Homework 1

Jatin Abacousnac

September 27, 2019

Problem 1

Differentiate the functions cos(x) and e^x at $x = 0.1, 10$ using single precision forward-, central- and extrapolated-difference algorithms.
a) Write a code that implements these three methods.
b) Make a log-log plot of the relative error vs step size h and check whether the scaling and the number of significant digits obtained agrees with simple estimates.
c) Truncation and roundoff error manifest themselves in different regimes in these plots. Clearly identify these regimes.

A function $f(x)$ may be numerically differentiated via various methods, with increasing accuracy. The forward-difference method, being the simplest implementation, is formulated as:

$$\lim_{h\to 0} f'(x) = \frac{f(x+h) - f(x)}{h}, \tag{1}$$

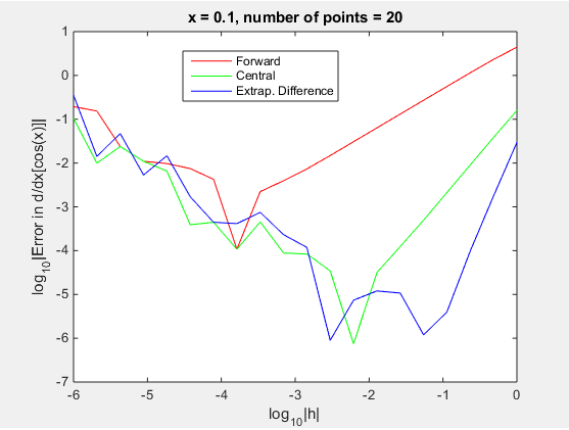
where h is known as the step size. An improved algorithm known as the central-difference method is expressed as:

$$\lim_{h\to 0} f'(x) = \frac{f(x+h) - f(x-h)}{2h}. \tag{2}$$

Finally, the user may opt for the extrapolation-difference method, which provides even greater accuracy:

$$\lim_{h\to 0} f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}. \tag{3}$$

These three methods were implemented for $x = 0.1$ and $x = 10$ using MATLAB (the code is uploaded to my GitHub repository). Variables were stored as single precision values.



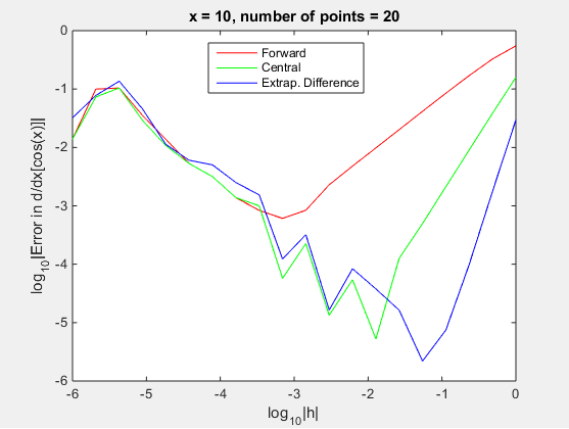
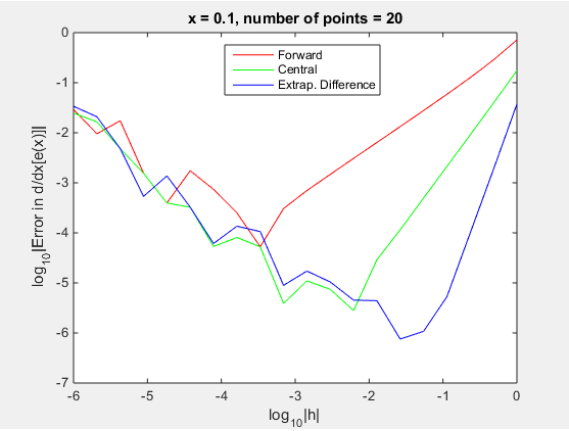


Figure 1: Log-log plots for the relative error v step size h, when evaluating the derivative of $\cos(x)$ at x=0.1 and x=10



