Appendix D. Excel Macro Code

' a_HETP_exp_shortcut

```
Option Explicit

Dim wblist(1 To 3)
Dim shlist(1 To 2)
Dim weightpercent_w() 'weightpercent_w at each sampling pt
Dim molefraction_x() 'molefraction_x at each sampling pt
Dim zR 'R-section packing height
Dim zS 'S-section packing height

Sub HETP_model()

zR = 0.19 + 0.235
zS = 0.225 + 0.225 + 0.25
'1. refer to workbook and sheets by name
  wblist(1) = "march8-TeamW9.xlsm"
  wblist(2) = "march10-TeamF5.xlsm"
  wblist(3) = "march11-teamS5.xlsm"
  shlist(1) = "run 1"
  shlist(2) = "run 2"
  'equilibrium table assign to array
  Dim eqm_table()
    Workbooks("equilibrium table.xlsx").Activate
    ActiveWorkbook.Sheets("Sheet1").Activate
    eqm_table = Range("A1", Range("A1").End(xlDown).End(xlToRight))
  'Assign Nt for R section to Nt_array
  Dim Nt_array
    Workbooks("NT_vapour.xlsx").Activate
    ActiveWorkbook.Sheets("Sheet1").Activate
    Nt_array = Range("C2", Range("C2").End(xlDown))
'2. activate wb and sheet 8 and 9
  Dim wB
  For wB = 1 To 3
    Workbooks(wblist(wB)).Activate

      'delete all previous generated charts
      Application.DisplayAlerts = False
      On Error Resume Next
      Charts.Delete
      Application.DisplayAlerts = True
    '3. run 1 and run 2
    'must use activatewb or you will only active sheet in wb where code is running
    Dim sh
    For sh = 1 To 2
   ActiveWorkbook.Sheets(shlist(sh)).Activate
   weightpercent_w = Range("B44:B52") 'weightpercent_w is a 2 dim array, 9row by 1Column
      'bottom mixture flow rate B
      Dim B 'kmol/h
        B = moleflowrate(Range("B25").Value, weightpercent_w(8, 1))
      'overhead mixture flow rate D
      Dim D 'kmol/h
        D = moleflowrate(Range("E17").Value, weightpercent_w(1, 1))
      Dim f 'kmol/h
        f = moleflowrate(Range("B17").Value, weightpercent_w(9, 1))
      Dim R 'reflux ratio
        R = Range("E26").Value
      '*******************
      '3.1 calculate x at each sample point
      Range("E43").Select
      Range("E43") = "molefraction_x"
      '3.1.1 cal molefraction_x by call function fx1
      ReDim molefraction_x(1 To 9, 1 To 1)
      Dim i
      For i = 1 To 9 'row
```

```
    On Error GoTo Assign0toMolefraction
    'if no error anymore, then run following code
    molefraction_x(i, 1) = molepercent(weightpercent_w(i, 1))
    On Error GoTo 0
  Next i
  '3.1.2 assign molefraction_x to cells
  Range("E44:E52").Value = molefraction_x
  '*******************
  '3.2 R-section op.line
  '3.2.1 xD at distillate sample point(overhead)
  Dim xD
    xD = molefraction_x(1, 1)
  Dim s1
    s1 = molefraction_x(2, 1)
  Dim s2
    s2 = molefraction_x(3, 1)

  '3.2.2 y=R/(R+1)*x+1/(R+1)*xD
  'Y-axis of Rline
  Dim y_Rline
    y_Rline = yRline(R, xD, s1, s2)
  'assign y vs x in worksheet
  Dim nR
    nR = Rline(R, xD, s1, s2, y_Rline)

  '3.2.3 mole balance
  Dim l 'kmol/h R-section liquid
    l = R * D
  Dim V 'kmol/h R-section vapour
    V = (R + 1) * D
  '*******************
  '3.3 Feed line and q value
  '3.3.1 HANDOUT equation(30) get q, call fx3
  Dim zF 'feed mole fraction
    zF = molefraction_x(9, 1)
  Dim q
    q = qf(zF)
  '3.3.2 qline y=q/(q-1)*x+1/(1-q)*zF
  Dim s3
    s3 = molefraction_x(4, 1)
  'Y-axis of Rline
  Dim y_qline
    y_qline = yqline(q, zF, s3)
  'assign y vs x in worksheet
  Dim nQ
    nQ = Qline(zF, s3, y_qline)
'*******************
  '3.4 S-section op.line
  '3.4.1 xB at distillate sample point(overhead)
  Dim s4
    s4 = molefraction_x(5, 1)
  Dim s5
    s5 = molefraction_x(6, 1)
  Dim s6
    s6 = molefraction_x(7, 1)
  Dim xB
    xB = molefraction_x(8, 1)
  '3.4.2 y=Lb/Vb*x-B/Vb*xB
  'MB calculate Lb and Vb
  Dim Lb
    Lb = q * f + l
  Dim Vb
    Vb = q * f + l - B

  'Y_axis of S-line
  Dim y_Sline
```

```
            y_Sline = ySline(Lb, Vb, B, s3, s4, s5, s6, xB)

        'assign y vs x in worksheet
        Dim nS
            nS = Sline(y_Sline, molefraction_x)

        '3.5 assign equilibrium array to worksheet at same location
        Range("A77", Range("A77").Offset(UBound(eqm_table, 1) - 1, UBound(eqm_table, 2) - 1)) = eqm_table
        Range("A76").Value = "equilibrium line"

        '3.6 assign V and Vb value to excel
        Range("G5") = "V_Rsection[kmol/h]"
        Range("G5").Offset(1, 0) = V
        Range("G5").Offset(0, 1) = "Vb_Ssection[kmol/h]"
        Range("G5").Offset(1, 1) = Vb
        '3.6.1 assign L and Lb value to excel
        Range("I5") = "L_Rsection[kmol/h]"
        Range("I5").Offset(1, 0) = l
        Range("I5").Offset(0, 1) = "Lb_Ssection[kmol/h]"
        Range("I5").Offset(1, 1) = Lb

        '3.7 plot oplines, qline, and equilibrium lines in one graph
        'note: do everything before creating chart, cos after chart created. active sheet changed to chart, not sheet" run 1 or 2"
        Dim nplot1
            nplot1 = plot1(shlist, sh)

        '3.8 HETP=zR/Nt_R for Rsection
        'reactivate current WORKsheet (Not chart)
        ActiveWorkbook.Sheets(shlist(sh)).Activate
        'assign value to Nt_R from Nt_array
        Dim Nt_R
            Nt_R = Nt_array((wB - 1) * 2 + sh, 1)
        'HETP
        Dim HETP_R_shortcut
            If wB = 3 And sh = 2 Then
                    zR = 0.19  'S5 run2, s2 wrong data, so only pick xD to s1'
            End If
            HETP_R_shortcut = zR / Nt_R

        'assign value to excel sheet
        Range("H55") = "Nt_R"
        Range("H55").Offset(0, 1) = "HETP_R_shortcut"
        Range("H55").Offset(1, 0) = Nt_R
        Range("H55").Offset(1, 1) = HETP_R_shortcut

      Next sh
   Next wB


Exit Sub
Assign0toMolefraction: 'the userdefined name of error handler
   molefraction_x(i, 1) = molefraction_x(i - 1, 1) 'on error, assign last value(value at i-1)to this value
   Resume Next



End Sub
' b_HETP_exp_byNTUmodel
Option Explicit
        '3.6 NTU model f=1/(ys-y),

        'ys=equilibrium y: (y*)
        Range("F43") = "y*(equilibrium_y)"
        Dim ys(1 To 9)
            Dim iy
             For iy = 1 To 9
```

```
    ys(iy) = ystar(molefraction_x, iy)
    Range("F43").Offset(iy, 0) = ys(iy)
 Next iy
'ys is a vector

'3.6.1 R-section: y=y_Rline
Dim sim_y(1 To 16)
Dim sim_ystar(1 To 16)
Dim f_R(1 To 16)
Dim sim_x(1 To 16)
sim_y(1) = y_Rline(3)

Dim intv
    intv = (y_Rline(1) - y_Rline(3)) / 15

If wB = 3 And sh = 2 Then  'S5 run2, s2 wrong data, so only pick xD to s1'
    sim_y(1) = y_Rline(2)
    intv = (y_Rline(1) - y_Rline(2)) / 15
End If
Dim sim_i
For sim_i = 1 To 15
    sim_y(sim_i + 1) = sim_y(sim_i) + intv
Next sim_i
Dim sim_i2
For sim_i2 = 1 To 16
    sim_x(sim_i2) = ((R + 1) * sim_y(sim_i2) - xD) / R
    sim_ystar(sim_i2) = ystar_sim(sim_x(sim_i2))
    f_R(sim_i2) = 1 / (sim_ystar(sim_i2) - sim_y(sim_i2))
Next sim_i2
'NTU_R_section
Dim NTU_R
    NTU_R = 0
Dim count1
    For count1 = 1 To 15
NTU_R = NTU_R + 0.5 * ((sim_y(count1 + 1) - sim_y(count1)) * (f_R(count1) + f_R(count1 + 1)))
    Next count1
Range("E55") = "NTU_R_section"
Range("E56") = NTU_R
'HTU_R_section
If wB = 3 And sh = 2 Then
    zR = 0.19
End If
Dim HTU_R
    HTU_R = zR / NTU_R

Range("F55") = "HTU_R_section"
Range("F56") = HTU_R

'slope m_Rsection
Dim m_R
    m_R = m(molefraction_x(1, 1), molefraction_x(3, 1))
    If wB = 3 And sh = 2 Then
    m_R = m(molefraction_x(1, 1), molefraction_x(2, 1))
    End If
'HETP_Rsection
Dim HETP_R 'log() in VBA is natural log
    HETP_R = HTU_R * Log(m_R * V / l) / (m_R * V / l - 1)
    Range("G55") = "HETP_R_section"
    Range("G56") = HETP_R


'3.6.2 S-section: y=y_Sline
Dim R_pts
    R_pts = 3
Dim S_pts
S_pts = 7 - R_pts '4 sample pts in Ssection: s3,s4,s5,s6 (EXCLUDE xB, since zS not include xB!!!)
```

```
        Dim f_Sline()
        ReDim f_Sline(1 To S_pts) 'f_Sline = 1 / (ys - y_Sline)
        'f_Sline is a vector
        Range("C69") = "y*_S-section"
        Range("D69") = "f_Sline"
          Dim count2
          For count2 = 1 To S_pts
            Range("C69").Offset(count2, 0) = ys(R_pts + count2)
            f_Sline(count2) = 1 / (ys(R_pts + count2) - y_Sline(count2))
            Range("D69").Offset(count2, 0) = f_Sline(count2)
          Next count2

        'NTU_S_section
        Dim NTU_S
          NTU_S = 0
          For count2 = 1 To S_pts - 1
            NTU_S = NTU_S + 0.5 * ((y_Sline(count2) - y_Sline(count2 + 1)) * (f_Sline(count2) + f_Sline(count2 + 1)))
          Next count2
        Range("E69") = "NTU_S_section"
        Range("E70") = NTU_S
        'HTU_S_section
        Dim HTU_S
          HTU_S = zS / NTU_S
        Range("F69") = "HTU_S_section"
        Range("F70") = HTU_S
        'slope m_Ssection
        Dim m_S
          m_S = m(molefraction_x(S_pts, 1), molefraction_x(7, 1)) 'Does not include xB
        'HETP_Ssection
        Dim HETP_S 'log() in VBA is natural log
          HETP_S = HTU_S * Log(m_S * Vb / Lb) / (m_S * Vb / Lb - 1)
          Range("G69") = "HETP_S_section"
          Range("G70") = HETP_S


    Next sh
  Next wB


Exit Sub
Assign0toMolefraction: 'the userdefined name of error handler
  molefraction_x(i, 1) = molefraction_x(i - 1, 1) 'on error, assign last value(value at i-1)to this value
  Resume Next
End Sub
```

