Excel Bootcamps 1, 2, 3 and 4

- ✓ 1: Getting up to speed with Excel
- ✓ 2: Introducing VBA
- 3: Learning to use Excel to solve typical problem scenarios
- 4: Detailed modeling of packed-bed and plug-flow reactors

Bootcamp 3 Outline

- Solving Single Algebraic Equations 2
- Solving Sets of Linear Algebraic Equations 10
- Solving Sets of Nonlinear Algebraic Equations 16
- Ordinary Differential Equation Models 25
- Solving Multiple Differential Equations 30
- Optimization 48
- Curve-Fitting 54

Solving Single Algebraic Equations

Excel Tools

Goal Seek

Solver

Numerical Methods

- Bracketing methods
  - Bisection
  - False position
- Open methods
  - Newton-Raphson
  - Modified secant
- Hybrid
  - Brent's method

$$f(x) = 0$$

- Circular scenario
  - Substitution
  - Wegstein method

$$x = g(x)$$

For details on numerical methods, see Chapra and Clough, **Applied Numerical Methods with Python for Engineers and Scientists**, McGraw-Hill, 2022, or Chapra, **Applied Numerical Methods with MATLAB for Engineers and Scientists**, 5th Edition, McGraw-Hill, 2023.

Solving Single Algebraic Equations

Water-gas shift equilibrium $$CO + H_2O \Leftrightarrow H_2 + CO_2$$

$$\frac{[H_2]\cdot[CO_2]}{[H_2O]\cdot[CO]} = K_{eq}(T) \qquad ln\left[K_{eq}(T)\right] = -3.112 + \frac{3317}{T} \qquad T(K)$$

Feed

	kmol/hr
H_2	450
CO	500
CO_2	50
H_2O	1150

Product

T = 1200 °C
P = 1 atm

$$f(x) = \frac{\left[Feed_{H_2} + x\right]\cdot\left[Feed_{CO_2} + x\right]}{\left[Feed_{H_2O} - x\right]\cdot\left[Feed_{CO} - x\right]} - K_{eq}(T) = 0$$

where x is the shift to equilibrium in kmol/hr.

Solve for x and the reactor product flow rates for the given temperature.

Solving Single Algebraic Equations

Water-gas shift equilibrium

Water-gas shift equilibrium						
Starter					Feed Rates	Product Rates
	T	1200	degC		kgmol/h	
	TK	1572.2	K	H_2	450	550.0
	Keq	0.380		CO_2	50	150.0
				H_2O	1150	1050.0
	x	100.0		CO	500	400.0
	Eqn error	-0.1835				

$$f(x) = \frac{\left[Feed_{H_2} + x\right] \cdot \left[Feed_{CO_2} + x\right]}{\left[Feed_{H_2O} - x\right] \cdot \left[Feed_{CO} - x\right]} - K_{eq}(T) = 0$$

water-gas_starter.xlsx

Solving Single Algebraic Equations

Water-gas shift equilibrium

Using Goal Seek

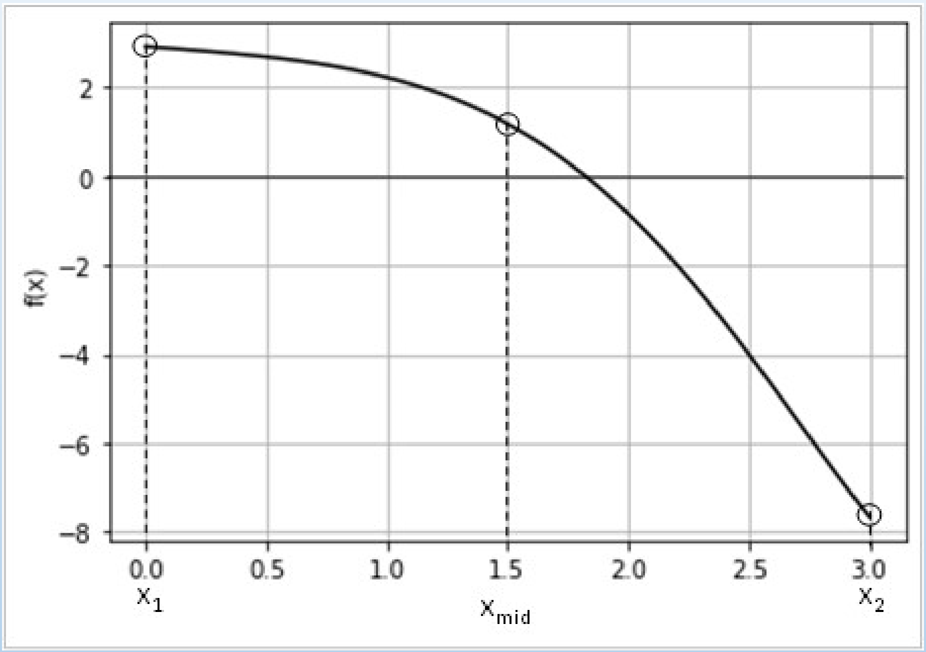
Not convenient for a case study because the solution is not "live" on the spreadsheet. Goal Seek has to be run each time an input changes.

Water-gas shift equilibrium

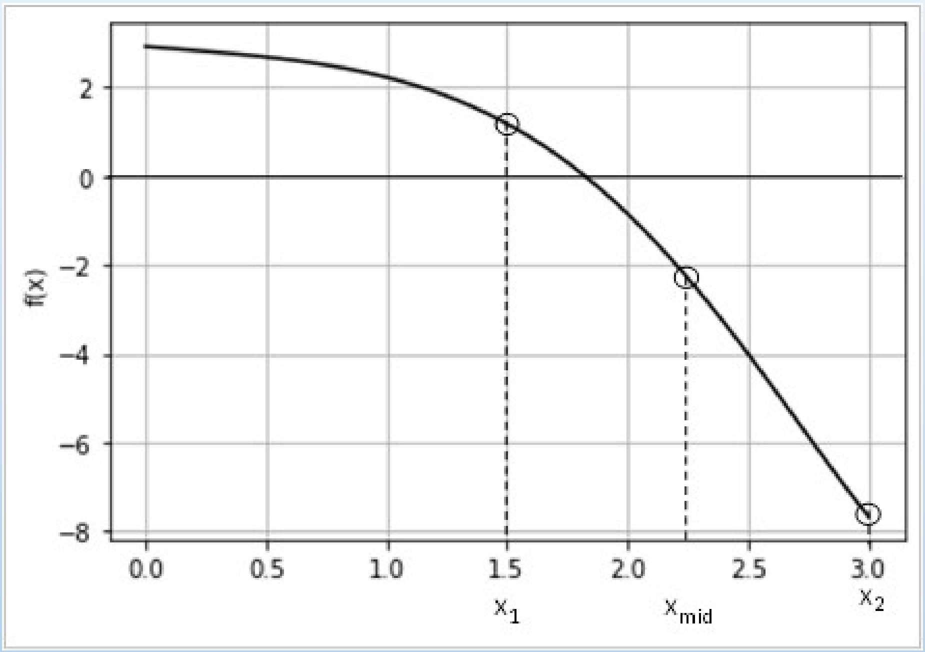
				Feed Rates	Product Rates
T	1200	degC		kgmol/h	
TK	1572.2	K	H_2	450	609.8
Keq	0.380		CO_2	50	209.8
			H_2O	1150	990.2
x	159.8		CO	500	340.2
Eqn error	-0.0001				

$$f(x) = \frac{\left[Feed_{H_2} + x\right] \cdot \left[Feed_{CO_2} + x\right]}{\left[Feed_{H_2O} - x\right] \cdot \left[Feed_{CO} - x\right]} - K_{eq}(T) = 0$$

Solving Single Algebraic Equations - Bisection



First iteration



Second iteration

Solving Single Algebraic Equations

Water-gas shift equilibrium

using Bisection on the sheet

	B	C	D	E	F	G	H
10	Iteration	x1	f(x1)	x2	f(x2)	xm	f(xm)
11	1	0	-0.34079	450	12.47723	225	0.34981
12	2	0	-0.34079	225	0.349813	112.5	-0.15256
13	3	225	0.34981	112.5	-0.15256	168.75	0.0365
14	4	112.5	-0.15256	168.75	0.036499	140.625	-0.06954
15	5	168.75	0.0365	140.625	-0.06954	154.688	-0.01979
16	6	168.75	0.0365	154.6875	-0.01979	161.719	0.00748
26	16	[illegible]	-4.1E-05	[illegible]	1.31E-05	[illegible]	-1.4E-05
27	17	159.837341	1.3E-05	159.83047	-1.4E-05	159.834	-4.5E-07
28	18	159.837341	1.3E-05	159.83391	-4.5E-07	159.836	6.3E-06
29	19	159.833908	-4.5E-07	159.83562	6.3E-06	159.835	2.9E-06
30	20	159.833908	-4.5E-07	159.83477	2.92E-06	159.834	1.2E-06
31	21	159.833908	-4.5E-07	159.83434	1.23E-06	159.834	3.9E-07

water-gas_bisectiontable.xlsx

Solving Single Algebraic Equations

Water-gas shift equilibrium

using Bisection on the sheet

Water-gas shift equilibrium						
					Feed Rates	Product Rates
On-sheet Bisection Strategy						
	T	1200	degC		kgmol/h	
	TK	1572.2	K	H_2	450	609.8
	Keq	0.380		CO_2	50	209.8
				H_2O	1150	990.2
	x	159.8		CO	500	340.2
	Eqn error	0.0000				

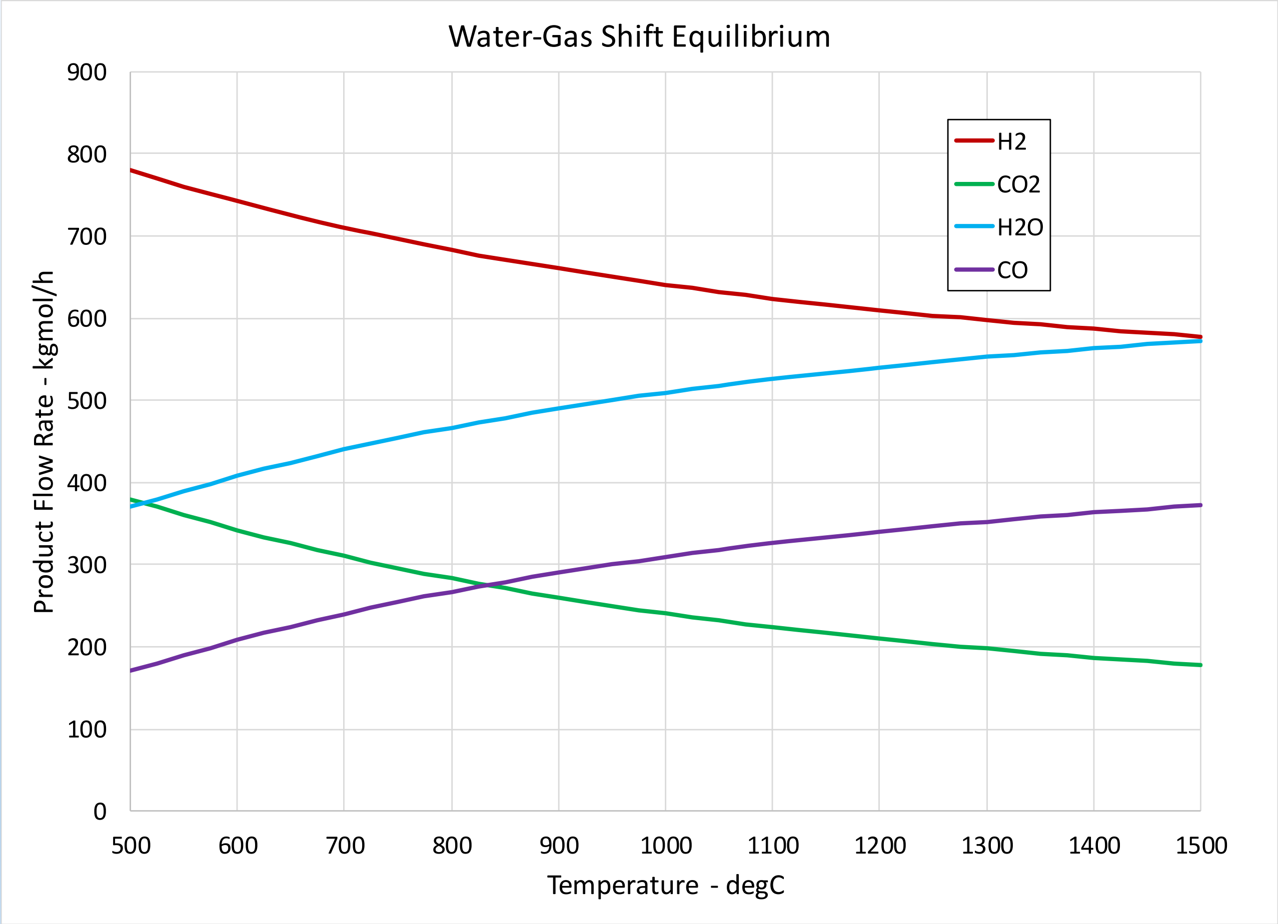
Live solution amenable to a Data Table case study.

Solving Single Algebraic Equations

Water-gas Shift Equilibrium

Data Table case study

	x				
Temperature (degC)	159.8341	H_2	CO_2	H_2O	CO
500	329.9	779.9	379.9	370.1	170.1
525	320.0	770.0	370.0	380.0	180.0
550	310.4	760.4	360.4	389.6	189.6
575	301.2	751.2	351.2	398.8	198.8
600	292.3	742.3	342.3	407.7	207.7
625	283.8	733.8	333.8	416.2	216.2
650	275.6	725.6	325.6	424.4	224.4
675	267.8	717.8	317.8	432.2	232.2
•	•		•		
1350	142.1	592.1	192.1	557.9	357.9
1375	139.4	589.4	189.4	560.6	360.6
1400	136.9	586.9	186.9	563.1	363.1
1425	134.5	584.5	184.5	565.5	365.5
1450	132.1	582.1	182.1	567.9	367.9
1475	129.8	579.8	179.8	570.2	370.2
1500	127.6	577.6	177.6	572.4	372.4



Solving Single Algebraic Equations

Water-gas Shift Equilibrium

Using VBA bisection user-defined function

```
Option Explicit
Function Bisect(x1, x2, T)
Dim xm, Count As Integer
For Count = 1 To 20
    xm = (x1 + x2) / 2
    If f(xm, T) * f(x1, T) > 0 Then
        x1 = xm
    Else
        x2 = xm
    End If
Next Count
Bisect = xm
End Function
```

	A	B	C	D	E	F	G	H
1	**Water-gas shift equilibrium**							
2	Bisection UDF Strategy					Feed Rates	Product Rates	
3		T	1200	degC		kgmol/h		
4		TK	1572.2	K	H_2	450	620.3	
5		Keq	0.300		CO_2	50	220.3	
6					H_2O	1150	979.7	
7		x	170.3		CO	500	329.7	
8		Eqn error	0.0000					
9								
10								
11								

$$f(x) = \frac{\left[Feed_{H_2} + x\right]\cdot\left[Feed_{CO_2} + x\right]}{\left[Feed_{H_2O} - x\right]\cdot\left[Feed_{CO} - x\right]} - K_{eq}(T) = 0$$

```
Function f(x, T)
Dim TK, Keq, FdH2, FdCO2, FdH2O, FdCO
TK = T + 273.15
Keq = Exp(-3.112 + 3317 / TK)
FdH2 = Range("FeedH2")
FdCO2 = Range("FeedCO2")
FdH2O = Range("FeedH2O")
FdCO = Range("FeedCO")
f = (FdH2 + x) * (FdCO2 + x) / (FdH2O - x) / (FdCO - x) - Keq
End Function
```

Solution is still live, so is amenable to case study. Much more compact on the spreadsheet.

water-gas_bisectionUDF.xlsx

Solving Single Algebraic Equations

Similar methods can be employed using the same on-sheet and VBA UDF live solution strategies:

- Root-finding
  - false position
  - Newton's method
  - secant method
  - Wegstein method [$x = g(x)$]
- Extremum-finding
  - binary search
  - Golden Section search
  - gradient method
  - hybrid methods, e.g., Levenberg-Marquardt

Solving Sets of Linear Algebraic Equations

n equations in *n* unknowns

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

Solving the equations:

1. matrix algebra and computations

$$\mathbf{A}^{-1} \cdot \mathbf{A} \cdot \mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b}$$

$$\mathbf{I} \cdot \mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b}$$

compute the inverse of **A** and multiply it by **b**

2. more efficient numerical method

- Gaussian elimination with enhancements
- LU decomposition

Solving Sets of Linear Algebraic Equations

Example

$$3x + 2y - z = 10$$
$$-x + 3y + 2z = 5$$
$$x - y - z = -1$$

$$\begin{bmatrix} 3 & 2 & -1 \\ -1 & 3 & 2 \\ 1 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \\ -1 \end{bmatrix}$$

	A	B	C	D	E
1	3	2	-1		10
2	-1	3	2		5
3	1	-1	-1		-1
4			A		b
5	-2				
6	5				
7	-6				
8	x				

	A
5	=MMULT(MINVERSE(A),b)
6	
7	
8	x

LinearEquationsStarter.xlsx

Solving Sets of Linear Algebraic Equations

Example problem: Six-stage absorber column

Equilibrium relationship on tray *i*

$$y_i = ax_i + b$$

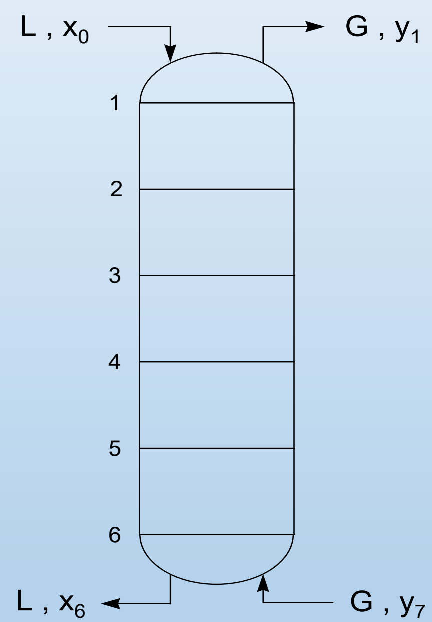
x_0 , y_7 , L and G specified

Component material balance on tray *i*

$$L \cdot x_{i-1} + G \cdot y_{i+1} = L \cdot x_i + G \cdot y_i$$

Incorporate equilibrium relationship

$$L \cdot x_{i-1} - \left(L + G \cdot a\right) \cdot x_i + G \cdot a \cdot x_{i+1} = 0$$

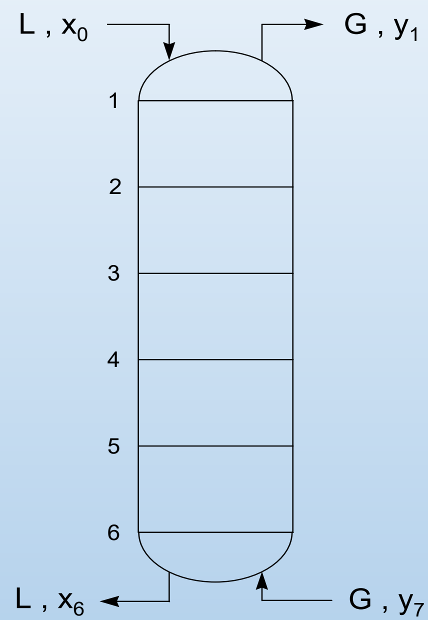


Solving Sets of Linear Algebraic Equations

Example problem: Six-stage absorber column

Write component material balances for each tray and rearrange with unknowns on the left and knowns on the right.

$$-(L+Ga)x_1 + Gax_2 = -Lx_0$$
$$Lx_1 - (L+Ga)x_2 + Gax_3 = 0$$
$$Lx_2 - (L+Ga)x_3 + Gax_4 = 0$$
$$Lx_3 - (L+Ga)x_4 + Gax_5 = 0$$
$$Lx_4 - (L+Ga)x_5 + Gax_6 = 0$$
$$Lx_5 - (L+Ga)x_6 = -G(y_7 - b)$$



This represents a set of six linear equations in the six unknown mass fractions.

Basic data: equilibrium model: a = 0.7, b = 0

Operating conditions: L = 20 mol/s, G = 12 mol/s

Inlet gas mole fraction: y_7 = 0.1

Inlet liquid mole fraction: x_0 = 0

Solving Sets of Linear Algebraic Equations

Example problem: Six-stage absorber column

Spreadsheet solution

Set up basic data and operating conditions:

	B	C	D	E	F
2	L	20	mol/s	yin	0.1
3	G	12	mol/s	a	0.7

Transfer the labels to name the cells to the right.

Set up a matrix for the coefficients of the linear equations:

	B	C	D	E	F	G	H
5	A_coef	Liquid composition coefficient					
6	Stage No.	x1	x2	x3	x4	x5	x6
7	1	-28.4	8.4	0	0	0	0
8	2	20	-28.4	8.4	0	0	0
9	3	0	20	-28.4	8.4	0	0
10	4	0	0	20	-28.4	8.4	0
11	5	0	0	0	20	-28.4	8.4
12	6	0	0	0	0	20	-28.4

	B	C	D
6	Stage No.	x1	=G*a
7	=-(L+G*a)		8.4
8	=L	20	-28.4

Name the matrix **A_coef** .

