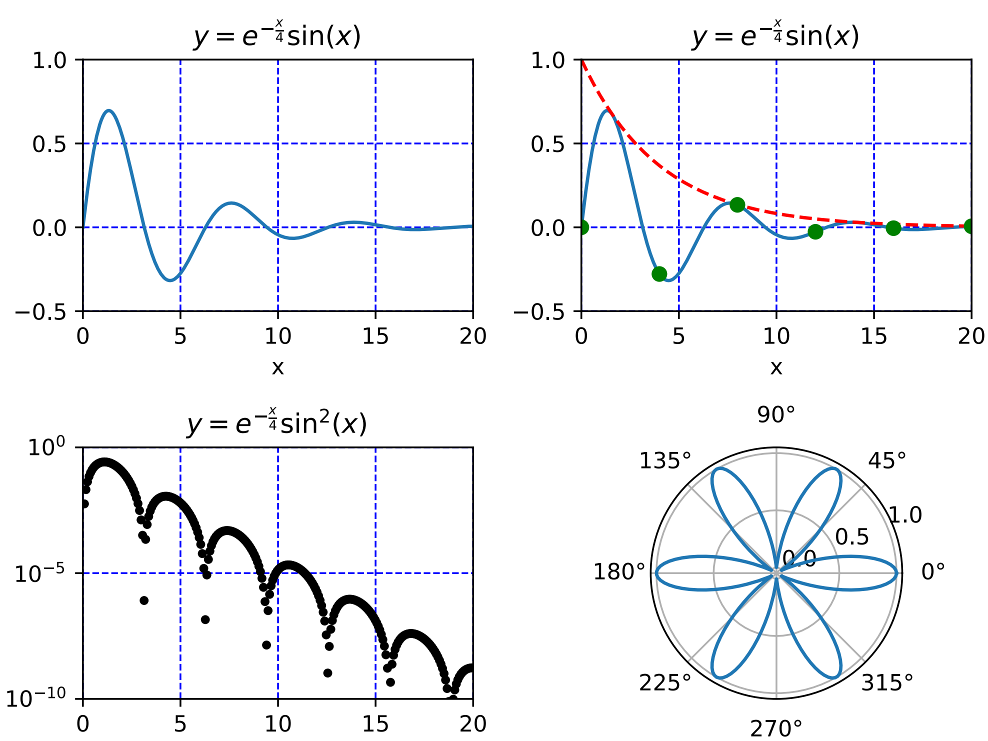
PHYS 416/517 - Problems from Chapter 1 – 2020

[due: Thursday, January 23 at 1 PM]

- **The purpose of this assignment is to familiarize you with the MATLAB and PYTHON programming environments.**
- *Numbered questions are set to be the same as the Garcia book; lettered questions are **not** from the book.*
- *When you turn in your programs, please use the same numbering convention, for example the programs for first problems below should be numbered ex3.m (.py), exA.m (.py), etc and so on.*
- *Keep all your programs and writeups in the same folder (for example, call this folder chapter1) and upload all that into canvas, preferably as one compressed archive such as a zip file.*
- *Ideally it would be better to include any write-ups, comments and derivations as a word or pdf file and include it in your submission folder.*
- *Try to make your programs as self-contained as possible, so that the grader can easily run them. In other words, the grader should be able to just run the program, without having to type a lot of inputs and get the desired output.*
- *Each problem is worth 10 points.*
- ***If you get stuck, ask for help – that's what I am here for!***

Exercises

3. Reproduce the following graph (on a single page). Try to be as close as possible in your reconstruction, include the original driver program and the graphic output. [Hint: use subplot and Latex style titles.]



