MATLAB Tutorial on ordinary differential equation solver (Example 12-1)

Solve the following differential equation for co-current heat exchange case and plot *X*, *Xe*, *T*, *Ta*, and *-rA* down the length of the reactor (*Refer LEP 12-1*, *Elements of chemical reaction engineering*, *5th edition*)

Differential equations

d(Ta)/d(V) = Ua*(T-Ta)/m/Cpcd(X)/d(V) = -ra/Fa0

d(T)/d(V) = ((ra*dH)-Ua*(T-Ta))/Cpo/Fa0

Explicit equations

Cpc = 28m = 500 Ua = 5000 Ca0 = 1.86 Fa0 = 14.67 dH = -34500 k = 31.1*exp((7906)*(T-360)/(T*360)) Kc = 3.03*exp((dH/8.314)*((T-333)/(T*333))) Xe = Kc/(1+Kc) ra = -k*Ca0*(1-(1+1/Kc)*X) Cpo = 159

Initial and final values

Ta(0) = 315T(0)= 305 X(0)=0 V(0)=0 V(f)= 5

Repeat the similar exercise for other cases:

- Counter-current heat exchange: Plot X, Xe, T, Ta, and -rA down the length of the reactor
- Constant ambient temperature, Ta: Plot X, Xe, T, and -rA down the length of the reactor
- Adiabatic operation: Plot X, Xe, T, Ta, and -rA, down the length of the reactor

Step 1: First, launch MATLAB which you can access on CAEN computers (University of Michigan computer) or download from <u>http://www.mathworks.com</u>. You will see that multiple window opens which looks like this

📣 MATLAB R2014	ь		and the state of t				0	x
HOME	PLOTS	APPS		LASE) 🕐 Se	arch Document	ation	P -
New New Op Script Part	Find Files	Import Save Data Workspace	New Variable	Analyze Code	Simulink Library		RESOURCES	
20	C: • Users	Foglerlab + Doc	uments + MATLAB					+ ,o
Current Folder	۲	Command Wine	dow					0
Details	^							
Workspace	۲	2						
Name A	Value							
•								
Ready								

The central window is called Command window. In the command window you can enter statements, run your files, generate output etc.

Step 2: Change the current folder location

The only folder which MATLAB can access is shown by red circle. In this case the folder name is "MATLAB". You must change the access location so that it refers to your particular folder which you are using for this project. Let's create a folder LEP-12-1 on desktop. Now change the MATLAB folder location to this location as shown below

MATLAB R2014b		×
HOME PLOTS	APPS 🛃 🔓 🔓 🚖 🔁 😨 Search Documentation 🔎	▲
Rew New Open Compare Script FillE	A Open Variable Variable A Bun and Time	
< 🔶 🔁 🔀 🌗 🗲 C: 🕨 Use	ers 🕨 foglerlab 🕨 Desktop 🕨 LEP-12-1 🔹 👻	٩
Current Folder 💿	Command Window	۲
Name ▼	fz >>	
Details ^		
Workspace 💿		
Name A Value		
IIII A Ready		

To solve ODE in MATLAB, you need to create two kind of program files:

- 1. Script file where you enter data such as integration span, initial guess, produce graphical outputs,etc
- 2. Function file where you enter all your explicit and differential equations

We will first create function file

<u>Creating function file</u> Step 3: On the toolbar, Click on the New menu and select Function

HOME	PLOTS	APPS				2 Searc	h Documentati	on 🔎
iew icript	Open Compare	Import Si Data Work	Ave space Clear W VARIABLE		Analyze Code	Simulink Library		RESOURCES
*	Script Ctrl+N	foglerlab	Desktop + LEP-	12-1				
urret fx	Function	mand Win	idow					
	Example	>>						
	Example Class	>>						
*	Class							
tails	Class							
6	Class System Object							

You will see a new window opens that looks like this. MATLAB automatically creates syntax for writing function file. To use solver in MATLAB, you need to write codes in the space provided.

MATLAB R2014b	T	×		-	-	- = -	-	-			53
HOME PLOTS	APPS EDITO	OR PUBLISH	VIEW		de	<u><u><u></u></u></u>	¢ E	🕽 🕐 Sea	arch Documentat	ion	<mark></mark>
New Open Save	Compare ▼ Print ▼	Go To V Find V NAVIGATE	Comment Indent	F× F₄ % ‰ ‰ ₩ F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F		Breakpoints	Run	Run and Advance	Run Section	Run and Time	
🗢 🔶 🔁 🔀 🚺	C: ► Users ► 1		sktop 🕨 LEI	P-12-1	_						- P
Current Folder Name Details		ditor - Untitled*	ED Summa		is f	unction g)		• x
Workspace Name A		imand Window									•
IIII - Ready					U	ntitled			Ln	8 Co	I 1

The first line of function starts with the keyword function followed by the output arguments. The right side contains function name (Untitled) and its input arguments. In this tutorial, we have chosen the function name as ODEfun which takes two input arguments i.e. V and Y. The first input argument "V" refers to integration span i.e. initial and final value of volume of reactor (in this case, Vinit=0, Vfinal=5).

Second input argument "Y" refers to initial values of the dependent variable i.e Ta, T, and X (in this case, Ta(0)=315, T(0)=305, and X(0)=0). The values of V and Y will be defined in the script file and then passed to the function file.

Step 4: SAVE your file.

Let's name the function file as ODEfun. MATLAB file is saved with extension ".m". In this case, your function file is saved with the name "ODEfun.m"

📣 Select File for Save	As			×
	P-12-1	-	✓ 4 Search LEP	-12-1 👂
Organize 🔻 Ne	w folder			:== 🗸 🔞
☆ Favorites	▲ Name	*	Date mo	dified Type
📃 Desktop 🚺 Downloads		No item	is match your search.	
🗐 Recent Places				
🥞 Libraries	E			
Documents				
J Music				
Pictures				
Videos 🛃				
🖳 Computer				
🏝 OSDISK (C:)	-			•
File name:	ODEfun			•
Save as type:	MATLAB Code files	(*.m)		•
Aide Folders			Save	Cancel

Step 5: Define function output arguments by f. The syntax for creating function file in our case becomes

Function f=ODEfun(V, Y)

Where V, Y are local to function.

Note that f, V, and Y are just variables. You can use whatever terms you like. Just remember that if you have defined a variable, then you have to always refer to it by the same name.

Now, edit the inbuilt format of function file for your case as shown below

📣 MATLAB R2014b		23
HOME PLOTS APPS	EDITOR PUBLISH VEW 🛃 🖷 🔏 🛍 🛱 🗇 🗭 🛱 😯 Search Documentation	<mark></mark>
New Open Save FILE FILE FILE Compa Print FILE FILE FILE Compa Print FILE FILE Compa Print FILE Compa FILE FILE Compa FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE FILE	re V Go To V Comment % % % Breakpoints Run Run and Advance Run and	→ <u>0</u>
Current Folder 💿	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m ODEfun.m +	▼ ×
Details	1 Function $f = ODEfun(V, V)$ 2 3 4 5 - end 6 7	
Name A Value		
4 <u> </u>	Command Window fx >>	•
Ready	ODEfun Ln 2 Col	1

Step 6: As the differential equation contains 3 dependent variables (Ta, T, & X), so Y vector contains initial values of these 3 variables, where, Ta is the first element, T is the second element and X is the third element of Y vector

So,

Ta=Y(1), T=Y(2) and X=Y(3)

Assign the initial values of these variable as shown below. Put a semi-colon after each line to prevent the value from being displayed on the command window each time you run the file

MATLAB R2014b	_	-	Sector State	
HOME PLOTS APPS	EDITOR	PUBLISH VIEW		Search Documentation 🛛 👂 🛣
New Open Save	e 👻 🖓 Go To 👻	Insert 5 fx 14 Comment % % % % Indert 5 45 15 EDIT	BREAKPOINTS BREAKPOINT BREAKPOINT BREAKPOINT BREAKPOINT BREAKPOINT	n Section vance Run and Time
<ri> Image: Second state Image: Second state</ri>	rs ▶ foglerlab ▶ De	esktop 🕨 LEP-12-1		٩ -
Current Folder 💿	📝 Editor - C:\Users	\foglerlab\Desktop\LEP-1	2-1\ODEfun.m	⊙×
D Name ODEfun.m	ODEfun.m M 1 - function 2 - Ta=Y(1) 3 - T=Y(2) 4 - X=Y(3) 5 - end 6	on <u>f</u> = ODEfun(<mark>V</mark> ,Y); ;		
Details ^	7			
Workspace 💿				
Name 📥 Value				
4 III +	Command Window	1		\odot
Ready			ODEfun	Ln 6 Col 1 🔡

