Problem Set 5

Dario Loprete

October 8, 2024

GitHub link

https://github.com/dariolop76/phys-ga2000

1 Problem 1

In the code `prob_1.py`, we create four user-defined functions:

- `f(x)` returns the value of the function $f(x) = 1 + \frac{1}{2}\tanh 2x$;
- `df_dx(x)` returns the derivative of the function f(x), which is given by $\frac{df}{dx} = 1 - \tanh^2 2x$;
- `df_dx_cd(x, dx)` evaluates the derivative of f(x) using the central difference method, namely $\frac{df}{dx} \simeq \frac{f(x+\frac{dx}{2})-f(x-\frac{dx}{2})}{dx}$;
- `f_jax` returns the function with the use of the package `jax`.

We use $dx = 10^{-5}$ and evaluate the derivative of the function $f(x)$ in the range $-2 \leq x \leq +2$ using the central difference method. Figure 1 shows the analytic result compared to the numerical estimate.

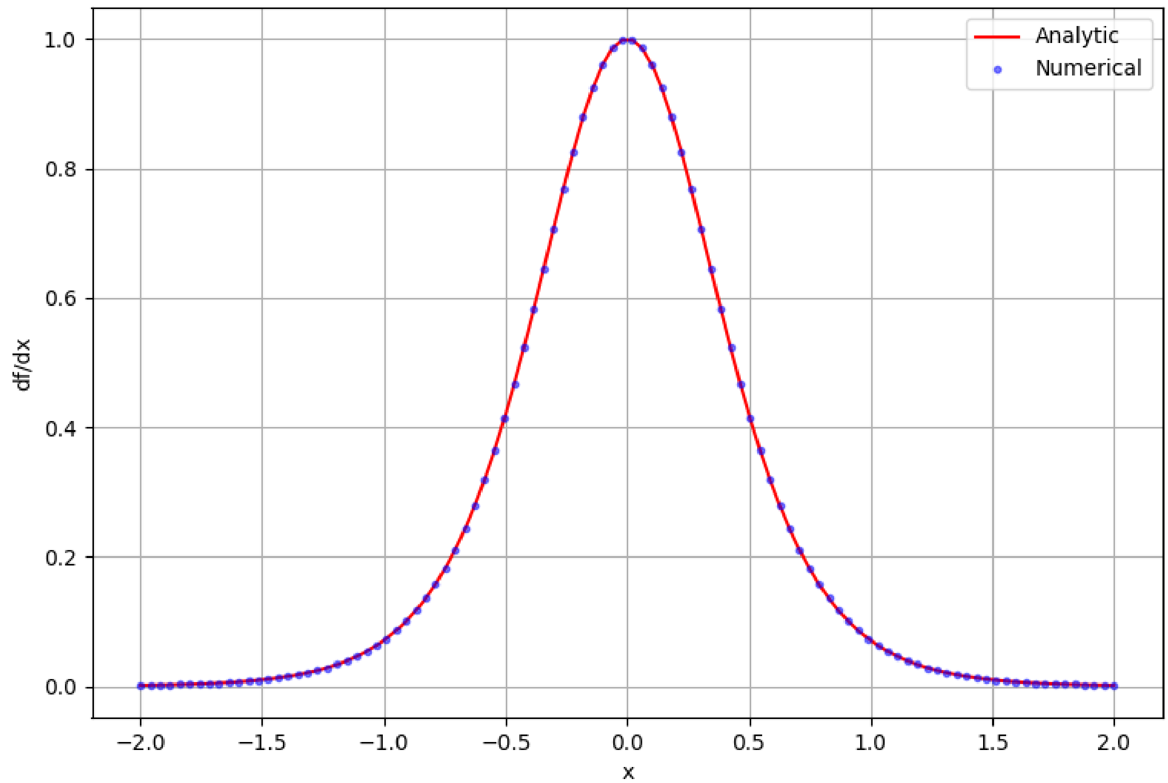

Figure 1: Derivative of the function $f(x)$: in red, the analytic form of the derivative, computed using `df_dx`; in blue, the numerical evaluation of the derivative using `df_dx_cd`, namely the central difference method.

Figures 2 and 3 show the difference between the analytic result and the numerical computation of the derivative using, respectively, the functions `df_dx_cd(x, dx)` and the `jax` package.

