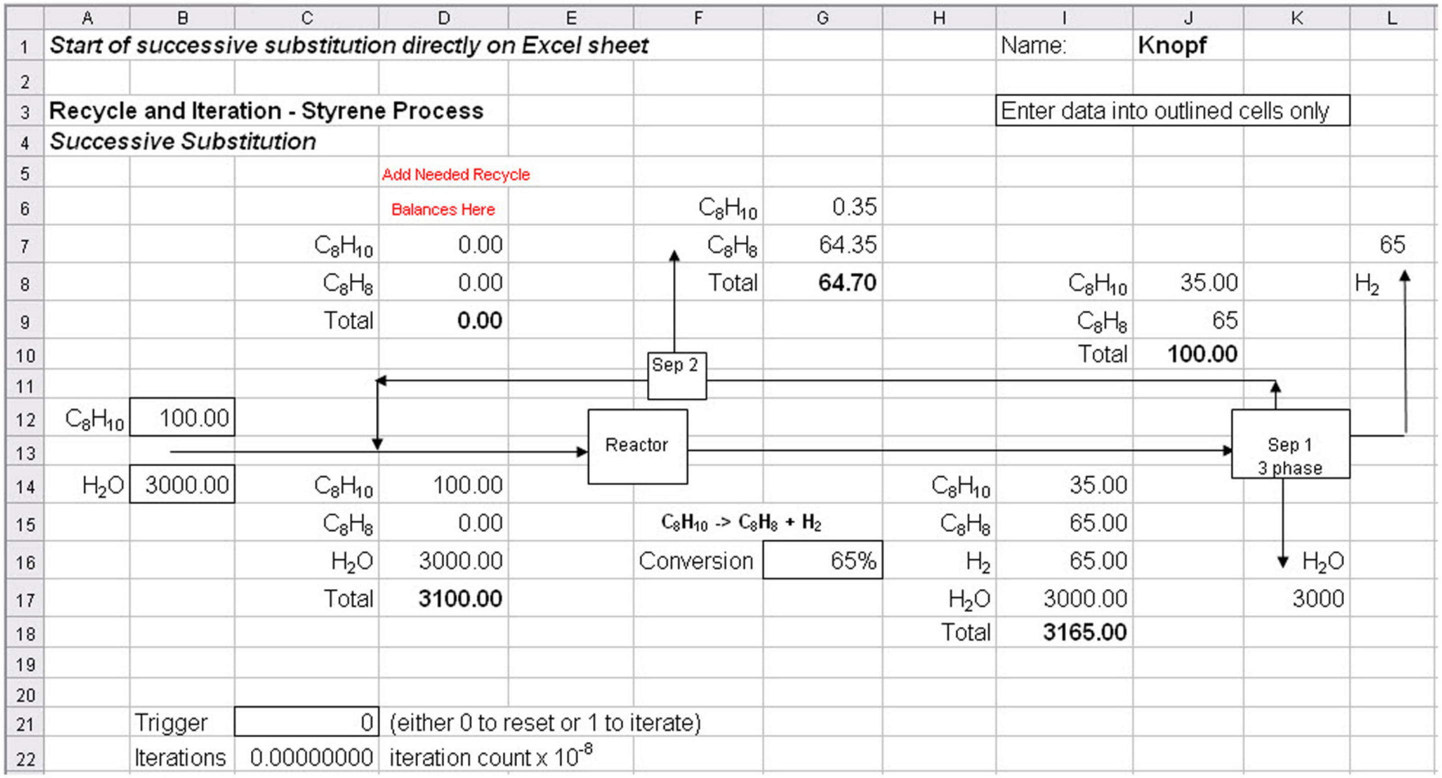
	A	B	C	D	E	F	G	H	I	J
1	*Styrene Production - Successive Substitution for Recycle Stream*									
2										
3		Trigger	0	(either 0 to reset or 1 to iterate)						
4		Iterations	0.00000000	iteration count x 10^{-8}						
5										
6		Reaction	C8H10 ---> C8H8 + H2		Conversion	0.65				
7										
8	**Iteration #k**									
9			(x_n^k)							$g_n(x_n^k)$
10		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
11	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
12	N(C8H8)	0	1.004691135	1.004691135	100.4691135	0	0	100.4691135	99.464422	1.004691135
13	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
14	N(H2O)	3000	0	3000	3000	3000	0	0	0	0
15	N(total)	3100	54.02687935	3154.026879	3253.491302	3000	99.46442234	154.0268794	100	54.02687935
16										
17										
18	**Iteration #k+1**									
19			(x_n^{k+1})							$g_n(x_n^{k+1})$
20		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
21	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
22	N(C8H8)	0	1.004691135	1.004691135	100.4691135	0	0	100.4691135	99.464422	1.004691135
23	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
24	N(H2O)	3000	0	3000	3000	3000	0	0	0	0
25	N(total)	3100	54.02687935	3154.026879	3253.491302	3000	99.46442234	154.0268794	100	54.02687935

FIGURE 3.10



Start of successive substitution directly on Excel sheet
Name: Knopf
Recycle and Iteration - Styrene Process
Enter data into outlined cells only
Successive Substitution
Add Needed Recycle
Balances Here
C8H10 0.35
C8H8 64.35
Total 64.70
C8H10 0.00
C8H8 0.00
Total 0.00
65
H2
C8H10 35.00
C8H8 65
Total 100.00
Sep 2
C8H10 100.00
Reactor
Sep 1
3 phase
H2O 3000.00
C8H10 100.00
C8H8 0.00
H2O 3000.00
Total 3100.00
C8H10 -> C8H8 + H2
Conversion 65%
C8H10 35.00
C8H8 65.00
H2 65.00
H2O 3000.00
Total 3165.00
H2O
3000
Trigger 0 (either 0 to reset or 1 to iterate)
Iterations 0.00000000 iteration count x 10-8

FIGURE 3.11

Styrene Production - Wegstein Method for Recycle Stream

Trigger	0	(either 0 to reset or 1 to iterate)
Iterations	0.00000000	iteration count x 10^{-8}
Precision	0.0000001	
Reaction	C8H10 ---> C8H8 + H2	Conversion 0.65

Iteration #1 (k=1)

	Feed	Recycle (x_n^1)	Reactor_In	Reactor_Out	Water_Out	Vapor_Out	Organic_Out	Product	New Recycle $g_n(x_n^1)$
N(C8H10)	100	0	100	35	0	0	35	0.35	34.65
N(C8H8)	0	0	0	65	0	0	65	64.35	0.65
N(H2)	0	0	0	65	0	65	0	0	0
N(H2O)	3000	0	3000	3000	3000	0	0	0	0
N(total)	3100	0	3100	3165	3000	65	100	64.7	35.3

Iteration #2 (k=2)