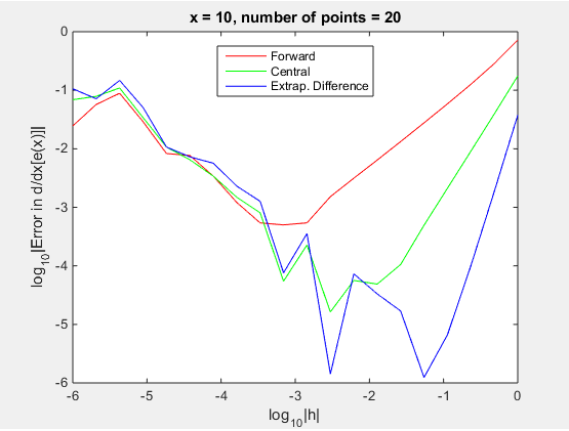


Figure 2: Log-log plots for the relative error v step size h, when evaluating the derivative of $\exp(x)$ at x=0.1 and x=10

The plots in Fig. 1 and Fig. 2 are shown for the three different methods of differentiation. From the plots it is clear that when a large number of bin sizes is used, the three methods are comparable, and when we use a lower number of bin sizes, the extrapolated-difference method largely outperforms the central-difference method, which, in turn, outperforms the forward-difference method.

From the plots, for larger values of h, we quickly notice that the forward-difference method has an error that scales linearly ($\propto h$), while the central-difference method has an error that scales with h^2, and the extrapolated difference outperforms both by having an error that scales with h^4. This regime corresponds to the errors associated with truncation error, and are in accordance with theory. For smaller values of h, we notice that ϵ scales roughly with $1/h$, which corresponds to roundoff error (itself related to machine precision).

The number of significant figures for the derivative being evaluated keeps improving, until h becomes too small, and roundoff error (being $\propto 1/h$) keeps increasing. This is clearly shown, and the switchover happens around $h = [0.0001, 0.01]$.

Problem 2

Consider the integral,

$$I = \int_0^1 e^{-t} dt, \tag{4}$$

and compare the relative error, ϵ, for the midpoint rule, trapezoid rule, and Simpsons rule for single precision.
a) Write code that implements each method.
b) Make a log-log plot of ϵ as a function of number of bins N. (Since this is a log-log plot, choose values that scale in a reasonable fashion). Make N large enough such that you see the effects of roundoff error.
c) Explain what you see in the plot.

Likewise, numerical integration can be performed using several methods. The most rudimentary is the midpoint rule, where an interval $[a, b]$ is divided into N bins, over which the function to be integrated is assumed to be linear. Each term is given by:

$$I = \sum_{i=1}^{N} \frac{b-a}{N} f(x_i), \tag{5}$$

where $h = (b\text{-}a)/N$ and $x_i = (x_i + \frac{1}{2}h)$. The trapezoid rule (a series of Riemann sums) computes the integral as

$$I = \frac{1}{2} h \sum_{i=1}^{N} [f(x_{i-1}) + f(x_i)], \tag{6}$$

and proves to be less accurate than the midpoint rule. An improvement upon these methods is Simpson's Rule, where we now model the function within a bin as being a quadratic. Then,

$$I = \frac{1}{3}h[f(a) + f(b) + 4\sum_{i=1}^{N-1} f(a+ih) + 2\sum_{i=2}^{N-2} f(a+ih)]. \tag{7}$$