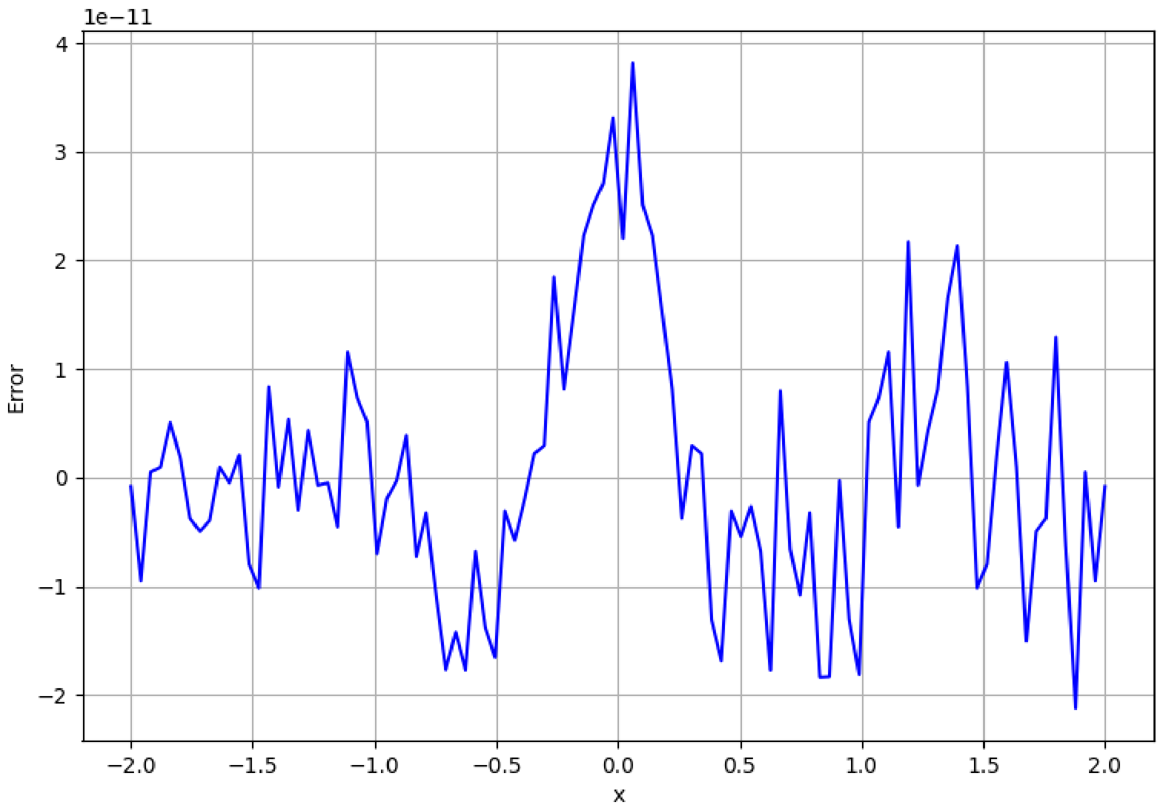

Figure 2: Difference between the analytic derivative and the numerical one obtained using the central difference method.

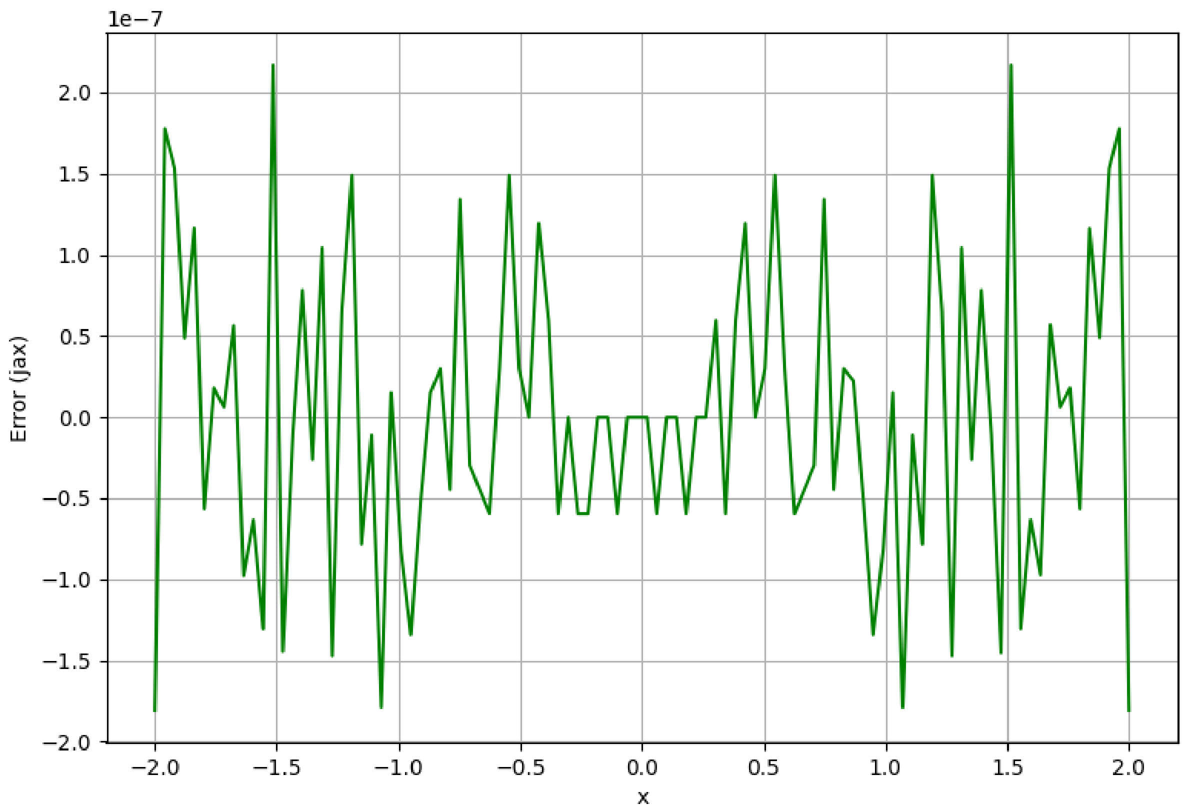

Figure 3: Difference between the analytic derivative and the numerical one obtained using the `jax` package.

