2 = 1.167*3600*4.1868*(323-305) // heat gained by
    water
31 r = fi2*100/fi1
32 printf("(a) \n \n The heat loss rate rate from the
    hot air in the bed = "+string(fi1)+" kW \n \n \n(
    b) \n \n The percentage heat recovery in hot
    water = "+string(r)+" percent.")
```

Scilab code Exa 6.18 Recovery of CS2 by adsorption

```
1 clear;
2 clc;
3
```

```
4 // Stoichiometry
5 // Chapter 6
6 // Stoichiometry and Unit Operations
7
8
9 // Example 6.18
10 // Page 390
11 printf("Example 6.18, Page 390 \n \n");
12
13 // solution
14
15 // basis 800 kmol of inlet CS2-H2 mix
16 Pi = 106.7 //kPa  Total Pressure
17 Pcs2i = 16.93 // kPa
18 n = 800 // kmol
19 ncs2i = Pcs2i*n/Pi // kmol
20 nh2i = n-ncs2i
21 Po = 101.325 // kPa
22 Pcs2o = 6.19 // kPa
23 nh2o = 673.1 // kmol
24 ncs2o = Pcs2o*nh2o/(Po-Pcs2o)
25 ncs2a = ncs2i-ncs2o
26 mcs2a = ncs2a*76.1407 //kg
27 r = 600 // kg/h  design adsorption rate
28 Mi = n*r/mcs2a // kmol/h
29 Vi = Mi*22.843 // m^3/h
30 mcs2ac = .32-.04 // kg  CS2 absorbed per kg BD
      activated carbon
31 qm = r*1.04/mcs2ac  // kg/h
32 C = ncs2o/nh2o // kmol CS2/kmol H2   = Pcs2/(P-Pcs2)
33 Pcs2 = 24.763 // kPa
34 T = 281.5  //K  by eq 5.24
35 printf("(a) \n \n Volumetric flowrate of entering
      mixture = "+string(Vi)+" m^3/h \n \n \n(b) \n \n
      Mass flowrate of activated carbon = "+string(qm)+
      " kg/h \n \n \n(c) \n \n Original mixture must be
       coole to "+string(T)+" K at 405 kPa for
      achieving same concentration of the outlet
```

```
    mixture with adsorption.")
```

Scilab code Exa 6.19 Hooker type diaphragm cell

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.19
// Page 391
printf("Example 6.19, Page 391 \n \n");

// solution

// basis 4000 kg/h of NaOH produced
Cl2p = 35.5*2*4000/80 // kg/h
Mcl2 = Cl2p/71 // kmol/h
P = 101.325 // kPa
Pw = 2.0624 // kPa
moist = (Pw/(P-Pw))*(18.0154/70.906) //
Tmoist = Cl2p*moist  // kg/h
// for 90% onc of acid
n = (10/18.0153)/(90/98.0776) /// kmol H2O/kmol acid
Q = 134477/(18*(n+1.7983)^2) //kJ/kg H2O   by eq (ii
    )
lambdav = 2459  // kJ/kg  (Appendix IV)
heatload = Q+lambdav
fi = heatload*18.74 //kJ/h
printf(" The heat liberation rate in the tower = "+
    string(fi)+" kJ/h.")
```

Scilab code Exa 6.20 Absorption of NH3 from pure gas

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 6
6  // Stoichiometry and Unit Operations
7
8
9  // Example 6.20
10 // Page 393
11 printf("Example 6.20, Page 393 \n \n");
12
13 // solution
14
15 // basis 100 kmol of feed gas
16 // using table 5.1
17 Sniai =
      20.6*29.5909+62*28.6105+4.1*20.7723+11.1*19.2494+2.2*25.6503

18 Snibi = [20.6*(-5.141)
      +62*1.0194+11.1*52.1135+2.2*33.4806]/1000
19 Snici = [20.6*13.1829+62*(-.1476)
      +11.1*11.973+2.2*.3518]/10^6
20 Snidi = [20.6*(-4.968)+62*.769+11.1*(-11.3173)
      +2.2*(-3.0832)]/10^9
21 Hgas = Sniai*(283-263) + Snibi*(283^2-263^2)/2 +
      Snici*(283^3-263^3)/3 + Snidi*(283^4-263^4)/4  //
      kJ
22 Hnh3 = 1533.8  //kJ
23 SniCmpi = (Hgas-Hnh3)/20 // kJ/(K 97.8 kmol gas)
      NH3 free gas
24 Go = 97.8/.99995  //kmol
```