Step 7: Before entering all the explicit equations, we will first write comments (which are not executed). To write comments, use percent symbol (%) followed by the comment. By default MATLAB uses green colour for writing comments. Let's put the comment "% Explicit equations"

📣 MATLAB R2014b		
HOME PLOTS APPS	EDITOR PUBLISH VIEW 🛃 🔚 🔏 🛱 🛱 🥱 🗟 🚍 🕐 Search Documentation	🔺 🔍
New Open Save	Insert fx fx	
<ri> 🔁 🔀 🌗 🕨 C: 🕨 Users 🕨</ri>	foglerlab > Desktop > LEP-12-1	- ₽
Current Folder 💿 📝	Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m	⊙×
Name ▼	ODEfun.m × + function f = ODEfun (V,Y) Ta=Y(1); T=Y(2); X=Y(3); * Explicit Equations end	-
Details ^ 7		
Workspace 💿 8		
Name A Value		
	mmand Window	
Ready	ODEfun Ln 5	Col 21 .::

📣 MATLAB R2014b		
HOME PLOTS APP	25 🛛 EDITOR PUBLISH 🔹 VIEW 🛛 🛃 🔚 🔏 🛍 🗐 定 🚍 🕐 Search Documentation	🔺 🔍
New Open Save	re 👻 🖓 Go To 👻 Comment % 🏂 🔅 Breakpoints Run Run and 🔍 Advance Run and	
< 🔶 🔁 🔀 🌗 K.: 🕨 Use	ers ▶ foglerlab ▶ Desktop ▶ LEP-12-1	م -
Current Folder 💿	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m	⊙×
Details Workspace Name Value	<pre>1</pre>	
۰ III +		:
Ready	ODEfun Ln 12	Col 10 .::

Step 8: Now, enter all the explicit equations with semi colon at end

Step 9: Next, you need to enter your differential equations. For this example, you have three differential equation in Ta, T and X.

In MATLAB, LHS of differential equations cannot be entered in derivative form (dy/dx), so you need to define variable representing left side of differential equation

In this case we will use the following definition for differential equation

dTa/dV=dTadV, dT/dV=dTdV, and

dX/dV=dXdV

Enter the comment for differential equation and then enter your differential equations. After all the equations are entered, you need to define the output f. In the function file, f contains the differential equation. So, define f as shown below. ODEfun must return column vectors, so, you need to put semicolon between differential equations to get column vector for different dependent variable.

MATLAB R2014b		• X
HOME PLOTS	APPS EDITOR PUBLISH VIEW 💽 🔚 🔏 🛱 🛱 🗇 🗭 🔁 🕐 Search Documentation	٩
New Open Save	e 🗸 🎲 Go To 👻 Comment % 🍇 🚱 Breakpoints Run Run and 🛃 Advance Run and	
	rs + foglerlab + Desktop + LEP-12-1	.
Current Folder 💿	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m ODEfun.m + +	⊙
 Temp_profile.m Rate_profile.m ODEfun.m Conversion_profile.m 	<pre>1</pre>	
etails Vorkspace Vame Ans (2;3;4) ans (2;3;4) ans (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;3;4) (2;	$\begin{array}{llllllllllllllllllllllllllllllllllll$	E
Vspan [0,5]	23	-
Xe 1x97 dc y 97x3 dc y0 [315,30 ~	Command Window fs >>	
	ODEfun Ln 18	Col 32

The function file returns the value in the form $[v \ y]$ where, v is a column vector $\begin{bmatrix} V_1 \\ V_2 \\ V_n \end{bmatrix}$ of independent variable (i.e. volume for this case) and y is a matrix $\begin{bmatrix} Ta1 & T1 & X1 \\ Ta2 & T2 & X2 \\ Tan & T2n & Xn \end{bmatrix}$ of dependent variable (i.e. Ta, T, & X for this case). Note that no of rows are same in vector v and matrix y. The return value of the formation will be made in the parity file matrix be made by dimensional parameters.

function will be used in the script file which would be discussed in next section

Creating a script file

Step 10: Go back to New menu and select Script

MATLAB R20146		- 0
HOME PLOTS APP	EDITOR PUBLISH VEW	ρ 🖛
New Open Save Print -	a • Age of the section of t	
Script Ctrl+N	NAVIGATE EDIT BREAKPOINTS RUN rs + foglerlab + Desktop + LEP-12-1	- p
f_X Function \odot	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m	⊙×
Class Class System Object > Details ^ Workspace © Name ~ Value	ODEfunm × + 1 function f = ODEfun(V,Y) 2 - Ta=Y(1); 3 = T=Y(2); 4 - X=Y(3); 5 * Explicit Equations 6 - Cac = 1.96; 7 - CPo = 159; 8 - dH = -34500; 9 - FAc = 14.67; 10 - UA = 5000;	н II I
۲ m))	Command Window	۲
IIII - Ready	ODEfun In 22	Col 2

A black window will appear like this

📣 MATU	AB R201	.4b	-		· . · ·			ſ				-			
HOME	F	LOTS	APP	s	EDITOR	PUBLISH	VIEW	K	B H 4	h h	90	E ? Search	Documentai	tion	🔺 🔍
New C	-	ave ,	G Find File E Compar Print ▼	• •	Go To Go To Find NAVIGATE	Comment		•	Breakpoints	Run	Run and Advance		Run and Time		
) •			foglerlab 🕨 l	· · ·	P-12-1								-
Current F					Editor - Untitle	-									⊙ ×
	lame ▼ DEfun.m	1		1	ODEfun.m	Untitled	• × + (
Details			^												
Workspa	ce		✐												
Name 🔺			Value	Cor	mmand Windo	w									•
IIII + Rea	ady								sc	ript			Ln	1	Col 1

Step 11: First we will create the codes for Temperature profile. So save your script file with the name "Temp_profile". You can also save it with other name as per your wish. We will save this file in the folder LEP-12-1 as shown

📣 Select File for Save A	5	×
	12-1 • • • S	Search LEP-12-1
Organize 🔻 New	folder	:= - 🔞
☆ Favorites	- Name	Date modified Type
🧮 Desktop	1 ODEfun	4/24/2017 3:23 PM MATLA
🐌 Downloads	-	
📃 Recent Places	E	
 ☐ Libraries ☐ Documents → Music ☐ Pictures ☐ Videos 	 ✓ <	•
File name:	Temp_profile	•
Save as type:	MATLAB Code files (*.m)	•
Alide Folders		Save Cancel

Step 12: In the blank space, Enter clc in the first line. It will clear all the input and output from the Command Window display, giving you a "clean screen"

Next you need to enter the integration time span. In this case we want to integrate the volume of reactor from V=0 to V=5. Let's define the integration time span variable as Vspan. To enter this in a row vector format, type "Vspan = [0 5]" with space between 0 & 5 else enter Vspan= [0; 5] to create a column vector. You can either create row or column vector, the output will remain same for this case. We will create a row vector.

