[PHYS-GA2000] Problem Set 5

Dylan Lane
Github: dl4729

October 17, 2023

Problem 1

Methods and Results

We have (1)

$$\Gamma(a) = \int_0^\infty x^{a-1}e^{-x}dx \tag{1}$$

, the integrand of which can be plotted on an interval $x \in [0, 5]$ for fixed a with no additional processing (Figure (1)). As described in (1), each curve for $a = 2, 3, 4$ starts at 0, rises to a maximum, then decays back down. To find the maximum, simply take the derivative and set it to 0; $(a-1)x^{a-2}e^{-x} - x^{a-1}e^{-x} = 0$ which implies $(a-1-x) = 0$, this the maximum is attained at $x = a - 1$.

To integrate this function, consider the change of variables $z = \frac{x}{c+x}$ for some c which maps the improper bounds of equation (1) onto $[0, 1]$. We want the peak of the data to occur at $x = 1/2$ so information is not lost during this rescaling; in particular, $1/2 = \frac{x_max}{c+x_max}$ which is trivially satisfied by $x_max = c$, so $c = a - 1$ is an appropriate choice of parameters. With some algebra, it can be shown that if $z = \frac{x}{c+x}$ then $x = \frac{zc}{1-z}$. Moreover, $dx = \frac{c}{(1-z)^2}dz$. Let $f(x) = x^{a-1}e^{-x}$; since $x^{a-1} = e^{ln(x^{a-1})} = e^{(a-1)ln(x)}$, we can rewrite $f(x) = e^{(a-1)ln(x)-x}$. In conclusion, equation (1) can be transformed to

$$\Gamma(a) = \int_0^1 \frac{c}{(1-z)^2} f(\frac{zc}{1-z})dz \tag{2}$$

. This can be computed numerically; after all the processing a bound scaling, simple Gaussian quadrature seemed the most natural way to compute the integral. The results are shown in Figure (2) and agree well with the true values of the gamma function. It is worth noting that equation (1) could be computed perfectly with Gauss-Laguerre integration and no modifications of the integral; since the problem prescribes these transformations, Gauss-Laguerre becomes harder to use because the bounds are from 0 to ∞.

Figures

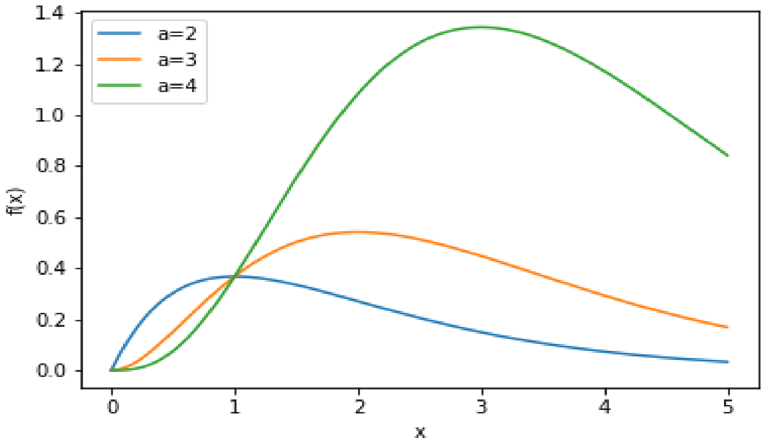

Figure 1: Integrand of Equation (1) Plotted for Multiple Values of a. Units not provided.

