#### 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/Cpc

d(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(0)=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 R20146	2		and the second second second				0	×
HOME	PLOTS	APPS		LDEDEE	) 🕐 Se	arch Document	ation	P -
New New Ope Script Piul		Import Save Data Workspace	New Variable  Open Variable  Clear Workspace  VARIABLE	Analyze Code	Simulink Library		RESOURCES	
	C: + Users	► foglerlab ► Doo	cuments + MATLAB					+ ,o
Current Folder Name   Details Workspace	^ @							⊙
Name +	Value							

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 🖉	Ŧ
New New Open Compar Script FILE	Open Variable   Run and Time	
		Q.
Current Folder 💿	Command Window	۲
Name 👻	ft >>	
Details ^		
Workspace 💿		
Name A Value		
IIII - 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			<li>Searce</li>	h Documentati	on 🔎
lew korpt	•	Import Save Data Workspace	New Variable	Analyze Code	Simulink Library		REBOURCES
4	Script Ctrl+N	foglerlab + Desi	Contraction of the second s		and a second		-
and the second se	Function	mand Window					
		>>					
8	Example	1000					
1000							
8	Class						
0-							
0-	Class System Object >						
etails 🛅		• •					
etailis 🚔 Vorksp	System Object >	-					
etails 🔚	System Object						
etailis 🚔 Vorksp	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		
HOME PLOTS APPS	EDITOR PUBLISH VIEW 🛃 🛱 🔏 🛱 🏦 🗇 🗭 🛱 🕐 Search Documentation	⊾ ۹
New Open Save	e 🗸 🎲 Go To 👻 Comment % 🎉 🐲 Breakpoints Run Run and 🔜 Advance Run and	1
	rs ▶ foglerlab ▶ Desktop ▶ LEP-12-1	- P
Current Folder   Name   Details	<pre>Editor - Untitled* Untitled* I Gutput args ] = Untitled( input args ) Gutput args ] = Untitled( input args ) Gutput args ) Gutput args ] = Untitled( input args ) Gutput args ) Gutput args ] = Untitled( input args ) Gutput args ) Gutput args ] = Untitled( input args ) Gutput args ) Gutput args ] = Untitled( input args ) Gutput args ] = Untitled( input args ) Gutput args ] Gutput args ] = Untitled( input args ) Gutput args ] Gutput args ] = Untitled( input args ) Gutput args ] Gutput</pre>	▼ ×
Workspace 💿	Command Window	۲
Name A Value	fx; >>	
Ready	Untitled Ln 8 C	ol 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" is a vector containing the integration span i.e. initial and final value of volume of reactor (in this case, Vinit=0, Vfinal=5).

So V= [Vinit Vfinal]

Or, V= [0 5]

Second input argument "Y" is also a vector and contains 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).

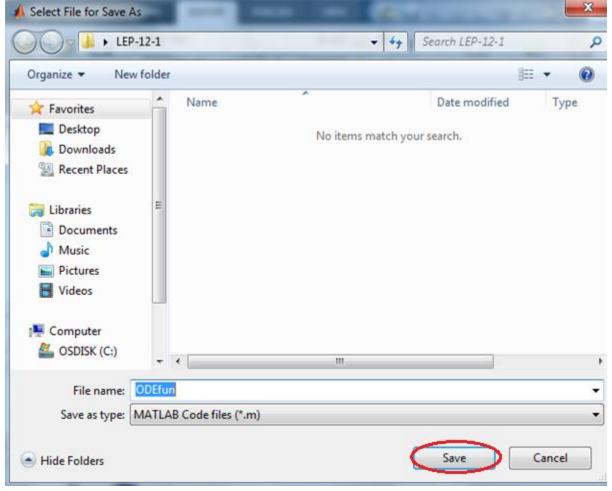
So Y = [Ta(0) T(0) X(0)]

Or, Y= [315 305 0]

The values of V and Y will be defined in the script file and then passed to the function file. This will become clearer as you go through the tutorial

#### 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"



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 VIEW 🛃 🔚 🔏 🖆 😭 😪 🛱 😯 Search Documentation	<mark>&gt;</mark> 🔺
Image: Save       Image: Compare Save       Image: Compare Save         New       Open       Save       Image: Print         FILE       Image: Print       Image: Print       Image: Print         Image: Print       Image: Print       Image: Print       Image: Print<	e 🗸 🎲 Go To 👻 Comment % ‰ ‰ Breakpoints Run Run and 🗔 Advance Run and	▼ ×
Details  Vorkspace  Value	7 Command Window fx >>	•
< Ⅲ ►		ol 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	_	-	and the second se	
HOME PLOTS APP	EDITOR	PUBLISH VIEW		)Search Documentation 🔎 🛣
New Open Save	e 💌 ᠫ Go To 👻	Insert , fx Comment % % % Indert : EDIT		tun Section dvance Run and Time
<ri>        Image: Second state       Image: Second state</ri>	rs ► foglerlab ► De	esktop 🕨 LEP-12-1		<b>م</b> •
Current Folder 💿	📝 Editor - C:\User:	s\foglerlab\Desktop\LEP-1	2-1\ODEfun.m	⊙×
D Name  ODEfun.m	0DEfun.m № 1 - functi 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				
	Command Window	/		•
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	And a second second	
HOME PLOTS APPS	EDITOR PUBLISH VIEW 🛃 🖷 🔏 🛱 🛱 🗇 📿 🛱 🕐 Search D	Documentation 🛛 🔎 🗖
New Open Save	🚬 👘 🔁 🔽 🖓 🔎 Run Section	Run and Time
<rp></rp>	s ▶ foglerlab ▶ Desktop ▶ LEP-12-1	۹ -
Current Folder Name ODEfun.m*	<pre>Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m ODEfun.m × + I</pre>	⊙ × 
Details  Workspace  Value Value	7 8	
Ready	Command Window ODEfun	© Ln 5 Col 21