'c_HETP_predicted

```
     '3.6 HETP_pred_Rsection
      Dim m_v_overhead
        m_v_overhead = Vap_massflowrate(V, y_Rline(1))
      Dim m_L_overhead
        m_L_overhead = liq_massflowrate(l, xD)

      Dim m_v_s2
        m_v_s2 = Vap_massflowrate(V, y_Rline(3))
      Dim m_L_s2
        m_L_s2 = liq_massflowrate(l, s2)

      If wB = 3 And sh = 2 Then
          m_v_s2 = Vap_massflowrate(V, y_Rline(2))
          m_L_s2 = liq_massflowrate(l, s1)
      End If

      Dim m_v_bottom
        m_v_bottom = Vap_massflowrate(Vb, y_Sline(5))
      Dim m_L_bottom
        m_L_bottom = liq_massflowrate(Lb, xB)
```

```
        'feed: Lf=f*q, Vf=f-Lf=f*(1-q) [kmol/h]
        Dim m_v_feed
          m_v_feed = Vap_massflowrate(f * (1 - q), y_qline(1))
        Dim m_L_feed
          m_L_feed = liq_massflowrate(f * q, zF)

        Range("G43") = "vapor mass flow rate[kg/h]"
        Range("G43").Offset(0, 1) = "liquid mass flow rate[kg/h]"
        Range("G43").Offset(1, 0) = m_v_overhead
        Range("G43").Offset(1, 1) = m_L_overhead
        Range("G43").Offset(8, 0) = m_v_bottom
        Range("G43").Offset(8, 1) = m_L_bottom
        Range("G43").Offset(9, 0) = m_v_feed
        Range("G43").Offset(9, 1) = m_L_feed
       'overhead
        Dim HETP_predicted_Rsection_overhead
          HETP_predicted_Rsection_overhead = HETP_pred(R, xD, y_Rline(1), m_v_overhead, m_L_overhead)
        's2
        Dim HETP_predicted_Rsection_s2
          HETP_predicted_Rsection_s2 = HETP_pred(R, s2, y_Rline(3), m_v_s2, m_L_s2)
        If wB = 3 And sh = 2 Then
          HETP_predicted_Rsection_s2 = HETP_pred(R, s1, y_Rline(2), m_v_s2, m_L_s2)
        End If
        'take average
        Dim AVG_HETP_pred_Rsection
          AVG_HETP_pred_Rsection = 0.5 * (HETP_predicted_Rsection_overhead + HETP_predicted_Rsection_s2)

          Range("A105") = "HETP_pred_Rsection[m]"
          Range("A105").Offset(1, 0) = "HETP_pred_overhead"
          Range("A105").Offset(1, 1) = "HETP_pred_s2"
          Range("A105").Offset(1, 2) = "AVG_HETP_pred_Rsection"
          Range("A105").Offset(2, 0) = HETP_predicted_Rsection_overhead
          Range("A105").Offset(2, 1) = HETP_predicted_Rsection_s2
          Range("A105").Offset(2, 2) = AVG_HETP_pred_Rsection

        Next sh
    Next wB


Exit Sub
Assign0toMolefraction: 'the userdefined name of error handler
    molefraction_x(i, 1) = molefraction_x(i - 1, 1) 'on error, assign last value(value at i-1)to this value
    Resume Next
End Sub
```

