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

November 9, 2013

[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

```
clear;
clc;

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


// Example 1.1
// Page 12
printf("Example 1.1, Page 12 \n \n");

// solution

// Using conversion factors from table 1.3 (Pg 9)
q1 = 75 // [gallon/min] (volumetric flow rate)
q2 = 75/(60*.219969) // [dm^3/s]
row = 0.8 // [kg/dm^3]
q3 = q2*row // [kg/s] (mass flow rate)
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");

```

```
// solution

// C = 89.2*A*(T/M^).5    [ft^3/s]
k = 89.2 //
C1 = 1 // [ft^3/s]
// Using conversion factors from table 1.3 (Pg 9)
C2 = 35.31467*C1
T1 = 1 //[dgree R]
T2 = 1.8*T1 // [K]
A1 = 1 // [ft^2]
A2 = 10.76391
k2 = (k*A2*(1.8)^.5)/35.34167
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

```
clear;
clc;

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


// Example 2.1
// Page 17
printf("Example 2.1, Page 17 \n \n");

// solution

// NH4Cl
M = 14+4+35.5 // [g] (molar mass of NH4Cl)
n=5 // [mol]
m = M*n // [g]
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
```

```
// Basic Chemical Calculations


// Example 2.3
// Page 17
printf("Example 2.3, Page 17 \n \n");

// solution

// K2CO3
m = 117 // [kg] (wt of K)
Mk = 39 // [g] (at wt of K)
a = m/Mk // [kg atoms]
// 1 mol of K2CO3 contains 2 atoms of K
n = a/2 // [kmol] (moles of K2CO3)
printf(" "+string(n)+" kmol of K2CO3 contains 117 kg
    of K.")
```

Scilab code Exa 2.4 no of atoms of BaCl2

```
clear;
clc;

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


// Example 2.4
// Page 18
printf("Example 2.4, Page 18 \n \n");

// solution

// BaCl2
```

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

Scilab code Exa 2.5 equivalent mass

```
clear;
clc;

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


// Example 2.5
// Page 19
printf("Example 2.5, Page 19 \n \n");

// solution

printf("(a) \n \n")
//PO4 radical
M = 31+4*16 //[g]
V = 3 // (valence of PO4)
eqm = M/V
printf("eq. mass of PO4 is "+string(eqm)+" [g] \n \n \n")
printf("(b) \n \n")
//Na3PO4
M = 3*23+95 //[g]
V = 3
eqm = M/V
```

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

Scilab code Exa 2.6 equivalents

```
clear;
clc;

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


// Example 2.6
// Page 19
printf("Example 2.6, Page 19 \n \n");

// solution

// AlCl3
v = 3 // valency of Al ion
eq = 3*3 // [mol]
printf("no. of equivalents in 3 kmol of AlCl3 is "+
    string(eq)+" keq.")
```

Scilab code Exa 2.7 composition of mixture

```
clear;
clc;

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

```
// Example 2.7
// Page 20
printf("Example 2.7, Page 20 \n \n");

// solution

// (a)
printf("(a) \n \n")
// mass %
m1 = 600 //[kg] (NaCl)
m2 = 200 //[kg] (KCl)
m = m1+m2 // total mass
Wa = (m1/m)*100
Wb = (m2/m)*100
printf("mass percentage of NaCl is "+string(Wa)+" \nmass percentage of KCl is "+string(Wb)+" \n \n \n")
// (b)
printf("(b) \n \n")
//mol %
M1 = 23+35.5 // molar mass of NaCl
n1 = m1/M1 // no. of moles of NaCl
M2 = 39+35.5 // molar mass of KCl
n2 = m2/M2 // no. of moles of KCl
n = n1+n2
N1 = (n1/n)*100
N2 = (n2/n)*100
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

```
clear;
```

```
clc;

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


// Example 2.8
// Page 21
printf("Example 2.8, Page 21 \n \n");

// solution
// CH.35O.35S.14
// mass %
C = 12.0107 //[kg]
H = 1.00794*.35 // [kg]
O = 15.9994*.35// [kg]
S = 32.065*.14 //[kg]
m = C+H+O+S
m1 = (C/m)*100
m2 = (H/m)*100
m3 = (O/m)*100
m4 = (S/m)*100
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")
M = m/(1+.35+.35+.14)
printf("molar mass = "+string(M)+" kg/kmol.")
```

Scilab code Exa 2.9 actual urea content

```
clear;
clc;

```

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


// Example 2.9
// Page 22
printf("Example 2.9, Page 22 \n \n");

// solution

// basis 100kg urea
m1 = 45 //[kg] (mass of N present)
Mu = 60 // (molar mass of urea)
m2 = 14*2 //[kg] (mass of N in 1 kmol of urea)
m = (Mu/m2)*m1
printf("The sample contains "+string(m)+" percent
    urea.")
```

Scilab code Exa 2.10 mass percent

```
clear;
clc;

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


// Example 2.10
// Page 22
printf("Example 2.10, Page 22 \n \n");

// solution

```

```
// NaOH

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

Scilab code Exa 2.11 no of ions

```
clear;
clc;

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


// Example 2.11
// Page 22
printf("Example 2.11, Page 22 \n \n");

// solution

Ca = 40.078 // at. wt of Ca
F = 18.9984032 // at wt of F
M1 = 3*Ca +2*(30.97762+(4*15.9994)) // molar mass of
    Ca3PO4
M2 = Ca +12.0107+3*15.9994 // molar mass of CaCO3
M3 = Ca+2*F // molar mass of CaF2
m1 = 800 //[mg] Ca3PO4
m2 = 200 //[mg] CaCO3
m3 = 5 //[mg] CaF2
n1 = ((3*Ca)/M1)*m1+(Ca/M2)*m2+(Ca/M3)*m3 // [mg]
    total Ca ions
n2 = (F/M3)*2*5 //[mg] total F ions
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
```

```
n1 = m1/M1 //[kmol] methanol
n2 = m2/M2 //[kmol] salicylic acid
n = n1+n2
N1 = n1/n*100
N2 = n2/n*100
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

```
clear;
clc;

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


// Example 2.13
// Page 24
printf("Example 2.13, Page 24 \n \n");

// solution


//mass %
m1 = 13.70 // HCl
m2 = 8.67 // NaCl
m3 = 100 // H2O
m = m1+m2+m3 // mass of solution
w1 = m1/m*100
w2 = m2/m*100
w3 = m3/m*100

```

```
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")
M1=36.4609 //HCl
M2=58.4428 //NaCl
M3=18.0153 //H2O
n1=m1/M1 //HCl
n2=m2/M2 //NaCl
n3=m3/M3 //H2O
n=n1+n2+n3
N1=n1/n*100
N2=n2/n*100
N3=n3/n*100
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

```
clear;
clc;

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


// Example 2.14
// Page 24
printf("Example 2.14, Page 24 \n \n");

// solution

m = 100 //[kg] Lye (basis)
m1 = 73 //[kg] NaOH
```

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

Scilab code Exa 2.15 TOC and ThOD

```
clear;
clc;

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


// Example 2.15
// Page 25
printf("Example 2.15, Page 25 \n \n");

// solution

//(CH2OH)3
M = 92 // molar mass of glycerin
C = 600 //[mg/l] glycerin conc.
TOC = (3*12/92)*600 //[mg/l]
// by combustion reaction we see 3.5 O2 is required
    for 1 mol of (CH2OH)3
ThOD = (3.5*32*600)/92 //[mg/l]
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;

```

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


// Example 2.17
// Page 26
printf("Example 2.17, Page 26 \n \n");

// solution

S = .68 // sulphur content by mass
d = .85 // kg/l
s = (S*d*10^6)/100 // [mg/l] or [ppm]
printf("Sulphur content in LDO is "+string(s)+" ppm.
   ")
```

Scilab code Exa 2.18 molarity normality and molality

```
clear;
clc;

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

// Example 2.18

// Page 26
printf("Example 2.18, Page 26 \n \n");

// solution

m1 = 100 //[kg] solution (basis)
m2 = 20 //[kg] NaCl
```

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

Scilab code Exa 2.19 molarity of solution

```
clear;
clc;

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

// Example 2.19

// Page 27
printf("Example 2.19, Page 27 \n \n");

// solution

m1 = 100 //[kg] TEA solution (basis)
m2 = 50 //[kg] TEA
M1 = 149 // molar mass of TEA
d = 1.05 //[kg/l]
V = m1/d // volume of 100 kg sol.
n = (m2/M1)*100 // [mol] NaCl
M = n/V //[M]
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");

```

```
// solution

n = 100 // [mol] air (basis)
n1 = 21 //[mol] O2
n2 = 78 //[mol] N2
n3 = 1 //[mol] Ar
M1 = 31.9988 // O2
M2 = 28.0134 // N2
M3 = 39.948 // Ar
m1 = n1*M1
m2 = n2*M2
m3 = n3*M3
Ma = (m1+m2+m3)/n
printf("average molar mass of air is "+string(Ma)+"
    g.")
```

Scilab code Exa 2.23 Composition and specific gravity

```
clear;
clc;

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

// Example 2.23

// Page 39
printf("Example 2.23, Page 39 \n \n");

// solution

// (a)
printf("(a) \n \n")
n = 100 // [kmol] cracked gas (basis)
```

```
n1 = 45 // methane
n2 = 10 // ethane
n3 = 25 // ethylene
n4 = 7 // propane
n5 = 8 // propylene
n6 = 5 // n-butane
M1 = 16
M2 = 30
M3 = 28
M4 = 44
M5 = 42
M6 = 58
m1 = n1*M1
m2 = n2*M2
m3 = n3*M3
m4 = n4*M4
m5 = n5*M5
m6 = n6*M6
m = m1+m2+m3+m4+m5+m6
M = m/n
printf("Average molar mass of gas is "+string(M)+" g.")
//(b)
printf("(b) \n \n")
// composition
p1 = (m1/m)*100
p2 = m2*100/m
p3 = m3*100/m
p4 = m4*100/m
p5 = m5*100/m
p6 = m6*100/m
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")
```

