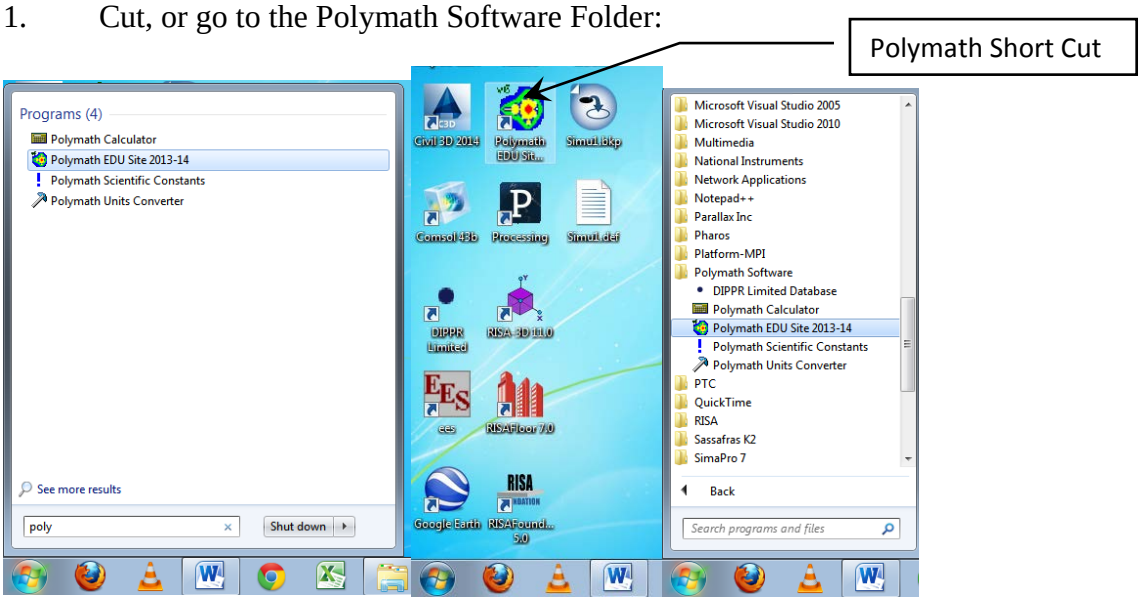
POLYMATH tutorial Objectives: A student will be able to

1. Enter and solve 3 differential equations for a batch chemical reactor problem .
2. Use the built-in dialog box buttons to enter a differential equation and its initial condition. This form is useful since you will not forget to enter the initial condition.
3. Prepare a word document that contains all required information for homework solution. Students must do more than just turn in a polymath program file. They must show how the model equations were derived, answer the questions, produce graphs and sample calculations.
4. How to copy POLYMATH output into an excel spreadsheet such that the produced output has headers in the first row.
5. find the help folder to determine that variable names are case sensitive
6. use the trouble shooting DEQ Message list to determine that a variable has been defined more than once or has not been defined
7. Identify problems that cause a program to stop running such as a divide by zero error
8. Use the comment feature in the polymath program (#)
9. How to use an if - then - else statement

1. Cut, or go to the Polymath Software Folder:



2. Within Polymath, Go to Help, Contents F1 or Press F1

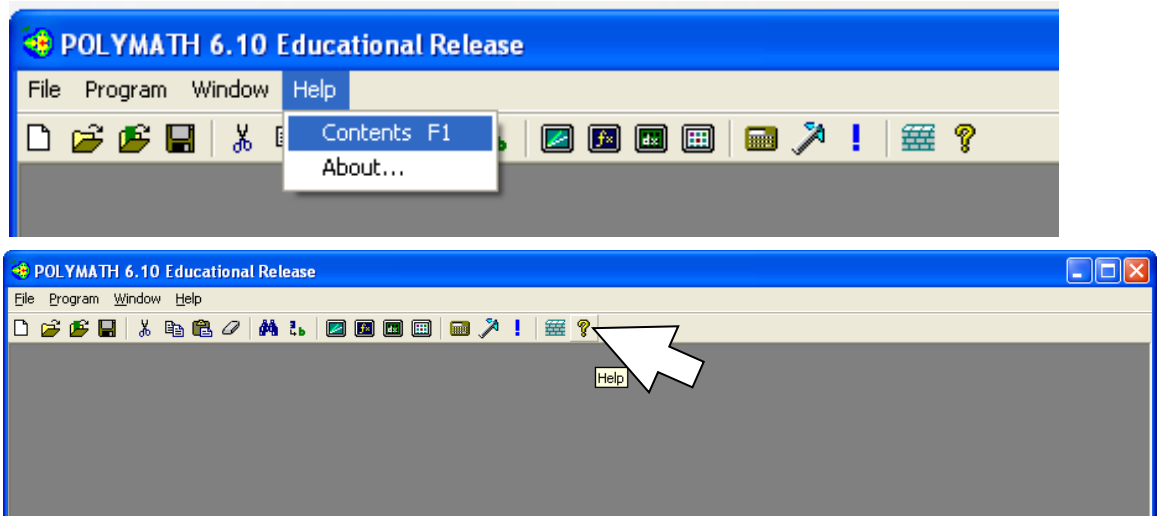
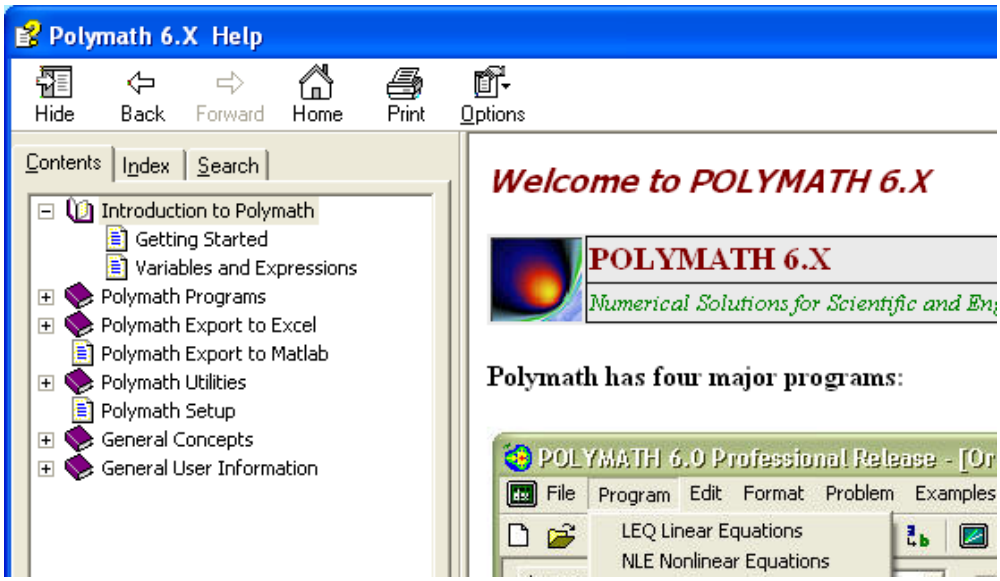
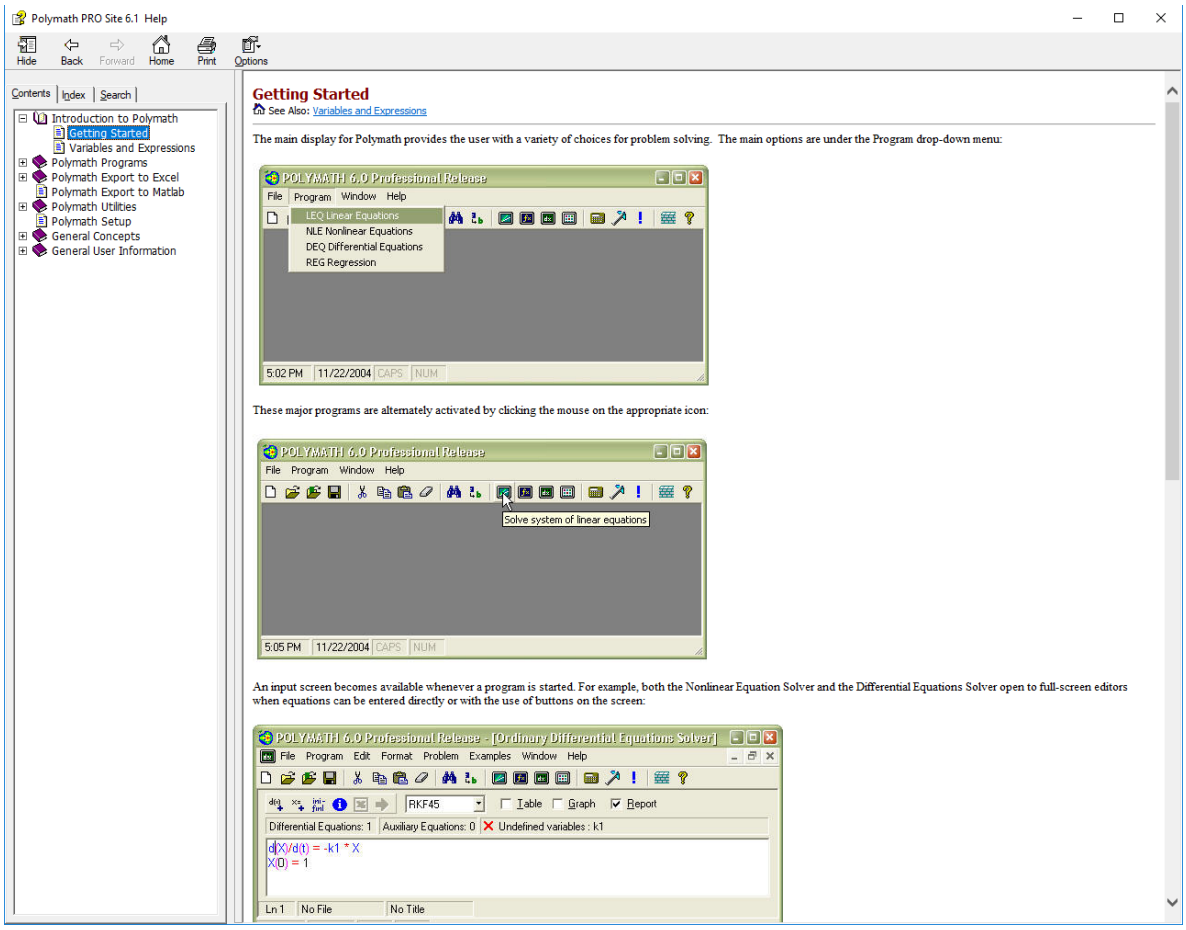
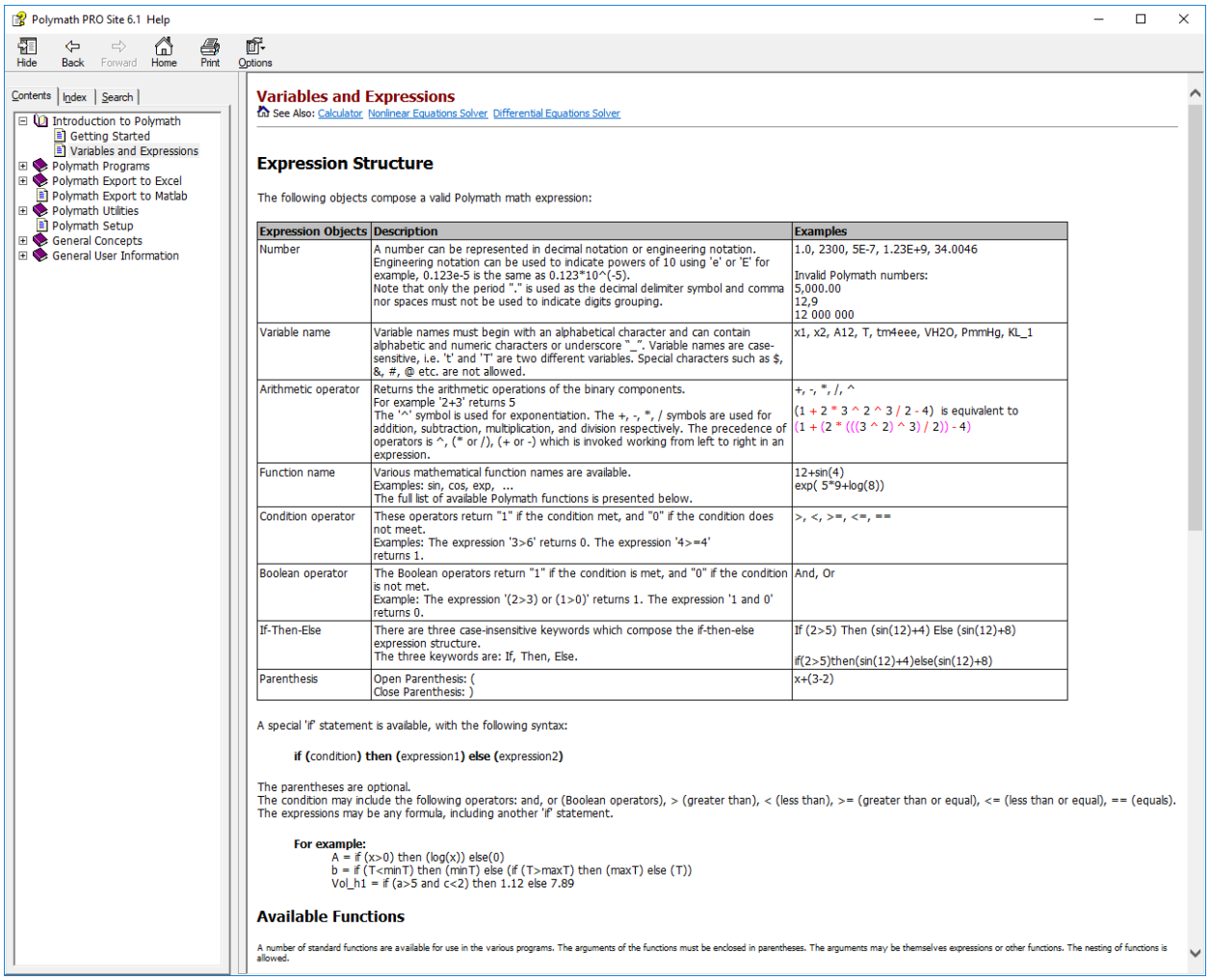