```
25 NH3a = (2.2-.005)*17  // kg
26 F1 = NH3a/.04 // flowrate of 4% NH3 solution
27 Water = F1-NH3a //kg
28 dT1 = Hgas/(Water*4.1868)  // K
29 Twater = 307-dT1  //K
30 Wvp = 2.116  //kPa
31 P = 5101.325  //kPa
32 moist = Go*Wvp/(P-Wvp) // kg
33 W = Water + moist // total demineralised water
34 dTactual = Hgas/(W*4.1868) //K
35 // from table 5.59
36 dHf1 = -80.093  //kJmol NH3  of 4% NH3 sol
37 dHf2 = -46.11  //kJ/molNH3
38 H = dHf1-dHf2  // heat of 4% NH3 sol
39 Hevl = -(H*NH3a*1000)/17  // total heat evolved
40 // in absorber gas is further heated from 283K to
      291.4K
41 Hsol = Hevl-(2854.1*(291.4-283.15)) // kJ
42 // c 0f 4% NH3 sol = c of water = 4.1868 kJ/kg K
43 dT2 = Hsol/(F1*4.1868)
44 To = 291.4+dT2
45 printf("(a) \n \n Temp of feed water to absorber = "
      +string(Twater)+"K. \n \n \n(b) \n \n Temp of aq
      NH3 sol leaving the absorber = "+string(To)+"K.")
```

Scilab code Exa 6.21 Direct contact counter current rotary drier

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 6
6 // Stoichiometry and Unit Operations
7
8
```

```
// Example 6.21
// Page 396
printf("Example 6.21, Page 396 \n \n");

// solution

// basis : product rate of 100 kg/h
H1 = .036 // kg moist/ kg dry solid
X1 = .25/.75 // kg /kg dry solid
X2 = .02/.98 // kg/kg dry solid
// moist balance
// ms*(X1-X2) = ma*(H2-H1)
To = 273.15 //K
is1 = 1.43*(30-0)+X1*4.1868*30
is2 = 1.43*80+.0204*4.1868*80
Tdb = 393.15 //K
Tdp1 = 308.15 //K
iwb1 = 2565.4  //kJ/kg
Ch1 = 1.006+1.84*.036
ia1 = 1.006*(Tdp1-273.15)+H1*iwb1+Ch1*(Tdb-Tdp1)
H2 = .056
Tdp2 = 315.55
iwb2 = 2578.7
ia2 = 1.006*(Tdp2-273.15)+H2*iwb2+(1.006+1.84*H2)
   *(323.15-Tdp2)
ma = .085/(.056-.036)
iaa = 1.006*(Tdp1-273.15)+H1*iwb1
fi = 4.25*(218.68-iaa)  //kW
lambda = 2133.0
steam = fi/lambda // kg/h
printf("(a) \n \n Flowrate of incoming air on dry
   basis = "+string(ma)+" kg/s \n \n \n(b) \n \n
   Humidity of air leaving the drier = "+string(H2)+
   " kg/kg dry air. \n \n \n(c) \n \n Steam
   consumption in the heater = "+string(steam)+" kg/
   h.")
```

Scilab code Exa 6.22 Hot air dryer of textile mill

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.22
// Page 398
printf("Example 6.22, Page 398 \n \n");

// solution
// basis cloth speed = 1.15 m/s
prod = 1.15*1.2*3600*.095
moisti = .90 // kg/kg bone dry cloth
moisto = .06
evp = 471.96*(moisti-moisto)
// using fig 6.15 and 6.16
H1 = .01805
H2 = .0832
dH = H2-H1
qm1 = evp/dH // kg dry air/h
Vh = .8837 //m^3/kg dry air
qv = qm1*Vh
DP1 = 296.5 //K
DP2 = 322.5 //K
lambdaV2 = 2384.1 //kJ/kg
To = 273.15 //K
fi1 = prod*1.256*(368-303)+prod*.06*(368-303)*4.1868
    // kJ/h
fi2 = evp*(322.5-303.15)+evp*lambdaV2 //kJ/h
```

