Project 2: Approximation by Trigonometric Polynomials (Due Date 14/10/2014)

Introduction: The name of Fourier is associated with the investigation of periodic phenomenon, Fourier series, Fourier Transforms, Discrete Fourier Transforms and Fast Fourier Transforms. These mathematical objects are fundamental to the extraction of periodic signals from noise and form the basis of the fields of Signal and Image Processing. The study and understanding of these objects is important to modern scientific computing.

Fourier Series: Let f be a given real valued function defined on the interval $[0, L]$ and suppose that f has a periodic component. How can both the period and magnitude of this component be determined? To investigate this periodicity, approximate the function by a trigonometric polynomial, $S(t)$ and look for the frequency having the largest coefficient. That is:

$$f(t) \simeq S(t) = c_0 + \sum_{n=1}^{M}[c_n \cos(\tfrac{2\pi n}{L}t) + d_n \sin(\tfrac{2\pi n}{L}t)]$$

or in complex exponential form:

$$f(t) \simeq S(t) = \sum_{n=-M}^{n=M} A_n \exp(\tfrac{i2\pi n}{L}t)$$

The size of A_n and A_{-n} or c_n and d_n gives the contribution of the frequency $2\pi n/L$ to the representation of f by the trigonometric approximation. Note that the trigonometric and exponential forms are equivalent. The following formulas define the transformation between the two (for real valued functions f) :

$$A_0 = c_0, A_n = (c_n - id_n)/2, A_{-n} = (c_n + id_n)/2, \text{ or } c_n = 2\,\mathrm{real}(A_n), d_n = -2\,\mathrm{Im}(A_n)$$

Note that A_n is the complex conjugate of A_{-n}. This approximation is determined by the coefficients of the trigonometric polynomial. The least squares condition will be used to select these coefficients. That is, the coefficients are determined by minimizing the integral of the square of the difference between the approximation and the function over the interval $[0, L]$. The function to be minimized is given by:

$$(c_0, c_1, \ldots, c_M, d_1, \ldots, d_M) = \int_0^1 (f(t) - S(t))^2 dt$$

Approximation Theorem: The coefficients of the Least Squares Approximation to the function f defined on the interval $[0, L]$ by trigonometric polynomials $S(t)$ are unique and are given by:

$$c_0 = \tfrac{1}{L}\int_0^L f(t)dt \quad c_n = \tfrac{2}{L}\int_0^L f(t)\cos(\tfrac{2\pi n}{L}t)dt \quad d_n = \tfrac{2}{L}\int_0^L f(t)\sin(\tfrac{2\pi n}{L}t)dt$$

or

$$A_n = \tfrac{1}{L}\int_0^L f(t)\exp(-\tfrac{2\pi n}{L}t)dt \quad \text{for } -M \le n \le M.$$

Proof: Only one case will be considered, that of the formula for d_k. The other cases follow from a similar calculation. To minimize the function E, compute the various

partial derivatives and set them equal to zero. Solve for the values of the coefficients which satisfies these equations and thus, minimize the function E.

$$\frac{\partial E}{\partial d_k} = 2\int_0^L (f(t) - S(t)) \sin((\tfrac{2\pi n}{L}t)dt = 0$$

or

$$\int_0^L f(t) \sin((\tfrac{2\pi n}{L}t)dt = \int_0^L S(t) \sin((\tfrac{2\pi n}{L}t)dt = \tfrac{L}{2}d_k$$

The last equality comes from using the orthogonality of the sine and cosine functions on the interval [0, L]. To verify uniqueness, let S be any M-term trigonometric polynomial

and let S_f be the M-term Fourier Approximation. Consider the mean square error for the function S.

$$(c_0, c_1, \ldots, c_M, d_1, \ldots, d_M) = \int_0^1 (f(t) - S(t))^2 dt = \int_0^1 (f(t) - S_f(t) + S_f(t) - S(t))^2 dt = \int_0^1 \{(f(t) - S_f(t))^2$$

Assume that f is represented by its Fourier series, then in the last term the quantity $(f(t) - S_f(t))$ contains only frequencies higher than $2\pi M/L$ and the term $(S_f(t) - S(t))$ contains only frequencies up through $2\pi M/L$. Orthogonality of sine and cosine functions on the interval [0, L] implies that all of these integrals are zero. Hence,

$$(c_0, c_1, \ldots, c_M, d_1, \ldots, d_M) = \int_0^1 (f(t) - S(t))^2 dt = \int_0^1 \{(f(t) - S_f(t))^2 + (S_f(t) - S(t))^2\} dt$$

Thus, the Least Squares Condition is satisfied by $S(t) = S_f(t)$. QED

The Least Squares Approximation to f by trigonometric polynomials on the interval

$[0, L]$ is just the truncated Fourier Series or the trigonometric polynomial $S(t)$ defined by the Approximation Theorem. This approximation property is true for each value of M or for the Fourier Series truncated at the Mth term. Note that c_0 is just the average value of f on the interval $[0, L]$.