Absorber.xlsx

Solving Sets of Linear Algebraic Equations

Example problem: Six-stage absorber column

Spreadsheet solution

Set up a vector **b** for the constants

	J	K
6	Stage No.	Constant
7	1	0
8	2	0
9	3	0
10	4	0
11	5	0
12	6	-1.2
13		b

=-G*yin

Solve for **x** using **A_coef^{-1}*b**

Liquid Mole Fractions
=MMULT(MINVERSE(A_coef),b)
0.00154
0.00413
0.0103
0.0249
0.0598

	M	N	O
6	Stage No.	Liquid Mole Fractions	
7	1	0.000456	
8	2	0.00154	
9	3	0.00413	
10	4	0.0103	
11	5	0.0249	
12	6	0.0598	

Since this is an array formula, remember to select all cells and *Ctrl-Shift-Enter*.

The **y** values could be computed using the equilibrium relationship.

This is a live solution and amenable to case studies using the Data Table.

Solving Sets of Nonlinear Algebraic Equations

$$f_1\left(x_1, x_2, \ldots, x_n\right) = 0$$
$$f_2\left(x_1, x_2, \ldots, x_n\right) = 0$$
$$\vdots$$
$$f_n\left(x_1, x_2, \ldots, x_n\right) = 0$$

or $\mathbf{f}\left(\mathbf{x}\right) = \mathbf{0}$

Common solution technique: Newton's Method

Start with an initial estimate of the solution: $\mathbf{x}^0$

Iterate with $\mathbf{x}^{i+1} = \mathbf{x}^i - \mathbf{J}^{-1}\left(\mathbf{x}^i\right) \cdot \mathbf{f}\left(\mathbf{x}^i\right)$ until a convergence criterion is met.

Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

or, where analytical derivatives are difficult:

$$\frac{\partial f_1}{\partial x_1}\left(\mathbf{x}^i\right) \cong \frac{f_1\left(x_1^i + \delta, x_2^i, \ldots, x_n^i\right) - f_1\left(x_1^i - \delta, x_2^i, \ldots, x_n^i\right)}{2 \cdot \delta}$$

and so forth.

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

$$P \cdot V = \frac{m}{MW} \cdot R \cdot \left(T + 273.15\right) \qquad log_{10} P = A - \frac{B}{T + C}$$

ideal gas law — Antoine equation

P : absolute pressure, Pa

V : vapor volume, m^3

m : mass of vapor, kg

MW: H_2O molecular weight, ≅ 18.02 kg/kgmol

R : gas law constant, 8314 (Pa•m^3)/(kgmol•K)

T : temperature, °C

A, B, C : Antoine constants for H_2O

$A = 11.21 \quad B = 2354.7 \quad C = 280.71$

Operating conditions: $m = 3.755\ kg \qquad V = 3.142\ m^3$

Solve for P and T .

SteamEquilibriumStarter.xlsx

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Formulating the problem for solution

$$f_1(T,P) = P \cdot V - \frac{m}{MW} \cdot R \cdot (T + 273.15)$$

$$f_2(T,P) = log_{10} P - A + \frac{B}{T+C}$$

$$\mathbf{J}\left(\begin{bmatrix} P \\ T \end{bmatrix}\right) = \begin{bmatrix} V & -\frac{m}{MW} \cdot R \\ \frac{1}{ln(10) \cdot P} & -\frac{B}{(C+T)^2} \end{bmatrix}$$

analytical Jacobian practical in this case

A possible issue here is the comparative scaling of the two equations. Typical values for the PV term could be of magnitude 10^6; whereas, terms in the second equation are closer to unity. A practical approach to this is to scale the first equation by dividing it by, e.g., 100,000.

$$f_1(T,P) = \left(P \cdot V - \frac{m}{MW} \cdot R \cdot (T + 273.15)\right) / 100000$$

$$f_2(T,P) = log_{10} P - A + \frac{B}{T+C}$$

$$\mathbf{J}\left(\begin{bmatrix} P \\ T \end{bmatrix}\right) = \begin{bmatrix} V / 1e5 & -\frac{m}{MW} \cdot R / 1e5 \\ \frac{1}{ln(10) \cdot P} & -\frac{B}{(C+T)^2} \end{bmatrix}$$

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Solution with Excel's Solver

	A	B	C	D	E
1	Rgas	8314	Pa*m^3/(kgmol*K)		
2	MW	18.02	kg/kgmol		
3				A	11.21
4	V	3.142	m^3	B	2354.7
5	m	3.755	kg/kgmol	Cc	280.7

Set up basic data and operating conditions. Name cells according to labels.

	A	B
10	P	200000
11	T	110.0

Create cells for initial estimates.

SteamEquilibriumStarter.xlsx

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Solution with Excel's Solver

Add function evaluations and sum of squares of equation errors

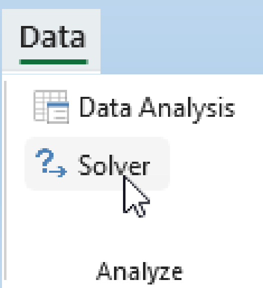
	A	B	C
10	P	200000	-0.3540
11	T	110.0	0.1179
12			0.1392

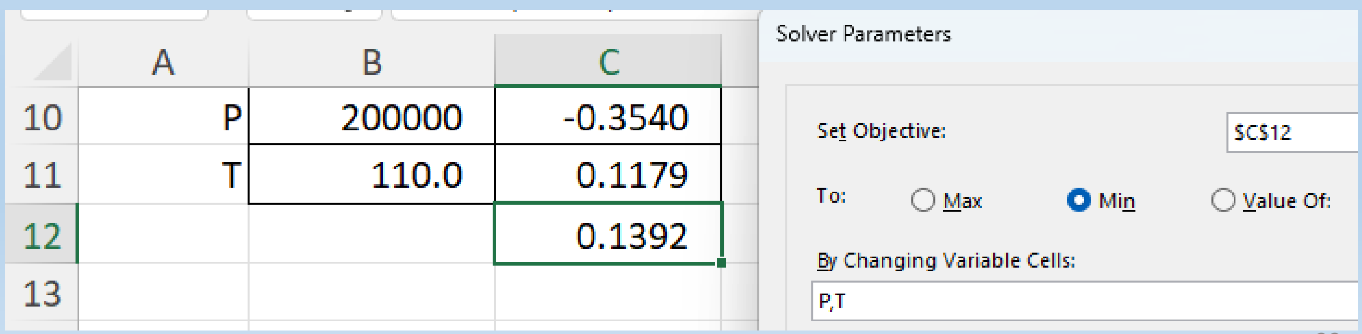
=(P*V-m/MW*Rgas*(T+273.15))/100000

=LOG10(P)-A+B/(T+Cc)

=SUMSQ(C10:C11)

Set up the Solver





Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Solution with Excel's Solver

Select Solve and accept the result

Solution

	A	B	C
10	P	216789	-0.0011
11	T	120.1	0.0013
12			0.0000

SteamEquilibriumSolverFinish.xlsx

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Solution with Excel - live Newton's Method

SteamEquilibriumStarter.xlsx

	A	B	C
9		Initial Estimates	Current Values
10	P	200000	=B10
11	T	110.0	

Enter pointer formulas to transfer Initial Estimates to Current Values.

Do not name the Initial Estimates cells this time.

	A	B	C	D
9		Initial Estimates	Current Values	Function Evaluation
10	P	200000	200000	-3.54E-01
11	T	110.0	110	1.18E-01

=(C10*V-m/MW*Rgas*(C11+273.15))/100000

=LOG10(C10)-A+B/(C11+Cc)

Evaluate functions given Current Values.

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Solution with Excel

Jacobian matrix evaluated in terms of Current Values

	A	B	C
13		Jacobian Matrix	
14	J	3.142E-05	-1.732E-02
15		2.171E-06	-1.543E-02

=V/100000

=-m/MW*Rgas/100000

=1/LN(10)/C10

=-B/(Cc+C11)^2

named **J**

	A	B	C
17		Inverse of Jacobian Matrix	
18	Jinv	3.451E+04	-3.875E+04
19		4.857E+00	-7.028E+01

=MINVERSE(J)

named **Jinv**

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Compute the new estimate using Newton's formula

	A	B	C	D	E
9		Initial Estimates	Current Values	Function Evaluation	New Estimate
10	P	200000	200000	-3.54E-01	216782
11	T	110.0	110	1.18E-01	120.0

=C10:C11-MMULT(Jinv,D10:D11)

Set up the iterative solver for a single calculation

File → Options → Formulas →

☑ Enable iterative calculation

Maximum Iterations: 1

Maximum Change: 0.00001

Map the New Estimate back to the Current Value for 1 iteration

Ctrl-Shift-Enter

	C	D	E
9	Current Values	Function Evaluation	New Estimate
10	=E10:E11	-3.54E-01	216782
11	110	1.18E-01	120.0

	C	D	E
9	Current Values	Function Evaluation	New Estimate
10	216782	-3.54E-01	233564
11	120	1.18E-01	130.0

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Press the Calc key (F9) a number of times to see whether the method converges.

	Initial Estimates	Current Values	Function Evaluation	New Estimate
P	200000	212874	-2.33E-15	212456
T	110.0	113	1.00E-01	112.2

In this case, it doesn't converge to the solution. This is typical of Newton's method which tends either to converge rapidly or not be stable.

To promote convergence, a common technique is to incorporate a decelerator shown as

$$\mathbf{x}^{i+1} = \mathbf{x}^{i} - decel \cdot \mathbf{J}^{-1}\left(\mathbf{x}^{i}\right) \cdot \mathbf{f}\left(\mathbf{x}^{i}\right) \qquad 0 < decel \le 1$$

	Initial Estimates	Current Values	Function Evaluation	New Estimate
P	200000	216880	0.00E+00	216880
T	110.0	120	0.00E+00	120.2
			decel	0.5

```
=C10:C11-decel*MMULT(Jinv,D10:D11)
```

110.0	120	0.00E+00	120.2

For a decelerator value of 0.5, the calculation converges to the solution.

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Set the iterative solver to 1000 Maximum Iterations, and the solution is always displayed.

☑ Enable iterative calculation	
Maximum Iterations:	1,000
Maximum Change:	0.00001

Add a provision to reset the calculation to the initial estimates.

	A	B	C	D	E
10	=if(Reset,B10:B11,E10:E11)			0.00E+00	216880
11	T	110.0	120.2	-7.59E-11	120.2

Reset
TRUE

	Initial Estimates	Current Values
P	200000	200000
T	110.0	110.0

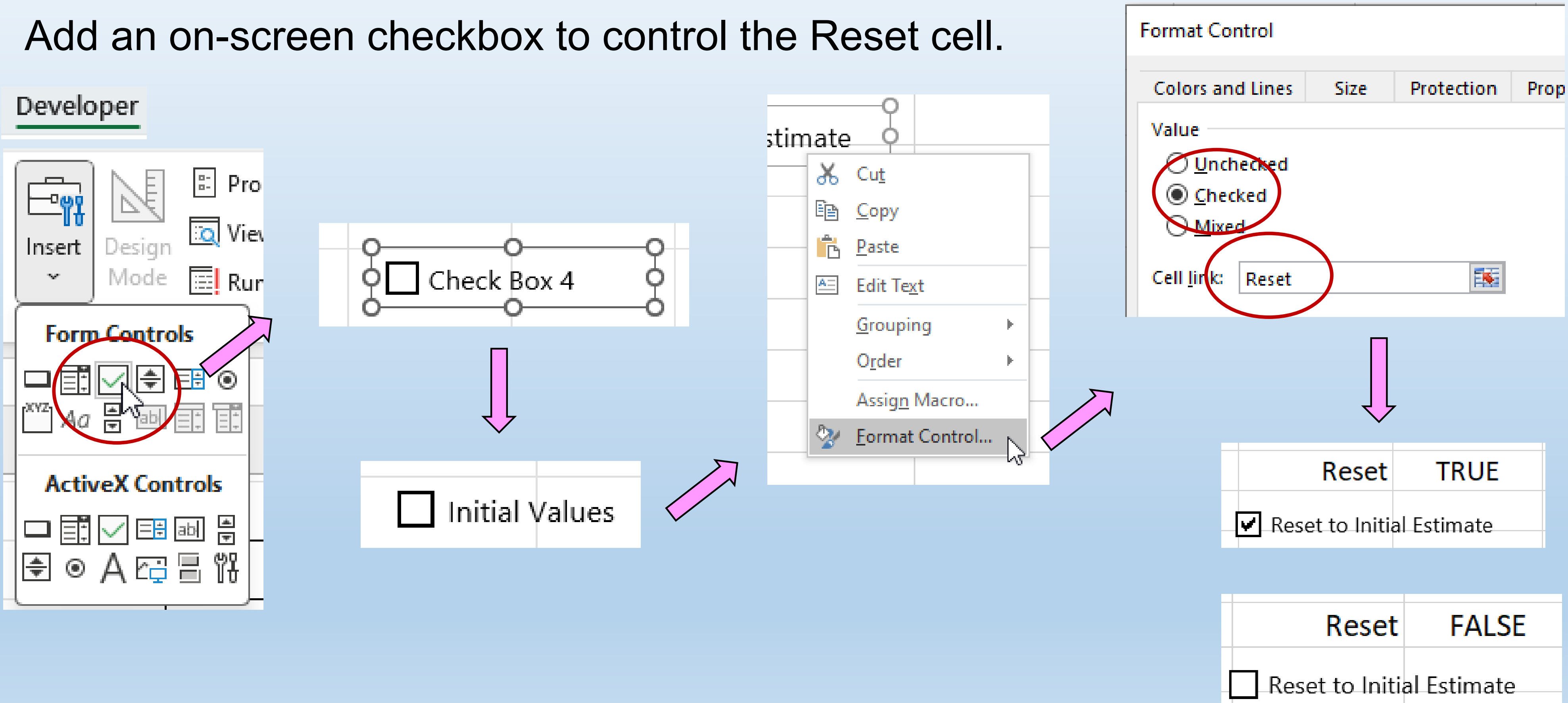
Reset
FALSE

Current Values	Function Evaluation
216880	1.16E-15
120.2	4.17E-10

Solving Sets of Nonlinear Algebraic Equations

Example problem: steam/water equilibrium

Add an on-screen checkbox to control the Reset cell.



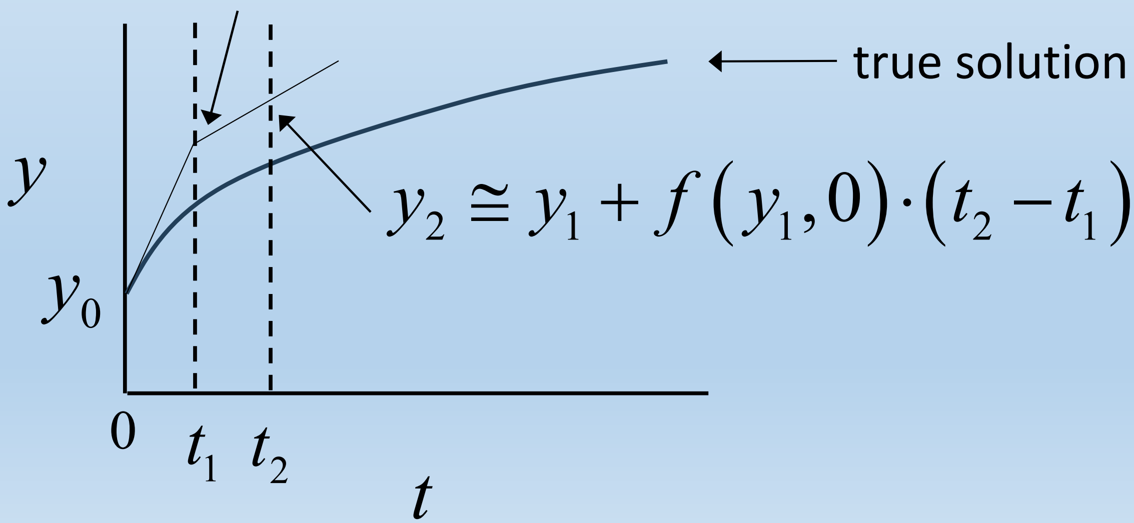
Ordinary Differential Equation Models

Using the Euler Method to Solve Differential Equations

Single equation with initial value

$$\frac{dy}{dt} = f\left(y, t\right) \qquad y\left(0\right) = 0$$

$$y_1 \cong y_0 + f\left(y_0, 0\right) \cdot \left(t_1 - 0\right)$$



Approximations include errors that accumulate as the solution proceeds.
Errors are reduced as step size

$$h_i = t_{i+1} - t_i$$

decreases. Very small step sizes increase computational effort and may lead to round off errors.
Other more complicated schemes, such as the Runge-Kutta and predictor-corrector methods control error better but are more difficult to implement in Excel/VBA.

Ordinary Differential Equation Models

Single Equation Example – Isothermal Batch Reactor $A + B \overset{k}{\Rightarrow} C$

Rate of disappearance of A: $$\frac{dC_A}{dt} = -k \cdot C_A \cdot C_B$$

Initial conditions: $C_A(0) = C_{A0} \quad C_B(0) = C_{B0} \quad C_C(0) = C_{C0}$

Basic data: $k = 14.7 \, \frac{1}{mol/L} \cdot \frac{1}{min}$

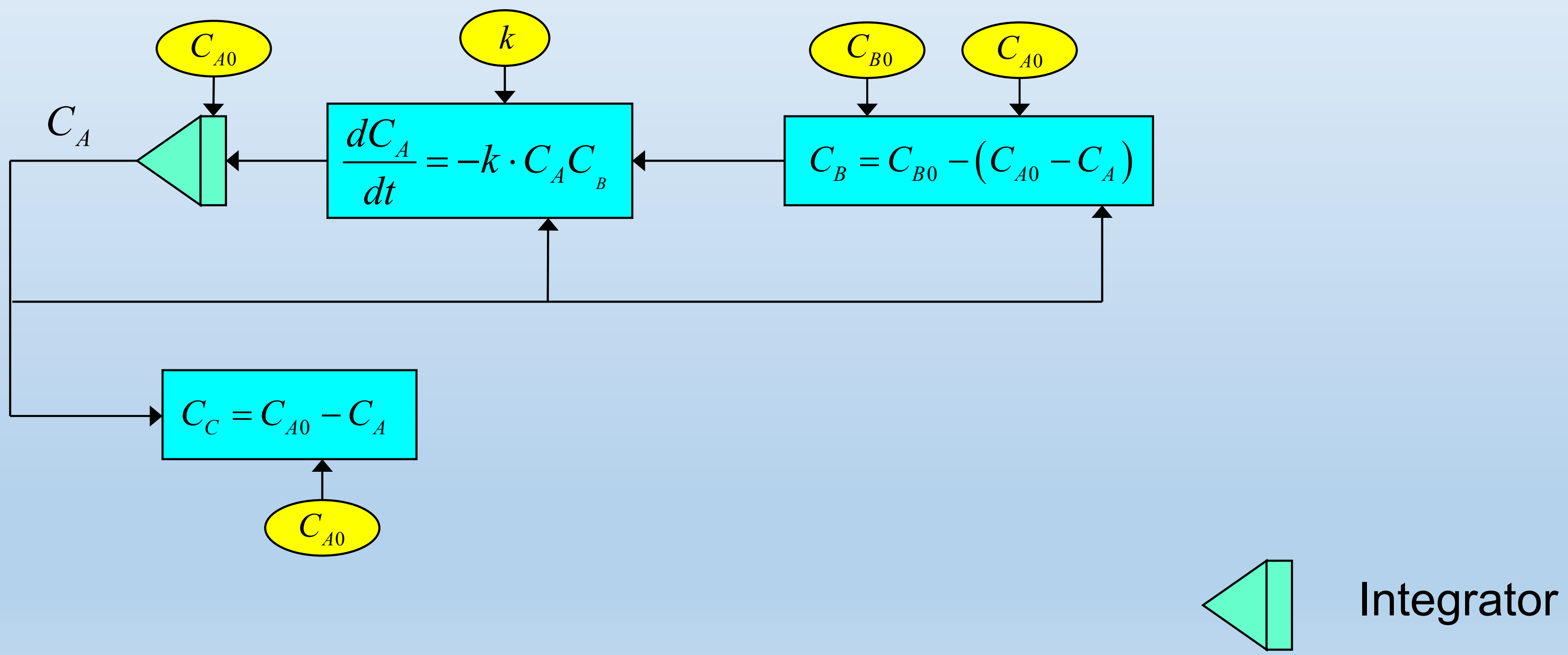
Initial conditions: $C_{A0} = 0.0209 \, \frac{mol}{L} \quad C_{B0} = C_{A0}/3 \quad C_{C0} = 0$

Stoichiometric relationships:
$$C_B(t) = C_{B0} - \left(C_{A0} - C_A(t)\right)$$
$$C_C(t) = C_{C0} + \left(C_{A0} - C_A(t)\right)$$

Ordinary Differential Equation Models

Single Equation Example – Isothermal Batch Reactor

Information Flow Diagram



Ordinary Differential Equation Models

Single Equation Example – Isothermal Batch Reactor

Spreadsheet solution using the Euler method

$$C_A(t_i+\Delta t)=C_A(t_i)+\frac{dC_A}{dt}(t_i)\cdot\Delta t$$

Set up the basic data, initial conditions, and step size

	B	C	D	E
2	k	14.7	1/((mol/L)*min)	
3	CA0	0.0209	mol/L	
4	CB0	0.00697	mol/L	
5	CC0	0	mol/L	
6	dt	0.1	min	

Name cells according to labels to the left.