'd_pressureDrop_predicted

```
'3.6 DeltaP_pred
        Dim m_v_s5
          m_v_s5 = Vap_massflowrate(Vb, y_Sline(3))
        Dim m_v_s2
          m_v_s2 = Vap_massflowrate(V, y_Rline(3))
        Dim m_L_s5
          m_L_s5 = liq_massflowrate(Lb, s5)
        Dim m_L_s2
          m_L_s2 = liq_massflowrate(l, s2)

        Dim avg_m_v
          avg_m_v = 0.5 * (m_v_s5 + m_v_s2)
        Dim avg_m_L
          avg_m_L = 0.5 * (m_L_s5 + m_L_s2)

        Dim DeltaP_predicted_Takahashi
          DeltaP_predicted_Takahashi = delta_P_pred_Takahashi(avg_m_v, avg_m_L, (s2 + s5) / 2, (y_Rline(3) + y_Sline(3)) / 2)
          Range("A35") = "PressureDrop_pred_Takahashi[Pa]"
          Range("A35").Offset(1, 0) = "PressureDrop_pred_Takahashi"
          Range("A35").Offset(2, 0) = DeltaP_predicted_Takahashi
        Dim DeltaP_predicted_Robbins
```

```
            DeltaP_predicted_Robbins = delta_P_pred_Robbins(avg_m_v, avg_m_L, (s2 + s5) / 2, (y_Rline(3) + y_Sline(3)) / 2)
            Range("B35") = "PressureDrop_pred_Robbins[Pa]"
            Range("B35").Offset(1, 0) = "PressureDrop_pred_Robbins"

            Range("B35").Offset(2, 0) = DeltaP_predicted_Robbins
          'experimental pressure drop unit conversion( in of water to Pa)
          Range("B32").Offset(1, 0) = Range("B32").Value * 248.84
          Range("B32").Offset(1, 1) = "Pa"
          Range("B32").Offset(1, -1) = "Experimental pressureDrop"
          Next sh
    Next wB


Exit Sub
Assign0toMolefraction: 'the userdefined name of error handler
    molefraction_x(i, 1) = molefraction_x(i - 1, 1) 'on error, assign last value(value at i-1)to this value
    Resume Next

End Sub
```

'fx1_weightToMole

```
Option Explicit
'weightpercent to molepercent
Function molepercent(weightpercent)
   Dim MWc 'MW of CH3OH
    MWc = 32.04 'g/mol
    Dim MWh 'MW of H2O
    MWh = 18 'g/mol
    molepercent = (1 / MWc) / (1 / MWc + (1 - weightpercent / 100) / (weightpercent / 100 * MWh)) '/100 is to convert percent to fraction
End Function

'massflowrate to moleflowrate
Function moleflowrate(massflowrate, weightpercent)
    Dim MWc 'MW of CH3OH
    MWc = 32.04 'g/mol
    Dim MWh 'MW of H2O
    MWh = 18 'g/mol
    moleflowrate = massflowrate * ((weightpercent / 100) / (MWc) + (1 - (weightpercent / 100)) / MWh) '/100 is to convert percent to fraction
End Function
```

'fx2_R_line

```
Option Explicit

Function yRline(R, xD, s1, s2)
    'x values:
    Dim Xaxis(1 To 3)
    Xaxis(1) = xD
    Xaxis(2) = s1
    Xaxis(3) = s2
    Dim yaxis(1 To 3)
    Dim x
    Dim counter
    For Each x In Xaxis
        counter = counter + 1
        yaxis(counter) = R / (R + 1) * x + 1 / (R + 1) * xD
    Next x
    yRline = yaxis
End Function

Function Rline(R, xD, s1, s2, y_Rline)
'assign value to excel
          Range("A54").Value = "Rline"
          Range("A55").Value = "Xaxis"
          Range("A56") = xD
          Range("A57") = s1
          Range("A58") = s2
          Range("B55").Value = "Yaxis"
```

```
        Dim ra
        ra = y_Rline
        Range("B56").Value = ra(1)
        Range("B57").Value = ra(2)
        Range("B58").Value = ra(3)
End Function
```

'fx3_q_line

```
Option Explicit

Function qf(zF)
Dim Hv
Dim HF
Dim HL
' units all [kJ/Kg]
Dim TF '[C]
  TF = Range("B13")
Dim cp_H, cp_C 'kJ/(Kg*C)
  cp_H = 4.188 - 5.69 * 10 ^ -4 * TF + 8.49 * 10 ^ -6 * TF ^ 2
  cp_C = 2.33 + 7.82 * 10 ^ -3 * TF + 3.77 * 10 ^ -5 * TF ^ 2
Dim w 'weight fraction
  w = Range("B52") / 100

HF = (cp_H * (1 - w) + cp_C * w) * TF
Hv = 2658.6 - 1231 * zF
HL = 434.34 - 939.501 * zF + 1106.461 * (zF) ^ 2 - 437.716 * (zF) ^ 3

qf = (Hv - HF) / (Hv - HL)
Range("C62") = "q_value"
Range("C62").Offset(0, 1) = qf
End Function


Function yqline(q, zF, s3)
  'x values:
  Dim Xaxis(1 To 2)
  Xaxis(1) = zF
  Xaxis(2) = s3
  Dim yaxis(1 To 3)
  Dim x
  Dim counter
  For Each x In Xaxis
    counter = counter + 1
    yaxis(counter) = q / (q - 1) * x + 1 / (1 - q) * zF
  Next x
  yqline = yaxis
End Function

Function Qline(zF, s3, y_qline)
'assign values to excel
        Range("A62").Value = "qline"
        Range("A63").Value = "Xaxis"
        Range("A64") = zF
        Range("A65") = s3
        Range("B63").Value = "Yaxis"
        Dim ra1
        ra1 = y_qline
        Range("B64").Value = ra1(1)
        Range("B65").Value = ra1(2)
End Function
```