2 Problem 2

The code `prob_2.py` performs the evaluation of the integral which returns the Gamma function, defined as follows:

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

2.1 Part a

Figure 4 shows the integrand of the Gamma function in Equation 1 for $a = 2, 3, 4$ in the range $0 \leq x \leq 5$.

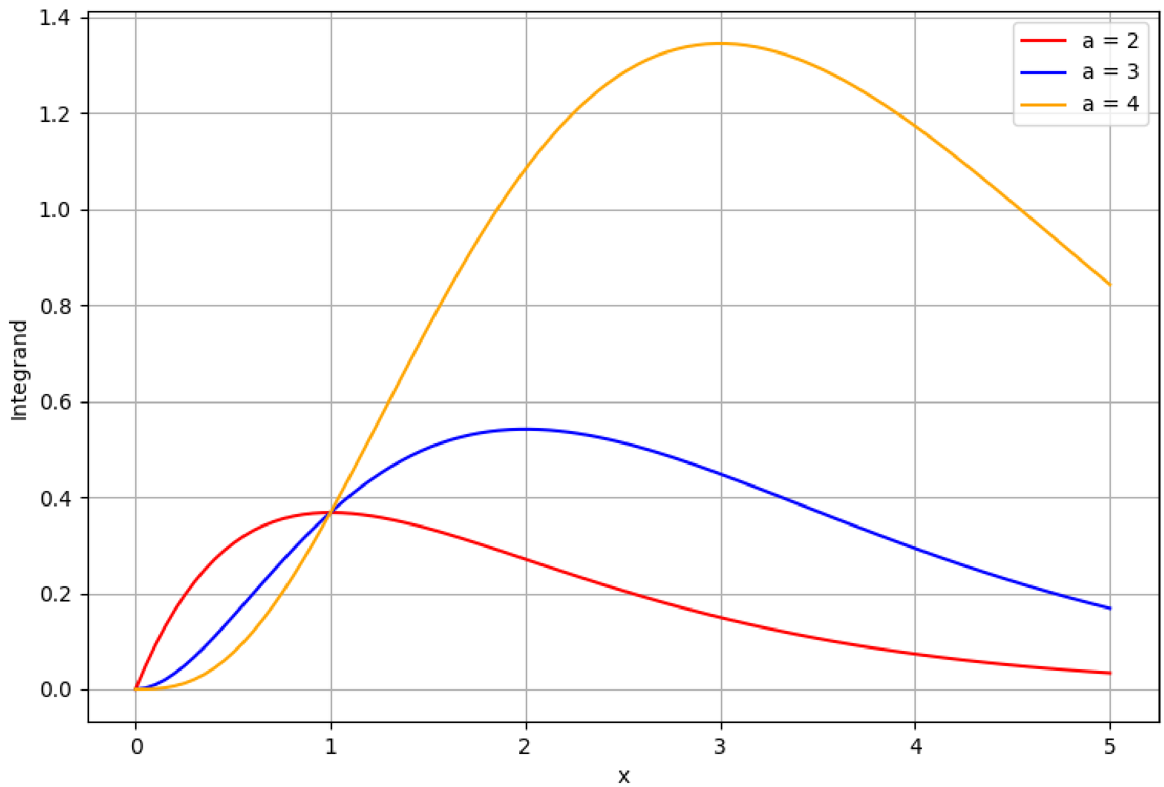

Figure 4: Integrand of the Gamma function for $a = 2, 3, 4$ in the range $0 \leq x \leq 5$.

As we can see, we obtain the expected behaviour, namely that the integrand starts at zero, rises to a maximum, and then decays again.

2.2 Part b

Let's show analytically that the maximum of the integrand falls at $x = a-1$. To do so, we compute its derivative with respect to x.

$$\frac{d}{dx}\left(x^{a-1}e^{-x}\right) = (a-1)x^{a-2}e^{-x} - x^{a-1}e^{-x} = \tag{2}$$

$$= x^{a-2}e^{-x}[(a-1)x^{-1} - 1] \tag{3}$$

By imposing that the first derivative be equal to zero we find $x = a - 1$. Let's now check that the second derivative is negative at this point to make show that it is a maximum:

$$\frac{d}{dx}\left(x^{a-1}e^{-x})\right)\Big|_{x=a-1} = -(a-1)^{a-2}e^{-a+1}\left(\frac{1}{a-1}\right) \tag{4}$$

which, for $a > 1$, is negative.

2.3 Part c

Given the change of variable

$$z = \frac{x}{c+x}, \tag{5}$$

we can see that $z = \frac{1}{2}$ is obtained when $x = c$. So, the appropriate choice of the parameter c that puts the peak of the integrand of Equation 1 at $z = \frac{1}{2}$ is $c = a - 1$.

2.4 Part d

In order to avoid potential overflow or underflow caused by the multiplication between a large value of x^{a-1} and a small value of e^{-x}, we can rewrite the integral in Equation 1 as

$$e^{(a-1)\left(\log x - \frac{x}{a-1}\right)}. \tag{6}$$

In this way, the computer will evaluate the difference between $\log x$ and $\frac{x}{a-1}$ which is less likely to be subject to overflow or underflow, since the logarithm increases gradually with x, while the term $\frac{x}{a-1}$ grows linearly. Hence, the exponent of the exponential is numerically more stable for large values of x.