	Feed	Recycle (x_n^2)	Reactor_In	Reactor_Out	Water_Out	Vapor_Out	Organic_Out	Product	New Recycle $g_n(x_n^2)$
N(C8H10)	100	34.65	134.65	47.1275	0	0	47.1275	0.471275	46.656225
N(C8H8)	0	0.65	0.65	88.1725	0	0	88.1725	87.290775	0.881725
N(H2)	0	0	0	87.5225	0	87.5225	0	0	0
N(H2O)	3000	0	3000	3000	3000	0	0	0	0
N(total)	3100	35.3	3135.3	3222.8225	3000	87.5225	135.3	87.76205	47.53795

Calculate Wegstein parameters - Must solve for every species

Theta max	10
Theta min	-10

	Slope	Theta Calc	Theta Lower	Theta Upper	Wegstein Recycle (x_n^3)
N(C8H10)	0.3465	1.530221882	1.530221882	1.530221882	53.02218822
N(C8H8)	0.3565	1.554001554	1.554001554	1.554001554	1.01010101
N(H2)	0	1	1	1	0
N(H2O)	0	1	1	1	0
N(total)					

Iteration #3 (k=3)

	Feed	Recycle (x_n^3)	Reactor_In	Reactor_Out	Water_Out	Vapor_Out	Organic_Out	Product	New Recycle $g_n(x_n^3)$
N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
N(C8H8)	0	1.01010101	1.01010101	100.4745234	0	0	100.4745234	99.469778	1.004745234
N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
N(H2O)	3000	0	3000	3000	3000	0	0	0	0
N(total)	3100	54.03228923	3154.032289	3253.496712	3000	99.46442234	154.0322892	100.00536	54.02693345

FIGURE 3.13

```
Option Explicit

    Public Sub Flowsheet_Iteration()
        Dim n_components As Integer
        n_components = 4

        ' Dimension x and g values
        Dim x_old() As Double
        ReDim x_old(n_components)

        Dim g_old() As Double
        ReDim g_old(n_components)

        Dim x_current() As Double
        ReDim x_current(n_components)

        Dim g_current() As Double
        ReDim g_current(n_components)


        ' Call Successive Substitution for first two iterations
        Call Successive(x_old, g_old, x_current, g_current, n_components)

        ' Call Wegstein Method
        Call Wegstein(x_old, g_old, x_current, g_current, n_components)

    End Sub

    Sub Successive(x_old, g_old, x_current, g_current, n_components)

        ' Read x1 and resulting g(x1) values from sheet
        Dim i As Integer
        For i = 0 To n_components - 1
            x_old(i) = Sheet1.Cells(i + 11, 3)
            g_old(i) = Sheet1.Cells(i + 11, 10)
        Next i

        ' Place g(x1) values on sheet as new x2 values
        For i = 0 To n_components - 1
            Sheet1.Cells(i + 11, 3) = g_old(i)
        Next i

        ' Read x2 and resulting g(x2) values from sheet
        For i = 0 To n_components - 1
            x_current(i) = Sheet1.Cells(i + 11, 3)
            g_current(i) = Sheet1.Cells(i + 11, 10)
        Next i

    End Sub

    Sub Wegstein(x_old, g_old, x_current, g_current, n_components)

        Dim Slope() As Double
        ReDim Slope(n_components)

        Dim Theta() As Double
        ReDim Theta(n_components)
```

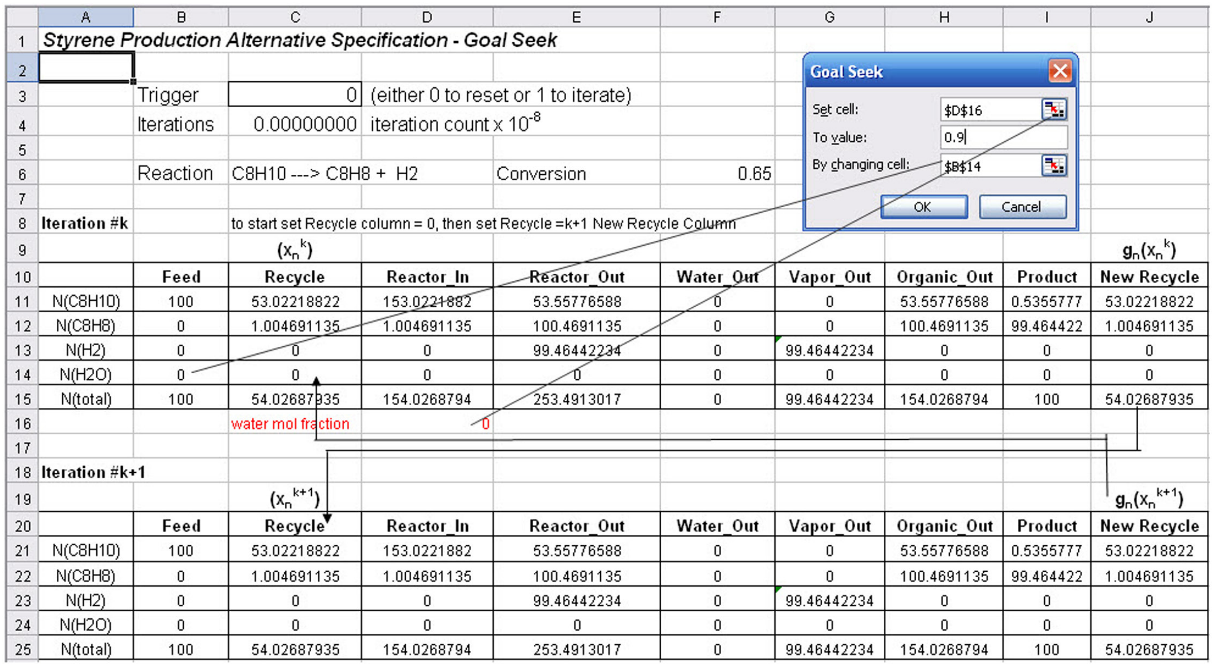
FIGURE 3.15a

```
    Dim x_new() As Double
    ReDim x_new(n_components)

   ' Set up Wegstein Method iteration loop using j as counter

    Dim j As Integer
    Dim i As Integer

    Dim Max_Iteration As Integer
    Max_Iteration = 10
    For j = 1 To Max_Iteration

        ' Then calculate Slope and Theta
        For i = 0 To n_components - 1
            If (x_current(i) - x_old(i) = 0.0#) Then
                Slope(i) = 0.0#
            Else
                Slope(i) = (g_current(i) - g_old(i)) / (x_current(i) - x_old(i))

            End If

            If ((1 - Slope(i)) = 0.0#) Then
                Theta(i) = 1
            Else
                Theta(i) = 1 / (1 - Slope(i))
            End If

            ' Check extrapolation bounds
            If Theta(i) >= 10.0# Then Theta(i) = 10.0#
            If Theta(i) <= -10.0# Then Theta(i) = -10.0#

            ' Calculate new x values for sheet
            x_new(i) = (1.0# - Theta(i)) * x_current(i) + Theta(i) * g_current(i)

        Next i

        ' Place new x values on sheet

        For i = 0 To n_components - 1
            Sheet1.Cells(i + 11, 3) = x_new(i)
        Next i

        ' Set x and g values
        For i = 0 To n_components - 1
            x_old(i) = x_current(i)
            g_old(i) = g_current(i)
            x_current(i) = Sheet1.Cells(i + 11, 3)
            g_current(i) = Sheet1.Cells(i + 11, 10)

        Next i

        ' Here we could check a convergence criterion parameter
        ' (Max_Change) to allow code exit or
        ' Continue to loop until Max_Iterations

    Next j
End Sub
```

