Spring 2024

Modeling Inflation Using a Fast Fourier Transform (FFT)

Blake Smith
brs146@uakron.edu

Modeling Inflation Using a Fast Fourier Transform (FFT)

Blake R. Smith, The University of Akron, Department of Mathematics

Abstract

This paper utilizes a Fast Fourier Transform (FFT) algorithm to construct a trigonometric interpolant for the Consumer Price Index (CPI), which is then differentiated and used to obtain a continuous function for "instantaneous" (i.e., month-wise) inflation, as opposed to a 12-month percent-change. Fourier coefficients are analyzed to investigate underlying periodicities in the newly constructed function. This metric does not hold significant predictive value but it may prove helpful in retroactive analysis of inflation trends.

Keywords: FFT, inflation, Consumer Price Index, CPI, Fourier, calculus, instantaneous percent change

1 Introduction

In the United States, inflation is a measure of the upward trend in prices of goods and services.[1] Inflation is most commonly calculated using the 12-month percent-change of the Consumer Price Index (CPI), a monthly economic index that captures the overall price of goods and services to consumers. The CPI is calculated by means of a weighted sum of prices of 243 commodities and services across 32 urban regions in the United States, which is then adjusted to account for various other economic factors and considerations.[2] CPI data was first collected in January 1913 and has been collected monthly since then. CPI data can be used for a wide range of applications in economic analysis, but the most common applications are calculating inflation and adjusting other economic series for the change in value of the dollar (e.g., analyzing rent over time in "2024" USD by adjusting the value of USD according to CPI data). This paper takes interest with the former.

Since the CPI is a discrete data set, inflation can be calculated for any given month begininning in January 1914 by the formula

$$Inflation_k = \frac{CPI_k - CPI_{k-12}}{CPI_k} * 100 \quad (1)$$

This method of defining inflation answers the question, "how much more expensive are goods and services this month compared to a year ago?" In typical settings, especially for the average consumer, this measure is very practical. However, there are certain contexts in which it fails to yield important insights. For example, if inflation is 8% one month and 1% the next month, one cannot determine from these numbers alone how the CPI is behaving. Perhaps the change is due to an unusual price change within the past month, or it may be due to an ususual price change 12 months prior. Regardless of the correct answer, the ambiguity can be problematic at best and misleading at worst. Inflation, then, is a poor indicator of *current* trends of the CPI.

To remedy this problem, we set out to use existing CPI data to construct a function representing the "instantaneous" inflation for any given month. This function, then, will represent the expected percent-change from one month to the next. To do this, we choose to employ a Discrete Fourier Transform (DFT) to a set of CPI data. Given a discrete data set with N evenly-spaced points y_n, we can find the discrete Fourier transform by the following summation:

$$Y_k = \frac{1}{N}\sum_{n=0}^{N-1} y_n e^{-\frac{2\pi i}{N}nk} \quad (2)$$

for $k = 0,1,\dots,N-1$. Each Y_k is a complex number known as a Fourier coefficient. Consider Euler's formula

$$re^{i\theta} = r[\cos(\theta) + i\sin(\theta)] \quad . \quad (3)$$

Since each Y_k is calculated by a sum of complex exponentials, Y_k is the sum of sines and cosines that have θ values of every multiple $0,1,\ldots,N-1$ of k. Thus, the magnitude of Y_k can be interpreted as a relative measure of the k-periodicity of the discrete data. If the magnitude of Y_k is relatively high compared to other magnitudes, then there is evidence suggesting cyclic trends with period k.

Algorithms that implement the traditional DFT are very computationally expensive, being on the order of $\sim O(N^2)$ time complexity.* Thus, to compute the Fourier coefficients, we use the standard Fast Fourier Transform (FFT) algorithm, which has time complexity $\sim O(N \log N)$. The Fourier coefficients produced by the two methods are identical, but those produced by the FFT are computed far more efficiently, hence why it is considered the standard method.

Once the Fourier coefficients are obtained, we use the formula from Johnson (2014)[3] to construct the minimally-oscillating trigonometric interpolant. Being an interpolant, this function is equal to the discrete data points at each (x_k, y_k), but has the added property of being continuous between the data itself. Additionally, the interpolant is "minimally-oscillating" between data points, which creates a smoother interpolation of our discrete data. This particular interpolant was derived by minimizing the mean-square slope of a general trigonometric interpolant, the process for which is described in Johnson (2014). The interpolant we are interested in is given by

$$f(x) = Y_0 + \sum_{0<k<N/2} \left(Y_k e^{+\frac{2\pi i}{N} kx} + Y_{N-k} e^{-\frac{2\pi i}{N} kx} \right) + Y_{N/2} \cos(\pi x) \quad , \qquad (4)$$

---

* Time complexity describes the "order" of computations/operations required to perform an action. $O(N^2)$ means that the time taken to perform the algorithm on N points increases as a constant times N^2.

where the $Y_{N/2}$ term is absent if N is odd. This absence results from the symmetry of Fourier coefficients about $N/2$. The summation captures all of the terms where k is strictly less than $N/2$, and all of the terms where k is strictly greater than $N/2$ are redundant. If N is odd, then $Y_{\frac{N-1}{2}} = Y_{\frac{N+1}{2}}$, and there is no $Y_{N/2}$ term ($N/2$ is not an integer). However, if N is even, then the summation only captures everything before and after the $Y_{N/2}$ term, but not $Y_{N/2}$ itself. Thus, we must add the term in cases of even N.