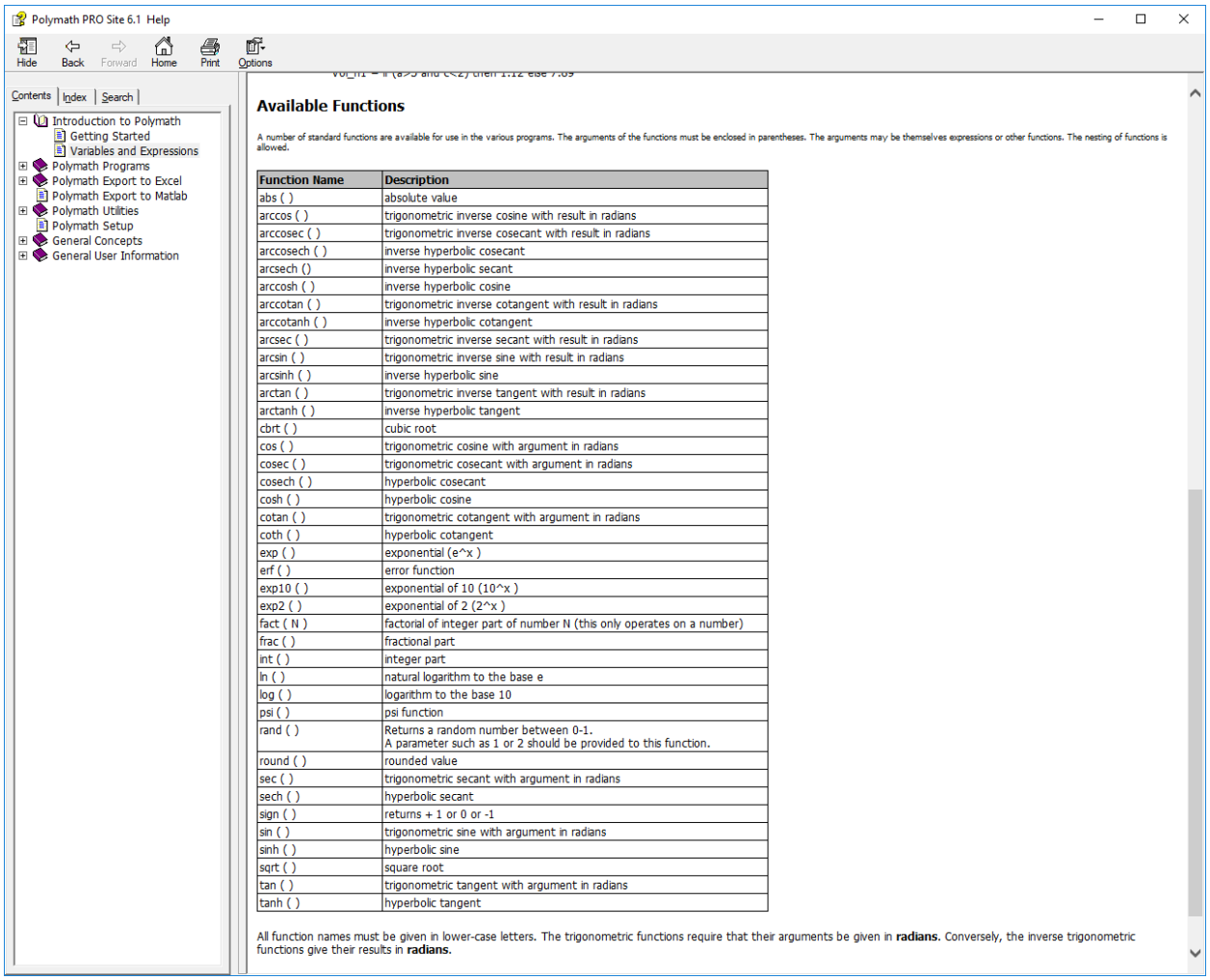
Figure 1: Bringing up help: F1, Help on menu, or selecting the yellow question mark

3. This help section is designed to quickly teach you how to create a POLYMATH program and run the program. Briefly review the section titled *Introduction to Polymath* both *Getting Started* and *Variables and Expressions*:




Polymath PRO Site 6.1 Help
Hide
Back
Forward
Home
Print
Options
Contents
Index
Search
Introduction to Polymath
Getting Started
Variables and Expressions
Polymath Programs
Polymath Export to Excel
Polymath Export to Matlab
Polymath Utilities
Polymath Setup
General Concepts
General User Information
Getting Started
See Also: Variables and Expressions
The main display for Polymath provides the user with a variety of choices for problem solving. The main options are under the Program drop-down menu:
POLYMATH 6.0 Professional Release
File
Program
Window
Help
LEQ Linear Equations
NLE Nonlinear Equations
DEQ Differential Equations
REG Regression
5:02 PM
11/22/2004
CAPS
NUM
These major programs are alternately activated by clicking the mouse on the appropriate icon:
POLYMATH 6.0 Professional Release
File
Program
Window
Help
Solve system of linear equations
5:05 PM
11/22/2004
CAPS
NUM
An input screen becomes available whenever a program is started. For example, both the Nonlinear Equation Solver and the Differential Equations Solver open to full-screen editors when equations can be entered directly or with the use of buttons on the screen:
POLYMATH 6.0 Professional Release - [Ordinary Differential Equations Solver]
File
Program
Edit
Format
Problem
Examples
Window
Help
RKF45
Table
Graph
Report
Differential Equations: 1
Auxiliary Equations: 0
Undefined variables : k1
d(X)/d(t) = -k1 * X
X(0) = 1
Ln 1
No File
No Title


Polymath PRO Site 6.1 Help
Hide
Back
Forward
Home
Print
Options
Contents
Index
Search
Introduction to Polymath
Getting Started
Variables and Expressions
Polymath Programs
Polymath Export to Excel
Polymath Export to Matlab
Polymath Utilities
Polymath Setup
General Concepts
General User Information
Variables and Expressions
See Also: Calculator Nonlinear Equations Solver Differential Equations Solver
Expression Structure
The following objects compose a valid Polymath math expression:
Expression Objects
Description
Examples
Number
A number can be represented in decimal notation or engineering notation. Engineering notation can be used to indicate powers of 10 using 'e' or 'E' for example, 0.123e-5 is the same as 0.123*10^(-5). Note that only the period "." is used as the decimal delimiter symbol and comma nor spaces must not be used to indicate digits grouping.
1.0, 2300, 5E-7, 1.23E+9, 34.0046
Invalid Polymath numbers:
5,000.00
12,9
12 000 000
Variable name
Variable names must begin with an alphabetical character and can contain alphabetic and numeric characters or underscore "_". Variable names are case-sensitive, i.e. 't' and 'T' are two different variables. Special characters such as $, &, #, @ etc. are not allowed.
x1, x2, A12, T, tm4eee, VH2O, PmmHg, KL_1
Arithmetic operator
Returns the arithmetic operations of the binary components. For example '2+3' returns 5 The '^' symbol is used for exponentiation. The +, -, *, / symbols are used for addition, subtraction, multiplication, and division respectively. The precedence of operators is ^, (* or /), (+ or -) which is invoked working from left to right in an expression.
+, -, *, /, ^
(1 + 2 * 3 ^ 2 ^ 3 / 2 - 4) is equivalent to
(1 + (2 * (((3 ^ 2) ^ 3) / 2)) - 4)
Function name
Various mathematical function names are available. Examples: sin, cos, exp, ... The full list of available Polymath functions is presented below.
12+sin(4)
exp(5*9+log(8))
Condition operator
These operators return "1" if the condition met, and "0" if the condition does not meet. Examples: The expression '3>6' returns 0. The expression '4>=4' returns 1.
>, <, >=, <=, ==
Boolean operator
The Boolean operators return "1" if the condition is met, and "0" if the condition is not met. Example: The expression '(2>3) or (1>0)' returns 1. The expression '1 and 0' returns 0.
And, Or
If-Then-Else
There are three case-insensitive keywords which compose the if-then-else expression structure. The three keywords are: If, Then, Else.
If (2>5) Then (sin(12)+4) Else (sin(12)+8)
if(2>5)then(sin(12)+4)else(sin(12)+8)
Parenthesis
Open Parenthesis: (Close Parenthesis:)
x+(3-2)
A special 'if' statement is available, with the following syntax:
if (condition) then (expression1) else (expression2)
The parentheses are optional.
The condition may include the following operators: and, or (Boolean operators), > (greater than), < (less than), >= (greater than or equal), <= (less than or equal), == (equals).
The expressions may be any formula, including another 'if' statement.
For example:
A = if (x>0) then (log(x)) else(0)
b = if (T<minT) then (minT) else (if (T>maxT) then (maxT) else (T))
Vol_h1 = if (a>5 and c<2) then 1.12 else 7.89
Available Functions
A number of standard functions are available for use in the various programs. The arguments of the functions must be enclosed in parentheses. The arguments may be themselves expressions or other functions. The nesting of functions is allowed.



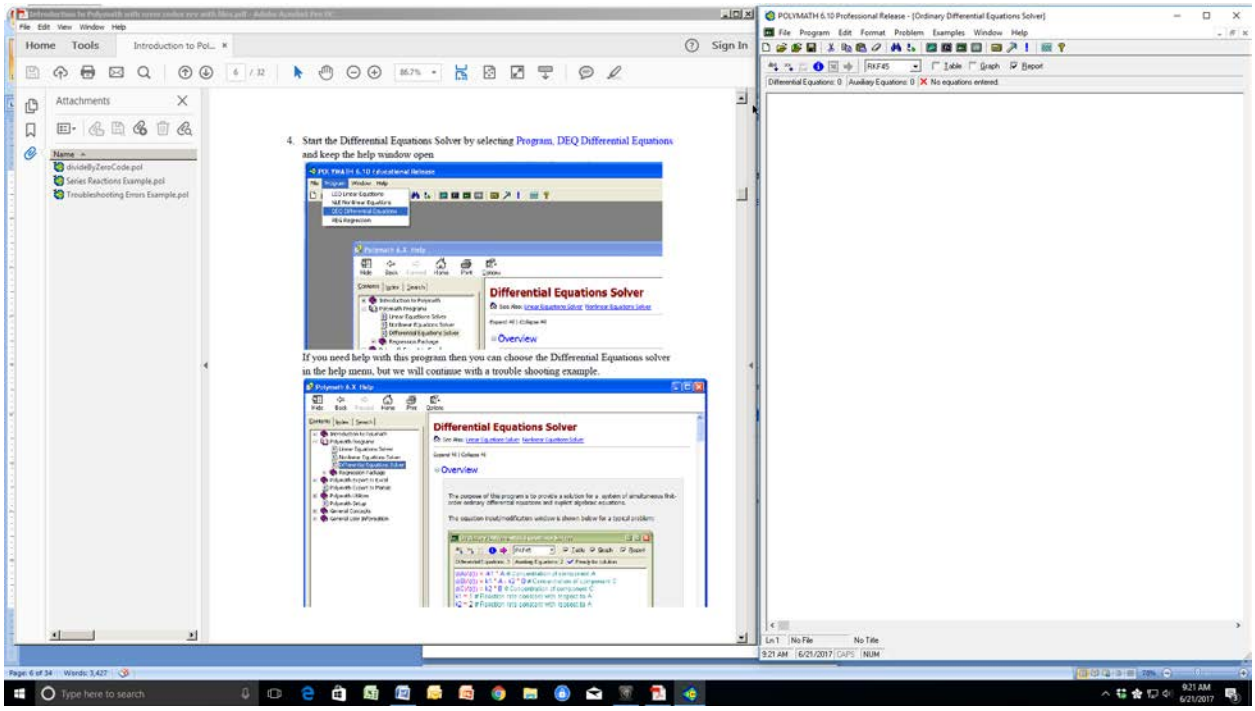
Available Functions

A number of standard functions are available for use in the various programs. The arguments of the functions must be enclosed in parentheses. The arguments may be themselves expressions or other functions. The nesting of functions is allowed.

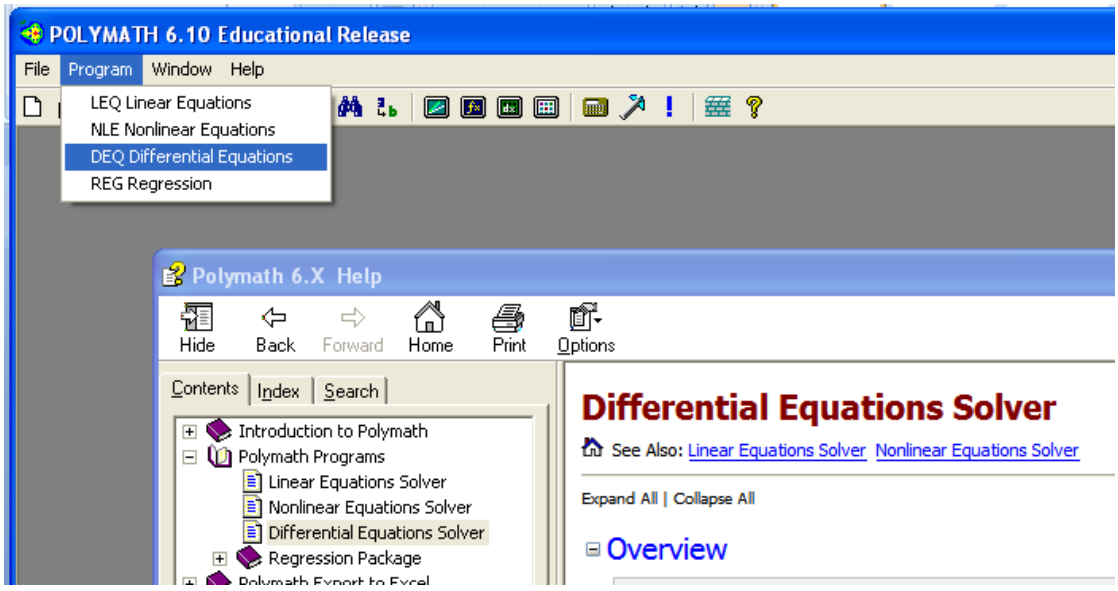
Function Name	Description
abs ( )	absolute value
arccos ( )	trigonometric inverse cosine with result in radians
arccosec ( )	trigonometric inverse cosecant with result in radians
arccosech ( )	inverse hyperbolic cosecant
arcsech ()	inverse hyperbolic secant
arccosh ( )	inverse hyperbolic cosine
arccotan ( )	trigonometric inverse cotangent with result in radians
arccotanh ( )	inverse hyperbolic cotangent
arcsec ( )	trigonometric inverse secant with result in radians
arcsin ( )	trigonometric inverse sine with result in radians
arcsinh ( )	inverse hyperbolic sine
arctan ( )	trigonometric inverse tangent with result in radians
arctanh ( )	inverse hyperbolic tangent
cbrt ( )	cubic root
cos ( )	trigonometric cosine with argument in radians
cosec ( )	trigonometric cosecant with argument in radians
cosech ( )	hyperbolic cosecant
cosh ( )	hyperbolic cosine
cotan ( )	trigonometric cotangent with argument in radians
coth ( )	hyperbolic cotangent
exp ( )	exponential (e^x)
erf ( )	error function
exp10 ( )	exponential of 10 (10^x)
exp2 ( )	exponential of 2 (2^x)
fact (N)	factorial of integer part of number N (this only operates on a number)
frac ( )	fractional part
int ( )	integer part
ln ( )	natural logarithm to the base e
log ( )	logarithm to the base 10
psi ( )	psi function
rand ( )	Returns a random number between 0-1. A parameter such as 1 or 2 should be provided to this function.
round ( )	rounded value
sec ( )	trigonometric secant with argument in radians
sech ( )	hyperbolic secant
sign ( )	returns + 1 or 0 or -1
sin ( )	trigonometric sine with argument in radians
sinh ( )	hyperbolic sine
sqrt ( )	square root
tan ( )	trigonometric tangent with argument in radians
tanh ( )	hyperbolic tangent

All function names must be given in lower-case letters. The trigonometric functions require that their arguments be given in **radians**. Conversely, the inverse trigonometric functions give their results in **radians**.

4. We suggest that you open two windows so that you can see this pdf file and the polymath program as shown below.



5. Start the Differential Equations Solver by selecting Program, DEQ Differential Equations and keep the help window open



Again, if you need help with the Differential Equations Solver program or want more information about the methods used to solve differential equations then you can choose the Differential Equations solver in the help menu, but we will continue with solving a batch reactor problem involving 3 differential equations and then a trouble shooting example.

