**PROJECT REPORT**

|  |  |  |
| --- | --- | --- |
| **Course Title: MATLAB Core II - Programming techniques** | | |
| **Course Code: UE19EC257F** | | |
| **Semester: IV** | **Section: A** | |
| **PES1UG19EC026** | **Akshata Kumble** | |
| **PES1UG19EC035** | **Ambika S Rao** | |

**Problem Statement:**

Using MATLAB to create a curve fit for the given set of data containing the specific heat of oxygen as a function of temperature

**Description:**

Specific heat is important as it gives an indication of how much energy is required to heat or cool an object of a given mass by a given amount. But, we know that the specific heat varies with temperature. When calculating mass and volume flow of a substance in heated or cooled systems with high accuracy, the specific heat should be corrected to get accurate results.

Curve fitting is an important tool when it comes to developing equations that best describes a set of given data points. It examines the relationship between one or more predictors or independent variables, and creates a response variable or dependent variable with the goal of defining a ‘best fit’ model for the relationship.

Here, we have tried to obtain the curve fit for data containing the specific heat of oxygen using various matlab commands like polyfit(), polyval() etc.

**Concepts Used**

* Polyfit is an in built function used for Polynomial curve fitting

[**p**](https://in.mathworks.com/help/matlab/ref/polyfit.html?searchHighlight=polyfit&s_tid=srchtitle#bue6sxq-1-p) **= polyfit(**[**x**](https://in.mathworks.com/help/matlab/ref/polyfit.html?searchHighlight=polyfit&s_tid=srchtitle#bue6sxq-1-x)**,**[**y**](https://in.mathworks.com/help/matlab/ref/polyfit.html?searchHighlight=polyfit&s_tid=srchtitle#bue6sxq-1-y)**,**[**n**](https://in.mathworks.com/help/matlab/ref/polyfit.html?searchHighlight=polyfit&s_tid=srchtitle#bue6sxq-1-n)**) r**eturns the coefficients for a polynomial p(x) of degree n that is a best fit (in a least-squares sense) for the data in y. The coefficients in p are in descending powers, and the length of p is n+1



Eg. co\_eff1 = polyfit(T,cp,1) returns the values of coefficients that correspond to the fitting curve of degree 1. The equation will be of the form ax +b (a, b being the coefficients, cp being the specific heat and T being the temperature).

Similarly co\_eff3 = polyfit(T,cp,3) returns the values of coefficients of a degree 3 polynomial ax3+bx2+cx+d

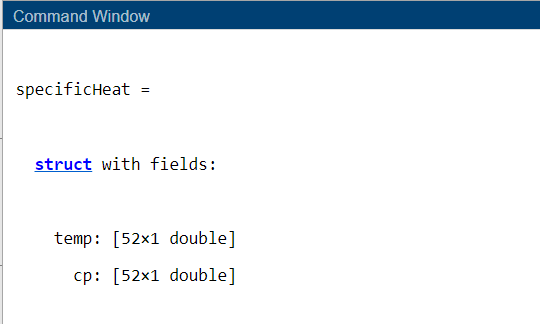
* Polyval is used to evaluate the polynomial

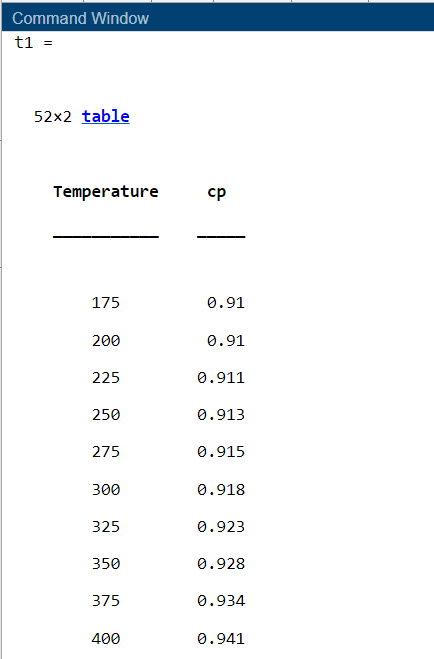
[y](https://in.mathworks.com/help/matlab/ref/polyval.html?s_tid=doc_ta#d123e1047347) = polyval([p](https://in.mathworks.com/help/matlab/ref/polyval.html?s_tid=doc_ta#f93-999699_sep_shared-p),[x](https://in.mathworks.com/help/matlab/ref/polyval.html?s_tid=doc_ta#d123e1047200)) evaluates the polynomial p at each point in x. The argument p is a vector of length n+1 whose elements are the coefficients (in descending powers) of an nth-degree polynomial

Eg. predicted\_cp3 = polyval (co\_eff3, Temp). This creates the predicted\_cp3 curve as per equation ax3+bx2+cx+d

* Extracting data

Structural array is created from which the variables for the table are extracted.