```
32 ia1 = 1.006*(303.15-273.15)+2556.4*.01805 //kJ/kg
     dry air
33 ia2 = 1.006*(322.8-273.15)
     +2591.5*.0832+(1.006+1.84*.0832)*(393-328.8)
34 fi2 = ia2-ia1
35 hlost = fi2-fi1 // kJ/h
36 // using Appendix IV
37 h = 720.94 //kJ/kg
38 lambdav = 2046.5 // kJ/kg
39 steami = (h+lambdav)*885 // kJ/h
40 fi4 = h*885 //kJ/h
41 qm2 = 885/evp
42 printf("(a) \n \n Bone dry production of the dryer =
     "+string(prod)+" kg/h. \n \n \n(b) \n \n The
     evaporation taking place in the dryer = "+string(
     evp)+" kg/h. \n \n \n(c) \n \n The air
     circulation rate = "+string(qv)+" m^3/h.")
```

Scilab code Exa 6.23 Quadruple effect forward feed evaporator

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 6
6 // Stoichiometry and Unit Operations
7
8
9 // Example 6.23
10 // Page 401
11 printf("Example 6.23, Page 401 \n \n");
12
13 // solution
14
15 // basis : weak liquor flowrate = 1060 kg/h
```

```
16 s1 = 1060*.04 //kg/h   solids in weak liquor
17 liqr = s1/.25 // kg/h  conc liquor leaving 4th
      effect
18 evp = 1060-liqr // kg/h
19 lambdas = 2046.3  // kJ/kg
20 Wf = 1060 // kg/h
21 C1f = 4.04
22 T1 = 422.6
23 Tf = 303
24 lambdav1 = 2114.4
25 // enthalpy balance of 1st effect
26 // Ws*lambdas = Wf*C1f*(T1-Tf) + (Wf-W1)*2114.4
27 //putting values we get
28 // Ws = 1345.57 - 1.033*W1
29 // 2nd effect
30 // W1 = 531.38+.510*W2
31 // 3rd effect
32 // W1 - 1.990*W2 = -1.027*W3
33 // 4th effect
34 // W2 - 1.983*W3 = -176.84
35 //solving above eqs
36 W1 = 862 // kg/h
37 W2 = 648.2 // kg/h
38 W3 = 416.7 // kg/h
39 Ws = 455.2 // kg/h
40 eco = evp/Ws  // kg evaporation/kg steam
41 spcon = 1/eco // kg steam/kg evaporation
42 printf("Specific heat consumption of the system is "
      +string(spcon)+" kg steam/kg evaporation.")
```

Scilab code Exa 6.24 Triple effect evaporation system

```
1 clear;
2 clc;
3
```

```
// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.24
// Page 403
printf("Example 6.24, Page 403 \n \n");

// solution

Fspd1 = 4300 // kg/h
Bcrtn = Fspd1*600*10^-6 // kg/h
Fspd2 = Bcrtn/.00645 // kg/h
evp1 = Fspd1-Fspd2
Fspd3 = Bcrtn/.057
evp2 = Fspd2-Fspd3
C3 = Bcrtn/.4
evp3 = Fspd3-C3
fi1 = Fspd1*2.56*(468.15-373.15)+3900*450 // kJ/h
fi2 = Fspd2*2.56*(463.15-468.15)+354.737*450 // kJ/h
fi3 = Fspd3*2.56*(453.15-463.15)+38.813*450 // kJ/h
fi = fi1+fi2+fi3
mt = fi/(2.95*(503.15-478.15)) // kg/h
qt = mt/.71 // l/h
mccw1 = 1755000/(8*4.1868)  // kg/h
mccw2 = mccw1*.9
dT2 = 159632/(mccw2*4.1868)
mccw3 = mccw1-mccw2
dT3 = 17466/(mccw3*4.1868)
dT = (1755000+159632+17466)/(mccw1*4.1868)
Fw = 1932098/(8*4.1868) // kg/h
printf("By mass balance, required cooling water flow
    in external cooler = "+string(Fw)+" kg/h.\n\nBy
   enthalpy balance, overall rise in CCW temperature
    = "+string(dT)+" K.")
```