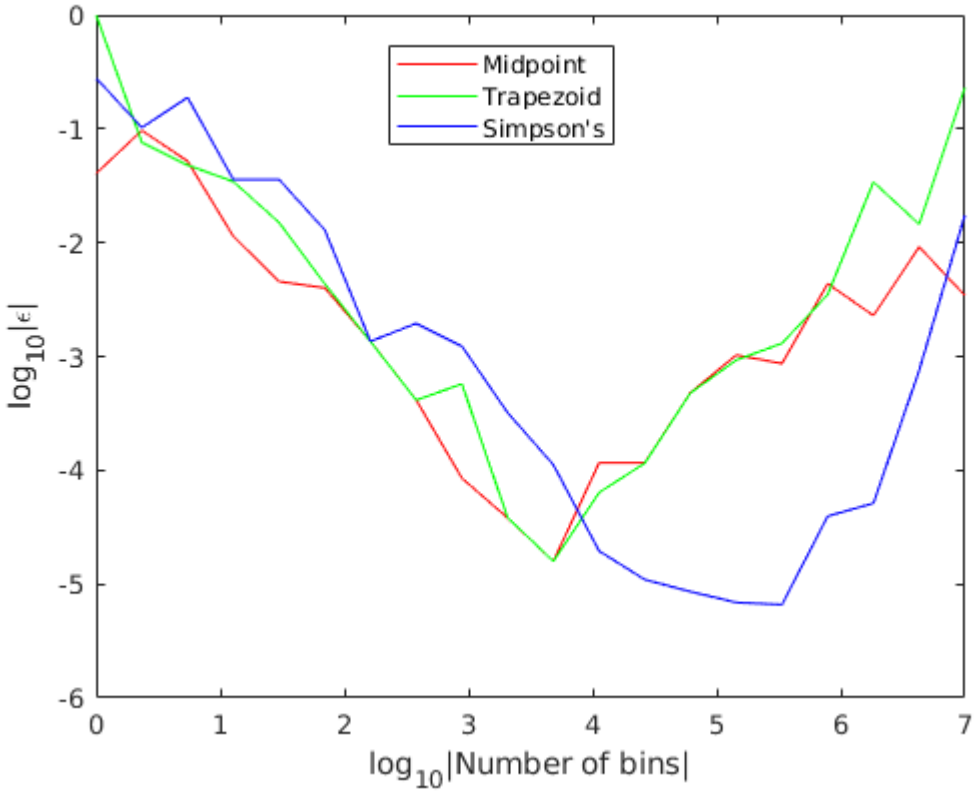


Figure 3: The three methods outlined are used to evaluate the integral of e^{-t} over the interval $t = [0, 1]$. The relative error associated with each method is evaluated for a different number of bins.

We note, from the plot, that there are two regimes of interest:

- In the small N (or, large h regime), roundoff error is absent and the three methods are comparable. However, I expected Simpson's Rule to outperform these two methods, which did not appear to be the case. I expected the error to scale in roughly the same way but to be systematically lower than the other two methods. This could indicate

an error in my implementation of the algorithm, which was checked repeatedly, but ended up giving the same results. It is interesting to note that, while the relatives errors for the midpoint and trapezoid methods scale in the same way with the number of bins, the midpoint method performs slightly better (as discussed in Lecture 7, dated 26 September 2019).

- In the large N regime, Simpson's Rule outperforms the other two methods, and round-off error kicks in for all three methods, around $N = 1 \times 10^4$.

Problem 3

In cosmology, density fluctuations in the matter distribution are characterized by a power spectrum, $P(k)$, the rms amplitude fluctuations of the density waves, as a function of wavenumber k (with units of h/Mpc). In configuration space, these density fluctuations are described by the correlation function, $\xi(r)$, at a given scale r, usually in Mpc/h. These two are quantities are related

$$\xi = \frac{1}{2\pi^2} \int dk\, k^2 P(k) \frac{sin(kr)}{kr} \tag{8}$$

With this homework, I have attached a tabulated power spectrum. The first column is k and the second column is $P(k)$ (pay no attention to the third column). Using whatever integration method you prefer, use the above equation to calculate $\xi(r)$ in the range r = [50,120] Mpc/h. The power spectrum is tabulated in logarithmic intervals in k, due to its power-law like nature. You may choose to use an interpolation technique, such as cubic spline, to help evaluate the integral.