Programming exercises

A. *The hailstone or 3n+1 problem.* Let *n* be a positive integer; we iterate *n* with the following procedure: if *n* is even, we divide *n* by 2, if *n* is odd we replace *n* by *3n+1*.
In summary:

$$n = \begin{cases} \dfrac{n}{2}, & \text{for n even} \\ 3n+1, & \text{for n odd} \end{cases}$$

The algorithm repeats until it reaches the value 1. For example, starting at 3, the sequence is [3 10 5 16 8 4 2 1]. The above process terminates when it gets to 1, currently no starting number has yet been found in which the sequence does not terminate at 1. (A proof that the sequence always converges to 1, for any starting number, is known as the Collatz conjecture, is an unsolved problem in mathematics. Finding a proof of the Collatz conjecture, or a counterexample, would probably make you famous, but not rich. The mathematician Paul Erdös offered a $500 reward for its proof.) Some of the interesting features of this sequence are:

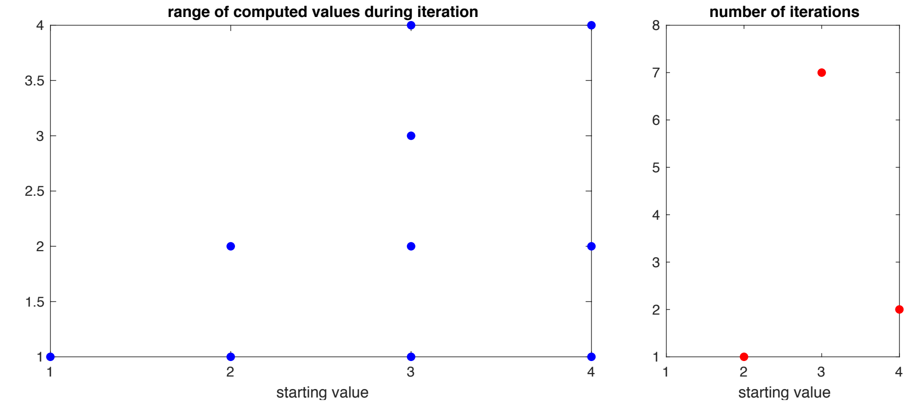
- The reason it is called the hailstone problem, is that the sequence can go up and down, just like a hailstone in a cloud.
- Once the number n becomes a power of 2 it will converge directly to 1.
- If you apply the rule to starting at the number 1, the sequence you get is [1,4,2,1] which then repeats.
- Every odd number is followed by an even number in the sequence, which is easy to prove.
- If you replace the 3*n*+1 rule with 3*n*-1, it is easy to find an example where the sequence does not converge to 1, for example starting at 5 the sequence you get cycles back to 5, never passing through 1.
- I am not aware of any other rule that always cycles back to 1, but you are welcome to try to find one!

Write a program that inputs a value of *n* and iterates until *n* becomes 1. Plot the computed number versus iteration for a starting value of 27. [Hint: Use the while statement.] A slightly more general version of this program would also check to see if your sequence repeats or cycles. Such a program could then be used to check or look for other rules.

B. Now modify your program so that it computes 3 vectors. The first vector *s(k)* is the starting number (*N*), the vector *f(k)* stores the computed number as a function of starting number *N* at iteration *k* and *g(N)* is the number of iterations needed to get to the number 1. For example, if *N*=4, *s, f* and *g* would look like:

$$s = [1,2,2,3,3,3,3,3,3,3,3,4,4,4]$$
$$f = [1,2,1,3,10,5,16,8,4,2,1,4,2,1] \qquad g = [0,1,7,2]$$

And the corresponding plots would like something like this:



Make a plot of these values (in one window if possible), and make its *x* and *y* axis range from 1 to *N*. Use the number range from 1 to about 200 for both axes. (You can try a larger upper number, but it will take considerably longer and the resulting plots will be rather full.)

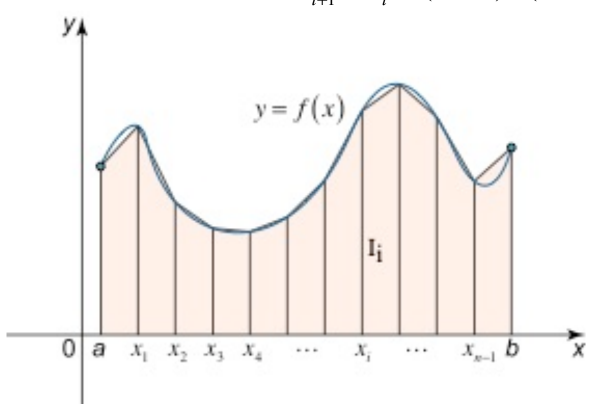
C. (Chapter 10, exercise 16) The Trapezoidal rule for integrating a function *f(x)* is a simple numerical scheme for determining the integral

