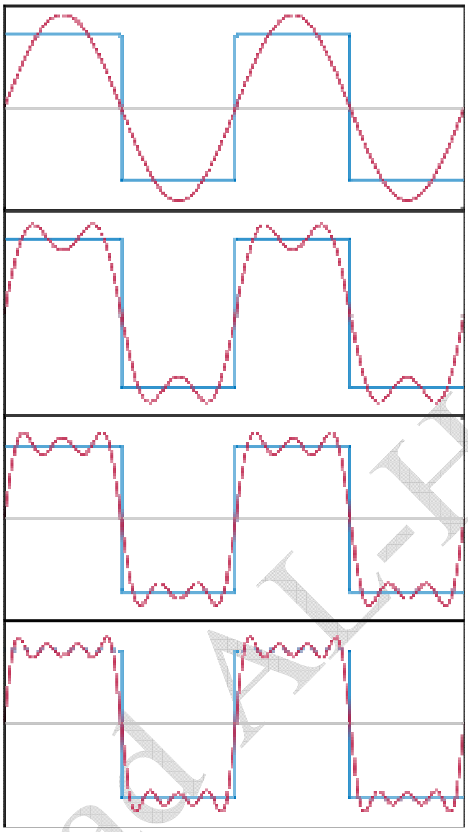
Fourier series and Applications



The first four Fourier series approximations for a square wave.

In mathematics, a **Fourier series** decomposes any periodic function or periodic signal into the sum of a (possibly infinite) set of simple oscillating functions, namely sines and cosines (or complex exponentials). The study of Fourier series is a branch of Fourier analysis. Fourier series were introduced by Joseph Fourier (1768–1830) for the purpose of solving the heat equation in a metal plate.

The heat equation is a partial differential equation. Prior to Fourier's work, there was no known solution to the heat equation in a general situation, although particular solutions were known if the heat source behaved in a simple way, in particular, if the heat source was a sine or cosine wave. These simple solutions are now sometimes called eigensolutions. Fourier's idea was to model a complicated heat source as a superposition (or linear combination) of simple sine and cosine waves, and to write the solution as a superposition of the corresponding eigensolutions. This superposition or linear combination is called the Fourier series.

Although the original motivation was to solve the heat equation, it later became obvious that the same techniques could be applied to a wide array of mathematical and physical problems.

The Fourier series has many applications in electrical engineering, vibration analysis, acoustics, optics, signal processing, image processing, quantum mechanics, econometrics,[1] thin-walled shell theory,[2] etc.

Fourier series is named in honour of Joseph Fourier (1768-1830), who made important contributions to the study of trigonometric series, after preliminary investigations by Leonhard Euler, Jean le Rond d'Alembert, and Daniel Bernoulli. He applied this technique to find the solution of the heat equation, publishing his initial results in his 1807 *Mémoire sur la propagation de la chaleur dans les corps solides* and 1811, and publishing his *Théorie analytique de la chaleur* in 1822.

From a modern point of view, Fourier's results are somewhat informal, due to the lack of a precise notion of function and integral in the early nineteenth century. Later, Dirichlet and Riemann expressed Fourier's results with greater precision and formality.

Revolutionary article:

“

$$\varphi(y) = a\cos\frac{\pi y}{2} + a'\cos 3\frac{\pi y}{2} + a''\cos 5\frac{\pi y}{2} + \cdots.$$

Multiplying both sides by $\cos(2k+1)\frac{\pi y}{2}$, and then integrating from $y = -1$ to $y = +1$ yields:

$$a_k = \int_{-1}^{1} \varphi(y)\cos(2k+1)\frac{\pi y}{2}\,dy.$$

”

In these few lines, which are close to the modern formalism used in Fourier series, Fourier revolutionized both mathematics and physics. Although similar trigonometric series were previously used by Euler, d'Alembert, Daniel Bernoulli and Gauss, Fourier believed that such trigonometric series could represent arbitrary functions. In what sense that is actually true is a somewhat subtle issue and the attempts over many years to clarify this idea have led to important discoveries in the theories of convergence, function spaces, and harmonic analysis.

When Fourier submitted a later competition essay in 1811, the committee (which included Lagrange, Laplace, Malus and Legendre, among others) concluded: *...the manner in which the author arrives at these equations is not exempt of difficulties and...his analysis to integrate them still leaves something to be desired on the score of generality and even rigour*.

Birth of harmonic analysis:

Since Fourier's time, many different approaches to defining and understanding the concept of Fourier series have been discovered, all of which are consistent with one another, but each of which emphasizes different aspects of the topic. Some of the more powerful and elegant approaches are based on mathematical ideas and tools that were not available at the time Fourier completed his original work. Fourier originally defined the Fourier series for real-valued functions of real arguments, and using the sine and cosine functions as the basis set for the decomposition.

