San Francisco State University

Engineering 305

Laboratory #5 – Fourier series

Purpose

In this lab, you will investigate the properties of orthogonal functions and the Fourier series. This is a long lab, so plan accordingly!

Background reading includes:

- Lathi, Chapter3
- Holton notes, Unit 4

Background

Orthogonal basis functions are central to the analysis of linear systems. The most important of these functions is the complex exponential, $e^{j\omega t}$, where ω is the angular frequency, which is real. In this laboratory exercise, we are specifically interested in the family of complex exponential basis functions, $\varphi_k(t)$, that are related to each other by having angular frequencies that are multiples of a common fundamental frequency, ω_0

$$\varphi_k(t) = e^{jk\omega_0 t} \tag{L4.1}$$

Each function can be viewed a vector of length 1 rotating at a different frequency in the complex plane. $\varphi_1(t)$ rotates at the fundamental frequency, $\omega_0 = 2\pi/T_0$, where T_0 is the fundamental period. $\varphi_k(t)$ rotates at a frequency $k\omega_0$ radians/sec, where k is an integer that ranges from $-\infty < k < \infty$.

Complex exponentials are important for the study of linear systems for two reasons: 1) they are *orthogonal* and 2) they are *eigenfunctions* (scary word!) of linear systems. In this lab we will look at the orthogonality property. In the next lab, we will look at the eigenfunction property.

Orthogonality

The key observation is that complex exponentials form an orthogonal set of basis functions over an interval $0 \le t < T_0$. Mathematically this means that

$$\frac{1}{T_0}\int_0^{T_0} \varphi_n(t)\varphi_m^*(t)dt = \begin{cases} 1, & n = m \\ 0, & n \ne m \end{cases} \tag{L4.2}$$

The proof is simple. Plugging Equation (L4.1) into Equation (L4.2) gives

$$\begin{aligned} &\frac{1}{T_0}\int_0^{T_0} e^{jn\omega_0 t}e^{-jm\omega_0 t}dt = \frac{1}{T_0}\int_0^{T_0} e^{j(n-m)\omega_0 t}dt = \\ &\frac{1}{j(n-m)\omega_0 T_0} e^{j(n-m)\omega_0 t}\Big|_0^{T_0} = \frac{e^{j(n-m)\omega_0 T_0} - 1}{j(n-m)\omega_0 T_0} = \frac{e^{j(n-m)2\pi} - 1}{j(n-m)2\pi}, \end{aligned} \tag{L4.3}$$

where we've recognized that

$$\omega_0 T_0 = \frac{2\pi}{\cancel{T_0}}\cancel{T_0} = 2\pi \ .$$

If $n \neq m$, the numerator in Equation (L4.3) is zero and the denominator is nonzero; hence, the entire integral is zero. If $n = m$, both the numerator and denominator are zero and hence the integral is indeterminate. We can determine the value of the integral directly when $n = m$

$$\tfrac{1}{T_0}\int_0^{T_0} e^{j(n-m)\omega_0 t}dt = \tfrac{1}{T_0}\int_0^{T_0} e^{j0\omega_0 t}dt = \tfrac{1}{T_0}\int_0^{T_0} 1dt = 1 \qquad \text{(L4.4)}$$

So Equation (L4.3) reduces to

$$\tfrac{1}{T_0}\int_0^{T_0} e^{jn\omega_0 t}e^{-jm\omega_0 t}dt = \begin{cases} 1, & n = m \\ 0, & n \neq m \end{cases} \qquad \text{(L4.5)}$$

as described in Equation (L4.2).

Fourier series

A large number of time functions, $x(t)$, can be represented over a given time interval, $\left[T_{\min}, T_{\max}\right)$, by the sum of an infinite number of complex exponential basis functions, each scaled by a different complex constant. This summation is called the *Fourier series* summation:

$$x(t) = \sum_{k=-\infty}^{\infty} x_k e^{jk\omega_0 t} \qquad \text{(L4.6)}$$

where x_k are complex coefficients called the *Fourier series coefficients*. You can think of this as a summation an infinite number of complex exponential vectors each one rotating at a different frequency, $k\omega_0$ radians/sec that is a multiple of the fundamental frequency, ω_0 , and each one with a magnitude and phase given by the Fourier series coefficient, x_k .