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

Scilab code Exa 2.24 percentage error

```
clear;
clc;

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


// Example 2.24
// Page 40
printf("Example 2.24, Page 40 \n \n");

// solution

p = 100 //[bar]
T = 623.15 //[K]
R = .083145
V = R*T/p // [l/mol] molar volume
v = V/18.0153 //
printf("Specific volume = "+string(v)+" m^3/kg.")
```

Scilab code Exa 2.25 molar volume

```
clear;
clc;
```

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


// Example 2.25
// Page 40
printf("Example 2.25, Page 40 \n \n");

// solution

p = 4 //[bar]
T = 773.15 //[K]
R = .083145
V = R*T/p // [l/mol] molar volume
printf("Molar volume = "+string(V)+" l/mol.\n \n \n"
    )
 // using appendix III
 // calculating Tc and Pc of different gases
    according to their mass fractions
Tc1 = .352*32.20 // H2
Tc2 = .148*190.56 // methane
Tc3 = .128*282.34 //ethylene
Tc4 = .339*132.91 // CO
Tc5 = .015*304.10 // CO2
Tc6 = .018*126.09 // N2
Tc = Tc1+Tc2+Tc3+Tc4+Tc5+Tc6 // Tc of gas
 // similarly finding Pc
Pc1=.352*12.97
Pc2=.148*45.99
Pc3=.128*50.41
Pc4=.339*34.99
Pc5=.015*73.75
Pc6=.018*33.94
Pc=Pc1+Pc2+Pc3+Pc4+Pc5+Pc6 // Pc of gas
a = (27*R^2*Tc^2)/(64*Pc) // [bar/mol^2]
b = (R*Tc)/(8*Pc)  // l/mol
```

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

Scilab code Exa 2.26 ternary mix analysis

```
clear;
clc;

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


// Example 2.26
// Page 43
printf("Example 2.26, Page 43 \n \n");

// solution
m = 6.5065 //[g] mixture (basis)
Pv = 2.175 //[kPa] V.P. of water over KOH
Pa = 102.5-2.175 //[kPa] Partial P of n-butane and 1 butene
V = 415.1*10^-3 //[l]
R = 8.314472
T = 296.4 //[K]
n = (Pa*V)/R*T // moles of butene and butane
n1 = n*.431 // n-butane
m1 = n1*58 // [g]
n2 = n-n1 // 1 butene
m2 = n2*56 //[g]
m3 = m-m1 // [g] furfural
n3 = m3/96
```

```
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

```
clear;
clc;

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


// Example 2.27
// Page 44
printf("Example 2.27, Page 44 \n \n");

// solution

P = 5.7+1.01 //[bar] absolute total P
// using Roult's law
vp = 3.293*.7737 //[kPa] vap P of furfural
// using Dalton's law of partial P
n1 = vp/(P*100) // mol fraction of furfural
n2 = 1-n1 // mol fraction of 1-butene
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");
```

```

// solution

//Ti-Tf = mu*(Pi-Pf)
Pi = 20.7 //[bar]
Pf = 8.7 // [bar]
mu = 1.616 //[K/bar]
Ti = 355.15 //[K]
Tf = Ti-mu*(Pi-Pf)
printf("Outlet temperature is "+string(Tf)+" K")
```

Chapter 3

Material Balances without Chemical Reaction

Scilab code Exa 3.1 Lancashire boiler

```
clear;
clc;

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


// Example 3.1
// Page 60
printf("Example 3.1, Page 60 \n \n");

// solution

m = 1 //[kg] feed water
m1 = 1200 //[mg] dissolved solids in 1 kg feed water
m2 = 3500 //[mg] max dissolved solid content
x = (m*m1)/m2 // [kg] blown down water
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
```

```


// Example 3.3
// Page 61
printf("Example 3.3, Page 61 \n \n");

// solution

m = 100 //[kg] babul bark (basis)
m1 = 5.8 //[kg] moisture
m2 = 12.6 //[kg] Tannin
m3 = 8.3 //[kg] soluble non tannin organic material
m4 = m-m1-m2-m3 // [kg] Lignin
// lignin content remains unaffected during leaching
m5 = 100-.92-.65 // [kg lignin/kg dry residue]
x = (m4*100)/m5 // [kg]
T1 = x*.0092 //[kg] Tannin present in residue
T2 = m2 - T1 // [kg] Tannin recovered
T = (T2/m2)*100
printf("Percentage of Tannin recovered during
    leaching is "+string(T)+".")
```

Scilab code Exa 3.4 Extraction of dry neem leaves

```
clear;
clc;

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


// Example 3.4
// Page 62
printf("Example 3.4, Page 62 \n \n");
```

```

// solution

m = 1 //[kg] dry neem leaves (basis)
m1 = .01/100 //[kg] beta cartene content of leaves
Ex = (m1*100)/.41 //[kg] extract quantity
Tc1 = Ex*.155 //[kg] Alpha Tocopherol in the extract
Tc2 = .46/100 //[kg] Alpha Tocopherol in the neem
    leaves
R = (Tc1*100)/Tc2 // recovery of Alpha Tocopherol
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

```
clear;
clc;

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


// Example 3.5
// Page 62
printf("Example 3.5, Page 62 \n \n");

// solution

m= 100 //[kg] original mixture (basis)
A = 27.8 //[kg]
B = 72.2 //[kg]
// let x and y be uper and lower layer amounts
```

```
// total mixture = (x+y) kg
// balancing A and B
X = [.075 .203;.035 .673]
d = [27.8;72.2]
x = X\d
M = X(1,1)+X(2,1) // [kg] total mixture
Ms = M - m //[kg] mixed solvent
Mr = Ms/m // mixed solvent/original mixture
S1 = x(1,1)*.574+x(2,1)*.028 //[kg] water balance
S2 = x(1,1)*.316+x(2,1)*.096 //[kg] acetic acid
    balance
Qs = S1+S2
pS1 = (S1*100)/Qs
pS2 = 100-pS1
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

```
clear;
clc;

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


// Example 3.6
// Page 63
printf("Example 3.6, Page 63 \n \n");

```

```
// solution

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

Scilab code Exa 3.7 Required Oleum strength

```
clear;
clc;

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


// Example 3.7
// Page 64
printf("Example 3.7, Page 64 \n \n");

// solution

m = 100 //[kg] SO3 free mixed acid (basis)
m1 = 55 //[kg] HNO3
m2 = 45 //[kg] H2SO4
// SO3 + H2O --> H2SO4
m3 = (80/18)*3 //[kg] SO3 equivalent to 3 kg of
    water
Q = m2+m3 //[kg] oleum to be mixed
```

```
S = (m3/Q)*100 // strength of oleum
R = m1/Q
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

```
clear;
clc;

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


// Example 3.8
// Page 64
printf("Example 3.8, Page 64 \n \n");

// solution

m = 1000 //[kg] mixed acid (basis)
// doing overall mass balance, H2SO4 balance and
   HNO3 balance
A = [1 1 1;.444 0 .98;.113 .9 0]
d = [1000;600;320]
x = A\d
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

```
clear;
clc;

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


// Example 3.10
// Page 67
printf("Example 3.10, Page 67 \n \n");

// solution

// see examples 3.5 and 3.8
```

Scilab code Exa 3.11 Flowrate calculation

```
clear;
clc;

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


// Example 3.11
// Page 68
printf("Example 3.11, Page 68 \n \n");
```

```

// solution

// basis : 1000 kg/h of feed
// balancing H2SO4, HNO3 and H2O in all the three
   product streams
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]
v = [400;100;500;4;94;60;16;6;400]
s = M\v
A = s(1)+s(2)+s(3)
B = s(4)+s(5)+s(6)
C = s(7)+s(8)+s(9)
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

```
clear;
clc;

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


// Example 3.12
// Page 70
printf("Example 3.12, Page 70 \n \n");

// solution
m = 100  // kg
x = linspace(70,110,5);
```

```
y = linspace(100,115,4);
y1 = 27.8/.203 - .075*x/.203
y2 = 72.2/.673 - .035*x/.673
x = linspace(70,110,5);
plot(x,y1,style=4)
plot(x,y2,style=8)
// from graph its clear x = 93.4 kg and y = 102.4 kg
   .
x = 93.4;
y = 102.4;
M = x+y // [kg] total mixture
Ms = M - m //[kg] mixed solvent
Mr = Ms/m // mixed solvent/original mixture
S1 = x*.574+y*.028 //[kg] water balance
S2 = x*.316+y*.096 //[kg] acetic acid balance
Qs = S1+S2
pS1 = (S1*100)/Qs
pS2 = 100-pS1
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

```
clear;
clc;

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


```

```
// 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
```

```

f1 = 1.25 //[m^3/s] fresh ambient air as feed (basis)
f2 = 5.806 //[m^/s] air entering auditorium
v1 = 8.314*290/101.3 //[m^3/kmol] sp. vol. of moist air at 101.3 kPa and 290 K
na1 = f2*1000/v1 // [mol/s] molar flow rate of air entering auditorium
nw1 = 243.95*.0163/1.0163 // [mol/s]
na2 = 243.95 - nw1 //[mol/s] dry air flow
nw2 = 240.04*.0225 //[mol/s] moisture enterin air conditioning plant
// using table 3.8
m1 = (nw2-nw1) //[kg/h] moisture removed in a c plant
m2 = na2-.0181 //[mol/s] moisture in air leaving auditorium
m3 = (m2-nw1)*18 // [kg/h] moisture added in auditorium
Vm2 = 8.314*308/101.3 // [m^3/kmol]
na3 = (f1/25.28)*1000 //[mol/s]
n4 = 5.40-1.925 //[mol/s] moisture in recycle stream
mr = 240.04-47.525 //[mol/s] molar flow rate of wet recycle stream
R = mr/na3
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

```
clear;
```

```
clc;

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


// Example 3.16
// Page 78
printf("Example 3.16 Page 78 \n \n");

// solution

// screen 1
// feed = N kg
// Oversize particle = NE1 kg
// Undersize particle = N-NE1

//screen 2
//feed = NE1+X kg
// Oversize particle = (NE1+X)*E2 kg
// Undersize particle = (NE1+X)(1-E2) kg


//screen 3
// feed = (NE1+X)*E2 kg
// Oversize particle = (NE1+X)*E2*E3 kg
// Undersize particle = (NE1+X)*E2*(1-E3) kg
printf("Overall Efficiency = (E1 E2 E3)*100/[(1-E1)
    (1-E2)+E2 E3].")
```