Next you need to enter the initial values of the dependent variable, Ta, T, X i.e. Ta (0) =315, T (0) =305, and X (0) =0

Enter the initial value of the dependent variable in the vector form

y0= [315 305 0]

Again putting semi-colon at the end of each statement prevents the value from being displayed on the command window. We will also put comment against each line as shown below

📣 MATLAB F	R2014b					-						x
НОМЕ	PLOTS	APPS	EDITOR	PUBLISH			h h	50	🔁 🕐 Search	Documentati	ion	<mark>></mark> 🔺
New Open	Save	Q Find Files Compare Print ▼	🔹 🔹 🖓 Go To 👻	Comment g	<pre> fx</pre>	Breakpoints	Run	Run and Advance	Run Section	n 🕑 Run and Time		
🔶 🔶 🔁 🎗	ء 🌒 🖾	C: 🕨 User	rs ► foglerlab ► D	esktop 🕨 LEP	-12-1							- 2
Current Fold	er	۲	📝 Editor - C:\User	s\foglerlab\De	sktop\LEP-12-	1\Temp_profile.	m					⊙×
🖄 ODEfu	n.m		-		-	or the indep al values :						
Details		^										
Workspace		$\overline{\mathbf{v}}$										
Name 🔺		Value										
						11	1					•
۰ III		4	Command Window	v								
A Ready						sc	ript			Ln	4 Co	ol 1 🧠

Step 13: Next, you need to choose your ODE solver. There are different kind of solver available in MATLAB which you can use as per your problem requirement. The following is the list of all the solver with details:

Solver	Proble m Type	Order of Accuracy	Method	When to Use
ode45	Nonstiff	Medium	Explicit Runge-Kutta	Most of the time. This should be the first solver you try.
ode23	Nonstiff	Low	Explicit Runge-Kutta ,pair of Bogacki and Shampine	For problems with crude error tolerances or for solving moderately stiff problems.
ode113	Nonstiff	Low to high	Adams-Bashforth- Moulton PECE	For problems with stringent error tolerances or for solving computationally intensive problems
ode15s	Stiff	Low to medium	Numerical differentiation formulas (NDFs)	If ode45 is slow because the problem is stiff.
ode23s	Stiff	Low	Modified Rosenbrock	If using crude error tolerances to solve stiff systems and the mass matrix is constant.
ode23t	Moderately Stiff	Low	Trapezoidal rule using a "free" interpolant.	For moderately stiff problems if you need a solution without numerical damping
ode23tb	Stiff	Low	Implementation of TR- BDF2, an implicit Runge-Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a backward differentiation formula of order two	If using crude error tolerances to solve stiff systems.

The first choice for solving differential equation should be Ode45 as it performs well with most ODE problems. Hence, we will use ode45 solver. To use ODE solver, MATLAB uses following **Syntax**

[v y] = solver (@ODEfun, Vspan, y0)

Where ODEfun is the function file which you have created. The function file name must be same as that is invoked/called from the script file. Vspan is a vector specifying the interval of integration, and y0 is a vector of initial conditions

Step 14: Write down the solver equation in the required format as shown below. In the script file, we call/invoke function file and pass input arguments to function file. In this case input arguments are Vspan and y0.

MATLAB R2014b							
HOME PLOTS APPS	EDITOR	PUBLISH VIEW		h h 9 ¢	🗄 🕐 Search D	ocumentation	🔺 🔍
New Open Save FillE FillE Compare Print ~ FillE Compare Print ~ FillE Compare Print ~ FillE		Insert 🛃 fx 🦌 Comment % ‰ % Indent 🖡 📲 🚱 EDIT	Breakpoints	Run Run and Advance	Run Section	Run and Time	<u>م</u> -
		esktop ▼ LEP-12-1 s\foglerlab\Desktop\LEP-	12-1\Temp_profile				• ⊅ ⊙ ×
ODEfun.m	3 - y0 = [:	Temp_profile.m × = [0 5]; % Range 315 305 0]; % Ini ode45(@ODEfun,Vag	for the inde	-			
Details Workspace Value Value							
	•						•
۰ III >	Command Window	/					\odot
IIII A Ready			S	cript		Ln 4	Col 31 .::

When you run the script file, it will call function file and evaluate the differential equation for different values of independent variable and the output will be stored in [v y]. As described earlier, v is a column vector of volume and y is a column vector of [T, Ta, X]. We will not run the script file at this moment.

In this tutorial, you will learn to plot temperature profile, conversion profile and rate profile along the volume of the reactor.

a) **Temperature profile**

Step 15: The 'y' output of the function file contains value of Ta, T, and X where the value of Ta, T and X are in first, second column and third column respectively, so the values of these variables can be obtained as

Ta = y(:,1); T = y(:,2); X = y(:,3) where, Ta, T and X are column vector

To plot Ta and T along the volume of the reactor, we will use MATLAB plot function.

The syntax for using plot function is

plot(X1,Y1,...,Xn,Yn)

Where X1, Y1 is the first set of data point. Similarly Xn, Yn is the nth set of data point. You can put multiple graph on the same plot by using comma between two data sets (X1, Y1) and (X2, Y2)

So to plot Ta vs v, the syntax is plot (v, y(:,1))This will take the values of v (column vector) and 1^{st} column vector of y. To plot T vs v on the same graph, the syntax becomes

Plot (v, y(:1),v, y(:,2))

This will plot Ta and T on Y axis and v on the X axis. You may or may not put semi-colon at the end of Plot statement as this gives only graph and does not return any value on command window

We can also put legend, axis label, title, and range to your plot. The syntax are:

1) For legend : legend('comments', 'comments', 'comments')

You can put your legend name under inverted comma. To put legend to different graphs, use comma between different legend names. By default, the order of the legend is same as the order of the graph defined in plot function

- 2) y axis label :ylabel('comments')
- 3) x axis label : xlabel('comments')
- 4) title: title('comments')
- 5) range: axis([a b c d]), where a, b refers to the range of X axis and c, d refers to the range of Y axis

The word in green can be replaced with the word you want to be displayed in your graph. In this case, there are two graphs: 1^{st} is for Ta and 2^{nd} for T. On X axis you want volume V (m3) and temperature T (K) on Y axis. The graph is made for co-current case, so accordingly define all the graphical features to your plot.

Enter the above codes as shown below

🔥 MATLAB R2014b		
HOME PLOTS APPS	EDITOR PUBLISH VEW	tion 🔎 🗖
New Open Save	Insert Insert fx Fx </td <td></td>	
	foglerlab + Desktop + LEP-12-1	- p
Current Folder	Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Temp_profile.m	⊙×
🗋 Name 🕶	ODEfun.m × Temp_profile.m × +	
	2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume o	
Details ^ Workspace (*)	<pre>3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - plot(v,y(:,1),v,y(:,2)); 6 - legend('T_a','T') 7 - ylabel('T(K)'); 8 - xlabel('V(m^3)'); 9 - axis([0 5 300 380]);</pre>	,T, and X
Details ^ Workspace ③ Name * Value 1 y 97x1 dou 1	<pre>3 - y0 = [315 305 0]; * Initial values for the dependent variables Ta 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - plot(v,y(:,1),v,y(:,2)); 6 - legend('T_a','T') 7 - ylabel('T_(K')'; 8 - xlabel('V(m^3)'); 9 - axis([0 5 300 380]); 0 - title('Temp profile down the reactor for co-current heat exchange 1 2 3 </pre>	,T, and X
Details ^ Workspace • Name + Value 1 V 97x1 dou 1 V Vspan [0,5] y 97x3 dou 1	<pre>3 - y0 = [315 305 0]; * Initial values for the dependent variables Ta 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - plot(v,y(:,1),v,y(:,2)); 6 - legend('T_a','T') 7 - ylabel('T_(K')'; 8 - xlabel('V(m^3)'); 9 - axis([0 5 300 380]); 0 - title('Temp profile down the reactor for co-current heat exchange 1 2 3 </pre>	,T, and X
Details ^ Workspace • Name * Value V 97x1 dou Vspan [0,5] y 97x3 dou y 97x3 dou 1 1 1 1 1 1 1 1 1 1 1 1 1	<pre>3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta 4 - [v y]=ode45(@ODEfun,Vepan,y0); 5 - plot(v,y(:,1),v,y(:,2)); 6 - legend('T_a','T') 7 - ylabel('T(K)'); 8 - xlabel('V(m^3)'); 9 - axis([0 5 300 380]); 10 - title('Temp profile down the reactor for co-current heat exchange 11 22 33 34 44 </pre>	,T, and X

Now you have created both the script file and function file. You need to run only script file as the function file will be automatically called from the script file. Don't run function file as it will give error. This is because function file takes input arguments which are defined in the script file. Make sure that the function file is present in the same location as that of script file, else MATLAB won't be able to find the function file when you invoke the function file from script file.

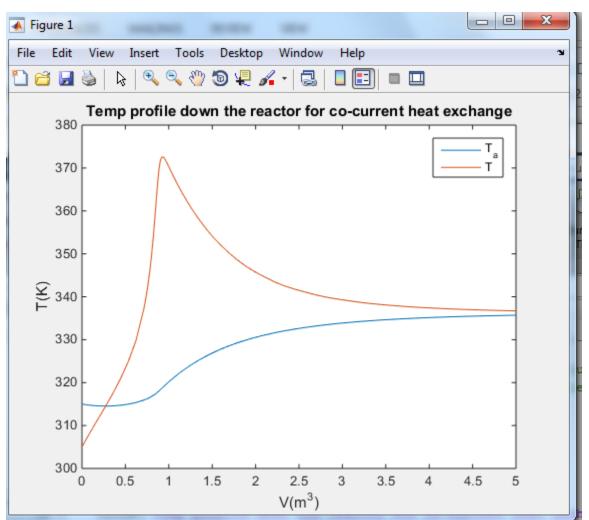
In this case, we have put both the files under the folder "LEP-12-1"

On the left side window, you will find that under current folder (shown by red rectangular box), both the files are present. If it is not so, then change the current folder location (shown by blue box) to the folder containing your files.