MATLAB R2014b		• ×
HOME PLOTS APPS	EDITOR PUBLISH VIEW 🛃 🔚 🚣 🖆 😂 🖻 😨 Search Documentation	🖬 🤇
New Open Save	Indent Set     Indent Set <td></td>	
FILE ← → 🔁 🔀 📙 → C: → User	NAVIGATE     EDIT     BREAKPOINTS     RUN       rs • foglerlab • Desktop • LEP-12-1	<del>-</del> 0
Current Folder  Name	Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m     ODEfun.m    X +	Θ×
Molefun.m       Details     ▲       Workspace     ⑦       Name     Value	1 Gunction $f = ODEfun(V, Y)$ 2 - Ta=Y(1); 3 - T=Y(2); 4 - X=Y(3); 5 * Explicit Equations 6 - Cao = 1.86; 7 - CPo = 159; 8 - dd = -34500; 9 - FAo = 14.67; 10 - UA = 5000; 11 - m = 500; 12 - Opc = 28; 13 - k = 31.1 * exp(7906 * (T - 360) / (T * 360)); 14 - Kc = 3.03 * exp((-34500/8.314) * (T - 333) / (T * 333)); 15 - ra = 0 - (k * Cao * (1 - ((1 + 1 / Kc) * X))); 16 - Xe = Kc / (1 + Kc); 17 18 - end 19	
4	Command Window J≭ >>	<ul> <li>Image: A state of the state of</li></ul>
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			
HOME P	LOTS	APPS EDITOR PUBLISH VIEW 🔀 🖶 🔏 🛱 🛱 🗇 🗭 🔁 🕐 Search Documentation	2 م
New Open Save	Find Files	e V V Go To V Comment % % % P Breakpoints Run Run and Advance Run and	
🗘 🔶 🖸 🌾	C: Users	rs ▶ foglerlab ▶ Desktop ▶ LEP-12-1	<del>-</del> ۶
Current Folder	۲		⊙ ×
Temp_profile.n  Rate_profile.n  ODEfun.m  Conversion_pro		<pre>1</pre>	
Details Workspace Name A ans Cao	Value     [2;3;4]     1.8600	12 - Cpc = 28; 13 - k = 31.1 * exp(7906 * (T - 360) / (T * 360)); 14 - Kc = 3.03 * exp((-34500/8.314) * (T - 333) / (T * 333)); 15 - ra = 0 - (k * Cao * (1 - ((1 + 1 / Kc) * X))); 16 - Xe = Kc / (1 + Kc); 17 * Differential equations	
i K Kc ra V Vspan	97 1x97 dc 1x97 dc 1x97 dc 97x1 dc [0,5]	<pre>18 - dTadV = UA * (T - Ta) / (m*Cpc); 19 - dTdV = (ra * dH - (UA * (T - Ta))) / (CFo * FAo); 20 - dXdV = 0 - (ra / FAo); 21 - f = [dTadV; dTdV; dXdV]; 22 - end 23</pre>	
Xe y y0	1x97 dc 97x3 dc [315,30 ▼	Command Window	6
( m	•		
*		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,

 $\begin{bmatrix} Tan & T2n & Xn \end{bmatrix}^T$ T, & X for this case). Note that no of rows are same in vector v and matrix y. The return value of the 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 R2014b		7					100		X
HOME PLOTS APPS	EDITOR PUBL	ISH VEW		10 . IC	90	🔁 🕐 Search i	Documentation		۵ 🛪
New pen Save Print + Script Ctrl+N	e 👻 🔂 Go To 👻 Com	nsert 🔜 fx 🗐 🔹 ment % 🙀 🖓 dent 🔽 💽 🕞 EDIT	Breakpoints	Run 👻	Run and Advance	Run Section	Run and Time		
: Vise	rs 🕨 foglerlab 🕨 Desktop	LEP-12-1							- ,0
fx     Function       Function     Example       Example     System Object       Details     A       Workspace     Second       Name A     Value	2 - Ta=Y(1); 3 - T=Y(2); 4 - X=Y(3); 5 & Explicit 6 - Cac = 1.86 7 - CPc = 159; 8 - dH = -3450 9 - FAc = 14.6	= ODEfun(V,Y) Equations ; ;							
۰ m )	10 - UA = 5000; Command Window								۲
IIII- Ready	No.5.		0	DEfun			Ln 22	Col	2

A black window will appear like this

📣 MATLAB R201	.4b														
HOME	LOTS	APPS	;	EDITOR	PUBLISH	VIEW	1	A 🔒 🚣	h i	90	🗄 🕐 Search	Documenta	tion	🔺 🔍	5
	ave )	G Find File E Compare Print ▼	• •	Go To - Go Find - NAVIGATE	Comment	5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	•	Breakpoints	Run	Run and Advance	Run Section	Run and Time			
🗢 🌩 🖸 🞘	<b>)</b> •	C: ► Use	rs 🕨	foglerlab 🕨 D	esktop 🕨 LE	P-12-1								<del>,</del> 2	2
Current Folder		$\overline{\mathbf{v}}$	2	Editor - Untitle	14									⊙ ×	
📄 Name 🔻				ODEfun.m	Juntitled4	× [ + ]									
lefun.m			1												
Details		^													
Workspace															
Name 🔺		Value													
٠		×	Cor	nmand Window	N		_								
IIII + Ready								so	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

					0
Organize 👻 Nev	v folde			· III (*	0
🔆 Favorites	<u> </u>	Name		Date modified	Type
E Desktop		1 ODEfun		4/24/2017 3:23 PM	MATLA
🐌 Downloads					
1 Recent Places	E				
词 Libraries					
Documents					
J Music					
Fictures					
Videos	-	· [	m		
File name:	Temp	profile			-
Save as type:	MATU	AB Code files (*.m)			-