Set up headings for the solution table

	B	C	D	E	F
8	Time	CA	CB	CC	dCA/dt

BatchReactorSingleEqnStarter.xlsx

Ordinary Differential Equation Models

Single Equation Example – Isothermal Batch Reactor

Spreadsheet solution using the Euler method

Create the *initialization row* of the table

	B	C	D	E	F
8	Time	CA	CB	CC	dCA/dt
9	0	0.0209	0.00697	0	-0.002140

	B	C	D	E	F
8	Time	CA	CB	CC	dCA/dt
9	0	=CA0	=CB0	=CC0	=-k*C9*D9

Enter the first *operational* row

	B	C	D	E	F
8	Time	CA	CB	CC	dCA/dt
9	0	0.02090	0.00697	0	-0.002140
10	0.1	0.02069	0.00675	0.00021	-0.002053

	B	C	D	E	F
8	Time	CA	CB	CC	dCA/dt
9	0	=CA0	=CB0	=CC0	=-k*C9*D9
10	=B9+dt	=C9+F9*dt	=CB0-(CA0-C10)	=CA0-C10	=-k*C10*D10

Ordinary Differential Equation Models

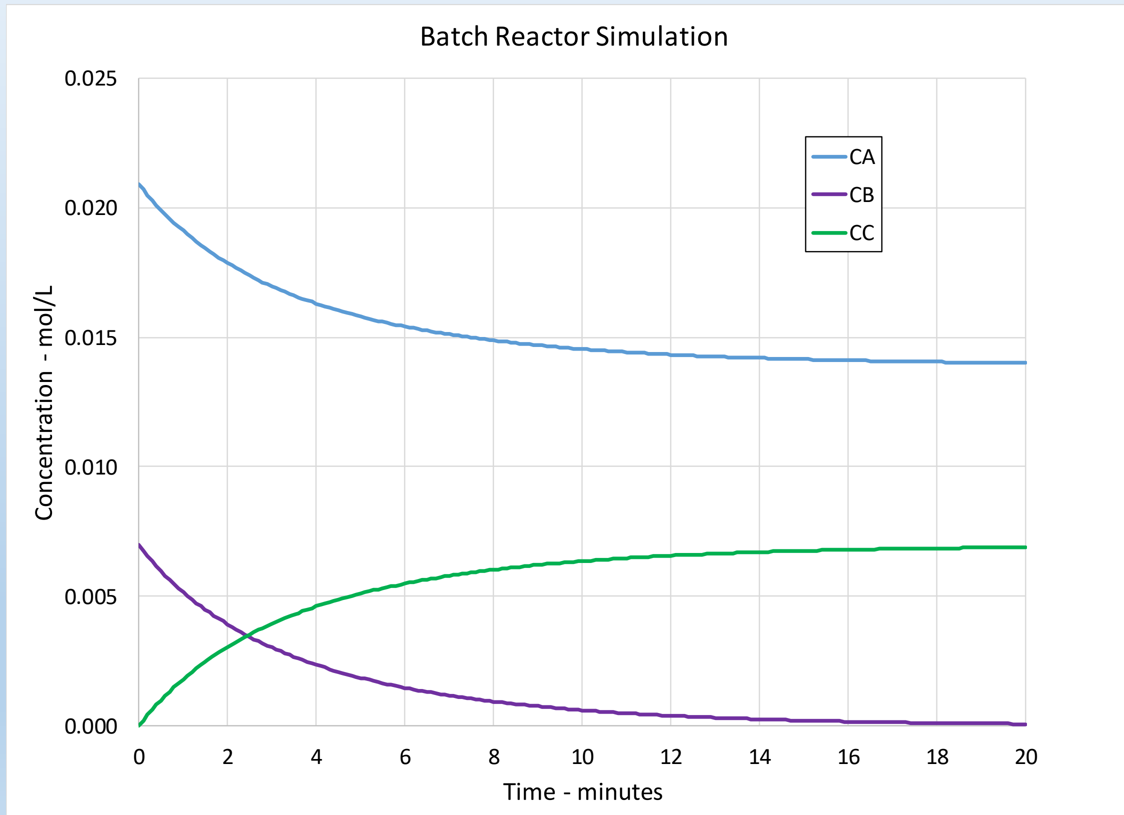
Single Equation Example – Isothermal Batch Reactor

Copy the operational row down to Time = 20

	B	C	D	E	F
8	Time	CA	CB	CC	dCA/dt
9	0	0.02090	0.00697	0	-0.002140
10	0.1	0.02069	0.00675	0.00021	-0.002053
11	0.2	0.02048	0.00655	0.00042	-0.001971
12	0.3	0.02028	0.00635	0.00062	-0.001893
13	0.4	0.02009	0.00616	0.00081	-0.001820
14	0.5	0.01991	0.00598	0.00099	-0.001750
15	0.6	0.01974	0.00580	0.00116	-0.001684
16	0.7	0.01957	0.00564	0.00133	-0.001621

• • • •

	B	C	D	E	F
201	19.2	0.01402	0.00009	0.00688	-0.000018
202	19.3	0.01402	0.00009	0.00688	-0.000018
203	19.4	0.01402	0.00008	0.00688	-0.000017
204	19.5	0.01402	0.00008	0.00688	-0.000017
205	19.6	0.01401	0.00008	0.00689	-0.000017
206	19.7	0.01401	0.00008	0.00689	-0.000016
207	19.8	0.01401	0.00008	0.00689	-0.000016
208	19.9	0.01401	0.00008	0.00689	-0.000016
209	20	0.01401	0.00007	0.00689	-0.000015



EulerMethod.xlsx

Solving Multiple Differential Equations

Multiple Equation Models – Isothermal Batch Reactor

$$A + B \xrightarrow{k_1} C + F$$
$$A + C \xrightarrow{k_2} D + F$$
$$A + D \xrightarrow{k_3} E + F$$

$$\frac{dA}{dt} = -k_1 AB - k_2 AC - k_3 AD \qquad A(0) = 0.0209\ \frac{mol}{L}$$

$$\frac{dB}{dt} = -k_1 AB \qquad B(0) = \frac{A(0)}{3} \qquad k_1 = 14.7\ \frac{1}{mol/L} \cdot \frac{1}{min}$$

$$\frac{dC}{dt} = k_1 AB - k_2 AC \qquad C(0) = 0 \qquad k_2 = 1.53\ \frac{1}{mol/L} \cdot \frac{1}{min}$$

$$\frac{dD}{dt} = k_2 AC - k_3 AD \qquad D(0) = 0 \qquad k_3 = 0.294\ \frac{1}{mol/L} \cdot \frac{1}{min}$$

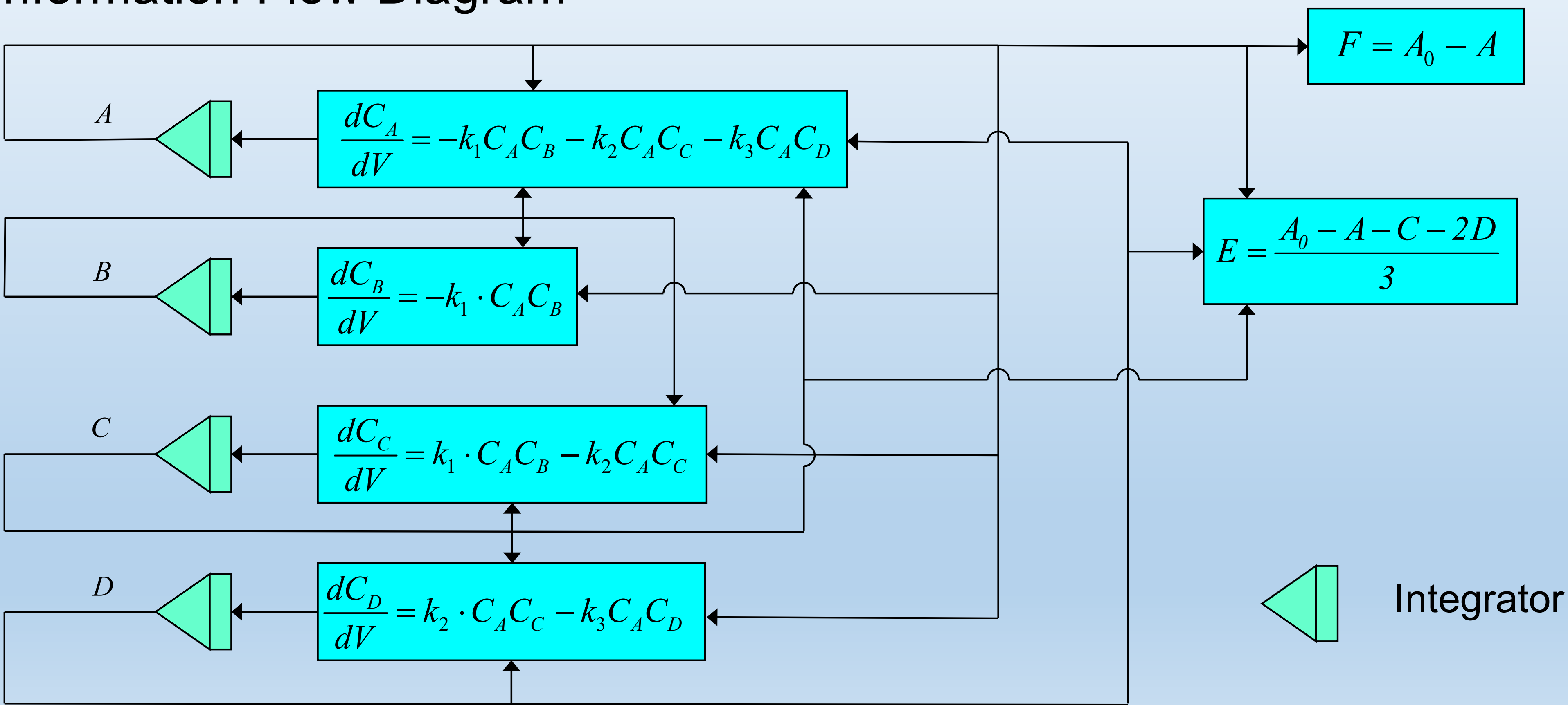
From stoichiometry: $E = \frac{A(0) - A - C - 2D}{3}$ and $F = A(0) - A$

Svirbely, W.J., and J.A. Blauer, *The Kinetics of Three-step Competitive Consecutive Second-order Reactions*, **J. Amer. Chem. Soc.**, **83**, 4115, 1961.
Svirbely, W.J., and J.A. Blauer, *The Kinetics of the Alkaline Hydrolysis of 1,3,5,Tricarbomethoxybenzene*, **J. Amer. Chem. Soc.**, **83**, 4118, 1961.

Solving Multiple Differential Equations

Multiple Equation Models – Isothermal Batch Reactor

Information Flow Diagram



Solving Multiple Differential Equations

Multiple Equation Models – Isothermal Batch Reactor

Spreadsheet solution using the Euler method with variable step size

Set up rate constants and initial conditions

	B	C	D	E	F	G	H
2	k_1	14.7	1/(mol/L)/min		A_0	0.0209	mol/L
3	k_2	1.53			B_0	0.0069667	
4	k_3	0.294			C_0	0	
5					D_0	0	
6							

Create headings and the initialization row

	A	B	C	D	E	F	G	H	I	J	K
7	Time (min)	A	B	C	D	E	F	dA/dt	dB/dt	dC/dt	dD/dt
8	0	0.0209	0.006967	0	0	0	0	-0.002140	-0.002140	0.002140	0

	A	B	C	D	E	F	G	H	I	J	K
7	Time (min)	A	B	C	D	E	F	dA/dt	dB/dt	dC/dt	dD/dt
8	0	=A_0	=B_0	=C_0	=D_0	=(A_0-B8-D8-2*E8)/3	=A_0-B8	=-k_1*B8*C8-k_2*B8*D8-k_3*B8*E8	=-k_1*B8*C8	=k_1*B8*C8-k_2*B8*D8	=k_2*B8*D8-k_3*B8*E8

MultipleReactionsEulerMethodStarter.xlsx

Solving Multiple Differential Equations

Multiple Equation Models – Isothermal Batch Reactor

Spreadsheet solution using the Euler method with variable step size

Create operational row with an initial time step of 0.1 min

	A	B	C	D	E	F	G	H	I	J	K
7	Time (min)	A	B	C	D	E	F	dA/dt	dB/dt	dC/dt	dD/dt
8	0	0.0209	0.006967	0	0	0	0	-0.002140	-0.002140	0.002140	0.000000
9	0.1	0.02069	0.00675	0.00021	0.00000	0.00000	0.00021	-0.002060	-0.002053	0.002047	0.000007

	A	B	C	D	E	F	G
7	Time (min)	A	B	C	D	E	F
8	0	=A_0	=B_0	=C_0	=D_0	=(A_0-B8-D8-2*E8)/3	=A_0-B8
9	=A8+dt_1	=B8+H8*dt_1	=C8+I8*dt_1	=D8+J8*dt_1	=E8+K8*dt_1	=(A_0-B9-D9-2*E9)/3	=A_0-B9

	I	J
2	dt_1	0.1
3	dt_2	1
4	dt_3	5
5	dt_4	10

	H	I	J	K
7	dA/dt	dB/dt	dC/dt	dD/dt
8	=-k_1*B8*C8-k_2*B8*D8-k_3*B8*E8	=-k_1*B8*C8	=k_1*B8*C8-k_2*B8*D8	=k_2*B8*D8-k_3*B8*E8
9	=-k_1*B9*C9-k_2*B9*D9-k_3*B9*E9	=-k_1*B9*C9	=k_1*B9*C9-k_2*B9*D9	=k_2*B9*D9-k_3*B9*E9

Solving Multiple Differential Equations

Multiple Equation Models – Isothermal Batch Reactor

Spreadsheet solution using the Euler method with variable step size

Copy the operational row through the following time step ranges:

	A	B	C	D	E	F	G	H	I	J	K
7	Time (min)	A	B	C	D	E	F	dA/dt	dB/dt	dC/dt	dD/dt
8	0	0.0209	0.006967	0	0	0	0	-0.002140	-0.002140	0.002140	0.000000
9	0.1	0.02069	0.00675	0.00021	0.00000	0.00000	0.00021	-0.002060	-0.002053	0.002047	0.000007
10	0.2	0.02048	0.00655	0.00042	0.00000	0.00000	0.00042	-0.001984	-0.001971	0.001958	0.000013
11	0.3	0.02028	0.00635	0.00061	0.00000	0.00000	0.00062	-0.001912	-0.001893	0.001874	0.000019
12	0.4	0.02009	0.00616	0.00080	0.00000	0.00000	0.00081	-0.001844	-0.001819	0.001795	0.000025
13	0.5	0.01991	0.00598	0.00098	0.00001	0.00000	0.00099	-0.001779	-0.001750	0.001720	0.000030

• • • • • • •

	A	B	C	D	E	F	G	H	I	J	K
534	76	0.00955	0.00000	0.00295	0.00366	0.00036	0.01135	-0.000053	0.000000	-0.000043	0.000033
535	77	0.00955	0.00000	0.00294	0.00366	0.00036	0.01135	-0.000053	0.000000	-0.000043	0.000033
536	78	0.00954	0.00000	0.00294	0.00366	0.00037	0.01136	-0.000053	0.000000	-0.000043	0.000033
537	79	0.00953	0.00000	0.00293	0.00367	0.00037	0.01137	-0.000053	0.000000	-0.000043	0.000033

• • • • • • •

	A	B	C	D	E	F	G	H	I	J	K
612	440	0.00916	0.00000	0.00264	0.00389	0.00044	0.01174	-0.000047	0.000000	-0.000037	0.000026
613	450	0.00915	0.00000	0.00263	0.00389	0.00045	0.01175	-0.000047	0.000000	-0.000037	0.000026
614	460	0.00915	0.00000	0.00263	0.00389	0.00045	0.01175	-0.000047	0.000000	-0.000037	0.000026
615	470	0.00914	0.00000	0.00262	0.00389	0.00045	0.01176	-0.000047	0.000000	-0.000037	0.000026
616	480	0.00914	0.00000	0.00262	0.00390	0.00045	0.01176	-0.000047	0.000000	-0.000037	0.000026
617	490	0.00913	0.00000	0.00262	0.00390	0.00045	0.01177	-0.000047	0.000000	-0.000037	0.000026
618	500	0.00913	0.00000	0.00261	0.00390	0.00045	0.01177	-0.000047	0.000000	-0.000036	0.000026
619	510	0.00913	0.00000	0.00261	0.00390	0.00045	0.01177	-0.000047	0.000000	-0.000036	0.000026

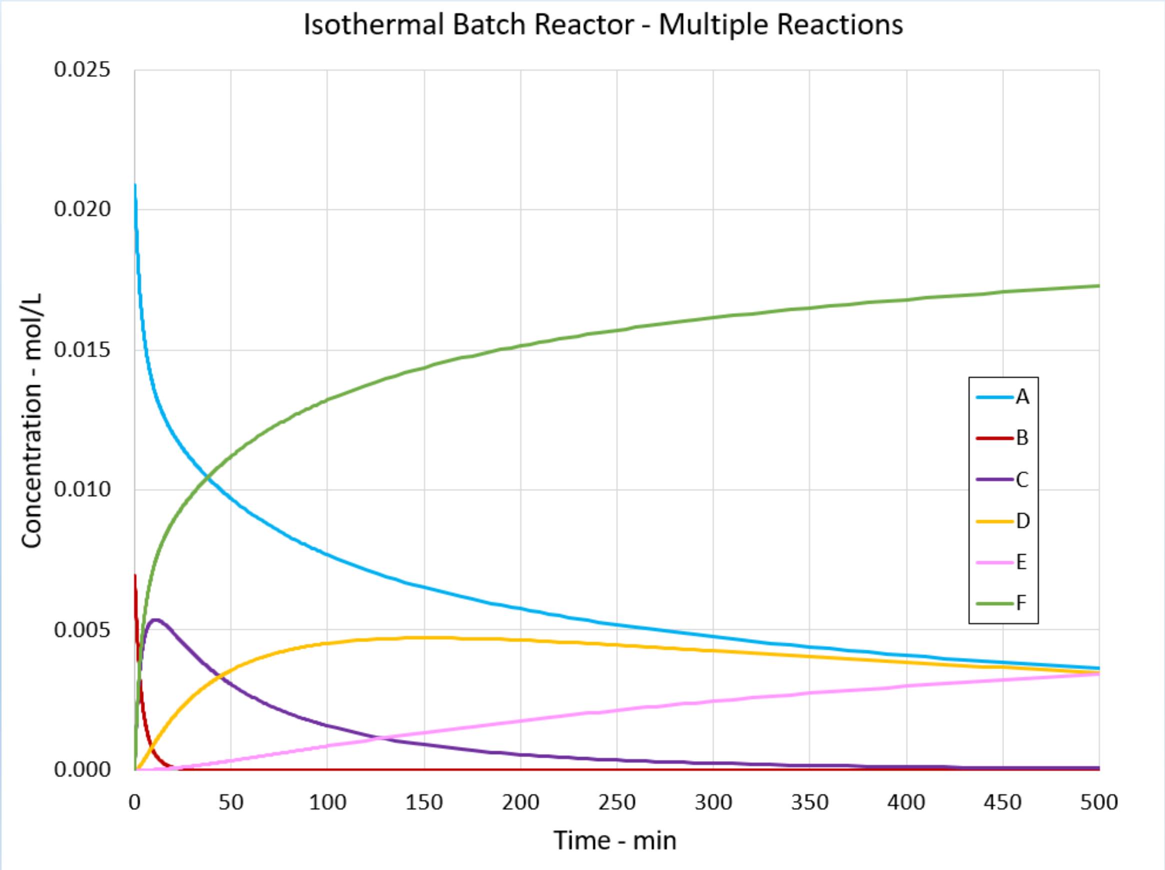
Time Range	Time Step
0-50	0.1
50-100	1
100-300	5
300-500	10

Solving Multiple Differential Equations

Multiple Equation Models – Isothermal Batch Reactor

Spreadsheet solution using the Euler method with variable step size

Create a plot of the solution



MultipleReactionsEulerMethod.xlsx

Solving Multiple Differential Equations

Second-order differential equation with split boundary conditions

$$\frac{d^2y}{dt^2} = \frac{1}{4}\frac{dy}{dt} + y \qquad y(0) = 5 \qquad y(10) = 8 \qquad 0 \le t \le 10$$

Decompose into two first-order ODEs

$$\frac{dy}{dt} = y_1 \qquad y(0) = 5 \qquad y(10) = 8$$

$$\frac{dy_1}{dt} = \frac{1}{4}y_1 + y$$

"Shooting" Strategy

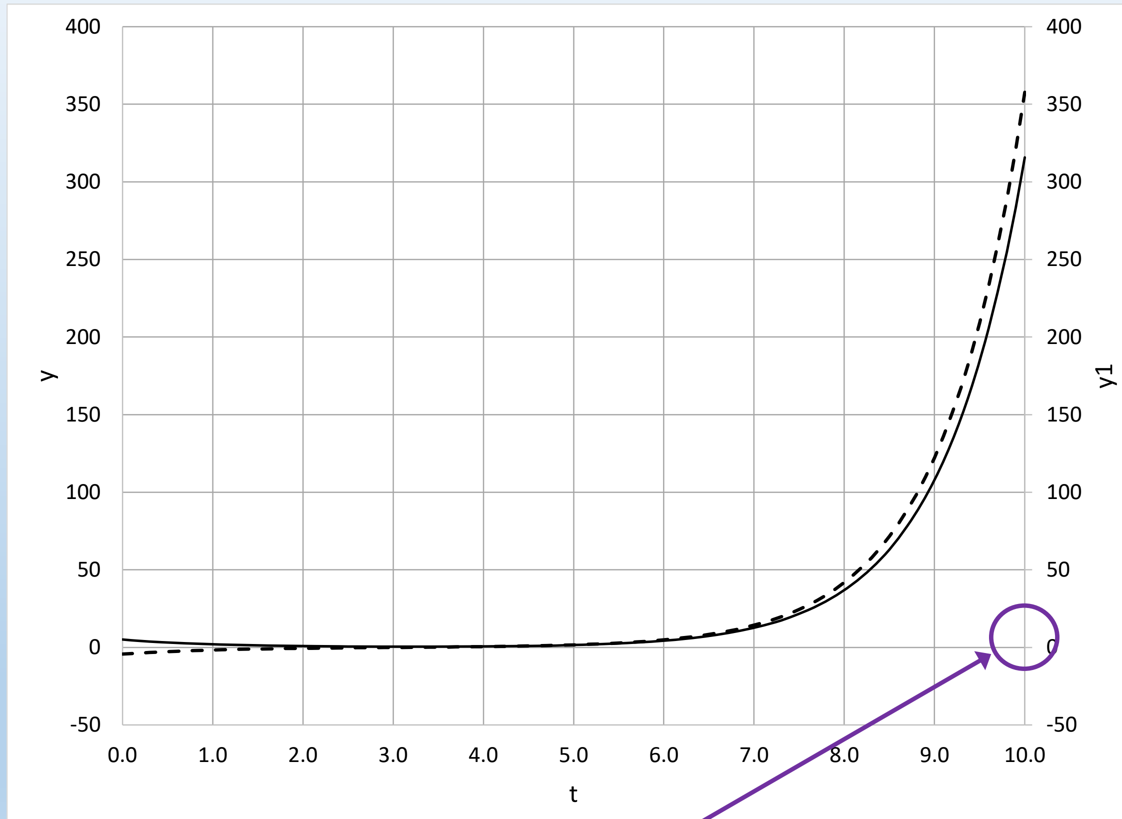
1. Estimate a value for y1 (dy/dt) at t = 0.
2. Solve the ODEs to t = 10
3. Check y(10) versus the required value, 8.
4. Adjust the y1(0) value and repeat steps 2 and 3 until the desired y(10)=8 value is obtained.