When the data are real-valued, $Y_k = \bar{Y}_{N-k}$, and the formula simplifies due to the cancellation of the imaginary parts of the calculation. Let $Y_k = a_k + ib_k$, $Y_{N-k} = \bar{Y}_k = a_k - ib_k$, and $\theta = \frac{2\pi}{N}$. We apply these substitutions and, after simplification, the above equation becomes

$$f(x) = Y_0 + 2 * \left[\sum_{0<k<\frac{N}{2}} (a_k \cos(\theta k x) - b_k \sin(\theta k x)) \right] + Y_{\frac{N}{2}} \cos(\pi x) \quad . \qquad (5)$$

After we compute $f(x)$ using our Fourier coefficients, we have a sum of sines and cosines with a constant being added. Thus, we can analytically compute $f'(x)$ by termwise differentiation. Now, we have functions $f(x)$ and $f'(x)$, and we define the function

$$P(x) = \frac{f'(x)}{f(x)} \qquad (6)$$

to be our function of interest. $P(x)$ will be our model for "instantaneous" percent-change. Assume that $f'(x_n)$ is a good approximation for $\frac{f(x_{n+1})-f(x_n)}{x_{n+1}-x_n}$, the forward difference approximation for $f'(x_n)$. Then

$$f'(x_n) \approx \frac{f(x_{n+1}) - f(x_n)}{x_{n+1} - x_n} = f(x_{n+1}) - f(x_n) \quad . \qquad (7)$$

This equality is true because the data points are evenly spaced and x is measured in the unit time of 1 month, so $x_{n+1} - x_n = 1$. Then we have

$$P(x_n) = \frac{f'(x_n)}{f(x_n)} \approx \frac{f(x_{n+1}) - f(x_n)}{f(x_n)} \quad . \tag{8}$$

The final expression above is the exact value of the percent-change between two consecutive months. Thus, we can construct the function $P(x)$ which, using our assumption about $f'(x)$, approximates the percent-change between consecutive months. Such a function provides information about the "instantaneous inflation," which more directly measures the movement of CPI measures at a particular point in time.

One can test the validity of assumption above about $f'(x_n)$ by performing the following algorithm on x_n for $n = 0,1, \ldots, N-1$:

1. Compute the derivative of the interpolant, $f'(x)$;
2. Compute and store the values of $y_{n+1} = f(x_n) + f'(x_n)$ for all n;
3. Find the absolute error $|y_k - f(x_k)|$ and/or the relative error (divide the quanity by $|f(x_k)|$) for $k = 1,2, \ldots, N-1$; and
4. Confirm that the errors are within some tolerance ϵ in the interval of interest.

The subset of CPI data used in this work passed the above test with a relative error tolerance of 1%, so we continue with our analysis accepting this approximation as "good."

The last last quantity we want to consider is some measure of what a "good" monthly inflation rate might look like. The Federal Reserve has a target goal of 2% annual inflation for the sake of a healthy economy.[4] Thus, we want to find a similar benchmark for monthly inflation that corresponds to the Federal Reserve's 2% target. In order to match an annual growth of 2%, we must find p such that, for some CPI_k,

$$CPI_{k+12} = 1.02 * CPI_k = (1+p)^{12} CPI_k \quad . \tag{9}$$

Intuitively, we are finding the growth percentage that, when compounded monthly for 12 months, produces the same net growth as an annual simple growth rate of 2%. Solving the above equation for p, we get

$$p = e^{\ln(1.02)/12} - 1 \approx 0.00165 \quad . \tag{10}$$

Thus, for 12 consecutive months to have the same cumulative inflation as the Federal Reserve's target, they must have $P(x_k), P(x_{k+1}), \dots, P(x_{k+11})$ such that

$$p = \sqrt[12]{|P(x_k)P(x_{k+1}) \dots P(x_{k+11})|} \quad . \tag{11}$$

In other words, if the geometric mean of $P(x_i)$ for 12 consecutive months is greater than p, then inflation is higher than the "healthly" amount.

2 Data and Methods

The CPI dataset was obtained from the U.S. Bureau of Labor Statistics website.[5] The table was exported to a spreadsheet (see Appendix A), and the data was formatted into a single vector in column Q. Note that prior to 2007, only one decimal place of precision was used, which was expanded to three decimal places as of January 2007. There are 1328 total data points, 1128 data points with ≤ 4 significant figures, and 200 data points with 6 significant figures. Due to less decimal precision, many early CPI values do not vary from month-to-month, which negatively affects the quality of the analysis. Therefore, for our analysis, we opt to use only the data with 6 significant figures.

A MATLAB (version R2023a) algorithm was written which peforms the following actions:

1. Imports the CPI data into a vector variable;
2. Computes Fourier coefficients for $f(x_n)$ using matlab protocol `fft`;
3. Computes the minimally-oscillating trigonometric interpolant $f(x)$;
4. Differentiates the trigonometric interpolant $f'(x)$;

5. Defines and computes the "instantaneous percent-change" function $P(x)$; and
6. Computes the Fourier coefficients of $P(x_n)$.

Note that, for the purpose of this analysis, one unit of time is equal to one month. The program also returns plots and graphs of the data that are utilized in this paper. The full algorithm is contained in the MATLAB file CPI_Model.mlx (see Appendix B).

When plotting and analyzing the results, we must omit a total of ~20% of the data (the first and last 10%) due to extreme oscillations near the endpoints of the interval. This inaccuracy is a common problem with non-piecewise interpolants, such as the one used in this paper, with large N (in our case, N = 200). Figure 1a and 1b demonstrate this problem with a LaGrange polynomial interpolant for 50 points sampled from the line $y = x + \sin(2\pi x)$ on $[0,2]$. Note the relatively extreme error near the endpoints in Figure 1b compared to everywhere else on the interval.

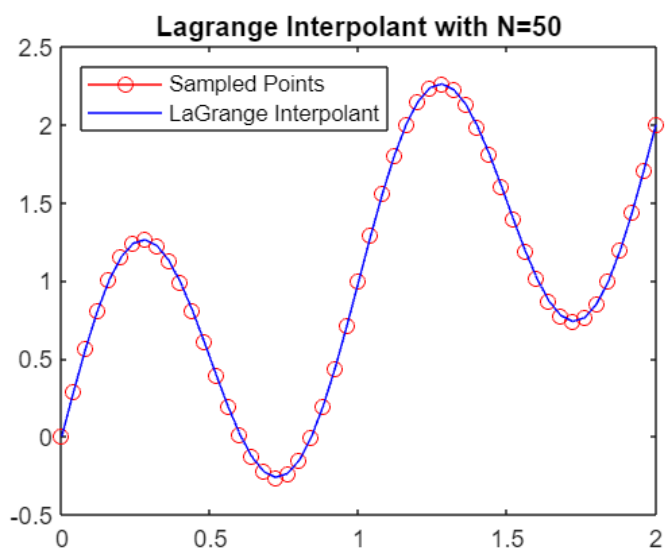

Figure 1a. LaGrange polynomial interpolant for $f(x) = x + \sin(x)$.