**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 R	R2014b					-						x
НОМЕ	PLOTS	APPS	EDITOR	PUBLISH	VIEW		<b>h h</b>	<b>9</b> ¢	🔁 🕐 Search	Documenta	tion	⊼ ۹
New Open	Save	Find File Compare Print V	e 👻 🚽 Go To 👻	Comment S Indent	★ ▲ · · · · · · · · · · · · · · · · · ·	Breakpoints	Run	Run and Advance	Run Section	Run and Time		
🔶 🔶 🔁 🌾	ه 🐌 🖾	C: ► Use	rs ► foglerlab ► D	esktop 🕨 LEP	-12-1							<del>-</del> 2
Current Folde	er	۲	📝 Editor - C:\Use	s\foglerlab\De	sktop\LEP-12-	-1\Temp_profile.	m					⊙×
🖺 Temp_ 🖄 ODEfu			-		-	or the indep ial values :						
Details		^										
Workspace												
Name 🔺	1	Value										
			•			11	1					•
< <u> </u>		4	Command Windo	N								
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	Problem 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 ODE fun 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 🛃 🔚 🔏 🛍 🛱	j 코 🖻 🔁 🕐 Search Documentation 🛛 👂 🗖
New Open Save	Indent     Indent     Image: Finitian state     Breakpoints     Run       NAVIGATE     EDIT     BREAKPOINTS	Run Section Run and Advance Run and Time RUN
	foglerlab	<ul> <li>Q</li> <li>∞</li> </ul>
Name ▼       ① Temp_profile.m       ▲ ODEfun.m	<pre>Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Temp_profile.m ODEfun.m X Temp_profile.m X + clc Vspan = [0 5]; % Range for the independe y0 = [315 305 0]; % Initial values for t [v y]=ode45(@ODEfun,Vspan,y0);</pre>	nt variable i.e. Volume of the reactor
Details  Workspace  Value Value		
<		
✓ IIII → Ready	ommand Window script	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									00	100	3
HOME PLOTS APPS	EDITOR	PUBLISH	VIEW			00	earch Doc	umentatio	n	P	Ŧ
Image: New Open Save     Image: Compare Print	<ul> <li></li></ul>	Comment 🌺 Indent 💽	1 19	Breakpoints	Run Advan	ce Carlana	nce R	un and Time			
FILE	NAVIGATE	EDI		BREAKPOINTS		RUN				•	0
Current Folder				1\Temp_profile.n	n					0	
Name *	ODEfun.m 💥	and the second se									
Temp_profile.m	1 - clc	1		2014 - C							T
	3 - y0 = 4 - [v y] 5 - plot(*	[315 305 0] =ode45(@ODE v,y(:,1),v,	<pre>; % Init fun,Vspa y(:,2));</pre>								
Details ^	3 - y0 = 4 - [v y] 5 - plot( 6 - legen: 7 - ylabe 8 - xlabe	<pre>[315 305 0] =ode45(@ODE v,y(:,1),v, d('T_a','T' l('T(K)'); l('V(m^3)')</pre>	; % Init fun,Vspa y(:,2)); ) ;	ial values n,y0);							
Details ^ Workspace ③	3 - y0 = $4 - [v y]^{2}$ 5 - plot(' $6 - legen(' 7 - ylabe)8 - xlabe)9 - axis(')$	[315 305 0] =ode45(@ODE v,y(:,1),v, d('T_a','T' 1('T(R)'); 1('V(m^3)') [0 5 300 38	; % Init fun,Vspa y(:,2)); ) ; ; ;0]);	ial values n,y0);	for the de	pendent v	/ariabl	es Ta,I	I, and 1		
Details ^ Workspace • Name ^ Value 1	3 - y0 = $4 - [v y]^{2}$ 5 - plot(' $6 - legen(' 7 - ylabe)8 - xlabe)9 - axis(')$	[315 305 0] =ode45(@ODE v,y(:,1),v, d('T_a','T' 1('T(R)'); 1('V(m^3)') [0 5 300 38	; % Init fun,Vspa y(:,2)); ) ; ; ;0]);	ial values n,y0);	for the de	pendent v	/ariabl	es Ta,I	I, and 1		
Details Workspace Value V 97x1 dou V 5pan [0,5] y 97x3 dou 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3 - y0 = 4 - [v y] 5 - plot( 6 - legen 7 - ylabe 8 - xlabe 9 - axis( 0 - title 11 12 13	[315 305 0] =ode45(@ODE v,y(:,1),v, d('T_a','T' 1('T(R)'); 1('V(m^3)') [0 5 300 38	; % Init fun,Vspa y(:,2)); ) ; ; ;0]);	ial values n,y0);	for the de	pendent v	/ariabl	es Ta,I	I, and 1		
Details A Workspace Name A Value V 97x1 dou Vspan [0,5] y 97x3 dou 1 1 1 1 1 1 1 1 1 1 1 1 1	3 - y0 = 4 - [v y] 5 - plot( 6 - legen 7 - ylabe 8 - xlabe 9 - axis( 0 - title 11 12 13	[315 305 0] =ode45(@ODE v,y(:,1),v, d('T_a','T' 1('T(R)'); 1('V(m^3)') [0 5 300 38	; % Init fun,Vspa y(:,2)); ) ; ; ;0]);	ial values n,y0);	for the de	pendent v	/ariabl	es Ta,I	I, and 1		
Details Workspace Value V 97x1 dou Vspan y y y y y y y y y y y y y	3 - y0 = 4 - [v y] 5 - plot( 6 - legen 7 - ylabe 8 - xlabe 9 - axis( 0 - title 11 12 13	<pre>[315 305 0] =ode45(@ODE v,y(:,1),v, d('T_a','T' 1('T(K)'); 1('V(m'3)') [0 5 300 38 ('Temp prof</pre>	; % Init fun,Vspa y(:,2)); ) ; ; ;0]);	ial values n,y0); the reacto	for the de	pendent v	/ariabl	es Ta,I	I, and 1	×	•

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.

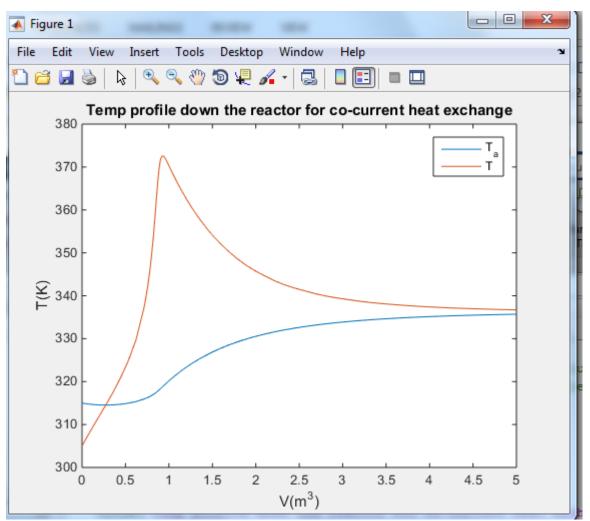
In this case, we have put both the files under the folder "LEP-12-1"

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

📣 MATLAB R2014b		• X
HOME PLOTS APP	S EDITOR PUBLISH VIEW 🛃 🔚 🔏 🛍 🛱 🤝 🔁 🕐 Search Documentation	۲ 🔍
New Open Save	e V V Go To V Comment % % % V Breakpoints Run Run and Advance Run and	
	rs 🕨 foglerlab 🕨 Desktop 🕨 LEP-12-1	- P
Current Folder 💿	Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Temp_profile.m	▼ ×
Name  Name	DDEfun.m X Temp_profile.m X +	
	<pre>2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the n 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(K)'); 8 - xlabel('V(m^3)'); </pre>	
Details ^	<pre>9 - axis([0 5 300 380]); 10 - title('Temp profile down the reactor for co-current heat exchange')</pre>	
Workspace 💿	11	-
Name         Value           v         97x1 dou           Vspan         [0,5]           y         97x3 dou           y0         [315,305,	III       Command Window       fx >> Temp_profile	
IIII.*		

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.

**Step 17:** If you wish to see the values of the Ta, T and X at various reactor volumes, you can do so by typing v and y in the file as shown below. Remember that typing without semi-colon will display the values in the Command window. Press run.

