Interpolation

A Linear Interpolation

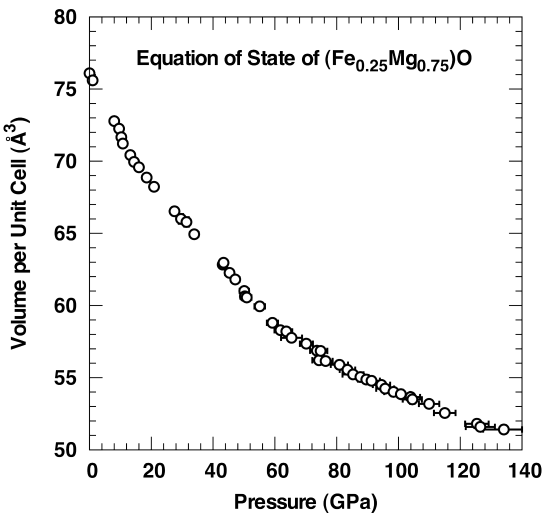


Figure 1. Equation of state of $(Fe_{0.25}Mg_{0.75})O$. Values are taken from J.F. Lin, V.V. Struzhkin, S.D. Jacobsen, M.Y. Hu, P. Chow, J. Kung, H. Liu, H.K. Mao, R.J. Hemley, Spin transition of iron in magnesiowüstite in the Earth's lower mantle, *Nature*, 436, 377-380, (2005).

In research, we are often interested in how one property varies as a function of another. For example, in geophysics we are interested in how the unit cell volume of lower mantle minerals varies with pressure, since it provides insight into how their density will change, with depth, within the Earth's interior. The pressure dependence of the unit cell volume of magnesiowüstite is shown above in Figure 1. Due to the complex nature of high-pressure experiments, only a limited number of measurements were made. In order to estimate the unit cell volume at pressures other than those studied, we can use interpolation.

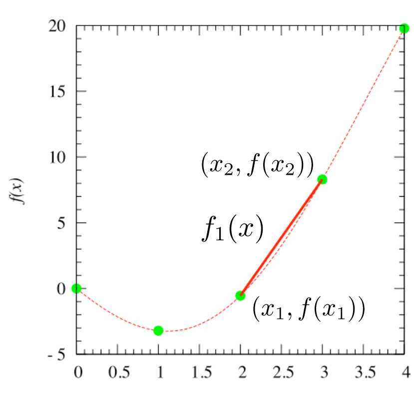
Linear Interpolation

The simplest form of interpolation is linear interpolation:

- locate the two data points that bracket the x value of interest
- find the equation of the straight line that connects them

$$f(x_1) = mx_1 + c$$

$$f(x_2) = mx_2 + c$$



- solving for m and c gives:

$$m = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$c = f(x_1) - \frac{f(x_2) - f(x_1)}{x_2 - x_1} x_1$$

$$\boxed{f_1(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1)}$$

Let us consider a general experiment, where we measure the property $f(x)$, as a function of a second property x. In the course of the experiment we will obtain a series of data points $(x_i, f(x_i))$, where i goes from 1 to n and n is the number of measurements. Using interpolation, it is possible to estimate $f(x)$ at any x value that lies within the range studied.

Your task is to write a MATLAB script that takes any set of experimental data and uses linear interpolation to estimate the value of $f(x)$ at any x value that lies within the range studied. You script should do the following:

1. Prompt the user for the name of the file containing the experimental data and read it in. (The format of this file is discussed below.)
2. Prompt the user for the x value at which $f(x)$ is desired and read it in.
3. Check if the input x value lies within the range studied in the experiment. If not, inform the user and ask them to input another value.
4. Check if the input x value is equal to one of those at which a measurement was made. If so, the program should print the corresponding $f(x)$ value to screen and end. If not, it should proceed to Step 5.
5. Loop through the experimental data to determine the pair of data points that bracket the input x value.
6. Perform linear interpolation using the data points from Step 5.
7. Write the result to the screen.

As usual, remember to describe the program and all the variables in a header; write your program in a clear manner and include helpful descriptive comments; and format your output.

Data File

The file containing the experimental data will comprise two columns, the first being the x values and the second the $f(x)$ values. To make searching through the experimental data simpler, you can assume that the values are ordered with respect to x, with the lowest x value at the top and the highest at the bottom. If this were not the case use could the MATLAB function `sortrows()`.

Testing Your MATLAB Script

The values shown in Figure 1 have been stored in a file called `data`, which is available on the VLE. You can download the file from:

```
/nfs/ugrad-library/SOEE2250/practical_7
```

The data file comprises two columns of values. The first column is the pressure (in GPa) and the second is the unit cell volume (in Å^3). Use your program to estimate the unit cell volume of magnesiowüstite at 40 GPa.

B Interpolating Polynomials

Write a MATLAB function to interpolate using Newton interpolating polynomials. The pairs of x and $f(x)$ values to use for interpolation and the value of x at which to interpolate should be passed to the function at the command line. It should return the interpolated value of $f(x)$.

```
>> x = [0; 1.0; 8.0; 9.6];
>> fx = [76.10; 75.60; 72.78; 72.25]
>> xint = 5.0
>> [fxint] = newton(x,fx,xint)
```

Base you script on the equations for the general case – these lend themselves to an algorithm. The function should require only 15 executable lines of code.

Testing Your MATLAB Function

You can check that your function works by comparing it to the values that you get using `polyfit` and `polyval`. They should be identical.

C Bringing it all Together

Make a copy of the script that you wrote for PART A, and combine it with the function that you wrote for PART B, such that instead of just performing linear interpolation, the user is able to select the order of the polynomial used for interpolation. Your script should determine the data points to use and pass them to your function, which should then return the correct interpolated value.