To extract the contents from multiple columns we specify the rows and columns using numerical or named indexing.

variableData = tableName{rowIndices,colIndices}

Using dot notation with the variable name to extract the contents from a table variable

variableData = tableName.VariableName

* Eg. legend('Original dataset','location','southeast')

The legend function also accepts two optional arguments: the keyword 'Location' and a text description of the location which is given by compass points, such as 'north' or 'southwest'.

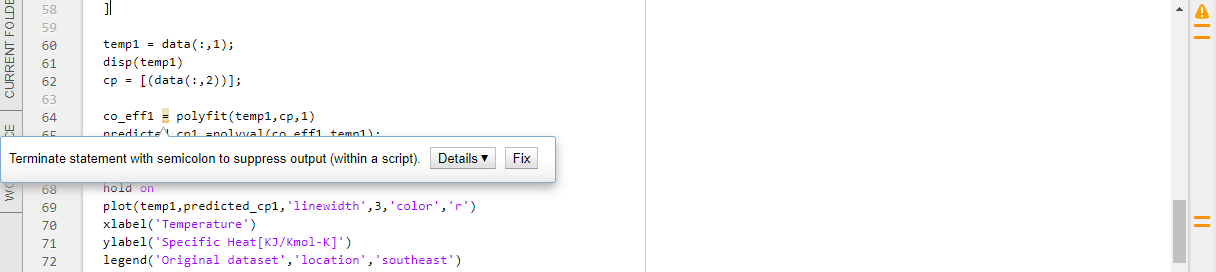
* b=cell2table(data); This is used to convert a cell array into a table
* plot(X,Y,LineSpec) sets the line style, marker symbol, and color. Line Properties include line style, line colour, line width

eg. plot(temp,predicted\_cp1,'linewidth',3,'color','r'). This plots a red coloured line of width 3

* 'VariableNames' property is used to specify column names

Eg. tbl = table(var1,var2, 'VariableNames',{'Temperature','Specific heat'})

* *hold on* retains plots in the current axes so that new plots added to the axes do not delete existing plots. *hold off* sets the hold state to off so that new plots added to the axes clear existing plots and reset all axes properties.
* Use of Code Analyzer

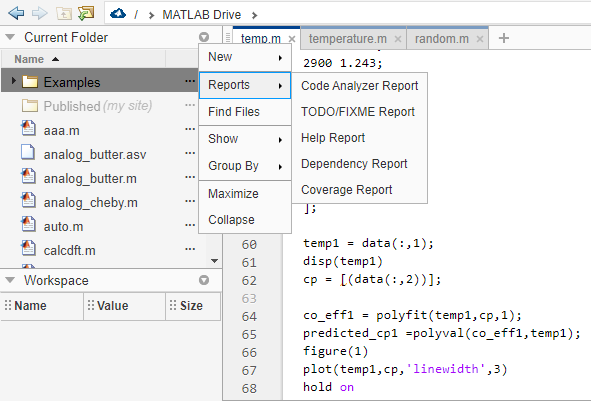
The code analyzer can help you identify some bugs before running your code

The orange indicator is for warnings for unexpected behavior or inefficient code. The code might still run even when the warnings are present.The lack of an orange or red indicator indicates there are no syntax errors or other warnings, but does not imply anything about run-time errors*.*

* Folder Reports

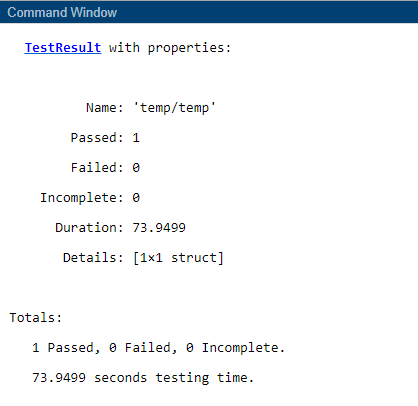
To run a folder report on the current folder, access the Reports menu from the Current Folder Actions menu.