Step 16: In the Temp_profile file, press the run button screenshot. Alternatively, you can also run your script file by using the command window. In the command window type "Temp_profile and" press enter as shown below

HOME PLOTS APPS EDITOR PUBLISH VEW Image: Constraint on the second s	∡ م
New Open Save Compare Indext Comment % % % Breakpoints Run Run and Advance Run and Indext	
	- <u></u>
Current Folder 💿 🛛 Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Temp_profile.m	
Name ODEfun.m × Temp_profile.m × Temp profile.m 1 - clc	
<pre> ODEfun.m 2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the 3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta,T,and 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - plot(v,y(:,1),v,y(:,2)); 6 - legend('T_a','T') 7 - ylabel('T_a','T') 7 - xlabel('V(m^3)'); 8 - xlabel('V(m^3)'); 9 - axis((0 5 300 380)); </pre>	
Details A I Workspace 0 11	
Name Value	• •
v 97x1 dou Vspan [0,5] y 97x3 dou y0 [315,305, ✓	\odot

You will see that following graph is generated which gives Ta and T profile down the length of the reactor.



You can also check that axis title, legend, chart title and axis range are as per defined by you in the code.

Next, you need to create conversion and rate profile along the length of the reactor

b) Conversion profile

Step 17: Create a new Script file and save it as "Conversion_profile" in the folder LEP-12-1

In this, you need to plot both actual conversion (X) and equilibrium conversion (Xe) along the length of the reactor.

Step 18: The function ODEfun gives conversion, X = y(:,3) as its output at different values of reactor volume. To get the value of Xe at different reactor volume you need to write few more codes.

The function file for this case will remain same as all the explicit equation and differential equations are going to be same. All you need to do is modify your script file to include the expression for Xe.

The function file will return the value of T at different values of v from which you can calculate value of Kc and Xe at different value of T. To get the total no of elements present in T vector, use MATLAB inbuilt "size" function which returns the size (no of rows and columns) of an array or matrix.

The values of temperature are stored in 1st column of matrix y, so you need to find the size of y

The syntax for using size function is

Z=size (y); this will return the size of matrix y. Suppose size of y is n x m then Z= [n m]

📣 MATLAB R2014b EDITOR **B** Ð ? Search Documentation ۵ HOME PLOTS APPS PUBLISH 5 Find Files <- ⇒ Insert 🛃 fx 🕞 0 0 I_{iili}il \Diamond ÷ H Run Section Compare Go To 👻 Comment % 🏂 🐲 New Open Save Run Run and Run and Breakpoints Advance 🔍 Find 💌 Advance Time 🚔 Print Indent 🛐 🏭 🎼 RI < Implementation 🗘 🕞 🕨 C: 🕨 Users 🕨 foglerlab 🕨 Desktop 🕨 LEP-12-1 Current Folder ۲ Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Conversion_profile.m 🗑 🗙 ODEfun.m 🗙 Temp_profile.m 💥 Conversion_profile.m Name 🔻 × + clc 🕙 Temp_profile.m 1 -DDEfun.m 2 -Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor Conversion_profile.m 3 y0 = [315 305 0]; % Initial values for the dependent variables Ta,T,and X 4 -[v v]=ode45(@ODEfun,Vspan,v0); 5 z=size(y); Details ~ Command Window ۲ Workspace $f_{\frac{x}{2}} >>$ Name 4 Value 🔣 v 97x1 (🏩 • Þ script Ln 5 Col 1

- p

1

Write the codes as shown below

Step 19: We are only concerned with the row no of y i.e. z (1, 1)

Now Xe=Kc/(1+Kc)

And Kc = $3.03 \exp((-34500/8.314)*((T-333)/(T*333)))$

To evaluate the value of Xe at z (1,1) number of points, we need to create a "for" loop. We will first evaluate the value of Kc at different temperature and then calculate the value of Xe at different Kc. In the equation of Kc and Xe, T can be replaced by y(i, 2) as temp is the second dependent variable of y matrix

Now we will evaluate the value of Xe at i=1: z(1,1)

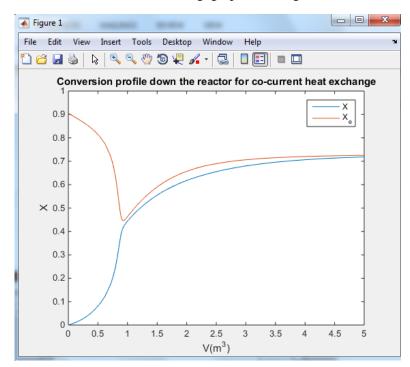
MATLAB R2014b		x
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🖷 🔏 🖆 🛱 🐨 🔄 🖗 Search Documentation	⊼ ۹
Image: Second secon	Insert ➡ fx ➡ x fx ➡ x Fx x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x <td>• 0</td>	• 0
Current Folder 💿	📝 Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Conversion_profile.m	⊙×
Name 🔻	ODEfun.m × Temp_profile.m × Conversion_profile.m × +	
1 Temp_profile.m 2 ODEfun.m 1 Conversion_profile.m	<pre>1 - clc 2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor 3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta,T, and X 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - [for i=1 : z(1,1)</pre>	_
Details ^	7 %Calculating the values of equilibrium conversion from the following expressions	3
Workspace Name Value	8 - Kc(i)=3.03*exp(-4149.63*(y(i,2)-333)/(y(i,2)*333)); 9 - Xe(i)=Kc(i)/(1+Kc(i)); 10 end	-
v 97x1 dc ▲		
Vspan [0,5] y 97x3 dc →	Command Window fx >>	•
	script In 10 Col	4

This will create a column vector of Xe with row no same as that of y matrix

Step 20: Next, you just need to add plot function as was done for case of temperature profile. The output y has X as the third element i.e. X=y(:,3). The vector Xe has been generated just now. So, write down plot function along with the graphical features. It is not necessary to specify axis range always as MATLAB can select the range automatically.

📣 MATLAB R2014b	
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🖪 🔚 🔏 🛍 🛱 🗇 🗭 🗗 🕐 Search Documentation 🔎 🗖
New Open Save Print +	Insert ➡ fx ➡ v fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ fx ➡ v Insert ➡ fx ➡ v Insert ➡ fx ➡ fx ➡ v Insert ➡ fx ➡ fx ➡ v
< 🔶 💽 🔀 🌗 C: 🕨 User:	Foglerlab ► Desktop ► LEP-12-1
Current Folder Name Temp_profile.m ODEfun.m Conversion_profile.m	<pre>Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Conversion_profile.m</pre>
Details	<pre>10 end 11 - plot(v,y(:,3),v,Xe); 12 - legend('X','X_e') 13 - ylabel('X'); 14 - xlabel('V(m^3)'); 15 - title('Conversion profile down the reactor for co-current heat exchange') Command Window (* fx >></pre>
۲	script Ln 15 Col 74

Step 21: Now run the conversion file either by clicking run button or typing "Conversion_profile" and pressing enter in command window. The following graph will be generated



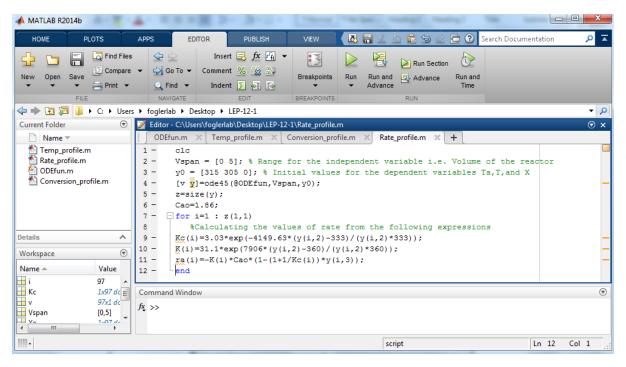
c) Rate profile

Step 22: Create a new Script file and save it as "Rate_profile" in the folder LEP-12-1