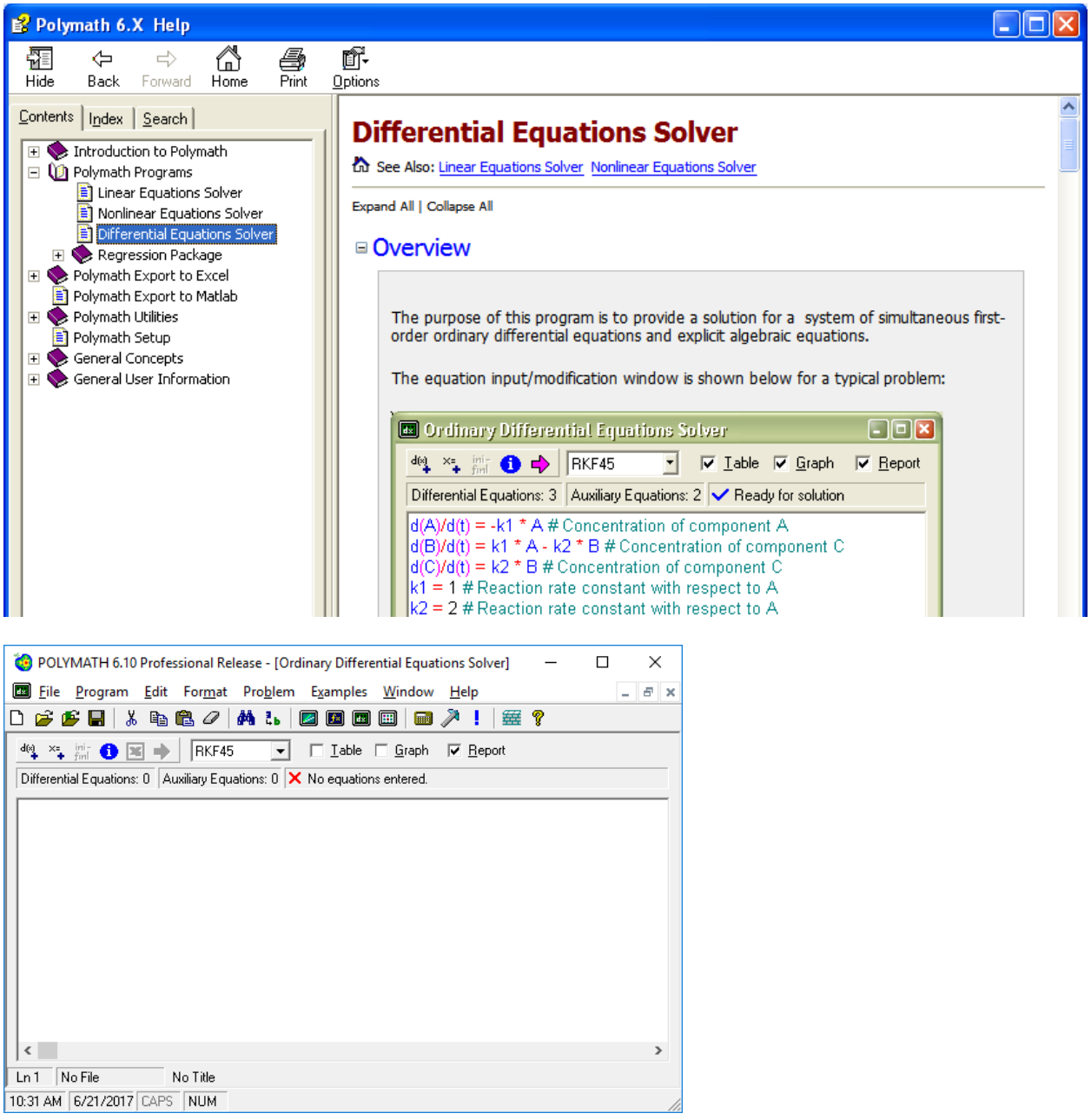


Figure 2: Blank screen of Differential Equation Solver

6. Now you will create a simple POLYMATH file: Series Reactions Example.pol. We will enter the 3 differential Equations and supporting explicit algebraic equations similar to that given in the POLYMATH help example. (The only difference is that concentrations are defined as CA, CB and CC).

This example is based on a batch reactor with 2 simultaneous chemical reactions in series.

$$A \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C$$

A component mole balance is constructed for each chemical species. Since this is a batch reactor, then the mole balances are differential equations. If this problem was an assigned homework problem then the first page of the problem would be a hand written setup of the problem on green engineering paper. This page would contain:

a) Setup of the component species mole balances including a diagram of the process (process flow diagram, pfd)

b) Initial conditions

c) Sample calculations showing that the correct units have been used and an order of magnitude estimate of the results.

For example the mole balances for A, B and C are given by

$$\frac{d(C_A)}{dt} = -k_1 C_A \tag{1}$$

$$\frac{d(C_B)}{dt} = k_1 C_A - k_2 C_B \tag{2}$$

$$\frac{d(C_C)}{dt} = k_2 C_B \tag{3}$$

The initial conditions in the batch reactor at t=0 min are $C_A = 1$ kmol/L, $C_B = 0$ kmol/L and $C_C = 0$ kmol/L. These are known as initial values. The integration will proceed from 0 min to t=3 min. The rate constants are $k_1 = 1\ \text{min}^{-1}$ and $k_2 = 2\ \text{min}^{-1}$.

Sample calculations of all equations are required to be submitted on green engineering paper. These calculations will help you to troubleshoot your program. Sample calculations for explicit equations should be straight forward always showing the number and units. For differential equations I suggest that you show an **order of magnitude** estimate as the sample calculation.

For this example for the batch reactor mole balance using the initial conditions the initial change in concentration of A with time is:

$$\left.\frac{d(C_A)}{dt}\right|_{t=0} = -1\ min^{-1}\left(1\frac{kmol}{L}\right) = -1\ kmol/(L\,\text{min}) \tag{4}$$

An estimate of the value of concentration after 1 minute would be
(NOTICE that this is not a correct integration of the differential equation. This is ONLY an ESTIMATE and a check on the units. To check for order of magnitude changes an assumption is made that the rate is constant.)

$$\int_{C_A=1\ kmol/L}^{C_A} d(C_A) \sim \int_{t=0min}^{1\ min} -1\ kmol/(L\,\text{min})dt = C_A - 1kmol/L = -1\ kmol/L \tag{5}$$

The above result gives the final value of the concentration of A to be zero. In other words if the reaction rate was at 1 kmol/(L min) for 1 minute, then there would be no reactant A left. It is then up to the student doing the problem to evaluate if this reaction rate is what was specified or should the rate be 10 times lower. The above equations do not need to be typed, and can be written by hand and then scanned in B&W using your phone with an free App such as CamScanner. This scan will then be inserted into the word document that will be submitted on Blackboard.

7. Of special note is the usefulness of the wizard menu's in entering a differential equation. This will write the equation using the proper syntax. For example to enter the first differential equation click on the d(x)+ button. d(CA) / d(t) = -k1*CA #Concentration of component A and then the initial condition that concentration of A at t=0 is 1.

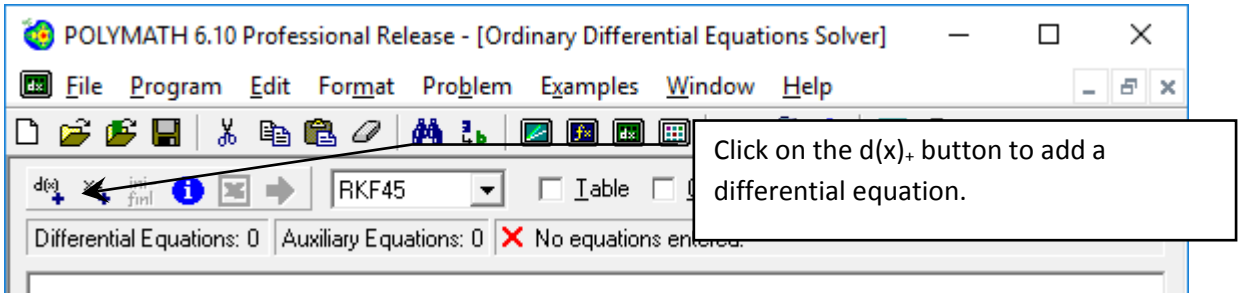


Figure 3: Add a differential Equation

8. Fill out the form shown below and then select done.

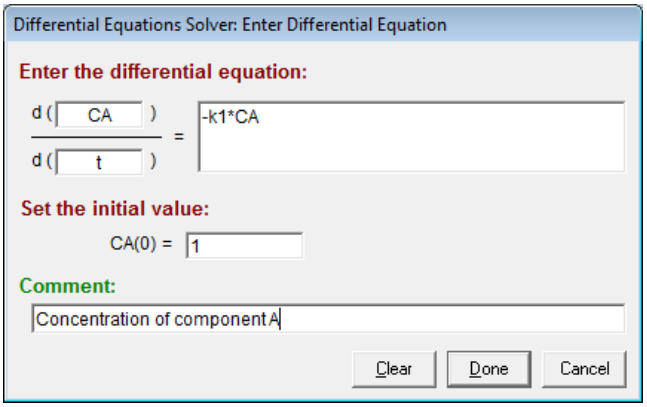


Figure 4: Fill out the menu screen as shown

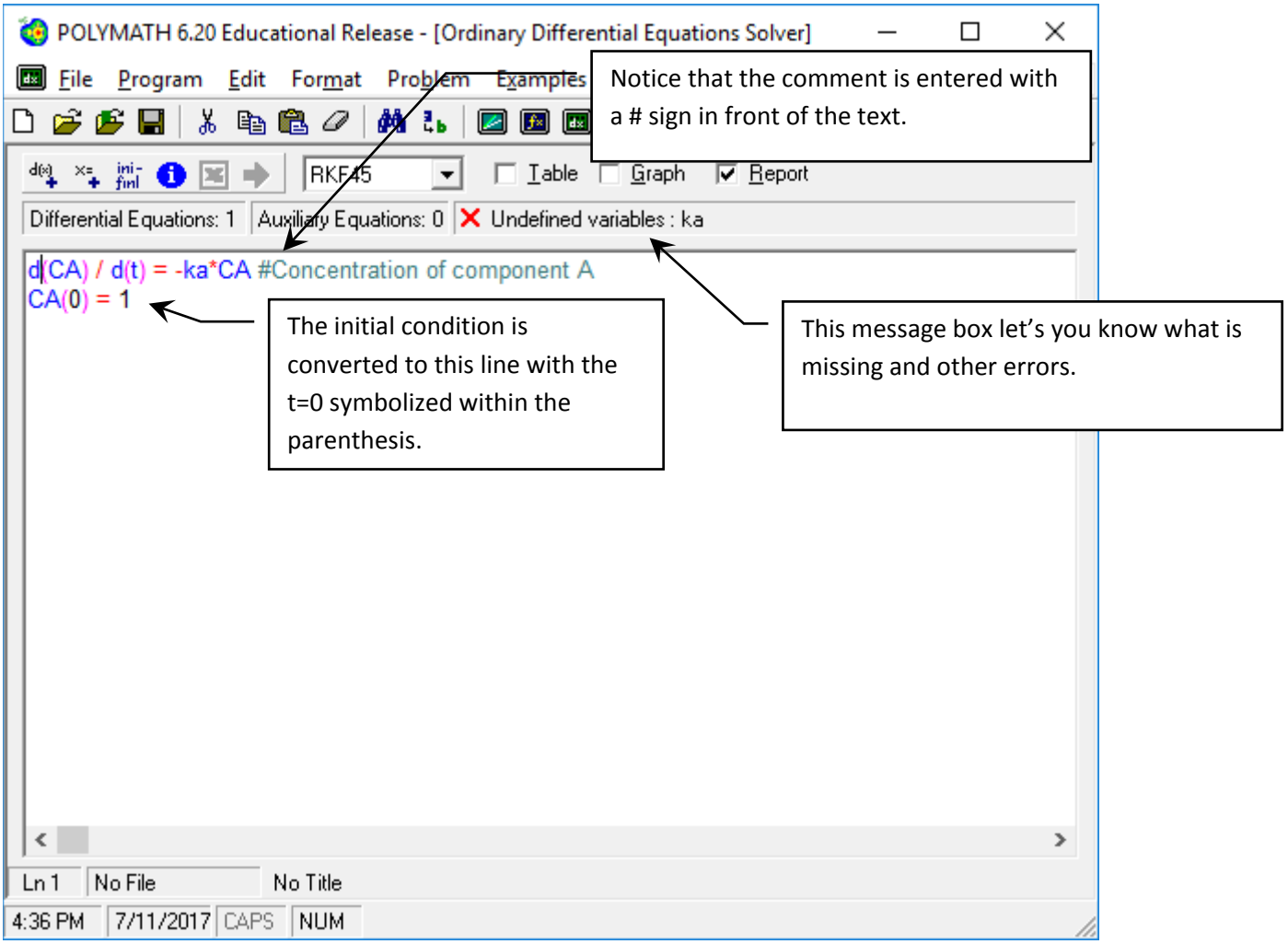
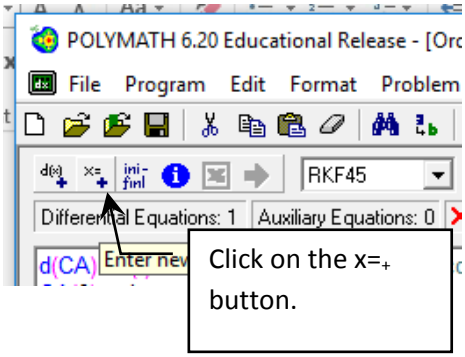


Figure 5: Result of filling out the menu for one differential equation

9. Notice that an error has appeared stating that there is an undefined variable, k1. To remove this error you can add the explicit equation k1=1 min^{-1}. Open up the wizard for an explicit equation



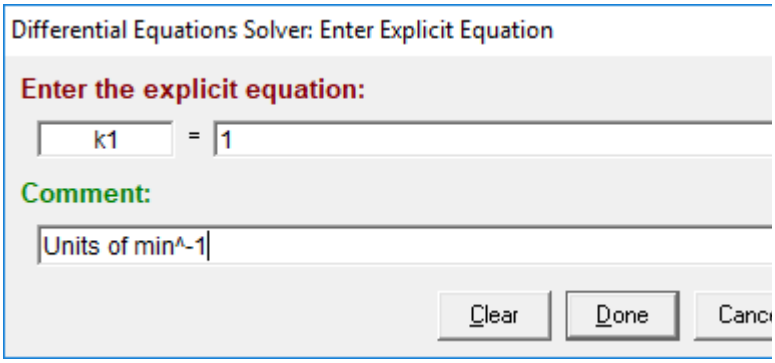


Figure 6: Explicit Equation Entry Form

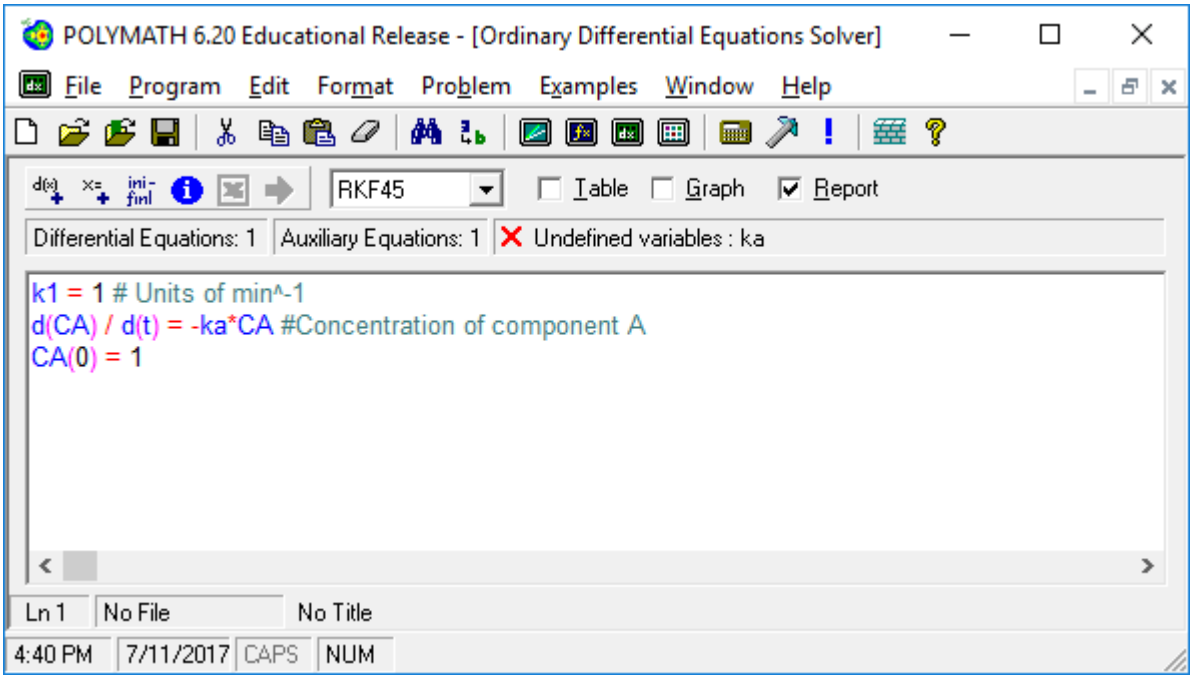


Figure 7: Result of entering explicit equation

10. Enter the second differential equation using the wizard. Notice that the second time that you open this wizard the independent variable of time is already entered.