'fx4_S_line

```
Option Explicit

Function ySline(Lb, Vb, B, s3, s4, s5, s6, xB)
  'x values:
  Dim Xaxis(1 To 5)
```

```
    Xaxis(1) = s3
    Xaxis(2) = s4
    Xaxis(3) = s5
    Xaxis(4) = s6
    Xaxis(5) = xB
    Dim yaxis(1 To 5)
    Dim x
    Dim counter
    For Each x In Xaxis
        counter = counter + 1
        yaxis(counter) = Lb / Vb * x - B / Vb * xB
    Next x
    ySline = yaxis

End Function
Function Sline(y_Sline, molefraction_x)
'assign values to Excel
    Dim ra2
            ra2 = y_Sline
    Range("A68").Select
    ActiveCell.Value = "S-line"
    ActiveCell.Offset(1, 0).Value = "Xaxis"
    ActiveCell.Offset(1, 1).Value = "Yaxis"
    Dim count
    For count = 1 To 5
        'Xaxis
        ActiveCell.Offset(count + 1, 0).Value = molefraction_x(count + 3, 1)
        'Yaxis
        ActiveCell.Offset(count + 1, 1).Value = ra2(count)
    Next count
End Function
```

'fx5_plot1

```
Option Explicit
Function plot1(shlist, sh)

            Dim xRange1 As Range
            Dim xRange2 As Range
            Dim xRange3 As Range
            Dim xRange4 As Range
            Dim y1 As Range 'eqm line
            Dim y2 As Range 'Rline
            Dim y3 As Range 'qline
            Dim y4 As Range 'Sline
            Dim legend(1 To 4)
            legend(1) = Range("A76").Value
            legend(2) = Range("A54").Value
            legend(3) = Range("A62").Value
            legend(4) = Range("A68").Value
            Dim Figure1 As Chart

            Set xRange1 = Range("B78", Range("B78").End(xlDown))
            Set xRange2 = Range("A56", Range("A56").End(xlDown))
            Set xRange3 = Range("A64", Range("A64").End(xlDown))
            Set xRange4 = Range("A70", Range("A70").End(xlDown))
            Set y1 = xRange1.Offset(0, 1)
            Set y2 = xRange2.Offset(0, 1)
            Set y3 = xRange3.Offset(0, 1)
            Set y4 = xRange4.Offset(0, 1)

            Dim yaxis(1 To 4)
            yaxis(1) = y1
            yaxis(2) = y2
            yaxis(3) = y3
            yaxis(4) = y4

            'select empty area
```

```
Range("G1").Select
Set Figure1 = Charts.Add
Figure1.Move after:=Sheets(shlist(sh))

'editing chart:
With Figure1
  .ChartType = xlXYScatter

  'insert series 1 to empty seriescollection
  .SeriesCollection.NewSeries
  .SeriesCollection(1).XValues = xRange1
  .SeriesCollection(1).Values = y1.Value
  .SeriesCollection(1).MarkerStyle = xlMarkerStyleCircle

  'insert series 2 to empty seriescollection
  .SeriesCollection.NewSeries
  .SeriesCollection(2).XValues = xRange2
  .SeriesCollection(2).Values = y2.Value
  .SeriesCollection(2).MarkerStyle = xlMarkerStyleX
  'insert series 3 to empty seriescollection
  .SeriesCollection.NewSeries
  .SeriesCollection(3).XValues = xRange3
  .SeriesCollection(3).Values = y3.Value
  .SeriesCollection(3).MarkerStyle = xlMarkerStyleDiamond
  'insert series 4 to empty seriescollection
  .SeriesCollection.NewSeries
  .SeriesCollection(4).XValues = xRange4
  .SeriesCollection(4).Values = y4.Value
  .SeriesCollection(4).MarkerStyle = xlMarkerStyleStar

  'y=x line
  .SeriesCollection.NewSeries
  Dim xll(1 To 2)
   xll(1) = 0
   xll(2) = 1
  Dim yll(1 To 2)
   yll(1) = 0
   yll(2) = 1
  .SeriesCollection(5).XValues = xll
  .SeriesCollection(5).Values = yll
  .SeriesCollection(5).MarkerStyle = xlMarkerStyleNone

  'change linestyle/weight of series at the end
  .SeriesCollection(1).Format.Line.DashStyle = msoLineSolid
  .SeriesCollection(1).Format.Line.Weight = 1
  .SeriesCollection(5).Format.Line.DashStyle = msoLineSolid
  .SeriesCollection(5).Format.Line.Weight = 1
  'chart title
  .HasTitle = False
  'Adding axis titles
  .Axes(xlCategory).HasTitle = True 'xlcategory is xaxis
  .Axes(xlCategory).AxisTitle.Text = "x(mole)"

  .Axes(xlValue).HasTitle = True 'xlValue is yaxis
  .Axes(xlValue).AxisTitle.Text = "y(mole)"

  'adding legends to series
  .HasLegend = True
  Dim s
  Set s = Figure1.SeriesCollection
  s(1).Name = legend(1)
  s(2).Name = legend(2)
  s(3).Name = legend(3)
  s(4).Name = legend(4)
  s(5).Name = "y=x"
  'modifying Legend position to the Right
```

```
            .SetElement (msoElementLegendRight)

            'add border to chart area
             With .PlotArea.Format.Line
                .Visible = msoCTrue
                .Style = msoLineSingle
                .Weight = 1
            End With
         End With
            'remove gridline
            Dim axs
            For Each axs In Figure1.Axes
                axs.HasMajorGridlines = False
                axs.HasMinorGridlines = False
            Next axs
End Function
```

'fx7_equilibrium_y

```
Option Explicit

Function ystar(molefraction_x, iy)
 'VIP: number ^ exp , need space between number and ^ and exponential!!!!!!
        ystar = -20.8777 * molefraction_x(iy, 1) ^ 6 + 72.4933 * molefraction_x(iy, 1) ^ 5 - 100.3125 * molefraction_x(iy, 1) ^ 4 _
            + 70.9239 * molefraction_x(iy, 1) ^ 3 - 27.4288 * molefraction_x(iy, 1) ^ 2 + 6.1815 * molefraction_x(iy, 1) + 0.0136

End Function

Function ystar_sim(x)
        ystar_sim = -20.8777 * x ^ 6 + 72.4933 * x ^ 5 - 100.3125 * x ^ 4 _
            + 70.9239 * x ^ 3 - 27.4288 * x ^ 2 + 6.1815 * x + 0.0136

End Function
```

'fx8_slope_m

```
Option Explicit

Function m(x1, x2) 'input pt1 and pt2, calculate avg slope
    Dim m1, m2
        m1 = -6 * 20.8777 * x1 ^ 5 + 5 * 72.4933 * x1 ^ 4 - 4 * 100.3125 * x1 ^ 3 _
            + 3 * 70.9239 * x1 ^ 2 - 2 * 27.4288 * x1 + 6.1815
        m2 = -6 * 20.8777 * x2 ^ 5 + 5 * 72.4933 * x2 ^ 4 - 4 * 100.3125 * x2 ^ 3 _
            + 3 * 70.9239 * x2 ^ 2 - 2 * 27.4288 * x2 + 6.1815
    m = 0.5 * (m1 + m2)
End Function
```