Step 23: For plotting rate profile, we need to determine the value of ra at different points. Again you need to edit only your script file. Create a "for" loop and replace T by y(i,2) and X by y(i,3) in the expression of k, Kc, and rate. So the equation for rate becomes:

 $\begin{aligned} &k(i) = 31.1*exp((7906)*(y(i,2)-360)/(y(i,2)*360)) \\ &Kc(i) = 3.03*exp((-34500/8.314)*((y(i,2)-333)/(y(i,2)*333))) \\ &ra(i) = -k(i)*Cao*(1-(1+1/Kc(i))*y(i,3)) \end{aligned}$

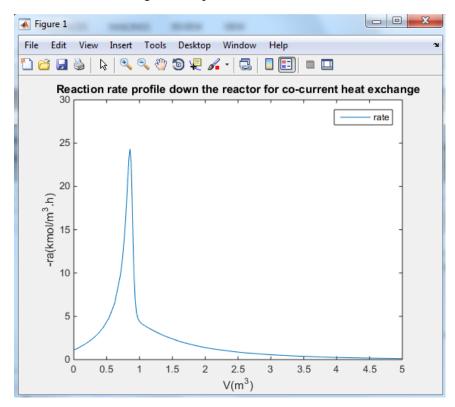
Insert the codes in the file as shown below. This will create a column vector of k, Kc and ra



Step 24: Next, add plot function, label your axis, define the range, legend and title on the graph. If the graph doesn't fit it the axis range you have provided, then go back to file and re-set the axis range. You need to plot rate function which is negative of ra (rate= - ra). So, in the plot function '-ra' is used to plot rate.

📣 MATLAB R2014b		• X
HOME PLOTS	APPS EDITOR PUBLISH VEW 🖪 🖬 🛣 🛱 🛱 🗭 📿 🔁 🕐 Search Documentation	∡ ۹
New Open Save	Run Section	
Image: A state of the	s 🕨 foglerlab 🕨 Desktop 🕨 LEP-12-1	- P
Current Folder 💿	📝 Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Rate_profile.m	Θ×
🗋 Name 🔻	ODEfun.m × Temp_profile.m × Conversion_profile.m × Rate_profile.m × +	
1 Temp_profile.m Rate_profile.m ODEfun.m Conversion_profile.m	<pre>1 - clc 2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor 3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta,T, and X 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - Cao=1.86; 7 - ☐ for i=1 : z(1,1) 8 %Calculating the values of rate from the following expressions 9 - Kc(i)=3.03*exp(-4149.63*(y(i,2)-333)/(y(i,2)*333)); 10 - K(i)=31.1*exp(7906*(y(i,2)-360)/(y(i,2)*360));</pre>	-
Details ^	11 - ra(i) =-K(i) *Cao*(1-(1+1/Kc(i))*y(i,3));	_
Workspace 💿	12 end $13 - plot(v, -ra);$	
Name 🔺 Value	14 - ylabel('-ra(kmol/m^3.h)');	
i 97	15 - xlabel('V(m^3)');	
Kc 1x97 doub	16 - axis([0 5 0 30])	
v 97x1 doub	17 - legend('rate')	
Η Vspan [0,5]	<pre>18 - title('Reaction rate profile down the reactor for co-current heat exchange')</pre>	
Xe 1x97 doub		
y 97x3 doub		
y0 [315,305,0] z [97,3]	Command Window	
	fg >>>	
۲ III +		
1111	script Ln 18	Col 77 .:

Step 25: Now run the file. You will get an output that looks like this



Counter- current heat exchange

Step 26: For counter-current flow, we only need to make two changes in the program. First, we modify the expression for Ta for counter-current flow. Multiply the right hand side of differential equation for Ta with -1. The following equation is obtained

$$d(Ta)/d(V) = -Ua*(T-Ta)/m/Cpc$$

To modify equation for Ta, open the function file ODEfun and put a minus sign on the right hand side of Ta equation as shown below:

A MATLAB R2014b		-										0 🛛
HOME PLO	отѕ	APPS	EDITOR	PUBLISH	VIEW	2		i i 9 ¢	🗗 🕐 S	earch Document	ation	<mark></mark>
New Open Save	G Find Files E Compare Print ▼	Q	Go To 👻 Comment	fx Fn · % ‰ ‰ F • • • EDIT • • •	Breakpoints	Run	Run and Advance	Run Section	Run and Time			
4 🔶 🔁 🔁 🌗 🕨	C: • Users	 foglerl 	lab 🕨 Desktop 🕨 l	EP-12-1								- 2
Current Folder	$\overline{\mathbf{v}}$	📝 Editor	r - C:\Users\foglerlal	\Desktop\LEP-1	2-1\ODEfun.m							₹ ®
Name 🔻		ODE	fun.m × +									
Temp_profile.m		1	function f =	• ODEfun (V,	Y)							
🐴 Rate_profile.m		2 -	Ta=Y(1);									_
🙆 ODEfun.m		3 -	T=Y(2);									
Conversion_prof	ile.m	4 —	X=Y(3);									
		5	<pre>% Explicit H</pre>	quations								
		6 -	Cao = 1.86;									
		7 -	CPo = 159;									
		8 -	dH = -34500;									
		9 -	FAo = 14.67;									
		10 -	UA = 5000;									
		11 -	m = 500;									E
Details	^	12 -	Cpc = 28;	10000								=
Workspace		13 - 14 -	k = 31.1 * e									
		14 -	Kc = 3.03 * ra = 0 - (k									
Name 🔺	Value	16 -	Xe = Kc / (1)		- ((1 + 1 /	KC)	^ A)))	,				
ans	[2;3;4] 🔺	17	<pre>% Differenti</pre>		ne							
Cao	1.8600 97	18 -	dTadV = -UA	-								
i K	97 1x97 dc	19 -	dTdV = (ra *			1 10	Po * F	Ao):				
Kc	1x97 dc	20 -	dXdV = 0 -		(//)	· · · ·		,,,				
ra ra	1x97 dc	21 -	<pre>f = [dTadV;</pre>		1;							
ra v	97x1 dc	22 -	end									
Η Vspan	[0,5]	23										-
📥 Xe	1x97 dc											
Vspan Xe y	97x3 dc	Comman	nd Window									۲
y 0 ∢	[315,30 -	$f_{\tilde{\star}} >>$										
	4											
							00)Efun		Lr	121 C	ol 26 🧮

Step 27: Save your file.

Step 28: Next, we guess *Ta* at V = 0 and see if it matches Tao=315 at V=5 m3. If it doesn't, we guess again. In this example, we will make first guess *Ta* (V=0) = 330 K and see if Ta = Ta0 = 315 K at V = 5 m3.

Change the initial value of y0 in all the script file as described below

a) **Temperature profile**

To make the changes, open your script file "Temp_profile" and change the 1^{st} guess value in y0 to 330 instead of 315 as shown below. The value of Ta at the end of reactor will be the last element of 1st column vector of y. So find the size of y and evaluate the value of y(n,1), where n is the last point

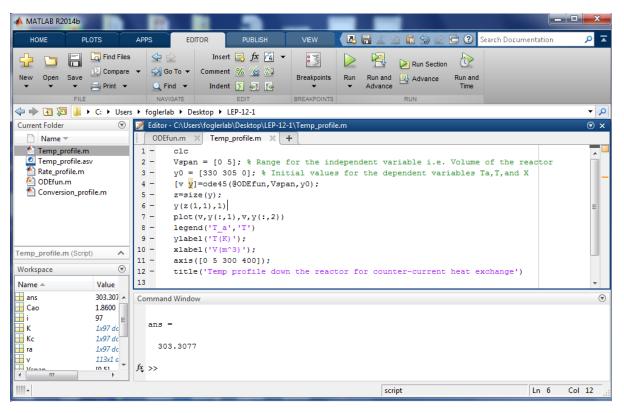
So,

z=size(y);

The value of Ta at the end of reactor will be given by y(z(1,1),1), where z(1,1) gives the row number of last element

Don't put semi-colon at the end of above line as you want the value of Ta to be displayed on command window. Also change the title of the graph from co-current to counter-current.