Many other Fourier-related transforms have since been defined, extending the initial idea to other applications. This general area of inquiry is now sometimes called harmonic analysis. A Fourier series, however, can be used only for periodic functions.

Exponential Fourier series:

We can use Euler's formula,

$$e^{inx} = \cos(nx) + i\sin(nx),$$

where *i* is the imaginary unit, to give a more concise formula:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx}.$$

The Fourier coefficients are then given by:

$$c_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x) e^{-inx}\, dx.$$

The Fourier coefficients a_n, b_n, c_n are related via

$$a_n = c_n + c_{-n} \quad \text{for } n = 0, 1, 2, \dots$$
$$b_n = i(c_n - c_{-n}) \quad \text{for } n = 1, 2, \dots$$

and

$$c_n = \begin{cases} \frac{1}{2}(a_n - ib_n) & n > 0 \\ \frac{1}{2}a_0 & n = 0 \\ \frac{1}{2}(a_{-n} + ib_{-n}) & n < 0 \end{cases}$$

The notation c_n is inadequate for discussing the Fourier coefficients of several different functions. Therefore it is customarily replaced by a modified form of f (in this case), such as F or $\hat{f}$, and functional notation often replaces subscripting. Thus:

$$f(x) = \sum_{n=-\infty}^{\infty} \hat{f}(n) \cdot e^{inx}$$
$$= \sum_{n=-\infty}^{\infty} F[n] \cdot e^{inx} \quad \text{(engineering)}.$$

In engineering, particularly when the variable x represents time, the coefficient sequence is called a frequency domain representation. Square brackets are often used to emphasize that the domain of this function is a discrete set of frequencies.

Fourier series on a square

We can also define the Fourier series for functions of two variables x and y in the square $[-\pi, \pi]\times[-\pi, \pi]$:

$$f(x, y) = \sum_{j,k\in\mathbb{Z} \text{ (integers)}} c_{j,k} e^{ijx} e^{iky},$$

$$c_{j,k} = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} f(x,y) e^{-ijx} e^{-iky} \, dx \, dy.$$

Aside from being useful for solving partial differential equations such as the heat equation, one notable application of Fourier series on the square is in image compression. In particular, the jpeg image compression standard uses the two-dimensional discrete cosine transform, which is a Fourier transform using the cosine basis functions.

General case:

There are many possible avenues for generalizing Fourier series. The study of Fourier series and its generalizations is called harmonic analysis.

Generalized functions:

One can extend the notion of Fourier coefficients to functions which are not square-integrable, and even to objects which are not functions. This is very useful in engineering and applications because we often need to take the Fourier series of a periodic repetition of a Dirac delta function. The Dirac delta δ is not actually a function; still, it has a Fourier transform and its periodic repetition has a Fourier series:

$$\hat{\delta}(n) = \frac{1}{2\pi} \text{ for every } n.$$

This generalization to distributions enlarges the domain of definition of the Fourier transform from $L^2([-\pi, \pi])$ to a superset of L^2. The Fourier series converges weakly.

Fourier Methods in Signal Processing

The smoothness of a signal

A signal or function is referred to as being ``smooth'' if the function and ``sufficiently many'' of its derivatives are continuous. The more derivatives that are continuous, the smoother is our signal.

Since Fourier analysis means representing a signal with respect to a basis of periodic functions, our signal should also be periodic for an efficient representation in this basis. This is of course not true for all signals. Since we will study finite signals in this lab, we therefore think of our finite signal to be periodically extended as described in the beginning of Chapter 2.1 in [1]. Once we have periodically extended our signal in this manner, there is a useful ``rule of thumb'' that relates the structure of our signal to its spectrum:

- If a (periodically extended) signal is smooth, then the spectrum decays ``relatively'' fast and vice versa.

This rule of thumb can of course be stated (and proved) more rigorously. However, for this lab the above statement should be sufficient to explain some of the spectra you will encounter.

The fundamental principle for signal processing using Fourier analysis

A fundamental principle when processing a signal using Fourier analysis is to *manipulate the spectrum of a signal* rather that manipulating the signal itself. Hence, the usual procedure is to find the DFT of a signal, manipulate the DFT vector by, for example, letting some of its elements equal zero to get rid of unwanted frequencies, and then transform back our signal using the inverse DFT.

Compression

When we compress a signal using Fourier analysis we neglect frequencies that have zero or almost zero magnitude in the spectrum. In many cases this means that we only need a small fraction of the spectrum to represent a signal. Instead of storing the whole signal, we just store the coefficients for the largest frequencies in the spectrum.

