TOPIC 4
The practical computation of Fourier (e.g., with Matlab)
PART 3

Fast Fourier Transform (FFT)

- JUST a FAST WAY of computing DFT
- FFT is faster since takes advantage of the structure of the matrix of DFT
- MATLAB (or Octave) USES FFT !

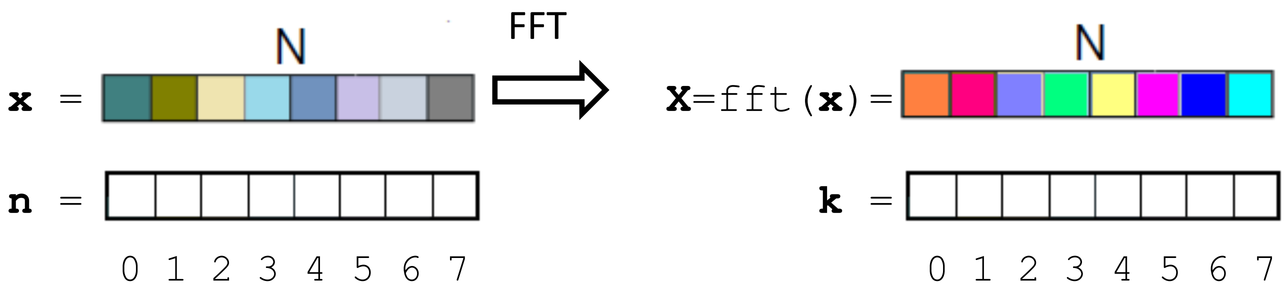
FFT = DFT but faster

With Matlab: fft(x,N) and ifft(x,N)

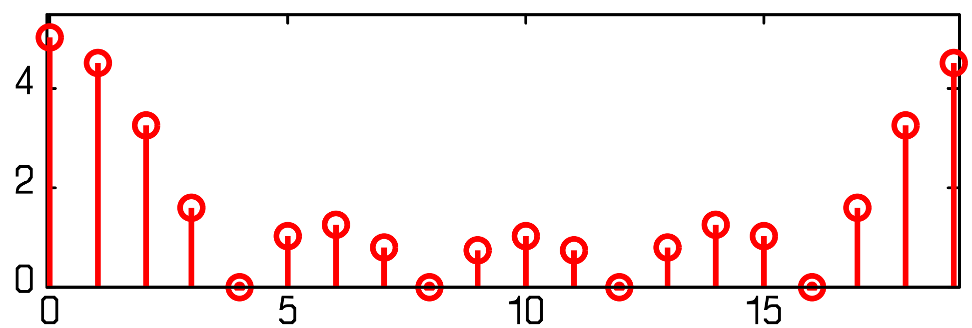
X=fft(x,N) is the N-point FFT of x (i.e., it is computing the DFT), padded with zeros if x has less than N points and truncated if it has more.

x=ifft(X,N) is the inverse discrete Fourier transform (Inverse DFT) of X.

With Matlab (or Octave)

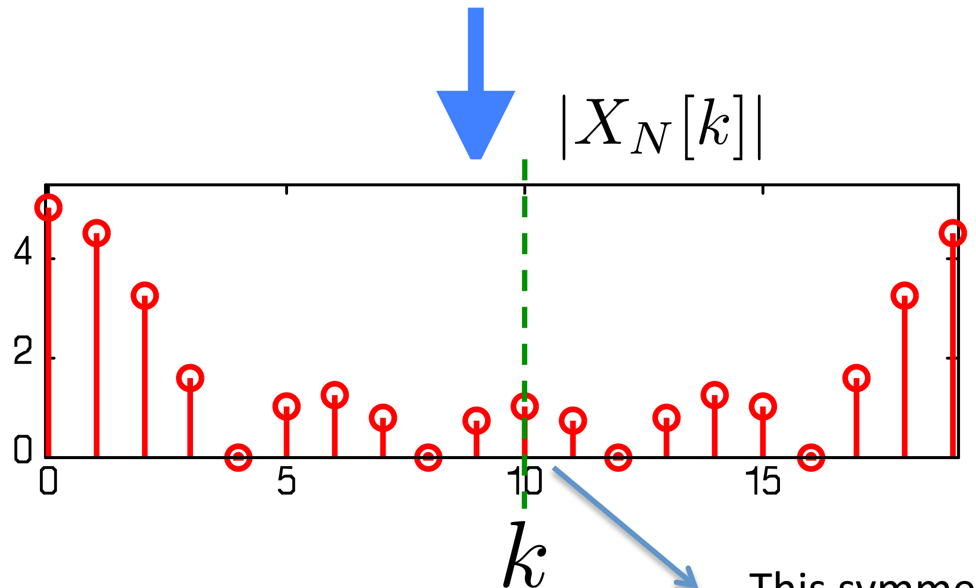



```
N=20;
x=[1 1 1 1 1 zeros(1,15)];
X=fft(x,N);
k=0:(N-1);
figure;
stem(k,abs(X));
```