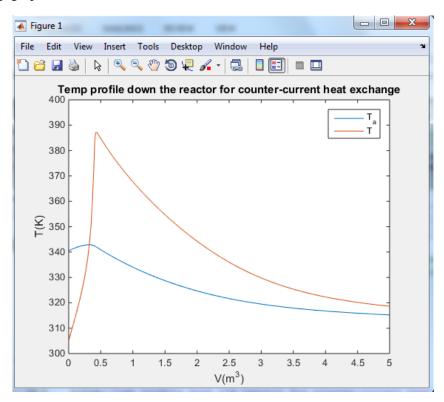
Step 29: Now save your file and run the program. In the command window, you can see that outlet temperature of Ta is 303.3 K but you want Ta=315 K



Step 30: Make another guess and check the Value of Ta. The final guess we obtain is Ta (V)=340.3 K where Ta (0)=315 K

MATLAB R2014b				-			_	-		• X
HOME PLOTS	APPS	EDITOR	PUBLISH	VIEW		h £ 9 ¢	E 🕐 Se	arch Documer	ntation	🔺 🔍
New Open Save	mpare		EDIT	Breakpoints	Run Run a	nd 🖳 Advance	n Run and Time			<u>م</u> -
Current Folder	🕤 📝 Edito	or - C:\Users\foglerl	ab\Desktop\LEP-12	-1\Temp_profile.	m					⊙×
🗋 Name 🔻	OD	Efun.m 🛛 Tem	p_profile.m 🛛 🕄	+						
Temp_profile.m Temp_profile.asv Rate_profile.asv ODEfun.m Conversion_profile.m Temp_profile.m (Script) Workspace Name Value Value Value	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13	<pre>y0 = [340.3 [v y]=ode43 z=size(y); y(z(1,1),1) plot(v,y(:, legend('T_a ylabel('T(i xlabel('V(i axis([0 5 3]</pre>	1),v,y(:,2)) (','T') ()'); ()'); ()');	nitial valu an,y0);	es for th	e dependent '	variables	Ta, T, and		
ans 314.9	77 A Comma	and Window								
Cao 1.860 i 97 K 1.97 Kc 1.97 ra 1.97 v 129x1 V 129x1 V 129x1	d_c ans d_c 3 d_c 3									
					[script			Ln 12	Col 73 .:
1111-						scupe			LII 12	COI 75

The following graph is obtained

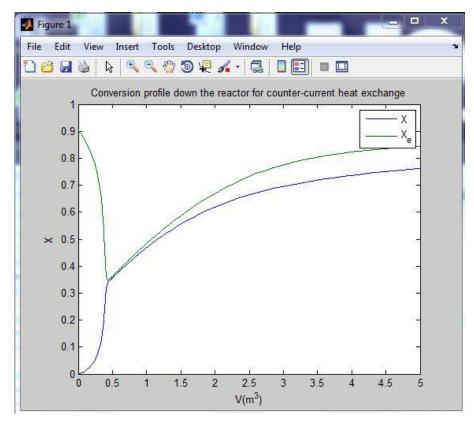


b) **Conversion profile**

Step 31: Change the initial value of Ta (i.e. Ta=340.3) and graph title in "Conversion_profile.m" file as shown

📣 MATLAB R2014b	
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🔚 🛣 🗇 🗭 🔁 🕐 Search Documentation 🔎 🖛
New Open Save Image: Compare Find Image: Compare Image: Compare FillE Image: Compare Image: Compare FillE Image: Compare Image: Compare Image: Compare Image: Compare Image: Compare <	Insert fx fx Image: Comment % % % % P Insert Image: Comment % % % % P Breakpoints Run Run and Advance Advance Run and Time NAVIGATE EDIT BREAKPOINTS RUN V V > foglerlab > Desktop LEP-12-1 V V V
Current Folder 💿	Zelitor - C:\Users\foglerlab\Desktop\LEP-12-1\Conversion_profile.m Image: Conversion_profile.m ODEfun.m X Temp_profile.m
Name Temp_profile.m Temp_profile.asv Rate_profile.m ODEfun.m Conversion_profile.m Rate_profile.m Vorkspace ans 314.977 Cao 1.8600 i i 129	<pre>1 - clc 2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor 3 - y0 = [340.3 305 0]; % Initial values for the dependent variables Ta,T, and X 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - [for i=1 : z(1,1) % Calculating the values of equilibrium conversion from the following expressions 8 - Kg(i)=3.03*exp(-4149.63*(y(i,2)-333)/(y(i,2)*333)); 9 - Xe(i)=Kc(i)/(1+Kc(i)); 10 - end 11 - plot(v,y(:,3),v,Xe); 12 - legend('X','X_e') 13 - ylabel('X'); 14 - xlabel('V(m^3)'); 15 - title('Conversion profile down the reactor for counter-current heat exchange')</pre>
	Command Window 🔊
ra 1x97 dc v 129x1 a v 129x1 a v 10 51	<i>μ</i> [*] >>
	script In 15 Col 79

When you run the program, you will get an output that looks like this

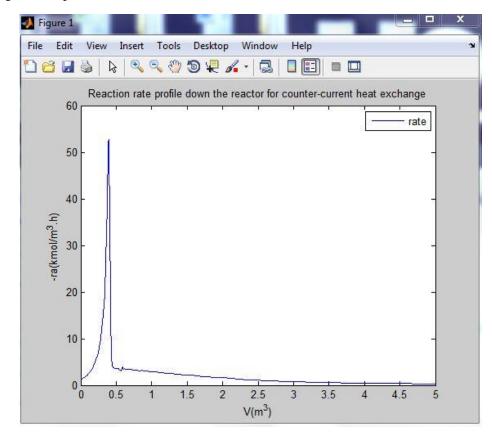


c) <u>Rate profile</u>

Step 32: Change the initial value of Ta and graph title in rate profile as shown

📣 MATLAB R2014b		x
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🖷 🔏 🖆 🛱 🗇 📿 📮 🕐 Search Documentation	◄ 🔍
New Open Save Compare V → → Print ▼ FILE → C:> User	📑 🔽 🔁 Run Section	• 0
Current Folder	Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Rate profile.m	⊙ x
Name T	ODEfun.m X Temp profile.m X Conversion profile.m X Rate profile.m X +	~ ~
Temp_profile.m Temp_profile.asv Rate_profile.asv ODEfun.m Conversion_profile.m	<pre>1 - clc 2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor 3 - y0 = [340.3 305 0]; % Initial values for the dependent variables Ta,T, and X 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - Cao=1.86; 7 - [for i=1 : z(1,1) 8 %Calculating the values of rate from the following expressions 9 - Kc(1)=3.03*exp(-4149.63*(y(1,2)-333)/(y(1,2)*333)); 10 - K(1)=3.1.1*exp(7906*(y(1,2)-360)/(y(1,2)*360));</pre>	
	11 - ra(i)=-K(i)*Cao*(1-(1+1/Kc(i))*y(i,3));	-
Workspace Image: Constraint of the system Name A Value ans 314,977 A Cao 1.8600 i 129 K 1x129 a Kc 1x129 a ra 1x129 a	<pre>12 - Lend 13 - plot(v,-ra); 14 - ylabel('-ra(kmol/m^3.h)'); 15 - xlabel('V(m^3)'); 16 - axis([0 5 0 60]) 17 - legend('rate') 18 - title('Reaction rate profile down the reactor for counter-current heat exchange')</pre>	
🕂 v 129x1 c	Command Window	
	tx >>	•
	script Ln 18 Col	5

You will get an output like this



Constant Ta case

Step 33: For Constant Ta, we only need to make one change i.e. modify the expression for Ta. Multiply the right hand side of differential equation for Ta with 0. The following equation is obtained

d(Ta)/d(V) = 0 *Ua*(T-Ta)/m/Cpc

In the function file, modify the equation of Ta only. All other equations will remain as it is.

