Computational Physics Homework 1

Michael Albergo

27 September 2019

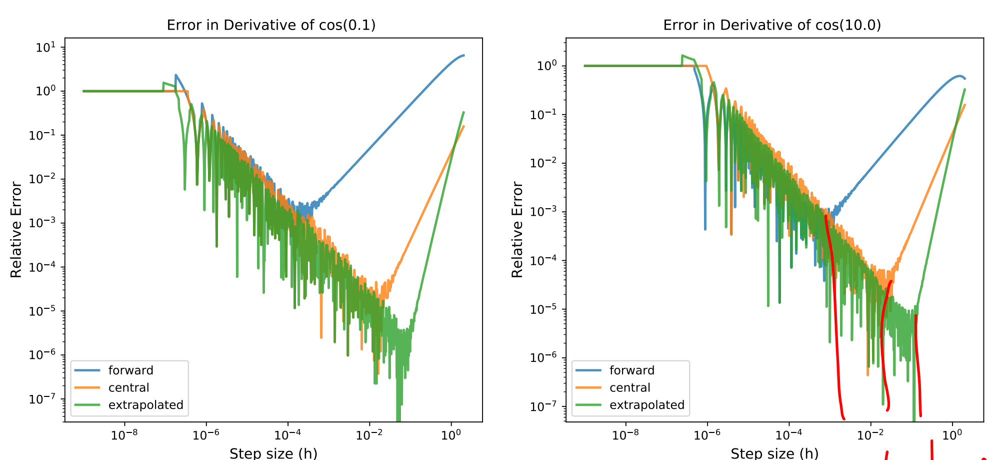
9/10

1 Problem 1

In this question we are asked to code up different numerical derivative methods and benchmark their error.

a) **Code up the three numerical derivative techniques.**

The code is available in the Jupyter notebook titled "HW1_MSA" in this repository.



do these agree w/ expectations? - ah, I see.

Figure 1: Relative error in numerical differential of cos(0.1) and cos(10.0) as a function of derivative approximation step size.

b) **Make a log-log plot of the relative error ϵ vs step size h and check whether the scaling and the number of significant digits obtained agrees with simple estimates.**

The relative error is computed as:

$$\epsilon = \left| \frac{\hat{x}_h - x}{x} \right| \tag{1}$$

where $\hat{x}_h$ is the estimate of the derivative at point x with derivative approximation step size h. Here I compare the relative error for $\cos(x)$ and $\exp(x)$ evaluated at 0.1 and 10. The optimal points are summarized in Table 1 along with a comparison of significant digits. We assume that the roundoff error from single precision is around C = 10^{-7}. All significant digits (exponents in the error) tend to agree with the theory, which

Good! :)

	Forward	Central	Extrapolated
$\cos(x)$	Between 10^{-3} and 10^{-4}	Between 10^{-5} and 10^{-6}	Between 10^{-6} and 10^{-7}
$\exp(x)$	Between 10^{-4} and 10^{-5}	Between 10^{-5} and 10^{-6}	Between 10^{-6} and 10^{-7}
Theory, $C \approx 10^{-7}$	$\epsilon \sim h, h_{opt} \approx C^{1/2} \to \epsilon \sim 10^{-7/2}$	$\epsilon \sim h^2, h_{opt} \approx C^{1/3} \to \epsilon \sim 10^{-5}$	$\epsilon \sim h^4, h_{opt} \approx C^{1/4} \to \epsilon \sim 10^{-7}$

Table 1: Summary of significant digit alignment for each numerical method against the theory according to the Newman textbook.

is summarized for each numerical method in the "Theory" row. These details are extracted from pages 190-196 in the Newman textbook.[1]

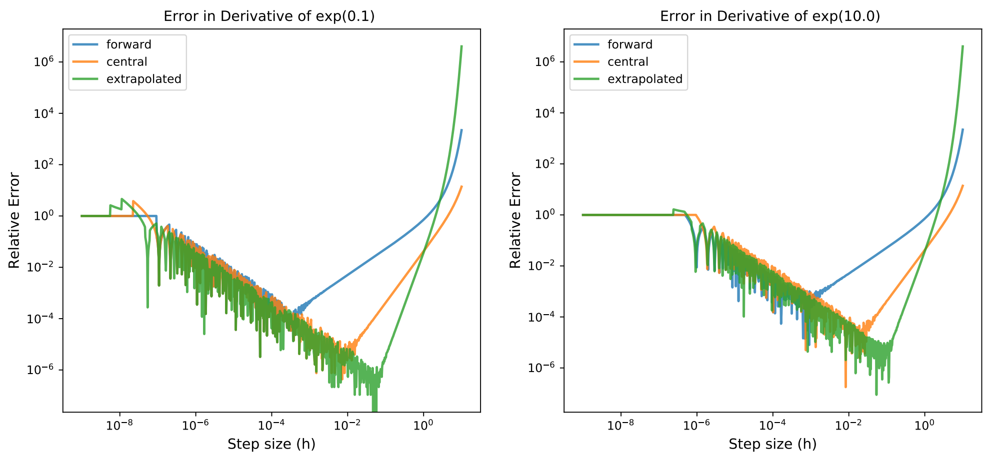


Figure 2: Relative error in numerical differential of exp(0.1) and exp(10.0) as a function of derivative approximation step size.

c) The effect of roundoff error is observed moving to the left of the optimal step h_{opt} (characterized by noise), and the truncation error is observed to the right of h_{opt} as the step sizes become too large. The beginning of roundoff varies between 10^{-1} (extrapolated) and 10^{-4} (forward) in step size based off each method. This agrees with what is known in the literature. For example, we are told on page 192 of the Newman textbook that h_{opt} is actually larger for the central difference method than for the forward method ($C^{1/3}$ vs $\sqrt{C}$), which agrees with our experiment. The roundoff error increases as the step size gets smaller until you've reached machine precision error (because we are using single precision, the derivative calculated there becomes zero). There isn't much difference in behavior between the two functions $cos(x)$ and $\exp(x)$.