Solving Multiple Differential Equations

Second-order differential equation with split boundary conditions

y_0	5	y1_0	-4.4

t	y	y1	dy/dt	dy1/dt
0.0	5	-4.4	-4.4	3.9
0.1	4.56	-4.01	-4.01	3.56
0.2	4.16	-3.65	-3.65	3.25
0.3	3.79	-3.33	-3.33	2.96
0.4	3.46	-3.03	-3.03	2.70
●	●	●	●	●
9.5	184.64	209.15	209.15	236.93
9.6	205.55	232.84	232.84	263.76
9.7	228.84	259.22	259.22	293.64
9.8	254.76	288.59	288.59	326.91
9.9	283.62	321.28	321.28	363.94
10.0	315.75	357.67	357.67	405.16



SecondOrderODEStarter.xlsx

y(10) = 8 clearly not met

Solving Multiple Differential Equations

Second-order differential equation with split boundary conditions

Use the Solver to adjust y1(0) to achieve y(10) = 8

Solver Parameters

Set Objective: C106

To: ○ Max ○ Min ◉ Value Of: 8

By Changing Variable Cells:

y1_0

Subject to the Constraints:

y1_0 <= -4
y1_0 >= -5

y_0	5	y1_0	-4.4136

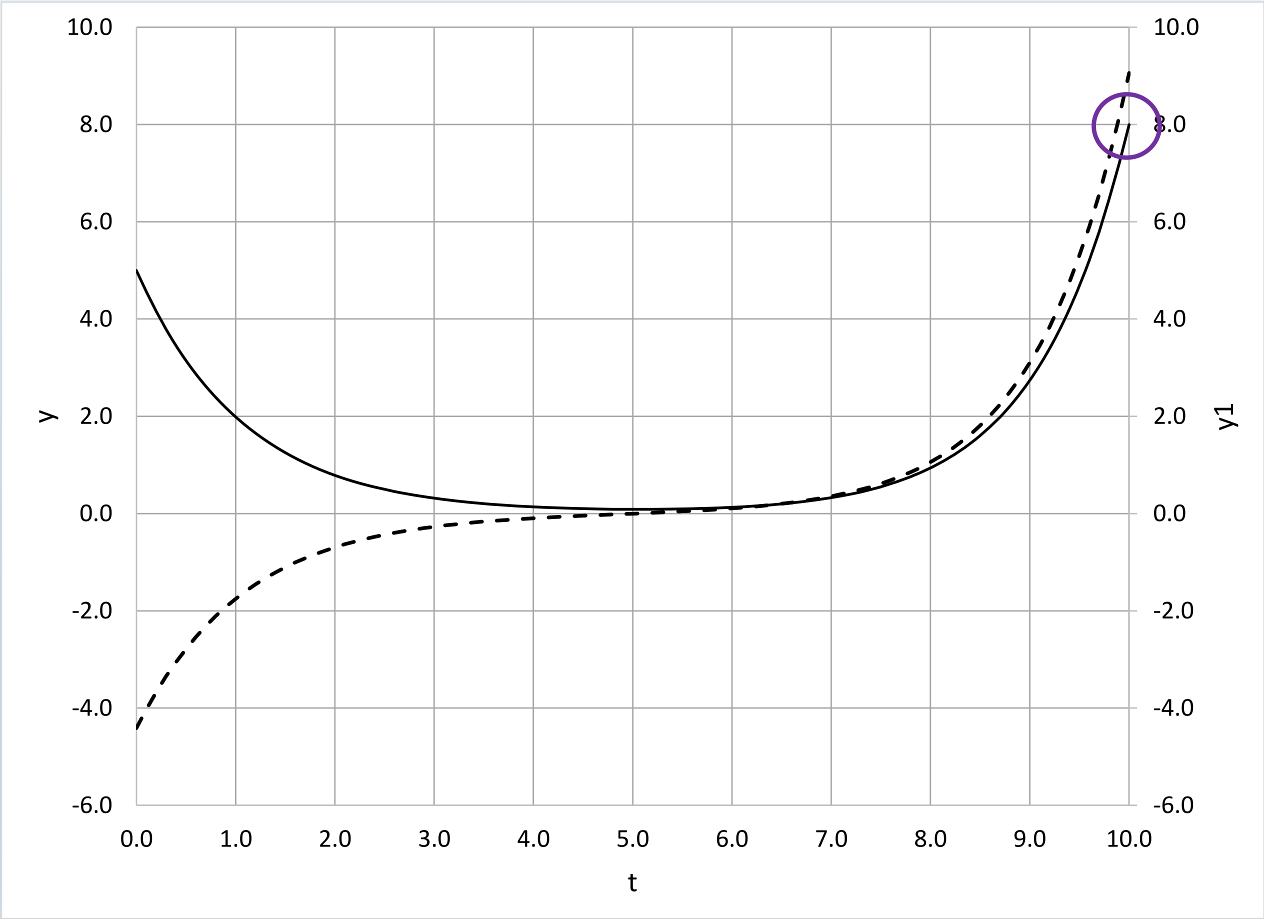
t	y	y1	dy/dt	dy1/dt
0.0	5	-4.4136	-4.41	3.8966
0.1	4.56	-4.02	-4.02	3.55
0.2	4.16	-3.67	-3.67	3.24
0.3	3.79	-3.34	-3.34	2.95
0.4	3.45	-3.05	-3.05	2.69
●	●	●	●	●
9.5	4.68	5.30	5.30	6.00
9.6	5.21	5.90	5.90	6.68
9.7	5.80	6.57	6.57	7.44
9.8	6.45	7.31	7.31	8.28
9.9	7.19	8.14	8.14	9.22
10.0	8.00	9.06	9.06	10.27

SecondOrderODE.xlsx

y(10) = 8 met

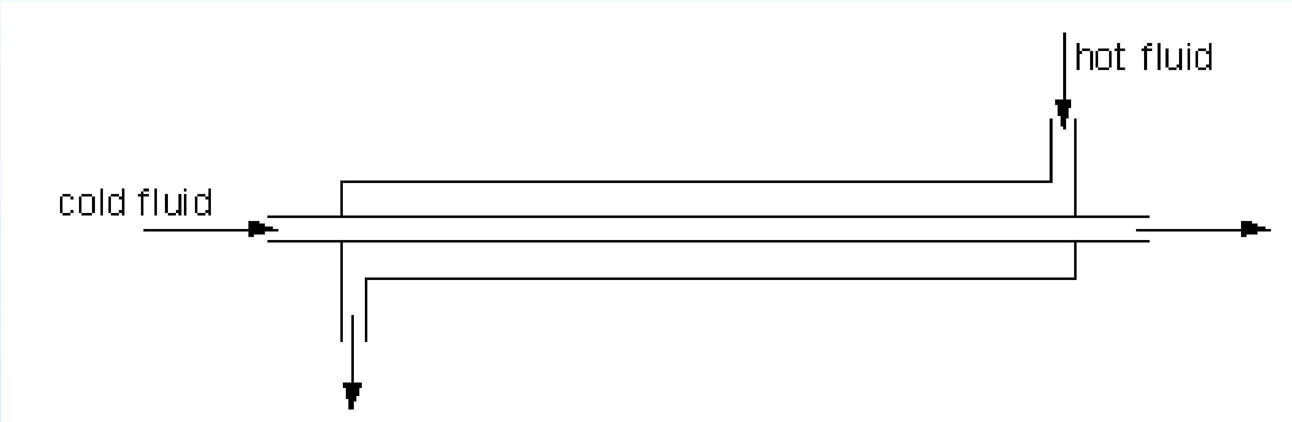
Solving Multiple Differential Equations

Second-order differential equation with split boundary conditions



Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger



$$\frac{dT_c}{dz} = \frac{h_i A_i}{w_c C_c}\left(T_h - T_c\right) \qquad T_c\left(0\right) = T_{ci}$$

$$\frac{dT_h}{dz} = \frac{h_o A_o}{w_h C_h}\left(T_h - T_c\right) \qquad T_h\left(L\right) = T_{hi}$$

$$h_o = \frac{h_i \cdot D_i}{D_o}$$

Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger

- z: distance down the heat exchanger from the cold fluid inlet (on the left)
- L: length of the heat exchanger
- T_c: temperature of the cold fluid, a function of z
- T_{ci}: cold water inlet temperature, at $z=0$
- T_{hi}: hot water inlet temperature, at $z=L$
- T_h: temperature of the hot fluid, a function of z
- w_c: mass flow rate of cold fluid
- w_h: mass flow rate of hot fluid
- C_c: heat capacity of cold fluid
- C_h: heat capacity of hot fluid
- A_i: inside area for heat transfer (cold fluid) per unit length
- A_o: outside area for heat transfer (hot fluid) per unit length
- h_i: inside heat transfer coefficient (cold fluid)
- h_o: outside heat transfer coefficient (hot fluid)

Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger

$$\frac{dT_c}{dz} = \frac{h_i A_i}{w_c C_c}\left(T_h - T_c\right) \qquad T_c\left(0\right) = T_{ci}$$

$$\frac{dT_h}{dz} = \frac{h_o A_o}{w_h C_h}\left(T_h - T_c\right) \qquad T_h\left(L\right) = T_{hi}$$

The issue we have with solving these equations is that the cold stream boundary condition is at z = 0 and the hot stream boundary condition is at z = L, the other end of the heat exchanger. A practical way to handle this is to estimate the hot stream temperature at z = 0, proceed with the solution, and adjust that estimate later on to meet the condition at z = L.

Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger

Basic data and operating conditions

Outer tube
11 BWG
OD 2 in, ID 1.76 in

Inner tube
11 BWG
OD 1 in, ID 0.76 in

Length 5 m

Inlet temperatures
Hot stream 50 °C
Cold stream 10 °C

Fluid density (H_2O)
988 kg/m^3

Heat capacity (H_2O)
4187 J/(kg·°C)

Hot stream flow rate 1 L/s
Cold stream 0.3 L/s

Heat transfer coefficient
h_i = 5000 W/(m^2·°C)

Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger

	A	B	C	D	E	F
1	Countercurrent Tube-in-tube Heat Exchanger					
2						
3	Tubing specs			density	988	kg/m^3
4	Outer	11	BWG	Heat capacity		
5	OD	2	inches	Cp	4186.8	J/kg/degC
6		0.0508	m	Flow rates		
7	ID	1.76	inches	Shell/hot	1.00	L/s
8		0.0447	m		0.001	m^3/s
9	Inside Area	2.43	in^2	wh	0.988	kg/s
10		0.00157	m^2	Tube/cold	0.30	L/s
11	Inner	11	BWG		0.0003	m^3/s
12	OD	1	inch	wc	0.296	kg/s
13	Do	0.0254	m	Linear velocities		
14	ID	0.76	inches	Shell/hot	0.94	m/s
15	Di	0.0193	m	Tube/cold	1.03	m/s

	A	B	C	D	E	F
16	Outside Area	0.785	in^2	Heat transfer coefficient		
17		0.000507	m^2	hi	5000	$W/m^2/degC$
18	Inside Area	0.454	in^2	ho	3800	$W/m^2/degC$
19		0.000293	m^2	Heat transfer areas/length		
20	Length -- L	5	m	Ai	0.0606	m^2/m
21				Ao	0.0798	m^2/m
22				Inlet temperatures		
23				Hot - Thi	50	degC
24				Cold - Tci	10	degC

CountercurrentHeatExchanger.xlsm

Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger

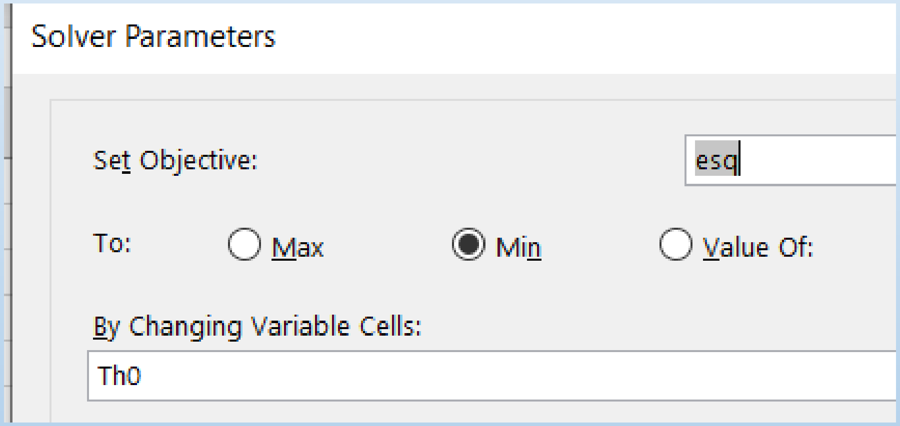
	H	I	J	K	L	M
2	Hot outlet temperature				$Error^2$	
3	Tho	40.0	degC		6.71	
4						
5	Index	Distance	Tc	Th	dTc/dz	dTh/dz
6	0	0	10	40.00	7.3	2.2
7	1	0.05	10.37	40.11	7.3	2.2
8	2	0.1	10.73	40.22	7.2	2.2
9	3	0.15	11.09	40.33	7.1	2.1
10	4	0.2	11.45	40.43	7.1	2.1
11	5	0.25	11.80	40.54	7.0	2.1
12	6	0.3	12.15	40.65	7.0	2.1
	•	•	•	•	•	•
100	94	4.7	33.74	47.12	3.3	1.0
101	95	4.75	33.90	47.17	3.2	1.0
102	96	4.8	34.07	47.22	3.2	1.0
103	97	4.85	34.23	47.27	3.2	1.0
104	98	4.9	34.39	47.32	3.2	0.9
105	99	4.95	34.54	47.36	3.1	0.9
106	100	5	34.70	47.41	3.1	0.9

Hot inlet temperature (50 degC) condition not met

Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger

Employ Solver to adjust the hot stream outlet temperature so that its inlet condition is met.

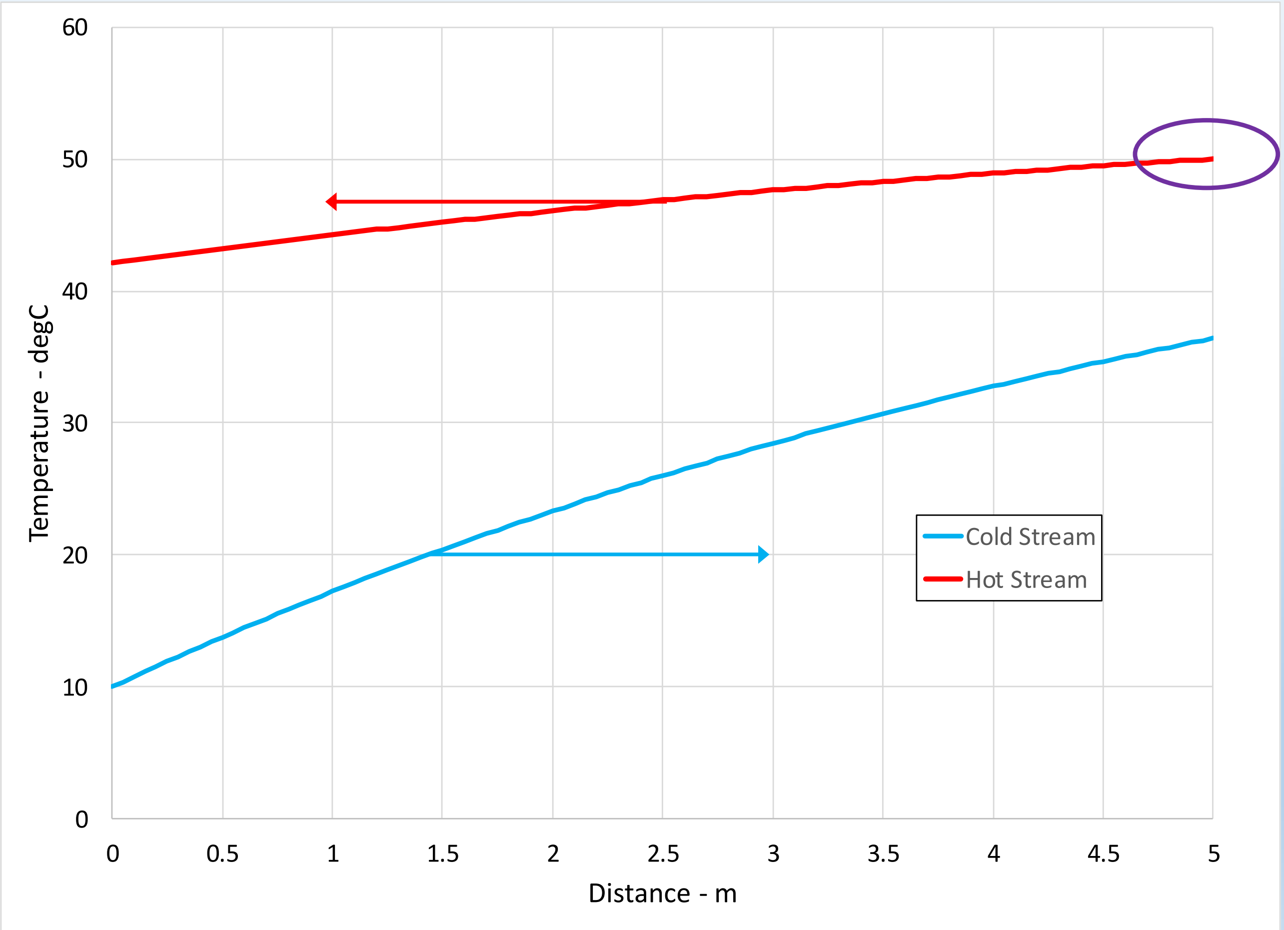


Hot outlet temperature				Error²
Tho	42.1	degC		0.00

99	4.95	36.24	49.95	3.3	1.0
100	5	36.41	50.00	3.3	1.0

Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger



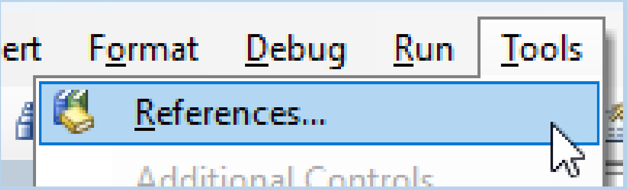
Solving Multiple Differential Equations

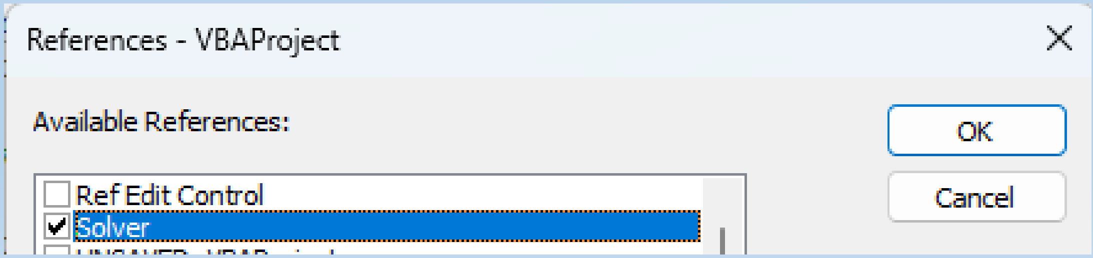
Example: tube-in-tube, countercurrent heat exchanger