Around $k \approx$ 0.1, you can see oscillatory behavior in $P(k)$. We call these the "baryon wiggles," and they manifest as a single "bump" in the correlation function at large scales. Using your calculation for $\xi(r)$, determine the scale, r, of the peak of this bump.

Make a plot of $r^2\xi(r)$ over the required range in r (multiplying by r^2 visually enhances the bump). Indicate on this plot the scale of the peak, also known as the "baryon acoustic oscillation" (BAO) peak.

Notes:
a) If you use spline interpolation, you are not required to code that up yourself. You may use a pre-packaged routine (or numerical recipes code).
b) Formally, the limits of the integral are from k = 0 to $k = \infty$. Note that $P(0)$ = 0. You may choose a finite upper limit, provided you can determine if your limit is robust.
c) The 'h' in the distance units refers to the Hubble constant $h = H_0/100$, which sets the distance scale and is thus incorporated into the distance units, since its value is unknown.

For this problem, we are given a text file with k and $P(k)$ values. The goal is to relate the correlation function $\xi(r)$ to some scale r.

First, we determine a value of k over which the integral may be evaluated. For these values of k, $P(k)$ must be obtained, and subsequently integrated over. For this purpose, MATLAB's built-in cubic spline function is used to interpolate the k and $P(k)$ data, so that $P(k)$ can be evaluated for any value of k.

The interpolation is shown below, as a log-log plot.

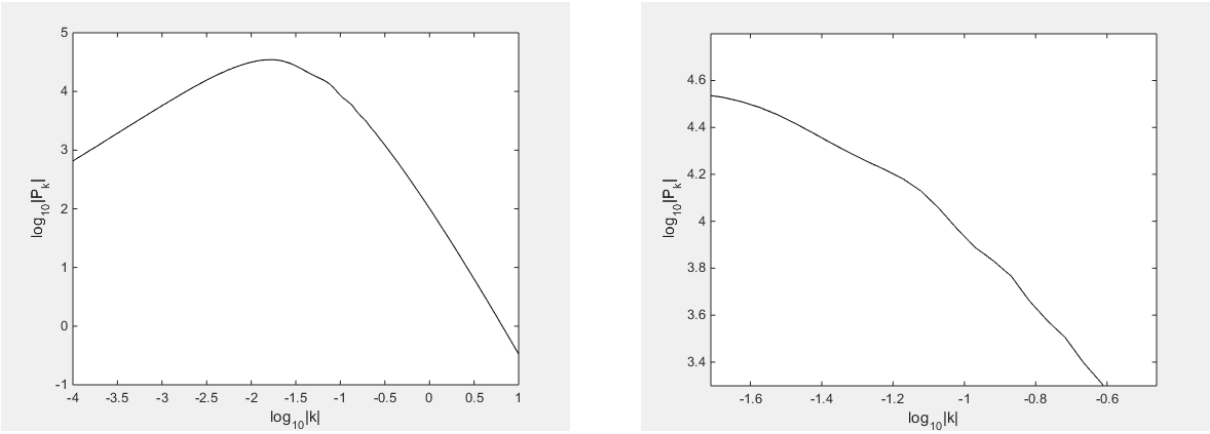


Figure 4: On the left is a log-log plot of $P(k)$ v. k, where the data has been interpolated using a cubic spline. At around $k = 0.1$, oscillatory behavior is noticeable, as shown on the right (zoomed-in portion of the plot on the left around $k = 0.1$).

To evaluate the correlation function, we employ the midpoint-method, with a large number of bins (here, the number of bins was set to be 2×10^6). For a value of k, $P(k)$ is evaluated using the spline interpolation. Then, the integrand from equation (8) is found and summed over a large interval of k (roughly 1000, here). This process is repeated for values of r in the interval [0, 150]. A visual representation of the result is shown in a log-log plot of $r^2\xi$ against r.