FIGURE 3.15b

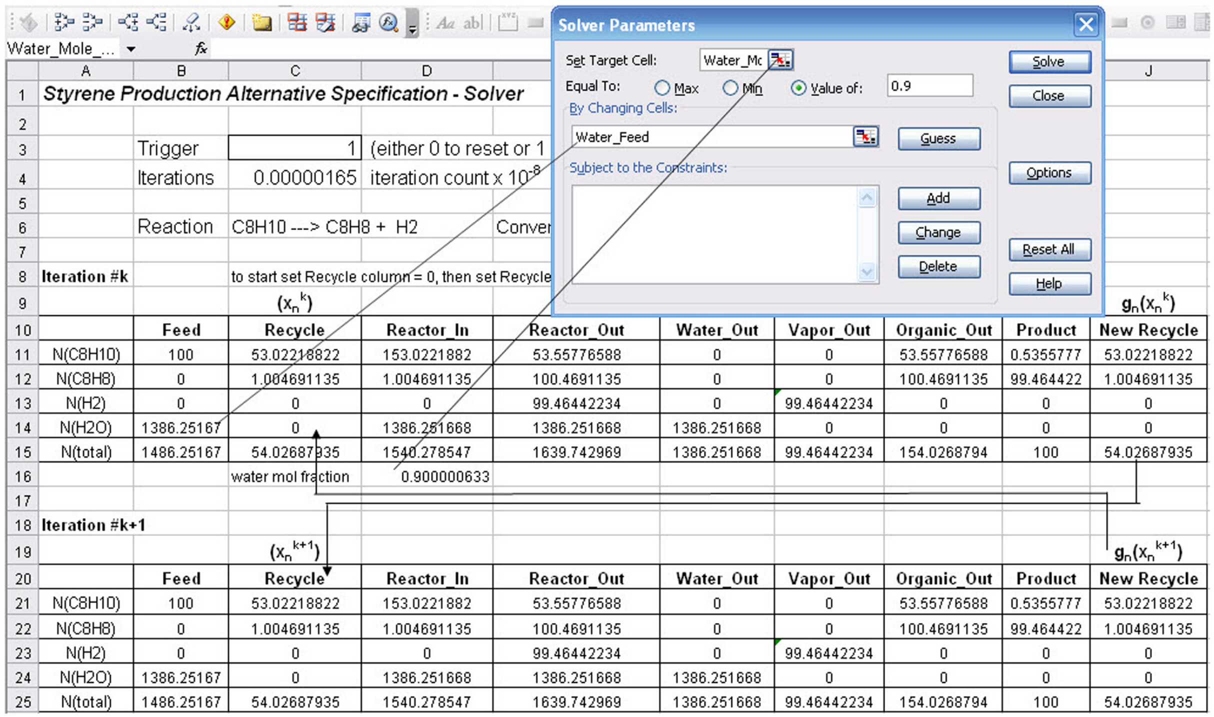
	A	B	C	D	E	F	G	H	I	J
1	***Styrene Production Alternative Specification - Trial and Error Solution***									
2										
3		Trigger	0	(either 0 to reset or 1 to iterate)						
4		Iterations	0.00000000	iteration count x 10^{-8}						
5										
6		Reaction	C8H10 ---> C8H8 + H2		Conversion	0.65				
7										
8	**Iteration #k**		to start set Recycle column = 0, then set Recycle =k+1 New Recycle Column							
9			(x_n^k)							$g_n(x_n^k)$
10		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
11	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
12	N(C8H8)	0	1.004691135	1.004691135	100.4691135	0	0	100.4691135	99.464422	1.004691135
13	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
14	N(H2O)	0	0	0	0	0	0	0	0	0
15	N(total)	100	54.02687935	154.0268794	253.4913017	0	99.46442234	154.0268794	100	54.02687935
16			water mol fraction	0						
17										
18	**Iteration #k+1**									
19			(x_n^{k+1})							$g_n(x_n^{k+1})$
20		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
21	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
22	N(C8H8)	0	1.004691135	1.004691135	100.4691135	0	0	100.4691135	99.464422	1.004691135
23	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
24	N(H2O)	0	0	0	0	0	0	0	0	0
25	N(total)	100	54.02687935	154.0268794	253.4913017	0	99.46442234	154.0268794	100	54.02687935

FIGURE 3.17



	A	B	C	D	E	F	G	H	I	J
1	*Styrene Production Alternative Specification - Goal Seek*									
2										
3		Trigger	0	(either 0 to reset or 1 to iterate)						
4		Iterations	0.00000000	iteration count x 10^{-8}						
5										
6		Reaction	C8H10 ---> C8H8 + H2		Conversion	0.65				
7										
8	**Iteration #k**		to start set Recycle column = 0, then set Recycle =k+1 New Recycle Column							
9			(x_n^k)							$g_n(x_n^k)$
10		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
11	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
12	N(C8H8)	0	1.004691135	1.004691135	100.4691135	0	0	100.4691135	99.464422	1.004691135
13	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
14	N(H2O)	0	0	0	0	0	0	0	0	0
15	N(total)	100	54.02687935	154.0268794	253.4913017	0	99.46442234	154.0268794	100	54.02687935
16			water mol fraction	0						
17										
18	**Iteration #k+1**									
19			(x_n^{k+1})							$g_n(x_n^{k+1})$
20		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
21	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
22	N(C8H8)	0	1.004691135	1.004691135	100.4691135	0	0	100.4691135	99.464422	1.004691135
23	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
24	N(H2O)	0	0	0	0	0	0	0	0	0
25	N(total)	100	54.02687935	154.0268794	253.4913017	0	99.46442234	154.0268794	100	54.02687935