Record macro to run the Solver

```
Option Explicit
Sub SolveHtExr()
'
' SolveHtExr Macro
'
    SolverOk SetCell:="$L$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$I$3", Engine:=1 _
        , EngineDesc:="GRG Nonlinear"
    SolverSolve
End Sub
```

Necessary to add Solver reference in VBE





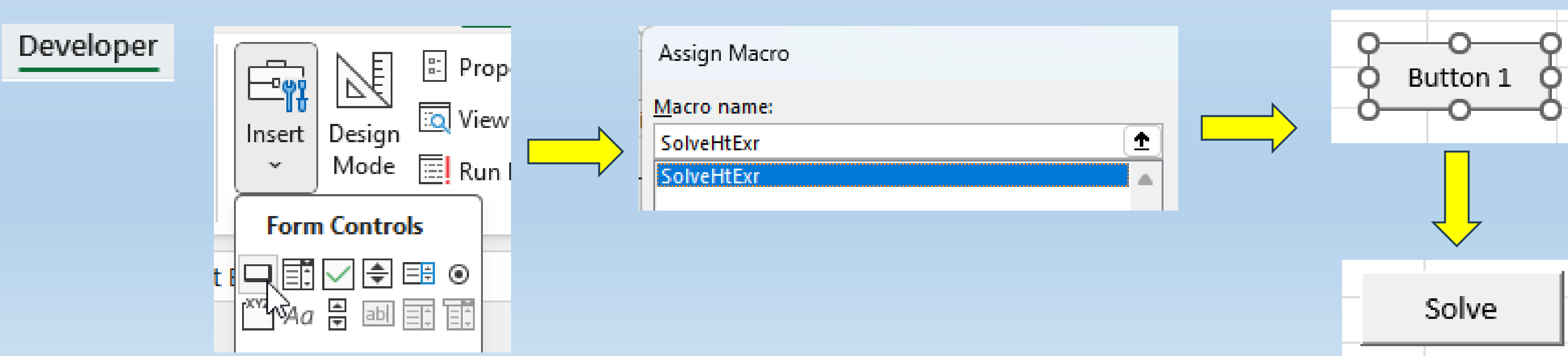
Solving Multiple Differential Equations

Example: tube-in-tube, countercurrent heat exchanger

Modify macro to bypass confirmation dialog box

```
Option Explicit
Sub SolveHtExr()
'
' SolveHtExr Macro
'
    SolverOk SetCell:="$L$3", MaxMinVal:=2, ValueOf:=0, ByChange:="$I$3", Engine:=1 _
        , EngineDesc:="GRG Nonlinear"
    SolverSolve UserFinish:=True
End Sub
```

Add button on spreadsheet to run macro



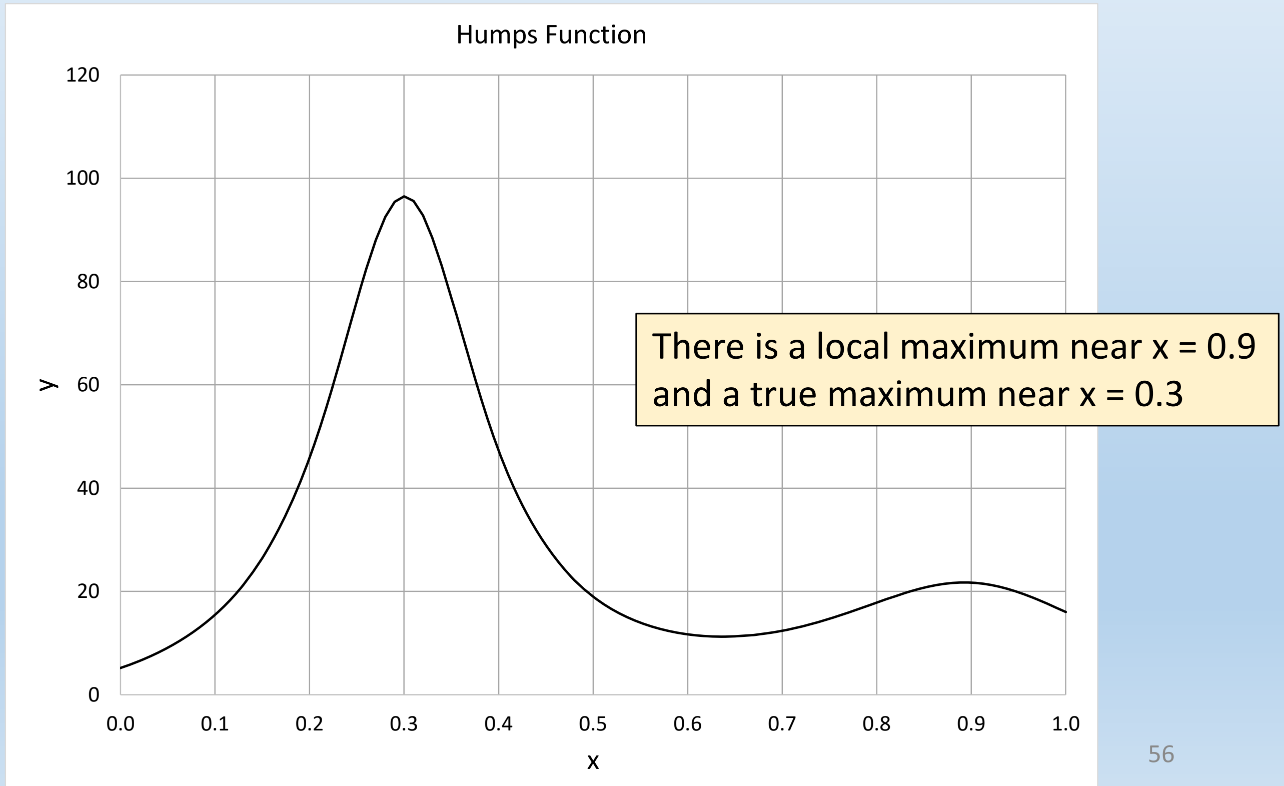
Optimization

Finding a maximum or minimum of a function with a single adjustable variable

HumpsOptimizationStarter.xlsx

Example $$y = \frac{1}{(x-0.3)^2 + 0.01} + \frac{1}{(x-0.9)^2 + 0.04} - 6$$

x	y
0.00	5.18
0.01	5.83
0.02	6.54
0.03	7.32
0.04	8.17
0.05	9.10
●	●
0.94	20.42
0.95	19.84
0.96	19.18
0.97	18.45
0.98	17.67
0.99	16.85
1.00	16.00

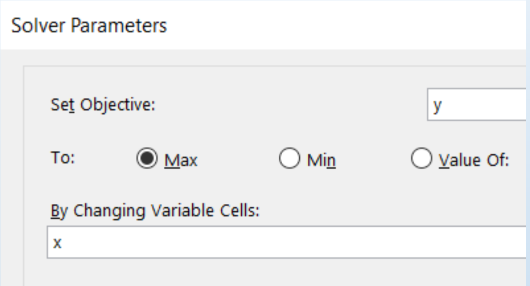


Optimization

Finding the maximum of a function with a single adjustable variable

=1/((x-0.3)^2+0.01)+1/((x-0.9)^2+0.04)-6

B	C	D	E
x	0.7		
y	12.382		



x	0.89272
y	21.735

Finds the local maximum

x	0.5
y	19.000

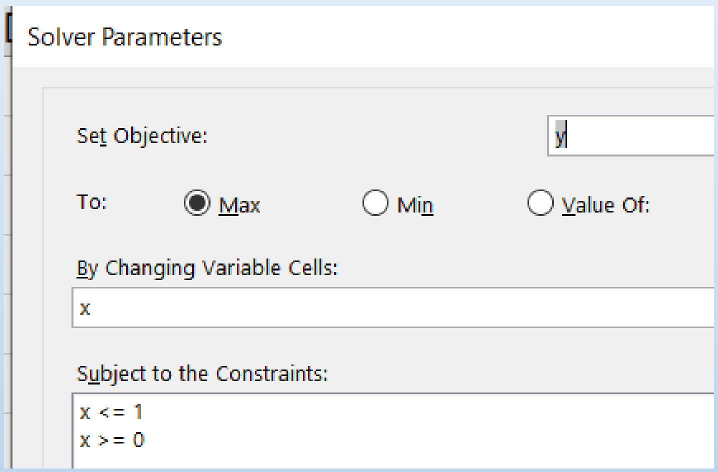
x	0.30038
y	96.501

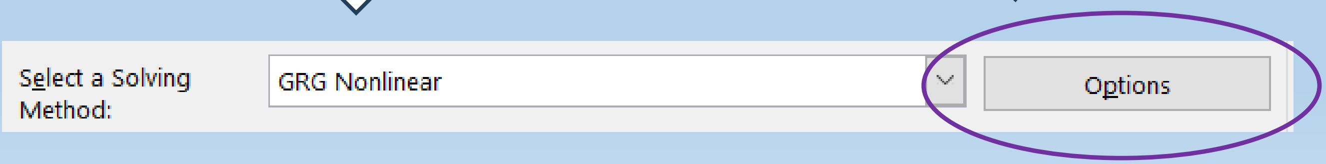
Finds the true maximum

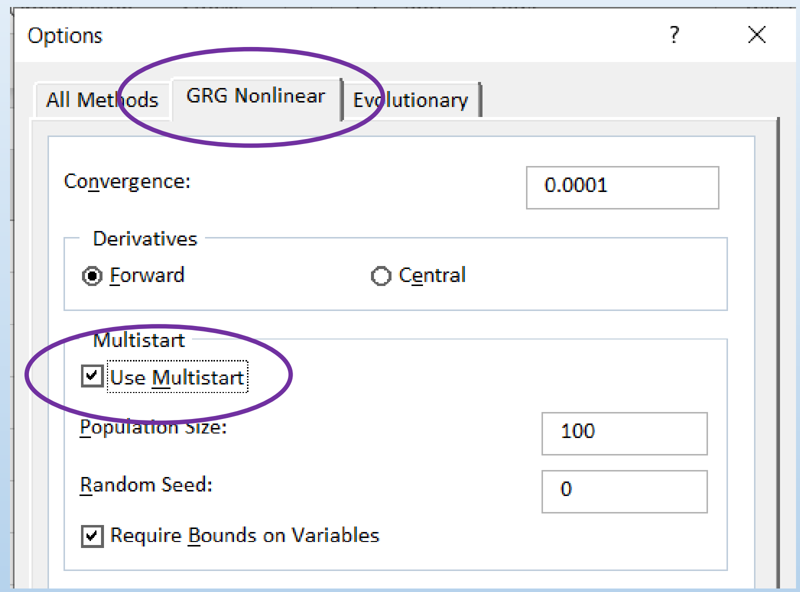
Optimization

Finding the maximum of a function with a single adjustable variable

Using the GRG Nonlinear Multistart option







Finds the true maximum

x	0.30038
y	96.501

Optimization

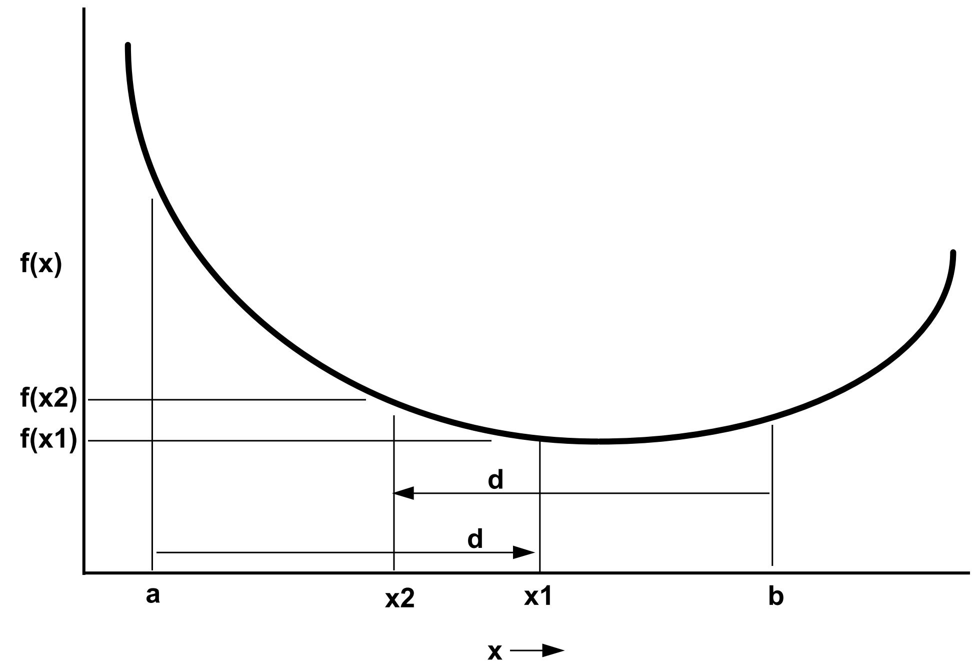
Finding a maximum or minimum of a function with a single adjustable variable - the Golden Section search

This is a bracketing method, similar to bisection. The figure shows a curve *f(x)* with a minimum between two initial estimates, *a* and *b*. Instead of using the midpoint between *a* and *b*, an overlapping interval *d* is used to compute x_1 and x_2. The interval is given by

$$d = \frac{\sqrt{5}-1}{2}\cdot(b-a)$$

where $\frac{\sqrt{5}-1}{2}$ is the Golden Ratio (*GR*)

with the unique property $GR = \frac{1}{1+GR}$



Optimization

Finding a maximum or minimum of a function with a single adjustable variable - the Golden Section search

For the figure to the right, we can see that

$$f(x_2) > f(x_1)$$

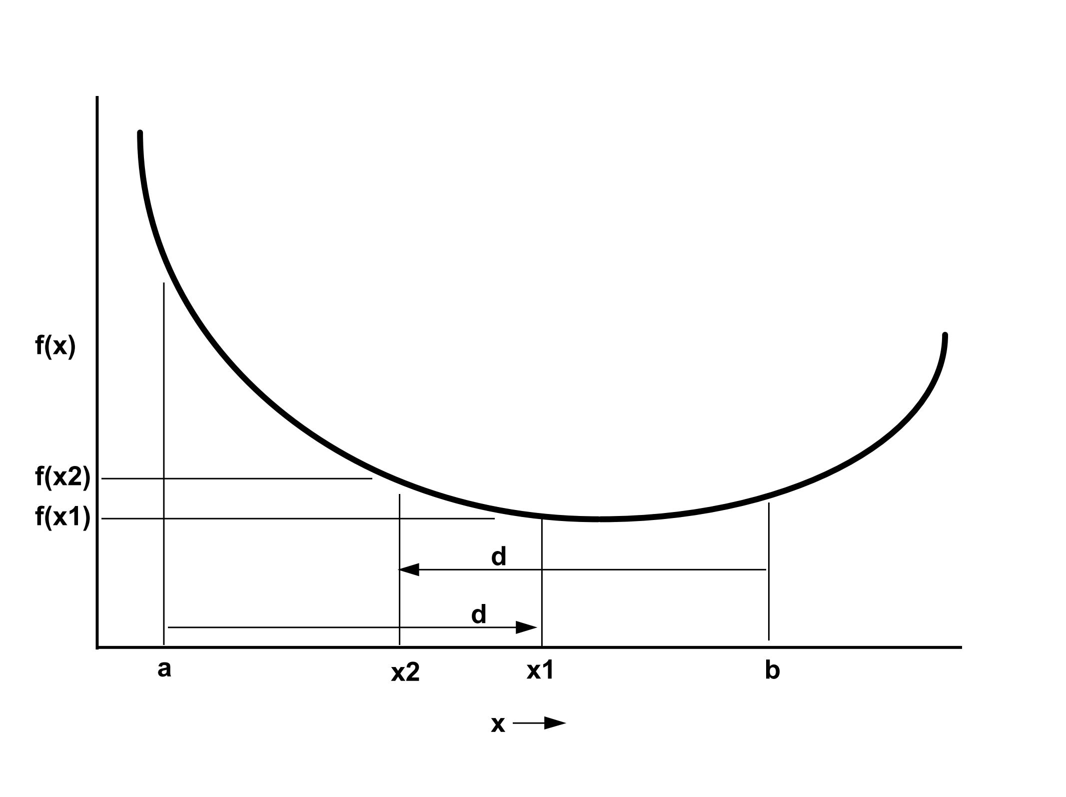
and that leads to the conclusion that the minimum must be between x_2 and b, and the interval $[a,x_2]$ can be excluded. This implies that x_2 becomes the new a for the next iteration of the method.

If

$$f(x_1) > f(x_2)$$

the interval $[x_1,b]$ would be excluded and x_1 would become the next b.

For each iteration, the interval containing the minimum is reduced by a factor of *GR*.



Any ratio other than *GR* that is greater than 0.5 and less than 1 could be used. The advantage of using the *GR* is that, for the first scenario above, x_1 is the x_2 value for the next iteration, avoiding the need to compute $f(x_2)$ then.

Optimization

HumpsOptimizationStarter.xlsx

Golden Section search to find a minimum of the humps function

	A	B	C	D	E	F	G	H
1	Humps Function Optimization			GR	0.618			
2	Using Golden Section Search							
3	Iteration	a	x2	f(x2)	x1	f(x1)	b	d
4	1	0.4	0.553	13.76	0.647	11.28	0.8	0.247

Initialization row

	A	B	C	D
1	Humps Fun			
2	Using Golde			
3	Iteration	a	x2	f(x2)
4	1	0.4	=G4-H4	=1/((C4-0.3)^2+0.01)+1/((C4-0.9)^2+0.04)-6

	E	F	G	H
1	=(SQRT(5)-1)/2			
2				
3	x1	f(x1)	b	d
4	=B4+H4	=1/((E4-0.3)^2+0.01)	0.8	=GR*(G4-B4)

Optimization

Golden Section search to find a minimum of the humps function

	A	B	C	D	E	F	G	H
5	2	0.553	0.647	11.28	0.706	12.58	0.8	0.153

First operational row

	A	B	C	D
5	=A4+1	=IF(D4>F4,C4,B4)	=G5-H5	=1/((C5-0.3)^2+0.01)+1/((C5-0.9)^2+0.04)-6

	E	F	G	H
5	=B5+H5	=1/((E5-0.3)^2+0.01)	=IF(F4>D4,E4,G4)	=GR*(G5-B5)

Copy down

Result from Row 33

Minimum at	0.637
Value	11.25

	A	B	C	D	E	F	G	H
30	27	0.637	0.637	11.25	0.637	11.25	0.637	0.000
31	28	0.637	0.637	11.25	0.637	11.25	0.637	0.000
32	29	0.637	0.637	11.25	0.637	11.25	0.637	0.000
33	30	0.637	0.637	11.25	0.637	11.25	0.637	0.000

Optimization

Golden Section search to find a minimum of the humps function

VBA Function for Golden Section search

```
Option Explicit
Function MinGold(a, b, maxit)
Dim GR, x1, x2, d, i As Integer
GR = (Sqr(5) - 1) / 2
For i = 1 To maxit
    d = GR * (b - a)
    x1 = a + d
    x2 = b - d
    If f(x2) > f(x1) Then
        a = x2
    Else
        b = x1
    End If
Next i
MinGold = (x1 + x2) / 2
End Function
```

MinGold VBA function	
a	0.4
b	0.8
maxit	30
Minimum at	0.637
Value	11.25

	J	K
7	MinGold VBA function	
8	a	0.4
9	b	0.8
10	maxit	30
11	Minimum at	=mingold(a,b,maxit)
12	Value	=f(K11)

```
Function f(x)
f = 1 / ((x - 0.3) ^ 2 + 0.01) + 1 / ((x - 0.9) ^ 2 + 0.04) - 6
End Function
```

Optimization

Finding a maximum or minimum of a function with multiple adjustable variables and one or more constraints