Scilab code Exa 3.17 2 stage membrane CO separation

```
clear;
clc;
```

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


// Example 3.17
// Page 79
printf("Example 3.17, Page 79 \n \n");

// solution

printf("(a) \n \n")
F = 5000 //[kmol/h] feed (basis)
m1 = F*.47 //[kmol/h] CO in F
m2 = F-m1 //[kmol/h] H2 in F
m3 = m1*.932 // CO in product stream
n2 = m3/.98 //[kmol/h]
printf("Flow rate of product stream is "+string(n2)+
    " kmol/h. \n \n \n(b) \n \n")
n2 = n2-m3 //[kmol/h] H2 in CO stream
printf("    Product H2 stream : \n  H2 = "+string(m2
    -n2)+" kmol/h \n  CO = "+string(m1-m3)+" kmol/h \
    n \n \n(c) \n \n")
nH2 = 2697.39 //[kmol/h]
nCO = 3000-nH2 // [kmol/h]
n4 = m2+nH2
n5 = m1+nCO
n6 = n4+n5

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
```

```
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

```
clear;
clc;

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


// Example 3.20
// Page 86
printf("Example 3.20, Page 86 \n \n");

// solution

// concetration of the component after n times
    introduction of v volume of inert gas :
// Cn = Co/(1+1/n)^n
// we know limn-->infinity (1+1/n)^n = e
// therefore Cv = Co/e
```

Chapter 4

Material Balances Involving Chemical Reactions

Scilab code Exa 4.1 Manufacture of MCA

```
clear;
clc;

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


// Example 4.1
// Page 116
printf("Example 4.1, Page 116 \n \n");

// solution

// basis one day operation
// Cl2 is the limiting component
n1 = 4536/71 //[kmol] Cl2 charged
// 1mol MCA requires 1 mol Cl2, so
n2 = 5000/94.5 // [kmol] Cl2 used for MCA production
```

```
// 1 mol DCA requires 2 mol of Cl2
n3 = 263*2/129 //[kmol] Cl2 used for DCA production
n4 = n2+n3 // total Cl2 used
a = n4*100/n1 // conversion %age
b = n2*100/n4 // yield % of MCA
s = n2/n3
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

```
clear;
clc;

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


// Example 4.2
// Page 117
printf("Example 4.2, Page 117 \n \n");

// solution

m = 700 //[kg] ONT charged to reactor (basis)
m1 = 505*.99 // [kg] OT produced
m2 = (4*137*500)/(4*107)  //[kg] ONT required
m3 = m*.98 // [kg] ONT reacted
n1 = m1*100/m3 // yield of OT
m4 = (9*56*m)/(4*137) // [kg] theoretical iron
   reqiurement
m5 = 800*.9 //[kg] iron charged
E = (m5-m4)*100/m4 // excess iron
```

```
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

```
clear;
clc;

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


// Example 4.3
// Page 118
printf("Example 4.3, Page 118 \n \n");

// solution

printf("(a) \n \n")
m = 100 //kg chlorobenzene (basis)
m1 = 106.5*.655 //kg HNO3
m2 = 108*.936 // kg H2SO4
m3 = 106.5*.345 +108*.064 //kg water
M = m1+m2+m3
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")
// (b)
// total charge mass is constant
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");
```

```
// solution

n=100 //[kmol] outgoing gas from 2nd scrubber
n1=.852*n //[kmol] N2
n2=21*n1/79 //[kmol] O2
n3=n2-2.1 // [kmol] reacted O2
// O2 balance
// O2 consumed in rxn (ii),(iii),(v) - O2 produced
   by rxn (iv) = 20.55 kmol
// let a,b,c be ethanol reacted (ii),(iii),(iv) and
   d be H2 reacted in (v)

// CO balance
a=2.3/2 //kmol

//CO2 balance
b = .7/2

//CH4 balance
c=2.6/2

//O2 balance
d = 41.1-a-3*b+c

//H2 balance
e = 7.1 +c+d //kmol (total H2 produced)
f = e-(3*b + 3*a) //kmol (H2 produced in (i) =
   ethanol reacted in (i))
g = f+a+b+c // total ethanol reacted
h = 2*(n1+n2) // total ethanol entering
c1 = g*100/h
printf("(a) \n \n Conversion percent of ethanol = "+
   string(c1)+" \n \n \n")
y = f*100/g
printf("(b) \n \n Yield of acetaldehyde = "+string(y
   )+".")
```

Scilab code Exa 4.5 Lime Soda process

```
clear;
clc;

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


// Example 4.5
// Page 121
printf("Example 4.5, Page 121 \n \n");

// solution

v = 1 //[l] water (basis)
// 1 mol (100mg) CaCO3 gives 1 mol (56) Cao
// use table 3.3 and eg 3.9
x = 56*390.6/100 //[mg/l] lime produced
printf("Amount of lime required = "+string(x)+" mg/l
    .")
```

Scilab code Exa 4.6 Manufacture of Ammonia by Fertilizer plant

```
clear;
clc;

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

```

```

// Example 4.5
// Page 121
printf("Example 4.5, Page 121 \n \n");

// solution

m=100 //[kmol] (basis) dry mixed gas
// x = kmol of water gas
// y =kmol of producer gas
// overall material balance :
// x+y = 100    (i)

//r2 =.43x+.25y // H2 formed by shift rxn
//r2=.51x+.25y // H2 entering with water and
    producer gas
//r = r1+r2 // taoal H2
///n =.02x+.63y // N2 entering
//N2:H2=1:3
// ==> x-1.807y = 0(ii)
//solving (i) and (ii)
A = [1 1;1 -1.807]
d = [100;0]
x = A\d
s = .43*x(1)+.25*x(2) // steam req.
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

```
clear;
clc;

// 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
```

```
n2 = 3 //[kmol] O2
n3 = 80.5 //[kmol] N2
// S + O2 = SO2
// S + 3/2 O2 = SO3
n4 = (21/79)*80.5 //[kmol] O2 supplied
n5 = n4-n1-n2 // [kmol] Unaccounted O2
// O2 used in 2nd eq is m5
n6 = (2/3)*n5 //[kmol] SO3 produced
n7 = n1+n6 // sulphur burnt
m7 = n7*32 //[kg]
f1 = n6/n7 // fraction of SO3 burnt
// O2 req. for complete combustion of S = n7
n8 = n4-n7 //[kmol] excess O2
p1 = n8*100/n7 // %age of excess air
n9 = n4+n3 //[kmol/s] air supplied
F1 = n9*.3/n7 // air supply rate
v = 22.414*(303.15/273.15)*(101.325/100) //[m^3/kmol] sp. vol of air
V1 = F1*v //[m^3/s] flow rate of fresh air
n10 = n+n7 //[kmol] total gas from burner
n11 = n10*.3/m7 // [kmol/s] gas req. for .3 kg/s S
V2 = 220414*1073.15*n11/273.15 // flowrate of burner gases
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

```
clear;
clc;

```

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


// Example 4.9
// Page 125
printf("Example 4.9, Page 125 \n \n");

// solution

m = 100 //[kg] soya fatty acid (basis)
// use table 4.6
M1 = m/.3597 // M(avg) of soya fatty acid
//3 mol of fatty acid + 1 mol of glycerol = 1 mol triglyceride + 3 mol of water
M2 = M1*3+92.09-3*18.02 // Mavg of soyabean oil
q1 = M2*m/(M1*3) // soyabean oil per 100kg fatty acid
// based on reactions occuring
q2 = .0967+.1822*2+.0241*3 // kmol H2 req. per 100 kg soya fatty acid
q3 = .5101 // kmol H2 req. per 100 kg soyabean oil
q4 = 11.434 // Nm^3/100kg soyabean oil
// x = linoleic acid converted to oleic acid
// y = oleic acid converted to stearic acid
q5 = 282.46*6.7/278.43 //
//q6 = 282.46*x/280.15 = 1.00717x [kg] oleic acid by linoleic acid
//q7 = 284.48*y/282.46 = 1.00715y [kg] stearic acid by oleic acid
//q8 = 100.097 + .00717x + .00715y  total fatty acid
//stearic balance : -.00105x + 1.00611y = 10.8142 (i)
//linoleic balance : 1.0019x + .00019y = 48.4975 (ii)
// solving (i) and (ii) we get
x = 48.5 //kg
```

```
y = 10.8 //kg
M3 = 100.52/.3596 // Mavg of fatty acid
H2req1 = .5334-.2864 // per 100kg fatty acid
H2req = 52.95 //Nm^3/t
I2s = 129.5 //kg I2 per 100 kg soyabean oil // for soyabean oil
I2h = 69.2 //kg I2 per 100 kg of fat
printf("(a) \n \n theoretical H2 required = "+string(q4)+" Nm^3/100kg 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

```
clear;
clc;

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


// Example 4.10
// Page 128
printf("Example 4.10, Page 128 \n \n");

// solution

F1 = 4000 //kg/h methanol (basis)
F2 = F1/32 //kmol/h
F3 = F2/.084 //kmol/h gaseous mix flowrate
F4 = F2-F3 //kmol/h flow of wet air
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

```
clear;
clc;

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


// Example 4.11
// Page 132
printf("Example 4.11, Page 132 \n \n");

// solution

m = 100 //kg pyrites (basis)
//(a)
printf("(a) \n \n")
S1 = 42 //kg
i1 = 58 //kg  inerts
// 8 moll S = 3 mol O2 in Fe2O3
m1 = 3*32*42/8*32 //kg O2 converted to Fe2O3
m2 = i1+m1 // mass of SO3 free cinder
//2.3 kg S is in 100kg cinder
m3 = 100-(2.3*80/32)
m4 = (100/m3)*m2
m5 = m4*.023 //kg  S in cinder
p1 = 1.8*100/42
printf("percentage of cinder remained in cinder = "+
    string(p1)+". \n \n \n(b) \n \n")
//(b)
```