FIGURE 3.18



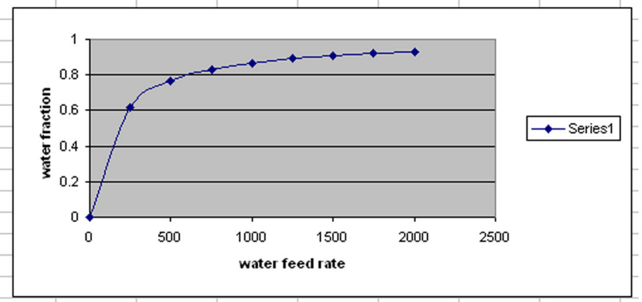
	A	B	C	D	E	F	G	H	I	J
1	*Styrene Production Alternative Specification - Solver*									
2										
3		Trigger	1	(either 0 to reset or 1						
4		Iterations	0.00000165	iteration count x 10^{-8}						
5										
6		Reaction	C8H10 ---> C8H8 + H2		Conver					
7										
8	**Iteration #k**		to start set Recycle column = 0, then set Recycle							
9			(x_n^k)							$g_n(x_n^k)$
10		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
11	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
12	N(C8H8)	0	1.004691135	1.004691135	100.4691135	0	0	100.4691135	99.464422	1.004691135
13	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
14	N(H2O)	1386.25167	0	1386.251668	1386.251668	1386.251668	0	0	0	0
15	N(total)	1486.25167	54.02687935	1540.278547	1639.742969	1386.251668	99.46442234	154.0268794	100	54.02687935
16			water mol fraction	0.900000633						
17										
18	**Iteration #k+1**									
19			(x_n^{k+1})							$g_n(x_n^{k+1})$
20		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
21	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.5355777	53.02218822
22	N(C8H8)	0	1.004691135	1.004691135	100.4691135	0	0	100.4691135	99.464422	1.004691135
23	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
24	N(H2O)	1386.25167	0	1386.251668	1386.251668	1386.251668	0	0	0	0
25	N(total)	1486.25167	54.02687935	1540.278547	1639.742969	1386.251668	99.46442234	154.0268794	100	54.02687935

FIGURE 3.19

	A	B	C	D	E	F	G	H	I	J
1	*Styrene Production Wegstein Method / VBA / WB Solver*									
2										
3		Trigger	1	(either 0 to reset or 1 to iterate)						
4		Iterations	0.00000097	iteration count x 10^{-8}						
5										
6		Reaction	C8H10 ---> C8H8 + H2		Conversion	0.65				
7										
8	**Iteration**									
9			(x_n)							$g_n(x_n)$
10		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Ou**	**Product**	**New Recycle**
11	N(C8H10)	100	53.02218822	153.022188	53.55776588	0	0	53.557766	0.535577659	53.02218822
12	N(C8H8)	0	1.004691135	1.00469113	100.4691135	0	0	100.46911	99.46442234	1.004691135
13	N(H2)	0	0	0	99.46442234	0	99.4644223	0	0	0
14	N(H2O)	1377.0512	0	1377.05116	1377.05116	1377.05116	0	0	0	0
15	N(total)	1477.0512	54.02687935	1531.07804	1630.542462	1377.05116	99.4644223	154.02688	100	54.02687935
16				0.89939972	H_20 mole fraction					
17				3.6034E-07	Objective Function					

FIGURE 3.20

	A	B	C
1	Objective Function Constrained Styrene Flowsheet		
2			
3	Water Feed	Water mole fraction	
4	moles/hr	Reactor_In	
5			
6	0	0	
7	250	0.618770712	
8	500	0.764494573	
9	750	0.82962135	
10	1000	0.866530943	
11	1250	0.890296346	
12	1500	0.906877644	
13	1750	0.919104672	
14	2000	0.92849352	
15			
16			
17			These cells contain
18			(Water mole fraction Reactor_In - 0.9)^2
19			
20	x	f(x)	
21	0	0.81	
22	250	0.079089912	
23	500	0.018361721	
24	750	0.004953154	
25	1000	0.001120178	
26	1250	9.41609E-05	
27	1500	4.7302E-05	
28	1750	0.000364988	
29	2000	0.000811881	
30			
31			
32			
33			
34			



