Lecture 8

Fourier Series

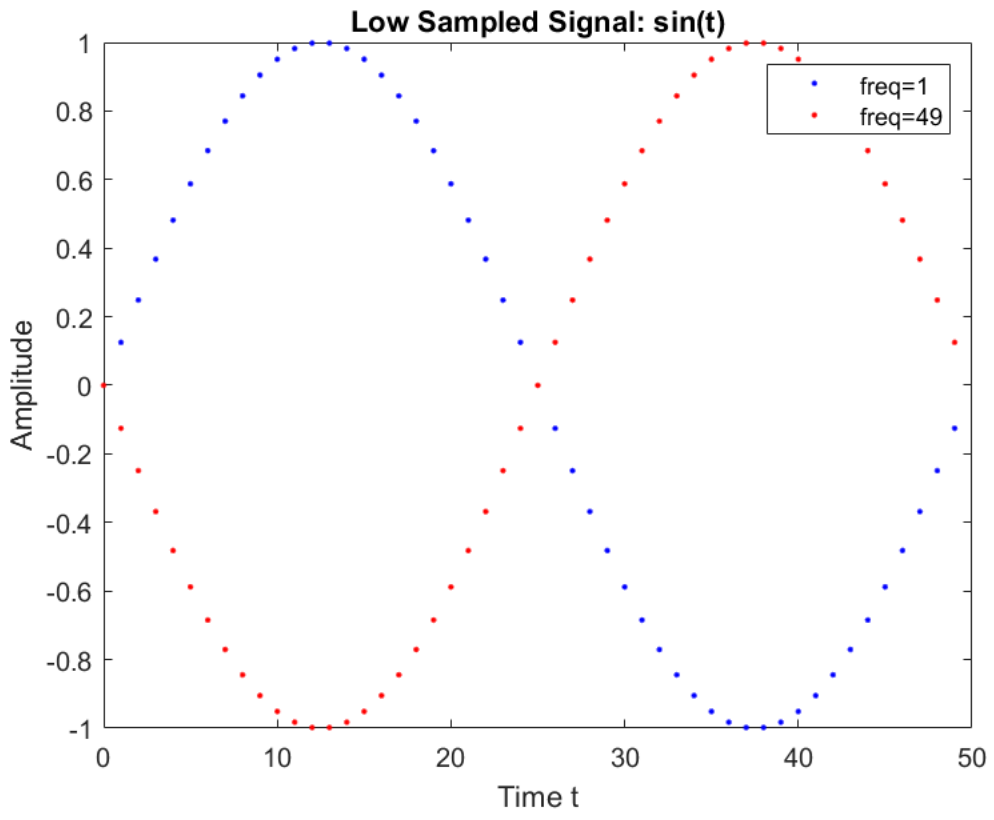
In this lecture we will discuss the composition of signals in time into frequencies using Fourier seriers and Fourier transformations. For example, the sound signal resulting of the note "A" can be either depict as a sinosoidal wave in time, or as a single frequency of 440Hz. In this case, representing the time signal in frequency space has many advantages. Instead of specifying the amplitude at every point in time, a single number of frequency and phase is sufficient, resulting in a large amount of compression of information. A major motivation of studying Fourier transformation is that periodic phenomena and oscillations are very common and that even fundamental physical theories are strongly tied to seeing waves as the fundamental building blocks of nature.

In previous lectures we already covered similar ideas of changing the underlying coordinate system of representation. In case of principal component analysis linear algebra techniques were used to represent data using a specific set of vector basis. The transformation between the original data and the transformed space was a result of a matrix multiplication. In the following we will derive a similar matrix operation that transforms between a vector of discrete points in time and a vector of frequencies.

A generalization of the discrete Fourier series into continues space is the continous Fourier transfrom. Similar to the Laplace transform it uses intergration as the underlying measure and it is a useful technique for solving differential equations.