2.5 Part e

We perform the evaluation of the integral using the Gauss-Legendre quadrature via the function `np.polynomial.legendre.leggauss(N)`, where the number of sample points is $N = 50$. For this purpose, we make use of the precautions from Part c) and Part d). The obtained value for $\Gamma(\frac{3}{2})$ and the difference with the analytic result of $\frac{\sqrt{\pi}}{2}$ are:

Numerical result for $\Gamma(\frac{3}{2})$: 0.8862272
Difference: 3e-07

2.6 Part f

a	$\Gamma(a)$ - numerical result	$a!$ - analytical result	Difference
3	2.00000000000006	2	$+6 \cdot 10^{14}$
6	119.9999999999994	120	$-6 \cdot 10^{13}$
10	362879.999999998	362880	$-2 \cdot 10^{-9}$

Table 1: Numerical results for $\Gamma(a)$ with $a = 3, 6, 10$ and comparison with analytic results.

3 Problem 3

3.1 Part a

Figure 5 shows the plot of the data.

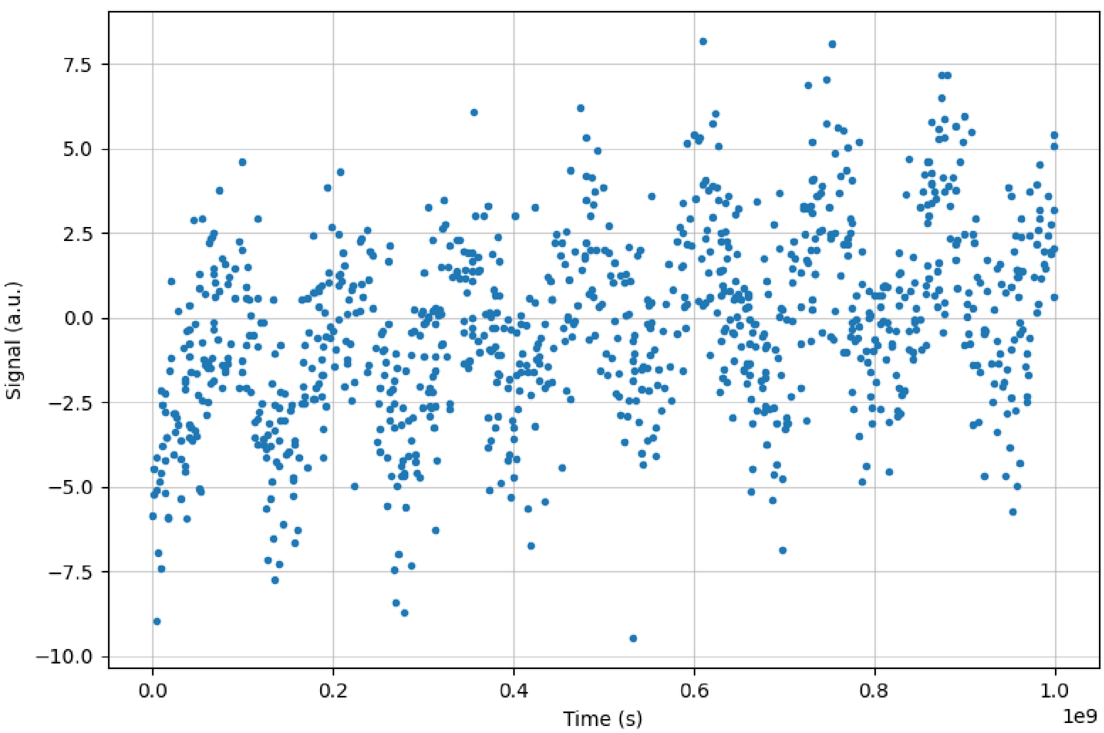

Figure 5: Signal vs time.

3.2 Part b

We apply the SVD decomposition to perform a fit of the data. For this part, our model is a 3^{rd}-order polynomial: $1 + x_1\, t + x_2\, t^2 + x_3\, t^3$, where t is the time, while x_i are the parameters we want to find. These are obtained as follows:

$$\vec{x} = A^{-1}\vec{b}, \tag{7}$$

where $\vec{b}$ is a vector containing our data and A is the design matrix we use for our model and to which we apply the SVD decomposition $A = UWV^t$. Our fit $\vec{m}$ is then obtained as $\vec{m} = AA^{-1}\vec{b}$. Additionally, to have a numerically stable SVD solution we rescale the temporal variable as follows:

$$t' = \frac{t - \bar{t}}{\sigma_t}, \tag{8}$$

where $\bar{t}$ and σ_t are the mean and the standard deviation for the time. We also compute the condition number, which is given by the ratio of the highest to lowest eigenvalues in W, to track the numerical stability of the fitting procedure.

Figures 6 and 7 show, respectively, the 3^{rd}-order polynomial fit of the data and the corresponding residuals, namely the difference between our model and the data. The obtained condition number is 6.