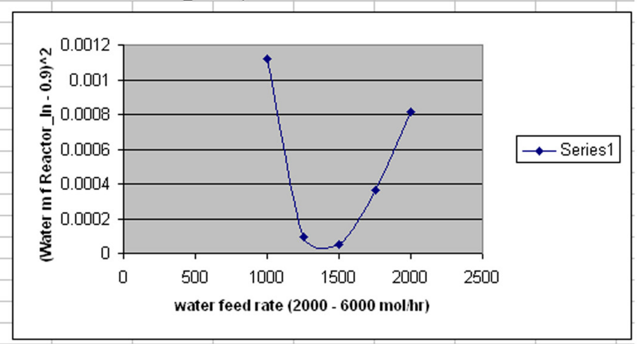


FIGURE 3.21

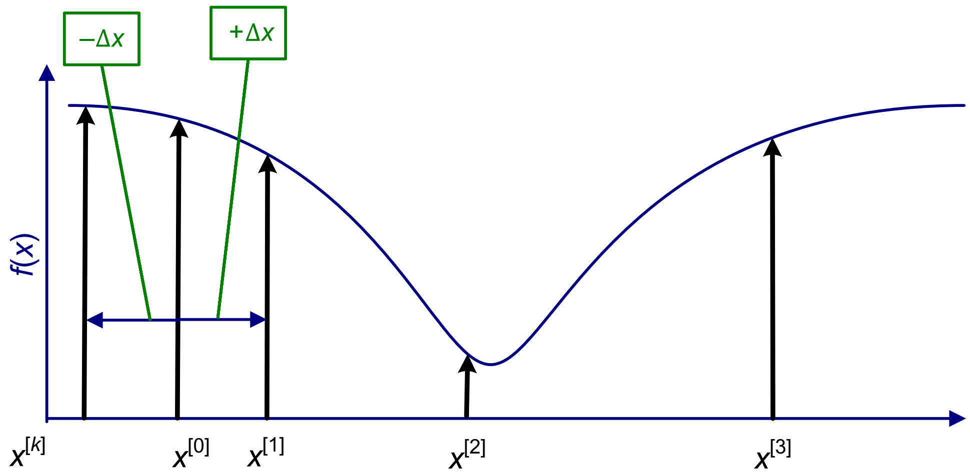


FIGURE 3.22

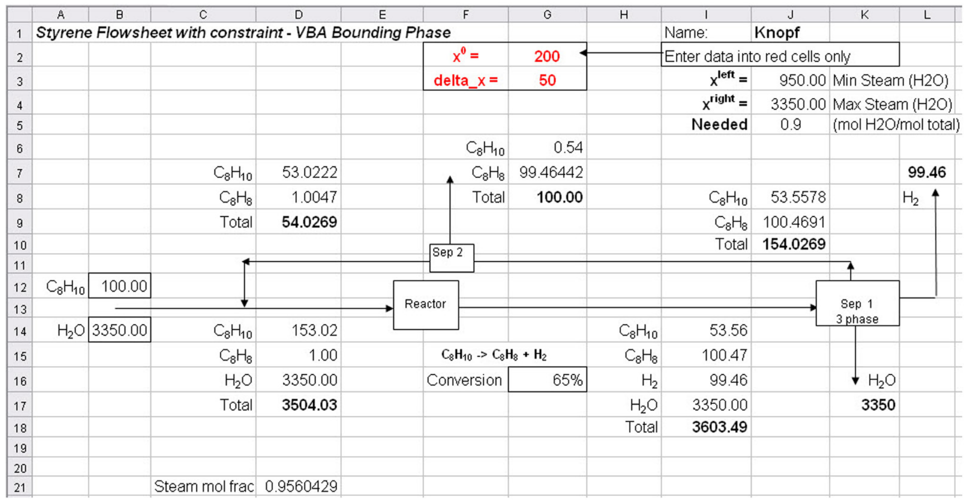
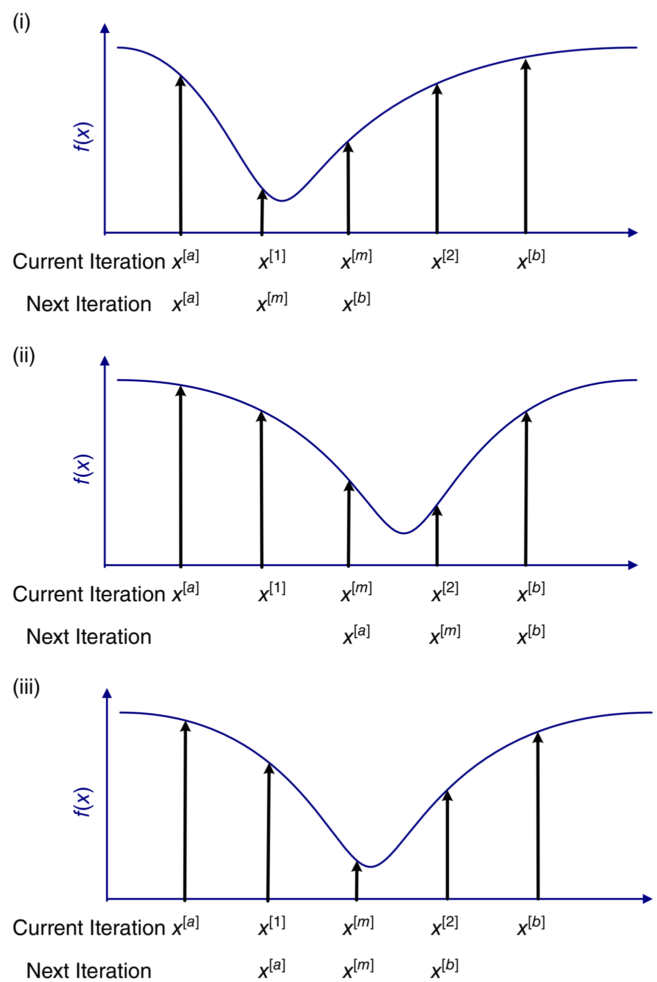

Styrene Flowsheet with constraint - VBA Bounding Phase
Name:
Knopf
x0 =
200
delta_x =
50
Enter data into red cells only
xleft =
950.00
Min Steam (H2O)
xright =
3350.00
Max Steam (H2O)
Needed
0.9
(mol H2O/mol total)
C8H10
0.54
C8H8
99.46442
Total
100.00
C8H10
53.0222
C8H8
1.0047
Total
54.0269
99.46
H2
C8H10
53.5578
C8H8
100.4691
Total
154.0269
Sep 2
C8H10
100.00
Reactor
Sep 1
3 phase
H2O
3350.00
C8H10
153.02
C8H8
1.00
H2O
3350.00
Total
3504.03
C8H10 -> C8H8 + H2
Conversion
65%
C8H10
53.56
C8H8
100.47
H2
99.46
H2O
3350.00
Total
3603.49
H2O
3350
Steam mol frac
0.9560429

FIGURE 3.23

```
Sub Bound()
    ' We want to bound the water feed rate between [xa, xb] or [xleft, xright]
    ' We get the initial guess, x0 now called x1, from cell G2 on the flowsheet

    x1 = Range("G2").Value
    ' We next calculate our objective function
    ' This is done on the flowsheet in a separate Sub Procedure
    Call Flowsheet(x1, F)
    f1 = F

    ' Next we determine the direction for the step delta_x

    delta_x = Range("G3").Value

    xright = x1 + delta_x
    Call Flowsheet(xright, F)
    fright = F
    xleft = x1 - delta_x
    Call Flowsheet(xleft, F)
    fleft = F

    Iteration_limit = 10000
    Iteration = 0
    Do
        If fleft > f1 And fright > f1 Then Exit Do

        If f1 > fright Then
            Iteration = Iteration + 1
            xleft = x1
            fleft = f1
            x1 = xright
            f1 = fright
            xright = x1 + (2 ^ Iteration) * delta_x
            Call Flowsheet(xright, F)
            fright = F

        Else
            Iteration = Iteration + 1
            xright = x1
            fright = f1
            x1 = xleft
            f1 = fleft
            xleft = x1 - (2 ^ Iteration) * delta_x
            ' Normally the next two statements (currenly comments) would not
            ' be used as the flow rate can not be negative
            '               If xleft < 0 Then xleft = 0
            '               If xleft = 0 Then Exit Do
            Call Flowsheet(xleft, F)
            fleft = F

        End If
    Loop Until Iteration > Iteration_limit
    ' Place bounded water flow on spreadsheet as [xa, xb] or [xleft, xright]
    Range("J3").Value = xleft
    Range("J4").Value = xright
End Sub
Sub Flowsheet(x, F)
    ' Here we place the water flow rate on the flowsheet in cell B14
    ' the flowsheet will converge before it returns control to the VBA code.
    Range("B14").Value = x
    ' The flowsheet is now converged with the x value (water flow rate)
    ' Cell D21 contains the converged (moles H2O)/(Total moles in Reator)
    ' Cell J5 contains the desired (moles H2O)/(Total moles in Reator)
    ' We square the difference between converged and desired (mole fraction)
    F = (Range("D21").Value - Range("J5").Value) ^ 2
    ' The value in F will be returned to the Bounding Sub Procedure
End Sub
```

FIGURE 3.24



(i)
f(x)
Current Iteration $x^{[a]}$ $x^{[1]}$ $x^{[m]}$ $x^{[2]}$ $x^{[b]}$
Next Iteration $x^{[a]}$ $x^{[m]}$ $x^{[b]}$
(ii)
f(x)
Current Iteration $x^{[a]}$ $x^{[1]}$ $x^{[m]}$ $x^{[2]}$ $x^{[b]}$
Next Iteration $x^{[a]}$ $x^{[m]}$ $x^{[b]}$
(iii)
f(x)
Current Iteration $x^{[a]}$ $x^{[1]}$ $x^{[m]}$ $x^{[2]}$ $x^{[b]}$
Next Iteration $x^{[a]}$ $x^{[m]}$ $x^{[b]}$