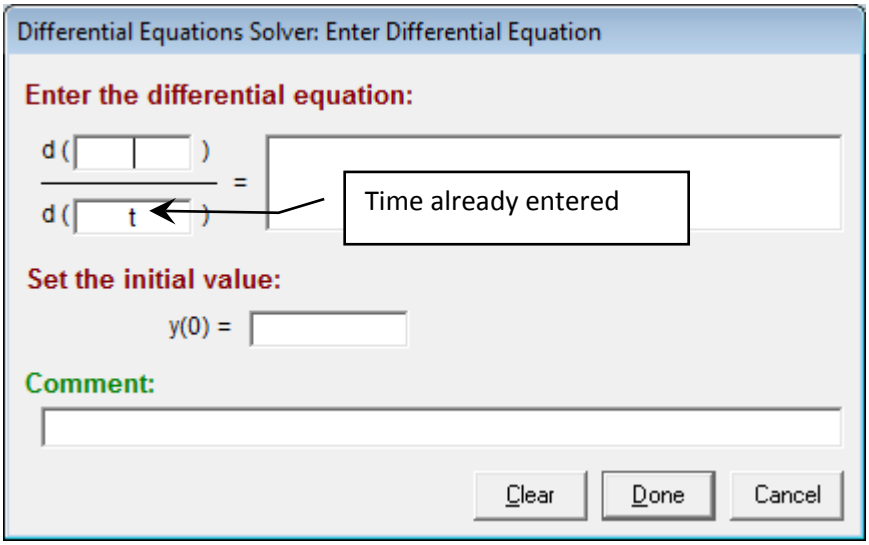


Figure 8: Second use of ODE wizard

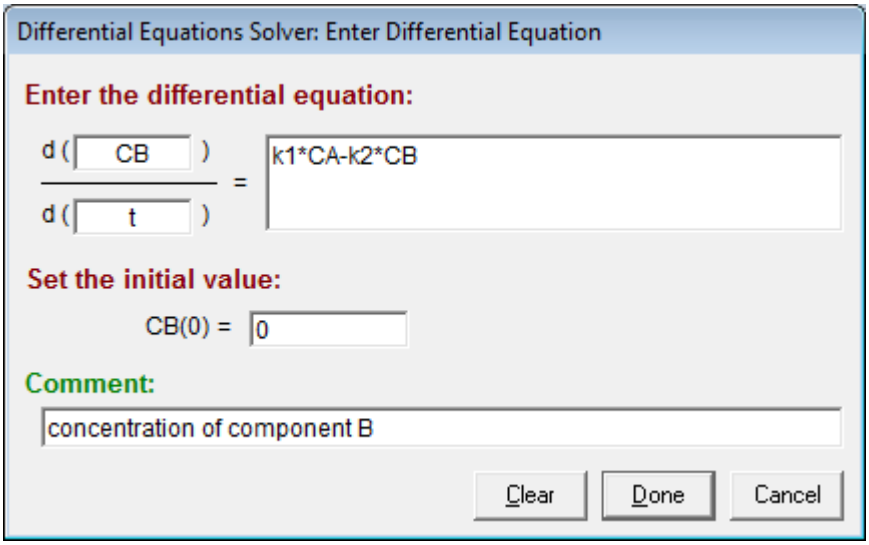
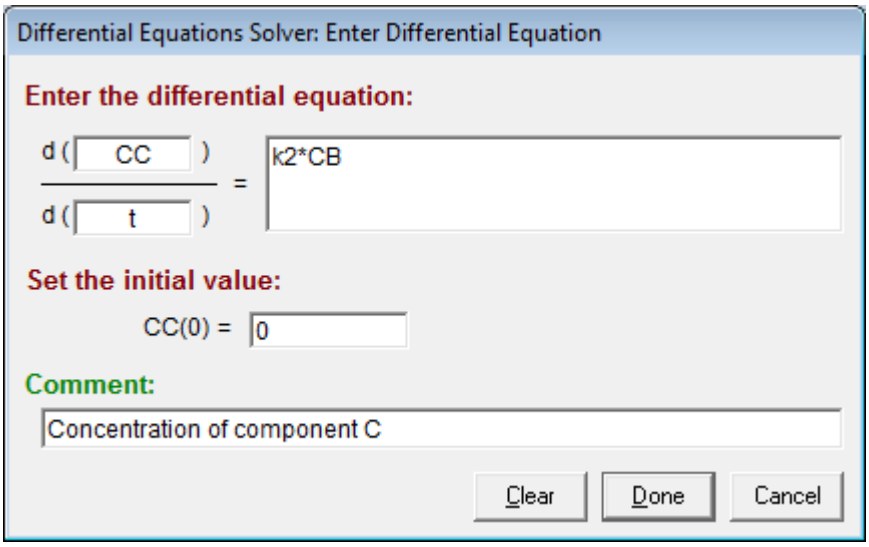
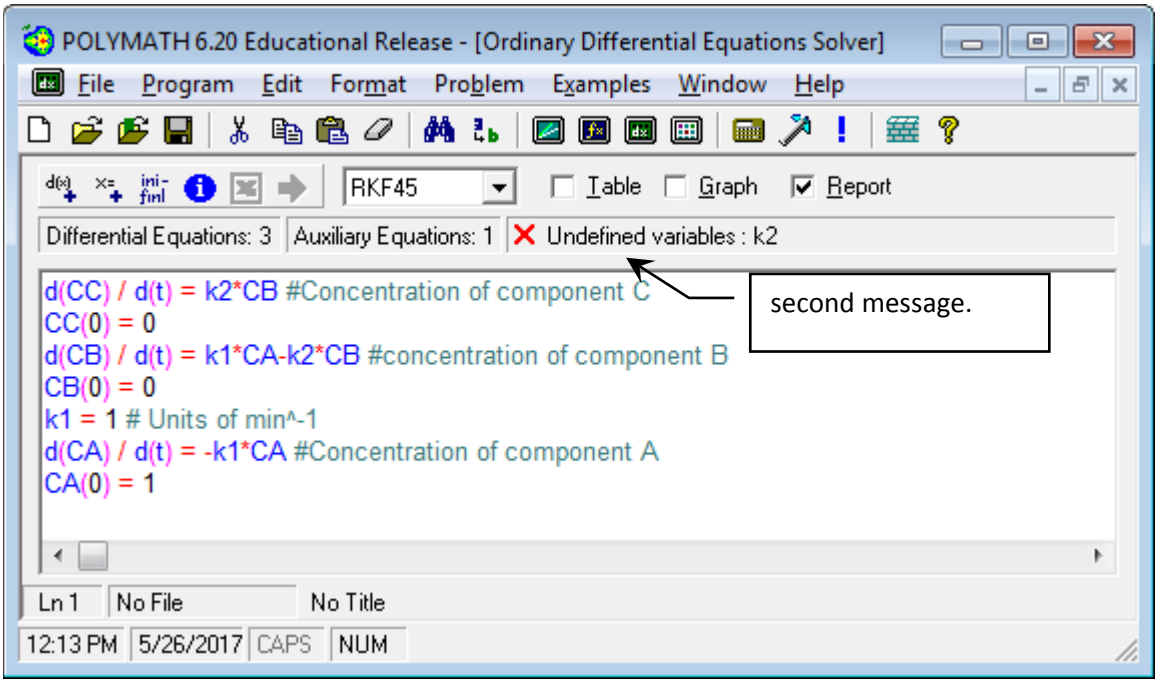


Figure 9: Second ODE

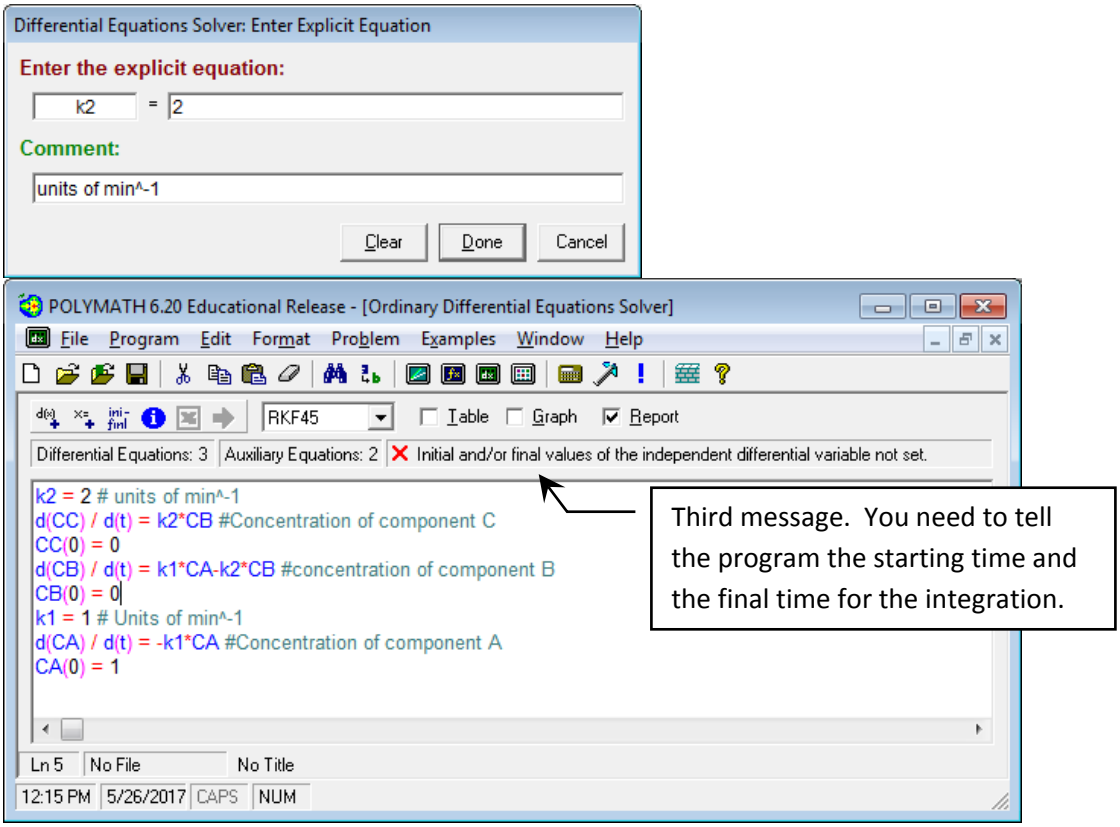
11. Finally enter the 3rd ODE



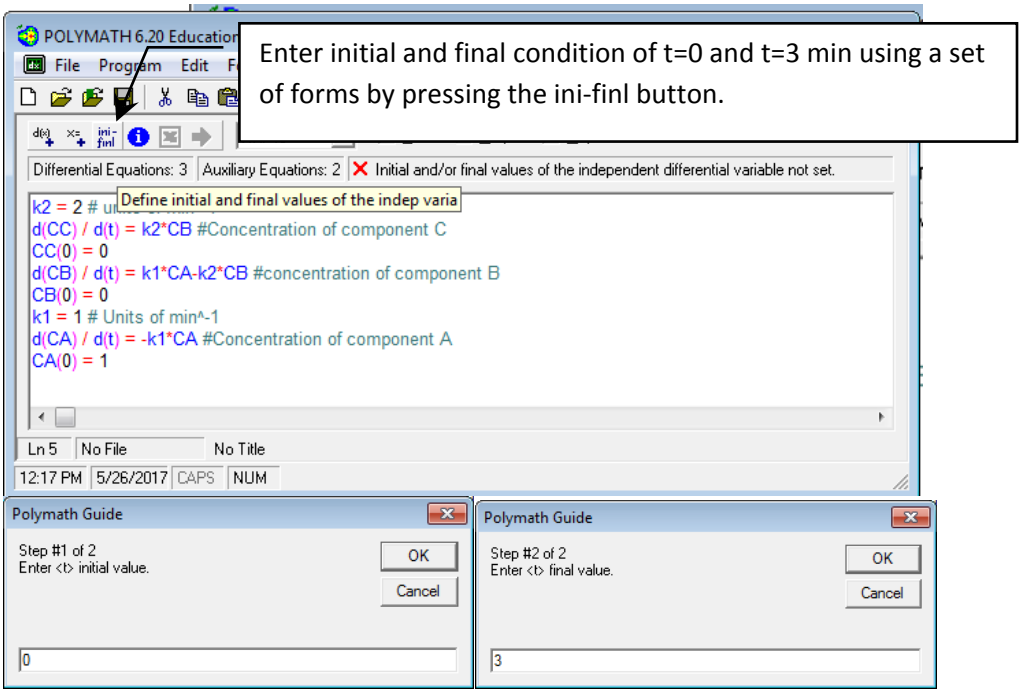
12. The result of entering the 3 ODE's is



13. To remove the error marked above add the value of the second rate constant



14. To remove the final marked error go to the ini-finl button to enter the initial and final values for the integration of the time variable. In this case t=0 to t=3 min.



15. Now the program can be run since the purple arrow appears, but I recommend that you use the Arrange equation feature to order your equations. This will be easier for your professor to troubleshoot and/or grade when put in this order.