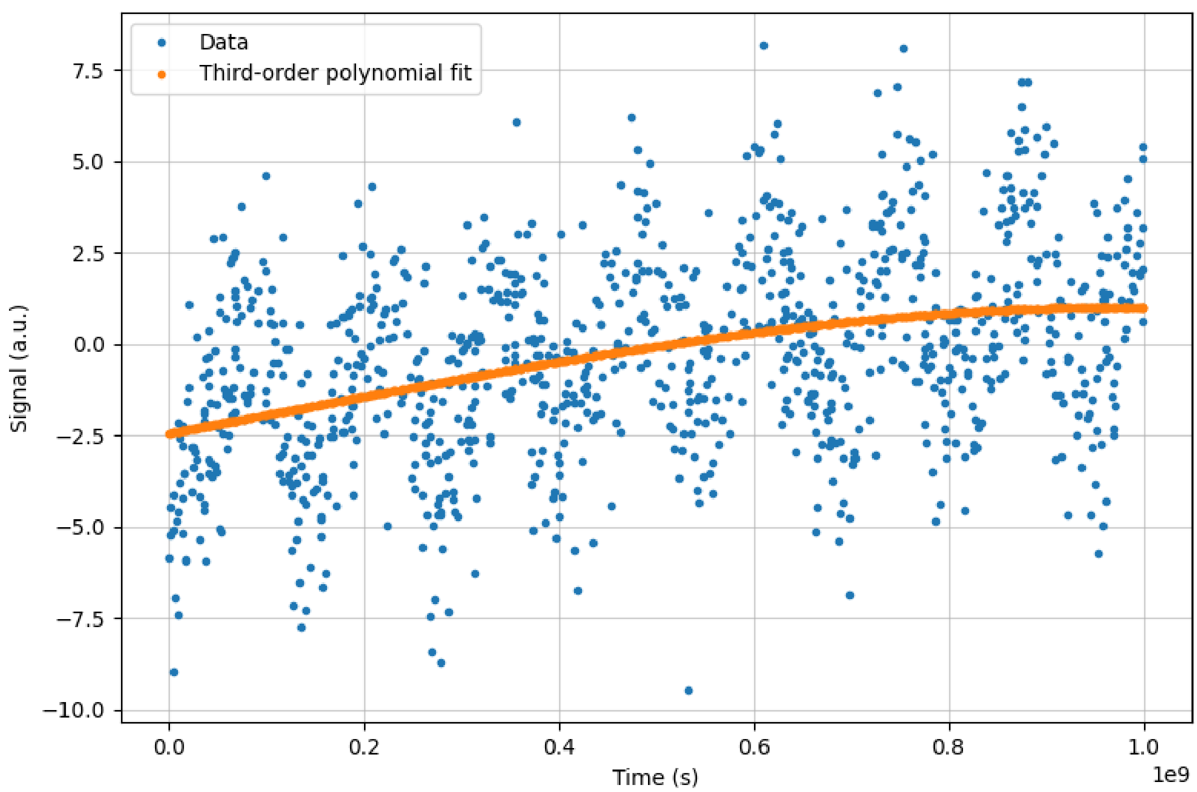

Figure 6: Fit of the data using a 3^{rd}-order polynomial.