Exercise #1: Consider the function given by $f(t) = 6\sin(\tfrac{2\pi 5}{L}t)$ already in the form of a trigonometric polynomial, compute the coefficients of the least squares trigonometric polynomial in any interval $[0, L]$.

Exercise #2: Suppose

$$f(t) = 4\cos(\tfrac{2\pi 5}{L}t + \varphi)$$

Using the trigonometric identity

$$\cos(a + b) = \cos(a)\cos(b) - \sin(a)\sin(b)$$

we obtain:

$$f(t) = 4\cos(\tfrac{2\pi 5}{L}t)\cos(\varphi) - 4\sin(\tfrac{2\pi 5}{L}t)\sin(\varphi)$$

Thus, Fourier series terms of the same frequency may be represented in two ways, as a linear combination of cosines and sines or as the cosine of the given frequency with a phase shift φ. Like Exercise #1, using the trigonometric identity, f is in the form of a trigonometric polynomial. Find the coefficients.

Exercise #3: This example considers the case when the frequency of f is not a harmonic (integer multiple) of the fundamental frequency of the trigonometric polynomial. Assume that $f(t) = 3cos(15t)$. Using the Symbolic library sympy of python find the Fourier Coefficients. Plot the function f and its Fourier approximation.

The approximating trigonometric polynomial is defined for all real numbers and not just those in the initial interval $[0, L]$. Since the sine and cosine functions in this polynomial are all periodic on $[0, L]$, the trigonometric polynomial defines an approximation to the periodic extension of the original function f beyond the initial interval $[0, L]$. Note that the periodic extension of the original function may not coincide with the original function outside of the interval $[0, L]$. The following Figure graphs both f and the approximation to f on a larger interval $[0, 3]$.
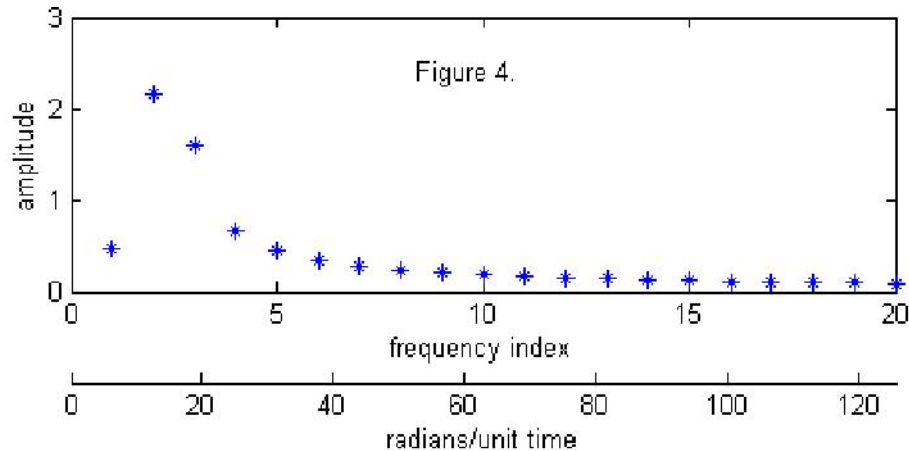