Folder Reports include: Code Analyzer report, Help Report, TODO/FIXME Report

****

* runtests function

A test allows you to verify that your code works correctly by testing specific inputs and validating that expected values have been created.

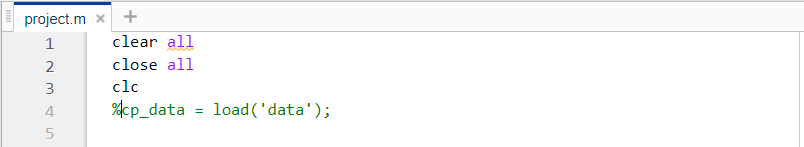
****

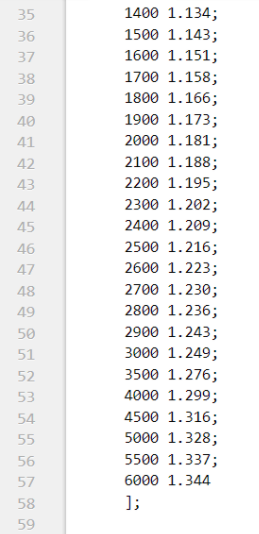
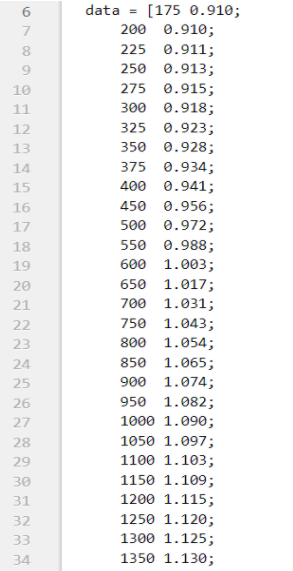
**Learning Outcome:**

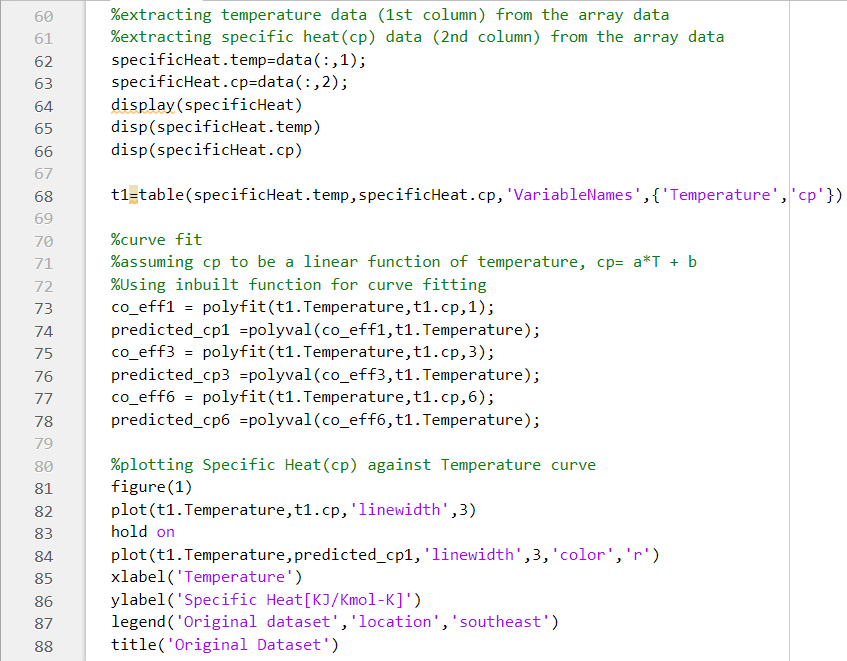
By this exercise/project, we learnt

* The importance of curve fitting in daily life.
* To appreciate various applications of graphs and its importance.
* To use fitting curves as an aid for data visualization so as to infer values of a function even when not much data is available.
* To use curve fitting to summarize the relationship between two or more variables.
* To extract data from cell array, structural array and table.
* To use various commands like polyfit(), polyval() etc.
* To use various concepts of matlab like functions, loops, data types
* The importance and use of testing and debugging

