

## EE202 Lab#8 Part 1 Numerical Methods

- [11.1] Make a new M-File named `pistonCylinder` as define in Appendix A (points will be deducted if you do not include the header instructions and comments). Add the following comment at the beginning of your script. `%% Lab 8 Question 1 Gas Volume and Pressure`. Consider a gas in a piston-cylinder device in which the temperature is held constant. As the volume of the device was changed, the pressure was measured. The volume and pressure values are reported in the following table:

Volume, m3	Pressure, kPa, When T = 300K
1	2494
2	1247
3	831
4	623
5	499
6	416

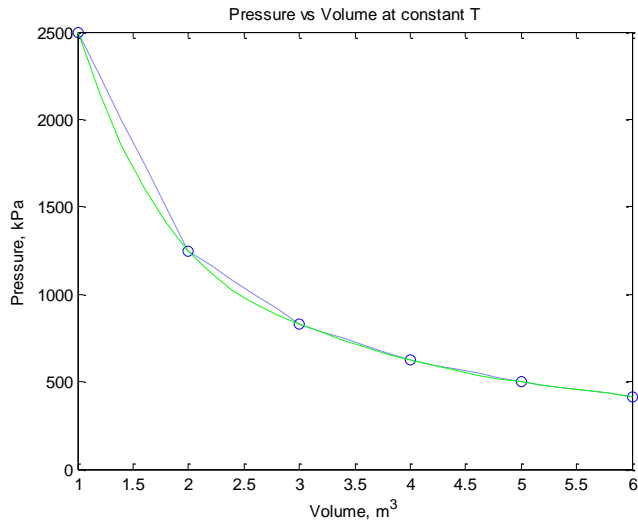
**interpolation** is a method of constructing new data points within the range of a [discrete set](http://en.wikipedia.org/wiki/Interpolation) of known data points (source: <http://en.wikipedia.org/wiki/Interpolation>) In this problem you will apply the Matlab `interp1` function with the `linear` (default) and `cubic spline` options to estimate the pressure and volume at a known point based on the data points from the table above.

- Use linear interpolation to estimate the pressure when the volume is  $3.8 \text{ m}^3$ .
  - Use cubic spline interpolation to estimate the pressure when the volume is  $3.8 \text{ m}^3$ .
  - Use linear interpolation to estimate the volume if the pressure is measured to be 1000 IPa.
  - Use cubic spline interpolation to estimate the volume if the pressure is measure to be 1000 kPa.
- [11.1] Working with the M-File from problem 1, add the following comment to the bottom of your script. `%% Lab 8 Question 2 Liner Interpolation`.
    - In this part of the problem we are going to use the data from problem 1 ( $P, V$ ) and `linear` interpolation (`interp1`) to create an expanded volume-pressure table with volume measurements every  $0.2 \text{ m}^3$ . Using colon notation, create a new volume array (`new_V`) of points from  $1 \text{ m}^3$  to  $6 \text{ m}^3$  at intervals of  $0.2 \text{ m}^3$ . Using the data from the table and `linear` interpolation create a new data set of pressures `new_P` at these points (`new_V`). Add the following Matlab code to your M-file to create your new table.

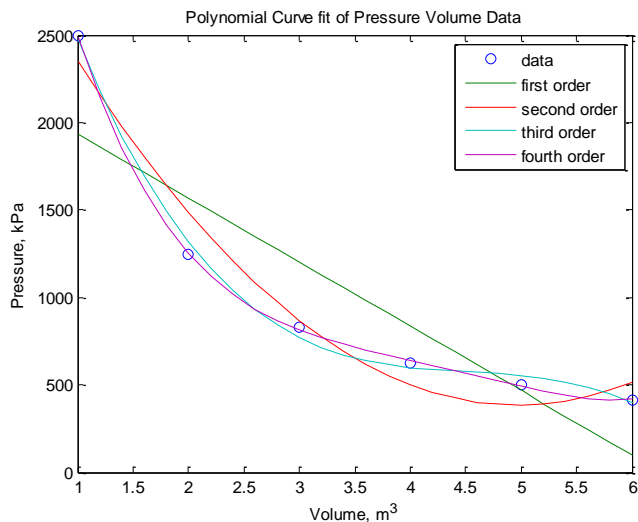
```
table=[new_V', new_P'];  
disp('Volume    Pressure')  
disp('  m^3      kPa')  
fprintf(' %4.1f  %8.1f \n', table')
```