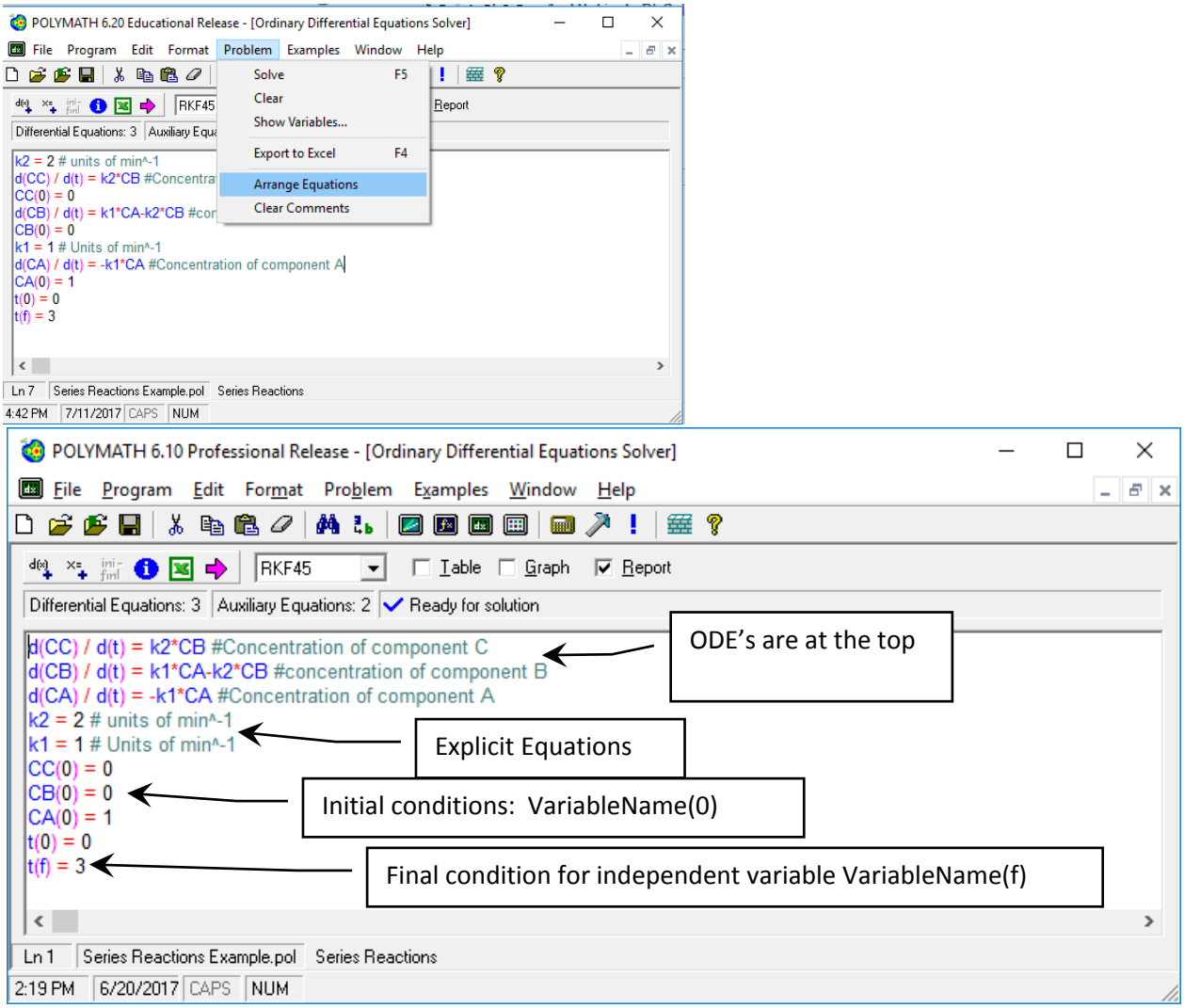
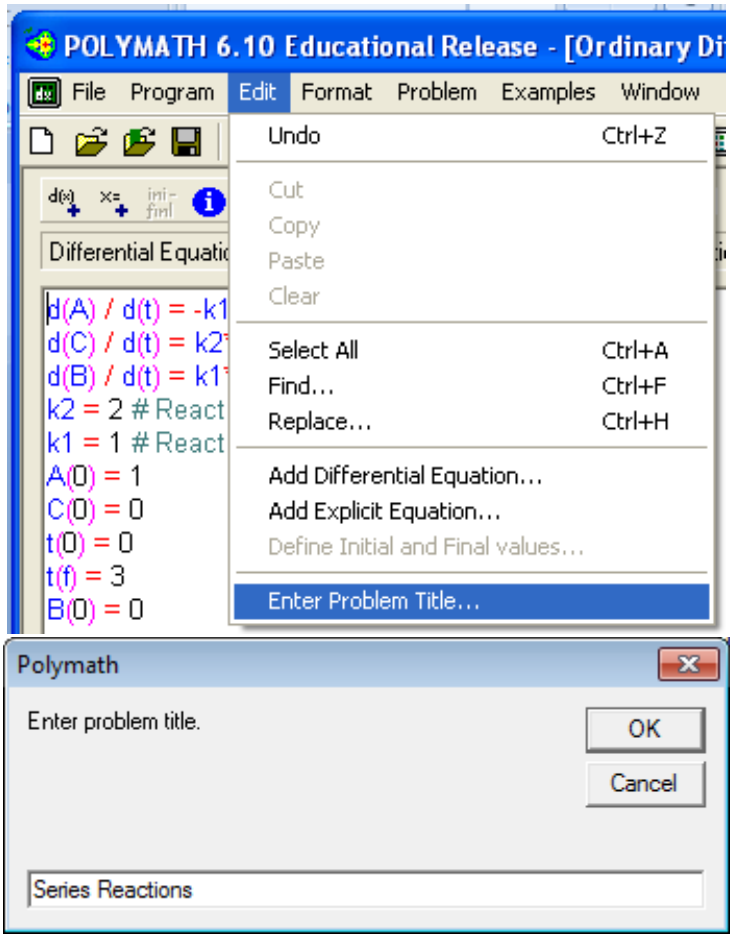
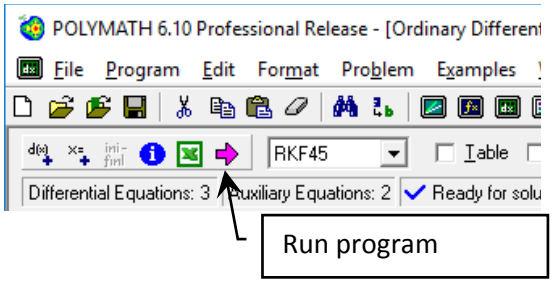


Figure 10: Result of the arrangement of equations (the ordering of the initial conditions may be different to that shown above)

16. Now enter a problem title by selecting Edit, Enter Problem Title…



17. Next save the program with a file name and then run the program by pressing on the pink arrow. The default output is the POLYMATH Report.



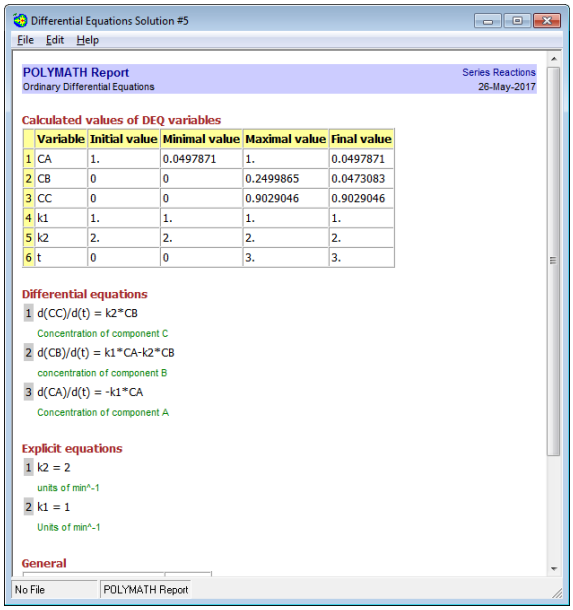
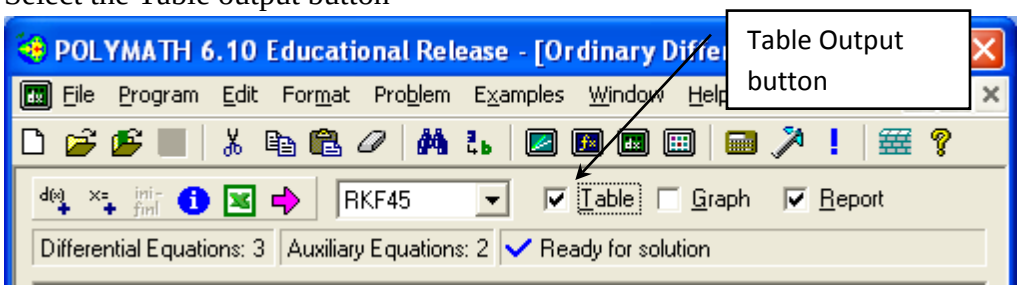


Figure 11: Default output from POLYMATH

18. The preferred method for graphs is to use excel to produce the graph. In this case you should do the following

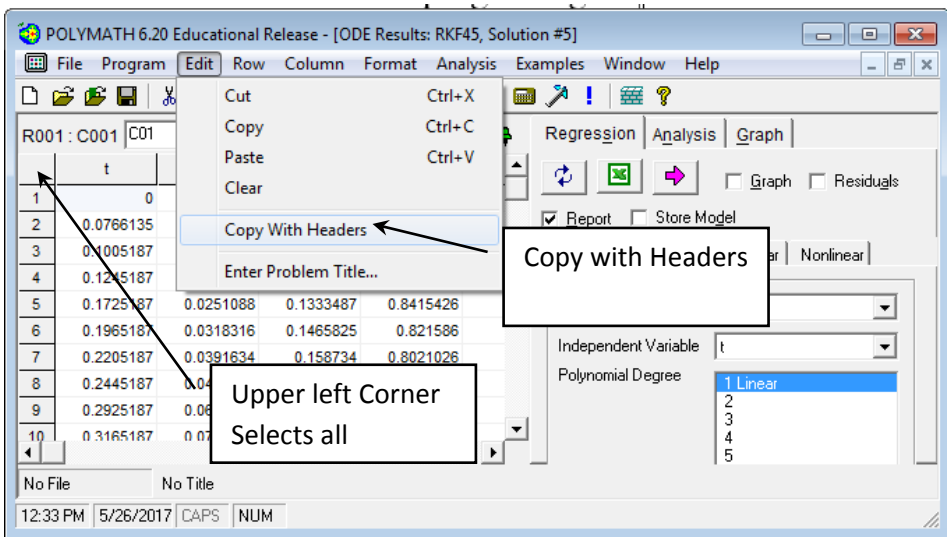
1. Select the Table output button



2. Run the program again

3. Select the table window

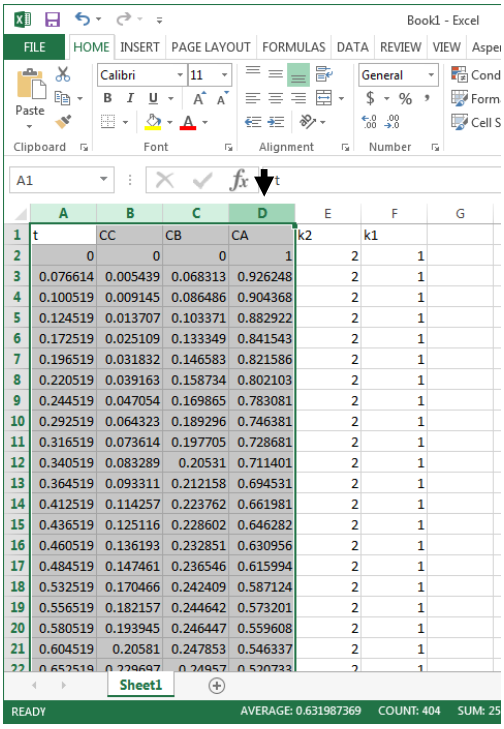
4. Click on the upper left corner of the table (similar to excel)



5. Then select Edit, Copy With Headers. (This will copy the names of the variables as well as the numbers)

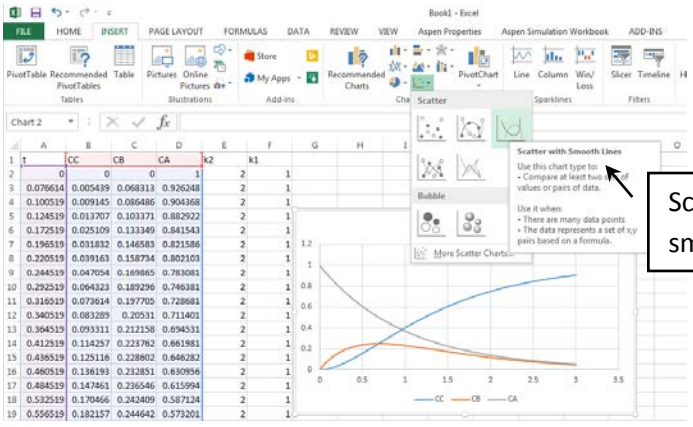
6. Paste this into an excel spreadsheet and produce a graph with all titles given and labels. Notice that for computer generated data, no markers are used. Draw this data using a line and not markers.

1. Select the columns A, B, C and D by holding the left mouse button and dragging over the cell headers of the columns A, B C and D

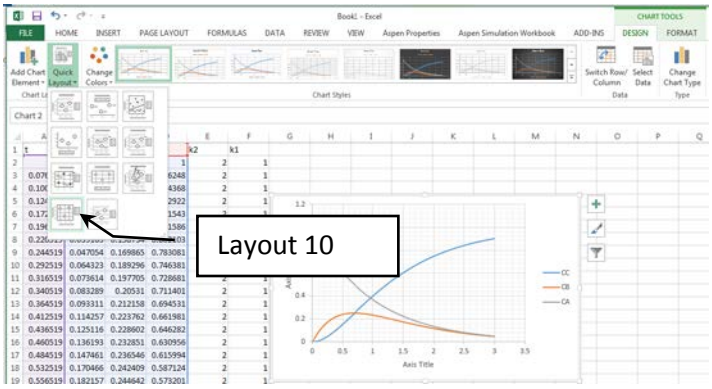


	A	B	C	D	E	F	G
1	t	CC	CB	CA	k2	k1	
2	0	0	0	1	2	1	
3	0.076614	0.005439	0.068313	0.926248	2	1	
4	0.100519	0.009145	0.086486	0.904368	2	1	
5	0.124519	0.013707	0.103371	0.882922	2	1	
6	0.172519	0.025109	0.133349	0.841543	2	1	
7	0.196519	0.031832	0.146583	0.821586	2	1	
8	0.220519	0.039163	0.158734	0.802103	2	1	
9	0.244519	0.047054	0.169865	0.783081	2	1	
10	0.292519	0.064323	0.189296	0.746381	2	1	
11	0.316519	0.073614	0.197705	0.728681	2	1	
12	0.340519	0.083289	0.20531	0.711401	2	1	
13	0.364519	0.093311	0.212158	0.694531	2	1	
14	0.412519	0.114257	0.223762	0.661981	2	1	
15	0.436519	0.125116	0.228602	0.646282	2	1	
16	0.460519	0.136193	0.232851	0.630956	2	1	
17	0.484519	0.147461	0.236546	0.615994	2	1	
18	0.532519	0.170466	0.242409	0.587124	2	1	
19	0.556519	0.182157	0.244642	0.573201	2	1	
20	0.580519	0.193945	0.246447	0.559608	2	1	
21	0.604519	0.20581	0.247853	0.546337	2	1	
22	0.652519	0.229697	0.24957	0.520733	2	1	

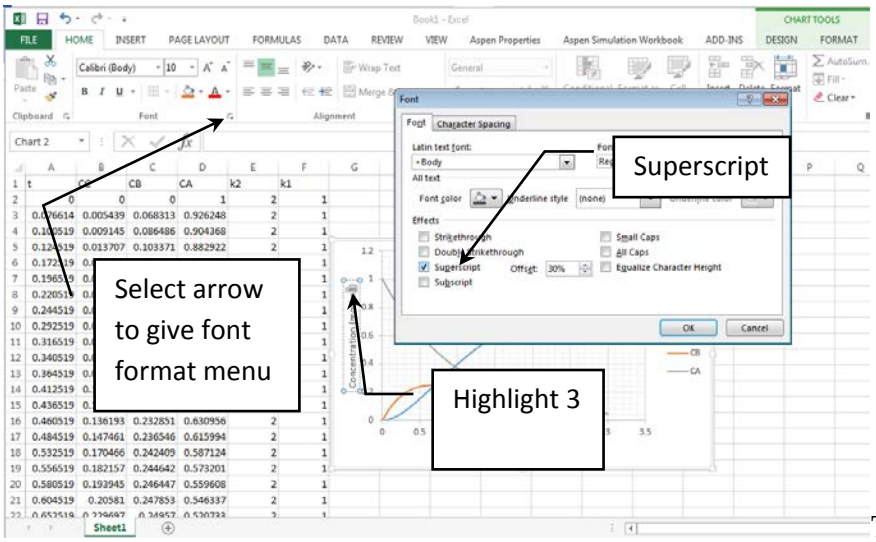
2. Next choose insert scatter with smooth lines



Scatter chart with smooth lines

3. Now add labels for the y and x axis. I usual start with one of the premade templates. 

4. Add a y-axis label of Concentration (mol/m^3) For the superscript on the y-axis label "Concentration (mol/m^3)" highlight the number 3 and then go to the home tab and click on the arrow to open the Font submenu and then select super script



Type

5. The x-axis label is time (minutes).

6. Copy and Paste the graph into the word document and format the text wrapping of the graph to either be tight or in front of text.

7. Right click on the graph and chose Insert Caption and type the title "Concentration Profile of a Series Reaction". Remember to never use the y and x axis labels as the title.

Now the output that you would submit for a grade is shown on the next page:

Calculated values of DEQ variables

	Variable	Initial value	Minimal value	Maximal value	Final value
1	CA	1.	0.0497871	1.	0.0497871
2	CB	0	0	0.2499865	0.0473083
3	CC	0	0	0.9029046	0.9029046
4	k1	1.	1.	1.	1.
5	k2	2.	2.	2.	2.
6	t	0	0	3.	3.

Differential equations

1 d(CC)/d(t) = k2*CB

Concentration of component C

2 d(CB)/d(t) = k1*CA-k2*CB

concentration of component B

3 d(CA)/d(t) = -k1*CA

Concentration of component A

Explicit equations

1 k2 = 2

units of min^-1

2 k1 = 1

Units of min^-1

General

Total number of equations	5
Number of differential equations	3
Number of explicit equations	2
Elapsed time	0.000 sec
Solution method	RKF_45
Step size guess. h	0.000001
Truncation error tolerance. eps	0.000001

This is what the electronic part of your homework should look like! But remember to also submit the hand derivation of the equations used in the POLYMATH model and answer any questions that were asked. For example the concentration of A, B, and C at t=3 s.