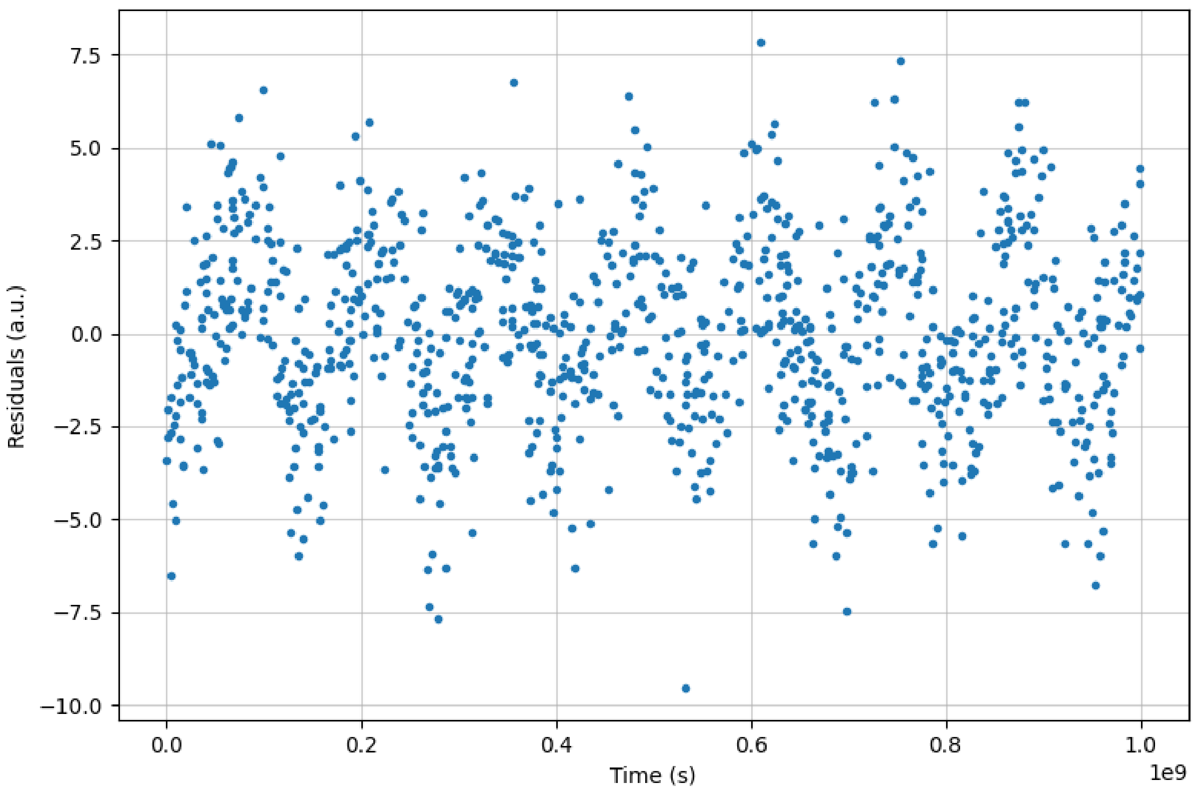

Figure 7: Residuals for the 3^{rd}-order polynomial fit.

3.3 Part c

As we can see from Figure 7, the 3^{rd}-order polynomial does not provide a good fit. Indeed, the residuals show a pattern instead of, say, being normally distributed around 0. Additionally, given that the uncertainty of the data is equal to $\sigma = 2$, we can also see that a non negligible part of the residuals is outside the range $[-2, 2]$, meaning that this model is not a good explanation of the data.

3.4 Part d

For this part, we wish to find the optimal order for a polynomial fit, which is of the form

$$1 + x_1\ t + x_2\ t^2 + ... + x_n t^n. \tag{9}$$

Our approach is the following: we fit the data with polynomials of increasing order, starting with order n = 3, and increase this order by one at each step. We do this until the condition number reaches a threshold, which we choose to be 10^{12} (smaller than the inverse of the machine precision). The optimal fit is obtained by finding for which polynomial fit the mean of the absolute value of the residuals is the smallest. The application of this method returns $n = 27$ as the optimal result. The obtained condition number is $3 \cdot 10^{11}$.

Figures 8 and 9 show, respectively, the optimal polynomial fit to the data and the corresponding residuals.

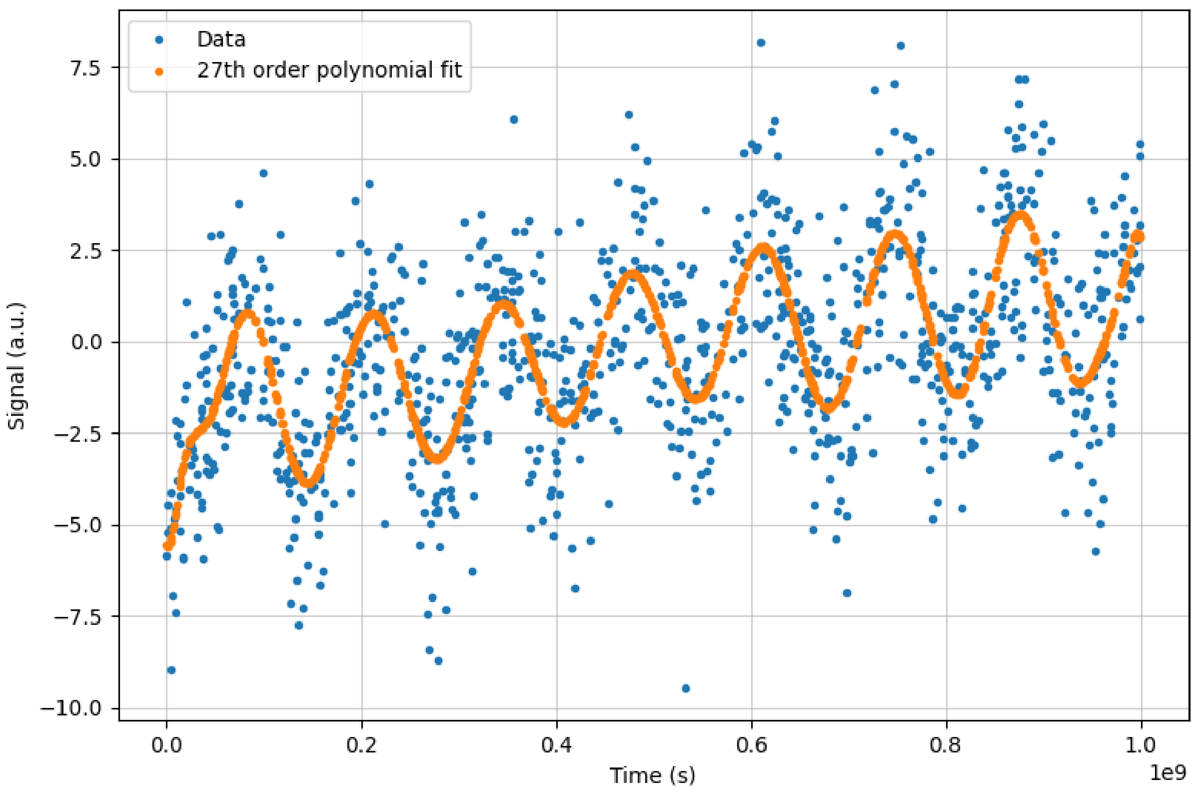

Figure 8: Fit of the data using an 27^{th}-order polynomial.