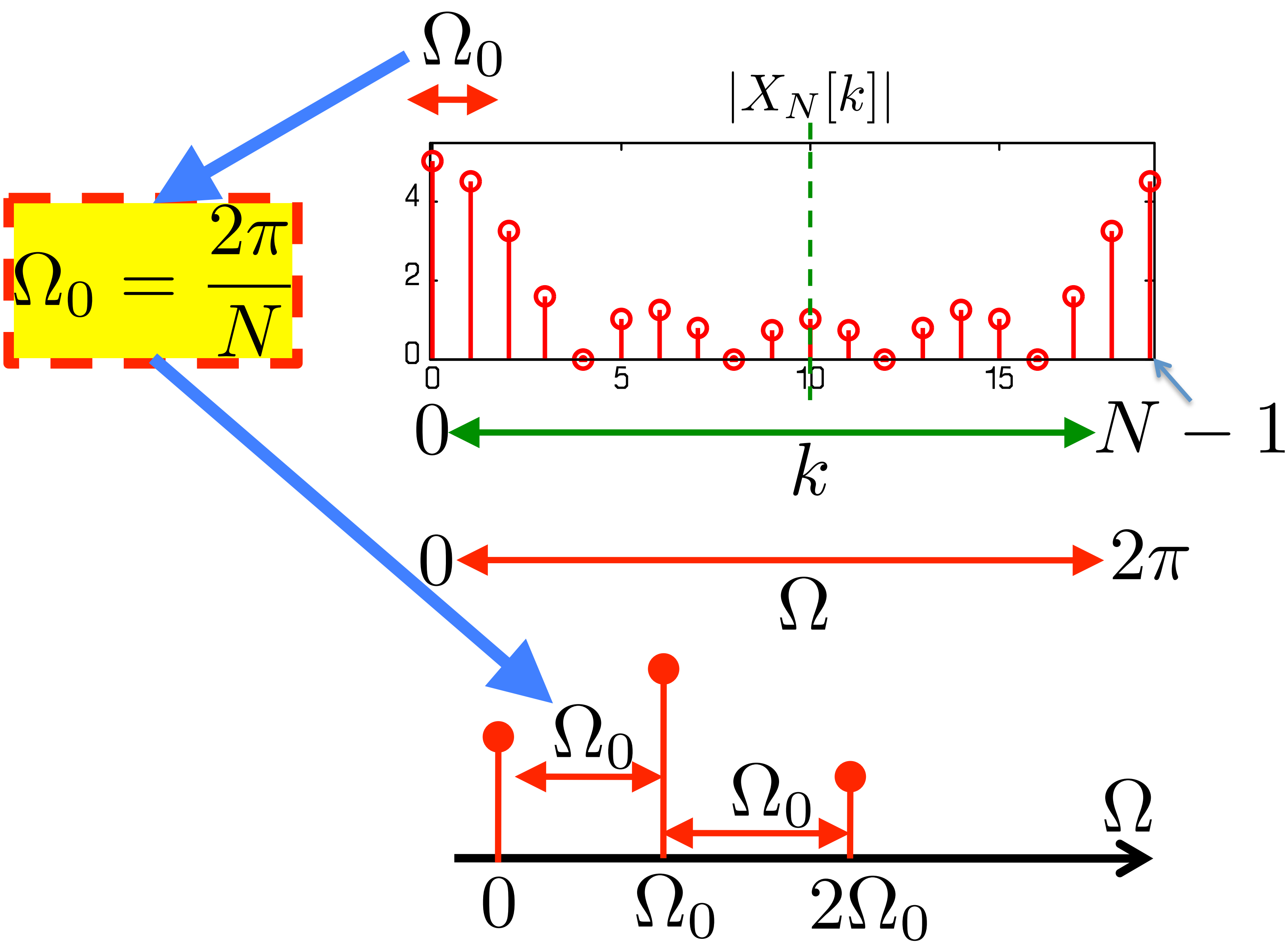
With Matlab (or Octave)

```
N=20;
x=[1 1 1 1 1 zeros(1,15)];
X=fft(x,N);
k=0:(N-1);
figure;
stem(k,abs(X));
```