You will see that values of v and y are displayed in Command window in vector form. The nth row of y contains value of Ta, T, and X at the reactor volume corresponding to nth row of v vector

📣 MATLAB R2014b		23
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🖬 🔏 🕼 🛱 📿 🔁 🕐 Search Documentation	<mark>∕</mark> ⊼
New     Open     Save       •     •     •	Comment % % %      Breakpoints     Comment % % %      Breakpoints     Comment % % %      Breakpoints     Comment % % %      Comment % % %      Comment % % %      Decomment % %      Decomment % % % %      Decomment % % %      Decomment % % %      Decomment % % %      Decomment % % % %      Decomment % % % %      Decomment % % % % % %      Decomment % % % % % % % % % % % % % % % % % % %	
FILE	NAVIGATE EDIT BREAKPOINTS RUN	- P
	s > foglerlab > Desktop > LEP-12-1a > LEP-12-1a	
Current Folder 💿	Editor - C:\Users\foglerlab\Desktop\LEP-12-1a\LEP-12-1a\Temp_profile.m	▼ ×
Name  Temp_profile.m	Temp_profile.m × +	
1 rate_profile.m 20 ODEfun.m 1 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 - plot(v,y(r,1),v,y(r,2)); 6 - legend('T_a(','T') 7 - ylabel('T(K)'); 8 - xlabel('V(m^3)'); 9 - axis([0 5 300 380]);</pre>	
Details ^	<pre>10 - title('Temp profile down the reactor for co-current heat exchange')</pre>	
Workspace 🐨	$\frac{11}{12} - \frac{1}{12}$	-
Name 🔺 Value		
ans 314.8908	Command Window	
🖬 v 97x1 doub	5.0000	<b>^</b>
🖶 Vspan [0,5]		
y 97x3 doub v0 [315:305:0		_
H y0 [315;305;0	<u> </u>	
	315.0000 305.0000 0	
۰ III ا	∫t 314.9976 305.0252 0.0001	-
<b>*</b>		

If you wish to see the value of a particular variable at a given volume, then you can do so by entering command "y(a,b)"where a and b are row and column number of y matrix. So by varying "b", you can evaluate the value of different dependent variable (Ta. T or X) at fixed reactor volume (i.e. volume corresponding to row "a" of v vector) and by varying "a" you can evaluate the value of that dependent variable (i.e. variable corresponding to "b" column) at different reactor volume.

So, suppose, you want to get the value of volume and T at  $20^{th}$  row ,then type "v(20)" to get the volume of reactor at this row and type "y(20,2)" to get the value of T at  $20^{th}$  row as shown below

📣 MATLAB R2014b		
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🗮 🔏 🛱 🛱 😂 🖨 🕐 Search Documentation	<mark></mark>
New Open Save	Comment % % %      Reapoints     Comment % % %      Run and     Comment % % % % % %      Run and     Comment % % % % % % % % % % % % % % % % % % %	
← → 🔁 🔀 🍌 → C: → User Current Folder	s + foglerlab + Desktop + LEP-12-1a + LEP-12-1a Editor - C:\Users\foglerlab\Desktop\LEP-12-1a\LEP-12-1a\Temp_profile.m	<u>♀</u> ▼ × (1)
Name -	Temp profilem X +	© X
Details A Workspace ()	<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 - 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 - (20) 12 - (20) 13</pre>	E
Name A Value	14	-
ans 329.7680	Command Window	$\odot$
v 97x1 doub Vspan [0,5] y 97x3 doub y0 [315;305;0	0.6212	
	ang =	=
< <u> </u>	<u>た</u> >>	-

From the below figure, you can see that T=329.7 K at V= $0.6212 \text{ m}^3$ 

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

#### b) **Conversion profile**

Step 18: 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.

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]

Write the codes as shown below

📣 MATLAB R2014b					
HOME PLOTS APPS	EDITOR PUBLISH	VIEW 🛃 🔒 🚣	🖺 🔓 🕤 🖻 🔁 🕐 Search Document	ation 👂 🛣	
New Open Save	Insert     Insert       Image: Solution of the sector of the sect	ん 2 後記 Breakpoints	Image: New Section     Image: New Section       Run Run and Advance     Run and Time       RUN     RUN		
	foglerlab 🕨 Desktop 🕨 LEP-	-12-1		<b>→</b> 🔎	
Name ▼       1 -         Temp_profile.m       2 -         ODEfun.m       3 -         Conversion_profile.m       4 -         5 -	ODEfun.m × Temp_prof clc Vspan = [0 5]; % y0 = [315 305 0] [v y]=ode45(@ODE	<pre>% Range for the indep % % Initial values f</pre>			
Workspace         Image: Workspace         Image: Figure Workspace         figure Workspace <th cold="" td="" th<="" workspace<=""><td>nmand Window</td><td></td><td></td><td><math>\odot</math></td></th>	<td>nmand Window</td> <td></td> <td></td> <td><math>\odot</math></td>	nmand Window			$\odot$
		2	script L	n 5 Col 1 <sub>.::</sub>	

**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)

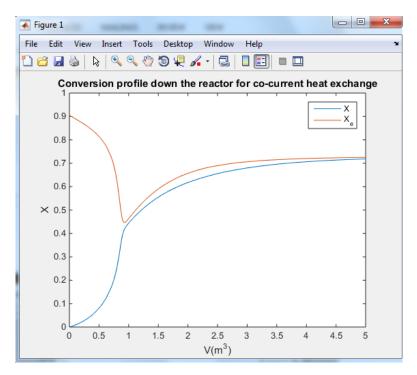
📣 MATLAB R2014b		_
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🔚 🔏 🖆 🛱 😒 🔁 🕐 Search Documentation 🔎	Ā
New Open Save FILE	🖳 🔽 🔁 Run Section	
	s → foglerlab → Desktop → LEP-12-1 v ,	٩
Current Folder 💿	📝 Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Conversion_profile.m 💿 :	×
Name Temp_profile.m ODEfun.m Conversion_profile.m Details	<pre>ODEfunm × Temp_profile.m × Conversion_profile.m × + 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*(v(i,2)-333)/(v(i,2)*333));</pre>	_
Workspace         Image: Constraint of the second seco	9 - Xxe(i)=Kc(i)/(1+Kc(i)); 10 - end	•
y 97x3 dc - ( IIII -	<i>f</i> <b>€</b> >> script	

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 Compare FILE	Indent     Indent     Image: Second s
	Foglerlab ► Desktop ► LEP-12-1       ✓     ✓       ✓     Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Conversion_profile.m
Name ▼         Temp_profile.m         ODEfun.m         Conversion_profile.m         Details         Workspace         ©         Name ▲       Value	ODEfun.m × Temp.profile.m ×       +         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 co-current heat exchange')
y0 [315,305,0]	Command Window (* /sj >>
	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



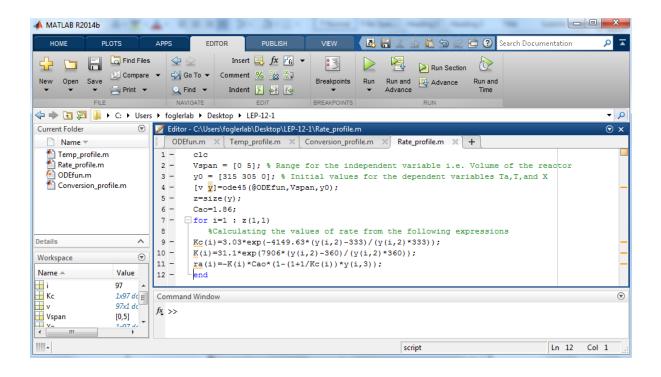
### c) <u>Rate profile</u>

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:

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

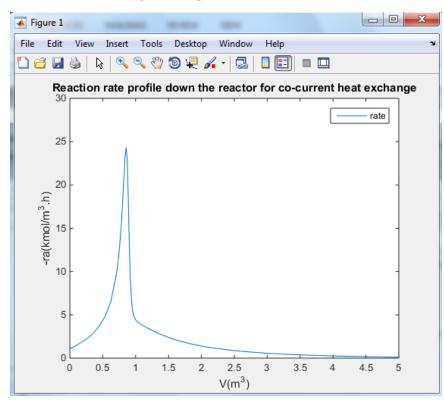
Insert the codes in the file as shown below. This will create a column vector of k, Kc and ra



**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	<mark></mark>
New Open Save FillE	e 🕶 🖓 Go To 👻 Comment % ‰ 🖏 Breakpoints Run Run and 🗔 Advance Run and	
	rs 🕨 foglerlab 🕨 Desktop 🕨 LEP-12-1	- P
Current Folder 💿	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Rate_profile.m	⊙×
🗋 Name 🔻	ODEfun.m × Temp_profile.m × Conversion_profile.m × Rate_profile.m × +	
🐴 Temp_profile.m	1 - clc	
🛐 Rate_profile.m	2 - Vspan = [0 5]; % Range for the independent variable i.e. Volume of the reactor	
ODEfun.m	3 - y0 = [315 305 0]; % Initial values for the dependent variables Ta,T, and X	
🖄 Conversion_profile.m	4 - [v y]=ode45(@ODEfun,Vspan,y0);	-
	5 - z=size(y);	
	6 - Cao=1.86; 7 - $\Box$ for i=1 : z(1.1)	
	<pre>7 - for i=1 : z(1,1) 8 %Calculating the values of rate from the following expressions</pre>	
	$9 - Kc(i) = 3.03 \exp(-4149.63*(y(i,2)-333)/(y(i,2)*333));$	_
	$10 - K(i) = 31.1 \exp(7906*(y(i,2) - 360) / (y(i,2) * 360));$	_
Details ^	$11 - \frac{x_{a}(i) = -K(i) *Cao* (1 - (1 + 1/Kc(i)) *y(i,3));}{2}$	_
	12 - end	
Workspace 💿	13 - plot(v,-ra);	
Name 🔺 Value	14 - ylabel('-ra(kmol/m^3.h)');	
🕂 i 97	15 - xlabel('V(m^3)');	
Kc 1x97 doub		
v 97x1 doub	17 - legend('rate') 18 - title('Reaction rate profile down the reactor for co-current heat exchange')	
Vspan [0,5] Xe 1x97 doub		
y 97x3 doub		
💾 у0 [315,305,0		
z [97,3]	Command Window	۲
۰ III +	$f_{x} >>$	
	script Ln 18 Co	177 .:

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:

MATLAB R2014b		
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🔚 🔏 🗟 🗟 😨 😧 Search Documentation	∡ ۹
🔓 🛅 🗐 🖾 Find Files	as 🧇 🖓 Insert 🛃 fx 🖪 👻 📑 🔛 🎽 🙀 Run Section	
Compare	V Comment % % %	
New Open Save	Q Find ▼ Indent P of the technology of	
FILE	NAVIGATE EDIT BREAKPOINTS RUN	
<rp></rp>	s ▶ foglerlab ▶ Desktop ▶ LEP-12-1	<del>-</del> ,0
Current Folder 💿	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\ODEfun.m	<b>∀</b> ×
Name 🔻	ODEfun.m × +	
Temp profile.m	1 _ function f = ODEfun(V,Y)	
Rate_profile.m	2 - Ta=Y(1);	<u> </u>
🖄 ODEfun.m	3 - T=Y(2);	
🛀 Conversion_profile.m	4 - X=Y(3);	
	5 % Explicit Equations	
	6 - Cao = 1.86;	
	7 - CPo = 159;	
	8 - dH = -34500;	
	9 - FAo = 14.67;	
	10 - UA = 5000;	
	11 - m = 500;	
Details ^	12 - Cpc = 28;	=
Workspace 💿	$13 - k = 31.1 * \exp(7906 * (T - 360) / (T * 360));$	
	$\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$	
Name 🔺 Value	15 - ra = 0 - (k * Cao * (1 - ((1 + 1 / Kc) * X))); 16 - Xe = Kc / (1 + Kc);	
ans [2;3;4] 🔺		
Cao 1.8600	<pre>17 % Differential equations 18 - dTadV = -UA * (T - Ta) / (m*Cpc);</pre>	
i 97 K 1x97 dc	$\frac{11}{19} = \frac{11}{12} \frac{11}{12} - \frac{11}{12} $	
K 1x97 dc Kc 1x97 dc	20 - dXdV = 0 - (ra / FAO);	
a 1x97 dc =	21 - f = [dTadV; dTdV; dXdV];	
v 97x1 dc	22 - end	
Vspan [0,5]	23	
Xe 1x97 dc		<b>T</b>
y 97x3 dc	Command Window	۲
y0 [315,30 ▼	$f_{\chi} >>$	
۲ III >		
<b> </b> +	ODEfun Ln 21	Col 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* = *Ta*0 = 315 K at V = 5 m3.

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

#### a) <u>Temperature profile</u>

To make the changes, open your script file "Temp\_profile" and change the 1<sup>st</sup> 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

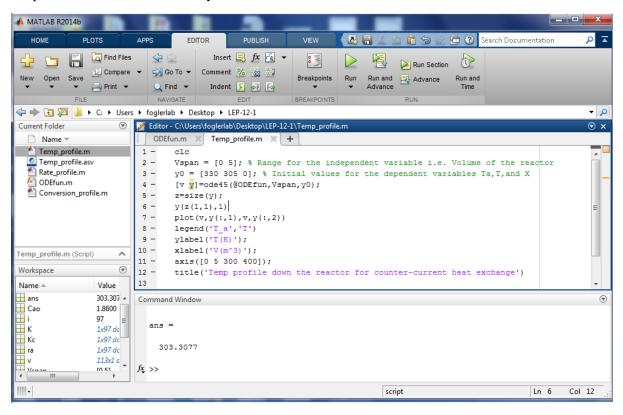
So,

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.