To get started consider a simple sin() function sampled on 50 points in time for frequencies 1 and 49.

```
N=50;
t=0:1:(N-1);
y1=sin(2*pi*1*t/N);
y49=sin(2*pi*49*t/N);
plot(t,y1,"b.");
hold on;
plot(t,y49,"r.");
title("Low Sampled Signal: sin(t)");
xlabel("Time t");
ylabel("Amplitude");
legend("freq=1","freq=49")
hold off;
```



Note that how in the plot above the higher frequency signal with f=49 looks just like the low frequency signal f=1! This is because the resolution of our time sampling rate puts a limit on the frequencies that we can resolve. The highest frequency due to the sampling called the Nyquest frequency and given by half the sampling rate

$$f_N = \frac{N}{2}$$

At the low sampling rate of just 50 points, the f=49 signal appears to be a phase shifted f=1 signal. The following calculation shows that under the low sampling rate of 50, the high frequency f=49 signal looks like the inverted f=1 signal

$$\sin\left(2\pi\ \frac{49}{50}\right) = \sin\left(2\pi\ \left(\frac{49}{50} - \frac{50}{50}\right)\right) = \sin\left(2\pi\ \left(-\frac{1}{50}\right)\right) = -\sin\left(2\pi\ \frac{1}{50}\right)$$

where we use that the sin() function does not change if adding or subtracting a phase of 2*pi.

In general we have the correspondence between the low and high frequencies:

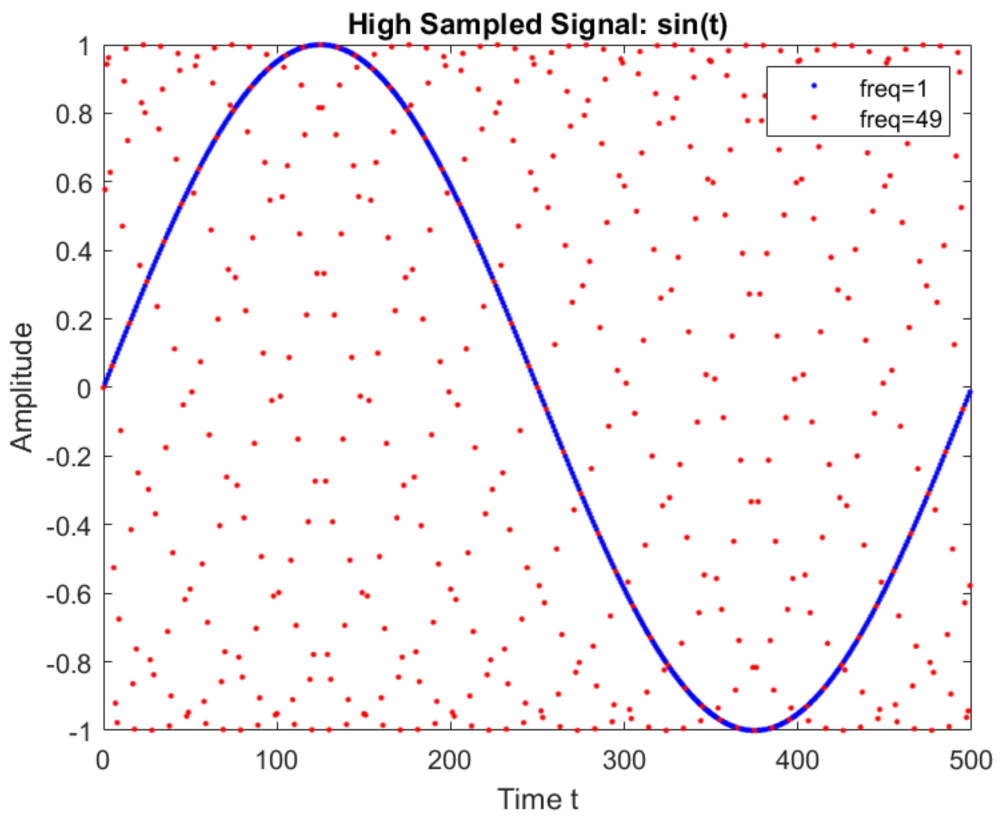
$$\sin\left(2\pi\ \frac{n}{N}\right) = \sin\left(2\pi\ \left(\frac{n}{N} - \frac{N}{N}\right)\right) = -\sin\left(2\pi\ \frac{N-n}{N}\right)$$