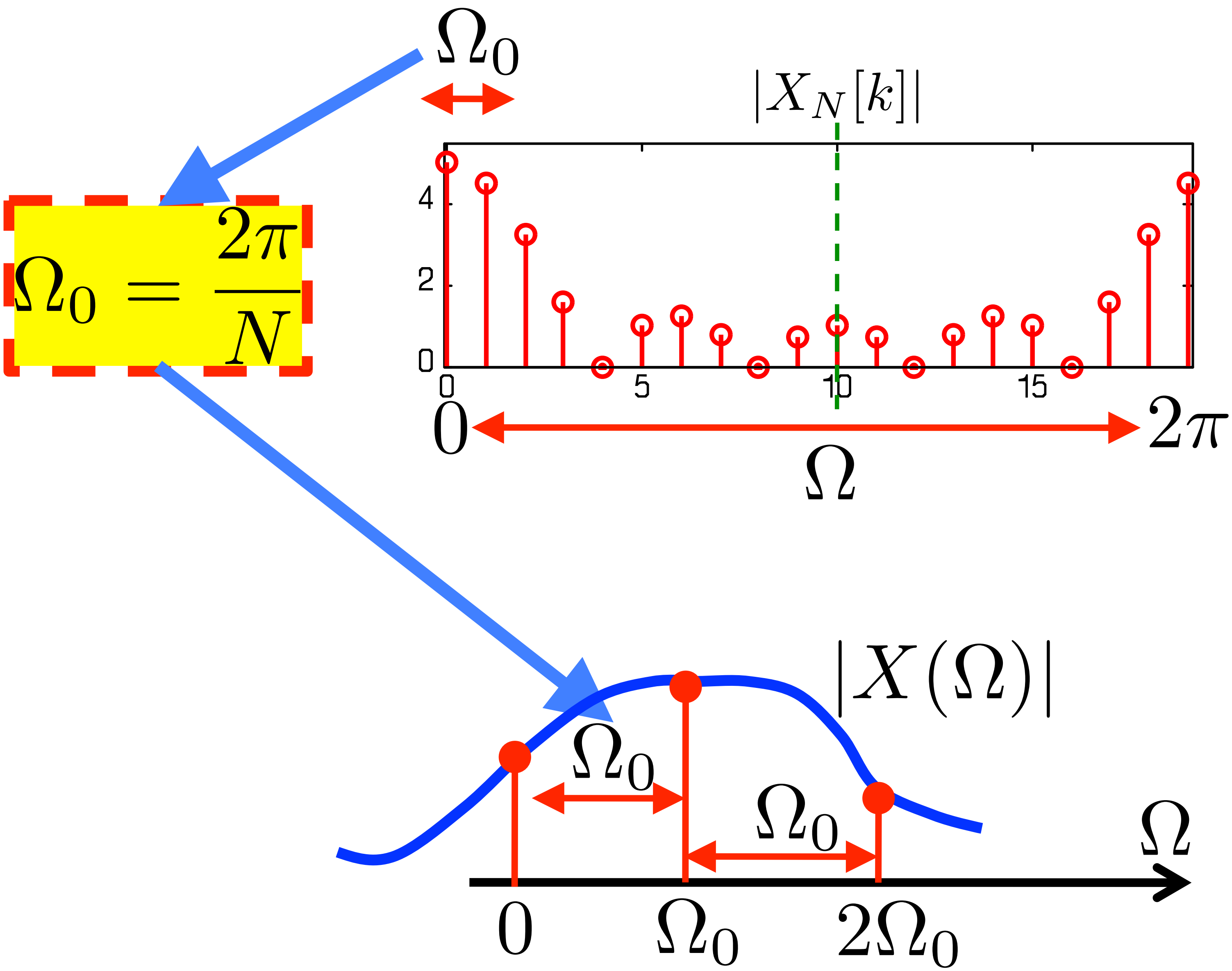


This symmetry is due to the fact that x[n] is real…

IMP: Interpreting the FFT output



IMP: Interpreting the FFT output



Recall this slide…

Matlab returns N complex numbers:

$$X_N[0], X_N[1], X_N[2] \ldots, X_N[N-1]$$

$$0 \qquad \Omega_0 \qquad 2\Omega_0 \qquad (N-1)\Omega_0$$

$$\Omega_0 = \frac{2\pi}{N}$$

$$X_N[k] = X(\Omega)\Big|_{k\Omega_0} = X(k\Omega_0)$$

$$X_N[k] = X(\Omega)\Big|_{k\frac{2\pi}{N}} = X\left(k\frac{2\pi}{N}\right)$$

Recall this slide…

$$X_N[k] = X(\Omega)\Big|_{k\frac{2\pi}{N}} = X\left(k\frac{2\pi}{N}\right)$$

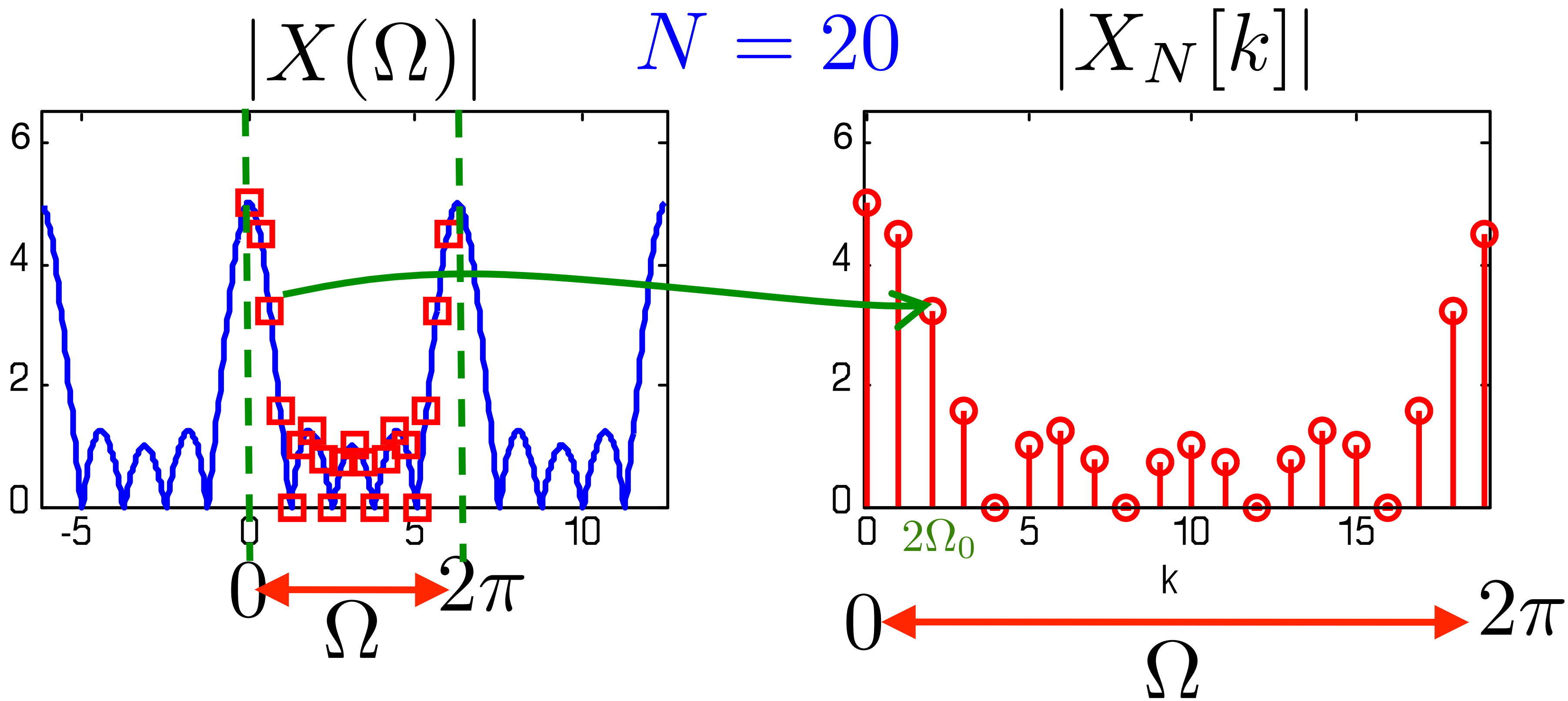
$$X_N[k] = X(\Omega)\Big|_{k\Omega_0} = X(k\Omega_0)$$

$$\Omega_0 = \frac{2\pi}{N}$$

$$x[n] \longleftrightarrow X(\Omega)$$

WE ARE "SAMPLING" the FOURIER TRANSFORM of x[n] in frequency domain.