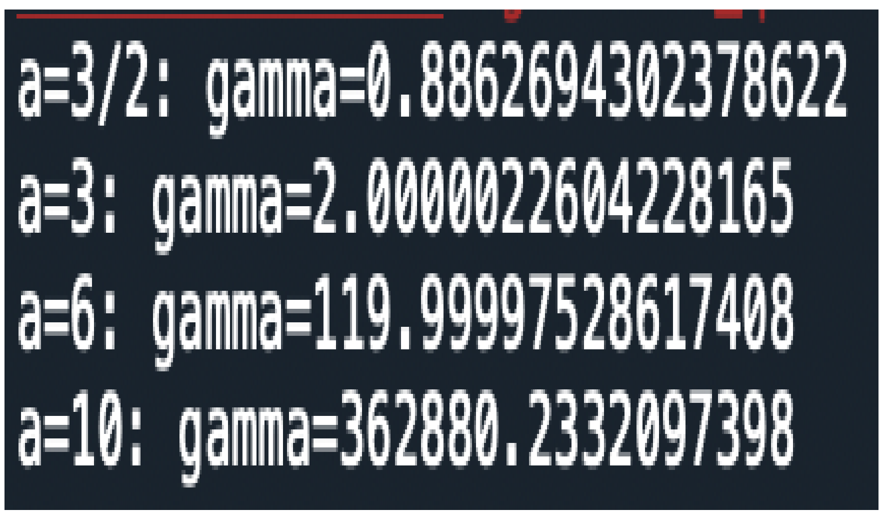

Figure 2: Calculation Results

Problem 2

Methods and Results

The data is first loaded and put into the desired form; Figure (3) shows all of the data plotted with the time axis rescaled to protect against numerical instabilities emerging from taking large powers of big numbers. Questions (b) and (d) are closely related, so I will describe the general process of generating these plots from which (b) will be a special case $N = 3$. For a given order N polynomial, we populate the matrix A (dimensions (N+1, len(time)) such that each row is a copy of the time data scaled to unity and raised to some power. As outlined in (1) (2) (3) we can do SVD on this matrix A so $A = UwV^T$, take the psuedoinverse of w, calculate $A^{-1} \approx Vw^{-1}U^T$, apply this inverse map to the signal data, then again apply A to find the best Nth order polynomial approximation of signal.

For $N = 3$, the fit is shown in Figure (4) and the residuals in Figure (5). The residuals reveal that this is a poor fit; assuming the uncertainty is Gaussian, there are several points whose residuals are up to 4 standard deviations from 0 which should be exceedingly rare if the fitting is good. Moreover, there is a clear periodicity to the residuals; non-uniform residuals indicate that the signal has features (like oscillations) which the fit is failing to capture. Figure (6) shows several higher order polynomial fits; the first polynomial to truly capture the oscillatory nature of the signal is $N = 25$. Indeed, the residuals look better too; they are mostly within two standard deviations of 0 and are uniformly distributed in x. However, the condition number for A is $1.4022367365824752e + 16$, a very large number which implies that this is prone to numerical instabilities. It therefore seems likely that there is no good polynomial fit; for N sufficiently large the features of the signal can be captured, but the condition number is very high so the solution is unreasonable.

Now consider a Fourier formulation of the signal which seems natural because the signal looks oscillatory. Assume (as stated in the problem) that $T = max(time)/2$, and $\omega = 2\pi/T$. Observe that ω will be very small and t very large, so ωt may suffer greatly from roundoff error. Instead, I choose to use $t/tmax$ as a parameter and then renormalize ω to be dimensionless so their product is still dimensionless. In particular, $\omega' = 2\pi t_{max}/T = 4\pi$. Then, letting t' denote the normalized time, we choose an order N and populate A such that each row is either constant, $\sin(n\omega' t')$, or $\cos(n\omega' t')$ for $1 \leq n \leq N$. I also include a linear term (3) since the signal clearly has some non-oscillatory increase in time which is difficult to approximate with a limited number of Fourier terms. The resulting fit and residuals for $N = 10$ is shown in Figures (7) and (8) respectively. Like the high order polynomial, the residuals are largely within two standard deviations of 0 as expected of a good fit; the condition number is of order 1 (in fact 4.630938888349239), so unlike the polynomials this is a reasonable solution.

Figures

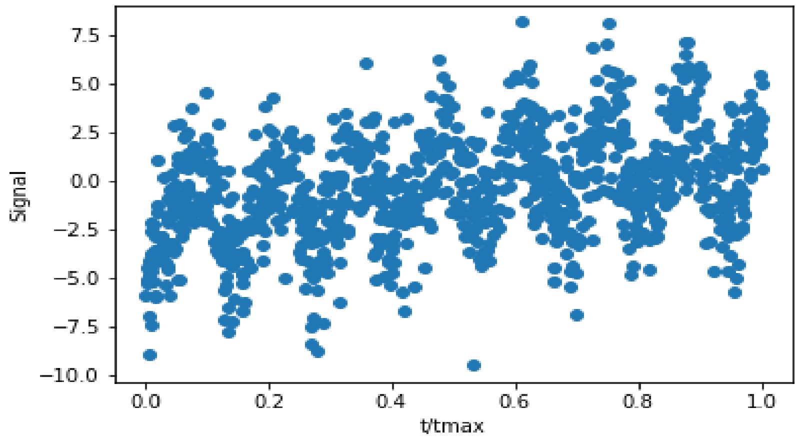

Figure 3: Data Plotted with Renormalized Time Axis.