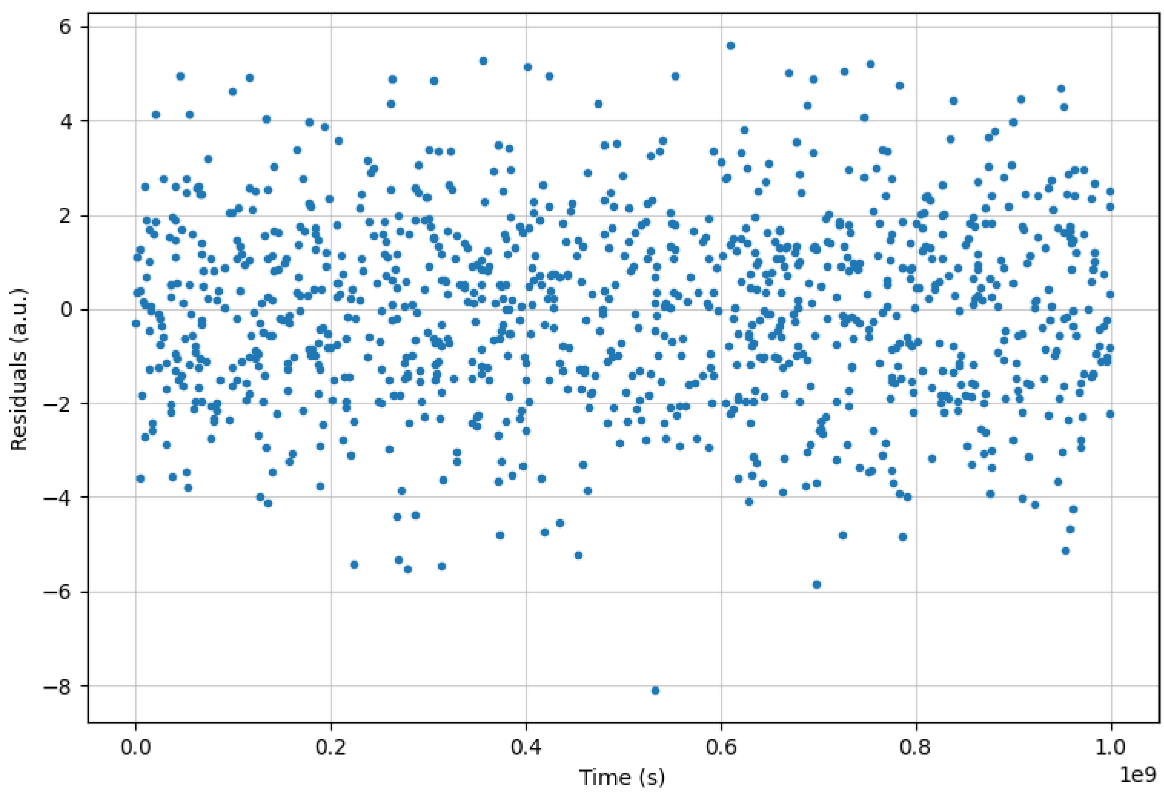

Figure 9: Residuals for the 27^{th}-order polynomial fit.

As we can see, this fit provides a better explanation of the data than that of Part b. The residuals mostly lie in the range $[-2, 2]$ and don't show a particular pattern.

3.5 Part e

For this part, we use a model based on a harmonic sequence, namely

$$1 + \sum_{k=1}^{n} \left[\sin\left(\frac{2\pi}{T} k \ t\right) + \cos\left(\frac{2\pi}{T} k \ t\right) \right], \tag{10}$$

starting with a period T equal to half of the time span covered. The order n which we use for this model is set to be $n = 10$. The choice of this number is based on a qualitative evaluation of the plot of the fitted data for different values of n. We chose this approach because we were not able to find an optimal n as in Part d; the condition numbers we found all exceeded the inverse of the machine precision and the residuals showed that the corresponding fit does not provide a very good explanation of the data, as in Part d. Among all values of n we tried, $n = 10$ is the one that provides an acceptable fitting plot. This is shown in Figure 10, while the residuals are displayed in Figure 11.