The Fourier series coefficients can be computed using the orthogonality property of complex exponentials. Letting $T_{\min} = 0$ and $T_{\max} = T$, and defining $T_0 \triangleq T_{\max} - T_{\min}$, we note that

$$\begin{gathered} \tfrac{1}{T_0}\int_0^{T_0} x(t)e^{-jk\omega_0 t}dt = \tfrac{1}{T_0}\int_0^{T_0}\sum_{m=-\infty}^{\infty} x_m e^{jm\omega_0 t}e^{-jk\omega_0 t}dt = \\ \sum_{m=-\infty}^{\infty} x_m \underbrace{\left(\tfrac{1}{T_0}\int_0^{T_0} e^{jm\omega_0 t}e^{-jk\omega_0 t}dt\right)}_{\substack{1,\ k=m \\ 0,\ k\neq m}} = x_k \end{gathered} \quad , \qquad \text{(L4.7)}$$

where we've noted from Equation (L4.5) that the quantity in the brackets in Equation (L4.7) is 1 only when $k = m$, so that only one term is picked out of the summation.

Thus, we arrive at a pair of relations

$$
\boxed{\begin{array}{ll} x(t) = \sum_{k=-\infty}^{\infty} x_k e^{jk\omega_0 t} & \text{Synthesis} \\ x_k = \frac{1}{T_0}\int_0^{T_0} x(t) e^{-jk\omega_0 t}\, dt & \text{Analysis} \end{array}} \tag{L4.8}
$$

The first relation, the *synthesis* relation, tell how we represent an arbitrary $x(t)$ over a given time interval, $[0,T)$ as the sum of complex exponentials, $e^{jk\omega_0 t}$ which are multiples of fundamental frequency, ω_0, each one with a magnitude and phase given by the Fourier series coefficient, x_k. The second relation, the *analysis* relation, tells how we determine the Fourier series coefficients from a given $x(t)$.

As an example, consider a pulse, $x(t)$, of length $t_0 = 1\,\text{sec}$ as shown in Figure 1.

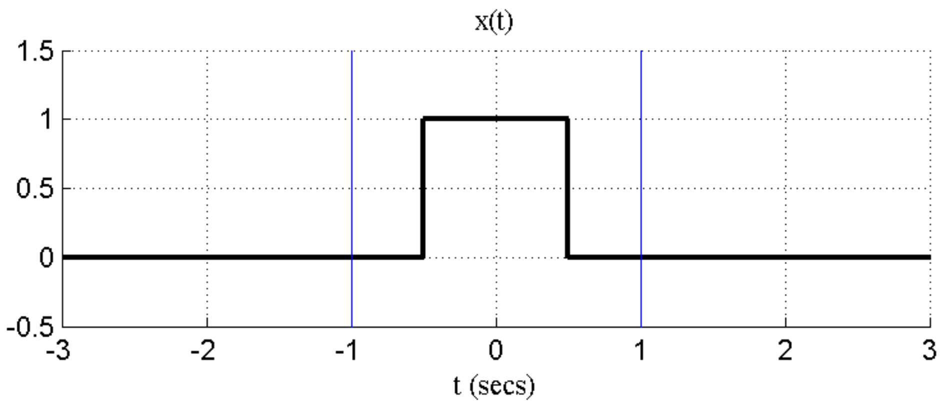

Figure 1: Pulse

1. The first step in performing the Fourier analysis of this function is to choose an analysis interval. Here, we choose $T_{\min} = -1$ and $T_{\max} = 1$. Hence, $T_0 = T_{\max} - T_{\min} = 2$.
2. Now, we compute the Fourier series coefficients, x_k, using the analysis relation of Equation (L4.8). For this case, you can show that $x_k = 0.5\,\text{sinc}(k\pi/2)$.
3. Finally, we resynthesize $x(t)$ using the synthesis relation of Equation (L4.8). For this example, let's split the summation up into three parts, for $k < 0$, $k = 0$ and $k > 0$:

$$
\begin{aligned} x(t) &= \sum_{k=-\infty}^{\infty} x_k e^{jk\omega_0 t} = \sum_{k=-\infty}^{-1} x_k e^{jk\omega_0 t} + x_0 + \sum_{k=1}^{\infty} x_k e^{jk\omega_0 t} \\ &= x_0 + \sum_{k=1}^{\infty}\left(x_k e^{jk\omega_0 t} + x_{-k} e^{-jk\omega_0 t}\right) \end{aligned} \tag{L4.9}
$$