Data file: h:\documents\cache\aseeworkshop\series reactions example.pol

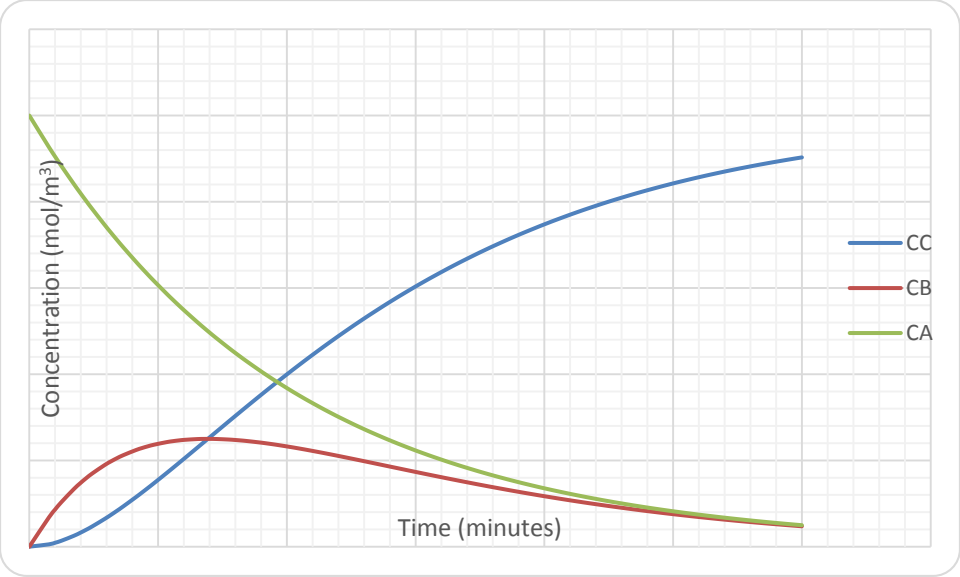


Figure 12: Concentration Profiles of a Series Reaction

Troubleshooting Example

As you can see POLYMATH is a very easy program to use. In the previous batch reactor example you probably produced a POLYMATH Report in less than 5 minutes (or if you did it a second time it would be even shorter). Some of the problems that you solve will be more complex than this and it is useful to see how to avoid errors in placing your model equations into polymath. Of course one of the biggest errors that students make is that they don't write out the equations. In addition they don't place numbers with units in them to make sure that all the units are consistent. This is probably the biggest error done by students. Numbers and units!!! Unfortunately POLYMATH can not help the student with errors in units other than putting the units in comment fields. What follows is the errors that POLYMATH can identify.

19. Common errors by students in using Polymath software:

1. What the user thinks are the same variables, but the computer uses them as different variables. This happens when you incorrectly spell a variable name or you do not match upper or lower cases. (e.g. $Tau \neq tau$)

2. Defining a variable more than once. Once a variable is on the left hand side of the equals sign then it is considered defined by the program.

3. Dividing by zero

4. Using too many parenthesis. POLYMATH uses the standard order of operators which is exponent, multiplication/division, addition/subtraction: ^, (* or /), (+ or -) which is invoked working from left to right in an expression.

Cut and paste the following program in the POLYMATH ODE Solver or load the program troubleshooting errors example.pol. Examine each of these errors and then correct them as directed in the error explanation below:

```
d(vtheta) / d(r) = vtheta/r-tau/mu
d(gamma) / d(r) = 0
tau = Gamma/r^2
tau=2
vtheta =omega*R1
r1=0.1
r(0) =0.1
r(f) = 0.12
vtheta(0) =0
gamma(0) =-6.5455E-06
error = (0.012-vtheta)/.012*100
```

20. To see the errors in your program look in the message area or alternatively to see a full list: Select Problem from the Polymath menu and then Show Variables… to see the following errors in this program. Alternatively you could just press on the information button 

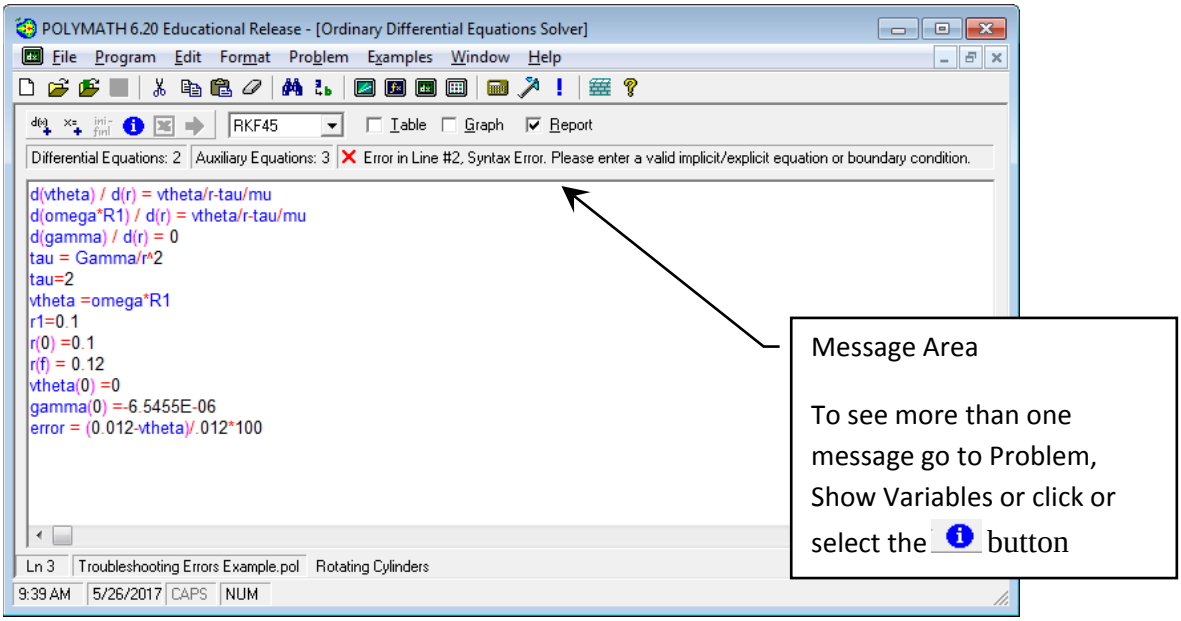


Figure 13: Original Code

Figure 14: Request to show all of the errors by selecting Problem and then Show Variables

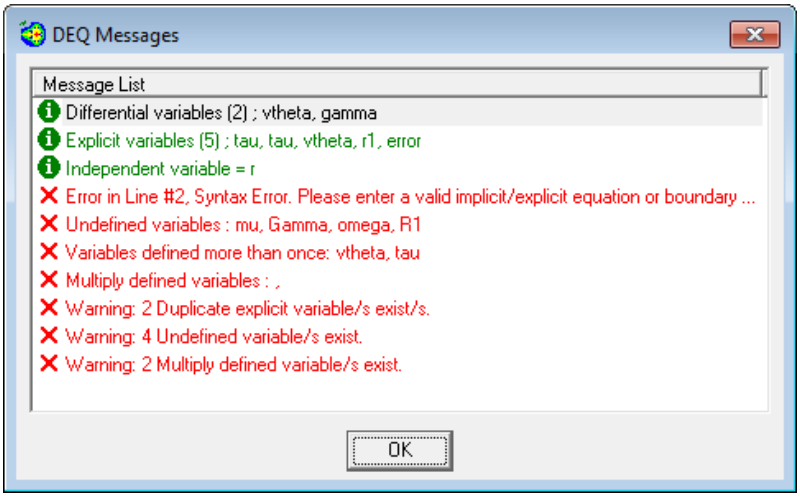


Figure 15: Original listing of variables and errors

Explanation of Errors in Message:

Differential variables (2); vtheta, gamma There are three ordinary differential equations, but one of the equations has a syntax error and is not recognized. See below.

Explicit variables (5); tau, tau, vtheta, r1, error There are 5 explicit (variable =) equations. Notice that it has tau twice which means that it was defined more than once as an explicit equation. This is an error that is also mentioned in the "Multiple (This is a spelling error!) defined variables"

Independent variable = r The ODE's are with respect to only one variable and that variable is r. This is OK

21. Now for the RED X's **X** Error in Line #2, Syntax Error. Please enter a valid implicit/explicit equation or boundary condition. d(omega*R1) / d(r) = vtheta/r-tau/mu The error in this equation is the use of an operator on the left hand side which is not allowed. If this equation was required you would need to define a new variable: junk=omega*R1. The correct form is using the line #1 ODE so to correct this error we will delete line 2.

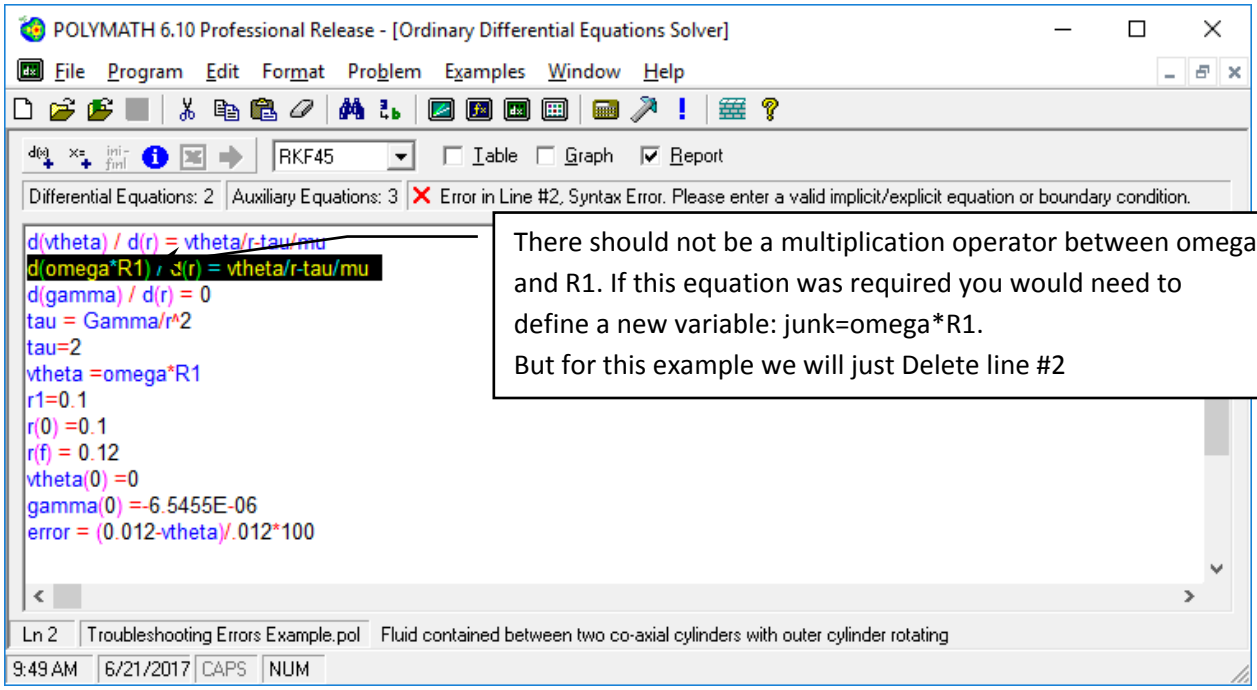


Figure 16: Error in line #2

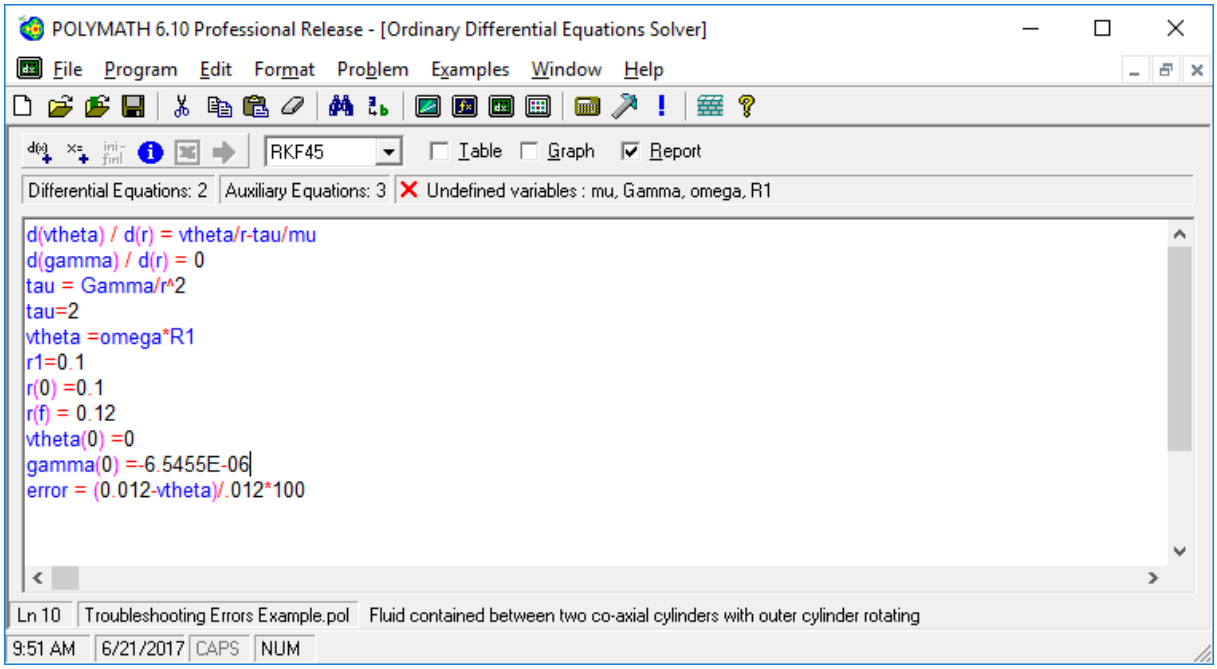


Figure 17: Resulting code after deleting line #2 from above. To see this code you need to click on another line and the message box will update to the next error.

22. **X** Undefined variables : mu, Gamma, omega, R1 The variable mu is used in line 1, but not defined. **You need to add an explicit equation mu=0.001**. The reason that Gamma is undefined is that it never appears on the left hand side of the equal sign, but the variable gamma does! **To correct this error change the case of the g in Gamma**. The remaining undefined variables we will correct in the next error.

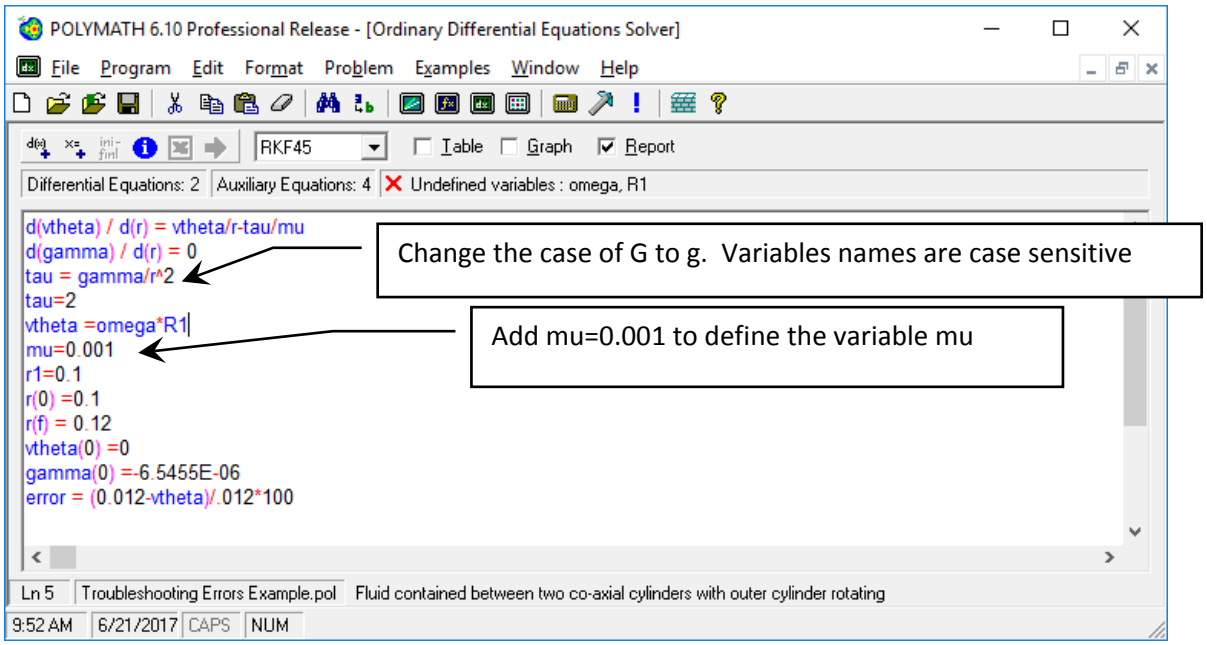


Figure 18: Result after adding definition of mu and change case of G to g

23. **X** Variables defined more than once: vtheta, tau There are two equations that have tau = on lines 3 and 4. The variable vtheta has been defined twice; once in line #1 as an ordinary differential equation and the second time in line #5 shown below. Equation 5 (vtheta =omega*R1) must be deleted in this problem. A variable is defined by having it on the left hand side of the equal sign. Additionally there can only be one variable on the left hand side of an equal sign.
Also on line #4 **you should delete the equation (**tau = 2**)** since it is already defined in the equation above it (#3).