$$\cos\left(2\pi\ \frac{n}{N}\right) = \cos\left(2\pi\ \left(\frac{n}{N} - \frac{N}{N}\right)\right) = \cos\left(2\pi\ \frac{N-n}{N}\right)$$

Thus, given a sampling of N points, we can only resolve frequencies up to N/2, as higher frequencies map back into the low frequency range.

A too low sampling rate results in aliasing, which is when a higher frequency signal appears to be of low frequency because is is sampled only a specific time slices. Graphing the same signal at 10 times higher sampling rate shows that the red curve is actually of much higher frequency.

```
t=0:1:499;
y1=sin(2*pi*1*t/500);
y49=sin(2*pi*49*t/500);
plot(t,y1,"b.");
hold on;
plot(t,y49,"r.");
title("High Sampled Signal: sin(t)");
xlabel("Time t");
ylabel("Amplitude");
legend("freq=1","freq=49")
hold off;
```



The discrete Fourier transformation

The above is the framework of the discrete Fourier transformation. Given a signal that is sampled as N amplitude on N time points , it derives N amplitudes for the corresponding N frequencies, where the lower half and upper half of the spectrum being connected due to sampling.

In the following we base our analyis on complex exponentials to represent the periodic functions, using Eulers formula

$$e^{ix} = \cos(x) + i\sin(x)$$

and the Fourier series expansion is defined as

$$x(n) = \frac{1}{N} \sum_{k=0..N-1} u(k) e^{2\pi i \frac{k}{N} n}$$

with time points n in between 0 and N-1 and frequencies k in between 0 and N-1. It connects the N points of the signal in time x(n) to the amplitudes u(k) that correspond to each frequency. In general both the signal and the frequency amplitudes can be complex numbers.

Because of the periodicity of the complex expondential, frequencies in the upper range between N/2 and N-1 correspond to negative frequencies

$$e^{2\pi i \frac{N-k}{N} n} = e^{2\pi i \frac{(-k)}{N} n}$$

and we have a choice to eather sum of frequencies from 0 to N-1 or from -(N-1)/2 to +(N-1)/2

$$\sum_{k=0..N-1} u(k) e^{2\pi i \frac{k}{N} n} = \sum_{k=-\frac{N-1}{2}..\frac{N-1}{2}} u(k) e^{2\pi i \frac{k}{N} n} = u(0) + \sum_{k=0..\frac{N-1}{2}} u(k) e^{2\pi i \frac{k}{N} n} + u(-k) e^{-2\pi i \frac{k}{N} n}$$

$$= \sum_{k=0..\frac{N-1}{2}} u(k) \cos\left(2\pi \frac{k}{N} n\right) + u(-k) \cos\left(-2\pi \frac{k}{N} n\right) + u(k) \sin\left(2\pi \frac{k}{N} n\right) i + u(-k) \sin\left(-2\pi \frac{k}{N} n\right) i$$

$$= \sum_{k=0..\frac{N-1}{2}} (u(k) + u(-k)) \cos\left(2\pi \frac{k}{N} n\right) + i(u(k) - u(-k)) \sin\left(2\pi \frac{k}{N} n\right)$$