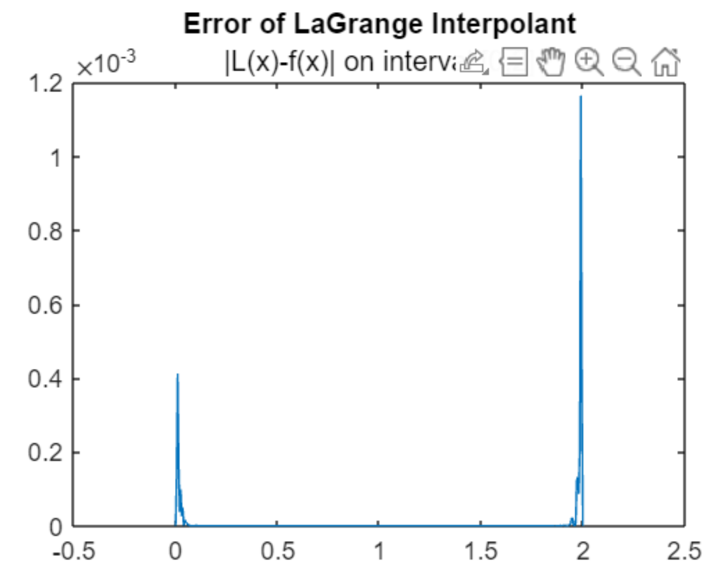

Figure 1b. Error plot of the degree 50 LaGrange polynomial compared to $f(x)$

3 Results

For this analysis, we utilize the CPI data from January 2007 to August 2023 (N = 200), visualized below in Figure 2.

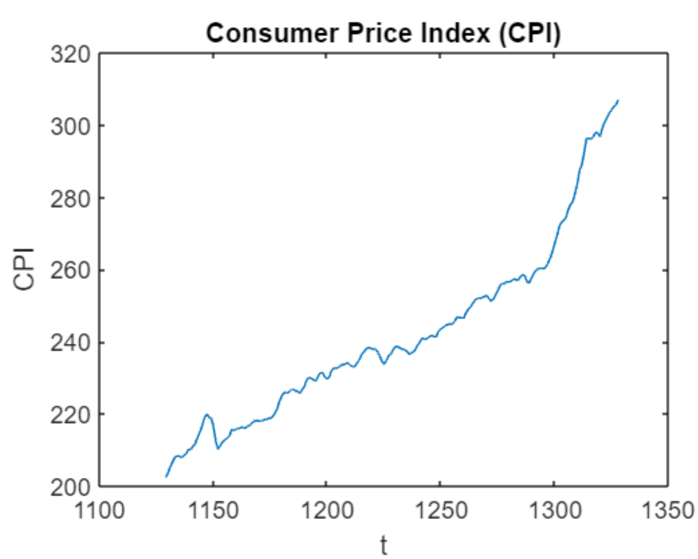

Figure 2. Plot of the CPI data from January 2007 to August 2023, corresponding to $t = 1129$ to $t = 1328$

An FFT was used on this data to acquire Fourier Coefficients Y_k. Equation (4) was then implemented and used to construct the unique minimally oscillating trigonometric interpolant, $f(x)$ (see Figures 3a and 3b).

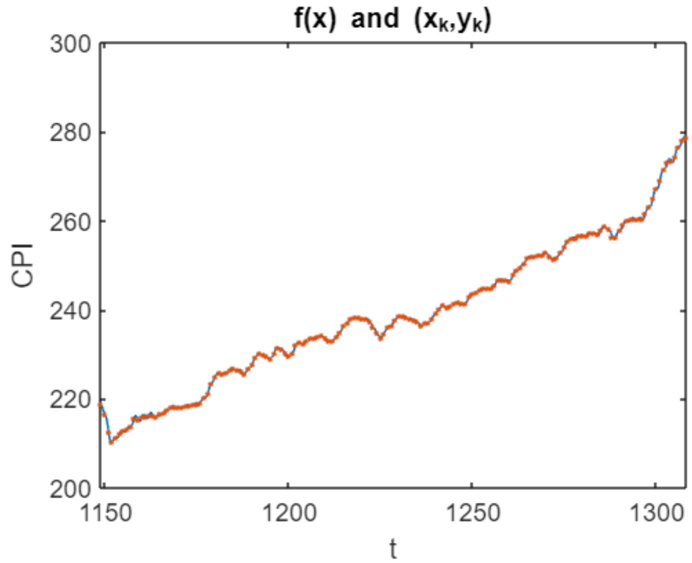

Figure 3a. Graph of $f(x)$ in blue, with plotted CPI data in orange. Note that we must omit the highest and lowest 10% of the data interval due to high oscillatory behavior near the endpoints.

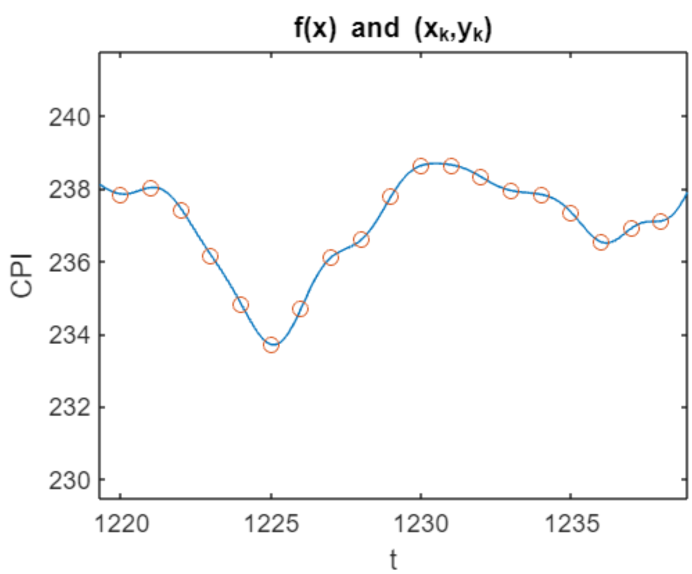

Figure 3b. The graph from Figure 3a, zoomed in to demonstrate interpolation.