Now note that the sinc function is even, so $x_k = x_{-k}$. Thus, the synthesis relation can be rewritten as

$$
x(t) = x_0 + \sum_{k=1}^{\infty} x_k \left(e^{jk\omega_0 t} + e^{-jk\omega_0 t}\right) = x_0 + 2\sum_{k=1}^{\infty} x_k \cos k\omega_0 t \tag{L4.10}
$$

Further simplifications are possible (see my Notes), but for now, this is good enough. It is useful to define $\hat{x}_M(t)$ as a resynthesis of the original $x(t)$ by using only M of the terms of the summation in Equation (L4.9):

$$\hat{x}_M(t) = x_0 + 2\sum_{k=1}^{M} x_k \cos k\omega_0 t \tag{L4.11}$$

We can show that as the number of terms in the resynthesis, M, increases, the difference between $x(t)$ and $\hat{x}_M(t)$ gets monotonically smaller. We can get a feeling for this graphically, by plotting $\hat{x}_M(t)$ for a few values of M (in red).

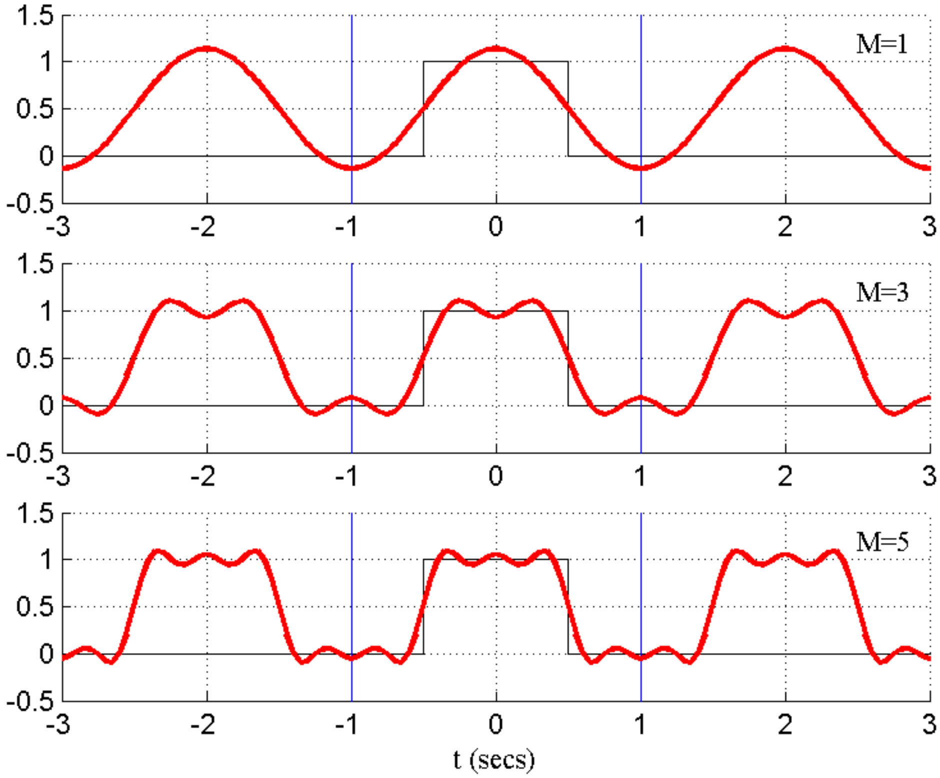

Figure 2: Fourier series resynthesis with 1, 3 and 5 terms

More precisely, we can show that E_M, the mean square error of the difference between $x(t)$ and $\hat{x}_M(t)$ is a monotonically decreasing function of M

$$E_M = \left[\int_{-T/2}^{T/2} |x(t)|^2 \, dt - T \sum_{-M}^{M} |x_k|^2 \right]^{1/2}. \tag{L4.12}$$

In the assignment, you will explore what happens as we analyze and resynthesis $x(t)$ using different parameters, such as t_0, T, and M.

3. Assignment

This assignment has three parts. First, we need to get some graphical sense of what the complex exponential functions look like. Then we discuss the orthogonality of these functions and finally, we look at how they can be used to represent arbitrary time functions using the Fourier series.