For x(t) to be real we need we need corresponing positive and negative frequencies to have equal real and opposing imaginary parts

$$\mathrm{Re}(u(k)) = \mathrm{Re}(u(-k))$$

$$\mathrm{Im}(u(k)) = -\mathrm{Im}(u(-k))$$

Similarly, using the range k=0 to N-1

$$x(n) = \frac{1}{N} \sum_{k=0..N-1} u(k) e^{2\pi i \frac{k}{N} n}$$

we split the sum into pairs of corresponding frequencies

$$x(n) = \frac{1}{N} u(0) + \frac{1}{N} \sum_{k=1..\frac{N}{2}} u(k) e^{2\pi i \frac{k}{N} n} + u(N-k) e^{2\pi i \frac{N-k}{N} n}$$

$$= \frac{1}{N} u(0) + \frac{1}{N} \sum_{k=1..\frac{N}{2}} u(k) e^{2\pi i \frac{k}{N} n} + u(N-k) e^{-2\pi i \frac{k}{N} n}$$

and substituting the Euler formula

$$e^{2\pi \frac{i\,k}{N} n} = \cos\left(2\pi \frac{\mathrm{kn}}{N}\right) + i\sin\left(2\pi \frac{\mathrm{kn}}{N}\right)$$

we get

$$x(n) = \frac{1}{N}u(0) + \frac{1}{N}\sum\nolimits_{k=1\ldots\frac{N}{2}} (u(k) + u(N-k))\cos\left(2\pi \frac{\mathrm{kn}}{N}\right) + i(u(k) + u(N-k))\sin\left(2\pi \frac{\mathrm{kn}}{N}\right)$$

Again in order for x(n) to be real valued, we need the following relation between the lower and upper band frequencies

$$\mathrm{Re}(u(k)) = \mathrm{Re}(u(N-k))$$

$$\mathrm{Im}(u(k)) = -\mathrm{Im}(u(N-k))$$

Constructing a time signal from frequencies is called the inverse Fourier transformation. For N=4 terms we have the following sum for time n=0,1,2,3

$$x(n) = \frac{1}{4}\left(u(0) + u(1)e^{2\pi i \frac{1}{4} n} + u(2)e^{2\pi i \frac{2}{4} n} + u(3)e^{2\pi i \frac{3}{4} n}\right)$$

giving

$$x(0) = \frac{(u(0) + u(1) + u(2) + u(3))}{4}$$

$$x(1) = \frac{u(0) + u(1)e^{2\pi i \frac{1}{4}} + u(2)e^{2\pi i \frac{2}{4}} + u(3)e^{2\pi i \frac{3}{4}}}{4}$$

$$x(2) = \frac{u(0) + u(1)e^{2\pi i \frac{2}{4}} + u(2)e^{2\pi i \frac{4}{4}} + u(3)e^{2\pi i \frac{6}{4}}}{4}$$

$$x(3) = \frac{u(0) + u(1)e^{2\pi i \frac{3}{4}} + u(2)e^{2\pi i \frac{6}{4}} + u(3)e^{2\pi i \frac{9}{4}}}{4}$$

or as matrix equation

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{2\pi i \frac{1}{4}} & e^{2\pi i \frac{2}{4}} & e^{2\pi i \frac{3}{4}} \\ 1 & e^{2\pi i \frac{2}{4}} & e^{2\pi i \frac{4}{4}} & e^{2\pi i \frac{6}{4}} \\ 1 & e^{2\pi i \frac{3}{4}} & e^{2\pi i \frac{6}{4}} & e^{2\pi i \frac{9}{4}} \end{bmatrix}\begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ u(3) \end{bmatrix}$$

or

$$x = M_{\mathrm{inv}}\,u$$

with

$$M_{\text{inv}} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{2\pi i\frac{1}{4}} & e^{2\pi i\frac{2}{4}} & e^{2\pi i\frac{3}{4}} \\ 1 & e^{2\pi i\frac{2}{4}} & e^{2\pi i\frac{4}{4}} & e^{2\pi i\frac{6}{4}} \\ 1 & e^{2\pi i\frac{3}{4}} & e^{2\pi i\frac{6}{4}} & e^{2\pi i\frac{9}{4}} \end{bmatrix}$$

Example: given frequencies u = [1; 1+i; 2; 1] compute the sampled signal in time x.

```
u = [1; 1+i; 2; 1;];
Minvft = exp(2*pi*i/4.*[0 1 2 3]'*[0 1 2 3])/4;
x = Minvft*u
```

```
x = 4×1 complex
   1.2500 + 0.2500i
  -0.5000 + 0.0000i
   0.2500 - 0.2500i
   0.0000 + 0.0000i
```

TASK:

1. Compare the ouput with the Matlab build-in function for the discrete inverse Fourier transformation ifft() applied to the frequency vector u above. Do you get the same result?