Because $f(x)$ is just the sum of sines and cosines, we then analytically differentiate it to acquire the continuous derivative function, $f'(x)$ (see Figure 4).

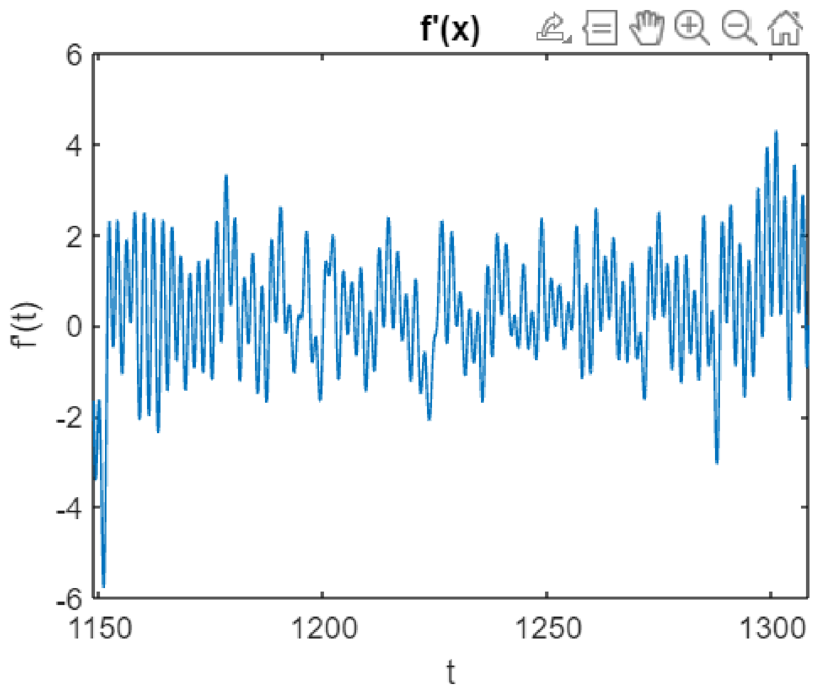

Figure 4. Graph of $f'(x)$ on the same subinterval as Figure 2.

Now that we have continuous expressions for $f(x)$ and $f'(x)$, we define the function $P(x) = \frac{f'(x)}{f(x)}$ to be the so-called "instantaneous" percent-change of f. Figure 5 shows the results for $P(x)$. Also pictured is the ideal geometric mean p required to achieve the Federal Reserve's target annual inflation of 2%.

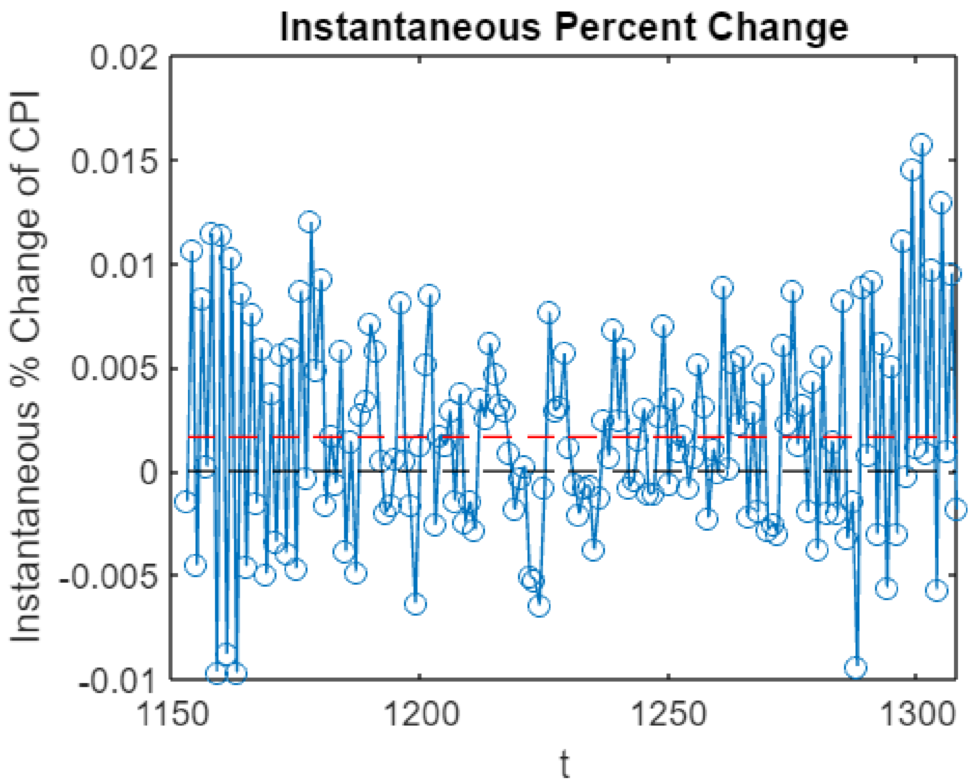

Figure 5. $P(x)$ graphed in blue, with growth threshold p from Eqn. (10) in red and 0 in black.

Finally, we apply an FFT to $P(x)$ above and analyze the magnitude of the Fourier coefficients Y_k (see Figure 6). In doing so, we identify major periodic attributes of $P(x)$. Namely, we see minor spikes at 13, 26, and 39 months. These values roughly correspond to cycles of one, two, and three years, respectively. Strangely, we see the largest major spike at 80 months (6.7 years), which would not have been predicted merely from conventional knowledge of inflation trends.