1. Visualization of the complex exponential basis functions, $\varphi_k(t) = e^{jk\omega_0 t}$. First consider a family of functions, $\varphi_k(t) = e^{jk\omega_0 t}$, with a fixed value of the fundamental frequency, $\omega_0 = 2\pi/64$. This means that $\varphi_1(t)$ is periodic with a period of 64 seconds. Use Matlab to create an array, $x[n]$, that corresponds to one period of $\varphi_1(t)$ sampled at 1 second intervals: $x[n] = e^{j\cdot 1\cdot\omega_0(0:63)}$. You should get the plot on the left side of Figure 3

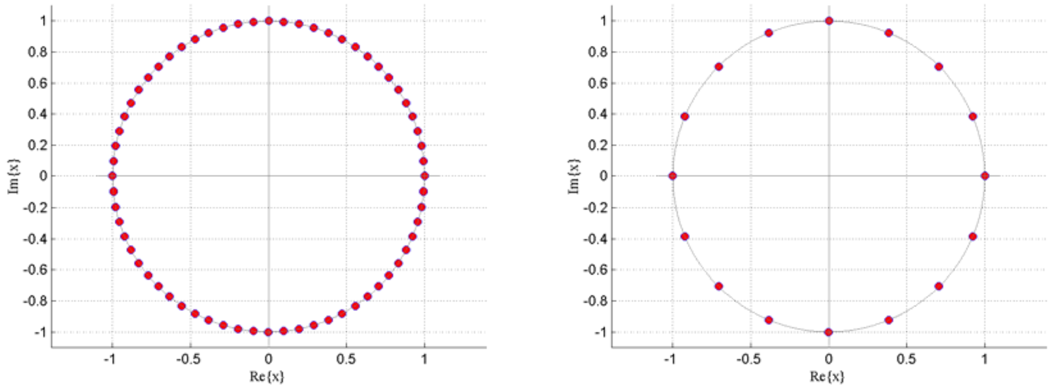

Figure 3: Complex exponential at ω_0 (left) and $4\omega_0$ (right)

Plot the real vs. imaginary parts of this array. Repeat for one period of an array that corresponds to one period of $\varphi_4(t)$, again sampled at one second intervals. You should get the plot on the right side of Figure 3. This should give you a visualization of $\varphi_k(t) = e^{jk\omega_0 t}$ as a vector spinning in the complex plane at a frequency $k\omega_0$ radians/sec. As k increases, the speed of the rotating vector increases.

Note that to plot a complex quantity, `x`, in Matlab you can always do this:

```
plot(real(x), imag(x));
```

but it is simpler to do this:

```
plot(x);
```

If `x` is a real array, Matlab plots it against the index of the array. But if `x` is a complex array, Matlab plots the real part against the imaginary part.

2. Orthogonality of $\varphi_k(t) = e^{jk\omega_0 t}$. We wish to investigate the orthogonality of complex exponential functions as expressed in Equation (L4.2). Although Matlab cannot deal numerically with continuous-time functions, we can create arrays, $\varphi_k[n]$, by sampling $\varphi_k(t)$ at multiples of some interval, Δ, which is an integer fraction, *N*, of the fundamental period. That is, $t = n\Delta$, where $\Delta = T_0/N$. The result is

$$\varphi_k[n] = e^{jk\omega_0 n\Delta} = e^{jk\frac{2\pi}{T_0}n\frac{T_0}{N}} = e^{jk\frac{2\pi}{N}n} \qquad \text{(L4.13)}$$

Notice that $\varphi_k[n]$ no longer depend on ω_0 and *t*, but on *N* and *n*. For example, the plots in Figure 3 correspond to $\varphi_1[n]$ and $\varphi_4[n]$ with $N = 64$.