FIGURE 3.25

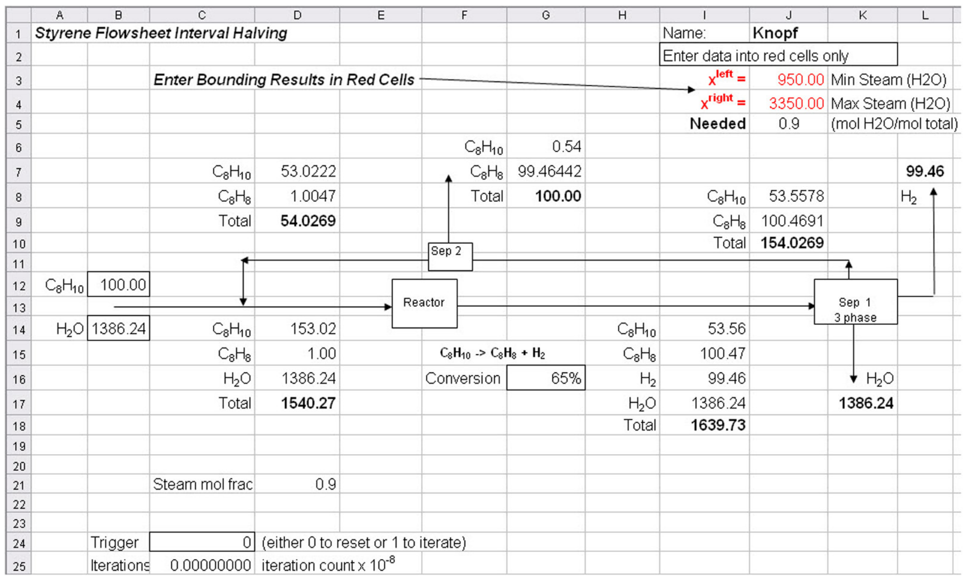

Styrene Flowsheet Interval Halving
Name: Knopf
Enter data into red cells only
Enter Bounding Results in Red Cells
x^left = 950.00 Min Steam (H2O)
x^right = 3350.00 Max Steam (H2O)
Needed 0.9 (mol H2O/mol total)
C8H10 0.54
C8H8 99.46442
Total 100.00
C8H10 53.0222
C8H8 1.0047
Total 54.0269
99.46
H2
C8H10 53.5578
C8H8 100.4691
Total 154.0269
Sep 2
C8H10 100.00
Reactor
Sep 1 3 phase
H2O 1386.24
C8H10 153.02
C8H8 1.00
H2O 1386.24
Total 1540.27
C8H10 -> C8H8 + H2
Conversion 65%
C8H10 53.56
C8H8 100.47
H2 99.46
H2O 1386.24
Total 1639.73
H2O
1386.24
Steam mol frac 0.9
Trigger 0 (either 0 to reset or 1 to iterate)
Iterations 0.00000000 iteration count x 10^-8

FIGURE 3.26

```
Sub Interval_Halving()
    ' We specify that our function has been bounded
    ' between [xa, xb] or [xleft, xright]
    ' and here xa and xb are given on the flowsheet (in cells J3 and J4,
respectively)
    '
    ' We get the lower bound, xa, from cell J3 on the flowsheet
    ' here xa is our lower bound on the water (steam) flow rate
    xa = Range("J3").Value
    ' Here we call our "function" - which is generally done in a separate Sub
Procedure
    Call Flowsheet(xa, F)
    fa = F
    ' Here we get the upper bound, xb, from cell J4 on the flowsheet
    xb = Range("J4").Value
    Call Flowsheet(xb, F)
    fb = F
    ' Here we determine the mid point
    xm = (xa + xb) / 2
    Call Flowsheet(xm, F)
    fm = F
    Max_Change = 0.000001
    Iteration_limit = 100000
    Iteration = 0
    Do
        Iteration = Iteration + 1
        delta_x = (xb - xa)
        If Abs(xb - xa) < Max_Change Then Exit Do
        ' Here we determine the points 1/4 and 3/4 of the total length
        x1 = xa + delta_x / 4
        Call Flowsheet(x1, F)
        f1 = F
        x2 = xb - delta_x / 4
        Call Flowsheet(x2, F)
        f2 = F
        If f1 < fm Then
            xa = xa
            xb = xm
            xm = x1
            fm = f1
        ElseIf f2 < fm Then
            xb = xb
            xa = xm
            xm = x2
            fm = f2
        Else
            xa = x1
            xb = x2
            xm = xm
            fm = fm
        End If
    Loop Until Iteration > Iteration_limit
    ' Here the optimum, xm, will be placed in the spreadsheet
    ' and one last convergence performed
    Range("B14").Value = xm
End Sub
Sub Flowsheet(x, F)
    ' Here we place the water flow rate on the flowsheet in cell B14
    ' the flowsheet will converge before it returns control to the VBA code.
    Range("B14").Value = x
    ' The flowsheet is now converged with the x value (water flow rate)
    ' Cell D21 contains the converged (moles H2O)/(Total moles in Reator)
    ' Cell J5 contains the desired (moles H2O)/(Total moles in Reator)
    ' We square the difference between converged and desired (mole fraction)
    F = (Range("D21").Value - Range("J5").Value) ^ 2
    ' F will be returned to the Interval Halving Sub Procedure
End Sub
```

FIGURE 3.27

	A	B	C	D	E	F	G	H	I	J
1	***Styrene Production Nonlinear Specification - Problem Start***									
2										
3		Trigger	0	(either 0 to reset or 1 to iterate)						
4		Iterations	0.00000000	iteration count x 10^{-8}						
5										
6		Reaction	C8H10 ---> C8H8 + H2		Conversion	0.65				
7										
8	**Iteration #1**		(k=1)							
9			(x_n^1)							$g_n(x_n^1)$
10		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
11	N(C8H10)	100	0	100	35	0	0	35	0.35	34.65
12	N(C8H8)	0	0	0	65	0	0	65	64.812884	0.187115573
13	N(H2)	0	0	0	65	0	65	0	0	0
14	N(H2O)	3000	0	3000	3000	3000	0	0	0	0
15	N(total)	3100	0	3100	3165	3000	65	100	65.162884	34.83711557
16										
17										
18	**Iteration #2**		(k=2)							
19			(x_n^2)							$g_n(x_n^2)$
20		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
21	N(C8H10)	100	34.65	134.65	47.1275	0	0	47.1275	0.471275	46.656225
22	N(C8H8)	0	0.187115573	0.187115573	87.70961557	0	0	87.70961557	87.458205	0.251410516
23	N(H2)	0	0	0	87.5225	0	87.5225	0	0	0
24	N(H2O)	3000	0	3000	3000	3000	0	0	0	0
25	N(total)	3100	34.83711557	3134.837116	3222.359616	3000	87.5225	134.8371156	87.92948	46.90763552

