- We start with a data stream \( x_0, x_1, \ldots, x_N \) of real numbers (sound file, Dow-Jones every minute, etc).
- We are interested in what frequencies or periodicities are present in the data stream.
- If the data stream was given as a continuous function \( f(t) \), we could find the Fourier Series as in the last lecture and use the coefficients as the amplitudes of each harmonic.
- But we have a discrete data set and so we need a discrete Fourier basis.
- Treat the data as a vector (column)
  \[
  \mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]^T
  \]
  (Notice indexing: start at time zero)
For simplicity we restrict the sample times to the interval $[0, 1]$ and assume we have sampled at uniform intervals and have $N$ samples starting with $0$.

- So the sample times are $0, \frac{1}{N}, \frac{2}{N}, \ldots$ or at times $t_j = j/N$ for $j = 0, \ldots, N-1$.

On $[0, 1]$ we transform the Fourier basis (ignoring normalization) to

$$ y_k = e^{2\pi i k t} = \cos(2\pi k t) + i \sin(2\pi k t) $$

Just looking at the real part $k = 1$

$$ k = 1 $$

Just looking at the real part $k = 2$

$$ k = 2 $$
Now we sample the $\Psi_k(t)$ at the times $t_j$ to get vectors

$$
\Psi_k = \left[ \Psi_k(t_0), \Psi_k(t_1), \ldots, \Psi_k(t_{N-1}) \right]^T
$$

$$
= \left[ e^{2\pi i k t_0}, e^{2\pi i k t_1}, \ldots, e^{2\pi i k t_{N-1}} \right]^T
$$

$$
= \left[ e^{2\pi i k 0/N}, e^{2\pi i k 1/N}, \ldots, e^{2\pi i k (N-1)/N} \right]^T
$$

$$
= \left[ (e^{2\pi i k 0/N}), (e^{2\pi i k 1/N}), \ldots, (e^{2\pi i k (N-1)/N}) \right]^T
$$

$$
= \left[ (W_N)^{k.0}, (W_N)^{k.1}, \ldots, (W_N)^{k(N-1)} \right]^T
$$

where $W_N = e^{2\pi i/N}$, the $N$-root of unity.
More succinctly, the $j$th component of $\vec{\Phi}_k$ is

$$ (\vec{\Phi}_k)_j = \omega_N^{kj} $$

It turns out that the $\vec{\Phi}_k$s are orthogonal but not orthonormal, so we define for $k = 0, \ldots, N-1$

$$ \vec{Z}_k = \frac{1}{\sqrt{N}} \left[ 1, \omega_N^k, \omega_N^{2k}, \ldots, \omega_N^{(N-1)k} \right]^T $$

**Theorem (Hall):** The $\sum \vec{Z}_k$s form an orthonormal basis for $\mathbb{C}^N$

**Examples:** For any $N$,

$$ \vec{Z}_0 = \frac{1}{\sqrt{N}} \left[ 1, 1, \ldots, 1 \right]^T $$
\[ N = 4 \]

\[ w = w_4 = e^{\frac{2\pi i}{4}} = e^{\frac{\pi i}{2}} = i \]

\[ \mathbf{Z}_3 = \frac{1}{2} \begin{bmatrix} 1, w^3, w^6, w^9 \end{bmatrix}^T \]

\[ = \frac{1}{2} \begin{bmatrix} 1, w^3, w^2, w^1 \end{bmatrix}^T \]

**Basic Fact:** If \( k = 4j + \ell \), then \( w^k = w_4^{4j+\ell} \).

\[ w_6^0 = w = e^{\frac{2\pi i}{6}} = e^{\frac{\pi i}{3}} \]

\[ \mathbf{Z}_5 = \frac{1}{\sqrt{5}} \begin{bmatrix} 1, w^5, w, w^5, w^{10}, w^{15}, w^{20}, w^{25} \end{bmatrix}^T \]

\[ = \frac{1}{\sqrt{5}} \begin{bmatrix} 1, w^5, w^4, w^3, w^2, w \end{bmatrix}^T \]
How do we change our data vector \( \mathbf{x} \) into the Discrete Fourier Basis \( \mathbf{\tilde{\mathbf{x}}} \)?