2. The result xfor the frequency vector u = [1 1+i 2 1] is complex. For a vector of the form

u = [1;1+i;2;u(3)], what should u(3) be to generate a vector x that has real numbers only? Verify your choice of u(3) with the ifft(u) function.

The Fourier Transformation

From x(time) to u(frequency), inverting the inverse

The Fourier transformation that given a sampled signal in time x(n) gives the corresponding frequencies u(k) is given by

$$u(k) = \sum_{n=0..N-1} x(n)e^{-2\pi i\,\frac{k}{N}\,n}$$

In matrix form transformation reads

$$u = M_{\text{inv}}\,x$$

or in case of just 3 points

$$\begin{bmatrix} u(0) \\ u(1) \\ u(2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-2\pi i\frac{1}{3}} & e^{-2\pi i\frac{2}{3}} \\ 1 & e^{-2\pi i\frac{2}{3}} & e^{-2\pi i\frac{4}{3}} \end{bmatrix}\begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix}$$

with

$$M_{\mathrm{ft}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-2\pi i\frac{1}{3}} & e^{-2\pi i\frac{2}{3}} \\ 1 & e^{-2\pi i\frac{2}{3}} & e^{-2\pi i\frac{4}{3}} \end{bmatrix}$$

This definition of the Fourier transformation is indeed inverse to the inverse. To illustrate consider the product

$$M_{\mathrm{inv}} * M_{\mathrm{fft}} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{2\pi i\frac{1}{3}} & e^{2\pi i\frac{2}{3}} \\ 1 & e^{2\pi i\frac{2}{3}} & e^{2\pi i\frac{4}{3}} \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-2\pi i\frac{1}{3}} & e^{-2\pi i\frac{2}{3}} \\ 1 & e^{-2\pi i\frac{2}{3}} & e^{-2\pi i\frac{4}{3}} \end{bmatrix} =$$

$$\frac{1}{3} \begin{bmatrix} 1+1+1 & 1+e^{-2\pi\frac{1}{3}}+e^{-2\pi\frac{2}{3}} & 1+e^{-2\pi i\frac{2}{3}}+e^{-2\pi i\frac{4}{3}} \\ 1+e^{2\pi i\frac{1}{3}}+e^{2\pi i\frac{2}{3}} & 1+e^{2\pi i\frac{1}{3}-2\pi i\frac{1}{3}}+e^{2\pi i\frac{2}{3}-2\pi i\frac{2}{3}} & 1+e^{2\pi i\frac{1}{3}-2\pi i\frac{2}{3}}+e^{2\pi i\frac{2}{3}-2\pi i\frac{4}{3}} \\ 1+e^{2\pi i\frac{2}{3}}+e^{2\pi i\frac{4}{3}} & 1+e^{2\pi i\frac{2}{3}-2\pi i\frac{1}{3}}+e^{2\pi i\frac{4}{3}-2\pi i\frac{2}{3}} & 1+e^{2\pi i\frac{2}{3}-2\pi i\frac{2}{3}}+e^{2\pi i\frac{4}{3}-2\pi i\frac{4}{3}} \end{bmatrix}$$

The off-diagonal terms are complex sum of complex roots and give zero while the diagonal terms result to ones.