📣 MATLAB R2014b		• X
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🖥 🔏 🗐 🗟 🖨 🕐 Search Documentation	<u>۹</u>
Image: Print of the second	Image: Second	- 0
Current Folder	rs → foglerlab → Desktop → LEP-12-1	<mark>♀ ▼</mark> ∞ × ⊙
Name -		• •
Temp_profile.m Rate_profile.m ODEfun.m Conversion_profile.m	<pre>1</pre>	
Rate_profile.m (Script)	13 - k = 31.1 * exp(7906 * (T - 360) / (T * 360)); 14 - Kc = 3.03 * exp((-34500/8.314) * (T - 333) / (T * 333));	
Name Value ans 314.9770 Cao 1.8600 i 129 K 1x129 dou Kc 1x129 dou v 129x1 dou Vspan [0,5] Xe 1x129 dou y 129x1 dou y 129x3 dou	21 - f = [dTadV; dTdV; dXdV]; 22 - end 23 24	
y0 [340.3000, z [129,3]	Command Window fx >>	۲
< <u> </u>		ol 10

In the script file all the parameters and equation will remain same as for co-current case except the title of the graph

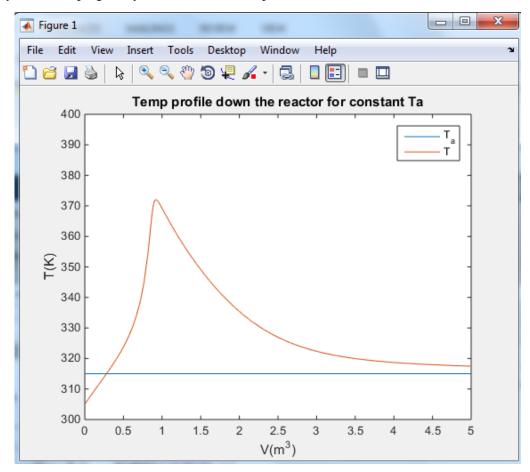
Change the title of the different graphs in all the script file and run the program to generate output.

a) **Temperature profile**

Step 34: The script file for Temp_profile should look like this

📣 MATLAB R2014b	10.00				
HOME PLOTS	APPS ED	ITOR PUBLISH	VIEW	🛃 🖥 🖌 🗄 🖆 🧇 d 🖻 🕐 😒	arch Documentation 🛛 🔎 🗖
New Open Save		Insert 🛃 fx 🏹 - Comment % % % Indent 🛐 🛃 💀 EDIT	Breakpoints	Image: New Section with the section of the section with the section with the section of the se	
<ri> <hr/> <hr/></ri>	s ▶ foglerlab ▶ D	esktop 🕨 LEP-12-1			ې 🗕
Current Folder 💿	📝 Editor - C:\Use	rs\foglerlab\Desktop\LEP-12	2-1\Temp_profile	.m	⊙ ×
🗋 Name 🔻	ODEfun.m	×∫ Temp_profile.m ×]	+		
1 Temp_profile.m	1 - clc				
🛃 Rate_profile.m 🖄 ODEfun.m 🏝 Conversion_profile.m	3 - y0 = 4 - [v y		tial values an,y0);	ependent variable i.e. Volume (for the dependent variables Ta	
Temp_profile.m (Script)	6 - lege	nd('T_a','T')			
Workspace 💿	-	el('T(K)');			
		el('V(m^3)');			
Name 🔺 Value		([0 5 300 400]);			
🖶 ans 314.977 🔺		e(.temb brotife dow	m the react	or for constant Ta <mark>')</mark>	-
🕂 Cao 1.8600 👻	11				
4	Command Windo	W			
IIII •				script	Ln 10 Col 53

When you run the program, you should see an output like this

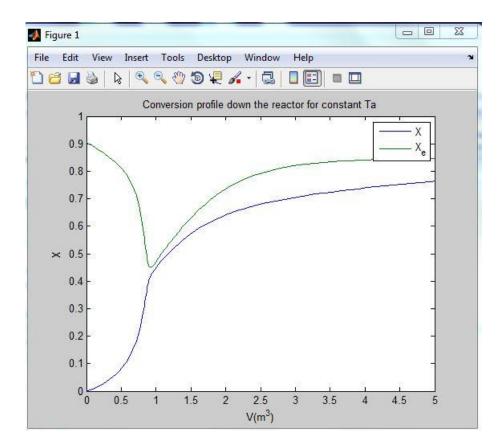


b) Conversion profile

Step 35: The script file for Conversion_profile should look like this

A MATLAB R2014b	
MATLAB R2014B	
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🔚 🔏 🖆 🛱 🥱 🗭 🔂 Search Documentation 🔎 🗖
New Open Save	Insert I → fx → Comment % % % % I indext I → Finite Insert I → fx → Comment % % % % I indext I → Finite Insert I → Finite <td< td=""></td<>
Current Folder	Kongenad & Beckop & Ell II I
Name ▼	ODEfun.m X Temp profile.m X Conversion profile.m X +
 Temp_profile.m Rate_profile.m ODEfun.m Conversion_profile.m 	<pre>1 - clc 2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor 3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta,T,and X 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - □ for i=1 : z(1,1) 7 %Calculating the values of equilibrium conversion from the following expressions 8 - Kc(1)=3.03*exp(-4149.63*(y(1,2)-333)/(y(1,2)*333)); 9 - Xc(1)=Kc(1)/(1+Kc(1));</pre>
Conversion_profile.m (Script) Workspace	10 - Lend 11 - plot(v,y(:,3),v,Xe); 12 - legend('X','X e')
Name A Value	<pre>13 - ylabel('X'); 14 - xlabel('X');</pre>
ans 314,977 × Cao 1.8600 i 129 E K 1x129 c Kc 1x129 c ra 1x129 c	<pre>If =</pre>
v 97x1 dc ▼	Command Window 🔊
	script Ln 15 Col 61

When you run this file, you should get an output like this

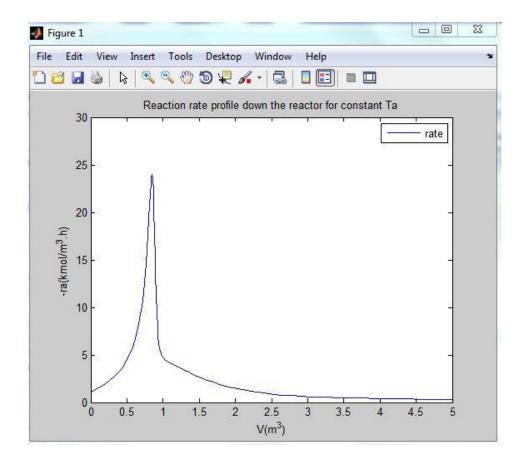


c) <u>Rate profile</u>

Step 36: The script file should look like this

📣 MATLAB R2014b		×
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🖬 🛣 🗇 🗭 🔁 🕐 Search Documentation	<mark>,</mark>
New Open Save Save File Print •	🔄 🔽 🔽 🔁 Run Section	م -
Current Folder	Kongenaal / Beskop / Ee 12 1 Kate profile.m	⊙ ×
Name 🔻	ODEfun.m X Temp_profile.m X Conversion_profile.m X Rate profile.m X +	○ ∧
 Rate_profile.m ODEfun.m Conversion_profile.m 	<pre>2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor 3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta,T, and X 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - Cao=1.86; 7 - [for i=1 : z(1,1) 8 %Calculating the values of rate from the following expressions 9 - Kc(i)=3.03*exp(-4149.63*(y(i,2)-333)/(y(i,2)*333));</pre>	-
Rate_profile.m (Script)	10 - K(i)=31.1*exp(7906*(y(i,2)-360)/(y(i,2)*360));	_
Workspace Name Value	11 - ra(i)=-K(i)*Cao*(1-(1+1/Kc(i))*Y(i,3)); 12 - end 13 - plot(y,-ra);	-
Cao 1.8600 ^	<pre>13 - plot(v,-ra); 14 - ylabel('-ra(kmol/m^3.h)'); 15 - xlabel('V(m^3)');</pre>	
K 1x97 dc Kc 1x97 dc	16 - axis([0 5 0 30]) 17 - legend('rate')	
ra 1x97 dc v 97x1 dc Vspan [0.5]	18 - title('Reaction rate profile down the reactor for constant Ta')	
Vspan [0,5] ▼	Command Window	۲
•		I 14 .::

When you run the above file, the output generated would be



Adiabatic operation

Step 37: For the adiabatic operation, heat exchange is zero i.e. Ua=0

In the function file, modify the expression for Ua (under explicit equation section). Make Ua=0 instead of 5000

Your modified function file should look like this

MATLAB R2014b		
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🔚 🔏 📾 🛱 🥱 🗭 🕗 Search Documentation	🔺 🔍
New Open Save Print +	Insert fx fx Fx Image: Comment fx Fx Image: Comment Fx Fx	
🔶 🔁 🔀 🚺 🕨 C: 🕨 Users	▶ foglerlab ▶ Desktop ▶ LEP-12-1	- ₽
Current Folder 💿	📝 Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m	⊙ ×
🗋 Name 🔻	ODEfun.m 🗶 🕂	
Temp_profile.m Rate_profile.m ODEfun.m Conversion_profile.m Rate_profile.m (Script)	<pre>1</pre>	× H
Workspace Name Value	$\begin{array}{l} 13 \\ 14 \\ 14 \\ 15 \\ 15 \\ 16 \\ 16 \\ 16 \\ 16 \\ 16 \\ 16$	-
Cao 1.8600 ^	17 & Differential equations	
K 1x97 dc	18 - dTadV = UA * (T - Ta) / (m*Cpc); $19 - dTdV = (ra * dH - (UA * (T - Ta))) / (CPo * FAo);$	
Kc 1x97 dc	20 - dXdV = 0 - (ra / FAO);	
ra 1x97 dc v 97x1 dc	<pre>21 - f = [dTadV; dTdV; dXdV];</pre>	
	22 - end	
🕂 Xe 1x97 dc	23 24	-
y 97x3 dc	Command Window	
y0 [315,30 +	fx >>	•
		Col 9 .:

All the other expression and equation will remain same as per co-current case.