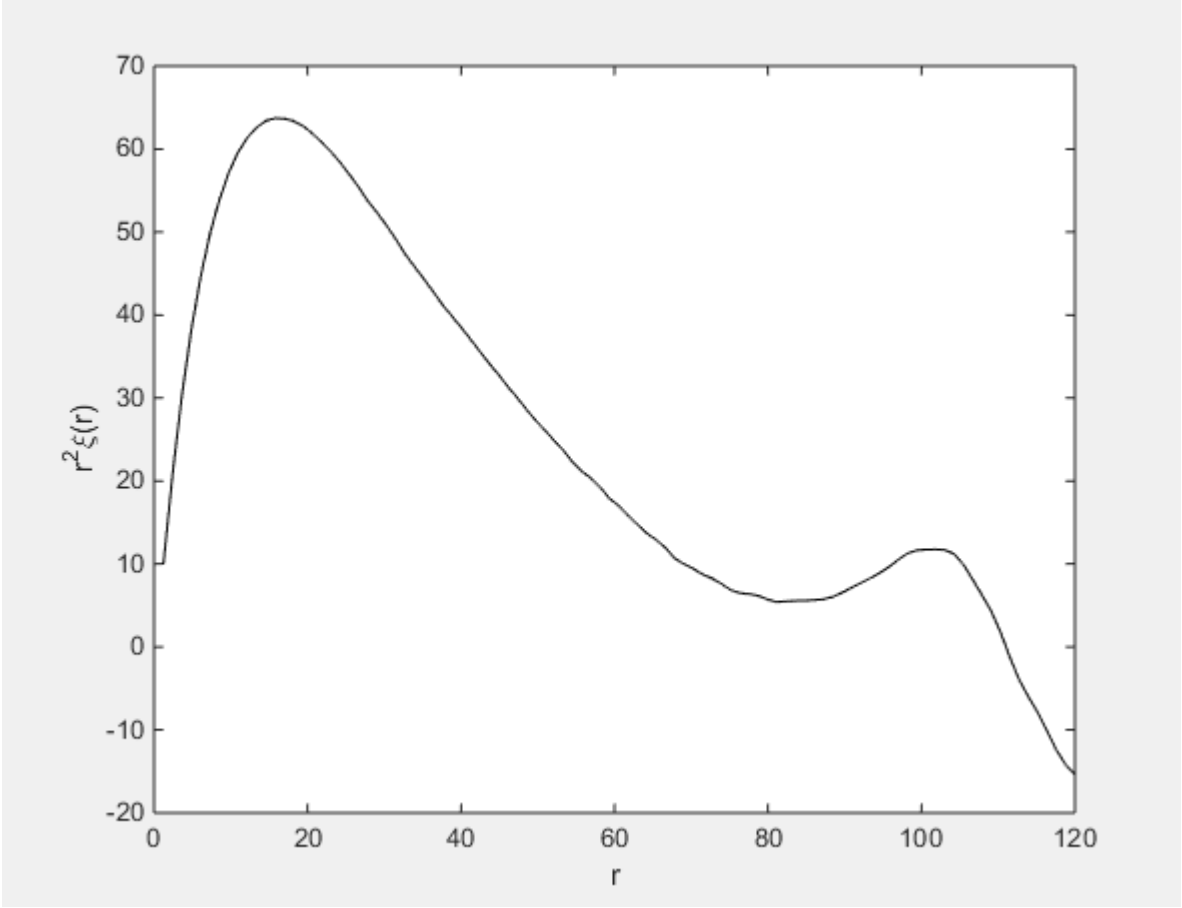


Figure 5: Here is a plot of ξr^2 v. r, showing the correlation of $P(k)$ with r. The oscillatory behavior in the vicinity of $k = 0.1$ in Fig. 4 is prominently shown as the second bump in this plot. This occurs roughly at a scale of $r = 100$, and is also known as the Baryonic Acoustic Oscillation peak.