- Repeat part 'a' using cubic spline interpolation.

- (c) Plot as figure (1) the data from your two tables along with the measured data on the same graph as shown in the figure below. Please title your plot, label the axis, and use line type, indicator, and colors shown.



3. [11.1] Working with the M-File from problem 1 and 2, add the following comment to the bottom of your script. `%% Lab 8 Question 3 Polyfit.`
- (a) Fit the data from problem 1 with first-, second-, third-, and fourth-order polynomials using the `polyfit` and `polyval` functions.
- Plot your results on the same graph.
  - Plot the actual data as a circle with no line.
  - Calculate the values to plot from your polynomial regression results at intervals of  $0.2 \text{ m}^3$ .
  - Do not show the calculated values on the plot, use solid lines (default) for the curves.
  - Use the `legend` function to create a legend.
- (b) Which model seems to do the best job? Add the answer to this question as a comment located at the end of your M-file.



4. [12.1, 19.2] Make a new M-File named **theQuad** as define in Appendix A (points will be deducted if you do not include the header instructions and comments). Add the following comment at the beginning of your script. **%% Lab 8 Question 4 Numerical Integration**. Consider the following equation:

$$\frac{dy}{dx} = 5x^3 - 2x^2 + 3$$

- (a) Apply what you have learned in calculus to solve for  $y$  over the interval from  $a$  to  $b$ . Show your work on a separate piece of paper to be turned in with your lab. In place of a pencil/pen you may use the Equation editor in Microsoft Word.

$$y = \int_a^b (5x^3 - 2x^2 + 3)dx$$

In your M-file set  $a = -1$  and  $b=1$  and then enter your solution (use array operators  $.^*$   $.^{\wedge}$ ), thus *analytically* solving your equation over the interval  $a = -1$  to  $b = 1$ .

- (b) Use the `quad` function to *numerically* find the integral of the differential expression with respect to  $x$ , evaluated over the interval -1 to 1.
- (c) Use the `quad1` function to *numerically* find the integral of the differential expression with respect to  $x$ , evaluated over the interval -1 to 1.
- (d) [19.2] Finally use the *symbolic* toolbox function, `int` to solve for  $y$ . You can find out more about in section 19.2 of your textbook or by using Google “Matlab int” and following the second link. Remember that the `quad` and `quad1` functions take input expressed with the array operators such as  $.^*$  or  $.^{\wedge}$ , but the `int` function takes a symbolic representation that does not use these operators.
5. [12.2, 19.2] Working with the M-File from problem 4, add the following comment to the bottom of your script. **%% Lab 8 Question 5 Differential Equations**.

$$\frac{dy}{dt} + \sin(t) = 1$$

- (a) Solve the following differential equation with the initial condition of  $y = 1$  when  $t = 0$ . Show your work on the same piece of paper with your analytical solution to problem 4. In place of a pencil/pen you may use the Equation editor in Microsoft Word.