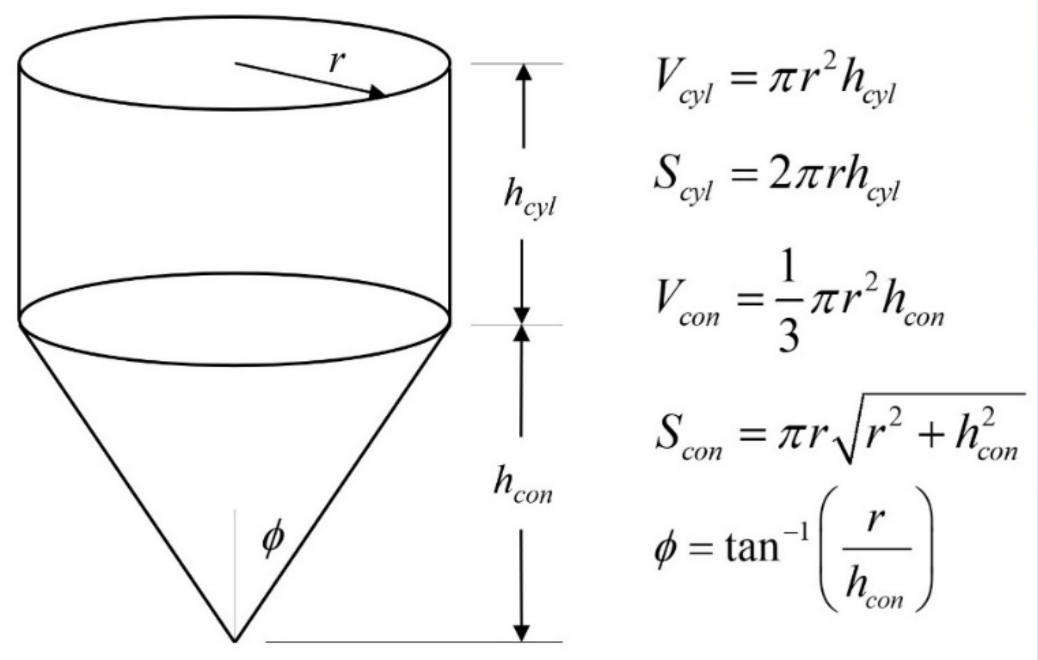
Example:
Optimal grain bin design

Minimize surface area not including the top

Constraints

$$V = V_{cyl} + V_{con} = 10 \text{ m}^3$$

$$\phi_{\max} = 20.4°$$



Optimization

Example:
Optimal grain bin design

GrainBin.xlsx

Initial Scenario

radius	1	m	Vcyl	3.14159	m^3	Scyl	6.28319	m^2	phi	0.7854	radians
hcyl	1	m	Vcon	1.0472	m^3	Scon	4.44288	m^2	phid	45	degrees
hcon	1	m	V	4.18879	m^3	S	10.7261	m^2			

Volume spec not met

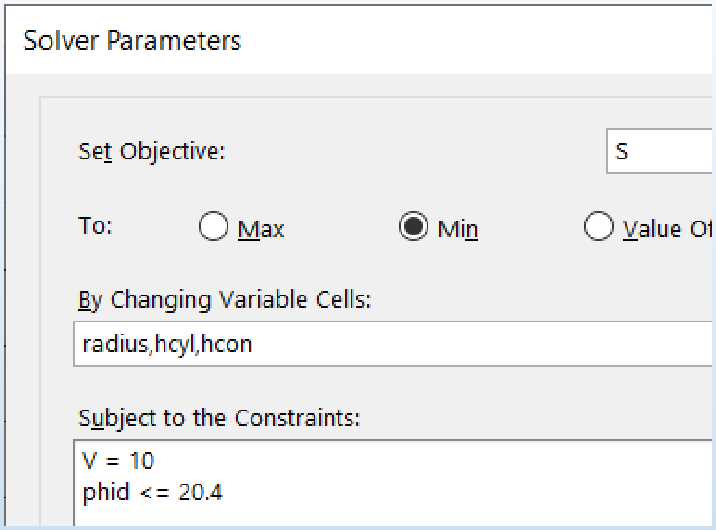
Angle limit violated

Formulas

radius	1	m	Vcyl	=PI()*radius^2*hcyl	m^3	Scyl	=2*PI()*radius*hcyl	m^2	phi	=ATAN(radius/hcon)	radians
hcyl	1	m	Vcon	=PI()*radius^2*hcon/3	m^3	Scon	=PI()*radius*SQRT(radius^2+hcon^2)	m^2	phid	=DEGREES(phi)	degrees
hcon	1	m	V	=Vcyl+Vcon	m^3	S	=Scyl+Scon	m^2			

Optimization

Example:
Optimal grain bin design



radius	1.44	m	Vcyl	1.67	m^3	Scyl	2.33	m^2	phi	0.356	radians
hcyl	0.26	m	Vcon	8.33	m^3	Scon	18.57	m^2	phid	20.4	degrees
hcon	3.86	m	V	10	m^3	S	20.90	m^2			

Small cylindrical part

Specifications met
Angle at the constraint

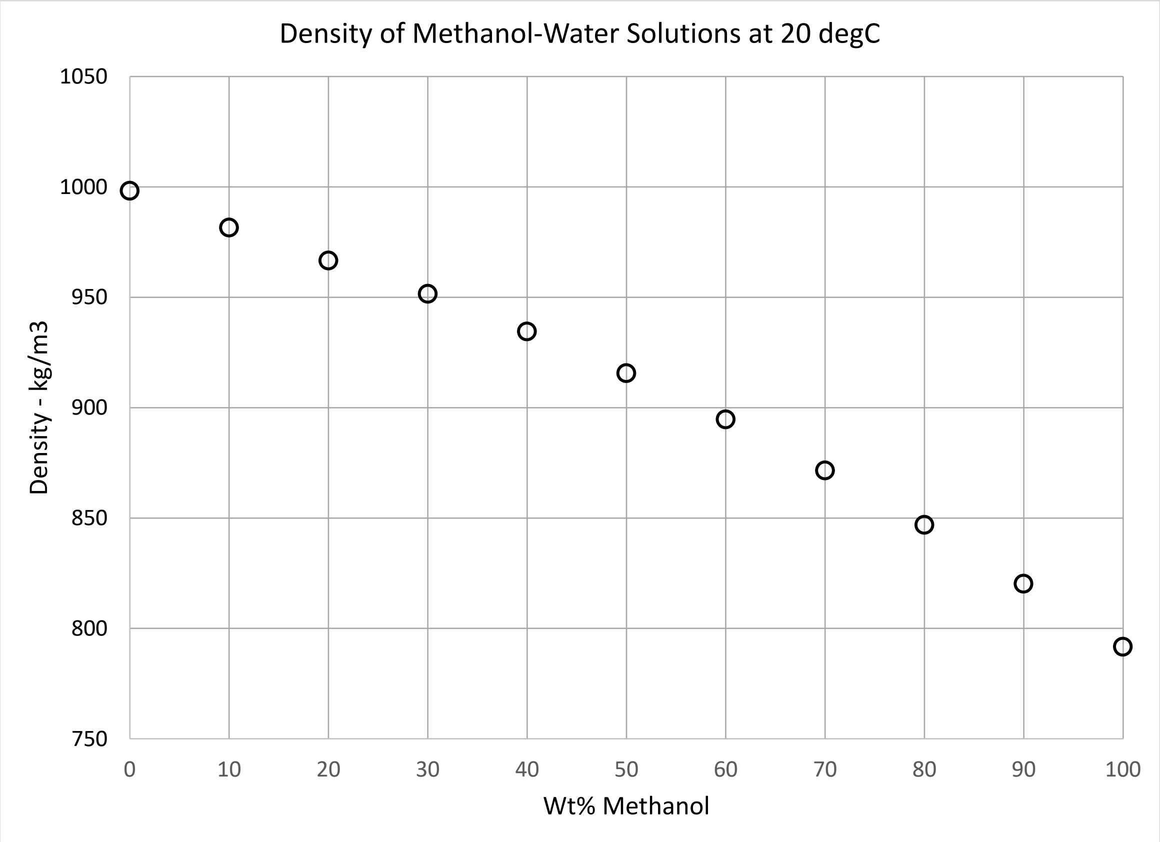
Minimum area

Curve-Fitting

Polynomial regression

Example:

	A	B	C	D	E
1	Density of Methanol-Water Solutions at 20 degC				
2	Wt% Methanol	Density (kg/m^3)			
3	0	998.2			
4	10	981.5			
5	20	966.6			
6	30	951.5			
7	40	934.5			
8	50	915.6			
9	60	894.6			
10	70	871.5			
11	80	846.9			
12	90	820.2			
13	100	791.7			

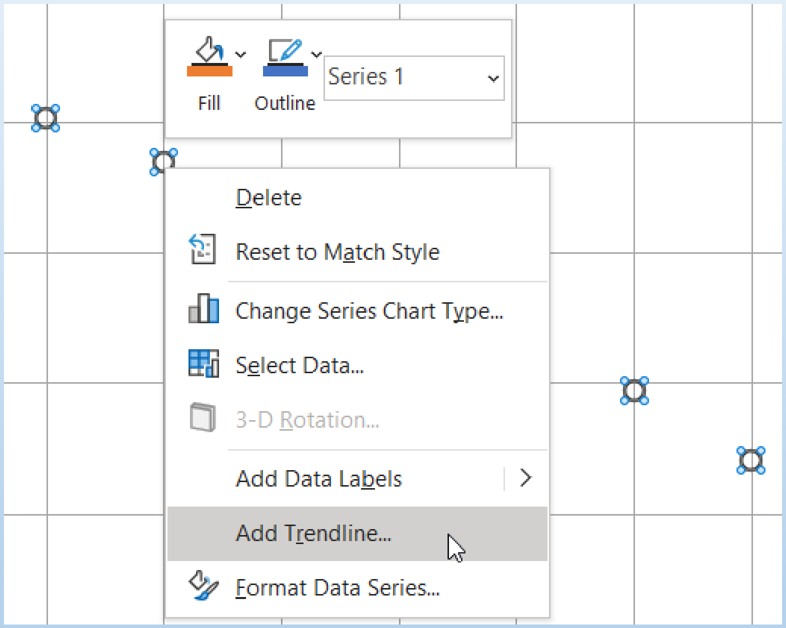


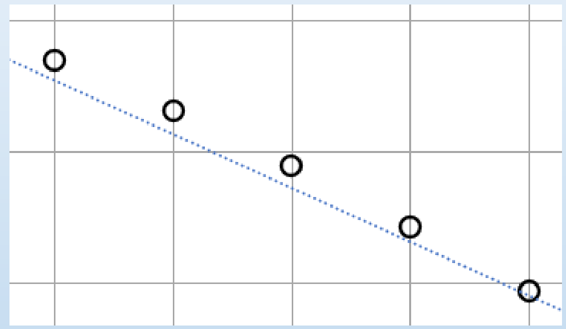
MethanolWaterDensityStarter.xlsx

Curve-Fitting

Polynomial regression

Using Trendline



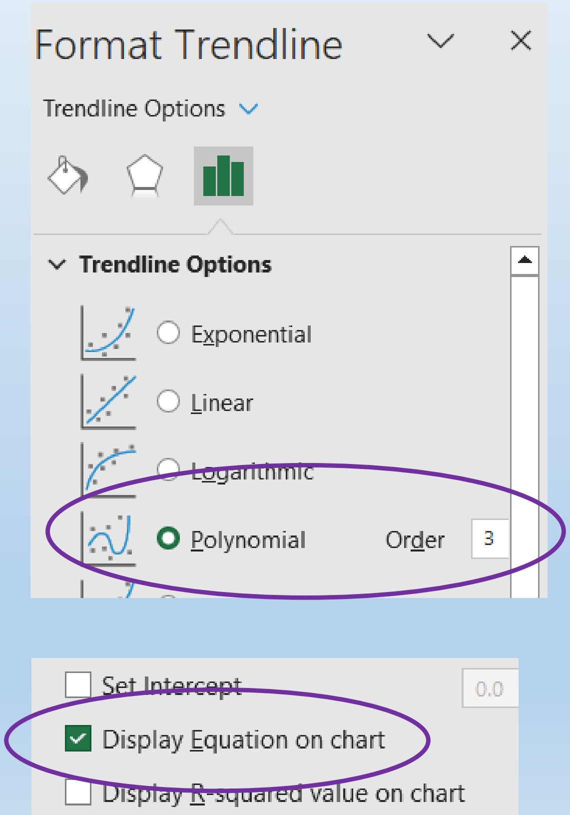


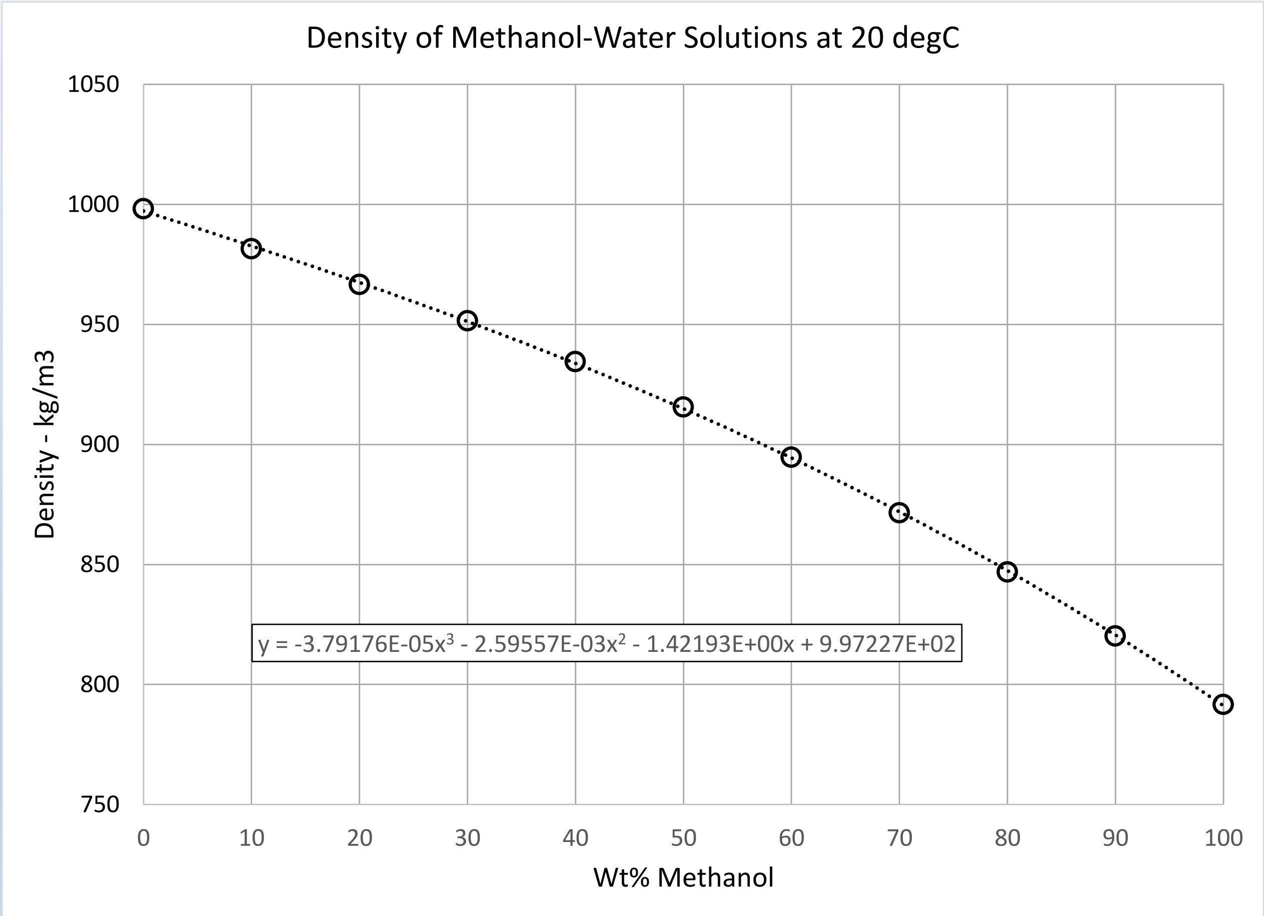
Initial fit is a straight line – clearly inadequate

Curve-Fitting

Polynomial regression

Using Trendline

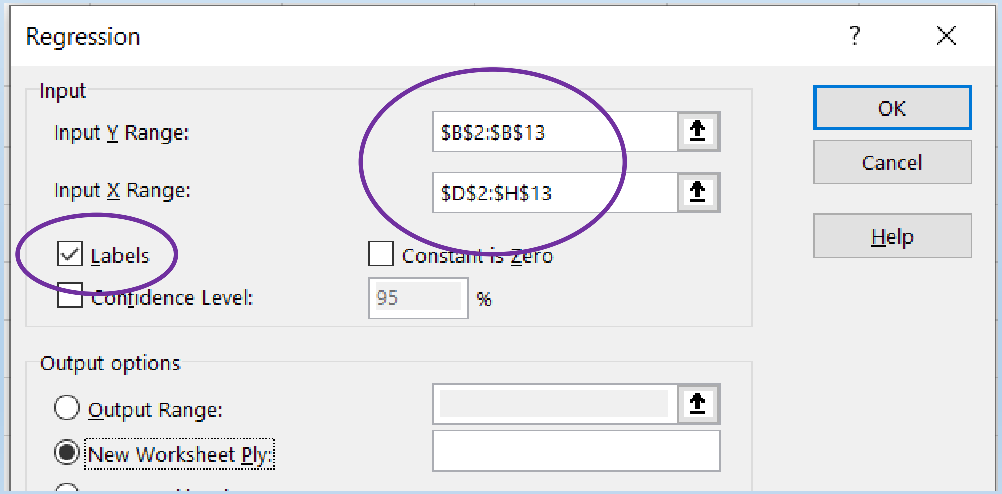




Curve-Fitting

Polynomial regression   Using the Data Analysis Regression tool

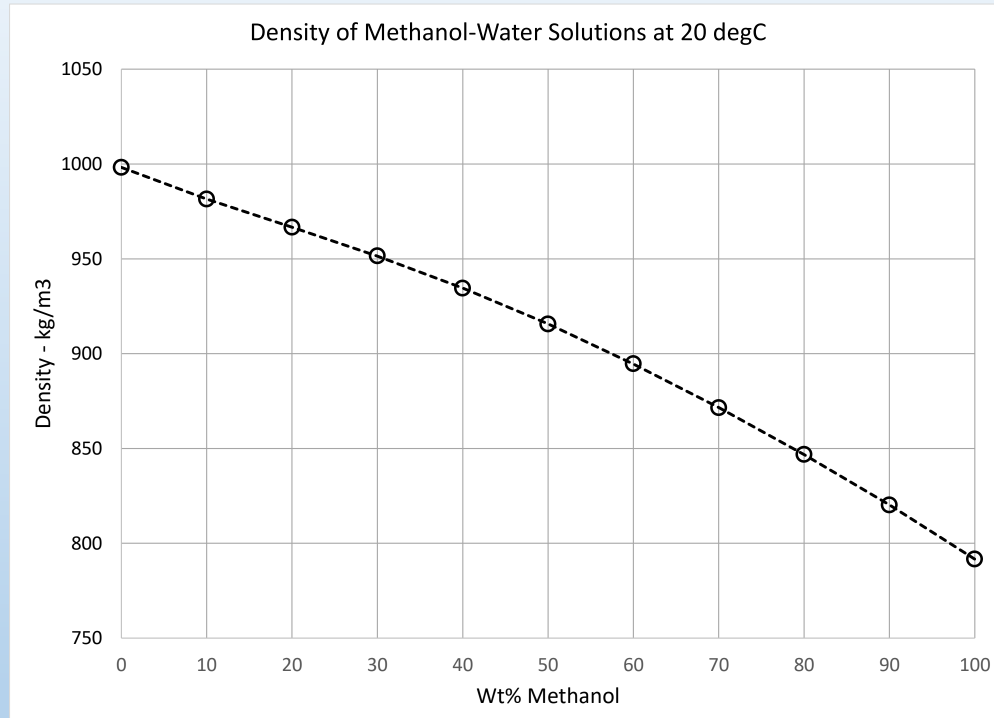
	A	B	C	D	E	F	G	H
2	Wt% Methanol	Density (kg/m^3)		1	2	3	4	5
3	0	998.2		0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00
4	10	981.5		1.0E+01	1.0E+02	1.0E+03	1.0E+04	1.0E+05
5	20	966.6		2.0E+01	4.0E+02	8.0E+03	1.6E+05	3.2E+06
6	30	951.5		3.0E+01	9.0E+02	2.7E+04	8.1E+05	2.4E+07
7	40	934.5		4.0E+01	1.6E+03	6.4E+04	2.6E+06	1.0E+08
8	50	915.6		5.0E+01	2.5E+03	1.3E+05	6.3E+06	3.1E+08
9	60	894.6		6.0E+01	3.6E+03	2.2E+05	1.3E+07	7.8E+08
10	70	871.5		7.0E+01	4.9E+03	3.4E+05	2.4E+07	1.7E+09
11	80	846.9		8.0E+01	6.4E+03	5.1E+05	4.1E+07	3.3E+09
12	90	820.2		9.0E+01	8.1E+03	7.3E+05	6.6E+07	5.9E+09
13	100	791.7		1.0E+02	1.0E+04	1.0E+06	1.0E+08	1.0E+10



Curve-Fitting

Polynomial regression   Using the Data Analysis Regression tool

	A	B	C	D	E
1	SUMMARY OUTPUT				
2					
3	*Regression Statistics*				
4	Multiple R	0.9999995			
5	R Square	99.99990%			
6	Adjusted R Square	99.99980%			
7	Standard Error	9.695E-02			
8	Observations	11			
9					
10		*Coefficients*	*Standard Error*	*t Stat*	*P-value*
11	Intercept	9.982E+02	9.567E-02	1.043E+04	0.000%
12	1	-1.853E+00	2.318E-02	-7.994E+01	0.000%
13	2	2.463E-02	1.629E-03	1.512E+01	0.002%
14	3	-6.489E-04	4.345E-05	-1.494E+01	0.002%
15	4	5.673E-06	4.875E-07	1.164E+01	0.008%
16	5	-1.859E-08	1.940E-09	-9.580E+00	0.021%
17					