'fx9_HETP_pred

```
Option Explicit
'NOTE: function name can't be the same as sub name in main code, it will confuse VBA
Function HETP_pred(Reflux_Ratio, x, y, m_v, m_L)
'only calculate for R section, Artin said


'input mass flow rate in [kg/hr]
Dim namda, u_GS, u_LS, k_G, k_L, ae

'1.namda
Dim slope_opLine
    slope_opLine = Reflux_Ratio / (Reflux_Ratio + 1)
Dim m 'slope of equilibrium line at mole fraction=x
    m = -6 * 20.8777 * x ^ 5 + 5 * 72.4933 * x ^ 4 - 4 * 100.3125 * x ^ 3 _
            + 3 * 70.9239 * x ^ 2 - 2 * 27.4288 * x + 6.1815

namda = m / slope_opLine

'2. u_GS, u_LS [m/s]
Dim A, rho_v, rho_L
'm_v:mass flow rate of vapor [kg/hr] from lab data
        A = (Application.WorksheetFunction.Pi() * 0.1 ^ 2) / 4
```

```
        rho_L = 920.614 - 363.411 * x + 263.143 * x ^ 2 - 73.895 * x ^ 3 'kg/m3
        rho_v = 0.5493 + 0.4518 * y + 0.078 * y ^ 2 'kg/m3

        u_GS = 1 / 3600 * (m_v / (A * rho_v)) 'm/s
        u_LS = 1 / 3600 * (m_L / (A * rho_L)) 'm/s

'3.ae 'm2/m3
Dim at
    at = 341 'm2/m3 =ap
Dim sigma_c, sigma, L_m, mu_L
Dim g 'm/s2
    g = 9.81
    sigma_c = 75 * 0.001 'N/m
    sigma = 1 / 1000 * (58.8602 - 23.1408 * x - 32.048 * x ^ 2 + 15.5362 * x ^ 3) 'N/m
    L_m = 1 / 3600 * m_L / A '[kg/(m2*s)](m_L:mass flow rate of liquid [kg/hr] from lab data
    If x < 0.31 Or x = 0.31 Then 'mu_L in [kg/(m*s)]
        mu_L = 2.863 * 10 ^ (-4) + 5.3724 * 10 ^ (-4) * x - 1.6676 * 10 ^ (-3) * x ^ 2 + 2.0517 * 10 ^ (-3) * x ^ 3
    Else
        mu_L = 3.236 * 10 ^ (-4) + 1.6497 * 10 ^ (-4) * x - 1.7563 * 10 ^ (-4) * x ^ 2 + 5.6272 * 10 ^ (-5) * x ^ 3
    End If


'[m2/m3] ONDA model
ae = at * (1 - Exp(-1.45 * (sigma_c / sigma) ^ 0.75 * (L_m / (at * mu_L)) ^ 0.1 * (L_m ^ 2 * at / (rho_L ^ 2 * g)) ^ (-0.05) * (L_m ^ 2 / (rho_L * sigma * at)) ^ 0.2))


'4. k_L [m/s]
Dim D_L, d_p

    'D_L: liquid diffusivity [m2/s]
    D_L = 3.655 * 10 ^ -9 - 9.4207 * 10 ^ -10 * Log(x) - 8.0893 * 10 ^ -11 * Log(x ^ 2)

    'D_p: nominal packing size [m]
    d_p = 0.0159

'm/s
k_L = (mu_L * g / (rho_L)) ^ (1 / 3) * 0.0051 * (L_m / (ae * mu_L)) ^ (2 / 3) * (mu_L / (rho_L * D_L)) ^ (-1 / 2) * (at * d_p) ^ (0.4)


'5. k_G [m/s]
Dim D_G '=D_v: diffusivity of gas in [m2/s]
    D_G = 2.3848 * 10 ^ -5 - 3.3964 * 10 ^ -6 * y - 1.1056 * 10 ^ -6 * y ^ 2
Dim G_m
    G_m = 1 / 3600 * m_v / A '[kg/(m2*s)](m_v:mass flow rate of vapor [kg/hr] from lab data
Dim mu_G 'in [kg/(m*s)]
    mu_G = 1.2416 * 10 ^ -5 + 5.3146 * 10 ^ -7 * y - 2.9807 * 10 ^ -6 * y ^ 2 + 9.7197 * 10 ^ -7 * y ^ 3

k_G = at * D_G * 5.23 * (G_m / (at * mu_G)) ^ 0.7 * (mu_G / (rho_v * D_G)) ^ (1 / 3) * (at * d_p) ^ -2


'6. Final HETP_pred 'all in SI units
HETP_pred = Log(namda) / (namda - 1) * (u_GS / (k_G * ae) + namda * u_LS / (k_L * ae))


End Function
```