FIGURE 3.28

	A	B	C	D	E	F	G	H	I	J
1	*Styrene Production Nonlinear Specification*									
2										
3		Trigger	0	(either 0 to reset or 1 to iterate)						
4		Iterations	0.00000000	iteration count x 10^{-8}						
5										
6		Reaction	C8H10 ---> C8H8 + H2		Conversion	0.65	mol fraction C_8H_8 product		0.99464422	
7										
8	**Iteration #k**						mol fraction C_8H_{10} recycle		0.99464422	
9			(x_n^k)							$g_n(x_n^k)$
10		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
11	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.53557766	53.02218822
12	N(C8H8)	0	0.28550409	0.28550409	99.74992643	0	0	99.74992643	99.4644223	0.28550409
13	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
14	N(H2O)	3000	0	3000	3000	3000	0	0	0	0
15	N(total)	3100	53.30769231	3153.307692	3252.772115	3000	99.46442234	153.3076923	100	53.30769231
16										
17										
18	**Iteration #k+1**									
19			(x_n^{k+1})							$g_n(x_n^{k+1})$
20		**Feed**	**Recycle**	**Reactor_In**	**Reactor_Out**	**Water_Out**	**Vapor_Out**	**Organic_Out**	**Product**	**New Recycle**
21	N(C8H10)	100	53.02218822	153.0221882	53.55776588	0	0	53.55776588	0.53557766	53.02218822
22	N(C8H8)	0	0.28550409	0.28550409	99.74992643	0	0	99.74992643	99.4644223	0.28550409
23	N(H2)	0	0	0	99.46442234	0	99.46442234	0	0	0
24	N(H2O)	3000	0	3000	3000	3000	0	0	0	0
25	N(total)	3100	53.30769231	3153.307692	3252.772115	3000	99.46442234	153.3076923	100	53.30769231

FIGURE 3.29

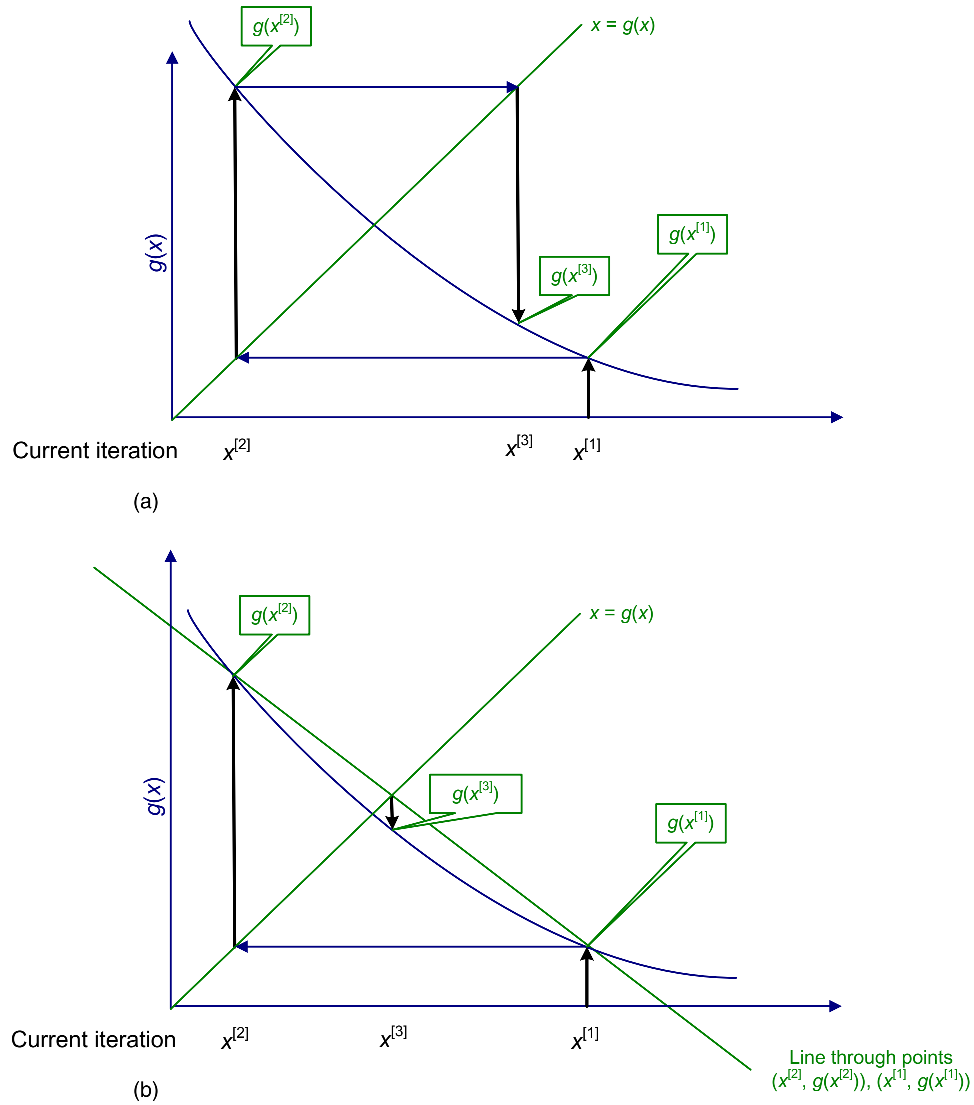

g(x^[2])
x = g(x)
g(x^[3])
g(x^[1])
g(x)
Current iteration
x^[2]
x^[3]
x^[1]
(a)
g(x^[2])
x = g(x)
g(x^[3])
g(x^[1])
g(x)
Current iteration
x^[2]
x^[3]
x^[1]
(b)
Line through points
(x^[2], g(x^[2])), (x^[1], g(x^[1]))