$$I = \int_a^b f(x)dx$$

It is done by dividing the interval [*a, b*] into *n* equal spaces, we can define a grid as

$$x_i = a + (b-a)\frac{i-1}{n-1} = a + h(i-1)$$

So that the spacing $h = x_{i+1} - x_i = (b-a)/(n-1)$



The area under the curve in each interval (i) is approximated by a trapezoid that has an area

$$I_i = \frac{1}{2}h[f(x_{i+1}) + f(x_i)]$$

The integral is then the sum of these values,

$$I = \sum_{i=1}^{n-1} I_i$$

this comes out to be

$$I \approx \frac{1}{2}hf(x_1) + hf(x_2) + ...hf(x_{n-1}) + \frac{1}{2}hf(x_n) = \frac{1}{2}h[f(x_1) + f(x_n)] + \sum_{i=2}^{n-1} hf(x_i)$$

This is the trapezoidal rule for numerical integration, it has an error term of $O(h^2)$.

Now for the problem: Debye theory tells us that the heat capacity of a solid is given by the integral

$$C_v(T) = 9kN\frac{T^3}{\theta_D^3}\int_0^{\theta_D/T} \frac{x^4e^x}{(e^x-1)^2}dx$$

Where θ_D is the Debye temperature, N is the number of atoms, and k is Boltzmann's constant. Produce a graph of the molar specific heat of copper ($\theta_D = 309K$) from T=0 to T=1083 K (melting point). Use a log scale for the x-axis. [Computer]

D. Modify the `intrpf` function (with a new name) so that it can handle any number of data points by using higher-order polynomials. Test your program, using the original `intrpf` function. Once you have convinced yourself your new function is working, try it out for 2 functions:

(a) The sine function: $f(x) = \sin(x)$ for $0 \leq x \leq 2\pi$ using 5 and 7 evenly spaced points for the data points. Plot the resulting Lagrange polynomial and the original sine function, along with the data points. Comment on how the results compare.

(b) The Heavyside function: $f(x) = 0.5(1 + \tanh(10x))$ for $-2 \leq x \leq 2$ using 5 and 7 evenly spaced points for the data points.

In both cases plot the resulting Lagrange polynomial and the original function, along with the data points. Comment on how the results compare. [Computer]

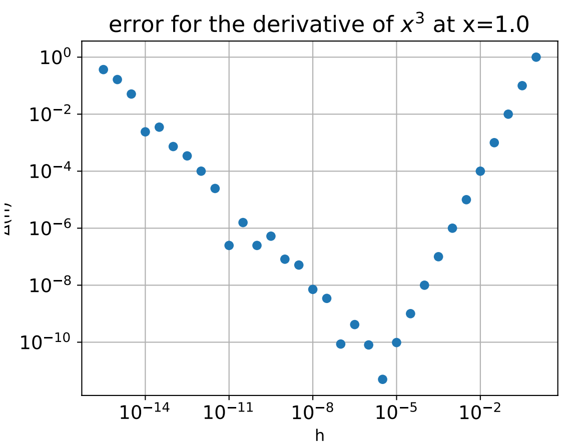
24. Write a program to reproduce similar to Figure 1.3 of the text (shown below) for the function $f(x) = x^2$at x=1, but for $f(x) = x^3$at x=1 using central differencing. [The basic idea here is to compute the difference between the exact numerical derivative $f'(1)$ and the numerical derivative given by

$$f'(1) \approx \frac{f(1+h) - f(1-h)}{2h} = \frac{(1+h)^3 - (1-h)^3}{2h}$$

so you will plot the function

$$\Delta(h) = \left| 3 - \frac{(1+h)^3 - (1-h)^3}{2h} \right|$$

Use h = 1, $10^{-0.5}$, 10^{-1}, $10^{-1.5}$, 10^{-2} ,…]



Show that the error term can be approximated as:

$$\Delta(h) = M\frac{h^2}{6} + \frac{\epsilon}{h}$$

Where $M = f'''(\theta) \approx 6$ and ϵ is the machine epsilon value. Plot this function along with the above plot. Why is the error in the estimate of the derivative increase with very small values of h? [Computer and Pencil]