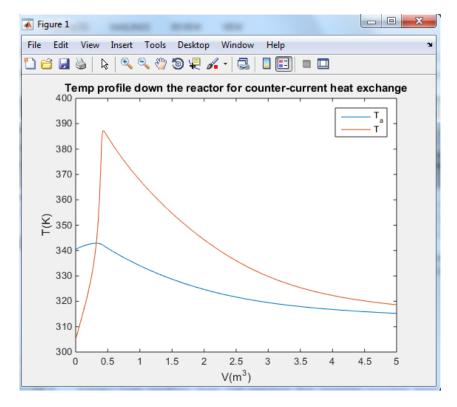
**Step 29:** Now save your file and run the program. In the command window, you can see that outlet temperature of Ta is 303.3 K but you want Ta=315 K



**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

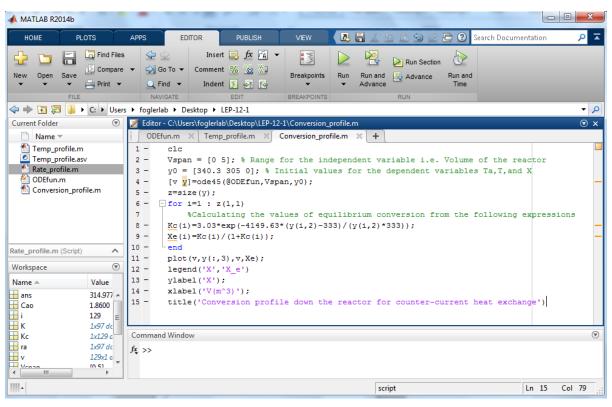
MATLAB R2014b		
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🖷 🔏 🛱 🛱 🗇 🧭 🚍 🕐 Search Documentation	<mark>&gt;</mark> 🔺
Image: New Open with the sector of the s	Run Section	
	s 🔸 foglerlab 🕨 Desktop 🕨 LEP-12-1	<b>▼</b>
Current Folder 💿	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Temp_profile.m	⊙×
🗋 Name 🔻	ODEfun.m X Temp_profile.m X +	_
Temp_profile.m Temp_profile.asv Ate_profile.asv ODEfun.m Conversion_profile.m Temp_profile.m (Script) Workspace Name  Value Value	<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 - y(z(1,1),1) 7 - plot(v,y(:,1),v,y(:,2)) 8 - legend('T_a','T') 9 - ylabel('T(K)'); 10 - xlabel('V(m^3)'); 11 - axis([0 5 300 400]); 12 - title('Temp profile down the reactor for counter-current heat exchange') 13</pre>	
ans 314.977 •	Command Window	
Gao 1.8600 i 97 ≡ K 1x97 dc Kc 1x97 dc v 1297 dc v 1297 dc v 1297 dc v 1297 dc	ans = 314.9770 fx >>	
IIII.*	script In 12 Co	ol 73 .::
<b>•</b>	script In 12 Co	<u>، 75 ار</u>

The following graph is obtained

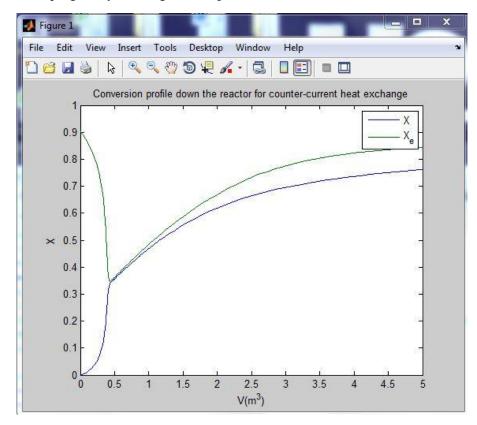


## b) **Conversion profile**

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



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

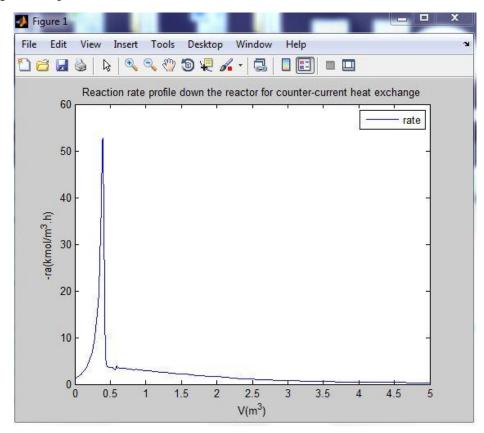


c) <u>Rate profile</u>

📣 MATLAB R2014b		-	from radio profil	is down the -	matter for co	-	-	heat made	-			• ×
HOME PL	отѕ	APPS	EDITOR	PUBLISH				1 <b>1</b> 9 ¢	0	Search Docum	entation	∡ ۹
FILE	Find Files		Go To ▼ Commen ind ▼ Inden /IGATE	EDIT	BREAKPOINTS		Run and Advance	Run Sectio	n C Run and Time			- 0
Current Folder	C: V Users		ab ► Desktop ► - C:\Users\foglerla		1101 01							<b>→</b> <u>&gt;</u>
Name      Temp_profile.m     Temp_profile.m     Rate_profile.m     ODEfun.m     Conversion_pro	v	ODE 1 - 2 - 3 - 4 - 5 - 6 -	fun.m × Temp clc Vspan = [0 y0 = [340.3	_profile.m × 5]; % Range 305 0]; % I (@ODEfun,Vsp	Conversion_prof for the ind nitial valu	ile.m ×	nt var					© ×
Rate_profile.m (Script) Workspace Name 🔺	∧ (▽) Value	8 9 - 10 - 11 - 12 - 13 -	<pre>Kc(i)=3.03* K(i)=31.1*e ra(i)=-K(i) end plot(v,-ra)</pre>		*(y(i,2)-33 ,2)-360)/(y /Kc(i))*y(i	3)/(y( (i,2)*	i,2)*3	33));	(pressio	ns		-
ans Cao i K Kc ra	314.977 ▲ 1.8600 129 1x129 c 1x129 c 1x129 c	14 - 15 - 16 - 17 - 18 -	<pre>xlabel('V(m axis([0 5 0 legend('rat</pre>	60])		the re	actor	for count	er-curre	nt heat ex	change')	
V Venan	129x1 c ro 51	Comman J× >>	d Window									•
A							scrip	pt			Ln 18 (	Col 5

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

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	⊾ ۹
New Open Save Compare FILE	🖳 🔽 🔁 🖓 Run Section	• 0
Current Folder	Z Editor - C:\Users\foqlerlab\Desktop\LEP-12-1\ODEfun.m	⊙ ×
Name -		^
Temp_profile.m Rate_profile.m ODEfun.m Conversion_profile.m Rate_profile.m (Script)	<pre>1</pre>	-
Name         Value           ans         314.9770           Cao         1.8600           i         129           K         1x129 down	<pre>17</pre>	
Kc         1x129 dout           ra         1x129 dout           v         129x1 dout           Vspan         [0,5]           Xe         1x129 dout           y         129x3 dout           y         129x3 dout           y0         [340.3000,2]	22 - end 23 24	
y0 [340.3000; z [129,3]	Command Window fx >>	۲
	ODEfun Ln 18 C	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

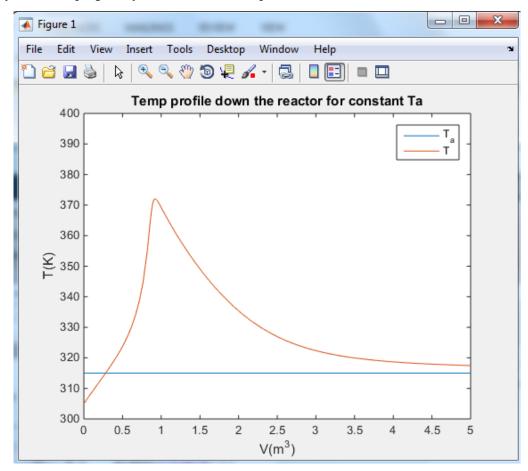
Change the title of the different graphs in all the script file and run the program to generate output.