Figure 2.
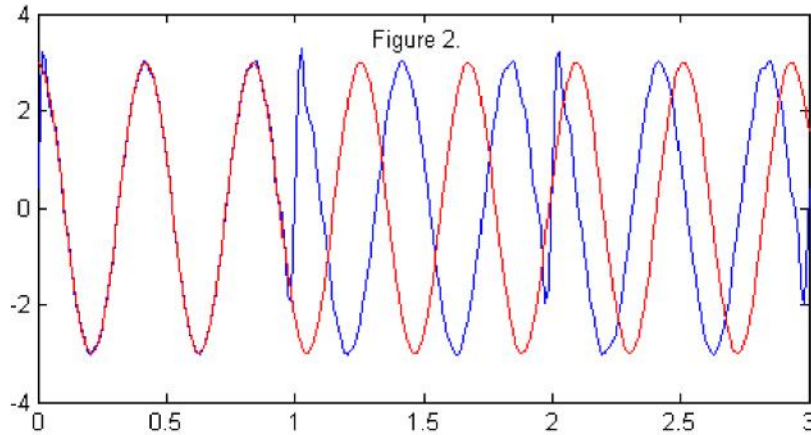
Remark

Note that the approximation and the original function f do not agree outside of the initial interval $[0, 1]$. The frequency of the function f is 15 radians/unit time which lies between two of the harmonics of the fundamental frequency of the trigonometric polynomial $2\pi 2 = 12.$
$566 < 15 < 2\pi 3 = 18.850$

Figure 3 graphs the values of the Fourier coefficients and Figure 4 graphs the square root of the sum of the squares of c_n and d_n for each value of n. Note that the trigonometric polynomial frequencies adjacent to the frequency of the function f are most prominent. The frequency is displayed in two ways, in terms of the frequency index, which represents the nth harmonic of the fundamental frequency $2\pi/L$, and in terms of radians/unit time.
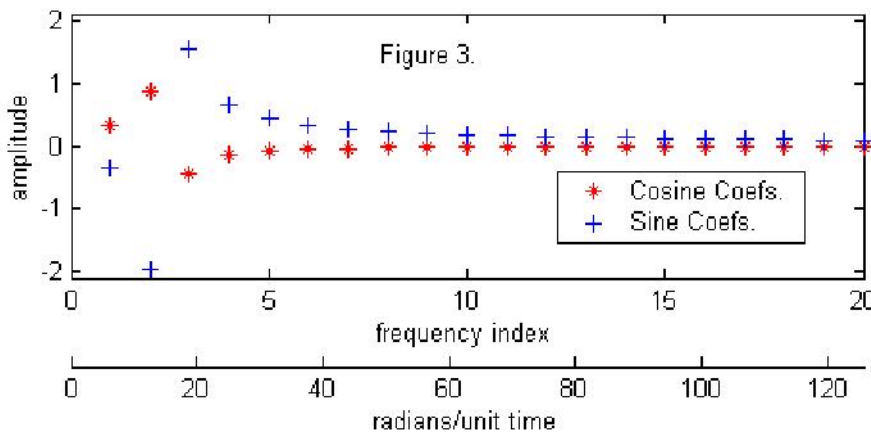
Figure 3.

Figure 4.

From Figure 1, it can be seen that the approximation does not seem as good near the ends of the initial interval. This is because the periodic extension of the function f is discontinuous at the ends of the interval. Discontinuities cause an oscillation in the approximating series independent of the number of terms used in the approximation.

This is phenomenon is call the Gibbs Phenomenon, which we will explore in exercise 4.

Exercise #4: Consider the unit jump at $x = a$ given by $f(t) = H(t - a)$. Where H is the

Heaviside Function

$H(t) = 0$ if $t < 0$ and 1 if $t \geqslant o$

Using the Sympy calculate the Fourier coefficients for this function and display in Figure 5 both the function and the first 20 terms of it's approximation and in Figure 6 displays 3 periods of the first 20 terms of the approximation, in Figures 7 and 8 display the first 40 terms of the approximation. Note that the overshoot at the jump does not decrease with the increase in number of terms, it remains at about 10% of the height of the jump.

The next exercise calculate the Fourier coefficients for a trend, that is for the line $f(t) = a + bt$ on the interval $[0, L]$.

Exercise #5: Assume that f(t) = a +bt on the interval [0, L]. Using the Sympy find the Fourier coefficients of its approximation. In Figure 9 graph the approximation to the periodic extension of this trend over three periods and Figure 10 graph the coefficients of the first twenty terms of the approximation.

If we study Figure 1 from Example #3, we find that the approximating series goes through the average value c_0 at the end points of the interval while the function does not. Hence, the periodic extension of the function has a discontinuity at each end point with the oscillatory behavior of the Gibbs Phenomenon.