Recall our basis formula when \( \mathbf{\tilde{\mathbf{x}}} = \sum_{k=0}^{2n-3} 2 \mathbf{\phi}_k \mathbf{\tilde{\mathbf{x}}} \) is an orthonormal basis.
If \( \vec{v} = \alpha_0 \vec{z}_0 + \cdots + \alpha_{n-1} \vec{z}_{n-1} \)

Then \( \alpha_k = \langle \vec{z}_k, \vec{v} \rangle = \vec{z}_k^T \vec{v} \)

Now making the coordinates or amplitudes \( \alpha_k \) into a vector \( \vec{a} = [\alpha_0, \alpha_1, \ldots, \alpha_{n-1}]^T \)

Then

\[
\begin{bmatrix}
\vec{z}_0^T \\
\vec{z}_1^T \\
\vec{z}_2^T \\
\vdots \\
\vec{z}_{n-1}^T
\end{bmatrix}
\begin{bmatrix}
\vec{a} \\
\vec{v}
\end{bmatrix}
= \vec{v}
\]

This matrix is the Fourier matrix \( \mathbf{F} \).
The $k^{th}$ row of $F$ is

$$Z_k = \frac{1}{\sqrt{N}} \left[ 1, \omega_k, \omega_k^2, \ldots, \omega_k^{(N-1)k} \right]$$

$$= \frac{1}{\sqrt{N}} \left[ 1, \omega_k^{-1}, \omega_k^{-2}, \ldots, \omega_k^{-(N-1)k} \right]$$

So we see the entries of $F$ are

$$F_{ij} = \frac{\omega_k^{-ij}}{\sqrt{N}} \quad \text{where} \quad i=0, \ldots, N-1 \quad j=0, \ldots, N-1$$

**Note:** $F_{ji} = F_{ij}$ so $F^T = F$, symmetric.

Also let's look at $Q = F^* F$, the adjoint.
\[ Q = F^* = \begin{bmatrix} \hat{z}_0 & \cdots & \hat{z}_{n-1} \end{bmatrix} \text{ has its columns}

\text{an orthonormal basis, so } F^* \text{ is unitary}

\text{or } Q \text{ is unitary or } Q^* = Q^{-1}

\text{So } (F^*)^* = (F^*)^{-1} \text{ or } F = (F^*)^{-1}

\text{or } F^{-1} = F^* \text{ so } F \text{ is unitary also.}

\text{Example: } N = 4, \quad w_4 = e^{\frac{2\pi i}{4}} = e^{\frac{\pi i}{2}} = i
- It is also useful to express this in coordinates.

- We now switch to more standard notation in the field: If \( x \) is the data (not a vector), then its Discrete Fourier transform is

\[
\hat{x} = Fx = \text{DFT}(x)
\]

In coordinates,

\[
\hat{x}_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k W_N^{-kj}
\]

\[
= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-2\pi i k j / N}
\]

For \( j = 0 \ldots N-1 \).

Also, sometimes \( \text{DFT}(x) = \hat{x} \), capital letters.
What about going back from \( \text{IDFT} \) to \( x \)?

(This is called going from frequency domain to the time domain)

Since \( \hat{x} = Fx \), \( x = F\hat{x} = F^{*}\hat{x} = \Omega \hat{x} \)

\( F \) is unitary

So for \( j = 0, \ldots, n-1 \)

\[
X_j = \frac{1}{n} \sum_{k=0}^{n-1} \hat{X}_k W_n^{kj}, \quad X = \text{IDFT}(\hat{x})
\]

Notice similarity to forward transform.
There are many other considerations and conventions.

(1) In practice, \( \hat{X} = Fx \) is not computed by matrix multiplication, but rather by a very clever, much faster Fast Fourier Transform.

(2) While the \( \frac{1}{\sqrt{N}} \) is nice in both DFT and IDFT for symmetry and linear algebra (or monomial basis), it is usually not in practice. A single \( \frac{1}{\sqrt{N}} \) is put on the DFT or on the IDFT.

(3) Matlab uses index 0, ... , \( N-1 \) and puts \( \frac{1}{N} \) on the DFT.

\[
\begin{align*}
\mathbf{X}_j &= \sum_{k=0}^{N-1} x_k W^{-k(j-1)} \\
x_j &= \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{X}_k W^{(k-1)(j-1)}
\end{align*}
\]

\( W = e^{2 \pi i / N} \)