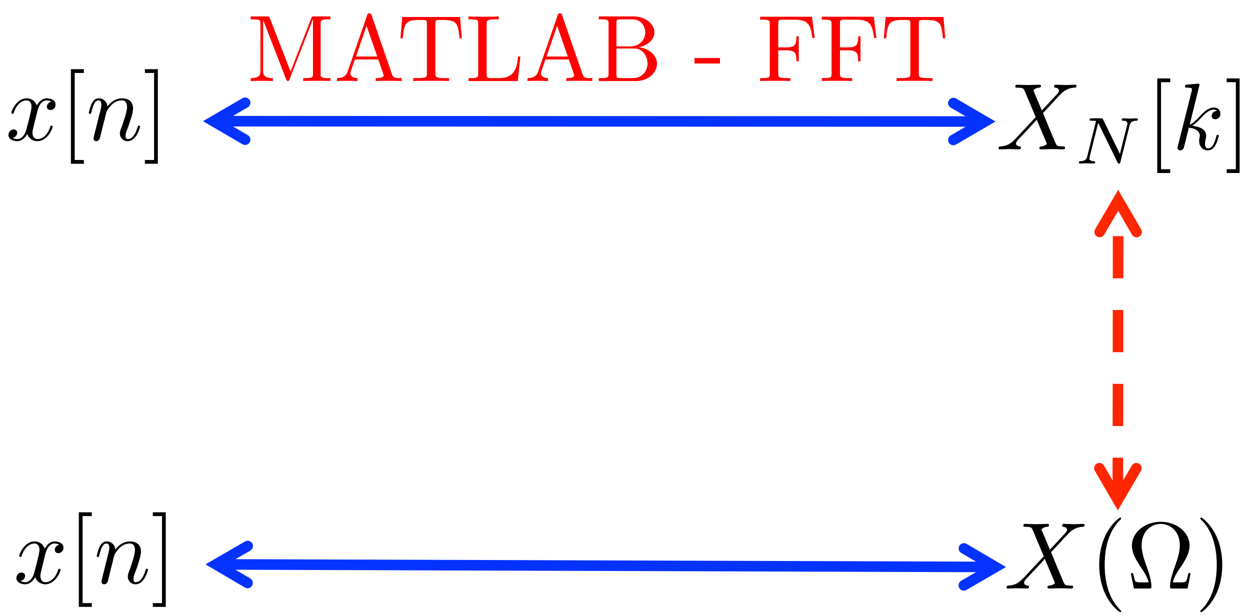
Another slide: for the same thing...



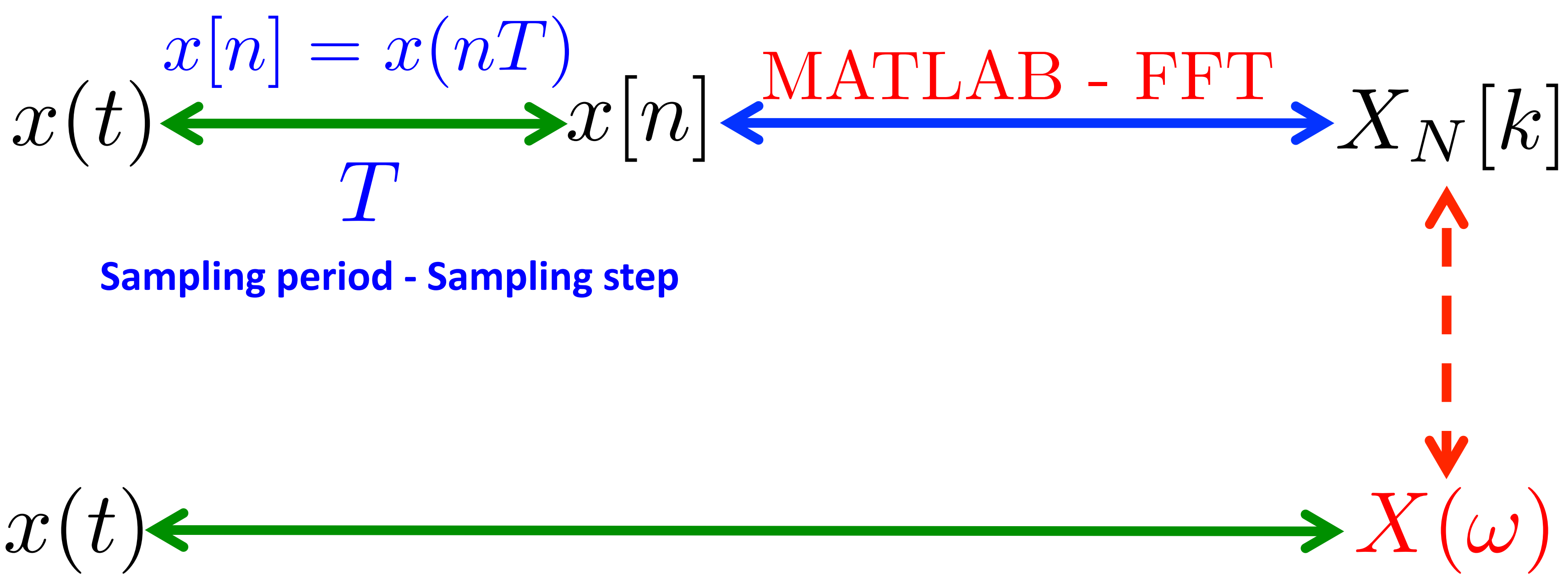
$$X_N[k] = X(\Omega)\Big|_{k\Omega_0} = X(k\Omega_0)$$