‘fx10_PressureDrop_pred

```
Option Explicit
Function u_G_S(m_v, y) 'in m/s

Dim A, rho_v
'm_v:mass flow rate of vapor [kg/hr] from lab data
        A = (Application.WorksheetFunction.Pi() * 0.1 ^ 2) / 4 '[m]'

        rho_v = 0.5493 + 0.4518 * y + 0.078 * y ^ 2 'kg/m3
```

```
        u_G_S = 1 / 3600 * (m_v / (A * rho_v)) 'm/s

End Function

Function u_L_S(m_L, x) 'in m/s

Dim A, rho_L
'm_L:mass flow rate of liquid [kg/hr] from lab data
        A = (Application.WorksheetFunction.Pi() * 0.1 ^ 2) / 4 '[m]'
        rho_L = 920.614 - 363.411 * x + 263.143 * x ^ 2 - 73.895 * x ^ 3 'kg/m3
        u_L_S = 1 / 3600 * (m_L / (A * rho_L)) 'm/s

End Function


Function delta_P_pred_Takahashi(m_v, m_L, x, y) 'Total pressure drop in Pa
'u_GS, u_LS in m/s
Dim u_GS, u_LS
    u_GS = u_G_S(m_v, y)
    u_LS = u_L_S(m_L, x)

Dim epsilon, z, k, d_p, u_Gsp, u_Lsp, mu_G, mu_L, mu_W, rho_G, rho_L, Re_G, Re_L, f, H_L

'a_p = 341
epsilon = 0.933 'porosity
z = 0.915 'distance between pressure taps
k = 52200  'for Pall Ring packing
'd_p = 6 * (1 - epsilon) / a_p 'particle diameter
d_p = 0.0159
u_Gsp = u_GS * 3600 'gas velocity in m/hr
u_Lsp = u_LS * 3600 'liquid velocity in m/hr

'vapour viscosity in kg/m/hr
mu_G = ((1.2416E-05) + (5.3146E-07) * y - (2.9807E-06) * (y ^ 2) + (9.7197E-07) * (y ^ 3)) * 3600
'liquid viscosity in kg/m/hr
If x <= 0.31 Then
    mu_L = ((0.0002863) + (0.00053724) * x - (0.0016676) * (x ^ 2) + (0.0020517) * (x ^ 3)) * 3600
Else
    mu_L = ((0.0003236) + (0.00016497) * x - (0.00017563) * (x ^ 2) + (5.6272E-05) * (x ^ 3)) * 3600
End If
'water viscosity in kg/m/hr
'average temperature of 65 and 100 = 82.5 degC
'at T = 90, mu = 0.315E-03
'at T = 80, mu = 0.354E-03
'at T = 82.5, mu = (0.354-0.315)/(80-90)*(82.5-80)+0.354 = 0.344E-03
mu_W = 0.000344 * 3600

rho_G = 0.5493 + 0.4518 * y + 0.078 * (y ^ 2) 'vapour density
rho_L = 920.614 - 363.411 * x + 263.143 * (x ^ 2) - 73.895 * (x ^ 3) 'liquid density
Re_G = rho_G * u_Gsp * d_p / (mu_G * epsilon) 'vapour Reynold's number
Re_L = rho_L * u_Lsp * d_p / (mu_L * epsilon) 'liquid Reynold's number

'friction factor
If Re_G < 200 Then
    f = 114 * Re_G ^ (-0.742)
Else
    f = 6.85 * Re_G ^ (-0.216)
End If

H_L = (0.000153 + 2.9E-05 * epsilon * (Re_L ^ 0.66) * ((mu_L / mu_W) ^ 0.75)) * (d_p ^ -1.2)  'liquid hold up


'Total pressure drop in Pa
delta_P_pred_Takahashi = 1 / (3600 ^ 2) * z * (4 * f / d_p * ((u_Gsp / (epsilon - H_L)) ^ 2) * (rho_G / 2) + k * (H_L ^ 3) * ((u_Gsp / (epsilon - H_L))
^ 2))
```

```
End Function


Function delta_P_pred_Robbins(m_v, m_L, x, y) 'Total pressure drop in Pa

Dim A, L_m, G_m
A = (Application.WorksheetFunction.Pi() * 0.1 ^ 2) / 4
' L and G in lb/hr/ft^2
L_m = m_L / A * 0.2048
G_m = m_v / A * 0.2048

Dim z, mu_G, mu_L, rho_G, rho_L, Fp, Gf, Lf, C1, C2, delta_Pz

z = 0.915 'distance between pressure taps in m
C1 = 7.4E-08
C2 = 2.7E-05

'viscosity in cP
mu_G = ((1.2416E-05) + (5.3146E-07) * y - (2.9807E-06) * (y ^ 2) + (9.7197E-07) * (y ^ 3)) * (10 ^ 3)
If x <= 0.31 Then
   mu_L = ((0.0002863) + (0.00053724) * x - (0.0016676) * (x ^ 2) + (0.0020517) * (x ^ 3)) * (10 ^ 3)
Else
   mu_L = ((0.0003236) + (0.00016497) * x - (0.00017563) * (x ^ 2) + (5.6272E-05) * (x ^ 3)) * (10 ^ 3)
End If

'density in lb/ft^3
rho_G = (0.5493 + 0.4518 * y + 0.078 * (y ^ 2)) * 0.062435
rho_L = (920.614 - 363.411 * x + 263.143 * (x ^ 2) - 73.895 * (x ^ 3)) * 0.06243

'Fpd in ft
Fp = 80

Gf = G_m * ((0.075 / rho_G) ^ 0.5) * ((Fp / 20) ^ 0.5)
Lf = L_m * (62.4 / rho_L) * ((Fp / 20) ^ 0.5) * ((mu_L) ^ 0.1)

'delta P/z in Pa/m
delta_Pz = 249.174 * 3.2808 * (C1 * (Gf ^ 2) * (10 ^ (C2 * Lf)) + 0.4 * ((Lf / 20000) ^ 0.1) * (C1 * (Gf ^ 2) * (10 ^ (C2 * Lf))) ^ 4)

delta_P_pred_Robbins = delta_Pz * z


End Function
```

'fx11_moleToWeight

```
Option Explicit


Function liq_massflowrate(moleflowrate, x) '[kg/hr]
'input in [kmol/h]
   Dim MWc 'MW of CH3OH
   MWc = 32.04 'g/mol
   Dim MWh 'MW of H2O
   MWh = 18 'g/mol
   liq_massflowrate = moleflowrate * (MWc * x + (1 - x) * MWh)
End Function

Function Vap_massflowrate(moleflowrate, y) '[kg/hr]
'input in [kmol/h]
   Dim MWc 'MW of CH3OH
   MWc = 32.04 'g/mol
   Dim MWh 'MW of H2O
   MWh = 18 'g/mol
   Vap_massflowrate = moleflowrate * (MWc * y + (1 - y) * MWh)
End Function
```

'z_clean

```
Option Explicit
Sub selectwb()
   MsgBox "ATTENTION: please select corresponding workbook then execute delectingAllCharts() program in z_clean module in VBA(alt+F11)!"
End Sub
Sub deleteAllCharts()

   Dim note As Variant
   note = MsgBox("ATTENTION: clean program will delete all CHARTS in THIS workbook,continue?", vbOKCancel)
   If note = vbCancel Then
   Exit Sub
   Else
   End If

'select workbook then run
   Application.DisplayAlerts = False
   On Error Resume Next
   Charts.Delete
   Application.DisplayAlerts = True
End Sub
```

'f_Final_plots

```
Option Explicit
Dim wblist(1 To 3)
Dim shlist(1 To 2)

Sub finalplots()
   wblist(1) = "march8-TeamW9.xlsm"
   wblist(2) = "march10-TeamF5.xlsm"
   wblist(3) = "march11-teamS5.xlsm"
   shlist(1) = "run 1"
   shlist(2) = "run 2"
         'delete all previous generated charts
         Application.DisplayAlerts = False
         On Error Resume Next
         Charts.Delete
         Application.DisplayAlerts = True
   Dim wB
   Dim count
   count = 0
   For wB = 1 To 3
      Workbooks(wblist(wB)).Activate
      Dim sh
      For sh = 1 To 2
      ActiveWorkbook.Sheets(shlist(sh)).Activate

         Dim HETP_shortcut_R
            HETP_shortcut_R = Range("I56")
         Dim HETP_HTU_R
            HETP_HTU_R = Range("G56")
         Dim dP_exp
            dP_exp = Range("B33")
         Dim dP_Taka
            dP_Taka = Range("A37")
         Dim dP_Robbin
            dP_Robbin = Range("B37")
         Dim V_m_Rsection
            V_m_Rsection = Range("G44")
         Dim HETP_pred
            HETP_pred = Range("C107")
         Dim vap
            vap = Range("G44")
         Dim R
            R = Range("E26")

            ThisWorkbook.Activate
            ActiveWorkbook.Sheets("Sheet2").Activate
```

```
            count = count + 1
            Range("A1") = "HETP_shortcut_R"
            Range("A1").Offset(count, 0) = HETP_shortcut_R
            Range("B1") = "HETP_HTU_R"
            Range("B1").Offset(count, 0) = HETP_HTU_R
            Range("C1") = "dP_exp"
            Range("C1").Offset(count, 0) = dP_exp
            Range("D1") = "dP_Taka"
            Range("D1").Offset(count, 0) = dP_Taka
            Range("E1") = "dP_Robbin"
            Range("E1").Offset(count, 0) = dP_Robbin
            Range("F1") = "Vapor_massflowrate_R section"
            Range("F1").Offset(count, 0) = V_m_Rsection
            Range("G1") = "HETP_pred"
            Range("G1").Offset(count, 0) = HETP_pred

            Range("A12") = "Run index"
            Range("A12").Offset(count, 0) = "run" & "#" & count
            Range("B12") = "Descriptions"
            Range("B12").Offset(count, 0) = wblist(wB) & " " & shlist(sh)
            Range("C12") = "vapor mass flow rates[Kg/h]"
            Range("C12").Offset(count, 0) = vap
            Range("D12") = "Reflux ratio"
            Range("D12").Offset(count, 0) = R

        Workbooks(wblist(wB)).Activate
        ActiveWorkbook.Sheets(shlist(sh)).Activate
        Next sh
    Next wB

    ' HETP vs vapor_mass_flow_rate
    ThisWorkbook.Activate
    ActiveWorkbook.Sheets("Sheet2").Activate

    Dim nplot2
        nplot2 = plot2()

    'Delta_P vs vapor_mass_flow_rate
    ThisWorkbook.Activate
    ActiveWorkbook.Sheets("Sheet2").Activate

    Dim nplot3
        nplot3 = plot3()


End Sub
```