Exercise #6: Suppose we multiply the original function in Exercise #3 by a function, which is zero at the ends of the initial interval. We will scale this function so that its maximum height is 1 at the center of the interval. Consider $f(t) = 3cos(15t)(16(t2)(L - t)2)/L4$. Then redo Exercise #3. In Figure 11 display the original function and the modified or windowed function, Figure 12 graph the coefficients, and Figure 13 graph the complex magnitude of the coefficients $(sqrt(c_n^2 + d_n^2))$.

The multiplication by a function, which is zero at the end points of the interval [0, L], to reduce the effect of discontinuities at the end points is called windowing. Compare Figures 4 and 13, note that

windowing has suppressed the high frequency components due to the discontinuities in the periodic extension of the given function.

From the above graphs (Figures 11, 12 and 13), notice that the original frequency of 15 radians/unit time which lies between the second and third harmonic of our trigonometric approximation stands out as the only feature.

(The second harmonic $2\pi2 = 12.57 < 15 < 2\pi3 = 18.85$ the third harmonic).

As the functions become more complicated, numerical integration must be used to evaluate the integrals defining the coefficients. This can be easily done in python using the built-in integration routines in scipy.integrate library.

Exercise #7: Continuing with Exercise #3, $f(t) = 3cos(15t)$, what happens when we change the length of the initial interval? From the formula for the fundamental frequency, $2\pi/L$, as L increase the fundamental frequency decreases. This decrease provides more detail in the graph of the complex magnitude of the coefficients for a given frequency. Thus, the longer the data set the more resolution is obtained in the frequency spectrum. Run the cases of L = 1, 2, 4 and 8 and plot the results in Figure 14.

Note that as L increases the complex magnitude of the coefficients are dominated by the given function's frequency of 15 radians/unit time. Thus, the original function must have the form

$f(t) = amplitude * \cos(frequency * t + \phi)$ where Φ is the phase angle. The phase angle is determined from the individual components of cosine and sine by the formula:

Phase Angle $\phi = \arctan(\sin n^{th} \text{ coef.}/\cos n^{th} \text{ coef.})$ where arctan is the four quadrant version.

In Figure 15 plot the phase angle for the case of L = 8.

From these two plots, the original function f(t) = 3 cos(15t) may be recovered.

Fourier Series of a Data Set (**Interpolation**): In many cases, the analytic expression for the function f is not known, only the values of f at a set of points in the interval [0, L]. If the Fourier series can be constructed from this data set then the results of Exercise #7 imply that periodic behavior of this data set can be determined. Suppose this data set is given. To define the Fourier series of this data set, construct the piecewise linear interpolant to this data and from this function compute the approximating Fourier series.
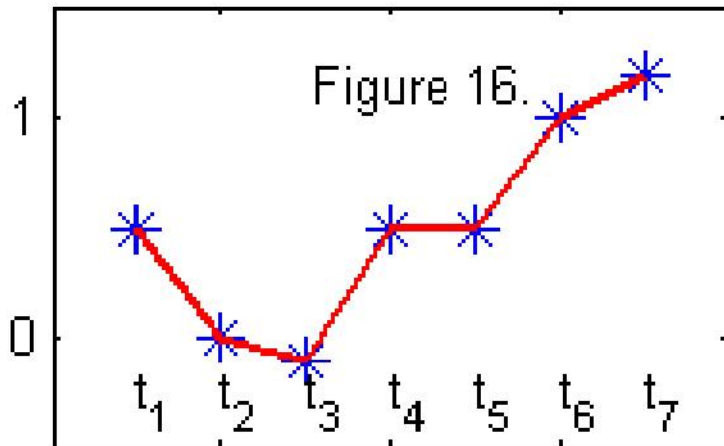
Piecewise Linear Interpolants

Suppose the following data set on the interval [0, L] is given.

$0 = t_1$	t_2	t_3	t_4	t_5	t_6	$L = t_7$
y_1	y_2	y_3	y_4	y_5	y_6	y_7

The graph of the piecewise linear interpolant is given in

Figure 16.

Figure 16.