```
m6 = 100 //kmol SO3 free roaster gas (basis)
m7 = 7.12 //kmol O2 as SO2
m8 = 10.6 //O2
m9 = 100-m8-m7//N2
m10 = (21/79)*m9 // O2 entering roaster along N2
m11 = m7+m8+(3*7.12/8) // accounted O2
m12 = m10-m11 // unaccounted O2
m13 = (8/15)*m12 // SO3 formed
m14 = m13+m7 // S burnt
p2 = (m13/m14)*100
printf("percentage of S burnt to form SO3 = "+string(p2)+" \n \n \n(c) \n \n")
// (c)
// basis 100kg pyrite
m15 = 37.81/32 // SO2 formed
m16 = (m9+m10)*1.181/m7 // air supplied
// 4 kg pyrite is roasted
m17 = m16*4/100 //kmol/s  total air supplied
v1 = m17*24.957
printf("volumetric flow rate of air = "+string(v1)+" m^3/s \n \n \n(d) \n \n")
// (d)
m18 = (100.455*m17)/(m9+m10) // roaster gases
v2 = m18*66.386
printf("volumetric flow rate of roaster gases = "+string(v2)+" m^3/s \n\n\n(f) \n\n")
//(f)
m19 = 4.838*10^-2*.98 // SO3 absorbed in absorber
// SO3 + H2O = H2SO4
m20 = (m19*98*24*3600)/(.98*1000) //[t/d]
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
```

```
m9 = 6.125*m7/m5 // SO3
m10 = .667*x // Fe2O3
m11 = m6+m10+m8+m9+I2
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")
S1 = (64/120)*69 + (32/97.4)*17 //[kg]  S charged to burner
S2 = .035*79.63 // S in cinder
p = S2*100/S1
printf("percentage of S left in cinder = "+string(p)+"")
```

Scilab code Exa 4.13 Raising pH with NaOH

```
clear;
clc;

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


// Example 4.13
// Page 138
printf("Example 4.13, Page 138 \n \n");

// solution

m1 = 1200*1.2 //[kg] mass of reactants
pOH1 = 14-6 //pOH of reactants
pOH = 14-9 //pOH of final mass
```

```
// ROWs = 1/sigma(Wi/ROWsi)
//Ms = mass of .5% NaOH required
//ROWs = density of final solution

//ROWs = 1/{((m1*10^3*1)/(((m1*10^3+Ms)*1.2)+(Ms/((
    m1*10^3+Ms)*1.005))}      (i)
//balance of OH- ions
//1200*10^-8 +Ms*10^-1.15/(1.005*10^-5) =
    (1200*1.2*10^3+Ms)*10^-5/ROWs*10^-5        (ii)
//solving (i) and (ii)
Ms = 170.21 //g
ROWs = 1.2016 //[kg/l]
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

```
clear;
clc;

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


// Example 4.14
// Page 140
printf("Example 4.14, Page 140 \n \n");

// solution

// using equations of example 4.10

// soving 4.10 by linear model method
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 -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;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
```

```
//at anode : SO4-- - 2e = SO4
eqwtCu = 63.5/2
depositedCu = eqwtCu*m2
E = (1130*18000)/96485//faradays  Total energy passed to cell
libCu = (1130*18000*eqwtCu)/96485 //[g]  theoritical liberation of Cu
eff = (depositedCu/libCu)*100 // current efficiency
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

```
clear;
clc;

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


// Example 4.16
// Page 144
printf("Example 4.16, Page 144 \n \n");

// solution

// basis : 1day operation
// NaCl = Na+ + Cl-
//H2O = H+ + OH-
//Na+ + OH- = NaOH
//H+ + e = (1/2)H2
//Cl- - e = (1/2)Cl2
E = (15000*3600*24)/96485 // faraday/day   Total
```

```
energy passed thrrough cell
NaOH = (15000*3600*24*40)/(96485*1000) //[kg/day] theoretical NaOH
eff = (514.1/NaOH)*100 // current efficiency
Cl2 = (35.5/40)*514.1
H2 = (456.3*2)/(35.5*2)
//40 g NaOH = 58.5g NaCl
consNaCl = (58.5/40)*514.1 // NaCl consumed
Tliquor = 514.1/.11 //[kg/day] total cell liquor
remNaCl = 514.1*1.4
totalNaCl = consNaCl+remNaCl
Fbrine = totalNaCl/.266 //feed rate of brine
consH2O = (18/40)*514.1
lossH2O = Fbrine-Tliquor-consH2O
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

```
clear;
clc;

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


// Example 4.17
// Page 146
printf("Example 4.17, Page 146 \n \n");

// solution
```

```

//M = mix feed rate , F = fresh feed rate , R = recycle stream
// using fig 4.3
// N2 balance
// a = 24.75M/(.25M+7.5M)      (i)
// P = (4.15M + 17.75a)/M      (ii)
// .585M -1.775a +(4.15M+17.75a)/M = 100   (iii)
//solving (i,) (ii), (iii)
M = 438.589 //[kmol/s]
a = (24.75*M)/((.25*M)+7.5) //kmol/s
P = (4.15*438.589+17.75*92.662)/M //kmol/s
R = M-100 // kmol/s
r = R/100 // recycle ratio
NH3 = (.585*M-2.275*a)*17.0305 //kg/s
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

```
clear;
clc;

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


// Example 4.18
// Page 149
printf("Example 4.18, Page 149 \n \n");

// solution
```

```
// given

// (.1*M*R1)/(.415M+1.775a) + (.1125a*P)/(.415M +
    1.775a) + 1 = .1M
// R1*(.315M-1.225a)/(.415M + 1.775a) = .9M-4a
// M = 100 + R1 + (2.25a*p)/(.415M + 1.775a)
// .1M*P/(.415M + 1.775a) - (.1125a*P)/(.415M1.775a)

//solving them
M = 457.011 // kmol/s
R1 = 350.771 // kmol/s
P = 10.368 // kmol/s
a = 96.608 // kmol/s
R2 = 2.25*96.608*10.369/(.415*457.011 +
    1.775*96.608) // kmol/s
F = M -R1 - R2
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

```
clear;
clc;

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


// Example 4.19
```

```
// Page 153
printf("Example 4.19, Page 153 \n \n");

// solution

m1 = (50/35.5)*312 //[mg/l]  Cl2 expressed as
    equivalent CaCO3
m2 = (50/48)*43.2 //[mg/l]  Sulphates as equivalent
    CaCO3
A = m1+m2 //[mg/l as CaCO3]    EMA in raw water
M1 = 550 // alkalinity of raw water
M2 = 50 // alkalinity of blend water
//let 100 l of raw water enters both ion exchangers
// balancing neutrilasion
x = 100*(M1-M2)/(A+M1) // raw water inlet to H2 ion
    echanger
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

```
clear;
clc;

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


// Example 4.20
// Page 155
printf("Example 4.20, Page 155 \n \n");

// solution

```

```
m1 = 1488.1 //kmol/h gas mix to reactor1 (basis)
m2 = m1*.0625 // CH3OH
m3 = m1-m2 // ambient air flow
m4 = m3/1.01772 // dry air flow rate
m5 = m3-m4 // moisture
m6 = m2*.99 // CH3OH conversion in R1
m7 = m2-m6 // unreacted CH3OH
//rxn i
m8 = m7*.9 // CH3OH reacted = HCHO produced = H2O produced
m9 = m8/2 // O2 consumed
m10 = m6-m8 // CH3OH reacted in rxn ii to v
//rxn ii
m11 = m10*.71 // CH3OH reacted = CO2 produced
m12 = m11*1.5 // O2 consumed
m13 = 2*m11 // H2O produced
//rxn iii
m14 = m10*.08 // CH3OH reacted = CO produced
m15 = 2*m14 // H2 produced
//rxn iv
m16 = m10*.05 // Ch3OH reacted = CH4 produced
m17 = m16/2 // O2 produced
//rxn v
m18 = m10-m16-m14-m11 // CH3OH reacted
m19 = m18/2 // (CH3)2O = H2O produced

m20 = 287.87-m9-m12+m17 // O2 in R1 exit stream
m21 = m5+m8+m13+m19 // H2O in R1
m = m7+m8+m11+m14+m15+m16+m19+m20+1082.93+m21
// R2
// x kmol/h CH3OH is added b/w reactors
// (m7+x)/(m+x) = .084 solving it
x = 140.548 //[kmol/h]
m22 = x+m7 // CH3OH entering R2
m23 = m22*.99 //CH3OH reacted
m24 = m22-m23 // CH3OH unreacted
//rxn i
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
```

```
// Material Balances involving Chemical Reaction


// Example 4.21
// Page 159
printf("Example 4.21, Page 159 \n \n");