We now seek to show that the sampled functions, $\varphi_k[n]$, are orthogonal, specifically that

$$\frac{1}{N}\sum_{n=0}^{N-1}\varphi_k[n]\varphi_m^*[n]=\begin{cases}1, & k=m\\ 0, & k\neq m\end{cases} \tag{L4.14}$$

a. Let $N=16$ Create arrays $\varphi_1[n]$ and $\varphi_2[n]$. Use Matlab to evaluate the summation in Equation (L4.14) for all permutations of k = 1 and 2 and m = 1 and 2. You should get four values. Do your results make sense? Note that you can be clever or not-so-clever in evaluating the sum in Matlab. For example, if you have two complex N-pt arrays, `x` and `y`, you can compute `sum(x.*conj(y))` (no-so-clever) or just `x*y'` (clever). Why does this work?

b. In the preceding part, we checked the orthogonality of $\varphi_k[n]$ at just two values of k. Now we want check orthogonality for *all* possible values of k. But how many k are there? Show theoretically that $\varphi_k[n]=\varphi_{k+N}[n]$ for all n. Why is this? Check it with Matlab by comparing $\varphi_1[n]$ and $\varphi_{1+N}[n]$ for $N=16$. This shows that for an arbitrary N, there are only N unique $\varphi_k[n]$.

c. From part b. above, we need to check orthogonality of N functions against each other. This means we have to make N^2 comparisons. The results are best shown in a matrix that we expect to be equivalent to the identity matrix. (Why?)

$$\begin{bmatrix}\sum\varphi_1,\varphi_1^* & \cdots & \sum\varphi_1,\varphi_N^*\\ \vdots & \ddots & \vdots\\ \sum\varphi_N,\varphi_1^* & \cdots & \sum\varphi_N,\varphi_N^*\end{bmatrix}=\begin{bmatrix}1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 1\end{bmatrix}. \tag{L4.15}$$

Again, there are clever and not-so-clever ways to compute this matrix with Matlab. The not-so-clever way is with a `for` loop. That's fine if you want to do it that way. This is not a course in the wonders of clever Matlab coding. However, if think about it a bit, you can exploit the properties of matrix multiplication to create and multiply two $N\times N$ matrices together with a line or two to get the result shown in Equation (L4.15).

3. Construction of Fourier series. Now that we've established the orthogonality of complex exponential functions, we are ready to look at the Fourier series in more detail. In this exercise, you'll design a Matlab function to produce the plots of Figure 2. You'll use the pulse, $x(t)$, of length $t_0=1$ sec and choose $T_{\min}=-1$ and $T_{\max}=1$ so your plots will match those of Figure 2. Choose the time limits of the plot to be $-3\le t\le 3$ and pick a large number of points per second (e.g. 1 msec) for your time vector. Here's the specification of the function:

```
function plotfs(t0, T, M)
% PLOTFS  Fourier synthesis
%   plotfs(t0, T, M) plots resynthesis of a pulse function
%   of width t0 over analysis interval [-T/2, T/2)
%   using the partial sum of the DC plus M terms.
```

Note a few things about your results.

a. As the number of terms in the partial summation of Equation (L4.11) increases, $\hat{x}_M(t)$ matches $x(t)$ increasingly well in the interval $\left[-T/2, T/2\right)$.

b. Outside the interval $\left[-T/2, T/2\right)$, the resynthesized function doesn't match the original function at all well because the resynthesized function, $\hat{x}_M(t)$, is *guaranteed* to be periodic, even if $x(t)$ is not. Why is this?

4. Error of the Fourier series approximation. As noted in the background section of this lab, E_M, the mean-square error between the partial Fourier series approximation, $\hat{x}_M(t)$ and the original $x(t)$,

decreases as the number of terms in the approximation increases, as indicated by Equation (L4.12). Given the pulse, $x(t)$, of length $t_0 = 1\,\text{sec}$, $T_{\min} = -1$ and $T_{\max} = 1$, plot E_M as a function of *M*, the number of terms on a log-log scale. Consider *M* ranging from 1 to 1000 terms. From your plot, you should see that

a. The error decreases monotonically as *M* increases.

b. Each term that you add to the summation reduces the error by roughly a constant percentage.