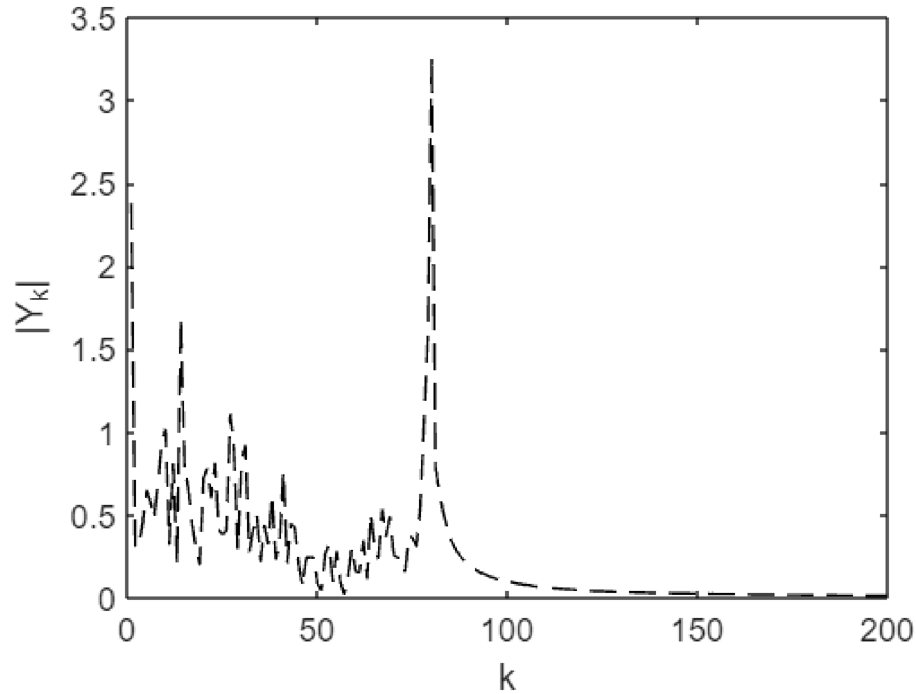

Figure 6. Plot comparing the magnitudes of the Fourier coefficients for $P(x)$.

4 Discussion and Applications

This paper used an FFT to generate Fourier coefficients for a certain series of CPI data. These coefficients were used to construct a trigonometric interpolant $f(x)$, which was then differentiated to acquire $f'(x)$. We then defined the function $P(x) = \frac{f'(x)}{f(x)}$ to be a continuosu function representing the instantaneous percent-change of the CPI and demonstrated that $P(x_n)$ is a fair approximation for the monthly percent-change at x_n. Finally, another FFT was performed on $P(x)$ to analyze any significant periods for $P(x)$. This final analysis yielded evidence that monthly inflation is at least somewhat cyclical, with significant periods of approximately one, two, and three years. Interestingly, that same analysis also suggests that there is a significant period of 80 months, which corresponds to approximately 6.7 years. While the yearly periods are intuitive and somewhat expected from the analysis, no conventional knowledge about inflation can reasonably explain the 80-month trend that. It is quite possible that this is merely a result of some real-world event or phenomenon relating to the time period chosen, being January 2007 to August 2023.

A major drawback of this method is that it holds virtually no predictive value. This is due to the choice of a non-piecewise interpolant. This type of interpolant tends to lose accuracy near the endpoints of the data set, as demonstrated in Figures 1a and 1b. In particular, the trigonometric interpolant used exacerbates this issue with its inherently oscillatory behavior. However, it is important to note that our definition of the monthly inflation function $P(x)$ did not depend on the choice of interpolant. Therefore, this model could be reconstructed with a different choice of interpolant, like a cubic spline, which minimizes the problem of inaccuracy near the endpoints of the data set. Another limitation of the method is that the concept of instantaneous inflation is not as practical in everyday life as the current convention for the metric, which is the 12-month percent change. However, in applications where it is important to know the source of any extreme behavior of inflation, it may prove to be more useful than manual analysis.

Finally, observe that we have developed a function $P(x)$ that represents the instantaneous percent-change of a function $f(x)$. Furthermore, given a set of discretely-sampled data, an interpolant can be computed and differentiated that then allows $P(x)$ to be constructed. Thus, this method can be applied to any set of sufficiently-often sampled data (e.g., Gross Domestic Product, stock exchanges, etc.), not just the CPI series used in this work. The primary obstacle to applying this method in other contexts, however, is the assumption that $f(x_{n+1}) \approx f(x_n) + f'(x_n)$. For sets of data where consecutive values in the series often fluctuate by extreme amounts, the assumption above may not necessarily be true, which diminishes the validity of $P(x)$. One can test the validity of this assumption above by performing the test described at the end of the introduction. For certain interpolants and, it may also be appropriate to apply some kind of smoothing to the $f'(x)$ function which, in turn, might normalize the behavior of $P(x)$.

References:

[1] U.S. Department of Labor. (2024, April 15). *Inflation and consumer information.* https://www.dol.gov/general/topic/statistics/inflation#:~:text=Inflation%20can%20be%20defined%20as%20the%20overall%20general,various%20indexes%20that%20measure%20different%20aspects%20of%20inflation.

[2] U.S. Bureau of Labor Statistics. (2023, September 6). *Consumer Price Index: Calculation.* https://www.bls.gov/opub/hom/cpi/calculation.htm#final-c-cpi-u

[3] Johnson, S. G. (2011, May 4). *Notes on FFT-based differentiation.* Massachusetts Institute of Technology, Applied Mathematics.