### a) <u>Temperature profile</u>

Step 34: The script file for Temp\_profile should look like this

📣 MATLAB R2014b	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		
HOME PLOTS	APPS EDITOR PUBLISH VIEW	v 🛛 🛃 🔏 🖆 🛱 🥱 🗭 🖨 🕐 Search Documentation	⊾ ۹
New Open Save		points Run Run and Advance Run and Time	
<ri>        Image: Second state       Image: Second state       Image: Second state       Image: Second state         Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state         Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state       Image: Second state<td>→ foglerlab → Desktop → LEP-12-1</td><td></td><td>- P</td></ri>	→ foglerlab → Desktop → LEP-12-1		- P
Current Folder 💿	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Temp	_profile.m	⊙×
🗋 Name 🔻	ODEfun.m × Temp_profile.m × +		
Temp_profile.m  Rate_profile.m  ODEfun.m  Conversion_profile.m		e independent variable i.e. Volume of the reactor alues for the dependent variables Ta,T,and X ;	
Temp_profile.m (Script)	<pre>6 - legend('T_a','T') 7 - ylabel('T(K)'); 8 - xlabel('V(m^3)'); 9 - axis([0 5 300 400]);</pre>		
🖬 ans 314.977 🔺	<pre>10 - title('Temp profile down the : 11</pre>	reactor for constant Ta')	-
Cao 1.8600 -	Command Window		
			۲
		script Ln 10 Co	ol 53 <sub>.::</sub>

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

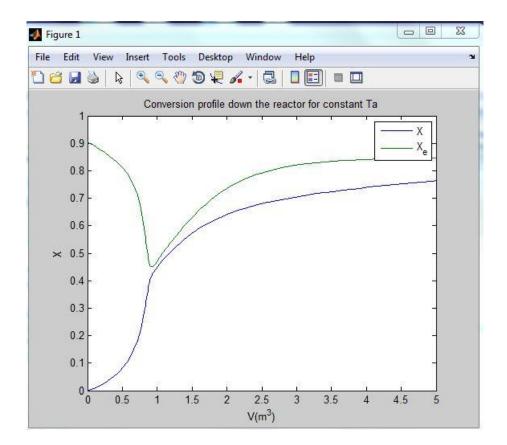


## b) **Conversion profile**

Step 35: The script file for Conversion\_profile should look like this

MATLAB R2014b	the second second	
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🔒	🔏 🛅 🛱 🗇 😥 🔁 🕐 Search Documentation 🛛 🔎 🗖
New Open Save	▼ 🖓 Go To ▼ Comment <u>%</u> ½ 2 Breakpoints Run Run	Run Section and Advance Run and Time RUN
Current Folder	Editor - C:\Users\foqlerlab\Desktop\LEP-12-1\Conversion_profile.m	⊙ x
Name		+
Conversion_profile.m (Script) ^	<pre>1 - clc 2 - Vspan = [0 5]; \$ Range for the independent 3 - y0 = [315 305 0]; \$ Initial values for the 4 - [v y]=ode45(@ODEfun,Vspan,y0); 5 - z=size(y); 6 - for i=1 : z(1,1)</pre>	variable i.e. Volume of the reactor dependent variables Ta,T, and X conversion from the following expressions
Workspace         Image: Constraint of the system           Name A         Value           ans         314.977 A           Cao         1.8600           i         129           K         1x129 c           Kc         1x129 c           ra         1x129 c	<pre>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</pre>	for constant Ta')
v 97x1 dc ▼	Command Window	•
IIII.*		script Ln 15 Col 61

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

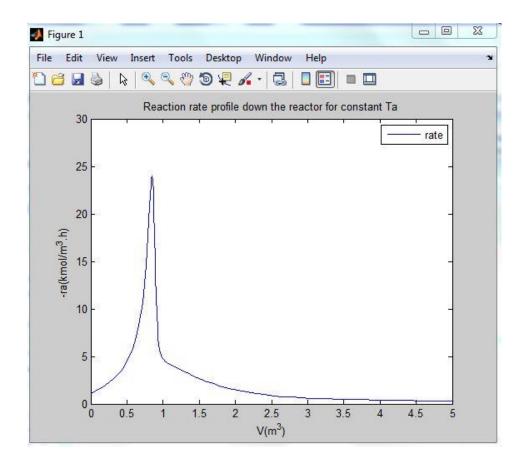


## c) <u>Rate profile</u>

Step 36: The script file should look like this

📣 MATLAB R2014b		• X
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🖥 🕼 🗇 🖻 🖗 Search Documentation	∡ م
New Open Save	Image: Second	
← → 🔁 💭 🍌 → C: → User Current Folder	rs > foglerlab > Desktop > LEP-12-1	<b>→</b> <u>&gt;</u>
	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Rate_profile.m	⊙×
Name  Name	DDEfun.m × Temp_profile.m × Conversion_profile.m × Rate_profile.m × +	
Rate_profile.m ODEfun.m Conversion_profile.m Rate_profile.m (Script)	<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(1)=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>	-
Workspace 💿	11 - ra(i)=-K(i)*Cao*(1-(1+1/Kc(i))*Y(i,3)); 12 - end	
Name         Value           Cao         1.8600           i         97           K         1x97 dc           Kc         1x97 dc           ra         1x97 dc           v         97x1 dc	<pre>13 - plot(v,-ra); 14 - ylabel('-ra(kmol/m^3.h)'); 15 - xlabel('V(m^3)'); 16 - axis([0 5 0 3b])) 17 - legend('rate') 18 - title('Reaction rate profile down the reactor for constant Ta')</pre>	
🕂 Vspan [0,5] 🔻	Command Window	
	script Ln 16 Ci	ol 14 .
100-1	script	01 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