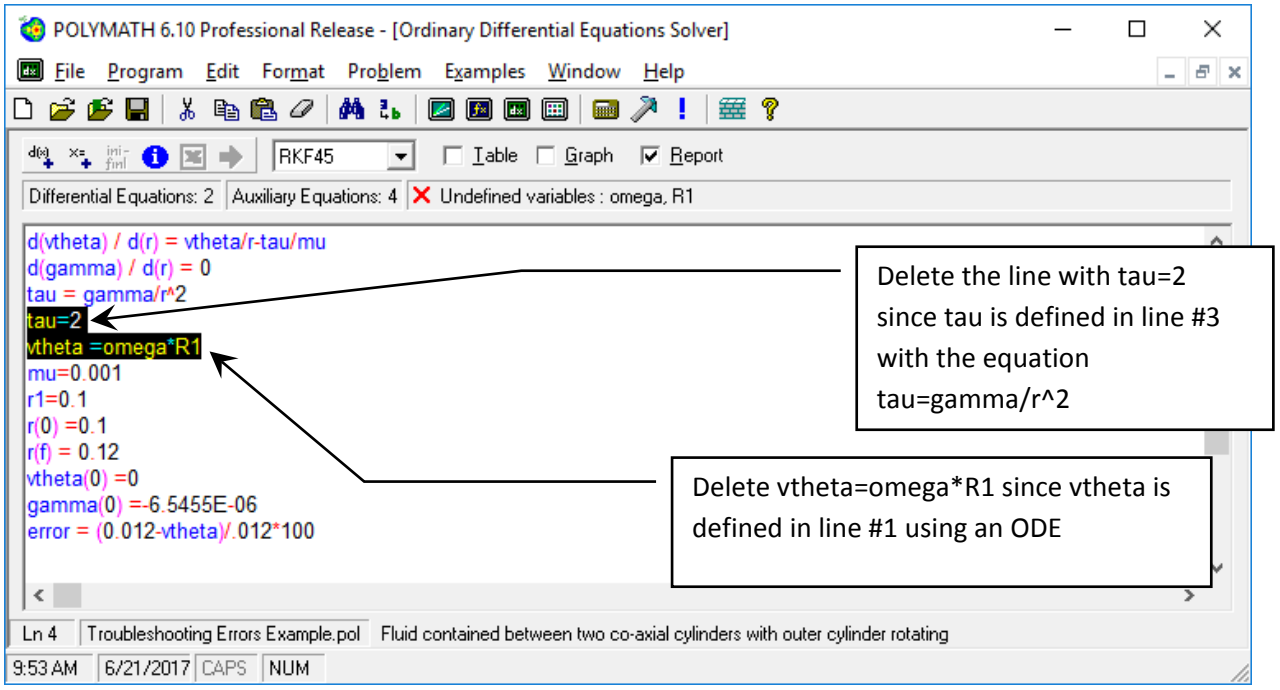


Figure 19: tau and vtheta corrections

24. **Now the program is ready to run:**

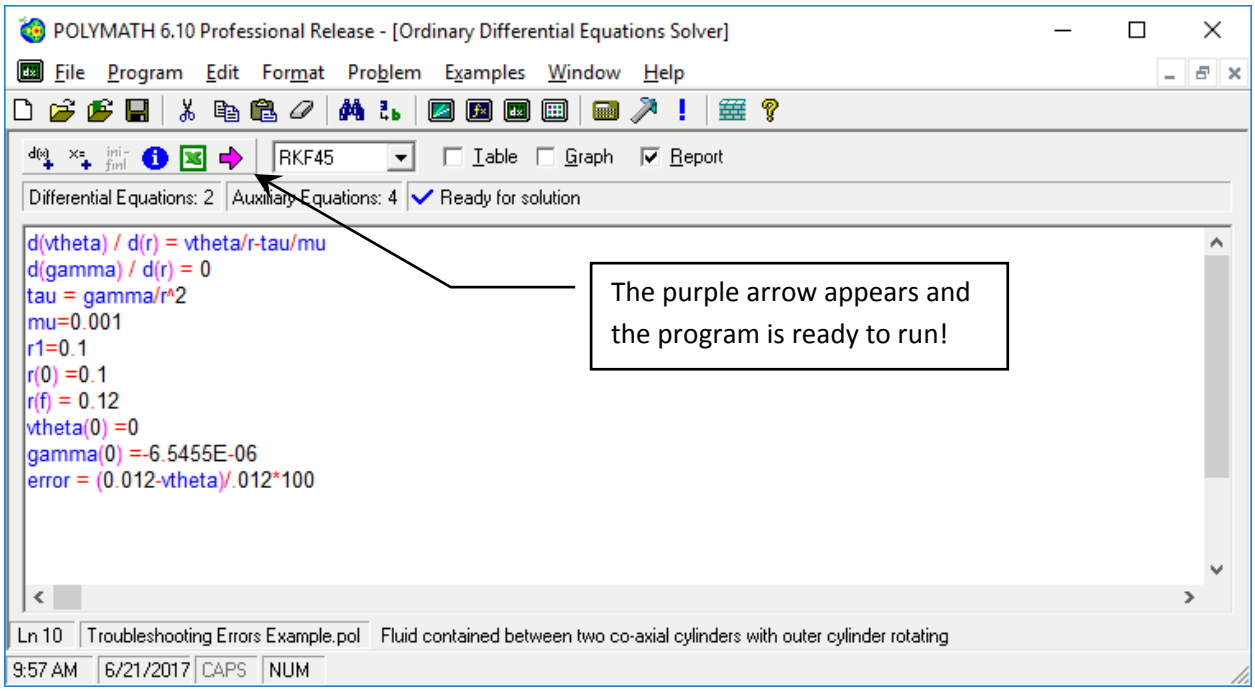


Figure 20: Program is ready to run

25. The following errors from the original code were fixed by following the steps above:

X Multiply (sp Multiple) defined variables : There are 2 variables (tau and vtheta) defined more than once that are involved in multiplication (This seems to be a strange error code). You already deleted the equation on line 4 **(**tau = 2**)** since you can only define it once.

X Warning: 2 Duplicate explicit variable/s exist/s. Again these are vtheta, tau.

X Warning: 4 Undefined variable/s exist. This is also giving the number of variables undefined as 4.

X Warning: 2 Multiply defined variable/s exist. Again a repeat of the above.

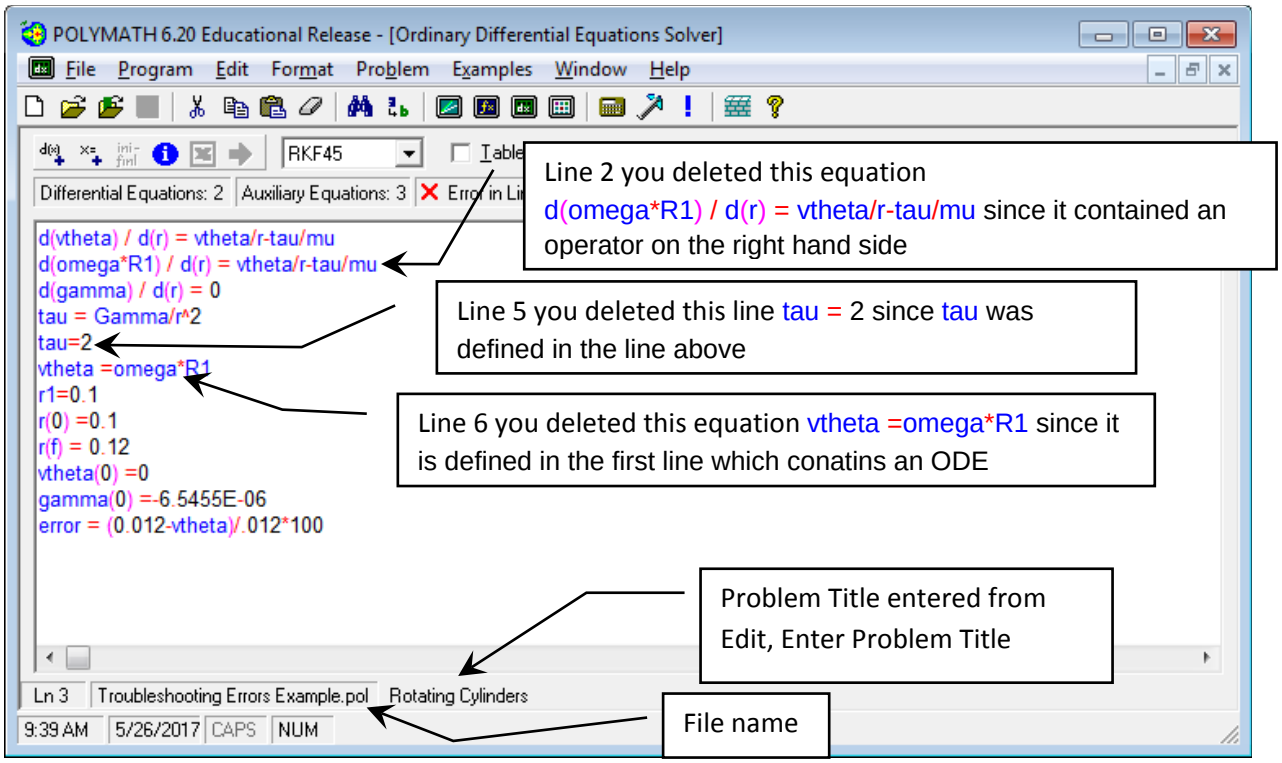


Figure 21: Original program will errors

26. One last step is to have POLYMATH automatically order your equations. This is optional but very useful for the professor to grade your homework. Go to the menu and select Problem and then Arrange equations.

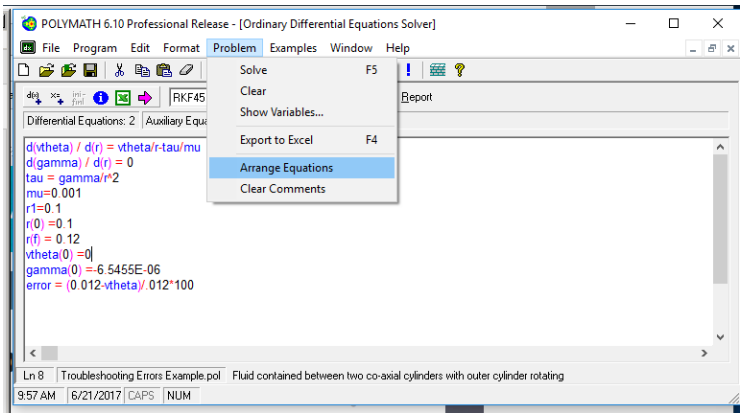
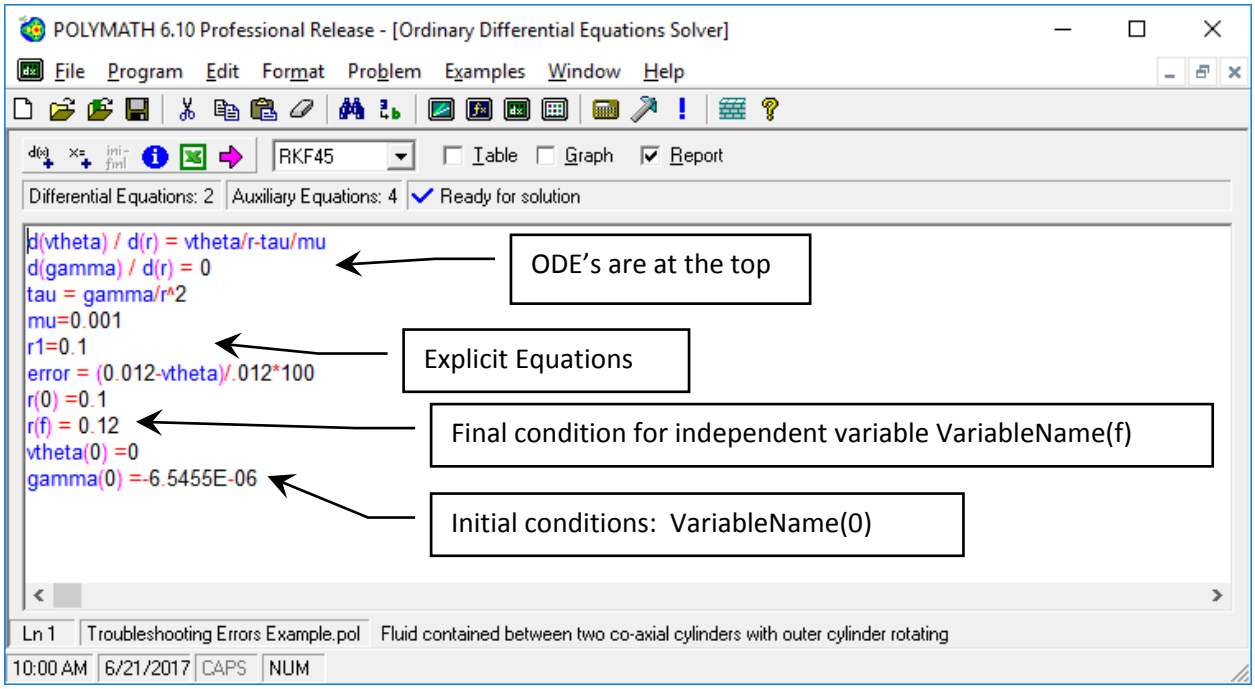
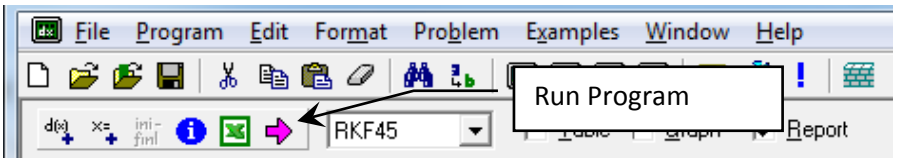