Scilab code Exa 6.25 Four compartment washing thickner

```
clear;
clc;

// Stoichiometry
// Chapter 6
// Stoichiometry and Unit Operations


// Example 6.25
// Page 405
printf("Example 6.25, Page 405 \n \n");

// solution

//stream M2
Vcaco3M2 = .349/2.711
VliqrM2 = .651/1.167
VslryM2 = Vcaco3M2+VliqrM2
spgM2 = 1/VslryM2
FsM2 = 2.845*3600*spgM2
sM2 = FsM2*.349 // kg/h
liqrM2 = FsM2*.651
Na2OM2 = liqrM2*.1342/1.167
//stream O2
FsO2 = 14.193*3600*1.037 // kg/h
sO2 = FsO2*.0003
liqrO2 = FsO2-sO2
Na2OO2 = liqrO2*.0272/1.037
//stream M1
VM1 = .194/2.711 + .806/1.037 // l
spgM1 = 1/VM1
FsM1 = 5206.9/.194
```

```
liqrM1 = FsM1 - 5206.9
Na2OM1 = liqrM1*.0252/1.034
// stream O1
FsO1 = FsO2+FsM1-FsM2
sO1 = FsO1*.0002
liqrO1 = FsO1 - sO1
Na2OO1 = liqrO1*.0096/1.014
// stream W
VW = .037/2.711 + .963
spgW = 1/VW
FsW = 14.977*3600*spgW
sW = FsW*.037
liqrW = FsW-sW
Na2OW = liqrW*.0024
// stream Mo
VMo = .402/2.711 + .598/1.022
spgMo = 1/VMo
FsMo = 3.627*3600*spgMo
sMo = FsMo*.402
liqrMo = FsMo - sMo
Na2OMo = liqrMo*.0162/1.022
printf(" Material balance thickener \n \n ITEM
                        STREAM, kg/h\n
                            M2                  O2
                M1                      O1                  W
                Mo\n Slurry                     "+string(
    FsM2)+"       "+string(FsO2)+"       "+string(FsM1)
    +"       "+string(FsO1)+"       "+string(FsW)+"
         "+string(FsMo)+"\n Suspended solids     "+
    string(sM2)+"       "+string(sO2)+"       "+string(
    sM2)+"       "+string(sO1)+"       "+string(sW)+"
         "+string(sMo)+"\n Liquor                    "+
    string(liqrM2)+"       "+string(liqrO2)+"       "+
    string(liqrM1)+"       "+string(liqrO1)+"          "
    +string(liqrW)+"       "+string(liqrMo)+" \n Na2O
                    "+string(Na2OM2)+"       "+string(
    Na2OO2)+"       "+string(Na2OM1)+"       "+string(
    Na2OO1)+"       "+string(Na2OW)+"       "+string(
```

```
Na2OMo)+"")
```

Chapter 7

Combustion

Scilab code Exa 7.1 GCV and NCV calculations

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 7
6  // Combustion
7
8
9  // Example 7.1
10 // Page 434
11 printf("Example 7.1, Page 434 \n \n");
12
13 // solution
14
15 // basis 100 kg as received coal
16 O2 = 18.04 //kg
17 nH2 = 2.79-(O2/8) //kg
18 printf("(a) \n \n Net H2 in coal = "+string(nH2)+"
      kg. \n \n \n (b) \n \n ")
19 cbW = 1.128*18 // kg
20 printf("Combined water in the coal = "+string(cbW)+"
```