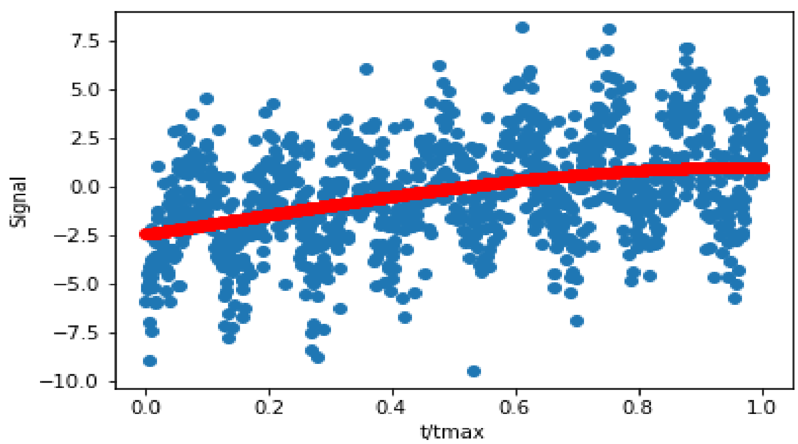

Figure 4: Third Order Polynomial Fit of Signal Data.

References

[1] Newman, M. 2012, Computational Physics (Createspace Independent Pub)

[2] https://blanton144.github.io/computational-grad/

[3] https://github.com/mcmorre/computational_TA/blob/main/PS5_hints.ipynb

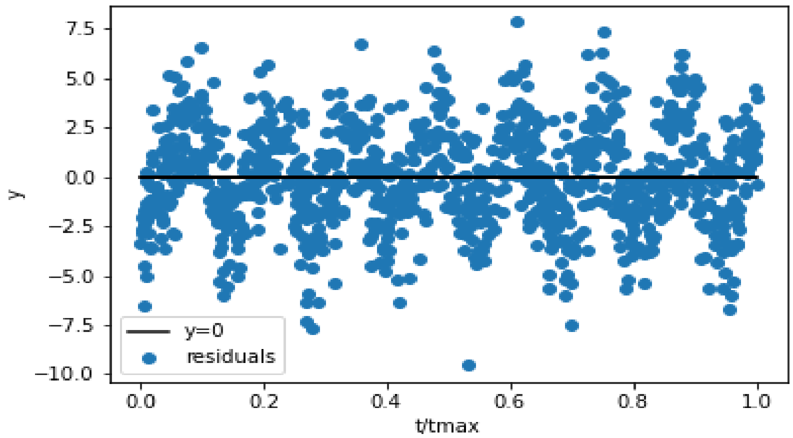

Figure 5: Residuals of Third Order Fit.

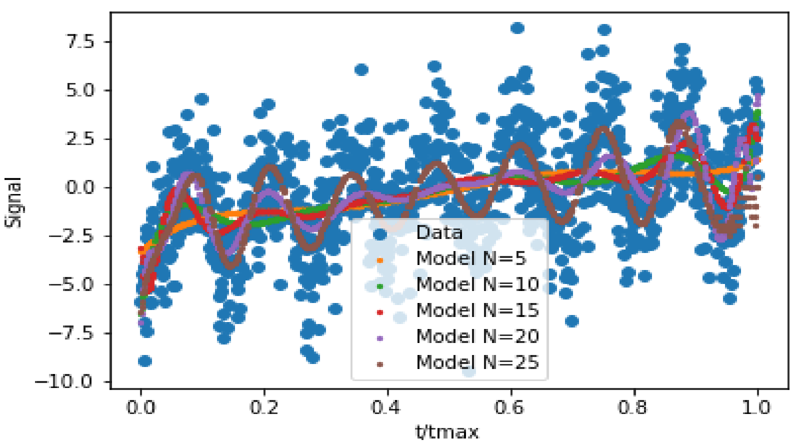

Figure 6: Several polynomial fits of higher order. Observe order 25 matches the form of the data fairly well.