Let us take an example. Consider a signal with 512 samples, that is, 512 data points. To store the whole signal requires 512 pieces of information to be stored. If we take the DFT (fft() in Matlab) of our signal, we get a new vector with 512 elements. Usually a great portion of this vector is ``almost'' zero. It may very well be that only 50 of the elements are ``large enough'' to significantly contribute to the signal. Therefore, we store only these 50 elements of the DFT vector. Once we need our signal, we just take an inverse transform of our DFT vector to reconstruct a good approximation of our original signal (ifft() in Matlab). For a definition of how to measure compression ratio, please see The Singular Value Decomposition of an Image.

The l^2-norm of a signal

When measuring the error in signal processing, we usually use the l^2-norm. Let our (finite) signal be given by $z = (z(0), z(1), \dots, z(N-1))$ and a processed version of the original signal be given by $\tilde{z} = (\tilde{z}(0), \tilde{z}(1), \dots, \tilde{z}(N-1))$. We define the relative l^2-error as

$$E_{rel} = \sqrt{\frac{\sum_{k=0}^{N-1} |z(k) - \tilde{z}(k)|^2}{\sum_{k=0}^{N-1} |z(k)|^2}} \quad (1)$$

Are Fourier methods still competitive in signal processing?

Even though wavelets outperform Fourier methods in many situations, there are still some signals which are better represented using Fourier methods. Also, even if we decide to use wavelets for an application, understanding wavelet analysis often requires solid knowledge in Fourier analysis.

Some useful Matlab commands

The following table gives a few commands that will be useful for this lab.

Operation:	**Matlab command**
The DFT of a signal (vector) *z*.	
(non-normalized)	fft(z)
The inverse DFT of a vector *fz*.	
(Normed with factor 1/*N*, where	
N is the length of the vector)	ifft(fz)
Switch position of first and second half	
of vector *fz*.	fftshift(fz)
Start a ``stop watch''	tic
Find the time elapsed since the stop watch started.	time=toc
Generate a symmetric Toeplitz matrix where	
the first column is given by the vector *c*	toeplitz(c)
Generate a Toeplitz matrix where	
the first column is given by the vector *c* and the first row	
is given by the vector *r*	toeplitz(c,r)

The real part of a scalar (or vector) *z*.	real(z)
The imaginary part of a scalar (or vector) *z*.	imag(z)
The magnitude part of a scalar (or vector) *z*.	abs(z)

Note that the last three commands in the table act element-wise on vectors. For example, if *z*=(-1,3+4*i*,-2*i*) then abs(z) will give *z*=(1,5,2) as output. When asked to find the magnitude of a signal in the exercises below, this means that you should use the command abs(). (As opposed to finding the norm of a vector for which you want to use the command norm().)

Warning! When multiplying a matrix *A* with a vector *x* to form *Ax*, the vector has to be a column vector! You can form the transpose in Matlab with the ' operator.

Displaying the spectrum

When you display the DFT of a signal in Matlab, the default setting is that the low frequencies are displayed at the right and left part of the plot with high frequencies in the center. This is how the spectra in Chapter 2.1 in [1] are shown. However, in many texts, one displays the spectrum with the low frequencies in the center, and the high frequencies at the left and right edge of the plot. Which way you choose is up to you. You can accomplish the latter alternative by using the command fftshift().

Fourier Methods Applied To Image Processing:

Background

The theory introduced for one dimensional signals above carries over to two dimensional signals with minor changes. Our basis functions now depend on two variables (one in the *x*-direction and one in the *y*-direction) and also two frequencies, one in each direction. See Exercises 2.15-2.18 in [1] for more details. The corresponding l^2-norm for a two dimensional signal now becomes

$$||A||_2 = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}|^2} \qquad (2)$$

Where a_{ij} are the elements in the $M \times N$ matrix representing the two dimensional signal. It is computed in Matlab using the Frobenius norm.

Some useful Matlab commands

The following table gives some commands that will be useful for this part of the lab.

Operation:	**Matlab command**
The DFT of a 2D signal (matrix) *A*.	
(non-normalized)	fft2(A)
The inverse DFT of an $M \times N$ matrix *fA*.	
(Normalized with a factor 1/*MN*.)	ifft2(fA)
Switch position of first and third quadrant and	
second and third quadrant of matrix *fA*.	fftshift(fA)

The commands real, imag and abs work on matrices elementwise just as in the one dimensional case.

Displaying the spectrum

When you display the DFT of a two dimensional signal in Matlab, the default setting is that the low frequencies are displayed towards the edges of the plot and high frequencies in the center. However, in many situations one displays the spectrum with the low frequencies in the center, and the high frequencies near the edges of the plot. Which way you choose is up to you. You can accomplish the latter alternative by using the command fftshift().

There is a very useful trick to enhance the visualization of the spectrum of a two dimensional signal. If you take the logarithm of the gray-scale, this usually give a better plot of the frequency distribution. In case you want to display the spectrum fA, I recommend to type imshow(log(abs(fA))) for a nice visualization of the frequency distribution. You may also want to use fftshift as described above, but that is more a matter of taste.