'fx_12_finalplots

```
Option Explicit

Function plot2()
        Dim xRange1 As Range
        Dim y1 As Range
        Dim y2 As Range
        Dim y3 As Range

        Dim legend(1 To 3)
        legend(1) = Range("A1").Value
        legend(2) = Range("B1").Value
        legend(3) = Range("G1").Value

        Dim Figure1 As Chart

        Set xRange1 = Range("F2", Range("F2").End(xlDown))
        Set y1 = Range("A2", Range("A2").End(xlDown))
```

```
Set y2 = Range("B2", Range("B2").End(xlDown))
Set y3 = Range("G2", Range("G2").End(xlDown))


Dim yaxis(1 To 3)
yaxis(1) = y1
yaxis(2) = y2
yaxis(3) = y3


'select empty area
Range("G100").Select
Set Figure1 = Charts.Add
Figure1.Move after:=Sheets("Sheet2")

'editing chart:
With Figure1
  .ChartType = xlXYScatter

  'insert series 1 to empty seriescollection
  .SeriesCollection.NewSeries
  .SeriesCollection(1).XValues = xRange1
  .SeriesCollection(1).Values = y1.Value
  .SeriesCollection(1).MarkerStyle = xlMarkerStyleCircle

  'insert series 2 to empty seriescollection
  .SeriesCollection.NewSeries
  .SeriesCollection(2).XValues = xRange1
  .SeriesCollection(2).Values = y2.Value
  .SeriesCollection(2).MarkerStyle = xlMarkerStyleX
  'insert series 3 to empty seriescollection
  .SeriesCollection.NewSeries
  .SeriesCollection(3).XValues = xRange1
  .SeriesCollection(3).Values = y3.Value
  .SeriesCollection(3).MarkerStyle = xlMarkerStyleDiamond

  .HasTitle = False

  'Adding axis titles
  .Axes(xlCategory).HasTitle = True 'xlcategory is xaxis
  .Axes(xlCategory).AxisTitle.Text = "Vapor mass flow rate(Rsection)[kg/h]"

  .Axes(xlValue).HasTitle = True 'xlValue is yaxis
  .Axes(xlValue).AxisTitle.Text = "HETP[m]"

  'adding legends to series
  .HasLegend = True
  Dim s
  Set s = Figure1.SeriesCollection
  s(1).Name = legend(1)
  s(2).Name = legend(2)
  s(3).Name = legend(3)

  'modifying Legend position to the Right
  '.SetElement (msoElementLegendRight)
  .SetElement (msoElementLegendRightOverlay)

  'add border to plot area
   With .PlotArea.Format.Line
     .Visible = msoCTrue
     .Style = msoLineSingle
     .Weight = 1
  End With

  'remove border for chart area
  .ChartArea.Border.LineStyle = xlNone
```

```
        End With
          'remove gridline
          Dim axs
          For Each axs In Figure1.Axes
             axs.HasMajorGridlines = False
             axs.HasMinorGridlines = False
          Next axs
End Function

Function plot3()

        Dim xRange1 As Range
        Dim y1 As Range
        Dim y2 As Range
        Dim y3 As Range


        Dim legend(1 To 3)
        legend(1) = Range("C1").Value
        legend(2) = Range("D1").Value
        legend(3) = Range("E1").Value

        Dim Figure1 As Chart

        Set xRange1 = Range("F2", Range("F2").End(xlDown))
        Set y1 = Range("C2", Range("C2").End(xlDown))
        Set y2 = Range("D2", Range("D2").End(xlDown))
        Set y3 = Range("E2", Range("E2").End(xlDown))


        Dim yaxis(1 To 3)
        yaxis(1) = y1
        yaxis(2) = y2
        yaxis(3) = y3


        'select empty area
        Range("G100").Select
        Set Figure1 = Charts.Add
        Figure1.Move after:=Sheets("Sheet2")

        'editing chart:
        With Figure1
          .ChartType = xlXYScatter

          'insert series 1 to empty seriescollection
          .SeriesCollection.NewSeries
          .SeriesCollection(1).XValues = xRange1
          .SeriesCollection(1).Values = y1.Value
          .SeriesCollection(1).MarkerStyle = xlMarkerStyleCircle

          'insert series 2 to empty seriescollection
          .SeriesCollection.NewSeries
          .SeriesCollection(2).XValues = xRange1
          .SeriesCollection(2).Values = y2.Value
          .SeriesCollection(2).MarkerStyle = xlMarkerStyleX
          'insert series 3 to empty seriescollection
          .SeriesCollection.NewSeries
          .SeriesCollection(3).XValues = xRange1
          .SeriesCollection(3).Values = y3.Value
          .SeriesCollection(3).MarkerStyle = xlMarkerStyleDiamond

          .HasTitle = False
          'Adding axis titles
          .Axes(xlCategory).HasTitle = True 'xlcategory is xaxis
```

```
            .Axes(xlCategory).AxisTitle.Text = "Vapor mass flow rate(Rsection)[kg/h]"

            .Axes(xlValue).HasTitle = True 'xlValue is yaxis
            .Axes(xlValue).AxisTitle.Text = "Pressure Drop[Pa]"

            'adding legends to series
            .HasLegend = True
            Dim s
            Set s = Figure1.SeriesCollection
            s(1).Name = legend(1)
            s(2).Name = legend(2)
            s(3).Name = legend(3)

            'modifying Legend position to the Right
            .SetElement (msoElementLegendRightOverlay)

            'add border to chart area
             With .PlotArea.Format.Line
               .Visible = msoCTrue
               .Style = msoLineSingle
               .Weight = 1
            End With

            'remove border for chart area
            .ChartArea.Border.LineStyle = xlNone

          End With
            'remove gridline
            Dim axs
            For Each axs In Figure1.Axes
               axs.HasMajorGridlines = False
               axs.HasMinorGridlines = False
            Next axs
End Function
```