[4] Board of Governors of the Federal Reserve System. (2020, August 27). *Why does the Federal Reserve aim for inflation of 2 percent over the longer run?*

[5] U.S. Bureau of Labor Statistics. (2024, April 15). *Consumer Price Index for all urban consumers (CPI-U).* https://data.bls.gov/pdq/SurveyOutputServlet

Appendix A

CPI-U Dataset Utilized

9.8	9.8	9.8	9.8	9.7	9.8	9.9	9.9	10.0	10.0	10.1	10.0
10.0	9.9	9.9	9.8	9.9	9.9	10.0	10.2	10.2	10.1	10.2	10.1
10.1	10.0	9.9	10.0	10.1	10.1	10.1	10.1	10.1	10.2	10.3	10.3
10.4	10.4	10.5	10.6	10.7	10.8	10.8	10.9	11.1	11.3	11.5	11.6
11.7	12.0	12.0	12.6	12.8	13.0	12.8	13.0	13.3	13.5	13.5	13.7
14.0	14.1	14.0	14.2	14.5	14.7	15.1	15.4	15.7	16.0	16.3	16.5
16.5	16.2	16.4	16.7	16.9	16.9	17.4	17.7	17.8	18.1	18.5	18.9
19.3	19.5	19.7	20.3	20.6	20.9	20.8	20.3	20.0	19.9	19.8	19.4
19.0	18.4	18.3	18.1	17.7	17.6	17.7	17.7	17.5	17.5	17.4	17.3
16.9	16.9	16.7	16.7	16.7	16.7	16.8	16.6	16.6	16.7	16.8	16.9
16.8	16.8	16.8	16.9	16.9	17.0	17.2	17.1	17.2	17.3	17.3	17.3
17.3	17.2	17.1	17.0	17.0	17.0	17.1	17.0	17.1	17.2	17.2	17.3
17.3	17.2	17.3	17.2	17.3	17.5	17.7	17.7	17.7	17.7	18.0	17.9
17.9	17.9	17.8	17.9	17.8	17.7	17.5	17.4	17.5	17.6	17.7	17.7
17.5	17.4	17.3	17.3	17.4	17.6	17.3	17.2	17.3	17.4	17.3	17.3
17.3	17.1	17.1	17.1	17.2	17.1	17.1	17.1	17.3	17.2	17.2	17.1
17.1	17.1	17.0	16.9	17.0	17.1	17.3	17.3	17.3	17.3	17.3	17.2
17.1	17.0	16.9	17.0	16.9	16.8	16.6	16.5	16.6	16.5	16.4	16.1
15.9	15.7	15.6	15.5	15.3	15.1	15.1	15.1	15.0	14.9	14.7	14.6
14.3	14.1	14.0	13.9	13.7	13.6	13.6	13.5	13.4	13.3	13.2	13.1
12.9	12.7	12.6	12.6	12.6	12.7	13.1	13.2	13.2	13.2	13.2	13.2
13.2	13.3	13.3	13.3	13.3	13.4	13.4	13.4	13.6	13.5	13.5	13.4
13.6	13.7	13.7	13.8	13.8	13.7	13.7	13.7	13.7	13.7	13.8	13.8
13.8	13.8	13.7	13.7	13.7	13.8	13.9	14.0	14.0	14.0	14.0	14.0
14.1	14.1	14.2	14.3	14.4	14.4	14.5	14.5	14.6	14.6	14.5	14.4
14.2	14.1	14.1	14.2	14.1	14.1	14.1	14.1	14.1	14.0	14.0	14.0
14.0	13.9	13.9	13.8	13.8	13.8	13.8	13.8	14.1	14.0	14.0	14.0
13.9	14.0	14.0	14.0	14.0	14.1	14.0	14.0	14.0	14.0	14.0	14.1
14.1	14.1	14.2	14.3	14.4	14.7	14.7	14.9	15.1	15.3	15.4	15.5
15.7	15.8	16.0	16.1	16.3	16.3	16.4	16.5	16.5	16.7	16.8	16.9
16.9	16.9	17.2	17.4	17.5	17.5	17.4	17.3	17.4	17.4	17.4	17.4
17.4	17.4	17.4	17.5	17.5	17.6	17.7	17.7	17.7	17.7	17.7	17.8
17.8	17.8	17.8	17.8	17.9	18.1	18.1	18.1	18.1	18.1	18.1	18.2
18.2	18.1	18.3	18.4	18.5	18.7	19.8	20.2	20.4	20.8	21.3	21.5
21.5	21.5	21.9	21.9	21.9	22.0	22.2	22.5	23.0	23.0	23.1	23.4
23.7	23.5	23.4	23.8	23.9	24.1	24.4	24.5	24.5	24.4	24.2	24.1
24.0	23.8	23.8	23.9	23.8	23.9	23.7	23.8	23.9	23.7	23.8	23.6
23.5	23.5	23.6	23.6	23.7	23.8	24.1	24.3	24.4	24.6	24.7	25.0
25.4	25.7	25.8	25.8	25.9	25.9	25.9	25.9	26.1	26.2	26.4	26.5
26.5	26.3	26.3	26.4	26.4	26.5	26.7	26.7	26.7	26.7	26.7	26.7
26.6	26.5	26.6	26.6	26.7	26.8	26.8	26.9	26.9	27.0	26.9	26.9
26.9	26.9	26.9	26.8	26.9	26.9	26.9	26.9	26.8	26.8	26.8	26.7
26.7	26.7	26.7	26.7	26.7	26.7	26.8	26.8	26.9	26.9	26.9	26.8
26.8	26.8	26.8	26.9	27.0	27.2	27.4	27.3	27.4	27.5	27.5	27.6
27.6	27.7	27.8	27.9	28.0	28.1	28.3	28.3	28.3	28.3	28.4	28.4
28.6	28.6	28.8	28.9	28.9	28.9	29.0	28.9	28.9	28.9	29.0	28.9
29.0	28.9	28.9	29.0	29.0	29.1	29.2	29.2	29.3	29.4	29.4	29.4
29.3	29.4	29.4	29.5	29.5	29.6	29.6	29.6	29.6	29.8	29.8	29.8
29.8	29.8	29.8	29.8	29.8	29.8	30.0	29.9	30.0	30.0	30.0	30.0
30.0	30.1	30.1	30.2	30.2	30.2	30.3	30.3	30.4	30.4	30.4	30.4
30.4	30.4	30.5	30.5	30.5	30.6	30.7	30.7	30.7	30.8	30.8	30.9
30.9	30.9	30.9	30.9	30.9	31.0	31.1	31.0	31.1	31.1	31.2	31.2
31.2	31.2	31.3	31.4	31.4	31.6	31.6	31.6	31.6	31.7	31.7	31.8
31.8	32.0	32.1	32.3	32.3	32.4	32.5	32.7	32.7	32.9	32.9	32.9
32.9	32.9	33.0	33.1	33.2	33.3	33.4	33.5	33.6	33.7	33.8	33.9
34.1	34.2	34.3	34.4	34.5	34.7	34.9	35.0	35.1	35.3	35.4	35.5
35.6	35.8	36.1	36.3	36.4	36.6	36.8	37.0	37.1	37.3	37.5	37.7
37.8	38.0	38.2	38.5	38.6	38.8	39.0	39.0	39.2	39.4	39.6	39.8
39.8	39.9	40.0	40.1	40.3	40.6	40.7	40.8	40.8	40.9	40.9	41.1
41.1	41.3	41.4	41.5	41.6	41.7	41.9	42.0	42.1	42.3	42.4	42.5