2 Problem 2

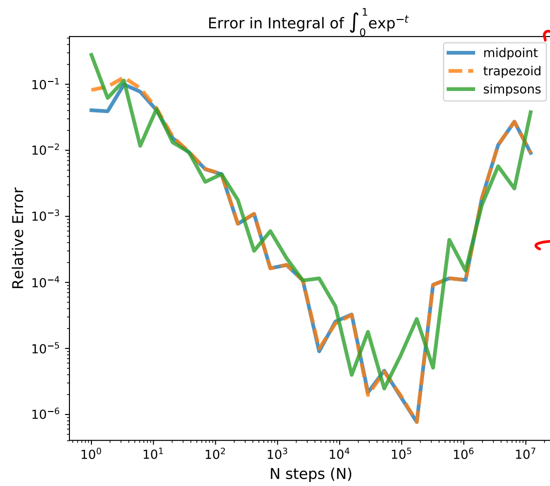
a) Code up the three numerical integration techniques.

The code is available in the Jupyter notebook titled "HW1_MSA" in this repository, just like the derivative techniques demonstrated above.

[1] These simple estimates apply when $f(x)$ and $f''(x)$ are of order 1.

b) **Make a log-log plot of the error as a function of number of bins in the integral.**

The log-log plot can be seen in Figure 3, comparing the three methods.



what is scaling w/ N?

but the three methods shouldn't agree. Need to explain.

−1

Figure 3: Relative error in numerical integration of $\exp(-t)$ from 0 to 1 as a function of derivative approximation step size.

c) **Explain what you see in the plot**

We can see the effects of truncation up to around the range of $N = 10^4$ or $N = 10^5$, where we reach optima in the error. From there, the roundoff effects take hold from the smaller and smaller bin widths that exacerbate the limits of single precision. Notably, midpoint and trapezoid agree almost everywhere, as would be expected from a monotonically decreasing (or increasing) function.

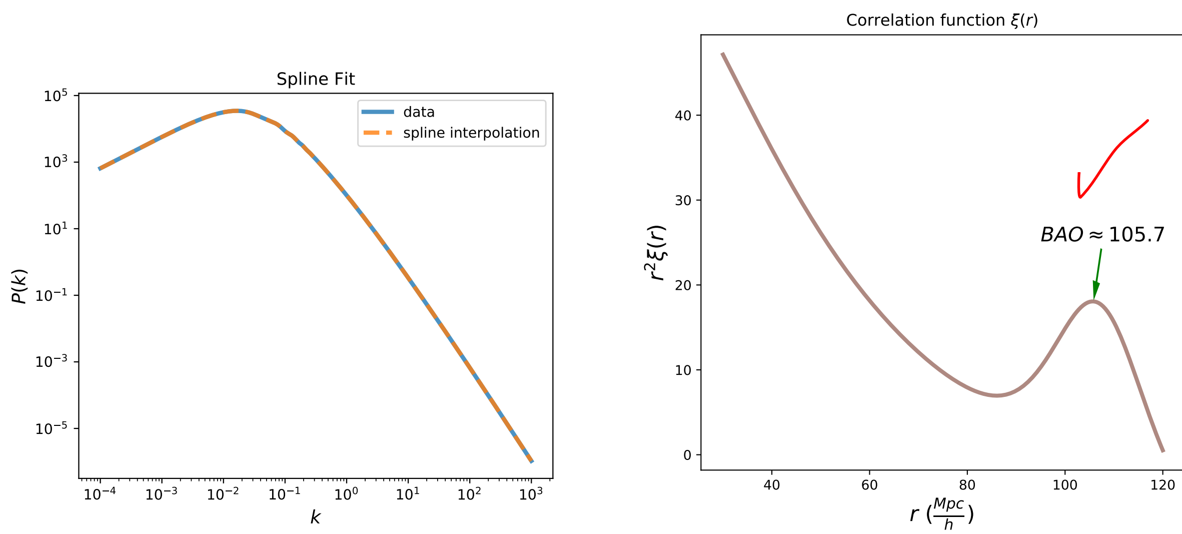
3 Problem 3

Determine the scale r of the peak of the bump, in the correlation function corresponding to BAOs and plot the enhanced correlation function $r^2\xi(r)$.

A spline interpolation is used to uniformly sample across the range of the power spectrum $P(k)$, which ran $10^{-4} \leq P(k) \leq 10^3$. The efficacy of the spline interpolation plot is demonstrated in left plot in Figure 4. Interpolating farther than this range resulted in non-physical deviations from the expected fit, and going lower was unnecessary for properly evaluating the integral, which was bounded below from 0. Moreover, $P(k)$ descreases exponentially for larger k, so the integrand would be near zero for those values, and so I did not need to (nor could with our interpolation techniques) integrate to $k = \infty$. The integral for each r value:

$$\xi(r)\frac{1}{2\pi^2}\int k^2 P(k)\frac{\sin(kr)}{kr}\mathrm{d}k \tag{2}$$

was calculated using $N = 10^6$ bins. This integral was performed for 200 values of r across the range $30 \leq r \leq 120$. The BAO bump was found to be $r_{BAO} \approx 105.7$, taken as the max value for $r > 80$. This can be seen in the right plot of Figure 4.



Got it!

Figure 4: Left: We see that the spline accurately interpolates $P(k)$. Right: A plot of the correlation function, enhanced by a factor of r^2 to highlight the Baryon Acoustic Oscillation (BAO) bump.