'e_MBEB

```
Option Explicit
Dim wblist(1 To 3)
Dim shlist(1 To 2)

Sub MBEB()
'1. refer to workbook and sheets by name
   wblist(1) = "march8-TeamW9.xlsm"
   wblist(2) = "march10-TeamF5.xlsm"
   wblist(3) = "march11-teamS5.xlsm"
   shlist(1) = "run 1"
   shlist(2) = "run 2"


   Dim count
   count = 0
   '2. activate wb and sheet 8 and 9
   Dim wB
   For wB = 1 To 3
      Workbooks(wblist(wB)).Activate
      '3. run 1 and run 2
      'must use activatewb or you will only active sheet in wb where code is running
      Dim sh
      For sh = 1 To 2

         ActiveWorkbook.Sheets(shlist(sh)).Activate
         Dim HF '[kJ/Kg]
         Dim TF '[C]
            TF = Range("B13")
         Dim TD
            TD = Range("E13")
         Dim Tb
```

```
    Tb = Range("B21")

Dim cp_HF, cp_CF 'kJ/(Kg*C)
    cp_HF = 4.188 - 5.69 * 10 ^ -4 * TF + 8.49 * 10 ^ -6 * TF ^ 2
    cp_CF = 2.33 + 7.82 * 10 ^ -3 * TF + 3.77 * 10 ^ -5 * TF ^ 2
Dim wF 'weight fraction
    wF = Range("B52") / 100

HF = (cp_HF * (1 - wF) + cp_CF * wF) * TF

Dim hD '[kJ/Kg]
Dim cp_HD, cp_CD 'kJ/(Kg*C)
    cp_HD = 4.188 - 5.69 * 10 ^ -4 * TD + 8.49 * 10 ^ -6 * TD ^ 2
    cp_CD = 2.33 + 7.82 * 10 ^ -3 * TD + 3.77 * 10 ^ -5 * TD ^ 2
Dim wD 'weight fraction
    wD = Range("B44") / 100

hD = (cp_HD * (1 - wD) + cp_CD * wD) * TD

Dim hB '[kJ/Kg]
Dim cp_HB, cp_CB 'kJ/(Kg*C)
    cp_HB = 4.188 - 5.69 * 10 ^ -4 * Tb + 8.49 * 10 ^ -6 * Tb ^ 2
    cp_CB = 2.33 + 7.82 * 10 ^ -3 * Tb + 3.77 * 10 ^ -5 * Tb ^ 2
Dim wBot 'weight fraction
    wBot = Range("B51") / 100

hB = (cp_HB * (1 - wBot) + cp_CB * wBot) * Tb


Dim Fm 'kg/h
    Fm = Range("B17")
Dim Dm
    Dm = Range("E17")
Dim Bm
    Bm = Range("B25")

'1. Utility:
'condenser utility: CW
Dim Tcw(1 To 2)
    Tcw(1) = Range("E7") 'inlet
    Tcw(2) = Range("E8") 'outlet
Dim Tcw_avg
    Tcw_avg = 0.5 * (Tcw(1) + Tcw(2))
Dim cp_cw
    cp_cw = 4.188 - 5.69 * 10 ^ -4 * Tcw_avg + 8.49 * 10 ^ -6 * Tcw_avg ^ 2
Dim mcool
    mcool = Range("E9")

Dim Qcon_ut 'KJ/h
    Qcon_ut = mcool * cp_cw * (Tcw(2) - Tcw(1))

'reboiler utility: steam
Dim Tst(1 To 2)
    Tst(1) = Range("B7") 'steam inlet
    Tst(2) = Range("B8") 'steam outlet= condensed steam
Dim Tst_avg
    Tst_avg = 0.5 * (Tst(1) + Tst(2))
Dim cp_st 'first reduce T. then change phase, so cp should use vapor phase
    cp_st = 1.813 + 7.439 * 10 ^ -4 * Tst_avg - 1.123 * 10 ^ -7 * Tst_avg ^ 2

Dim mst
    mst = Range("B9")

Dim delta_H_st
    delta_H_st = 2481.1 - 1.821 * Tst(2) - 4.236 * 10 ^ -3 * Tst(2) ^ 2
```

```
Dim Qreb_ut 'KJ/h
   Qreb_ut = mst * delta_H_st + mst * cp_st * (Tst(1) - Tst(2))


'2. process:
'condenser process:
Dim Lm
   Lm = Range("E25")
Dim delta_H_C_cond
   delta_H_C_cond = 1218.4 - 1.3849 * TD - 6.402 * 10 ^ -3 * TD ^ 2
Dim delta_H_H_cond
   delta_H_H_cond = 2481.1 - 1.821 * TD - 4.236 * 10 ^ -3 * TD ^ 2
Dim delta_H_condprocess
   delta_H_condprocess = wD * delta_H_C_cond + (1 - wD) * delta_H_H_cond
Dim Qcon_process 'KJ/h
   Qcon_process = (Lm + Dm) * delta_H_condprocess


'reboiler process:
Dim mboilup '= Vb_mass
   mboilup = Range("G51")
Dim delta_H_C_reb
   delta_H_C_reb = 1218.4 - 1.3849 * Tb - 6.402 * 10 ^ -3 * Tb ^ 2
Dim delta_H_H_reb
   delta_H_H_reb = 2481.1 - 1.821 * Tb - 4.236 * 10 ^ -3 * Tb ^ 2

Dim delta_H_rebprocess
   delta_H_rebprocess = wBot * delta_H_C_reb + (1 - wBot) * delta_H_H_reb
Dim Qreb_process 'KJ/h
   Qreb_process = mboilup * delta_H_rebprocess


'3. Qloss_utility
Dim Qloss_ut
Dim Qin_ut
   Qin_ut = Fm * HF + Qreb_ut
   Qloss_ut = Qin_ut - (Dm * hD + Bm * hB + Qcon_ut)

'4. Qloss_process
Dim Qloss_process
Dim Qin_process
   Qin_process = Fm * HF + Qreb_process
   Qloss_process = Qin_process - (Dm * hD + Bm * hB + Qcon_process)

'5. Mass_difference
Dim totalmass_diff
   totalmass_diff = Fm - (Bm + Dm)
Dim componentmass_diff
   componentmass_diff = Fm * wF - (Bm * wBot + Dm * wD)

ThisWorkbook.Activate
     ActiveWorkbook.Sheets("Sheet2").Activate


     count = count + 1
     Range("I1").Offset(count, 0) = wblist(wB) & " " & shlist(sh)

     Range("J1") = "Qloss_utility[KJ/H]"
     Range("J1").Offset(count, 0) = Qloss_ut
     Range("K1") = "Qin_utility"
     Range("K1").Offset(count, 0) = Qin_ut
     Range("L1") = "Qloss_utility%"
     Range("L1").Offset(count, 0) = Qloss_ut / Qin_ut * 100

     Range("M1") = "Qloss_process[KJ/H]"
     Range("M1").Offset(count, 0) = Qloss_process
```

```
        Range("N1") = "Qin_process"
        Range("N1").Offset(count, 0) = Qin_process
        Range("O1") = "Qloss_process%"
        Range("O1").Offset(count, 0) = Qloss_process / Qin_process * 100


        Range("P1") = "totalmass_diff[kg/h]"
        Range("P1").Offset(count, 0) = totalmass_diff
        Range("Q1") = "totalmass_IN"
        Range("Q1").Offset(count, 0) = Fm
        Range("R1") = "totalmass_diff%"
        Range("R1").Offset(count, 0) = totalmass_diff / Fm * 100

        Range("S1") = "componentmass_diff[kg/h]"
        Range("S1").Offset(count, 0) = componentmass_diff
        Range("T1") = "componentmass_IN"
        Range("T1").Offset(count, 0) = Fm * wF
        Range("U1") = "componentmass_diff%"
        Range("U1").Offset(count, 0) = componentmass_diff / (Fm * wF) * 100

    Workbooks(wblist(wB)).Activate
    ActiveWorkbook.Sheets(shlist(sh)).Activate
    Next sh
  Next wB


End Sub
```