```
    kg. \n \n \n (c) \n \n ")
// Dulong's formula
GCV1 = 33950*(50.22/100) + 144200*nH2/100 +
   9400*.37/100  // kJ/kg
printf("GCV by Dulongs formula = "+string(GCV1)+" kJ
   /kg. \n \n \n (d) \n \n ")
tH2 = 1.395  // kmol
wp = tH2*18 + 7
Hv = 2442.5*wp/100  // kJ/kg fuel
GCV2 = 23392*(1-.21-.07) // as of received coal
NCV = GCV2-Hv
printf("NCV of the coal = "+string(NCV)+" kJ/kg. \n
   \n \n (e) \n \n ")
// Calderwood eq
// Total C = 5.88 + .00512(B-40.5S) +-
   .0053[80-100*(VM/FC)]^1.55
C = 5.88 + .00512*(7240.8-40.5*.37)
   +.0053*[80-56.52]^1.55
printf("Total Carbon by Calderwood eq = "+string(C)+
   ".")
```

Scilab code Exa 7.2 NCV of crude oil

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.2
// Page 436
printf("Example 7.2, Page 436 \n \n");

```

```
13 // solution
14
15 // basis 1 kg crude oil
16 H2 = .125 // kg   burnt
17 H2O = H2*18/2
18 Lh = H2O*2442.5 //kJ
19 GCV = 45071
20 NCV = GCV-Lh  //kJ/kg oil
21 printf("NCV = "+string(NCV)+" kJ/kg.")
```

Scilab code Exa 7.3 Gaseous propane

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 7
6 // Combustion
7
8
9 // Example 7.3
10 // Page 444
11 printf("Example 7.3, Page 444 \n \n");
12
13 // solution
14
15 // basis 1 mol of gaseous propane
16 H2O = 4*18.0153  //g
17 NHV = 2219.17-(H2O*2442.5/1000)
18 printf("NHV = "+string(NHV)+" kJ/mol.")
```

Scilab code Exa 7.4 GCV NCV for natural gas

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.4
// Page 444
printf("Example 7.4, Page 444 \n \n");

// solution

// basis 1 mol of natural gas
// using table 7.7
H2O = [2*.894+3*.05+.019+5*(.004+.006)]*18  // g
Hv = H2O*2442.5/1000
NCV1 = 945.16-Hv
GCV = 945.16*1000/18.132
NCV = NCV1*1000/18.132
printf(" GCV = "+string(GCV)+" kJ/kg. \n NCV = "+
   string(NCV)+" kJ/kg.")
```

Scilab code Exa 7.5 Coal burnt in excess air

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.5
```

```
10 // Page 451
11 printf("Example 7.5, Page 451 \n \n");
12
13 // solution
14
15 // basis 100 kg fuel
16 O2req = 4.331*32  // kg
17 rO2req = O2req/100
18 N2in = (79/21)*4.331  // kmol
19 AIRreq = O2req+N2in*28  //kg
20 rAIRreq = AIRreq/100
21 R = AIRreq/100
22 AIRspld = R*2  // kg/kg coal
23 O2spld = 4.331*2  // kmol
24 N2spld = N2in*2
25 N2coal = 2.05/28  // kmol
26 tN2 = N2spld+N2coal
27 moist = 1.395+(7/18)  // kmol
28 printf("(a) \n \n Theoratical O2 requirement per
   unit mass of coal = "+string(rO2req)+" kg. \n \n
   \n(b) \n \n Theoratical dry air requirement = "+
   string(rAIRreq)+" kg/kg coal.")
```

Scilab code Exa 7.6 Residue fuel oil sample

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 7
6 // Combustion
7
8
9 // Example 7.6
10 // Page 452
```

```
11 printf("Example 7.6, Page 452 \n \n");
12
13 // solution
14
15 // basis 100 kg of RFO
16 O2req = 9.786 //kmol
17 N2req = (79/21)*O2req  //kmol
18 AIRreq = O2req+N2req //kmol
19 rAIRreq = AIRreq*29/100
20 AIRspld = AIRreq*1.25
21 rAIRspld = AIRspld/100
22 // using table 7.11 and 7.12
23 xSO2 = .07/(55.925+5.695)  // kmol SO2/kmol wet gas
24 vSO2 = xSO2*10^6  // ppm
25 mSO2 = 4.48*10^6/(1696.14+102.51)
26 // at 523.15 K and 100.7 kPa
27 V = [(55.925+5.695)*8.314*523.15]/100.7  // m^3
28 cSO2 = (4.48*10^6)/V  // mg/m^3
29 //from fig 7.3
30 dp = 424.4 //K
31 printf("(a) \n \n Theoretical air required = "+
   string(rAIRreq)+" kg/kg fuel. \n \n \n(b) \n \n
   Actual dry air supplied = "+string(rAIRspld)+" kg
   /kg fuel. \n \n \n(c) \n \n Concentration of SO2
   = "+string(mSO2)+" mg/kg. \n \n \n(d) \n \n
   Concentration of SO2 = "+string(vSO2)+" ppm vol/
   vol. \n \n \n(e) \n \n Concentration of SO2 if
   gases are discharged at 523.15K and100.7kPa = "+
   string(cSO2)+" mg/m^3. \n \n \n(f) \n \n Dew
   Point of flue gas = "+string(dp)+" K.")
```