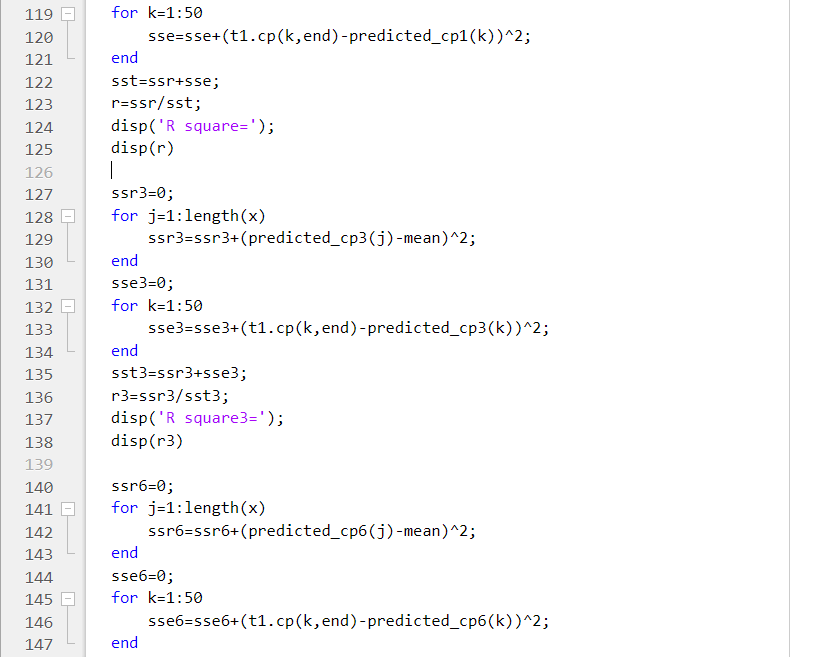
**Code Screenshots:**

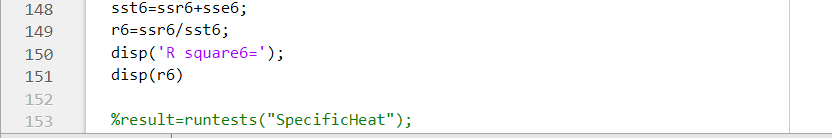
****

****

****

****

****

****

Note:

R square measures how closely the fit resembles the variation of the data. It is the correlation between the data set values and the predicted values. R square ranges from 0 to 1, higher the value of R square better the fit .

R2 = where SST= SSR+SSE

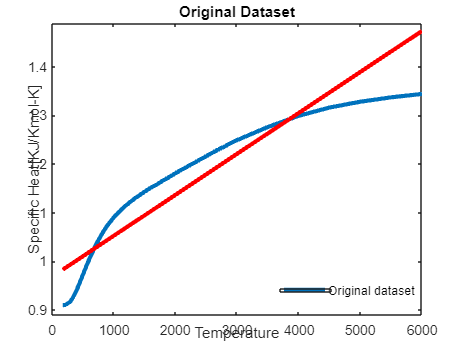
SSE is the sum of squared errors 

SSR is sum of squares of the regression

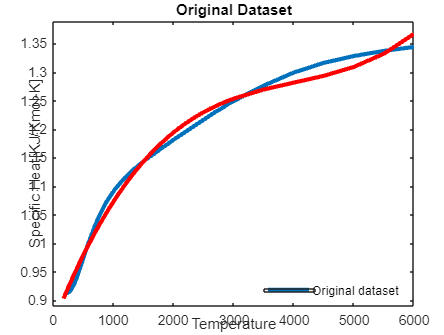


**Output Screenshots:**

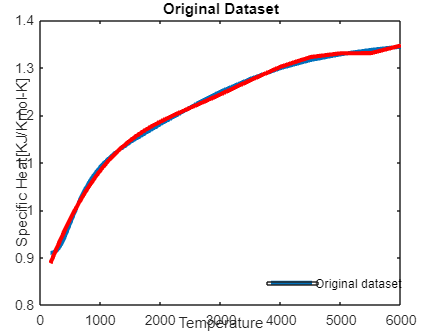
Curve fit of a linear polynomial

****

Curve fit of a cubic polynomial

****

Curve fit of a hexa polynomial

****

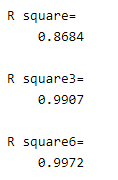
R square measures how closely the fit resembles the variation of the data. It is the correlation between the data set values and the predicted values. R square ranges from 0 to 1, higher the value of R square better the fit .

R square for the linear polynomial = 0.8684

R square for the cubic polynomial = 0.9907

R square for the hexa polynomial = 0.9972

Therefore by increasing the degree of the polynomial we get a better curve fit.

****