$$\frac{dy}{dt} + \sin(t) = 1$$

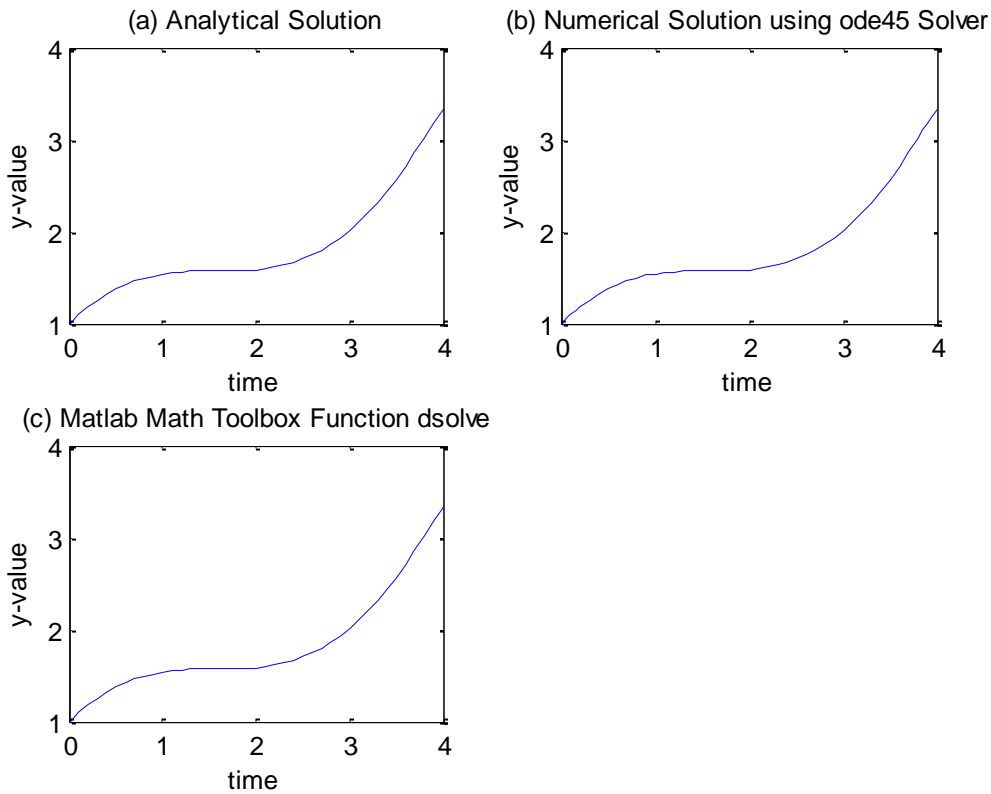
Make a subplot of your solution over the interval  $t= 0$  to 4, in increments of 0.1. Title and label your plot as shown in the figure below.

- (b) Solve the differential equation using the `ode45` solver.

- (c) [19.2] Solve the differential equation using the `dsolve` function. This function is not included in your textbook, so you will need to ask Google for help. Here is one problem that you will most likely run into along the way.

Let us assume that when you create your symbolic expression for  $y$  you use the letter  $t$ . You cannot plot a *symbolic expression* like  $t$  which is unrelated to the array *variable* containing time. To solve this problem you will need to use the `subs` function as shown here. The expression assumes your symbolic expression is assigned to `y_dsolve` and `time` is an array variable previously defined from 0 to 4 in increments of 0.1.

```
y_c=subs(y_dsolve,'t',time);
```



6. [11.2] Applying what you learned in labs 5, 6 and 7 update your rocket M-file to use the MATLAB `fzero` function to find the actual time, in seconds, it takes for the rocket to hit the ground. Display the answer as shown here.

The rocket hits the ground after nn.nn seconds.

## Appendix A      Creating Your Lab Report

### **How to Organize and Clean up Your Work**

To clean up past work, place the following line at the beginning of your M-File(s).

```
format compact  
clear,clc, close all % The close all command closes all figure windows
```

To make more than one figure in an M file, use the function `figure(n)`, where **n** is the next figure to be drawn.

### **How to Publish your Lab**

Matlab can format your M-file and the resulting outputs for publishing (i.e., a format you can turn in with your lab) by selecting File → Publish To → Word Document. If the conversion fails then try File → Publish To HTML, open in IE or Firefox and convert to PDF or simply print.