42.6	42.9	43.3	43.6	43.9	44.2	44.3	45.1	45.2	45.6	45.9	46.2
46.6	47.2	47.8	48.0	48.6	49.0	49.4	50.0	50.6	51.1	51.5	51.9
52.1	52.5	52.7	52.9	53.2	53.6	54.2	54.3	54.6	54.9	55.3	55.5
55.6	55.8	55.9	56.1	56.5	56.8	57.1	57.4	57.6	57.9	58.0	58.2
58.5	59.1	59.5	60.0	60.3	60.7	61.0	61.2	61.4	61.6	61.9	62.1
62.5	62.9	63.4	63.9	64.5	65.2	65.7	66.0	66.5	67.1	67.4	67.7
68.3	69.1	69.8	70.6	71.5	72.3	73.1	73.8	74.6	75.2	75.9	76.7
77.8	78.9	80.1	81.0	81.8	82.7	82.7	83.3	84.0	84.8	85.5	86.3
87.0	87.9	88.5	89.1	89.8	90.6	91.6	92.3	93.2	93.4	93.7	94.0
94.3	94.6	94.5	94.9	95.8	97.0	97.5	97.7	97.9	98.2	98.0	97.6
97.8	97.9	97.9	98.6	99.2	99.5	99.9	100.2	100.7	101.0	101.2	101.3
101.9	102.4	102.6	103.1	103.4	103.7	104.1	104.5	105.0	105.3	105.3	105.3
105.5	106.0	106.4	106.9	107.3	107.6	107.8	108.0	108.3	108.7	109.0	109.3
109.6	109.3	108.8	108.6	108.9	109.5	109.5	109.7	110.2	110.3	110.4	110.5
111.2	111.6	112.1	112.7	113.1	113.5	113.8	114.4	115.0	115.3	115.4	115.4
115.7	116.0	116.5	117.1	117.5	118.0	118.5	119.0	119.8	120.2	120.3	120.5
121.1	121.6	122.3	123.1	123.8	124.1	124.4	124.6	125.0	125.6	125.9	126.1
127.4	128.0	128.7	128.9	129.2	129.9	130.4	131.6	132.7	133.5	133.8	133.8
134.6	134.8	135.0	135.2	135.6	136.0	136.2	136.6	137.2	137.4	137.8	137.9
138.1	138.6	139.3	139.5	139.7	140.2	140.5	140.9	141.3	141.8	142.0	141.9
142.6	143.1	143.6	144.0	144.2	144.4	144.4	144.8	145.1	145.7	145.8	145.8
146.2	146.7	147.2	147.4	147.5	148.0	148.4	149.0	149.4	149.5	149.7	149.7
150.3	150.9	151.4	151.9	152.2	152.5	152.5	152.9	153.2	153.7	153.6	153.5
154.4	154.9	155.7	156.3	156.6	156.7	157.0	157.3	157.8	158.3	158.6	158.6
159.1	159.6	160.0	160.2	160.1	160.3	160.5	160.8	161.2	161.6	161.5	161.3
161.6	161.9	162.2	162.5	162.8	163.0	163.2	163.4	163.6	164.0	164.0	163.9
164.3	164.5	165.0	166.2	166.2	166.2	166.7	167.1	167.9	168.2	168.3	168.3
168.8	169.8	171.2	171.3	171.5	172.4	172.8	172.8	173.7	174.0	174.1	174.0
175.1	175.8	176.2	176.9	177.7	178.0	177.5	177.5	178.3	177.7	177.4	176.7
177.1	177.8	178.8	179.8	179.8	179.9	180.1	180.7	181.0	181.3	181.3	180.9
181.7	183.1	184.2	183.8	183.5	183.7	183.9	184.6	185.2	185.0	184.5	184.3
185.2	186.2	187.4	188.0	189.1	189.7	189.4	189.5	189.9	190.9	191.0	190.3
190.7	191.8	193.3	194.6	194.4	194.5	195.4	196.4	198.8	199.2	197.6	196.8
198.3	198.7	199.8	201.5	202.5	202.9	203.5	203.9	202.9	201.8	201.5	201.8
202.416	203.499	205.352	206.686	207.949	208.352	208.299	207.917	208.490	208.936	210.177	210.036
211.080	211.693	213.528	214.823	216.632	218.815	219.964	219.086	218.783	216.573	212.425	210.228
211.143	212.193	212.709	213.240	213.856	215.693	215.351	215.834	215.969	216.177	216.330	215.949
216.687	216.741	217.631	218.009	218.178	217.965	218.011	218.312	218.439	218.711	218.803	219.179
220.223	221.309	223.467	224.906	225.964	225.722	225.922	226.545	226.889	226.421	226.230	225.672
226.665	227.663	229.392	230.085	229.815	229.478	229.104	230.379	231.407	231.317	230.221	229.601
230.280	232.166	232.773	232.531	232.945	233.504	233.596	233.877	234.149	233.546	233.069	233.049
233.916	234.781	236.293	237.072	237.900	238.343	238.250	237.852	238.031	237.433	236.151	234.812
233.707	234.722	236.119	236.599	237.805	238.638	238.654	238.316	237.945	237.838	237.336	236.525
236.916	237.111	238.132	239.261	240.229	241.018	240.628	240.849	241.428	241.729	241.353	241.432
242.839	243.603	243.801	244.524	244.733	244.955	244.786	245.519	246.819	246.663	246.669	246.524
247.867	248.991	249.554	250.546	251.588	251.989	252.006	252.146	252.439	252.885	252.038	251.233
251.712	252.776	254.202	255.548	256.092	256.143	256.571	256.558	256.759	257.346	257.208	256.974
257.971	258.678	258.115	256.389	256.394	257.797	259.101	259.918	260.280	260.388	260.229	260.474
261.582	263.014	264.877	267.054	269.195	271.696	273.003	273.567	274.310	276.589	277.948	278.802
281.148	283.716	287.504	289.109	292.296	296.311	296.276	296.171	296.808	298.012	297.711	296.797
299.170	300.840	301.836	303.363	304.127	305.109	305.691	307.026				