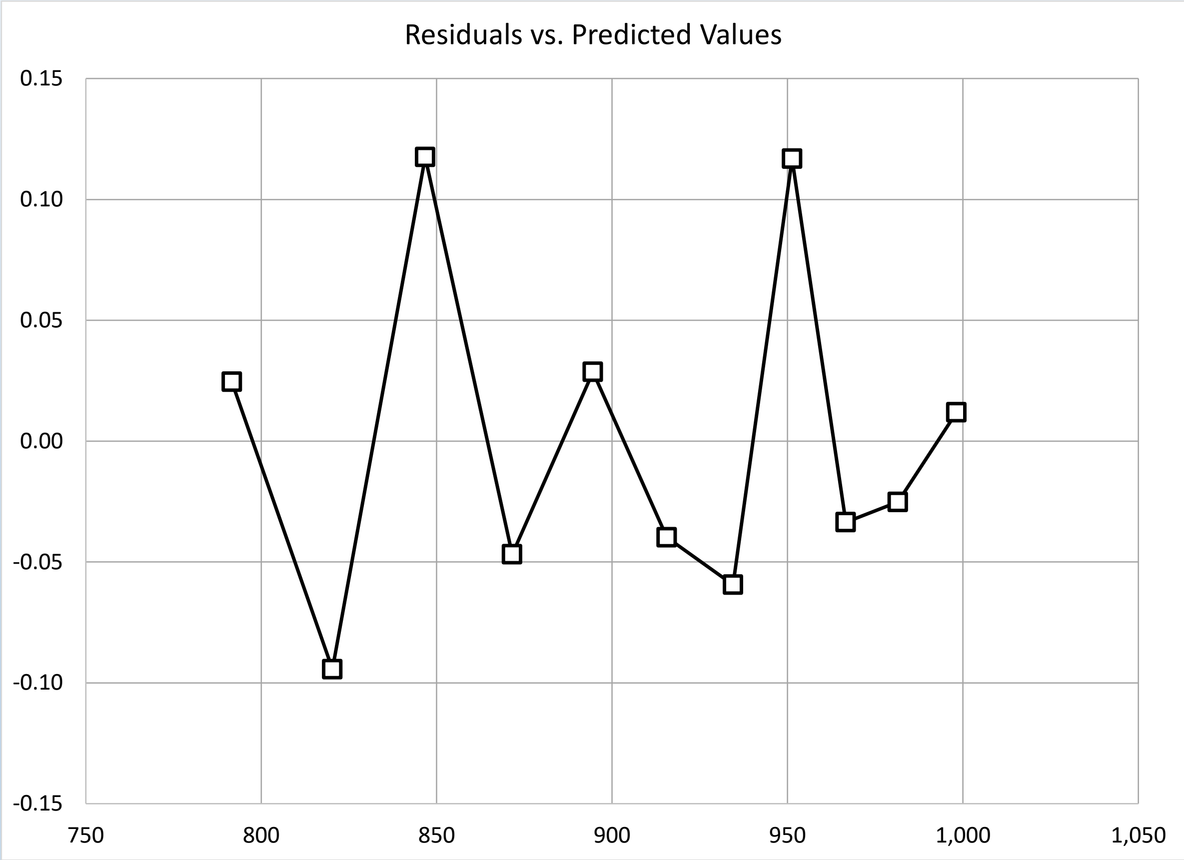


$$\rho = 998.2 - 1.853w + 0.02463w^2 - 6.489\times10^{-4}w^3 + 5.673\times10^{-6}w^4 - 1.859\times10^{-8}w^5$$

MethanolWaterDensity.xlsx

Curve-Fitting

Polynomial regression   Using the Data Analysis Regression tool



No significant pattern.
Model is adequate.

Curve-Fitting

Multilinear regression

Model $y = \beta_0 + \beta_1 f_1\left(x_j, j=1,\ldots,m\right) + \beta_2 f_2\left(x_j, j=1,\ldots,m\right) + \cdots + \beta_k f_k\left(x_j, j=1,\ldots,m\right)$

measurement responses

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

realization matrix of input levels

$$\mathbf{X} = \begin{bmatrix} 1 & f_{11} & f_{21} & \cdots & f_{m1} \\ 1 & f_{12} & f_{22} & \cdots & f_{m2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & f_{1n} & f_{2n} & \cdots & f_{mn} \end{bmatrix}$$

intercept

parameter estimates

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_k \end{bmatrix}$$

model predictions

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$$

residuals

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \qquad \mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$$

sum of squares criterion

$$V = \sum_{i=1}^{n} e_i^2 = \mathbf{e}^T \cdot \mathbf{e}$$

Dataset

$$\left\{y_i, x_{1i}, \ldots, x_{mi}, i = 1, \ldots, n\right\}$$

Minimize V by choice of $\hat{\boldsymbol{\beta}}$ via calculus

Normal equations

$$\left(\mathbf{X}^T\mathbf{X}\right)\mathbf{b} = \mathbf{X}^T\mathbf{y}$$

Fitted model parameters

$$\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

Curve-Fitting

Multilinear regression

Example

Density of NaCl Aqueous Solutions					
		Temperature			
		0 °C	10 °C	25 °C	40 °C
Wt % NaCl	1	1.00747	1.00707	1.00409	0.99908
	2	1.01509	1.01442	1.01112	1.00593
	4	1.03038	1.02920	1.02530	1.01977
	8	1.06121	1.05907	1.05412	1.04798
	12	1.09244	1.08946	1.08365	1.07699
	16	1.12419	1.12056	1.11401	1.10688
	20	1.15663	1.15254	1.14533	1.13774
	24	1.18999	1.18557	1.17776	1.16971
	26	1.20709	1.20254	1.19443	1.18614

from *Perry's Chemical Engineer's Handbook*, Green and Southard, Ed., 9th Ed., p. 2-103.

Model

$$\rho = \beta_0 + \beta_1 w + \beta_2 T + \beta_3 w^2 + \beta_4 T^2 + \beta_5 wT$$

NaClDensityRegressionStarter.xlsx

Curve-Fitting

Multilinear regression using Data Analysis Regression tool

Set up X-Input array

wt%	degC	dens
1	0	1.00747
2	0	1.01509
4	0	1.03038
8	0	1.06121
12	0	1.09244
16	0	1.12419
20	0	1.15663
24	0	1.18999
26	0	1.20709
1	10	1.00707
2	10	1.01442

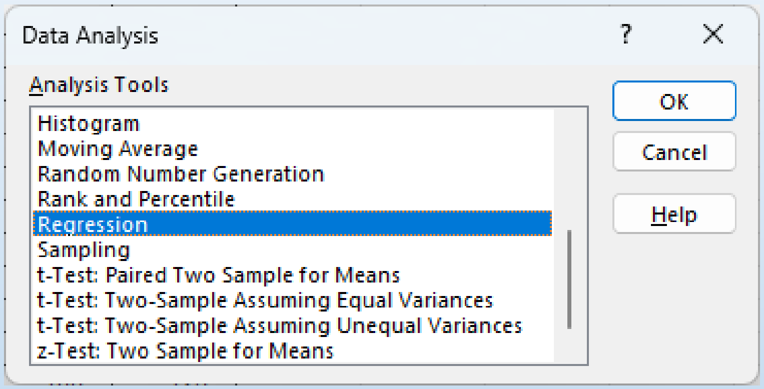
w	T	w2	T2	wT
1	0	1	0	0
2	0	4	0	0
4	0	16	0	0
8	0	64	0	0
12	0	144	0	0
16	0	256	0	0
20	0	400	0	0
24	0	576	0	0
26	0	676	0	0
1	10	1	100	10
2	10	4	100	20

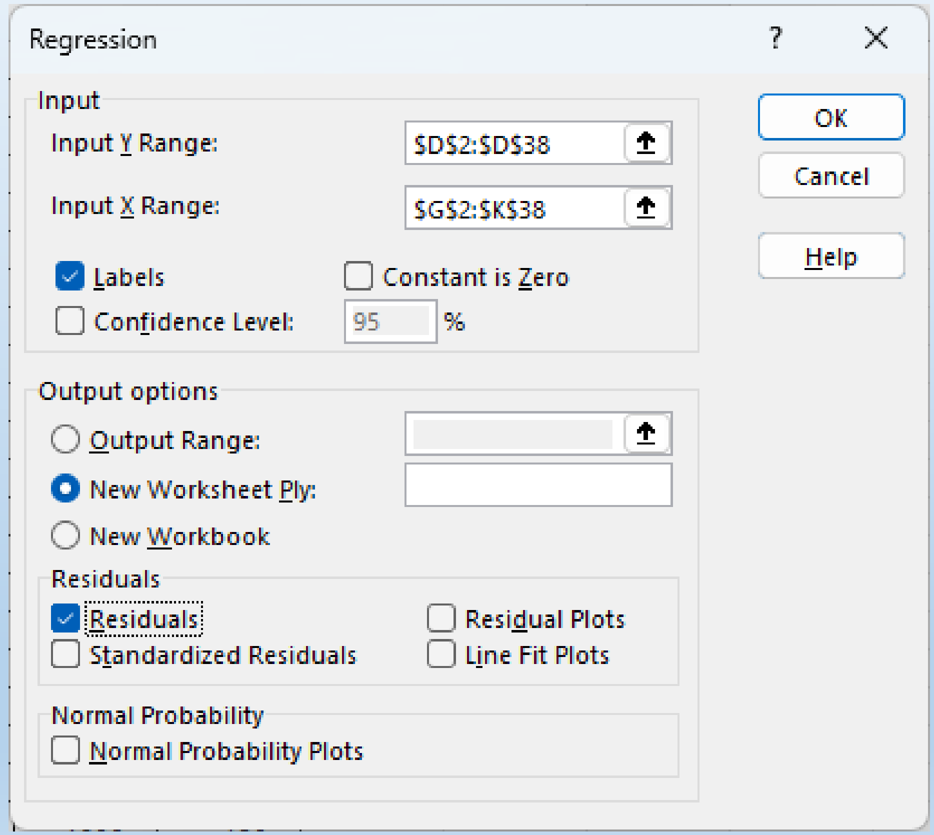
$$\rho = \beta_0 + \beta_1 w + \beta_2 T + \beta_3 w^2 + \beta_4 T^2 + \beta_5 wT$$

NaClDensityDataAnalysisRegression.xlsx

Curve-Fitting

Multilinear regression using Data Analysis Regression tool





Curve-Fitting

Multilinear regression using Data Analysis Regression tool

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.9999866
R Square	99.99731%
Adjusted R Square	99.99687%
Standard Error	3.919E-04
Observations	36

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	1.001E+00	2.186E-04	4.580E+03	0.000%	1.001E+00	1.002E+00
w	7.274E-03	3.184E-05	2.285E+02	0.000%	7.209E-03	7.339E-03
T	-9.780E-05	1.712E-05	-5.713E+00	0.000%	-1.328E-04	-6.284E-05
w2	2.496E-05	1.122E-06	2.225E+01	0.000%	2.266E-05	2.725E-05
T2	-2.997E-06	3.825E-07	-7.834E+00	0.000%	-3.778E-06	-2.215E-06
wT	-1.254E-05	4.825E-07	-2.599E+01	0.000%	-1.353E-05	-1.156E-05

$$\rho = 1.001 + 0.007274w - 9.780\times10^{-5}T$$

$$+2.496\times10^{-5}w^2 - 2.997\times10^{-6}T^2 - 1.254\times10^{-5}wT$$

Curve-Fitting

Multilinear regression using vector-matrix calculations

	B	C	D
2	wt%	degC	dens
3	1	0	1.00747
4	2	0	1.01509
5	4	0	1.03038
6	8	0	1.06121
7	12	0	1.09244
8	16	0	1.12419
9	20	0	1.15663
10	24	0	1.18999
11	26	0	1.20709
12	1	10	1.00707
13	2	10	1.01442
	●	●	●
29	26	25	1.19443
30	1	40	0.99908
31	2	40	1.00593
32	4	40	1.01977
33	8	40	1.04798
34	12	40	1.07699
35	16	40	1.10688
36	20	40	1.13774
37	24	40	1.16971
38	26	40	1.18614

	F	G	H	I	J	K
2	X matrix					
3	1	1	0	1	0	0
4	1	2	0	4	0	0
5	1	4	0	16	0	0
6	1	8	0	64	0	0
7	1	12	0	144	0	0
8	1	16	0	256	0	0
9	1	20	0	400	0	0
10	1	24	0	576	0	0
11	1	26	0	676	0	0
12	1	1	10	1	100	10
	●	●	●	●	●	●
27	1	20	25	400	625	500
28	1	24	25	576	625	600
29	1	26	25	676	625	650
30	1	1	40	1	1600	40
31	1	2	40	4	1600	80
32	1	4	40	16	1600	160
33	1	8	40	64	1600	320
34	1	12	40	144	1600	480
35	1	16	40	256	1600	640
36	1	20	40	400	1600	800
37	1	24	40	576	1600	960
38	1	26	40	676	1600	1040

Curve-Fitting

Multilinear regression using vector-matrix calculations

	M	N	O	P	Q	R
2	XtX					
3	36	452	675	8548	20925	8475
4	452	8548	8475	183236	262725	160275
5	675	8475	20925	160275	725625	262725
6	8548	183236	160275	4157572	4968525	3435675
7	20925	262725	725625	4968525	26645625	9110625
8	8475	160275	262725	3435675	9110625	4968525

	M	N	O	P	Q	R
9	XtXinv					
10	0.311026	-0.03147	-0.013447	0.000773	0.000166	0.000357
11	-0.03147	0.0066	0.0003568	-0.00022	1.27E-19	-2.8E-05
12	-0.01345	0.000357	0.0019075	-4.3E-20	-3.8E-05	-1.9E-05
13	0.000773	-0.00022	2.272E-19	8.19E-06	-4.6E-21	-1.9E-21
14	0.000166	4.92E-21	-3.84E-05	-9.1E-23	9.53E-07	-2E-22
15	0.000357	-2.8E-05	-1.9E-05	3.76E-21	-5.4E-22	1.52E-06

	M
16	Xty
17	39.30495
18	515.6119
19	733.8584
20	9920.177
21	22717.53
22	9620.406

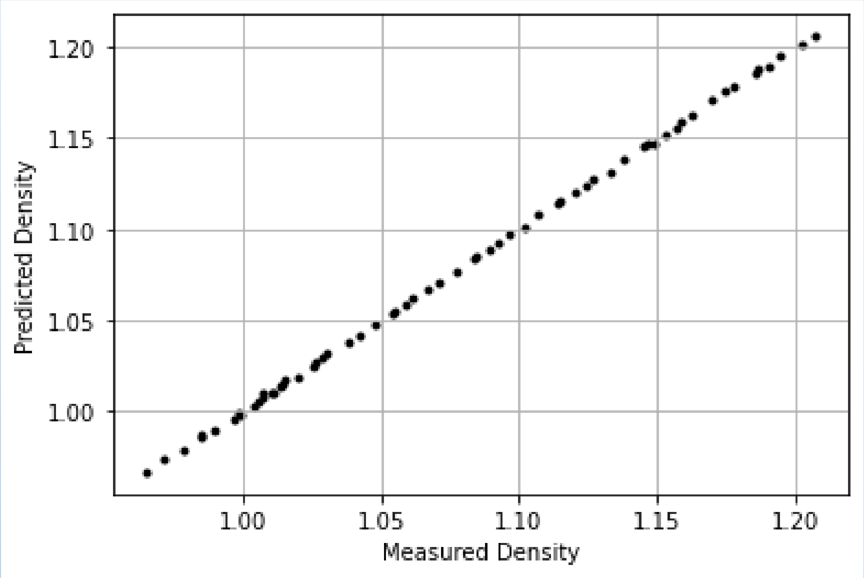
	O
16	b
17	1.0010766
18	0.0072741
19	-9.780E-05
20	2.496E-05
21	-2.997E-06
22	-1.254E-05

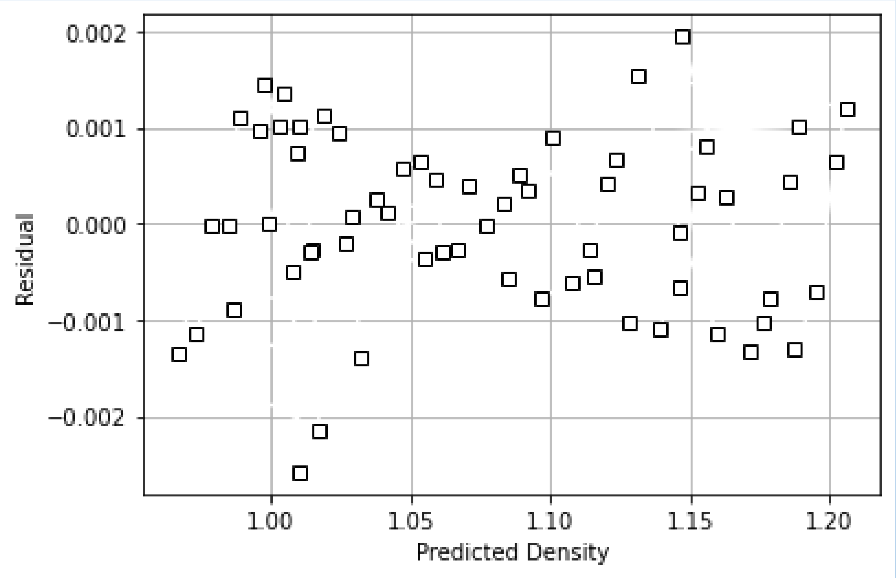
$$\rho = 1.00108 + 0.0072741w - 9.780\times10^{-5}T$$
$$+2.496\times10^{-5}w^2 - 2.997\times10^{-6}T^2 - 1.254\times10^{-5}wT$$

NaClDensityRegressionFinished.xlsx

Curve-Fitting

Multilinear regression using vector-matrix calculations





Very close to perfect agreement

Model appears to be adequate

Curve-Fitting

Nonlinear regression

Model: $y = f\left(\mathbf{x}, \boldsymbol{\beta}\right)$ Dataset: $\left\{y_i, x_{1i}, \ldots, x_{mi}, i = 1, \ldots, n\right\}$

$$\mathbf{f}\left(\mathbf{x}, \boldsymbol{\beta}\right) = \begin{bmatrix} f\left(\mathbf{x}_1\right) \\ f\left(\mathbf{x}_2\right) \\ \vdots \\ f\left(\mathbf{x}_n\right) \end{bmatrix}$$

$\mathbf{e} = \mathbf{y} - \mathbf{f}\left(\mathbf{x}, \hat{\boldsymbol{\beta}}\right)$ $\min_{\hat{\boldsymbol{\beta}}} \mathbf{e}^T \mathbf{e}$ using an optimization routine

Curve-Fitting Nonlinear regression

Example: fitting the Antoine equation to vapor pressure data

AntoineStarter.xlsx

$$\log_{10} P_V = A - \frac{B}{C+T}$$

Vapor Pressure of 95%(wt) Sulfuric Acid Aqueous Solution

Temperature (degC)	Vapor Pressure (torr)
35	0.0015
40	0.00235
45	0.0037
50	0.0058
55	0.00877
60	0.0133
65	0.0196
70	0.0288
75	0.0415
80	0.0606
85	0.0879
90	0.123
95	0.172
100	0.237
105	0.321
110	0.437

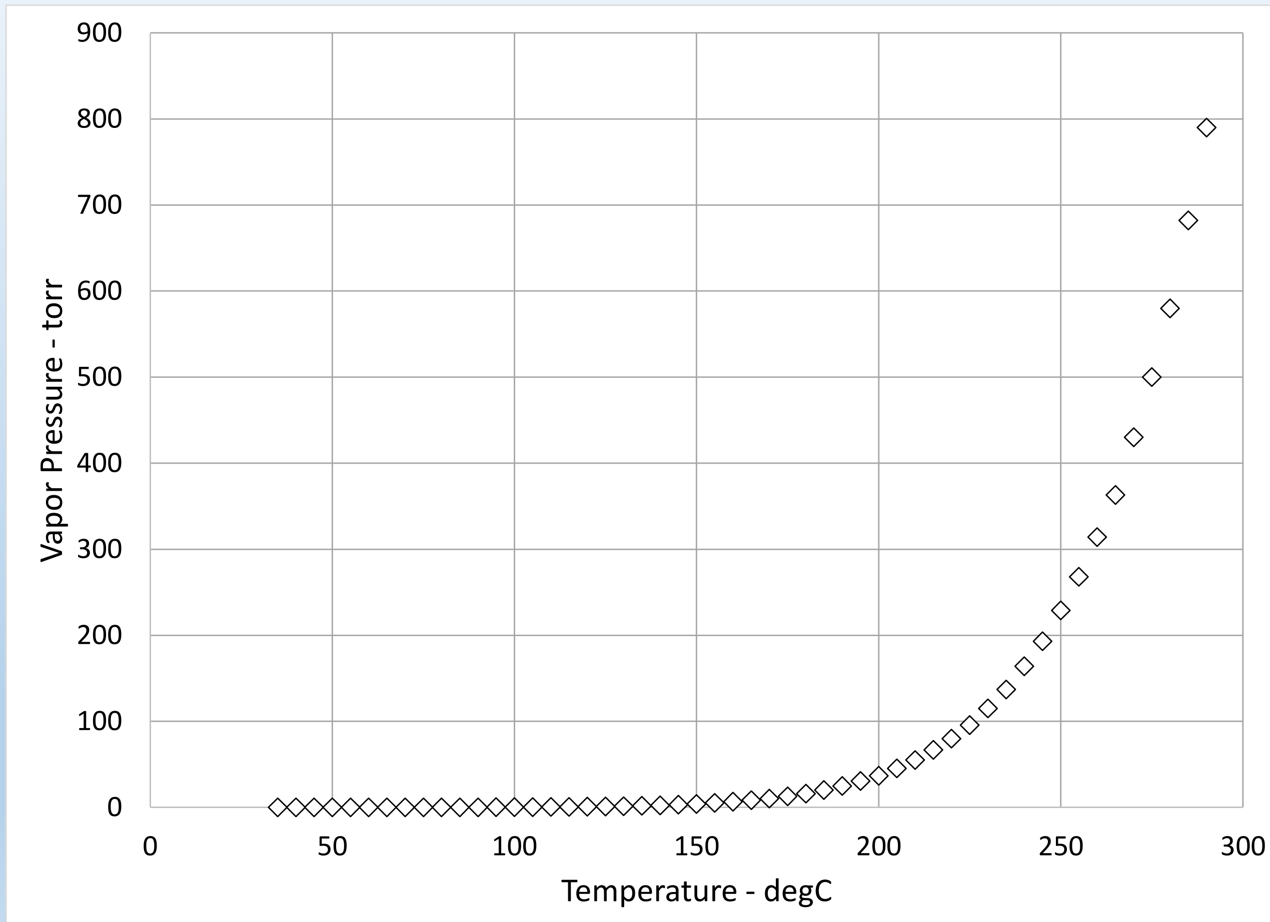
115	0.59
120	0.788
125	1.07
130	1.42
135	1.87
140	2.4
145	3.11
150	4.02
155	5.13
160	6.47
165	8.39
170	10.3
175	12.9
180	15.9
185	20.2
190	24.8
195	30.7
200	36.7