// solution

// basis 1 tonne of pig iron
coke = 1000 //kg
flux = 400 //kg
Fe1 = 1000*.95 // Fe in pig iron
Fe2 = (112/160)*.8 // Fe available per kg of ore
ore = Fe1/Fe2 // kg
Si = .014*1000 //kg  //Si in pig iron
si1 = (60/28)*14 // silica present in pig iron
si2 = ore*.12 // silica in ore
si3 = .1*coke // silica in coke
si4 = si2+si3-si1 // silica in slag
alumina = ore*.08 // Al2O3 in ore = Al2O3 in slag
CaO = flux*(56/100)
slag = si4+alumina+CaO
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")
C = .9*coke+(12/100)*flux-36 // total C available
// CO:CO2 = 2:1
C1 = C/3 // C converted to CO2
C2 = 2*C/3 // C converted to CO
O21 = C1*(32/12)+C2*(16/12) // O2 required for CO
   and CO2 formation
O22 = (32/28)*Si // O2 from SiO2
O23 = ore*(.8*48/160) // O2 from Fe2O3
O24 = flux*(32/100) // O2 from CaCO3
```

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

Chapter 5

Energy Balances

Scilab code Exa 5.1 Pumping of water

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.1
// Page 186
printf("Example 5.1, Page 186 \n \n");

// solution

// basis pumping of 1 l/s of water
Hadd = 52  // kW
Hlost = 21  // kW
fi = Hadd - Hlost  // kW
p1 = 101325  // Pa
p2 = p1
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
```

```

// (a)
Pc = 3732  // kPa
Tc = 630.3  // K
Tb = 417.6  //K
TBr = Tb/Tc
lambdav = 8.314472*417.6*(1.092*(log(3732)-5.6182)
   /(.930-.6625))
// (b)
T1 = 298.15  //K
lambdav1 = 36240*[(630.3-298.15)/(630.3-417.6)]^.38
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

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.11
// Page 225
printf("Example 5.11, Page 225 \n \n");

// solution

// (a)
Pc = 61.37 // bar
```

```
Tc = 514  //K
Tb = 351.4
P = 1 // atm
TBr = Tb/Tc
// Riedel eq
lambdav1 = 8.314472*Tb*1.092*(log(6137)-5.6182)
   /(.930-TBr)
// NIST eq
lambdav2 = 50430*exp(-(-.4475*TBr))*(1-TBr)^.4989
// (b)
T1 = 298.15
TBr1 = T1/Tc
// Watson eq
lambdav21 = 38563*[(514-298.15)/(514-351.4)]^.38
// NIST eq
lambdav22 = 50430*exp(-(-.4475*TBr1))*(1-TBr1)^.4969
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

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.12
```

```
// Page 227
printf("Example 5.12, Page 227 \n \n");

// solution

// using Appendix IV.2
Ps1 = 75
Ps2 = 80
T1 = 563.65
T2 = 568.12
T = 565.15
Ps = 75*exp((T2*(T-T1)*log(80/75)/(T*(T2-T1))))
printf(" Saturation Pressure of steam at 565.15K is
    "+string(Ps)+" bar.")
```

Scilab code Exa 5.13 Bubble and Dew pt calculations

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.13
// Page 236
printf("Example 5.13, Page 236 \n \n");

// solution

// basis  1 kmol equimolar mix
npent = .5 // kmol
nhex = .5 // kmol
P = 101.325  // kPa
```

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

Scilab code Exa 5.14 Hot air drying machine

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.14
// Page 237
printf("Example 5.14, Page 237 \n \n");

// solution

// basis 1000 kg/h of condensate at the saturation
    temperature corresponding to 8 bar a
// using Appendix IV.2
H = 720.94 // kJ/kg
Hm = 419.06 // kJ/kg
x = poly(0,'x')
condensate = 1000-x
Hcondensate1 = 1000*H
```

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

Scilab code Exa 5.15 Flow of saturated vapors of R134

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.15
// Page 238
printf("Example 5.15, Page 238 \n \n");

// solution

qv1 = 50  // l/s
qm = qv1*1.08  // kg/s
fi = qm*3.08*(263.15-258.15) // kW
lv = 384.19-168.7  // kJ/kg
qm2 = fi/lv
H = 256.35  // kJ/kg
x = poly(0, 'x')
p = H*(qm2+x) - 168.7*qm2-x*384.19
a = qm2+roots(p)
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)
```

```
R = Cl2recy/.116 // kg/kg fresh feed
// T4/T1 = (p2/p1)^[(gamma-1)/gamma]
gm = 1.355
p22 = 326.3
p21 = 101
T4 = Ts1*(p2/p1)^[(gm-1)/gm]
T5 = 313.15
fi3 = 1.88*10^-3*(343.1+91.6-26.2+2.5)   // kW
Fwater1 = fi3/(8*4.1868)   // kg/s
// similarly
T6 = 379.9
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
Fwater2 = fi4/(8*4.1868)   // kg/s
Wreq = Fwater1+Fwater2
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
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

```
clear;
clc;

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

```
// Example 5.17
// Page 242
printf("Example 5.17, Page 242 \n \n");

// solution

// basis 100 kg of tin
T1 = 303.15
T2 = 505.15
n = 100/118.7  // kmol
// Q1 = n*[intgr from T1 to T2 (Cms dT)]
Q1 = 4973.3 // kJ
lf = 7201
Q2 = n*lf  // kJ
Q = Q1+Q2
lv = 278 // kJ/kg
vp = Q/lv // kg
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

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.18
// Page 243
```

```
printf("Example 5.18, Page 243 \n \n");

// solution

Ts1 = (438.2+436)/2
Ta = 300
fi1 = .045*(Ts1-Ta)*3600
theta1 = 307293/fi1  //h
Ts2 = (436+434)/2
fi2 = .045*(Ts2-Ta)*3600
theta2 = 302415/fi2
Ts3 = (434+432.1)/2
fi3 = .045*(Ts3-Ta)*3600
theta3 = 313859/fi3
theta = theta1+theta2+theta3
printf(" total time required = "+string(theta)+" hrs.")
```

Scilab code Exa 5.19 Manufacture of dry ice

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.19
// Page 245
printf("Example 5.19, Page 245 \n \n");

// solution

H1 = 482.9  // kJ/kg
```

```
H2 = 273.4
fi1 = 100*(H1-H2) // kJ/h
T1 = 313.15
T2 = 403.15
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
// at 20 MPa
h1 = 211.1
Ts = 277.6
H11 = 427.8
x = poly(0, 'x')
p = x*h1+(100-x)*H11-100*H2
a = roots(p)
fi2 = (100-a)*(H11-h1)  // kJ/h
h2 = -148.39
H3 = 422.61
y = poly(0, 'y')
p1 = 100*176.18-(100-y)*H3+h2*y
b = roots(p1)
fi3 = 100*(h1-176.8)
H = fi3+24021
H4 = H/(100-43.16)
// from ref 23
T = 262.15
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

```
clear;
clc;

```

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


// Example 5.20
// Page 247
printf("Example 5.20, Page 247 \n \n");

// solution

// basis 200 kg/h of Sulphur firing
F = 200/32  // kmol/h
O2req = 6.25*1.1
airin = O2req/.21
N2in = airin-O2req
T1 = 1144.15
T2 = 463.15
fi = 788852.2 // kJ/h
H = 15*4.1868+1945.2
qm = fi*.9/2008  // kg/h
printf(" Amount of steam produced = "+string(qm)+"
   kg/h.")
```

Scilab code Exa 5.21 Equimoar pentane and hexane mix

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.21
```

```
// Page 248
printf("Example 5.21, Page 248 \n \n");

// solution

// enthalpy at Tbb
Tbb = 321.6
T1 = 298.15
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
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
Hsol = (H1+H2)/2  // kJ/kmol
// enthalpy at Tdp
lv1 = 25790*((469.7-329.9)/(469.7-309.2))^.38
lv2 = 28850*((507.6-329.9)/(507.6-341.9))^.38
Tdp = 329.9
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
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
Hmixig = (H21ig+H22ig)/2
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

```
clear;
clc;

// 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
```

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

Scilab code Exa 5.24 Refrigiration calculations

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.24
// Page 256
printf("Example 5.24, Page 256 \n \n");

// solution

// fi1 = integr (from 304.15 to 313.15) {11831.6+24997.4*10T^-3-5979.8*10^-6T^2-31.7*10^-9T3}dt
fi1 = 170787.7 // kJ/h
fi2 = 535.4*12086 - [344.36*8743.2+168.97*18036+22.07*15892]  // kJ/h
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;
```

```

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.30
// Page 267
printf("Example 5.30, Page 267 \n \n");

// solution

// basis 1 kmol of styrene
dH = 241749-189398 // kJ/mol
Cmpn = dH/(600-298.15) // kJ/kmol K
printf(" Mean heat capacity between 600K and 298.15
   K is "+string(Cmpn)+" kJ/kmol K.")
```

Scilab code Exa 5.31 Heat of reaction

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.31
// Page 269
printf("Example 5.31, Page 269 \n \n");

// solution

// basis 1 mol of SiO2 reacted
```

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

Scilab code Exa 5.32 Std heat of reaction

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.32
// Page 269
printf("Example 5.32, Page 269 \n \n");

// solution

// basis 100 kg of 2% ammonia solution
NH3 = 2 // kg
H2O = 98 // kg
Hr = -361.2-(-45.94-285.83)  // kJ/mol NH3 dissolved
Hd = -(Hr*2*1000/17.0305)  // kJ/100 kg sol.
printf(" heat of reaction = "+string(Hd)+" kJ/100 kg
     solution.")
```

Scilab code Exa 5.33 Burning of SO2

```
clear;
clc;
```

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


// Example 5.33
// Page 272
printf("Example 5.33, Page 272 \n \n");

// solution

// basis 1 kmol of SO2 reacted
a = 22.036-24.771-.5*(26.026)
b = (121.624-62.948-.5*11.755)
c = (-91.876+44.258-.5*(-2.343))
d = (24.369-11.122-.5*(-.562))
Hr = -395720+296810  // kJ/kmol
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
T = 778.15
Hrt = -Hro-15.748*T+26.4*10^-3*T^2-15.48*10^-6*T
    ^3+3.382*10^-9*T^4
printf(" Heat of reaction at 775K is "+string(Hrt)+"
     kJ/kmol.")
```

Scilab code Exa 5.34 Esterification of acetic acid

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances

```

```
// Example 5.34
// Page 272
printf("Example 5.34, Page 272 \n \n");