Scilab code Exa 7.7 Orsat analysis of flue gases

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

```
3
4 // Stoichiometry
5 // Chapter 7
6 // Combustion
7
8
9 // Example 7.7
10 // Page 454
11 printf("Example 7.7, Page 454 \n \n");
12
13 // solution
14
15 // basis 100 kmol of dry flue gas
16 O2acntd = 11.4+4.2  // kmol
17 O2avlbl = (21/79)*84.4 // kmol
18 O2excs = 4.2 //kmol
19 O2unactd = O2avlbl-O2acntd
20 H2brnt = O2unactd*2
21 O2req = 11.4+O2unactd
22 pexcsAIR = O2excs*100/O2req
23 mH2brnt = H2brnt*2 // kg
24 mCbrnt = 11.4*12
25 r = mCbrnt/mH2brnt
26 printf("(a) \n \n Percent excess air = "+string(
   pexcsAIR)+". \n \n \n(b) \n \n In fuel C:H = "+
   string(r)+".")
```

Scilab code Exa 7.8 Sugar factory boiler

```
1 clear;
2 clc;
3
4 // Stoichiometry
5 // Chapter 7
6 // Combustion
```

```
7
8
9  // Example 7.8
10 // Page 459
11 printf("Example 7.8, Page 459 \n \n");
12
13 // solution
14
15 // basis 100 kg of bagasse fired in th boiler
16 //(a)
17 O2req = 2.02 // kmol
18 N2in = (79/21)*O2req // kmol
19 AIRreq = (O2req+N2in)*29 // kg
20 rAIR = AIRreq/100
21 printf("(a) \n \n Theoretical air required = "+
      string(rAIR)+" kg dry air/kg fuel. \n \n \n (b) \
      n \n ")
22 // (b)
23 tflugas = 1.95/.1565  ///kmol
24 xcsO2N2 = tflugas - 1.95
25 x = (xcsO2N2-7.6)/4.76  // kmol
26 pxcsAIR = x*100/O2req
27 printf("Percent excess air = "+string(pxcsAIR)+". \n
       \n (c) \n \n ")
28 //(c)
29 pW = 100*.2677 // kPa    partial p of water vap
30 // from fig 6.13
31 dp = 339.85 //K
32 printf("Dew Point of flue gas = "+string(dp)+"K. \n
      \n \n (d) \n \n ")
33 // (d)
34 // from appendix IV
35 hfw = 292.97  //kJ/kg     enthalpy of feed water at
      343.15 K
36 Hss = 3180.15  // kJ/kg   enthalpy of super heated
      steam at 2.15 bar and 643.15K
37 Hgain = Hss - hfw
38 H6 = Hgain*2.6*100  // kJ    heat gained by water
```

```
39 H1 = 100*1030000  // kJ
40 GCV = H6*100/H1
41 printf("Thermal efficiency of the boiler = "+string(
   GCV)+".")
```