So far…

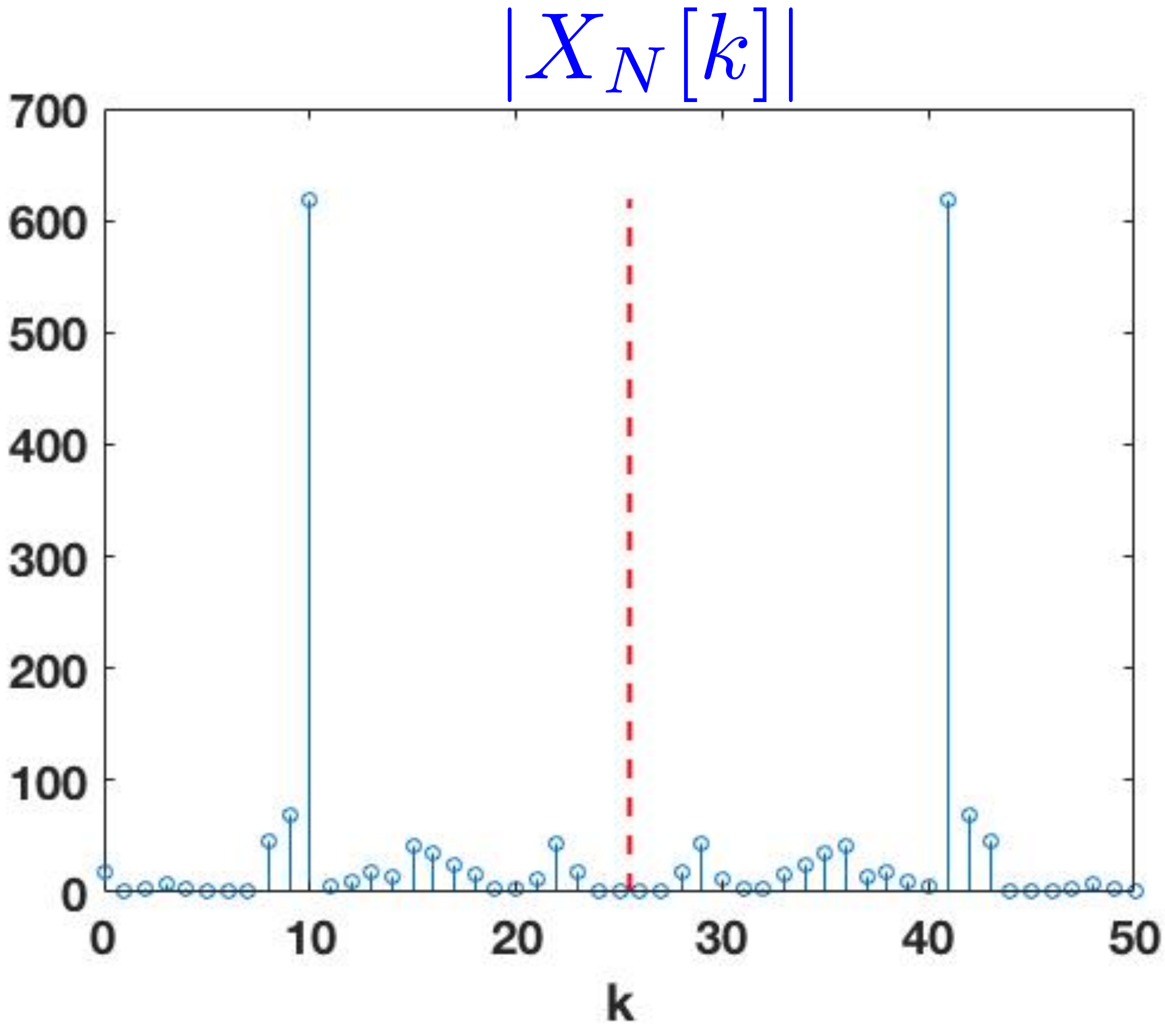
So far, we have just "repeated" several times the same thing… And we have seen/studied the relationship:



NOW: if x[n] is a sampled signal from x(t)?

