Computational Physics

Eitan Lees

2025-02-24

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

1 Introduction

I first learned to program in 2011 in Dr. Richard O. Gray's computational physics course.

It was a revelatory experience. It shaped the rest of my life.

I enjoyed the class so much that the following year I was the TA.

I went on to get a PhD in Computational Science.

Ever since I took the course I have kept the lecture notes on my various computers.

Here I am attempting to convert them to Quarto.

I am also thinking of updating the content to Python.

2 Colophon

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

3 Interpolation

3.1 Linear Interpolation

A common computational problem in physics involves determining the value of a particular function at one or more points of interest from a tabulation of that function. For instance, we may wish to calculate the index of refraction of a type of glass at a particular wavelength, but be faced with the problem that that particular wavelength is not explicitly in the tabulation. In such cases, we need to be able to interpolate in the table to find the value of the function at the point of interest. Let us take a particular example.

BK-7 is a type of common optical crown glass. Its index of refraction n varies as a function of wavelength; for shorter wavelengths n is larger than for longer wavelengths, and thus violet light is refracted more strongly than red light, leading to the phenomenon of dispersion. The index of refraction is tabulated in Table 3.1.

Let us suppose that we wish to find the index of refraction at a wavelength of 5000. Unfortunately, that wavelength is not found in the table, and so we must estimate it from the values in the table. We must make some assumption about how n varies between the tabular values. Presumably it varies in a smooth sort of way and does not take wild excursions between the tabulated values. The simplest and quite often an entirely adequate assumption to make is that the actual function varies linearly between the tabulated values. This is the basis of **linear interpolation**.

Exercise 4.1

Determine, by hand, the value of the index of refraction of BK7 at 5000 using linear interpolation.

How do we carry out linear interpolation on the computer? Let us suppose that the function is tabulated at N points and takes on the values $y_1, y_2, y_3 ... y_N$ at the points $x_1, x_2, x_3 ... x_N$, and that we want to find the value of the function y at a point x that lies someplace in the interval between x_1 and x_N.

The first thing that we must do is to bracket x, that is we must find a j such that $x_j < x \leq x_{j+1}$. This can be accomplished by the following code fragment:

Table 3.1: Refractive Index for BK7 Glass

$\lambda()$	n
3511	1.53894
3638	1.53648
4047	1.53024
4358	1.52669
4416	1.52611
4579	1.52462
4658	1.52395
4727	1.52339
4765	1.5231
4800	1.52283
4861	1.52238
4880	1.52224

$\lambda()$	n
4965	1.52165
5017	1.5213
5145	1.52049
5320	1.51947
5461	1.51872
5876	1.5168
5893	1.51673
6328	1.51509
6438	1.51472
6563	1.51432
6943	1.51322
7860	1.51106

$\lambda()$	n
8210	1.51037
8300	1.51021
8521	1.50981
9040	1.50894
10140	1.50731
10600	1.50669
13000	1.50371
15000	1.5013
15500	1.50068
19701	1.495
23254	1.48929

```
for(i=1;i<N;i++) {
    if(xn[i] < x && xn[i+1] >= x) {
        j = i;
        break;
    }
}
```

where the `xn`'s are the tabulated points. When the `if` statement is satisfied, `j` is assigned the value of `i` and the procedure drops out of the loop. Please note that this is not the most efficient way to accomplish this task, especially if N is very large. We will look at a more efficient way later on.

Once we have bracketed x, we can find the equation of the line between the points (x_j, y_j) and (x_{j+1}, y_{j+1}). This equation will be of the form $y = mx + b4where$m$ is the slope and b is the y-intercept. As we all know, the slope is given by

$$m = \frac{y_{j+1} - y_j}{x_{j+1} - x_j}$$

and the intercept can be found by substituting one point, say, (x_j, y_j) into the resulting equation. Thus,

$$b = y - mx = y_j - \frac{y_{j+1} - y_j}{x_{j+1} - x_j} x_j$$

yielding for the equation of the line, after some rearrangement,

$$y == y_j + \left(\frac{y_{j+1} - y_j}{x_{j+1} - x_j} \right) (x - x_j)$$

It is left to the student to show (for future reference) that this equation may be rewritten

$$y = Ay_j + By_{j+1}$$

where

$$A = \frac{x_{j+1} - x}{x_{j+1} - x_j}$$

and

$$B = \frac{x - x_j}{x_{j+1} - x_j}$$

Exercise 4.2

Write a C-function that will linearly interpolate the tabular data for the index of refraction of BK-7 and return a value for n for wavelengths between 3511 and 23254. Write a driver program that will use this function to prompt the user for a wavelength and then print to screen the corresponding value of n.

Exercise 4.3

The file `data/boiling.dat` contains data in two columns for the boiling point of water at different atmospheric pressures. The first column is the pressure in millibars, the second is the corresponding boiling point temperature in degrees Celsius. Write a C-function that initializes two vectors, `P` and `T` with the data in that data file (don't read in the datafile – hardwire the data into your program), accepts the pressure as a double floating-point parameter, and returns the value of the temperature of the boiling point at that pressure. You should also write a driver program that will prompt the user for an atmospheric pressure, check whether it is within the limits of the data ($50 \leq P \leq 2150$), calls your C-function, and prints to the screen the boiling point of water at that pressure.

4 Finding the Root of a Function

5 Minimization and Maximization of Functions

6 Numerical Integration

7 Numerical Integration of Ordinary Differential Equations

8 The Modeling of Data

References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.