// solution

Hr = -480-285.83+277.2+484.2  // kJ/mol
Hrt1 = Hr*1000 + [146.89+75.76-119.55-129.70]*75  //
    kJ/kmol
a = 4.2905+50.845-100.92-155.48
b = 934.378+213.08+111.8386+326.5951
c = -2640-631.398-498.54-744.199
d = 3342.58+648.746
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
T = 373.15
Hrt = Hro+a*T+792.949*10^-3*T^2-1504.712*10^-6*T
   ^3+997.832*10^-9*T^4
printf(" Heat of reaction at 373 K is "+string(Hrt)+
   " kJ/kmol reactant.")
```

Scilab code Exa 5.35 Heat transfer in intercoolers

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.35
// Page 273
printf("Example 5.35, Page 273 \n \n");
```

```
// solution

T2 = 800
T1 = 298.15
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
T3 = 875
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
Hr = -98910  // kJ/kmol SO2 reacted     by eg 5.33
fi3 = (8.8511-.351)*Hr/3600  // kW
dH = fi2/3600+fi3-fi1/3600
printf(" Net enthalpy change = "+string(dH)+" kW.")
```

Scilab code Exa 5.36 Enthalpy balance in the reactor

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.36
// Page 275
printf("Example 5.36, Page 275 \n \n");

// solution

// basis 100 kmol outgoing gas mixture from scrubber
moistin = 3127.7*.015/18  // kmol
```

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

Scilab code Exa 5.37 Calculation of circulation rate

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.37
// Page 276
printf("Example 5.37, Page 276 \n \n");

// solution

fi3 = 15505407   // kJ/h
lv = 296.2   // from table 5.6
Ht = 17131551   // kJ/h
r = Ht/lv   // kg/h
printf(" Downtherm circulation rate = "+string(r)+"
    kg/h.")
```

Scilab code Exa 5.38 Loop reactor for EDC manufacture

```
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
```

```
// Example 5.39
// Page 284
printf("Example 5.39, Page 284 \n \n");

// solution

To = 298.15
T1 = 483.15
// 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
fi1 = 2455874.6  // kJ/h
dHr = 2*(-45.94)  // kJ/mol N2 reacted
fi2 = 91.88*1000*23.168
fi3 = fi1+fi2
// 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
// solving it
T2 = 657.41  // K
printf("Temperature of the gas mixture leaving the
    reactor = "+string(T2)+" K.")
```

Scilab code Exa 5.40 Burning of HCl

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.40
// Page 292
```

```
printf("Example 5.40, Page 292 \n \n");

// solution

// basis 4 kmol of HCl gas
O2req = 1  // kmol
O2spply = 1.35*1
N2 = 1.35*79/21
air = O2spply+N2
HClbrnt = .8*4
HCl = 4-HClbrnt
O2 = O2spply-.8
Cl2 = .8*2
H2O = .8*2
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 ")
H2 = 114.4*1000*.8
// 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
// solving it
T = 599.5  // K
printf(" Adiabatic reaction temperature of product
    gas stream = "+string(T)+" K.")
```

Scilab code Exa 5.41 Dehydrogenation of EB

```
clear;
clc;
```

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


// Example 5.41
// Page 294
printf("Example 5.41, Page 294 \n \n");

// solution

// 1 kmol of EB vapors entering the reactor at
   811.15 K
// (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
// we get
T1 = 929.72  // K
To = 298.15
H1 = 493405  // kJ
EBr = .35
Styrenep = EBr*.9
Benzeneb = EBr*.03
Ethyleneb = Benzeneb
Cb = EBr*.01
Toulened = EBr*.06
Hr1 = 147.36-29.92  // kJ/mol EB
Hr2 = 82.93+52.5-29.92
Hr3 = -29.92
Hr4 = 50.17-74.52-147.36  // kJ/mol styrene
dHr = 1000*(Hr1*(Styrenep+Toulened)+Hr2*Benzeneb+Hr3
   *Cb+Hr4*Toulened)
H2 = H1-dHr
// H2 = (from To t0 T2)intgr{Comp2dT
// we get
T2 = 798.79  // K
```

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

Scilab code Exa 5.42 Heat of crystallization

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.42
// Page 297
printf("Example 5.42, Page 297 \n \n");

// solution

Hsol = 62.86  // kJ/mol solute
Mcrystal = 286.1414
Hcry = Hsol*1000/Mcrystal  // kJ/kg solute
printf(" Heat of crystallization of 1 kg crystal is
    "+string(Hcry)+" kJ.")
```

Scilab code Exa 5.43 Heat of crystallization

```
clear;
clc;

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

```


// Example 5.43
// Page 297
printf("Example 5.43, Page 297 \n \n");

// solution

Hf = -285.82  // kJ/mol    of H2O
Hcryst = -4327.26-(-1387.08+10*Hf)
printf(" Heat of crystallization = "+string(Hcryst)+
   " kJ/mol.")
```

Scilab code Exa 5.44 Heat of sol of Boric acid

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.44
// Page 297
printf("Example 5.44, Page 297 \n \n");

// solution

Hfs = -1094.33
Hfao = -1072.32
Hsol = Hfao-Hfs
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;

```

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


// Example 5.46
// Page 300
printf("Example 5.46, Page 300 \n \n");

// solution

// using chart 5.16 we get
T = 329.5  // K
printf(" T = "+string(T)+" K.")
```

Scilab code Exa 5.47 Using std heat of formations

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.47
// Page 300
printf("Example 5.47, Page 300 \n \n");

// solution
// basis 100(m1) kg 46% sol
NaOH = 46  // kg
H2O = 54  // kg
m2 = NaOH/.25
NaOHo = 25  // kg
```

```
H2Oo = 75   // kg
Hf1 = -453.138   // kJ/mol
Hf2 = -467.678   // kJ/mol
Hs = Hf2-Hf1
Hg = -Hs*1000*1.501
// using Appendix IV.1
Hw1 = 146.65
Hw2 = 104.9
Hadd = 84*(Hw1-Hw2)
H = Hg+Hadd
C1 = 3.55
T2 = 298.15+H/(184*C1)   // K
printf(" Final sol T = "+string(T2)+" K.")
```

Scilab code Exa 5.48 Heat effect of the solution

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.48
// Page 301
printf("Example 5.48, Page 301 \n \n");

// solution
// basis 100 kg of sol with 32% N
MNH4NO3 = 80.0434
MNH2CONO2 = 60.0553
MN2 = 28.0134
na = 32/(60.9516)
Ureadis = 1.1758*na*MNH2CONO2   // kg
```

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

Scilab code Exa 5.49 Integral heats of solution

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.49
// Page 302
printf("Example 5.49, Page 302 \n \n");

// solution
Hmix = 896
M1 = 88  // molar mass of n-amyl alcohol
M2 = 78  // molar mass of benzene
B = .473*M2
A = .527*M1
Ha = Hmix/A
Hb = Hmix/B
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
```

```
// Energy Balances


// Example 5.54
// Page 311
printf("Example 5.54, Page 311 \n \n");

// solution

// from ref 24
H = 1600.83
To = 273.15
h = 200
Hf1 = -79.3  // table 5.59
Hf2 = -46.11
Hsol = Hf1-Hf2
Hg = Hsol*1000*140/17.0305
Raq = 140/.15  // kg/h
dT = Hg/(4.145*Raq)
T = -dT+303
printf(" Temp of resultant sol = "+string(T)+" K.")
```

Scilab code Exa 5.55 Using table 5 60

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.55
// Page 311
printf("Example 5.55, Page 311 \n \n");
```

```
Hf1 = -80.14
Hf2 = -46.11
Hsol = Hf1-Hf2
Hg = Hsol*1000*2/17.0305
printf(" Heat generated for making 2 percent
    solution = "+string(Hg)+" kJ/100 kg sol.")
```

Scilab code Exa 5.56 Heat removed in cooler

```
clear;
clc;

// Stoichiometry
// Chapter 5
// Energy Balances


// Example 5.56
// Page 312
printf("Example 5.56, Page 312 \n \n");

// solution

fi3 = 15505407
fi4 = 11395056
fi5 = fi3-fi4  // kJ/h
fi6 = 111.375*62.75*1000
fi7 = 1063379
fi8 = 5532.15*4.1868*(303.15-298.15)
fi9 = 9030.4*3.45*(323.15-298.15)
fi = fi5+fi6+fi8-fi7-fi9
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

```
clear;
clc;

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


// Example 6.1
// Page 346
printf("Example 6.1, Page 346 \n \n");

// solution

// basis = 100kmol of feed
Benzene = 100*.72// kmol
Toulene = 100-Benzene //kmol
// use fig 6.1
// 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;
```

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


// Example 6.2
// Page 349
printf("Example 6.2, Page 349 \n \n");

// solution

// basis = 2000kg/h liquid feed rate
F = 2000/28.84 //kmol/h
//D = distillate , W = residue flow rate
//N2 balance
// F*.79 = .999D + .422W    (i)
// 54.840 = D + .4224W      (ii)
// solving it
W = 25.118 //kmol/h
D = 44.230 //kmol/h
//using fig 6.4 and 6.5
// trial method is used for flash calculations
// Trial I
x = .75
// from fig 6.4
y = .8833
// from fig 6.5
Hl = 1083.65
Hv = 6071.7
Hf = .3*Hv+Hv*.7
// calculating we get Emix is not close to 2592.2kJ/
    kmol
//Trial II
x = .71
y = .859
Hl = 1085.6
Hv = 6118.6
```

```
Hf = .3*Hv+.7*Hl //kJ/kmol
// which is aproox equal to 2595.2kJ/kmol, so
   flashing will occur
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

```
clear;
clc;

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


// Example 6.3
// Page 353
printf("Example 6.3, Page 353 \n \n");

// solution
// material balance

// V2 vapour mix is a ternary azeotrope in which all
    cyclohexane of D1 is recycled
// V2 stream
// Cyclohexane balance
// D1 = (.488/.024)*V2
// IPA in V2 = .206V2
// water in V2 = (1-.488-.206)*V2
// W2 stream
// IPA in W2 = (.23D1-.206V2)
// water in W2 = (1-.024-.23)*D1-.306V2
// W2 stream = 4.471V2 + 14.862V2
```