Appendix B

CPI_Model.mlx

```
% Student:    Blake R. Smith              brs146@uakron.edu
% Advisor:    Dr. Joseph Wilder           wilder@uakron.edu
%------------------------------------------------------------
% Modeling Inflation Using a Fast Fourier Transform (FFT)
% Senior Honors Research Project
% Spring 2024
% Department of Mathematics
% The University of Akron
%------------------------------------------------------------
clear all;
```

Section 1: CPI Data Importing

```
% Reads in CPI Data from an Excel Sheet
    CPI = readmatrix("CPI_Datasheet.xlsx","Range","Q2:Q1329");


% There are 1328 data entries (monthly, from Jan 1913 to Aug 2023)
% t is just the plotting window, does not affect calculations


% Good data (3 decimal precision) begins at index 1129; everything prior is
% single digit decimal precision


winstart = 1129;
winend = 1328;
t = winstart:winend;


% Plot to demonstrate CPI data
    figure(1)
    plot(t,CPI(t));
    title("Consumer Price Index (CPI)")
    ylabel("CPI")
    xlabel("t")
```

Section 2: Trigonometric Interpolant, $f(x)$

```
% Use the function in Section 6 to calculate the interpolant
[CPI_interp, Y] = trig_interp(CPI(t),t(1));


% Grid Plotting
    h = 0.1;
    plotgrid = winstart:h:winend;
```

```
    perc = round(0.1*length(plotgrid));
    m = plotgrid(perc:end-perc);


    plot(m,CPI_interp(m))
    hold on
    plot(t,CPI(t),'.')
    hold off
    axis([m(1) m(end) 200 300])
    title("f(x) and (x_k,y_k)")
    xlabel("t")
    ylabel("CPI")
    % % Note: lots of error within 10% distance of endpoints
```

Section 3: First Derivative, $f'(x)$

```
% Use the Y values from Section 2 and function from
% Section 6 to calculate f'(x)
CPI_deriv = trig_deriv(Y,t(1));
    plot(m, CPI_deriv(m))
    title("f'(x)")
    xlabel("t")
    ylabel("f'(t)")
```

Section 4: Percent-change function, P(x)

```
perc_change = @(x) CPI_deriv(x)./CPI_interp(x);


hold off
plot(t(25:end-20),perc_change(t(25:end-20)),'-o')
interv = length(t(25:end-20));
hold on
plot(t(25:end-20),0.00165*ones(interv,1),'r--')
plot(t(25:end-20),zeros(interv,1),'k--')
hold off;
title("Instantaneous Percent Change")
xlabel("t")
ylabel("Instantaneous % Change of CPI")
```

Section 5: FFT of P(x) function

```
K = fft(perc_change(t(25:end-20)));
plot(0:length(K)-1,abs(K),'k--')
axis([0 78 0 0.3])
xlabel("k")
```

```
ylabel("|Y_k|")
```

Secton 6: Function Definitions

```
function [f, Y] = trig_interp(y,x0)
% function f = trig_interp(y,x0)
% INPUT:
% y --> column vector of sample points
% x0 --> the x position of the first point
% OUTPUT:
% f --> minimally-oscillating trigonometric interpolant
% Y --> column vector of Fourier coefficients
%
% This function assumes that each y_(k+1) = y(t_k + 1)
% i.e., evenly spaced points sampled at each unit of time
    N = length(y);

    Y = fft(y)/(N);

    % Determine if N is odd or even, which decides whether to include the Nyquist
Term
    if mod(N,2) == 0
        isodd = 0;
        stopsum = (N/2);
    else
        isodd = 1;
        stopsum = (N+1) / 2;
    end

    % First term of the interpolant is just Y(1)
    f = @(x) real(Y(1));

    % The k'th term of the interpolant is 2(ac - bd)
    % a = Re(Y(k)), b = Im(Y(k)), c = cos(2*pi*i*(k-1)), d = sin(2*pi*i*(k-1))
    for k=2:stopsum
        a = real(Y(k));
        b = imag(Y(k));

        % Term multiplying x in the exponential
        M = 2*pi*(k-1)/N;

        % Building f via termwise iteration
        f = @(x) f(x) + 2*(a*cos(M*(x-x0)) - b*sin(M*(x-x0)));
    end

    % Nyquist Term (N/2) is only present if N is even
    if isodd == 0
        f = @(x) f(x) + Y(stopsum + 1)*cos(pi*(x-x0));
```

```
    end
end


function g = trig_deriv(Y,x0)
% function f = trig_deriv(Y,x0)
% INPUT:
% Y --> column vector of Fourier coefficients
% x0 --> the x position of the first point
% OUTPUT:
% g --> f'(x)
% Differentiation is done directly, modeling the format of the trig_interp
% function
% g = f'(x) = sum(-a*M*sin(M*x) - b*M*cos(M*x)) for 2<= k<= stopsum
    N = length(Y);
    % Determine if N is odd or even, which detects a Nyquist Term
    if mod(N,2) == 0
        stopsum = (N/2);
    else
        stopsum = (N+1) / 2;
    end

    % Initialize g(x)
    g = @(x) 0;


    for k=2:stopsum
        a = real(Y(k));
        b = imag(Y(k));

        % Term multiplying x in the exponential
        M = 2*pi*(k-1)/N;

        % Building f via termwise iteration
        % Shifted by x0
        g = @(x) g(x) + -a*M*sin(M*(x-x0)) - b*M*cos(M*(x-x0));
    end
    g = @(x) g(x)*2;
end
```