$$M_{\mathrm{inv}} * M_{\mathrm{fft}} = \frac{1}{3} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We used that the sum over the complete set of complex roots of the unit circle gives zero

$$e^{2\pi\frac{0}{3}} + e^{2\pi\frac{2}{3}} + e^{2\pi\frac{4}{3}} = e^{2\pi\frac{0}{3}} + e^{2\pi\frac{1}{3}} + e^{2\pi\frac{2}{3}} = 0$$

This follows from the geometric sum formula

$$s = r^0 + r^1 + r^2 + .. + r^{n-1}$$

$$\mathrm{rs} = r^1 + r^2 + .. + r^{n-1} + r^n = s - r^0 + r^n$$

$$s = \frac{r^n - 1}{r - 1}$$

Now with $r^n = e^{2\pi i} = 1$ and $r = e^{2\pi\frac{i}{n}}$ we get $s = 0$

In general, applying the inverse to the transformation u(k) gives

$$x(n) = \frac{1}{N} \sum_{k=0..N-1} u(k) e^{2\pi i\frac{n}{N}k} =$$

substituing the Fourier transformation of x(m) for u(k)

$$\frac{1}{N} \sum_{k=0..N-1} \sum_{m=0..N-1} x(m) e^{-2\pi i\frac{k}{N}m} e^{2\pi i\frac{k}{N}n} =$$

and summing first over k for each fixed m gives

$$= \frac{1}{N} \sum_{m=0..N-1} x(m) \sum_{k=0..N-1} e^{2\pi i \frac{k}{N}(n-m)} =$$

The inners sum is zero for n not equal to m, since its sums over a complete set of complex root. However for n=m the exponential is 1 and the sum gives N. Therefore we recover x(n)

$$\frac{1}{N} \sum_{m=0..N-1} x(m) \sum_{k=0..N-1} 1_{(n=m)} = \frac{1}{N} x(n) N = x(n)$$

This shows the the Fourier transformations as given above are indeed inverse to each other.

TASK:

Following the example above of a four point inverse Fourier transformation matrix

Minvft = exp(2*pi*i/4.*[0 1 2 3]'*[0 1 2 3])/4;

1. Generate the equivalent 6 point inverse Fourier transformation matrix M6invft
2. Following the discussion above, generate the Fourier transformation matrix M6ft, (the inverse of the inverse).
3. Use Matlab to calculate M6ft*M6invft. What result are you expecting?
4. The Matlab function for the descrete Fourier transformtion is fft(). Compare applying both M6ft and fft to the vector x=[1;2;3;3;2;1] . Confirm that the results agree.

The FAST Fourier Transform

The Fourier Transformation has many uses in computation and signal processing. Computing the Fourier transformation as discussed above requires a Matrix multiplication, which is computationally expensive. One full Matrix multiplication with a NxN matrix requires in the order of N^2 computations, because for each vector product with a row requires a summation of N products of elements and there are N or such rows. Even if each individual calculation is extremely fast, scaling N up to more points increases the computational time rapidely (quadratically).

The Fast Fourier Transform algorithm improves on the computational speed by recursively dividing the original problem into subproblems of half the size. Below we illustrate for example how the Fourier tranformation with a 4x4 matrix can be split into 2 calculations over a 2x2 matrix.

Succesive application of halfing the size N problem can be done Log(N) times. Each step still involes a total of N points. Such the total runtime scales with N*Log(N), which is an improvement over N^2