```
// D3 is an azeotrope containing 67.5 mol% IPA
// water in W3 stream = (1-.675)F
// basis = 100 kmol/h fresh feed
// W1+W3 = 100                    (i)
// .998W1 + .001W3 = 67.5       (ii)
// solving it
W1 = 67.603 //kmol/h
W3 = 32.397 //kmol/h
IPA1 = W3*.001 // IPA in W3
//IPA2 = 4.471*V2 - .032   IPA in D3
//C-1 = F+D3 = F1
// water in D3 = 6.624V2 - .047-4.471V2+.032
// water in W3 = 14.862V2-2.153V2+.015
// solving them
V2 = 2.624 //kmol/h
D3 = 2.153*V2-.015
D1 = 20.333*V2
F1 = 6.624*V2+99.953
R = 1.75*D1 // = V1+V2-D1
V1 = 144.1
r = D3/100 // recycle ratio
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

```
clear;
clc;

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

```

```

// Example 6.4
// Page 355
printf("Example 6.4, Page 355 \n \n");

// solution

// basis 0.625 l/s of MEA solution
c = 3.2 //M  conc of MEA
M = 61 // molar mass of MEA
C = M*c //g/l  conc of MEA in sol
MEAin = c*.625*3600/1000 // kmol/h
CO2diss = .166*7.2 //kmol/h    CO2 dissolved in lean
    MEA
v = 26.107 //m^3  sp. vol of gas at 318K and 101.3
   kPa  (table 7.8)
qv = 1000/v //kmol/h
CO2in = qv*.104 // moles of CO2 in inlet gas
CO2freegas = qv - CO2in
//outgoing has 4.5% CO2
GASout = CO2freegas/(1-.0455) //kmol/h
CO2abs = qv-GASout
CO2 = CO2diss + CO2abs
CO2conc = CO2/MEAin //kmol/kmol MEA
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

```
clear;
clc;

// Stoichiometry
// 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
```

```
NaNO2t = NaNO2pro2 + NaNO2pro1
NaNO3t = NaNO3pro2+NaNO3pro3
H2Ot = H2Opro1+H2Opro2+H2Opro3
NaOHt = NaOHreac1+NaOHreac2+NaOHreac3
liq = 37500 //kg/h
NaOHin = liq*.236
NaOHout = NaOHin-NaOHt
moist = n2*.045*18
water = liq-NaOHin-H2Ot-moist //kg/h
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)")
//(b)
//heat effect of scrubbing
//using tables 6.6 and 6.7
//fi1 = integ{59865.7+4545.8+10^-3 *T + 15266.3*10^-6*T^2-705.11*10^-9*T^3}
fi1 = -155941.3/3600 //kW
//similarly
fi2 = 75.778 //kW
dH1 = (-346.303-450.1-285.83-(2*(-468.257)+2*33.18))/2 //kJ/mol NO2
dH2 = -346.303-450.1-285.83-(2*(-468.257)+9.16) //kJ/mol N2O4
dH3 = (2*(-450.1)-285.83+90.25-(2*(-468.257)+3*33.18))/3 // kJ/mol NO2
dHdil = -469.837-(-468.257) //kJ/mol NaOH
fi3 = (dH1*1000*18.48+dH2*1000*27.32+dH3*1000*14.85+dHdil*1000*138.23)/3600 //kW
fi4 = -fi1+fi2+fi3
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
```

```
// F+S = M = E+R
M = E+R
S = D+E1
AAloss = W*.4*100/(100*.3)
AArec = 100-AAloss
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

```
clear;
clc;

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


// Example 6.9
// Page 367
printf("Example 6.9, Page 367 \n \n");

// solution

```

```
// 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 MgSO4.7H2O = "+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)
```

```
printf("(a) \n \n ")
// using fig 6.12
// peforming material balance at 290K
a1 = 5.76
b1 = 4.91
DCBs = b1*100/(a1+b1) // % of solid separated p-DCB
DCBr1 = DCBs*100/70 // recovery of p-DCB
printf("Percentage recovery of p-DCB = "+string(
    DCBr1)+". \n \n \n (b) \n \n ")

//(b)
//at 255K
a2 = 5.76
b2 = 10.22
DCBs = b2*100/(a2+b2)
DCBr2 = (DCBs*100)/70
Ar = DCBr2-DCBr1
printf("Additional recovery of p-DCB = "+string(Ar)+
    ".")
```

Scilab code Exa 6.12 Extractive Crystallization of o and p nitrochlorobenzenes

```
clear;
clc;

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


// Example 6.12
// Page 371
printf("Example 6.12, Page 371 \n \n");

```

```
// 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;

```

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


// Example 6.14
// Page 384
printf("Example 6.14, Page 383 \n \n");

// solution

//Pa = v.p. at DP
Pw = 2.0624 //kPa
P = 100 //kPa
Hm = Pw/(P-Pw)   // kmol water vapour / kmol dry air
H = .622*Hm  // kg moisture/kg dry air
// at saturation, DB = WB = DP
Ps = 4.004 //kPa
RH = Pw*100/Ps
Hs = (Ps/(P-Ps))*.622
s = H*100/Hs
Ch = 1.006+1.84*H //kJ/kg dry air K
Vh = (.00073+.03448)*22.414*1.1062*1.0133  //m^3/kg
    dry air
// using fig 6.15
WB = 294.55 //K
ias = 62.3 // kJ/kg dry air
d = -.28 // kJ/kg dry air
ia = ias + d
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
```

```
lamda= 2163.2 //kJ/kg          by appendix IV.2
SC = fi/lamda  //kg/h     steam consumption at the heater
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

```
clear;
clc;

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


// Example 6.16
// Page 387
printf("Example 6.16, Page 387 \n \n");

// solution

// M = E+B+W
Tav = (45+32)/2 +273.15 //K   avg cooling water T
// using steam tables (Appendix A IV.1)
lamda = 2410.5  //kJ/kg
E = 530/lamda  //kg/s
Cl = 4.1868
Ti = 45+273.15 //K
To = 32+273.15 //K
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
```

```
// Page 389
printf("Example 6.17, Page 389 \n \n");

// solution

// basis 1 kg dry air fed to tower
// from fig 6.16 we get
// at WB=330 K and DB=393 K
H1 = .0972 // kg/kg dry air
DP = 325.15 //K
// at 313 K
H2 = .0492 // kg/kg dry air
H = H1-H2 // moisture condensed in tower
Ch1 = 1.006 + 1.84*H1 // kJ/kg dry air
Ch2 = 1.006 + 1.84*H2
ia1 = 1.006*(325-273) + H1*2596 + 1.185*(393-325) // enthalpy of entering air
ia2 = 1.006*(313-273) + H2*2574.4 // enthalpy of outgoing air
i = ia1-ia2
qm = 2000/(1+H1)
fi1 = qm*i // heat loss rate
fi2 = 1.167*3600*4.1868*(323-305) // heat gained by water
r = fi2*100/fi1
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

```
clear;
clc;

```

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


// Example 6.18
// Page 390
printf("Example 6.18, Page 390 \n \n");

// solution

// basis 800 kmol of inlet CS2-H2 mix
Pi = 106.7 //kPa  Total Pressure
Pcs2i = 16.93 // kPa
n = 800 // kmol
ncs2i = Pcs2i*n/Pi // kmol
nh2i = n-ncs2i
Po = 101.325 // kPa
Pcs2o = 6.19 // kPa
nh2o = 673.1 // kmol
ncs2o = Pcs2o*nh2o/(Po-Pcs2o)
ncs2a = ncs2i-ncs2o
mcs2a = ncs2a*76.1407 //kg
r = 600 // kg/h  design adsorption rate
Mi = n*r/mcs2a // kmol/h
Vi = Mi*22.843 // m^3/h
mcs2ac = .32-.04 // kg  CS2 absorbed per kg BD activated carbon
qm = r*1.04/mcs2ac  // kg/h
C = ncs2o/nh2o // kmol CS2/kmol H2   = Pcs2/(P-Pcs2)
Pcs2 = 24.763 // kPa
T = 281.5  //K  by eq 5.24
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

```
clear;
clc;

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


// Example 6.20
// Page 393
printf("Example 6.20, Page 393 \n \n");

// solution

// basis 100 kmol of feed gas
// using table 5.1
Sniai = 20.6*29.5909+62*28.6105+4.1*20.7723+11.1*19.2494+2.2*25.6503
Snibi = [20.6*(-5.141)+62*1.0194+11.1*52.1135+2.2*33.4806]/1000
Snici = [20.6*13.1829+62*(-.1476)+11.1*11.973+2.2*.3518]/10^6
Snidi = [20.6*(-4.968)+62*.769+11.1*(-11.3173)+2.2*(-3.0832)]/10^9
Hgas = Sniai*(283-263) + Snibi*(283^2-263^2)/2 + Snici*(283^3-263^3)/3 + Snidi*(283^4-263^4)/4  // kJ
Hnh3 = 1533.8  //kJ
SniCmpi = (Hgas-Hnh3)/20 // kJ/(K 97.8 kmol gas) NH3 free gas
Go = 97.8/.99995  //kmol
```

```
NH3a = (2.2-.005)*17  // kg
F1 = NH3a/.04 // flowrate of 4% NH3 solution
Water = F1-NH3a //kg
dT1 = Hgas/(Water*4.1868)  // K
Twater = 307-dT1  //K
Wvp = 2.116  //kPa
P = 5101.325  //kPa
moist = Go*Wvp/(P-Wvp) // kg
W = Water + moist // total demineralised water
dTactual = Hgas/(W*4.1868) //K
// from table 5.59
dHf1 = -80.093  //kJmol NH3  of 4% NH3 sol
dHf2 = -46.11  //kJ/molNH3
H = dHf1-dHf2  // heat of 4% NH3 sol
Hevl = -(H*NH3a*1000)/17  // total heat evolved
// in absorber gas is further heated from 283K to 291.4K
Hsol = Hevl-(2854.1*(291.4-283.15)) // kJ
// c 0f 4% NH3 sol = c of water = 4.1868 kJ/kg K
dT2 = Hsol/(F1*4.1868)
To = 291.4+dT2
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

```
clear;
clc;

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