Scilab code Exa 7.9 Stoker fired water tube boiler

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 7
6  // Combustion
7
8
9  // Example 7.9
10 // Page 465
11 printf("Example 7.9, Page 465 \n \n");
12
13 // solution
14
15 // using mean heat capacity data Table 7.21
16 // basis 100 kmol of dry flue gas
17 H7 = 1.0875*100*30.31*(423.15-298.15)
18 H71 = 3633.654*(423.15-298.15)
19 fi7 = H71*3900*.7671/162.2  // kJ/h
20 fi1 = 3.9*1000*26170    // kJ/h
21 // performing heat balance
22 Hsteamgen = 23546.07
23 eff = Hsteamgen*100/fi1  // overall efficiency rate
24 printf("Overall efficiency rate = "+string(eff)+"
   percent.")
```

Scilab code Exa 7.10 Atomization of fuel

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.10
// Page 468
printf("Example 7.10, Page 468 \n \n");

// solution

// basis 100 kg of fuel oil
O2req =  9.364  // kmol
N2in = (79/21)*O2req
tN2 = N2in+.036
AIRreq = O2req*32 + tN2*28
rAIR = AIRreq/100   // kg/kg
wp = 4.5  // kmol
Hloss = 2442.8*wp*18/100  // kJ/kg fuel
NCV = 43540-Hloss
printf("(a) \n \n NCV = "+string(NCV)+" kJ/kg. \n \n
    \n (b) \n \n Theoretical air required = "+string
   (rAIR)+" kg/kg fuel. \n \n \n (c) \n \n ")
H1 = 100*41561.33  // kJ
// from table 5.1
H71 = 1349.726*(1500-298.15)+252.924*10^-3 *
   ((1500^2-298.15^2)/2)
   +257.436*10^-6*((1500^3-298.15^3)/3)
   -137.532*10^-9*((1500^4-298.15^4)/4)  // upto
   1500 K
H711 = H1-H71  // above 1500K
// F(T) = {1500 to T} integr[1477.301+375.2710*10^-3
   T-91.2760*10^-6T^2+8.146*10^-9T^3]dT-2147118
```

```
    (i)
// solving it for T = 2000
AFT = 2612.71 // K
printf("When fluid is burnt with theoretical air AFT
    = "+string(AFT)+" K. \n \n \n (d) \n \n ")
// with 30% excess air
O2spld = 9.364*1.3
xcsO2 = O2spld-O2req
N2in1 = (79/21)*O2spld
tN21 = N2in1+.036
// now, using table 7.26, table 7.27 and eq(i) we
   get
AFT1 = 2178.66  // K
// from fig 7.3
dp = 429  // K
// similarly for incomplete combustion we find
AFT2 = 2561.42 //K
printf("When 30 percent excess air is supplied AFT =
    "+string(AFT1)+" K. \n \n \n (d) \n \n Dew Point
    = "+string(dp)+" K. \n \n \n (e) \n \n For
   incomplete combustion AFT = "+string(AFT2)+" K.")
```

Scilab code Exa 7.11 Water tube boiler

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.11
// Page 473
printf("Example 7.11, Page 473 \n \n");
```

```
12
13 // solution
14
15 // basis 100 kg of fuel
16 // material balance of carbon
17 CO2 = 7.092+.047 //kmol   in flue gases
18 N2 = 11.94*7.139/7.01
19 O2 = 11.94*7.139/7.01
20 flue = CO2+N2+O2
21 // material balance of O2
22 O2air = 21*N2/79
23 airin = N2+O2air
24 tO2in = O2air+.078  // O2 in burner
25 O2xcs = tO2in-9.864
26 // material balance of water vapour
27 moistfrmd = 5.45 // kmol    from combustion of H2
28 H = .0331 // kmol/kmol of dry air        humidity at
       100.7 kPa
29 moistair = H*104.482  //kmol
30 tmoist = moistfrmd+moistair
31 pxcsair = O2xcs*100/9.786
32 // now using table 7.32
33 H7 = 3391.203*(563.15-298.15)  //kJ
34 Ff = 400 // kg/h   fuel firing rate
35 tH = 2791.7-179.99 // kJ/kg    total heat supplied
      in boiler
36 fi5 = tH*4365  // kJ/h
37 fi8 = 5.45*18*Ff*2403.5/100  // kJ/h
38 GCVf = 42260  //kJ/kg
39 fi1 = Ff*GCVf
40 Fdryair = 104.482*29*Ff/100
41 Cha = 1.006+1.84*.0205  // kJ/kg dry air K
42 fi3 = Fdryair*Cha*(308.15-298.15)
43 fi2 = Ff*1.758*(353.15-298.15)
44 BOILEReff1 = fi5*100/fi1
45 NCVf = GCVf-(18.0153/2.016)*.109*2442.8  // kJ/kg
46 BOILEReff2 = fi5*100/(Ff*NCVf)
47 r = 4365/Ff // steam:fuel
```