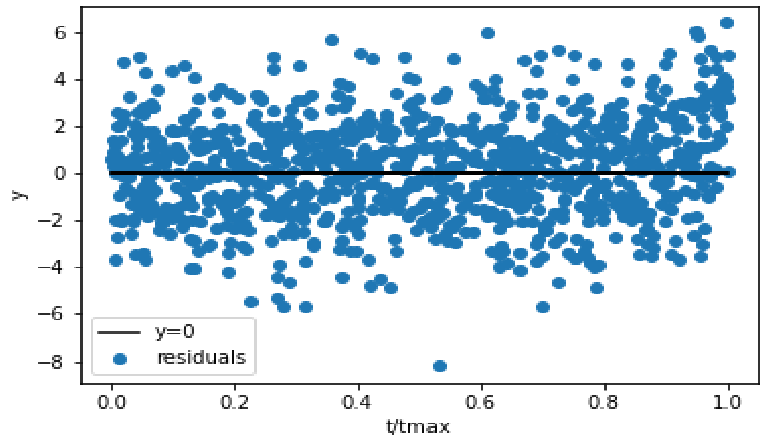

Figure 7: Residuals of 25th Order Polynomial.

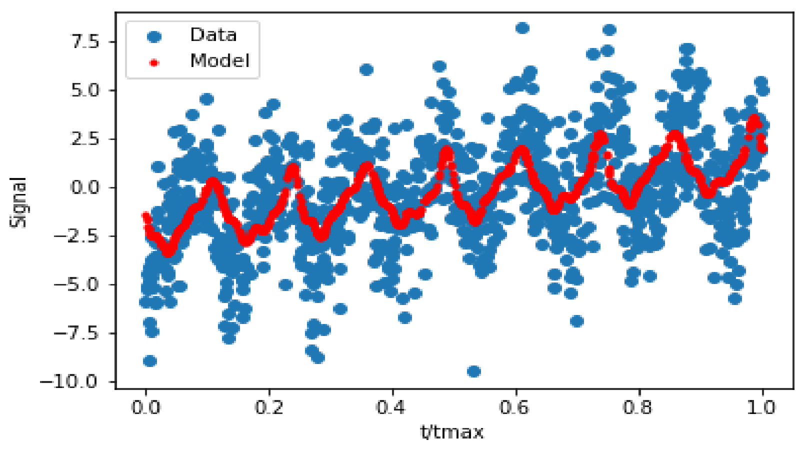

Figure 8: Fit of 10th Order Fourier Series plus a nonzero offset and linear term.

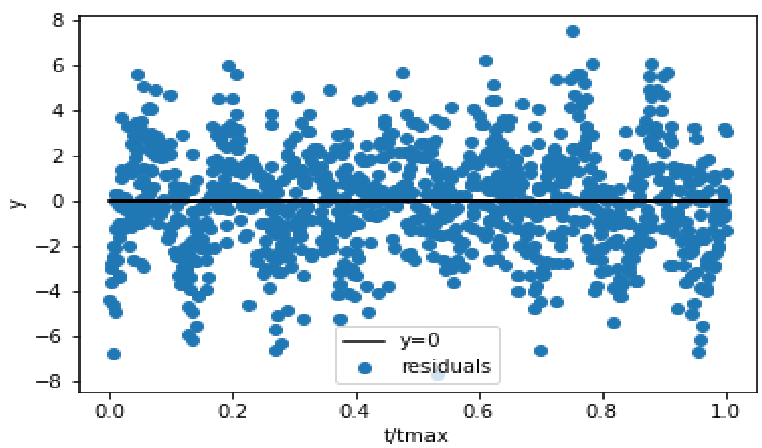

Figure 9: Residues of 10th Order Fourier fit.