205	45.3
210	55
215	66.9
220	79.8
225	95.5
230	115
235	137
240	164
245	193
250	229
255	268
260	314
265	363
270	430
275	500
280	580
285	682
290	790

Curve-Fitting Nonlinear regression

Example: fitting the Antoine equation to vapor pressure data

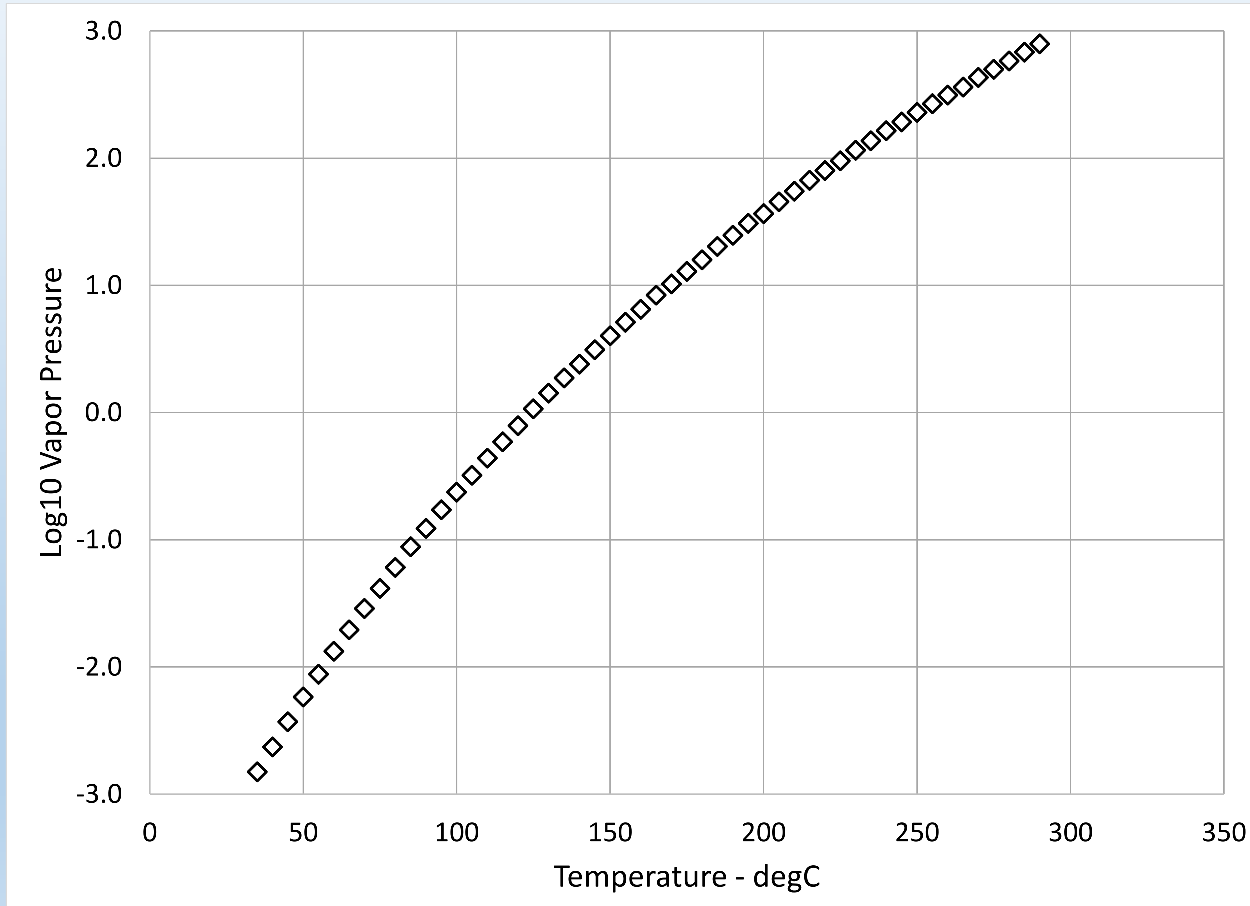
	A	B
2	Temperature (degC)	Vapor Pressure (torr)
3	35	0.0015
4	40	0.00235
5	45	0.0037
6	50	0.0058
7	55	0.00877
8	60	0.0133
9	65	0.0196
10	70	0.0288
●	●	●
47	255	268
48	260	314
49	265	363
50	270	430
51	275	500
52	280	580
53	285	682
54	290	790



Curve-Fitting Nonlinear regression

Example: fitting the Antoine equation to vapor pressure data

	A	B	C
2	Temperature (degC)	Vapor Pressure (torr)	Log10 Vapor Pressure
3	35	0.0015	-2.824
4	40	0.00235	-2.629
5	45	0.0037	-2.432
6	50	0.0058	-2.237
7	55	0.00877	-2.057
8	60	0.0133	-1.876
9	65	0.0196	-1.708
10	70	0.0288	-1.541
●	●	●	●
[illegible]	[illegible]	[illegible]	[illegible]
49	265	363	2.560
50	270	430	2.633
51	275	500	2.699
52	280	580	2.763
53	285	682	2.834
54	290	790	2.898

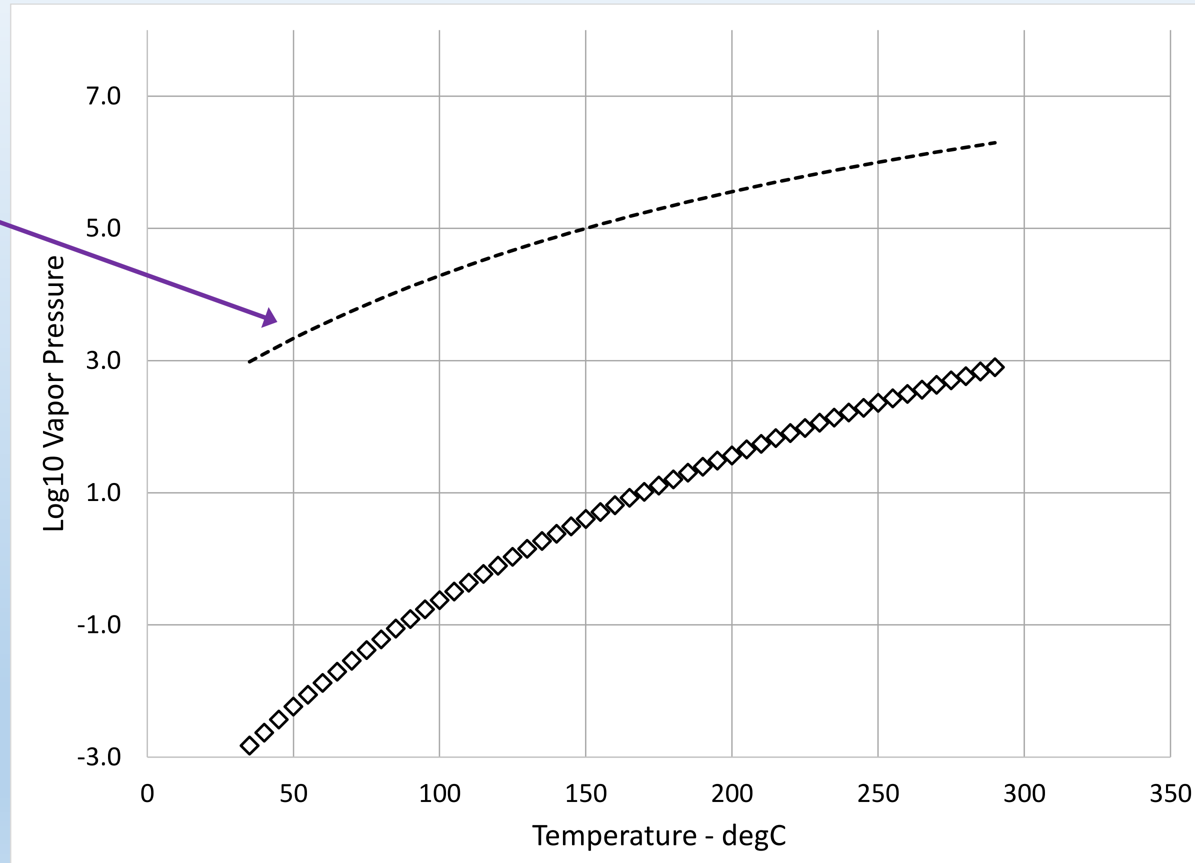


Curve-Fitting Nonlinear regression

Example: fitting the Antoine equation to vapor pressure data

	A	B	C	D
2	Temperature (degC)	Vapor Pressure (torr)	Log10 Vapor Pressure	Predicted Log10 VP
3	35	0.0015	-2.824	2.982
4	40	0.00235	-2.629	3.103
5	45	0.0037	-2.432	3.220
6	50	0.0058	-2.237	3.333
7	55	0.00877	-2.057	3.443
8	60	0.0133	-1.876	3.548
9	[illegible]	[illegible]	[illegible]	[illegible]

A	10
B	2000
Cc	250



Curve-Fitting Nonlinear regression

Example: fitting the Antoine equation to vapor pressure data

=C3-D3

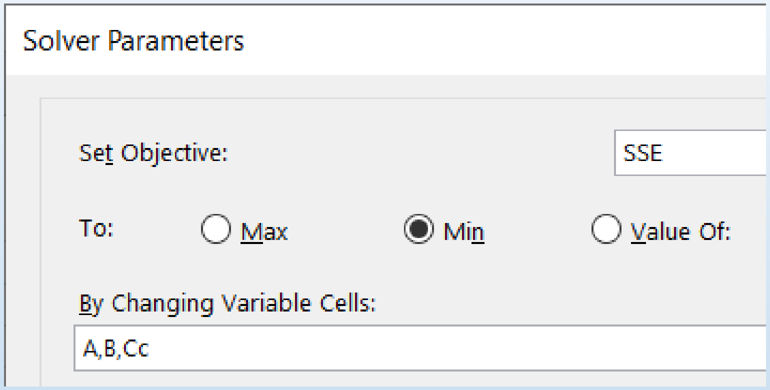
C	D	E
Log10 Vapor Pressure	Predicted Log10 VP	Error
-2.824	2.982	-5.806

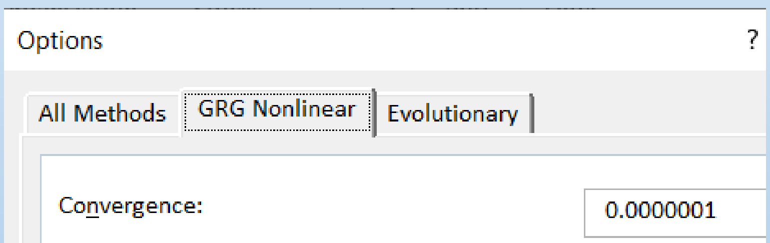
	A	B	C	D	E	F	G	H
2	Temperature (degC)	Vapor Pressure (torr)	Log10 Vapor Pressure	Predicted Log10 VP	Error			
3	35	0.0015	-2.824	2.982	-5.806		A	10
4	40	0.00235	-2.629	3.103	-5.732		B	2000
5	45	0.0037	-2.432	3.220	-5.652		Cc	250
6	50	0.0058	-2.237	3.333	-5.570			
7	55	0.00877	-2.057	3.443	-5.500		SSE	1029.227
8	60	0.0133	-1.876	3.548	-5.425			

SSE =SUMSQ(E3:E54)

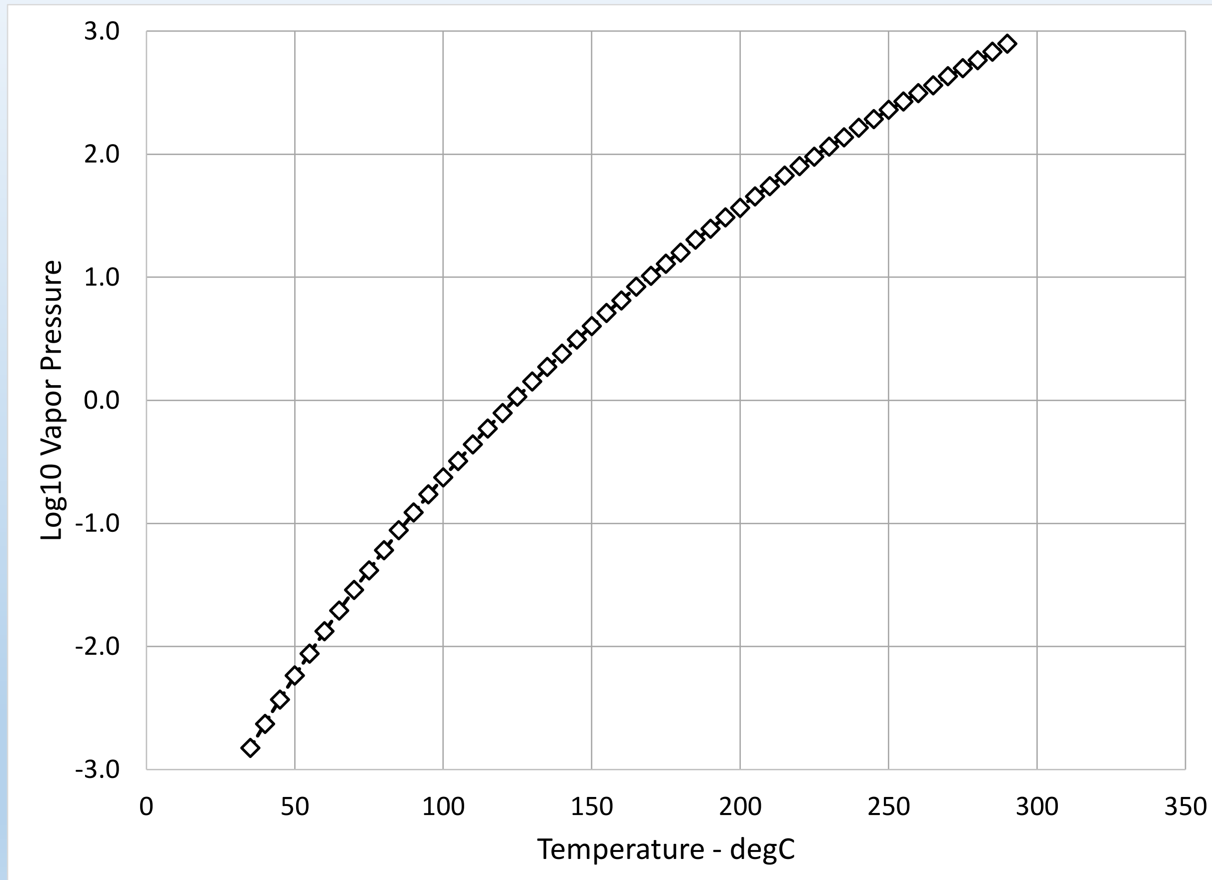
Curve-Fitting Nonlinear regression **AntoineFinished.xlsx**

Example: fitting the Antoine equation to vapor pressure data





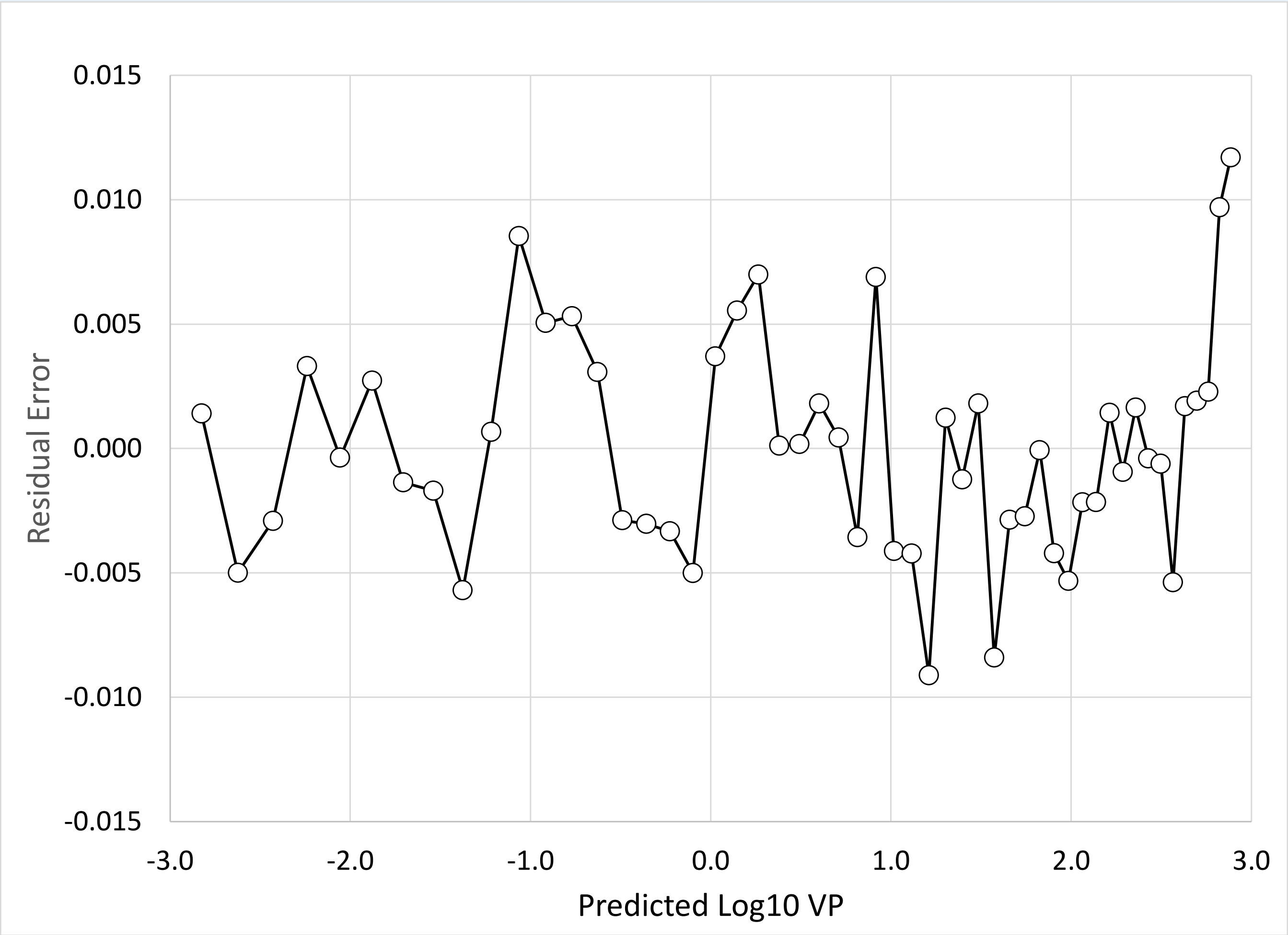
A	9.789
B	3887.5
Cc	273.19
SSE	0.0010



Curve coincides with data

Curve-Fitting Nonlinear regression

Example: fitting the Antoine equation to vapor pressure data



No apparent pattern.
Model is adequate.

Reference:

Spreadsheet Problem Solving and Programming for Engineers and Scientists,
David E. Clough and Steven C. Chapra,
CRC Press - Taylor & Francis Group, 2024.

"Prof. Clough, may I be excused? My brain is full."

What's next?

Excel Bootcamps 1, 2, 3 and 4

- ✓ 1: Getting up to speed with Excel
- ✓ 2: Introducing VBA
- ✓ 3: Learning to use Excel to solve typical problem scenarios
- 4: Detailed modeling of packed-bed and plug-flow reactors