📣 MATLAB R2014b		
HOME PLOTS	APPS EDITOR PUBLISH VEW 🛃 🖥 🛣 🖘 🖙 🔁 🕐 Search Documenta	tion 🔎 🔼
New Open Save	e 🗸 🖓 Go To 👻 Comment % 🎪 🕅 Breakpoints Run Run and 🖳 Advance Run and	<u>م</u> •
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) Workspace Name & Value	<pre>1</pre>	
Cao 1.8600 ▲ i 97 K 1x97 dc Kc 1x97 dc	<pre>17 % Differential equations 18 - dTadV = UA * (T - Ta) / (m*Cpc); 19 - dTdV = (ra * dH - (UA * (T - Ta))) / (CPo * FAo); 20 - dXdV = 0 - (ra / FAo);</pre>	
ra 1x97 dc v 97x1 dc	21 - f = [dTadV; dTdV; dXdV];	
v 97x1 dc Vspan [0,5] Xe 1x97 dc y 97x3 dc	22 - end 23 24	-
y0 [315,30 →	Command Window	۲
< III >	<i>f</i> <sub>*</sub> >>	÷
<b>Ⅲ</b> ▲	ODEfun Ln	10 Col 9 .::

Your modified function file should look like this

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

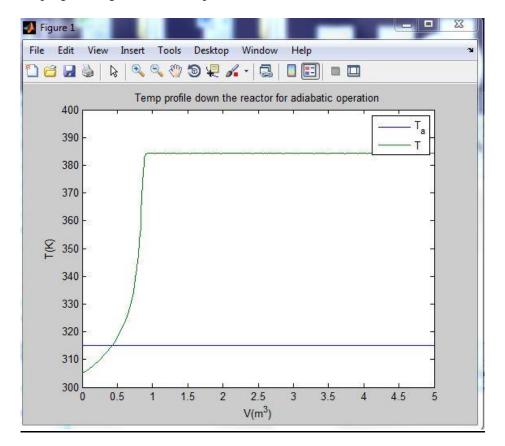
In the script file, all expression and equation will remain same as per co-current case except the graph title.

## a) <u>Temperature profile</u>

Step 38: The script file for Temp\_profile should look like this

📣 MATLAB R2014b			-						- 0 ×
HOME PLOTS	APPS	EDITOR	PUBLISH	VIEW		libioc	E ? S	earch Documentation	🔺 🤉
	nd Files ompare rint NAVIO	To ▼ Comment d ▼ Indent	f×         f₄            %         ‰         ‰         ↓           E         •         •         •	Breakpoints		Run Section Run and Advance RUN	Run and Time		
🔶 🔁 🔁 🚺 🕨 G 🕨	Users 🕨 foglerlat	> > Desktop > L	EP-12-1						- ,2
Current Folder		C:\Users\foglerlat in.m X Temp	o\Desktop\LEP-12 _profile.m ≈		.m				⊙ ×
Temp_profile.m Rate_profile.m ODEfun.m Conversion_profile.m Temp_profile.m (Script)	2 - 3 - 4 - 5 - 6 -		05 0]; % Ini (@ODEfun,Vsp .),v,y(:,2)) ,'T')	itial values pan,y0);	-	ent variable i.e. Che dependent var			_
Workspace	8 -	xlabel('V(m' axis([0 5 30	1.1.1						
i 97 K 1x97 Kc 1x97	11 12 13 14	title('Temp	profile dow	wn the react	or for	: Adiabatic opera	tion <mark>')</mark>		
	7 dc - Command	Window							۲
	▶ JX >>					script		Ln 10	Col 61

Run the above program to generate the output shown below

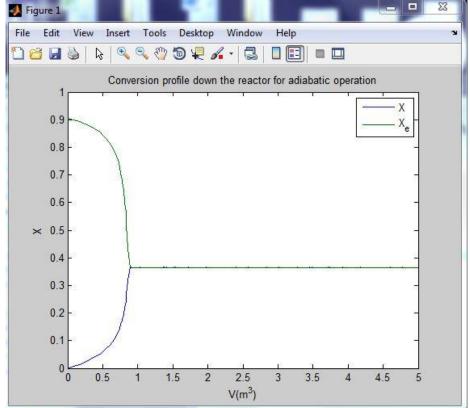


## b) **Conversion profile**

Step	39: '	The	script	file	for	Con	version	profile is	s as	shown	in s	screenshot

📣 MATLAB R2014b		X
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🛃 🖷 🔏 🖆 🛱 🗇 定 🛱 🕐 Search Documentation	<mark>0</mark> 🔺
New Open Save	🖳 🔽 🔁 Run Section	
		<del>-</del> 2
Current Folder  Name  Temp_profile.m Temp_profile.m ODEfun.m Conversion_profile.m Conversion_profile.m Value Conversion_profile.m Cao 1.8600	ODEfun.m         X         Temp_profile.m         X         +           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(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');	⊙ ×
i 97 K 1x97 dc Kc 1x97 dc v 581x1 c Vspan [0,5] Xe 1x97 dc ₹	<pre>14 - xlabel('V(m^3)'); 15 - title('Conversion profile down the reactor for Adiabatic operation') Command Window fs &gt;&gt;</pre>	•
· · · · · · · · · · · · · · · · · · ·	script In 15 Col (	57

The output generated after running the file is

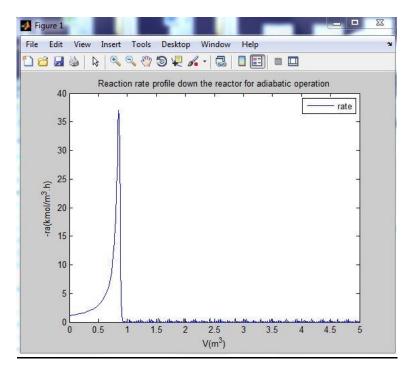


#### c) <u>Rate profile</u>

**Step 40**: The modified Rate\_profile for adiabatic operation is

MATLAB R2014b		-
HOME PLOTS	Image: Section of the section of t	
	s 🔸 foglerlab 🕨 Desktop 🔸 LEP-12-1	<del>,</del> 7
Current Folder   Name	Z Editor - C:\Users\foglerlab\Desktop\LEP-12-1\Rate_profile.m ODEfun.m X Temp_profile.m X Conversion_profile.m X Rate_profile.m X +	⊙ >
<ul> <li>Rate_profile.m</li> <li>ODEfun.m</li> <li>Conversion_profile.m</li> </ul>	<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</pre>	
ODEfun.m (Function)	<pre>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)); 11 - xa(i)=-K(i)*Cao*(1-(1+1/Kc(i))*y(i,3)); 12 - end 13 - plot(v,-ra);</pre>	
Cao 1.8600 ▲ i 581 K 1x581 c Kc 1x581 c r a 1x581 c v 581x1 c Vspan [0,5]	<pre>13 = plot(v,-ra; kmol/m^3.h)'); 14 = ylabel('-ra(kmol/m^3.h)'); 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	(
	script Ln 16 Col	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