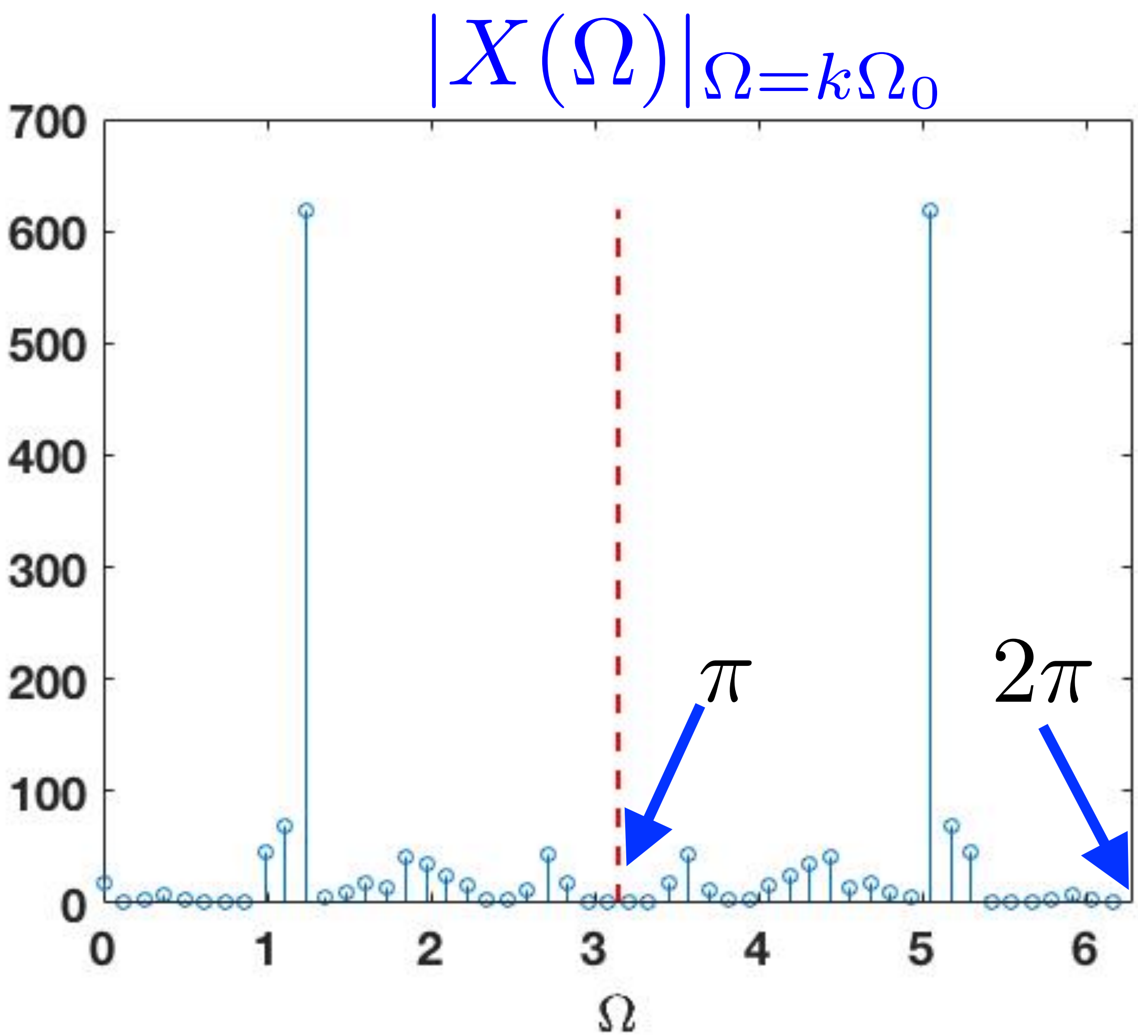


EXAMPLE; DFT=FFT (Matlab)



$N = 51$

FFT en Matlab: interpreting as FT of x[n]



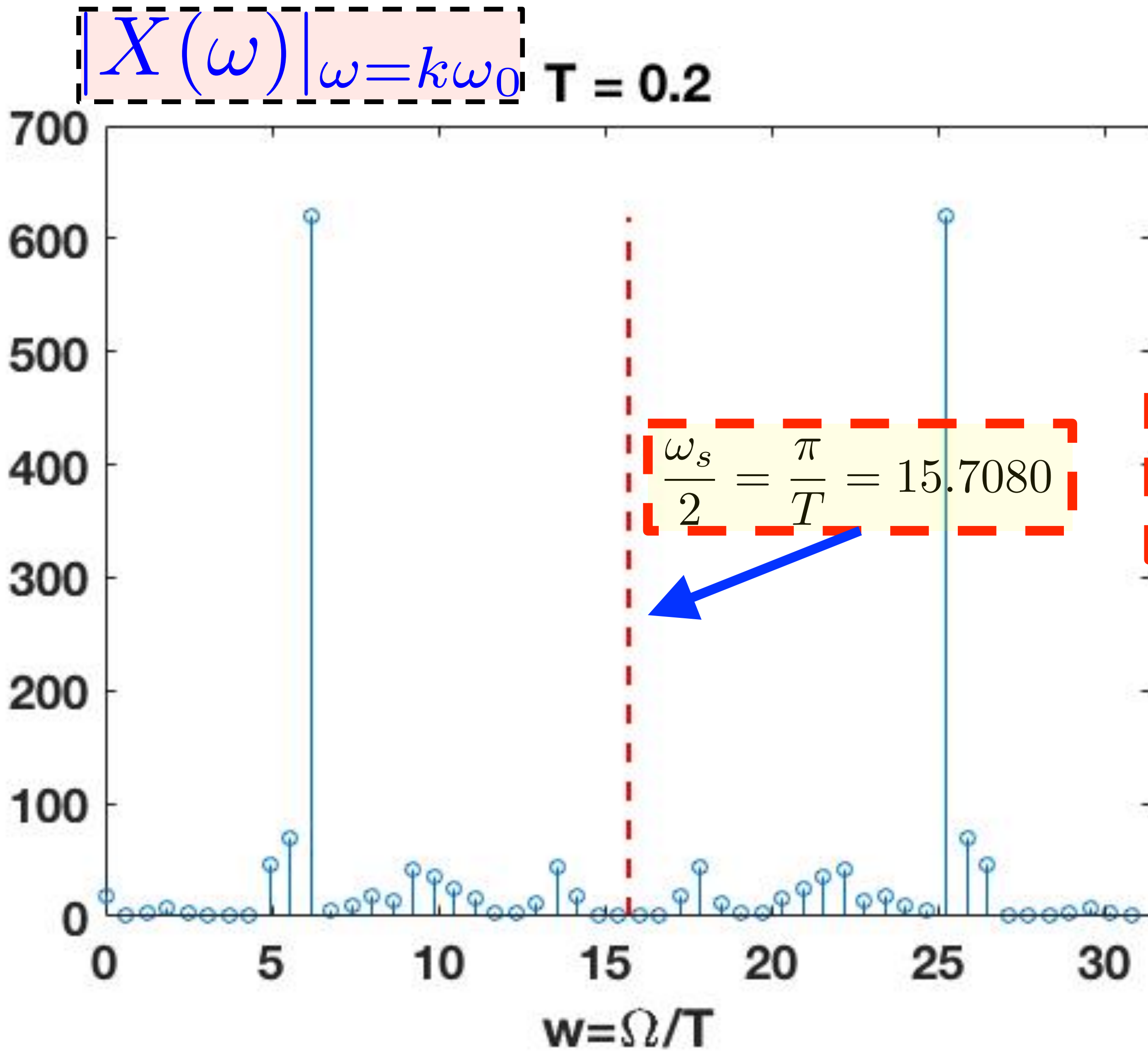
$N = 51$

BETWEEN TWO POINTS:

$$\Omega_0 = \frac{2\pi}{N} = 0.1232$$

FFT en Matlab: interpreting as FT of x(t)