Splitting the transform into even and odd points

$$\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-2\pi i \frac{1}{4}} & e^{-2\pi i \frac{2}{4}} & e^{-2\pi i \frac{3}{4}} \\ 1 & e^{-2\pi i \frac{2}{4}} & e^{-2\pi i \frac{4}{4}} & e^{-2\pi i \frac{6}{4}} \\ 1 & e^{-2\pi i \frac{3}{4}} & e^{-2\pi i \frac{6}{4}} & e^{-2\pi i \frac{9}{4}} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-2\pi i \frac{1}{4}} & e^{-2\pi i \frac{2}{4}} & e^{-2\pi i \frac{3}{4}} \\ 1 & e^{-2\pi i \frac{2}{4}} & e^{-2\pi i \frac{4}{4}} & e^{-2\pi i \frac{2}{4}} \\ 1 & e^{-2\pi i \frac{3}{4}} & e^{-2\pi i \frac{2}{4}} & e^{-2\pi i \frac{1}{4}} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & e^{-2\pi i \frac{2}{4}} & 0 \\ 1 & 0 & e^{-2\pi i \frac{4}{4}} & 0 \\ 1 & 0 & e^{-2\pi i \frac{6}{4}} & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ 0 \\ x_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & e^{-2\pi i \frac{1}{4}} & 0 & e^{-2\pi i \frac{3}{4}} \\ 0 & e^{-2\pi i \frac{2}{4}} & 0 & e^{-2\pi i \frac{6}{4}} \\ 0 & e^{-2\pi i \frac{3}{4}} & 0 & e^{-2\pi i \frac{9}{4}} \end{bmatrix} \begin{bmatrix} 0 \\ x_1 \\ 0 \\ x_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & e^{-2\pi i \frac{2}{4}} & 0 \\ 1 & 0 & e^{-2\pi i \frac{4}{4}} & 0 \\ 1 & 0 & e^{-2\pi i \frac{6}{4}} & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ 0 \\ x_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-2\pi i \frac{1}{4}} & 0 & 0 \\ 0 & 0 & e^{-2\pi i \frac{2}{4}} & 0 \\ 0 & 0 & 0 & e^{-2\pi i \frac{3}{4}} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & e^{-2\pi i \frac{2}{4}} \\ 0 & 1 & 0 & e^{-2\pi i \frac{4}{4}} \\ 0 & 1 & 0 & e^{-2\pi i \frac{6}{4}} \end{bmatrix} \begin{bmatrix} 0 \\ x_1 \\ 0 \\ x_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & e^{-2\pi i \frac{1}{2}} & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & e^{-2\pi i \frac{1}{2}} & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ 0 \\ x_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-2\pi i \frac{1}{4}} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -e^{-2\pi i \frac{1}{4}} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & e^{-2\pi i \frac{1}{2}} \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & e^{-2\pi i \frac{1}{2}} \end{bmatrix} \begin{bmatrix} 0 \\ x_1 \\ 0 \\ x_3 \end{bmatrix}$$

Such the single reduction step results in a problem involing two applications of the 2x2 Fourier transformation matrix over the even [x0 x2] and odd point vectors [x1 x3] to yield the lower half [u0 u1] and upper half [u2 u3] of the frequency spectrum.

$$\begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & e^{-2\pi i \frac{1}{2}} \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i \frac{1}{4}} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & e^{-2\pi i \frac{1}{2}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & e^{-2\pi i \frac{1}{2}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i \frac{1}{4}} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & e^{-2\pi i \frac{1}{2}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \text{u}$$

Task

Explore the how the size of the data vector x affects the runtime of the Fourier transformation.

Matlab provides functionalilty to measure the execution time of a function with

time = timeit(function)

For example the code below measures how long it takes to compute a Fourier matrix * vector multiplication with 3000 random data points.

Use the example as a template to measure the time it takes for a computation involving an inceasing number of points, for example 3000, 4000, 5000 ... . Depending on the particular speed and memory size of the computer actual working parameters that give meaningful measurements (not too fast, not too slow) will vary. Plot at least 5 data-size vs time-taken points. What kind of scaling do you get?

```
M1=exp(-2*pi*i/3000.*(0:1:2999)'*(0:1:2999))/3000;
x1=rand(3000,1);

fm1 = @() M1*x1;

tm1=timeit(fm1)
```

```
tm1 = 0.0088
```

Task

Apply a 1000 points Fourier transform to the sampling of the step function

x(n) = 1 for n <= 100

x(n) = 0 for n > 100

Plot x(n) and the transform (spectrum) of u(n) = fft(x(n)).