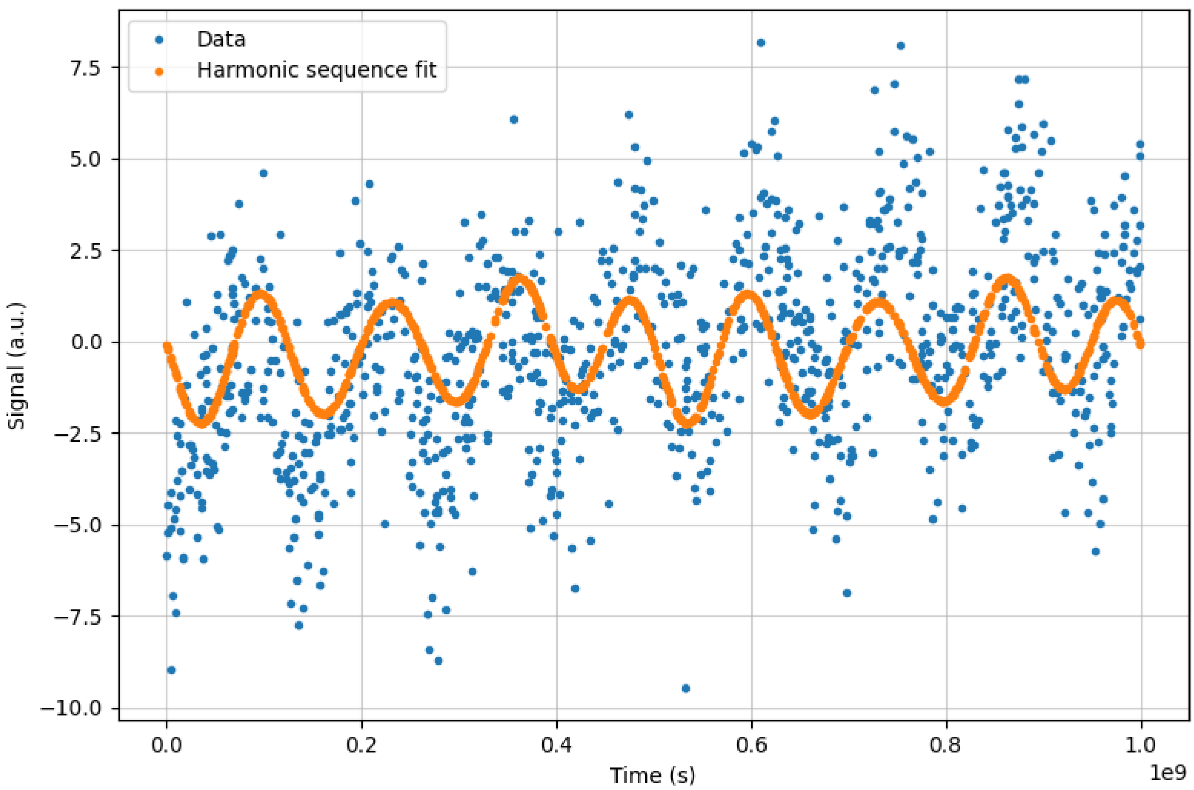

Figure 10: Fit of the data using a harmonic sequence with $n = 10$.

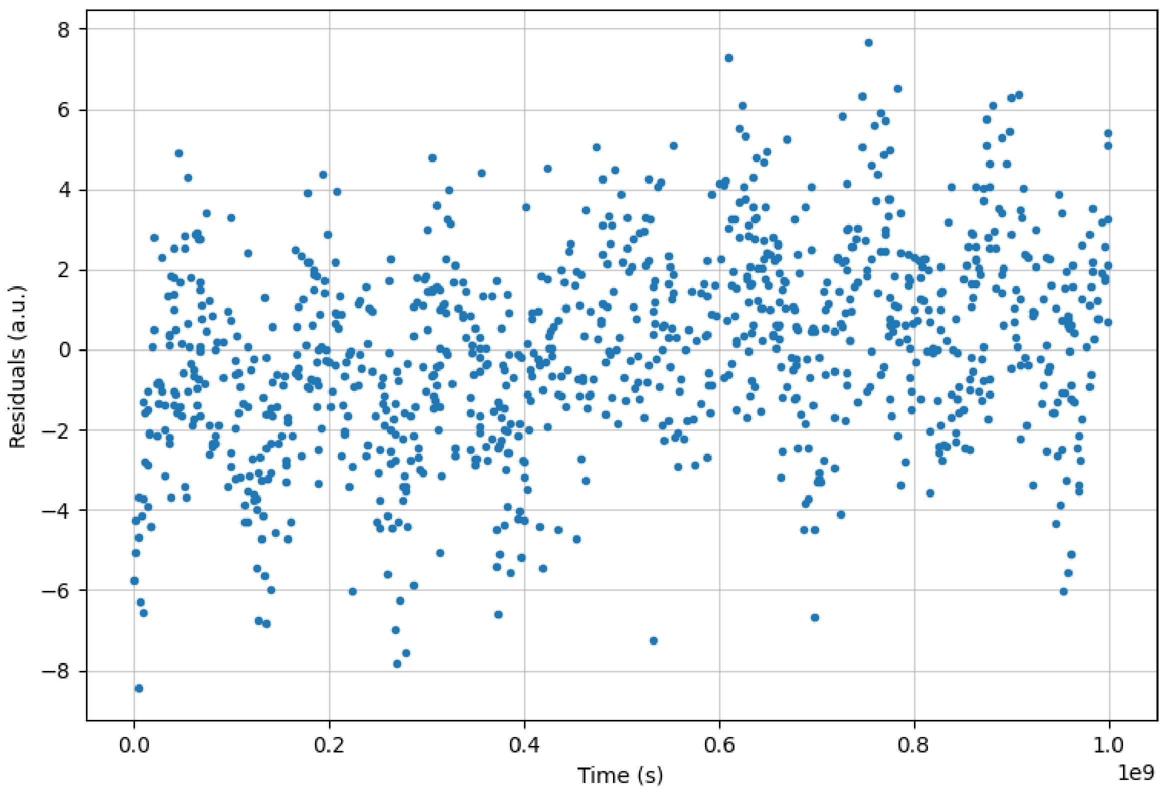

Figure 11: Residuals for the harmonic sequence fit with $n = 10$.

From Figure 10 and Figure 8 we can obtain an estimate for the period of the signal. This is given by the time span covered divided by the number of crests minus 1, which is 8-1 = 7. The obtained period is then

$$T = 1.43 \cdot 10^8 \ s. \tag{11}$$