$N = 51$

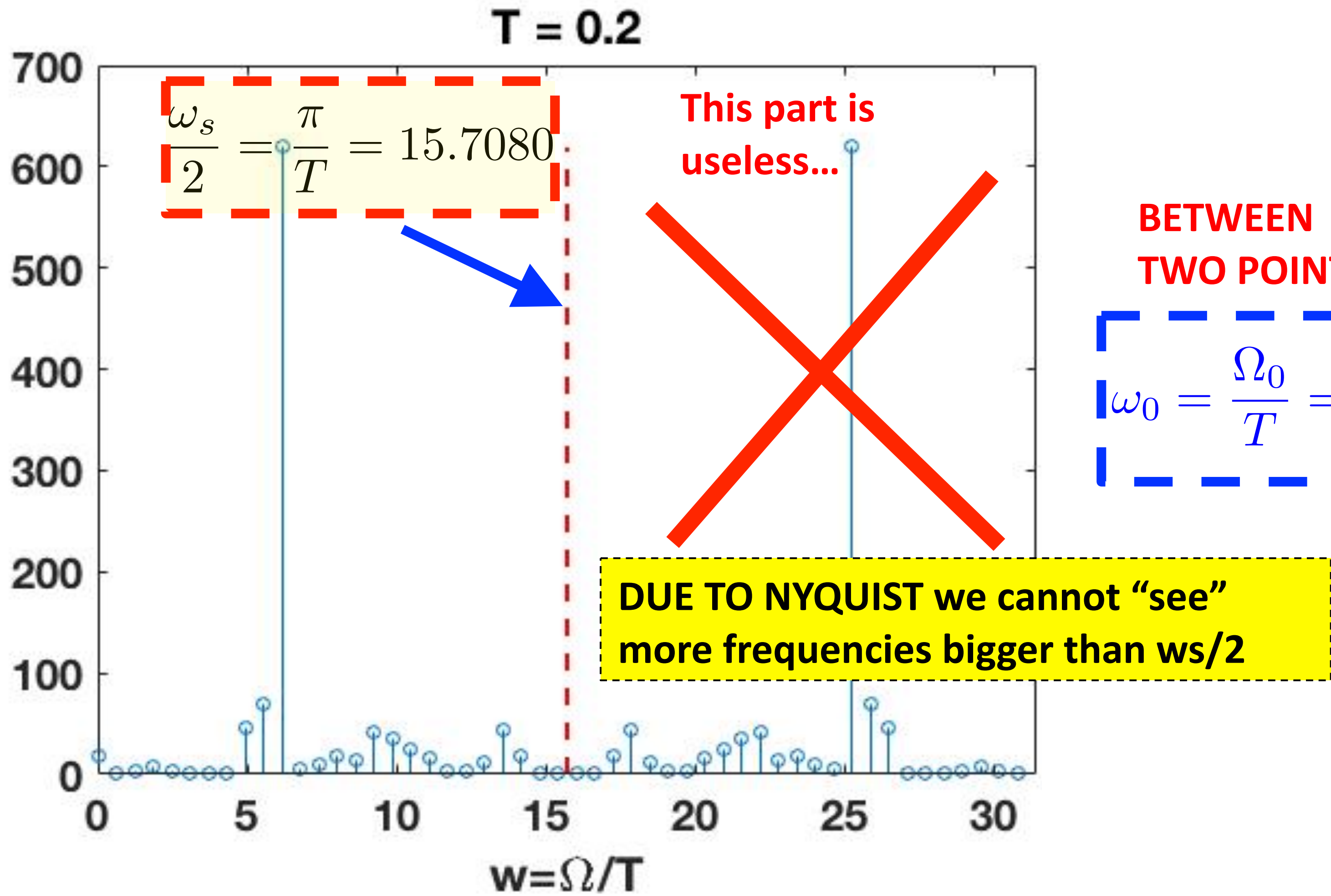


BETWEEN TWO POINTS:

$$\omega_0 = \frac{\Omega_0}{T} = \frac{2\pi}{NT} = 0.6160$$

$|X(\omega)|_{\omega=k\omega_0}$

IMP!!!



$N = 51$

BETWEEN TWO POINTS:

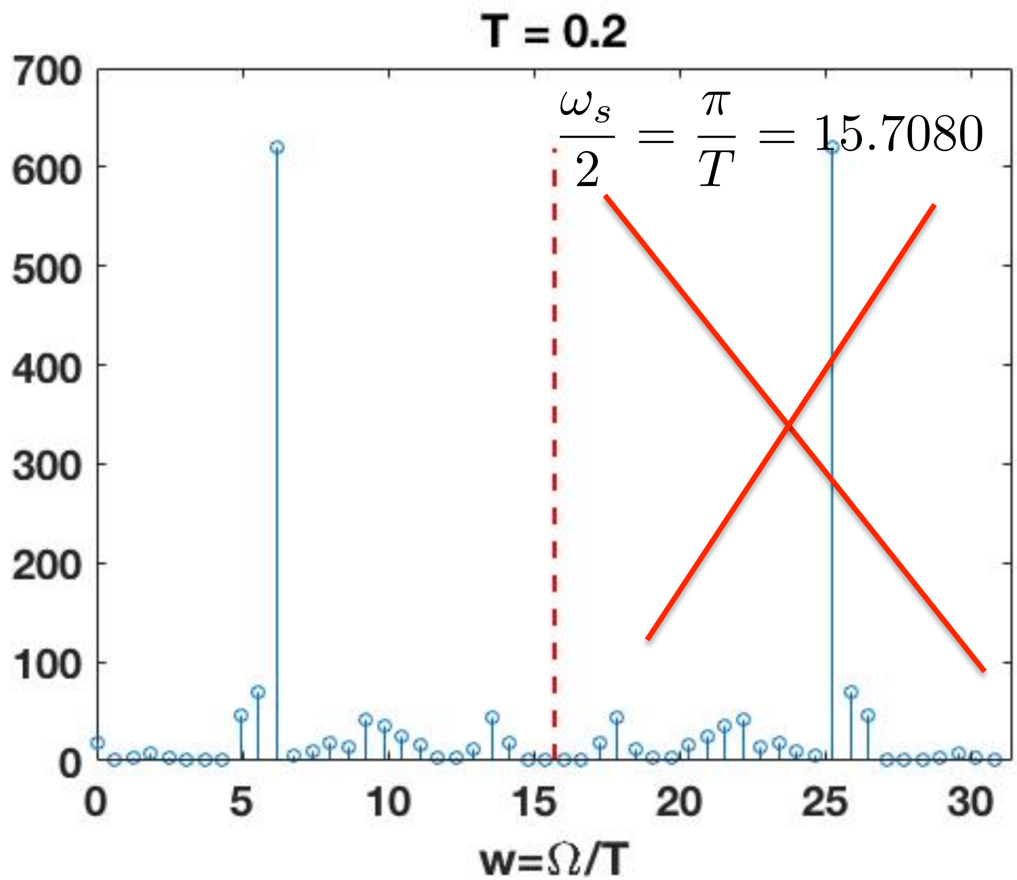
$$\omega_0 = \frac{\Omega_0}{T} = \frac{2\pi}{NT} = 0.6160$$