In order to efficiently compute the Fourier series coefficients for this piecewise linear interpolant, write this interpolant as a linear combination of the following basis functions. Each of these basis functions will be the interpolant for a very simple data set based on

the original set of t-values or nodes. The first piecewise linear interpolant basis function θ_1 and its data set are given by the following:

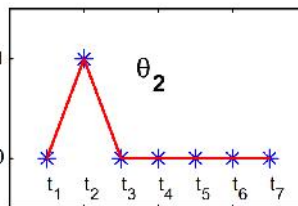| $0 = t_1$	t_2	t_3	t_4	t_5	t_6	$L = t_7$
1	0	0	0	0	0	0

$$\theta_1(t) = \begin{cases} \dfrac{t - t_2}{t_1 - t_2} & \text{for } 0 \le t \le t_2 \\ 0 & \text{otherwise} \end{cases}$$
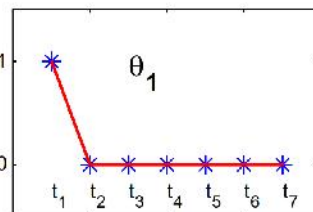
The second basis function θ_2 and its data set are given by the following:

$0 - t_1$	t_2	t_3	t_4	t_5	t_6	$L - t_7$
0	1	0	0	0	0	0

$$\theta_2(t) = \begin{cases} \dfrac{t - t_1}{t_2 - t_1} & t_1 \le t \le t_2 \\ \dfrac{t - t_3}{t_2 - t_3} & t_2 \le t \le t_3 \\ 0 & t_3 \le t \end{cases}$$

In a similar manner all of the remaining basis functions are defined θ_3 through θ_7.

The piecewise linear interpolant of the original data set may now be written in terms of these basis functions.

$$PL_{Data}(t) = \sum_{i=1}^{7} y_i \theta_i(t)$$

Since the computation of the Fourier series coefficients is a linear operation, it is enough to compute the coefficients for the basis functions θ_i and multiply by y_i and sum over all basis functions. Define the Fourier series of a data set on the interval [0, L] to be the Fourier series of the piecewise linear interpolant.

$$\mathfrak{J}(Data) \overset{\text{def}}{=} \mathfrak{J}(PL_{Data}(t)) = \sum_{i=1}^{7} y_i \, \mathfrak{J}(\theta_i(t))$$

Using the python Sympy lib, we can easily compute the Fourier coefficients for the basis functions. For θ_1 defined in the nodes (o,c) we obtain $c0first$

For θ_1 defined on the nodes (0,c) we obtain: c0first = c/(2*L)

cnfirst = -2*(cos((2*pi*n/L)*c)-1)/(L*c*(2*pi*n/L)^2)

dnfirst = 2*(-sin((2*pi*n/L)*c)+(2*pi*n/L)*c)/(L*c*(2*pi*n/L)^2)

The computation of the coefficients for the first basis function is given by the following MATLAB function subprogram.

```
function [c0,cn,dn]=FSPLBasisF(N,L,c)
%Calculates the Fourier Coefficients of the first PL Basis Function
%where N is the number of terms in the series
c0=c/(2*L);
n=1:N;
w=2*pi*n/L; Const=(2/(L*c))./(w.*w);
cn=-Const.*(cos(w*c)-1);
dn=Const.*(-sin(w*c)+w*c);
```

For θ_i

defined on the nodes (a,b,c) we obtain:

c0 = (c-a)/(2*L)

cn =2*(cos((2*pi*n/L)*b)*(c-a)+cos((2*pi*n/L)*a)*(b-c) +cos((2*pi*n/L)*c)*(a-b))/(L*(-b+a) *(-c+b)*(2*pi*n/L)^2)

dn = 2*(sin((2*pi*n/L)*b)*(c-a)+sin((2*pi*n/L)*a)*(b-c) +sin((2*pi*n/L)*c)*(a-b))/(L*(-b+a) *(-c+b)*(2*pi*n/L)^2)

The computation of the coefficients for the general basis function is given by the following MATLAB function subprogram.

```
function [c0,cn,dn]=FSPLBasis(N,L,a,b,c)
%Calculates the Fourier Coefficients for the PL Basis Functions
%where N is the number of terms in the series
c0=(c-a)/(2*L);
n=1:N;
w=2*pi*n/L; wsq=w.*w; Const=2/(L*(c-b)*(b-a))./wsq;
cn=Const.*(cos(w*b)*(c-a)+cos(w*a)*(b-c)+cos(w*c)*(a-b));
dn=Const.*(sin(w*b)*(c-a)+sin(w*a)*(b-c)+sin(w*c)*(a-b));
```