```
48 BOILERcapacity = fi5/2256.9
49 printf(" After performing material and thermal
   balance operations we get \n  \n Overall thermal
   efficiency of the boiler based on GCV of the fuel
    = "+string(BOILEReff1)+" percent. \n \n Overall
   efficiency of the boiler based on NCV of the fuel
    = "+string(BOILEReff2)+" percent. \n \n Steam to
    fuel ratio = "+string(r)+" at 16 bar. \n \n
   Equivalent boiler capacity = "+string(
   BOILERcapacity)+" kg/h.")
```

Scilab code Exa 7.12 Gassification by coal

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 7
6  // Combustion
7
8
9  // Example 7.12
10 // Page 478
11 printf("Example 7.12, Page 478 \n \n");
12
13 // solution
14
15 // basis 100 kmol of dry producer gas
16 C = 33*12 // kg
17 O2 = 18.5*32  //kg
18 H2 = 20*2 // kg
19 O2air = 21*51/79 // kmol
20 COALgassified = 396/.672  // kg
21 O2coal = COALgassified*.061/32  // kmol
22 tO2 = O2coal + O2air
```

```
23 O2steam = 18.5-tO2  // kmol
24 H2steam = 2*O2steam  // kmol
25 H2fuel = 20-H2steam
26 dryproducergas = 100*22.41/COALgassified  // Nm^3/kg
       coal
27 Pw = 2.642 // kPa
28 Ha = Pw/(100.7-Pw)  // kmol/kmol dry gas
29 water = Ha*100
30 moistproducergas = (100+water)*22.41/COALgassified
     // Nm^3/kg coal
31 dryair = (51*28+O2air*32)/COALgassified  // kg/kg
     coal
32 tsteamsupplied = H2steam+water-(COALgassified
     *.026/18)  // kmol
33 steam = tsteamsupplied*18/COALgassified
34 printf(" (a) \n \n Moistproducer gas obtained = "+
     string(moistproducergas)+" Nm^3/kg coal. \n \n \n
      (b) \n \n Air supplied = "+string(dryair)+" kg/
     kg coal gassified. \n \n \n (c) \n \n Steam
     supplied = "+string(steam)+" kg/kg coal.")
```

Scilab code Exa 7.13 Open Hearth steel furnace

```
1  clear;
2  clc;
3
4  // Stoichiometry
5  // Chapter 7
6  // Combustion
7
8
9  // Example 7.13
10 // Page 479
11 printf("Example 7.13, Page 479 \n \n");
12
```

```
// solution

// solving by alternate method on page 483
// basis 100 kmol of dry producer gas
// using tables 7.38 and 7.39
fi7 = 6469.67*(833.15-298.15)*(27650/2672)  // kJ/h
// heat output basis 1 kg of steam
// referring Appendix IV
H4 = 675.47-272.03  // kJ/kg
Ts = 463  // K
h = 806.69  // kJ/kg
lambdav = 1977.4  // kJ/kg
Hss = 2784.1  // kJ/kg at Ts
i = 3045.6  // kJ/kg
H6 = i-Hss
fi4 = H4*7100  // kJ/h
fi5 = (Hss-675.47)*7100  // kJ/h
fi6 = H6*7100  // kJ/h
recovery = fi4+fi5+fi6
BOILERcapacity = recovery*3600/2256.9  // kg/h
fi8 = 6125.47*(478.15-298.15)*(27650/2672)  // kJ/h
hloss = fi7-fi4-fi5-fi6-fi8  /// kJ/h
printf(" Heat Balance of Waste Heat Boiler \n \n \n
                                    kJ/h \n Heat Output
    \n Steam rising \n Economiser
                      "+string(fi4)+" \n Steam
   generator                "+string(fi5)+" \n Super
   heater                   "+string(fi6)+" \n \n
   Heat loss in flue gases        "+string(fi8)+" \n
   Unaccounted heat loss          "+string(hloss)+"")
```