```

```
// 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
```

```
ia1 = 1.006*(303.15-273.15)+2556.4*.01805 //kJ/kg
    dry air
ia2 = 1.006*(322.8-273.15)
    +2591.5*.0832+(1.006+1.84*.0832)*(393-328.8)
fi2 = ia2-ia1
hlost = fi2-fi1 // kJ/h
// using Appendix IV
h = 720.94 //kJ/kg
lambdav = 2046.5 // kJ/kg
steami = (h+lambdav)*885 // kJ/h
fi4 = h*885 //kJ/h
qm2 = 885/evp
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

```
clear;
clc;

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


// Example 6.23
// Page 401
printf("Example 6.23, Page 401 \n \n");

// solution

// basis : weak liquor flowrate = 1060 kg/h
```

```
s1 = 1060*.04 //kg/h   solids in weak liquor
liqr = s1/.25 // kg/h  conc liquor leaving 4th
    effect
evp = 1060-liqr // kg/h
lambdas = 2046.3  // kJ/kg
Wf = 1060 // kg/h
C1f = 4.04
T1 = 422.6
Tf = 303
lambdav1 = 2114.4
// enthalpy balance of 1st effect
// Ws*lambdas = Wf*C1f*(T1-Tf) + (Wf-W1)*2114.4
//putting values we get
// Ws = 1345.57 - 1.033*W1
// 2nd effect
// W1 = 531.38+.510*W2
// 3rd effect
// W1 - 1.990*W2 = -1.027*W3
// 4th effect
// W2 - 1.983*W3 = -176.84
//solving above eqs
W1 = 862 // kg/h
W2 = 648.2 // kg/h
W3 = 416.7 // kg/h
Ws = 455.2 // kg/h
eco = evp/Ws  // kg evaporation/kg steam
spcon = 1/eco // kg steam/kg evaporation
printf("Specific heat consumption of the system is "
    +string(spcon)+" kg steam/kg evaporation.")
```

Scilab code Exa 6.24 Triple effect evaporation system

```
clear;
clc;

```

```
// 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

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.1
// Page 434
printf("Example 7.1, Page 434 \n \n");

// solution

// basis 100 kg as received coal
O2 = 18.04 //kg
nH2 = 2.79-(O2/8) //kg
printf("(a) \n \n Net H2 in coal = "+string(nH2)+"
   kg. \n \n \n (b) \n \n ")
cbW = 1.128*18 // kg
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");

```

```
// solution

// basis 1 kg crude oil
H2 = .125 // kg   burnt
H2O = H2*18/2
Lh = H2O*2442.5 //kJ
GCV = 45071
NCV = GCV-Lh  //kJ/kg oil
printf("NCV = "+string(NCV)+" kJ/kg.")
```

Scilab code Exa 7.3 Gaseous propane

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.3
// Page 444
printf("Example 7.3, Page 444 \n \n");

// solution

// basis 1 mol of gaseous propane
H2O = 4*18.0153  //g
NHV = 2219.17-(H2O*2442.5/1000)
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
```

```
// Page 451
printf("Example 7.5, Page 451 \n \n");

// solution

// basis 100 kg fuel
O2req = 4.331*32  // kg
rO2req = O2req/100
N2in = (79/21)*4.331  // kmol
AIRreq = O2req+N2in*28  //kg
rAIRreq = AIRreq/100
R = AIRreq/100
AIRspld = R*2  // kg/kg coal
O2spld = 4.331*2  // kmol
N2spld = N2in*2
N2coal = 2.05/28  // kmol
tN2 = N2spld+N2coal
moist = 1.395+(7/18)  // kmol
printf("(a) \n \n Theoretical O2 requirement per
    unit mass of coal = "+string(rO2req)+" kg. \n \n
    \n(b) \n \n Theoretical dry air requirement = "+
    string(rAIRreq)+" kg/kg coal.")
```

Scilab code Exa 7.6 Residue fuel oil sample

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.6
// Page 452
```

```
printf("Example 7.6, Page 452 \n \n");

// solution

// basis 100 kg of RFO
O2req = 9.786 //kmol
N2req = (79/21)*O2req  //kmol
AIRreq = O2req+N2req //kmol
rAIRreq = AIRreq*29/100
AIRspld = AIRreq*1.25
rAIRspld = AIRspld/100
// using table 7.11 and 7.12
xSO2 = .07/(55.925+5.695)  // kmol SO2/kmol wet gas
vSO2 = xSO2*10^6  // ppm
mSO2 = 4.48*10^6/(1696.14+102.51)
// at 523.15 K and 100.7 kPa
V = [(55.925+5.695)*8.314*523.15]/100.7  // m^3
cSO2 = (4.48*10^6)/V  // mg/m^3
//from fig 7.3
dp = 424.4 //K
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

```
clear;
clc;
```

```
// Stoichiometry
// Chapter 7
// Combustion


// Example 7.7
// Page 454
printf("Example 7.7, Page 454 \n \n");

// solution

// basis 100 kmol of dry flue gas
O2acntd = 11.4+4.2  // kmol
O2avlbl = (21/79)*84.4 // kmol
O2excs = 4.2 //kmol
O2unactd = O2avlbl-O2acntd
H2brnt = O2unactd*2
O2req = 11.4+O2unactd
pexcsAIR = O2excs*100/O2req
mH2brnt = H2brnt*2 // kg
mCbrnt = 11.4*12
r = mCbrnt/mH2brnt
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

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion
```

```
// Example 7.8
// Page 459
printf("Example 7.8, Page 459 \n \n");

// solution

// basis 100 kg of bagasse fired in th boiler
//(a)
O2req = 2.02 // kmol
N2in = (79/21)*O2req // kmol
AIRreq = (O2req+N2in)*29 // kg
rAIR = AIRreq/100
printf("(a) \n \n Theoretical air required = "+
    string(rAIR)+" kg dry air/kg fuel. \n \n \n (b) \
    n \n ")
// (b)
tflugas = 1.95/.1565  ///kmol
xcsO2N2 = tflugas - 1.95
x = (xcsO2N2-7.6)/4.76  // kmol
pxcsAIR = x*100/O2req
printf("Percent excess air = "+string(pxcsAIR)+". \n
     \n (c) \n \n ")
//(c)
pW = 100*.2677 // kPa     partial p of water vap
// from fig 6.13
dp = 339.85 //K
printf("Dew Point of flue gas = "+string(dp)+"K. \n
    \n \n (d) \n \n ")
// (d)
// from appendix IV
hfw = 292.97  //kJ/kg      enthalpy of feed water at
    343.15 K
Hss = 3180.15  // kJ/kg    enthalpy of super heated
    steam at 2.15 bar and 643.15K
Hgain = Hss - hfw
H6 = Hgain*2.6*100  // kJ     heat gained by water
```

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

Scilab code Exa 7.9 Stoker fired water tube boiler

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.9
// Page 465
printf("Example 7.9, Page 465 \n \n");

// solution

// using mean heat capacity data Table 7.21
// basis 100 kmol of dry flue gas
H7 = 1.0875*100*30.31*(423.15-298.15)
H71 = 3633.654*(423.15-298.15)
fi7 = H71*3900*.7671/162.2   // kJ/h
fi1 = 3.9*1000*26170     // kJ/h
// performing heat balance
Hsteamgen = 23546.07
eff = Hsteamgen*100/fi1   // overall efficiency rate
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");
```

```

// solution

// basis 100 kg of fuel
// material balance of carbon
CO2 = 7.092+.047 //kmol   in flue gases
N2 = 11.94*7.139/7.01
O2 = 11.94*7.139/7.01
flue = CO2+N2+O2
// material balance of O2
O2air = 21*N2/79
airin = N2+O2air
tO2in = O2air+.078  // O2 in burner
O2xcs = tO2in-9.864
// material balance of water vapour
moistfrmd = 5.45 // kmol    from combustion of H2
H = .0331 // kmol/kmol of dry air        humidity at
    100.7 kPa
moistair = H*104.482  //kmol
tmoist = moistfrmd+moistair
pxcsair = O2xcs*100/9.786
// now using table 7.32
H7 = 3391.203*(563.15-298.15)  //kJ
Ff = 400 // kg/h   fuel firing rate
tH = 2791.7-179.99 // kJ/kg    total heat supplied
   in boiler
fi5 = tH*4365  // kJ/h
fi8 = 5.45*18*Ff*2403.5/100  // kJ/h
GCVf = 42260  //kJ/kg
fi1 = Ff*GCVf
Fdryair = 104.482*29*Ff/100
Cha = 1.006+1.84*.0205  // kJ/kg dry air K
fi3 = Fdryair*Cha*(308.15-298.15)
fi2 = Ff*1.758*(353.15-298.15)
BOILEReff1 = fi5*100/fi1
NCVf = GCVf-(18.0153/2.016)*.109*2442.8  // kJ/kg
BOILEReff2 = fi5*100/(Ff*NCVf)
r = 4365/Ff // steam:fuel
```

```
BOILERcapacity = fi5/2256.9
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

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.12
// Page 478
printf("Example 7.12, Page 478 \n \n");

// solution

// basis 100 kmol of dry producer gas
C = 33*12 // kg
O2 = 18.5*32  //kg
H2 = 20*2 // kg
O2air = 21*51/79 // kmol
COALgassified = 396/.672  // kg
O2coal = COALgassified*.061/32  // kmol
tO2 = O2coal + O2air
```

```
O2steam = 18.5-tO2  // kmol
H2steam = 2*O2steam  // kmol
H2fuel = 20-H2steam
dryproducergas = 100*22.41/COALgassified  // Nm^3/kg coal
Pw = 2.642 // kPa
Ha = Pw/(100.7-Pw)  // kmol/kmol dry gas
water = Ha*100
moistproducergas = (100+water)*22.41/COALgassified  // Nm^3/kg coal
dryair = (51*28+O2air*32)/COALgassified  // kg/kg coal
tsteamsupplied = H2steam+water-(COALgassified*.026/18)  // kmol
steam = tsteamsupplied*18/COALgassified
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

```
clear;
clc;

// Stoichiometry
// Chapter 7
// Combustion


// Example 7.13
// Page 479
printf("Example 7.13, Page 479 \n \n");

```

```
// 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)+"")
```