For $_{last}$ defined on the nodes (a,L) we obtain:

c0last = (L-a)/(2*L)

cnlast = 1/2*L*(-1+cos(2*pi*n/L*a))/((-L+a)*pi^2*n^2)

dnlast = -1/2*(-2*pi*n*(L-a)-sin(2*pi*n/L*a)*L)/((-L+a)*pi^2*n^2)

The computation of the coefficients for the last basis function is given by the following

MATLAB function subprogram.

```
function [c0,cn,dn]=FSPLBasisL(N,L,a)
%Calculates the Fourier Coefficients of the last PL Basis Function
%where N is the number of terms in the series
c0=(L-a)/(2*L);
n=1:N;
w=2*pi*n/L; Const=(2/(L*(L-a)))./(w.*w);
cn=Const.*(-cos(w*a)+1);
dn=Const.*(-sin(w*a)-(L-a)*w);
```

The Fourier series of a data set, can now be calculated using the above Fourier series

coefficient functions for the piecewise linear interpolation basis functions and our

definition. This methodology is exhibited in the next exercise.

Exercise #8:

Continuing with the function $f(t) = 3cos(15t)$, let $L = 8$ and choose 71 equally spaced points as samples of this function. Find the piecewise linear approximation and its Fourie approximation and plot the results. Suppose we now add noise to each of the sample points. Assume the noise is taken from a uniform distribution of amplitude 3 about zero. The following graphs give the results. In Figure 20 plot the original function, the PL interpolant and the sampled points.

Using this technique, Fourier coefficients may be computed for a PL interpolant for any data set having unique independent variable values. Exercise #8 provides evidence that periodic behavior can be recovered from noisy data. While this process is effective, it is computationally intensive.

Discrete Fourier Transform:

The Discrete Fourier Transform, or DFT, produces a complex vector X of length N from

an input vector x of length N by the following formula:

$$X(k) = \sum_{n=1}^{N} x(n) \exp\left(\frac{-2\pi i}{N}(k-1)(n-1)\right) \text{ for } 1 \le k \le N$$

The computation of the DFT is usually accomplished using the Fast Fourier Transform Algorithm or FFT. In python, the DFT is computed using the built-in function scipy.fft.fft()

The DFT operation has an inverse, or IDFT, given by the following formula:

$x(n) = \frac{1}{N} \sum_{k=1}^{N} X(k) \exp(\frac{2\pi i}{N}(k-1)(n-1))$ for $1 \le n \le N$

Exercise #9: Let the input vector x = [1, -2, 3, 6, 0, -1]. Thus, N = 6. Using the python routines for fft and ifft

calculates the DFT and IDFT and print k   vector x   X(k) = DFT(x)   IDFT(DFT(x))

For this particular vector x, note that IDFT(DFT) is just the identity and that

$X(k) = Conjugate(X(N - k - 2))$ for $2 \le k \le N$

Both of these observations can be proved for an arbitrary input vector.

Relationship of the DFT to the Interpolation of a Data Set:

Let x = [x1, x2, ... , xN] be a given vector of real values. Using these values form the following equally spaced data set on the interval [0, L]:

DATA SET

$t_1 = 0$	$t_2 = \Delta t$	......	$t_N = (N-1)\,\Delta t$	$t_{N+1} = N\,\Delta t = L$
x_1	x_2	......	x_N	x_1

Interpolation Theorem: The above data set is interpolated by the following trigonometric polynomial.

$$x(t) = a_0 + \sum_{k=1}^{k \le (N+1)/2} \left[a_k \cos\left(\frac{2\pi k}{L}t\right) + b_k \sin\left(\frac{2\pi k}{L}t\right) \right] \quad \text{where } a_0 = X(1)/N,$$

$$a_k = \frac{2 \, \text{Real}\,(X(k+1))}{N}, \quad b_k = \frac{-2 \, \text{Im}\,(X(k+1))}{N} \text{ , if N is even } a_{N/2} = \frac{X(N/2+1)}{N} \text{ and } X = DFT(x).$$

Note that if N is even X(N/2+1) is real, see Exercise #9.

The proof of this theorem begins with the definition of the IDFT, using the properties observed in Exercise #9, the theorem follows by a careful manipulation and recasting of the terms. Note that we may choose L arbitrarily.

Exercise #10: Consider the vector of values given in Example #9. Let L = 1, and

Data Set :

0	1/6	1/3	1/2	2/3	5/6	1
1	-2	3	6	0	-1	1

Graph both this data set and the Trigonometric Polynomial defined in the Theorem in ipython.