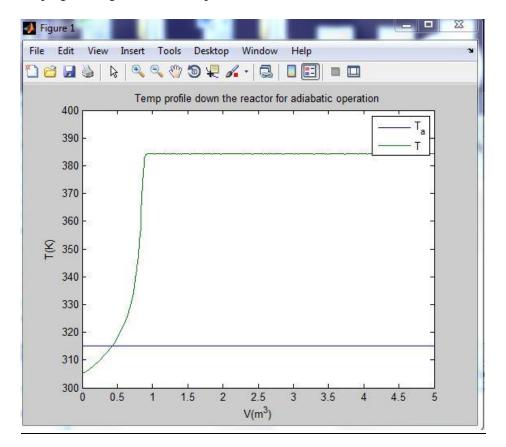
In the script file, all expression and equation will remain same as per co-current case except the graph title.

a) **Temperature profile**

Step 38: The script file for Temp_profile should look like this

📣 MATLAB R2014b				
HOME PLOTS	APPS EDITOR PUBLISH	VIEW 🛃	, 🖷 🔏 🛍 🛱 🗇 🛃 🔁 🕐 Search Documen	tation 🔎 🔼
Image: New Open with the second s		BREAKPOINTS	Run and Advance Run and Time RUN	
🗢 🔶 🔄 💭 퉬 🕨 C: 🕨 Users	s ▶ foglerlab ▶ Desktop ▶ LEP-12-1			- ₽
Current Folder Name Current Folder Name Current Folder Rate_profile.m	<pre>Editor - C:\Users\foglerlab\Desktop\LEP-1 DDEfun.m X Temp_profile.m X 1 - clc 2 - Vspan = [0 5]; % Range</pre>	+	ndent variable i.e. Volume of the rea	⊙ ×
ODEfun.m Conversion_profile.m	<pre>4 - [v y]=ode45(@ODEfun,Vsg 5 - plot(v,y(:,1),v,y(:,2)) 6 - legend('T_a','T')</pre>	pan,y0);	r the dependent variables Ta,T,and X	_
Temp_profile.m (Script) Workspace Image: Content of the second	<pre>7 - ylabel('T(K)'); 8 - xlabel('V(m^3)'); 9 - axis([0 5 300 400]);</pre>			
Name Value Cao 1.8600 i 97 K 1x97 dc Kc 1x97 dc		wn the reactor :	for Adiabatic operation)	
ra 1x97 dc –	Command Window			
				-

Run the above program to generate the output shown below

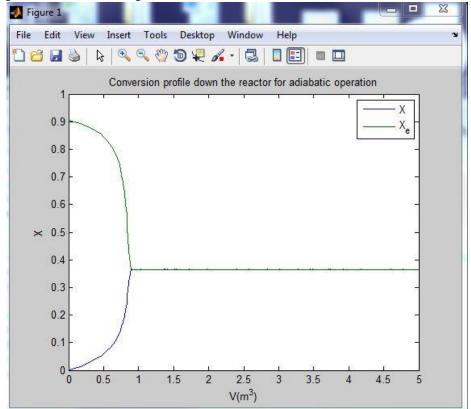


b) **Conversion profile**

Step 39: The script file for Conversion_profile is as shown in screenshot

📣 MATLAB R2014b	
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🖷 🔏 🛍 🛱 🥱 🗭 📮 🕐 Search Documentation 🔎 🗖
New Open Save	Insert fx Fx Fx Comment Go To ▼ Comment Comment % % % Mavigare Navigare EDIT Breakpoints Run Run and × Advance Run Section Mavigare Navigare EDIT Breakpoints Run Run and × Advance S + foglerlab + Desktop + LEP-12-1
Current Folder 💿	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Conversion profile.m
Name 🔻	ODEfun.m X Temp_profile.m X Conversion_profile.m X +
Temp_profile.m Rate_profile.m ODEfun.m Conversion_profile.m Conversion_profile.m Workspace Cao 1.8600 ▲ 1.97 K 1.x97 dc Kc 1.x97 dc Rate 1.x97 dc Kc 1.x97 dc	<pre>1 - clc 2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor 3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta,T, and X 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - For i=1 : z(1,1) 7 %Calculating the values of equilibrium conversion from the following expressions 8 - Kc(i)=3.03*exp(-4149.63*(y(i,2)-333)/(y(i,2)*333)); 9 - Xe(i)=Kc(i)/(1+Kc(i)); 10 - end 11 - plot(v,y(:,3),v,Xe); 12 - legend('X','X_e') 13 - ylabel('X'); 14 - xlabel('V(m^3)'); 15 - title('Conversion profile down the reactor for Adiabatic operation')</pre>
₩ 581x1 c	Command Window (*
Vspan [0,5] Xe 1x97 dc ≠ ✓ III	Command Window (*)
	script In 15 Col 67

The output generated after running the file is

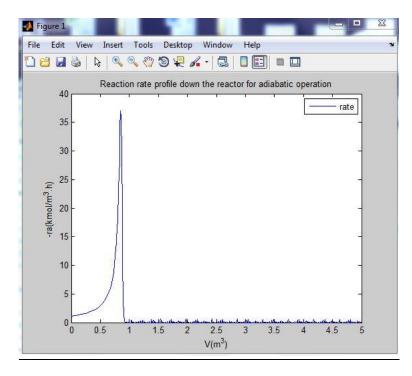


c) <u>Rate profile</u>

Step 40: The modified Rate_profile for adiabatic operation is

MATLAB R2014b) X
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🖥 🛱 🛱 🛱 😨 💽 Search Documentation	<mark></mark>
New Open Save	Run Section	
🜩 🔁 🔀 퉬 🕨 C: 🕨 User:	s 🔸 foglerlab 🕨 Desktop 🕨 LEP-12-1	- P
Current Folder Name	Editor - C:\User:\foglerlab\Desktop\LEP-12-1:\Rate_profile.m ODEfun.m × Temp_profile.m × Conversion_profile.m × Rate_profile.m × +	⊙ ×
Temp_profile.m Rate_profile.m Conversion_profile.m	<pre>1 - clc 2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor 3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta,T, and X 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - Cao=1.86; 7 - ☐ for i=1 : z(1,1) 8 %Calculating the values of rate from the following expressions 9 - Kc(i)=3.03*exp(-4149.63*(y(1,2)-333)/(y(1,2)*333));</pre>	_
ODEfun.m (Function) Image: Color of the second se	<pre>10 - K(i)=31.1*exp(7906*(y(1,2)=56)/y(1,2)*56)); 11 - ra(i)=-K(i)*Cao*(1-(1+1/Kc(i))*y(i,3)); 12 - end 13 - plot(v,-ra); 14 - ylabel('-ra(kmol/m^3.h)');</pre>	-
i 581 K 1x581 c Kc 1x581 c ra 1x581 c v 581x1 c Vspan [0,5]	<pre>15 - xlabel('V(m^3)'); 16 - axis([0 5 0 40]) 17 - legend('rate') 18 - title('Reaction rate profile down the reactor for Adiabatic operation')</pre>	
Xe 1x581 a y 581x3 a *	Command Window fx >>	۲
	script Ln 16 Co	ol 14 .:

When you run the above file, the output generated is



This is all basically all you need to get started with solving differential equations in MATLAB. If you face any problem then restart MATLAB or try to solve error. To get the value of particular parameter, remove the semi-colon from the file and you will find that value is displayed on the command window