Figure 22: Arrange Equations

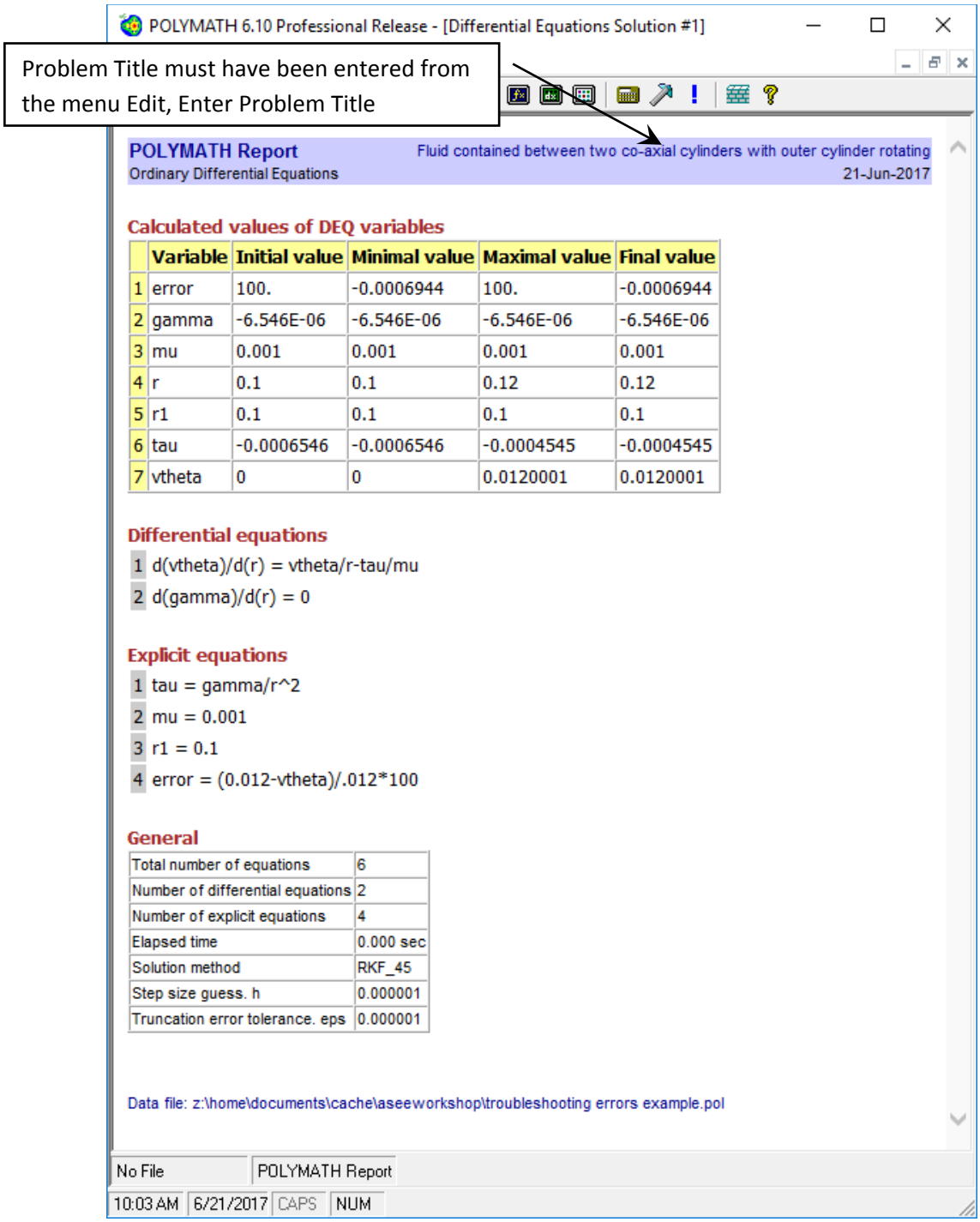
27. The result after arranging equations will put the ODE's at the top, followed by explicit eqautions and then initial and final conditions as shown below.



28. Now run your corrected program by pressing the Pink arrow



29. Your program should run and produce an output page like the following:



For homework assignments you will be required to copy and paste this page into a word document that will also contain answers to questions, graphs and sample calculations that will be uploaded to Blackboard.

Calculated values of DEQ variables

	Variable	Initial value	Minimal value	Maximal value	Final value
1	error	100.	-0.0006944	100.	-0.0006944
2	gamma	-6.546E-06	-6.546E-06	-6.546E-06	-6.546E-06
3	mu	0.001	0.001	0.001	0.001
4	r	0.1	0.1	0.12	0.12
5	r1	0.1	0.1	0.1	0.1
6	tau	-0.0006546	-0.0006546	-0.0004545	-0.0004545
7	vtheta	0	0	0.0120001	0.0120001

Differential equations

1 d(vtheta)/d(r) = vtheta/r-tau/mu

2 d(gamma)/d(r) = 0

Explicit equations

1 tau = gamma/r^2

2 mu = 0.001

3 r1 = 0.1

4 error = (0.012-vtheta)/.012*100

General

Total number of equations	6
Number of differential equations	2
Number of explicit equations	4
Elapsed time	0.000 sec
Solution method	RKF_45
Step size guess. h	0.000001
Truncation error tolerance. eps	0.000001

Data file: z:\home\documents\cache\aseeworkshop\troubleshooting errors example.pol

Remember to always enter a problem title by selecting Edit, Enter Problem Title…

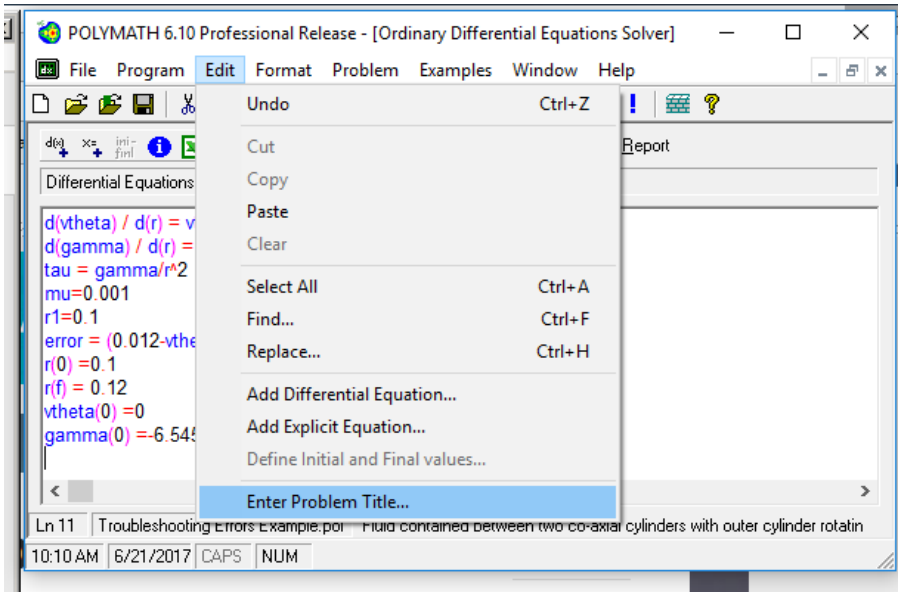


Figure 23: Always enter a problem title so that the professor will know what problem you solved.

30. Rename and Save this program for submission on blackboard. The new filename should have your last name as part of the title.

31. Other types of errors cause a program to stop running. This error starts after you press the Pink arrow to start solving the problem below and the solution stops with an error message window titled Polymath Guide. Below is a different example to show this error. Again copy and paste the below code into an ODE solver (or load the program divideByZeroCode.pol) and then run the code

```
d(tau_r)/d(r)= delP/L*r
d(vx)/d(r)=-(tau/K)^(1/n)
K=1e-6
delP=100
L=10
tau=tau_r/r
#tau=if(r>0) then (tau_r/r) else(0)
n=2
R=0.009295
r(0)=0
tau_r(0)=0
vx(0)=1.3358812
r(f)=0.009295
```

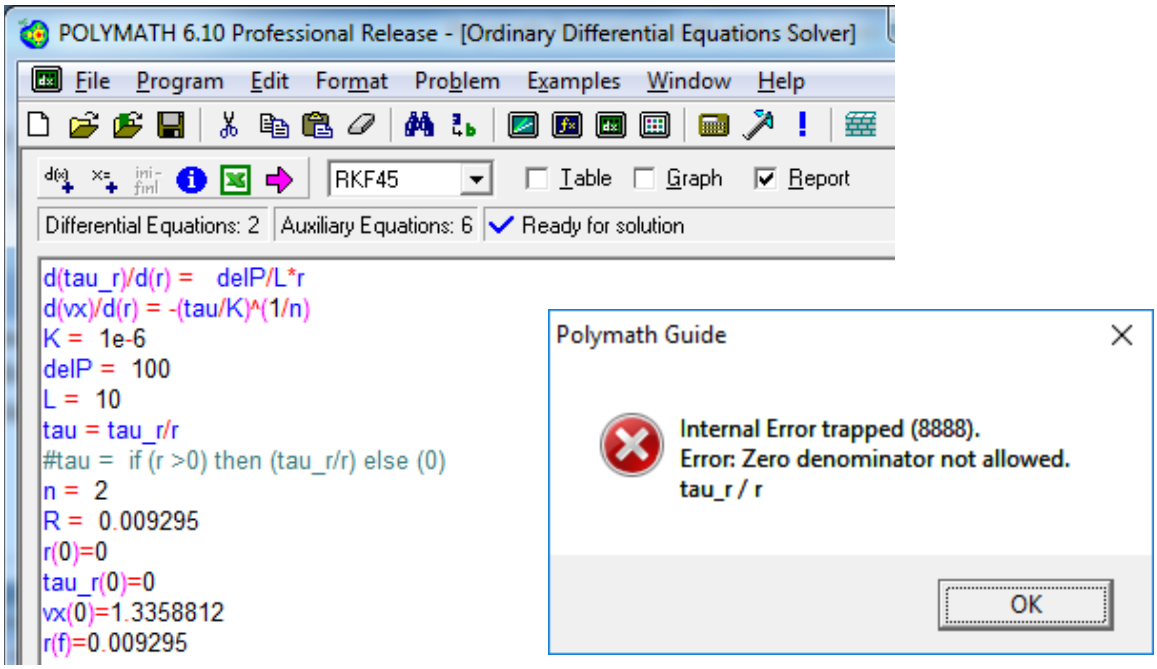
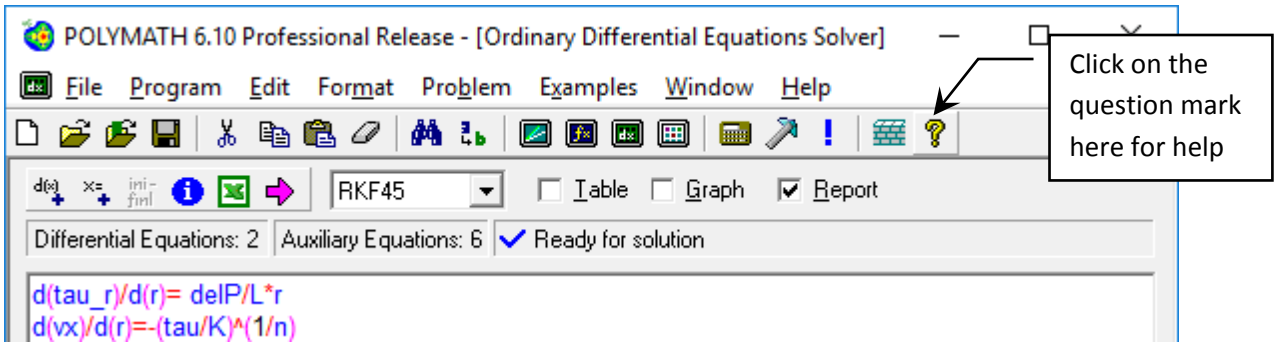


Figure 24: Divide by zero example after pressing the pink arrow

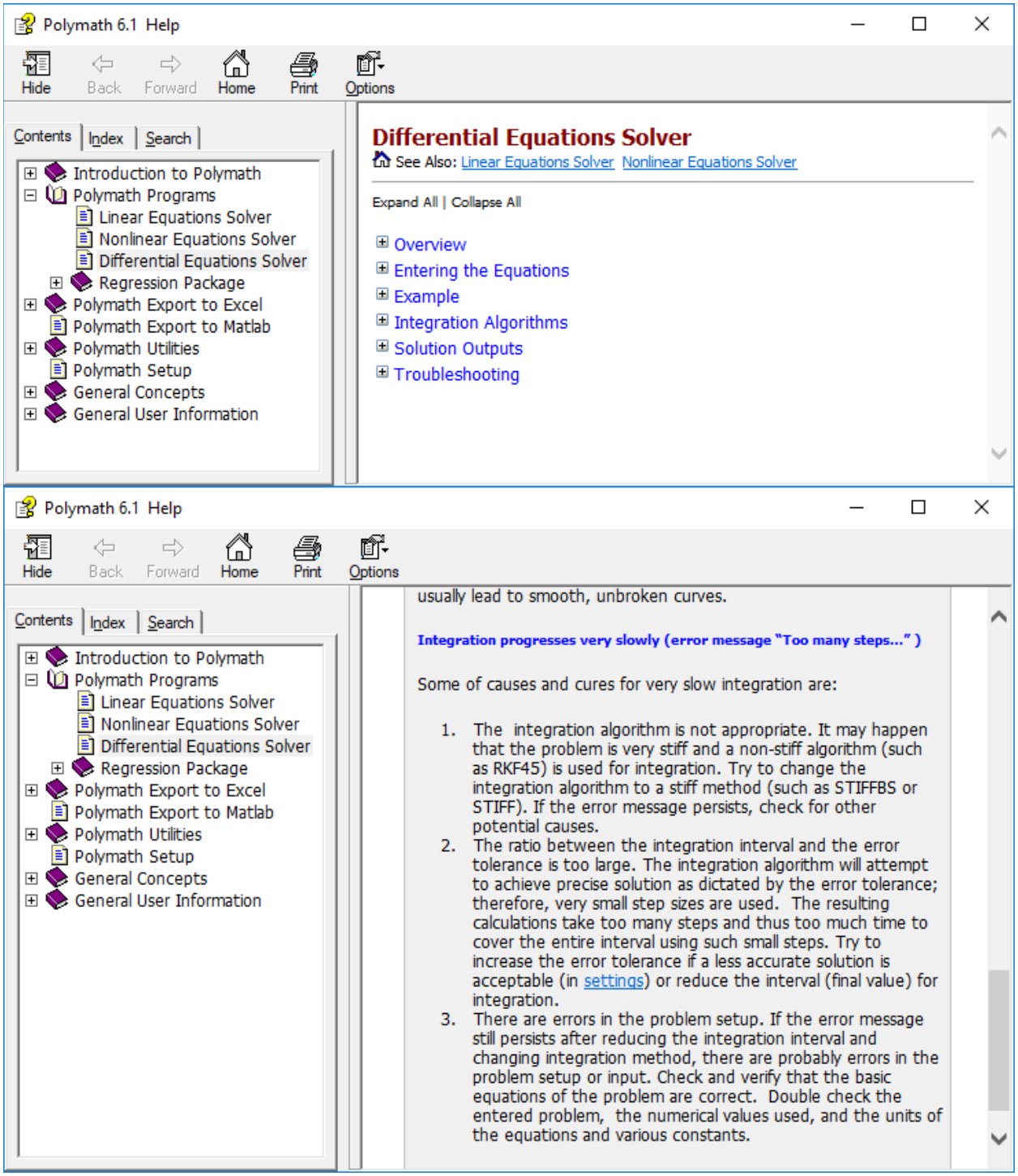
In line 6, the value of r at the beginning of the integration is zero (r(0)=0) so the program tries to divide by zero in the expression in line 6 (tau = tau_r/r). This causes the program to stop and this error code appears. To correct this error, the program needs an If, Then Else statement which has been commented out using a pound sign. This is shown in line #7: #tau = if (r >0) then (tau_r/r) else (0). The pound sign # is used as a comment marker. Delete the line: tau=tau_r/r and then remove the # that is used as a comment marker. Anything after the comment marker is ignored by the solver. Now the program will run.

32. Save this program for submission on blackboard. Again the filename should have your last name in it.

Additional tips for troubleshooting are found in the help menu for ODE's which can be found by clicking on the yellow question mark or pressing the F1 key.



Now scroll down or click the Collapse All to go to the section marked trouble shooting as shown below. Of particular interest will be what happens if Integration progresses very slowly and you need to use a stiff algorithm:



Submit to Canvas:

Submit in one pdf document per person. Title the pdf document using the following format: yournametutorial.pdf

1. pdf document containing (convert word to pdf using save as)
    1. page of required polymath output exactly like page 20/39 and graph
    2. page of required polymath output exactly like page 32/39

2. polymath progams (*.pol files)
    1. polymath program at the end of the reaction problem on page 16/39
    2. polymath program from the end of the error exercise on page 33/39
    3. polymath program from the end of the error exercise on page 34/39