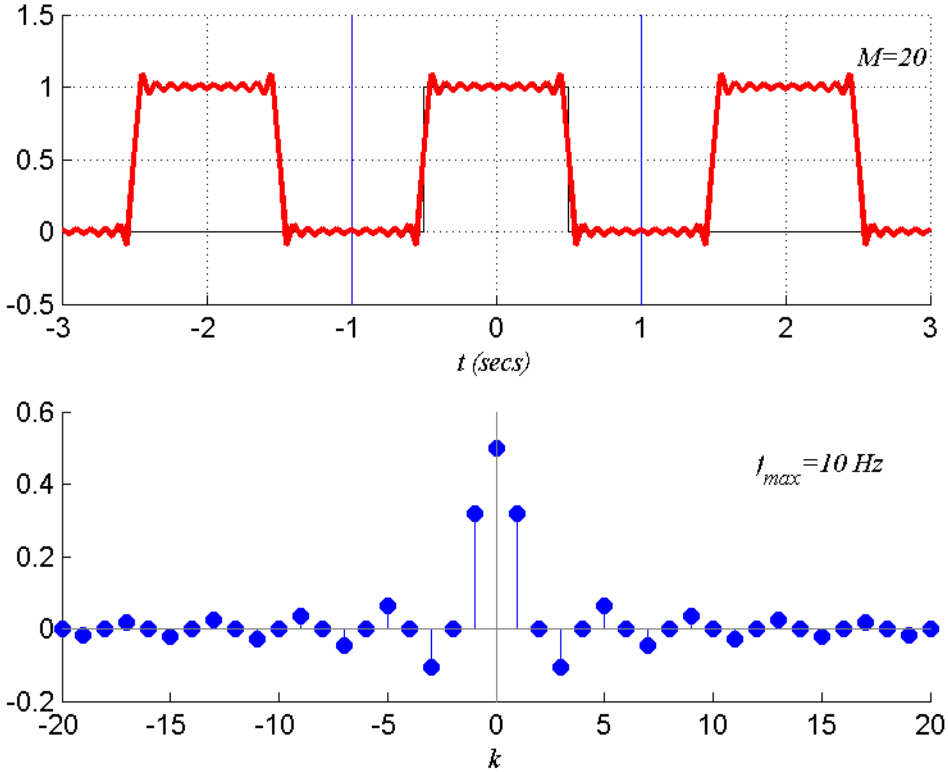

Figure 4: Fourier series coefficients

5. Relation of $x(t)$ to x_k. Create a function, `plotk(t0, T, M)`, to plot the first M Fourier series.

```
function plotk(t0, T, M)
% PLOTK  Fourier coefficients
%   plotk(t0, T, k) plots 2M+1 Fourier series coefficients
%   (i.e. –M<=k<=M) of a pulse of width t0
%   over analysis interval [-T/2, T/2).
```

You can use `stem(..., 'filled')` to do this. When you use both `plotfs` and `plotk` on two panels of a plot, your result should look that shown in Figure 4. Note that I have indicated in the top right corner of the bottom plot that the largest included Fourier-series component, $k = 20$, corresponds to a complex exponential with a frequency, $f_{\max} = 10\,Hz$.

a. Create plots with $t_0 = 1\,\text{sec}$, $T = 2$ and $M = 20$.

b. Create plots with $t_0 = 2\,\text{sec}$, $T = 4$ and $M = 20$. Compare the results of part a. and b. Note that the *x(t)* plotted in the upper panels are different, but the Fourier series coefficients plotted in the lower panels are exactly the same. How can this be? How can two different time functions have the same Fourier series coefficients?

c. Create plots with $t_0 = 1\sec$, $T = 4$ and $M = 40$. Compare the results of part a. and c. Note that the *x(t)* plotted in the upper panels look the same in the range $-1 < t \le 1$, but the Fourier series coefficients plotted in the lower panels are different. Why is this? How can two time functions that look the same have different Fourier series coefficients?

d. If you compare the Fourier series coefficients in the plots of part a. and c, you'll notice that $f_{\max} = 10\,Hz$ in both plots, but that the number of Fourier-series coefficients required to span this range is double in the plot of part c compared with that for part a. Why? Notice also that the Fourier series coefficients of the two plots are related. Specifically, note that the Fourier series coefficients for $k = 0, \pm 2, \pm 4, \ldots$ in the part c are exactly the same as the Fourier series coefficients for $k = 0, \pm 1, \pm 2, \ldots$ in part a , except scaled by ½. That is, the coefficients in the two plots that correspond to the same frequency are the same, when scaled by ½. Argue that as *T* doubles, we need twice as many Fourier series coefficients to span the same range of frequencies. What do you think would happen if you created a plot with $t_0 = 1\sec$, $T = 8$ and $M = 80$. Try it! Comment on what you expect to happen in the limit as $T \to \infty$.