FIGURE 3.8

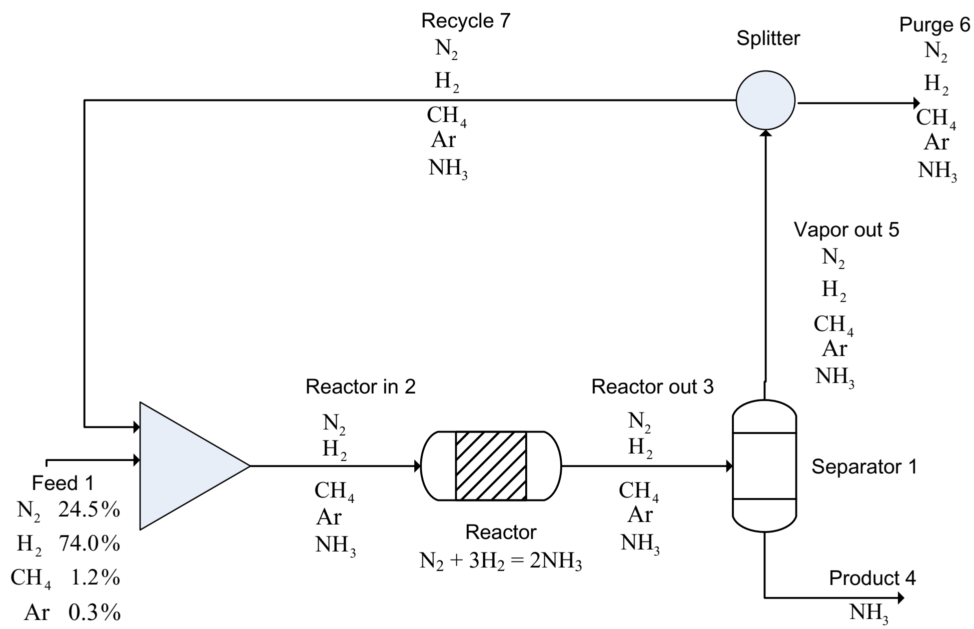
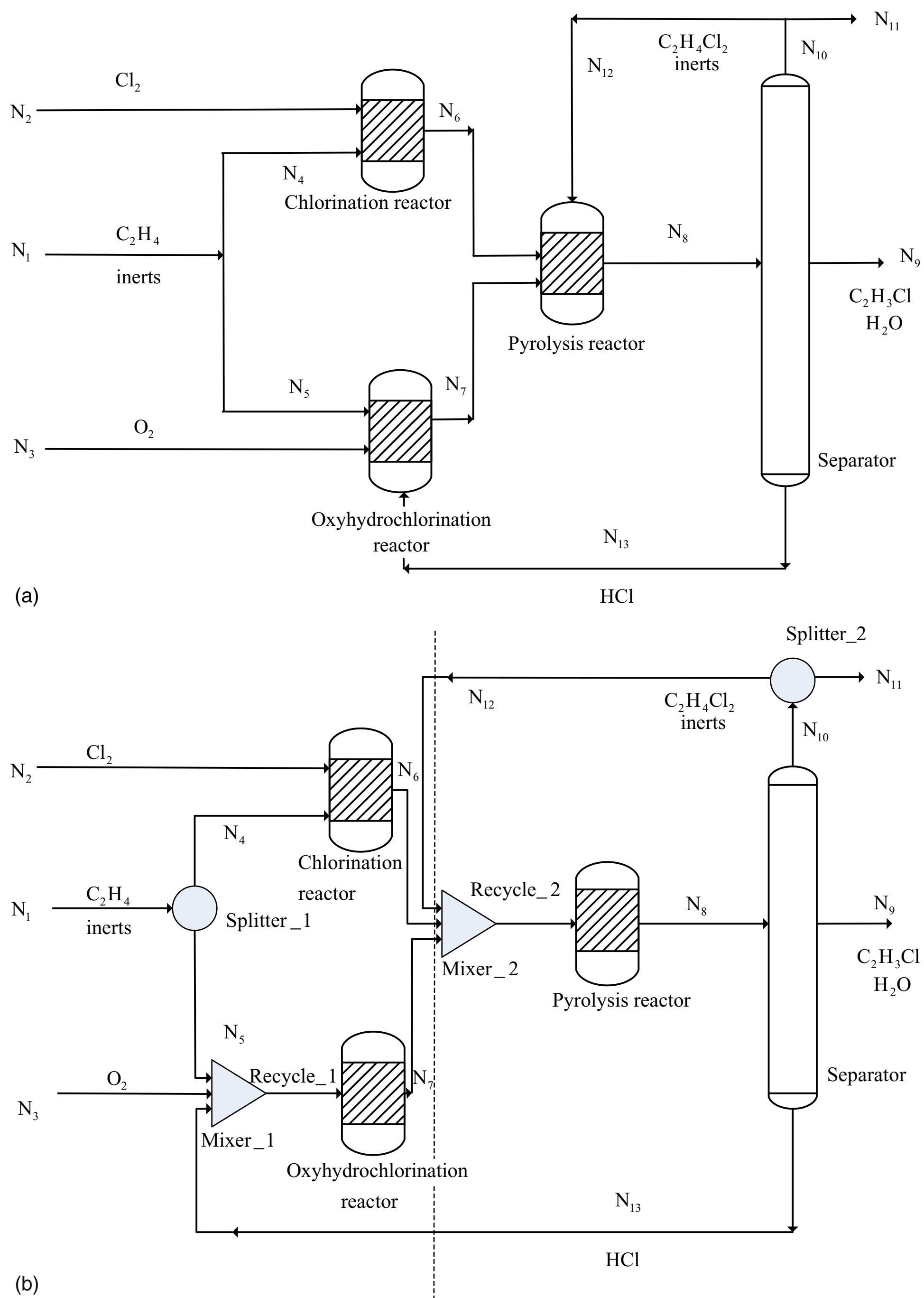
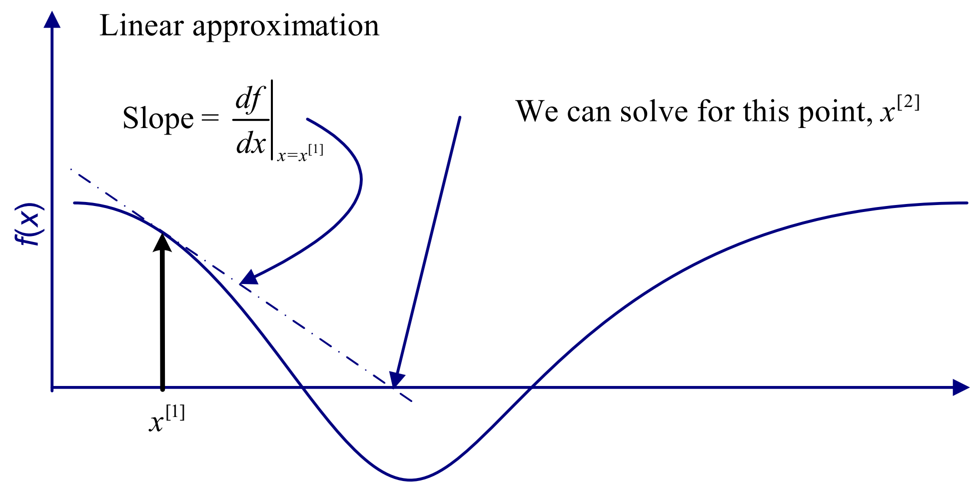

Recycle 7
N_2
H_2
CH_4
Ar
NH_3
Splitter
Purge 6
N_2
H_2
CH_4
Ar
NH_3
Vapor out 5
N_2
H_2
CH_4
Ar
NH_3
Reactor in 2
N_2
H_2
CH_4
Ar
NH_3
Reactor out 3
N_2
H_2
CH_4
Ar
NH_3
Separator 1
Feed 1
N_2 24.5%
H_2 74.0%
CH_4 1.2%
Ar 0.3%
Reactor
$N_2 + 3H_2 = 2NH_3$
Product 4
NH_3

FIGURE p3.15



N11
N12
C2H4Cl2
inerts
N10
Cl2
N2
N6
N4
Chlorination reactor
C2H4
N1
inerts
N8
N9
C2H3Cl
H2O
Pyrolysis reactor
N5
N7
O2
N3
Separator
Oxyhydrochlorination
reactor
N13
(a)
HCl
Splitter_2
N11
N12
C2H4Cl2
inerts
N10
Cl2
N2
N6
N4
Chlorination
reactor
Recycle_2
N1
C2H4
inerts
Splitter_1
N8
N9
Mixer_2
C2H3Cl
H2O
Pyrolysis reactor
N5
O2
Recycle_1
N7
Separator
N3
Mixer_1
Oxyhydrochlorination
reactor
N13
HCl
(b)

FIGURE p3.20



Linear approximation
Slope = $\left.\frac{df}{dx}\right|_{x=x^{[1]}}$
We can solve for this point, $x^{[2]}$
$f(x)$
$x^{[1]}$

FIGURE p3.4