If we satisfy Nyquist, we can "see" until $\frac{\omega_s}{2} = \frac{\pi}{T}$

IMP!!!

If we do not satisfy Nyquist, the FFT does not make any sense…. It is useless (we lose too much information in the sampling procedure).

FFT en Matlab: interpreting as FT of x(t)



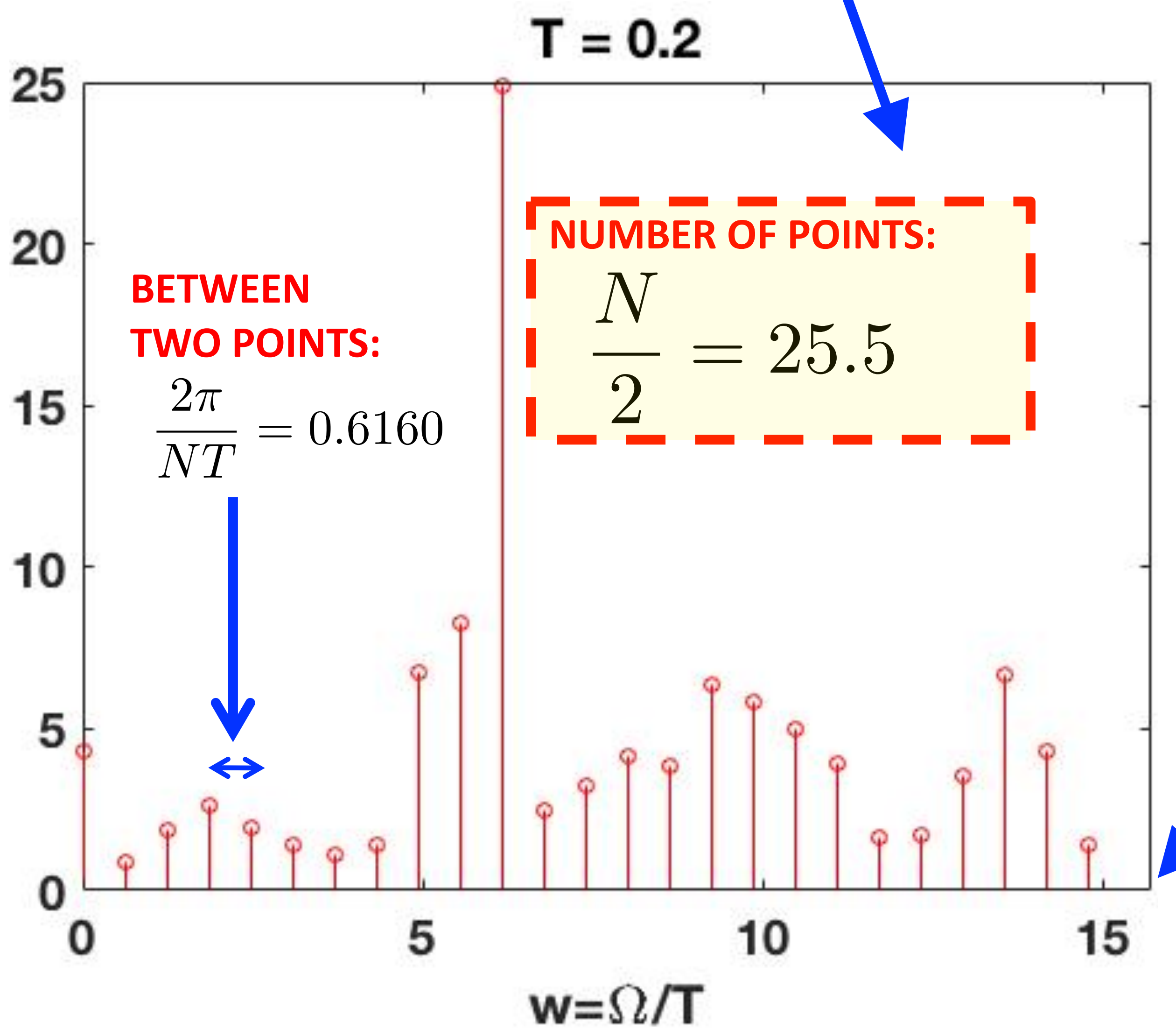
If the signal contains frequencies higher than $\frac{\omega_s}{2} = \frac{\pi}{T}$

THEN the sampling has been not done properly!

Interpreting as FT of x(t):

just HALF of points

$$N = 51$$



$$\frac{\omega_s}{2} = \frac{\pi}{T} = 15.7080$$