EE327 Digital Signal Processing Discrete Fourier Transform DFT

Yasser F. O. Mohammad

REMINDER 1: Common Impulse Responses

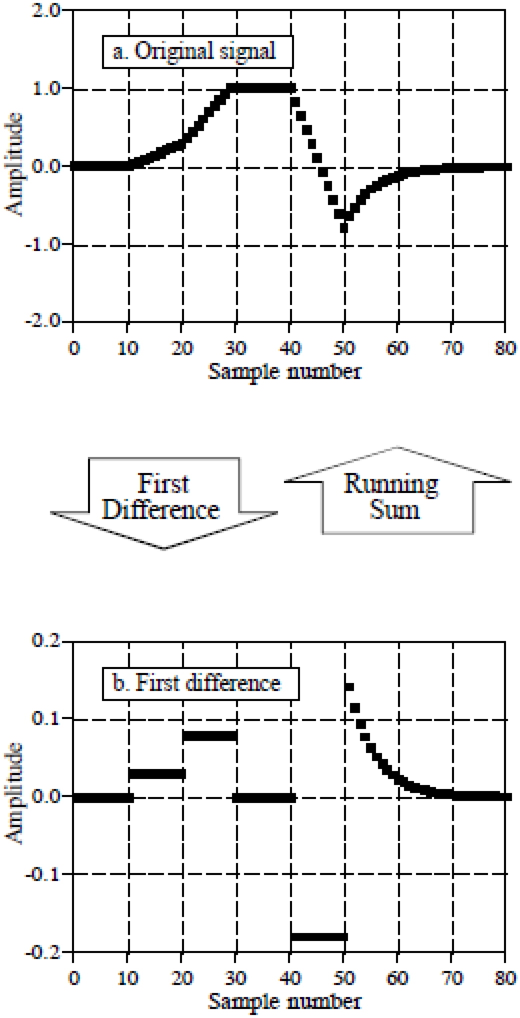
- Identity System: $x[n]*\delta[n]=x[n]$
- Amplifier/Attenuator: $x[n]*k\times\delta[n]=k\times x[n]$
- Delay/Shift: $x[n]*\delta[n+s]=x[n+s]$
- Echo: $x[n]*(\delta[n]+\delta[n+s])=x[n]+x[n+s]$

REMINDER 2: Discretizing Calculus

- First Difference :

$$y[n] = x[n] - x[n-1]$$

- Discrete equivalent of differentiation
  - Discrete Derivative



- Running Sum:

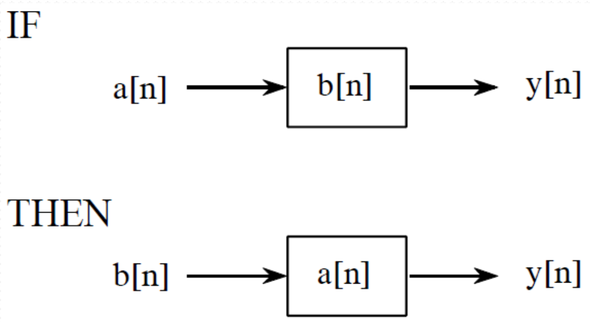
$$y[n] = \sum_{i=0}^{n} x[i] = x[n] + y[n-1]$$

  - Discrete equivalent of integration
  - Discrete Integral

REMINDER 3: Properties of Convolution

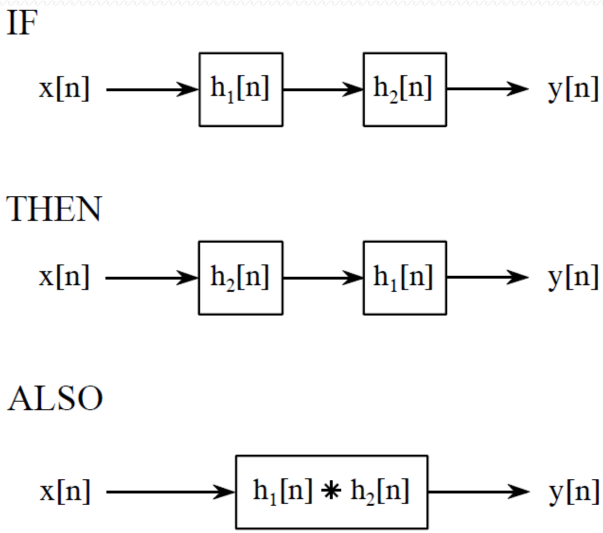
- Commutative Property:

$$a[n]*b[n]=b[n]*a[n]$$



- Associative Property:

$$a[n]*\left(b[n]*c[n]\right)=$$
$$\left(a[n]*b[n]\right)*c[n]$$



Now What?

- We can analyze systems in time domain using impulse response and convolution
- We will look at how to do the same thing in the frequency domain using Fourier analysis and just multiplication.
- Why?
  - More insight (sometimes)
  - Faster (sometimes)

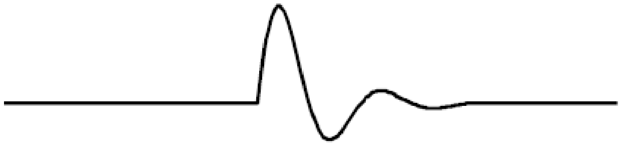
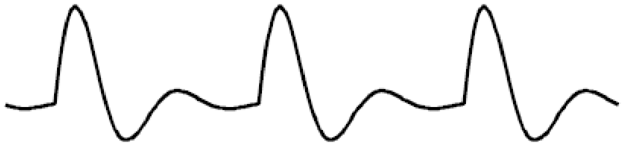
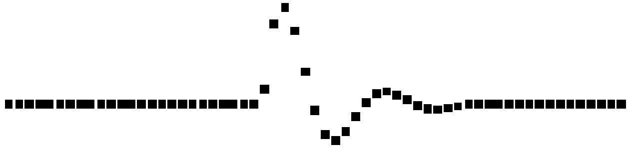
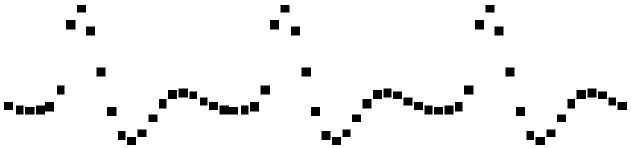
What is a transform

- A multi-input multi-output function

- We use it to see the data from a different prespective

- Examples:
  - Fourier transform
  - Laplace transform
  - Z transform
  - Discrete Cosine Transform
  - etc

Types of Fourier Decompositions

Type of Transform	Example Signal
Fourier Transform *signals that are continious and aperiodic*	
Fourier Series *signals that are continious and periodic*	
Discrete Time Fourier Transform *signals that are discrete and aperiodic*	
Discrete Fourier Transform *signals that are discrete and periodic*	

Fourier Decomposition

- Periodic Time Domain → Discrete Frequency Domain
- Discrete Time Domain → Periodic Frequency Domain

		Periodicity	
Continuity		Periodic	aperiodic
	continuous	Fourier Series FS Aperiodic Spectrum Discrete Spectrum	Fourier Transform FT Aperiodic Spectrum Continuous Spectrum
	discrete	Discrete Fourier Transform DFT Periodic Spectrum Discrete Spectrum	Discrete Time Fourier Transform DTFT Periodic Spectrum Continuous Spectrum

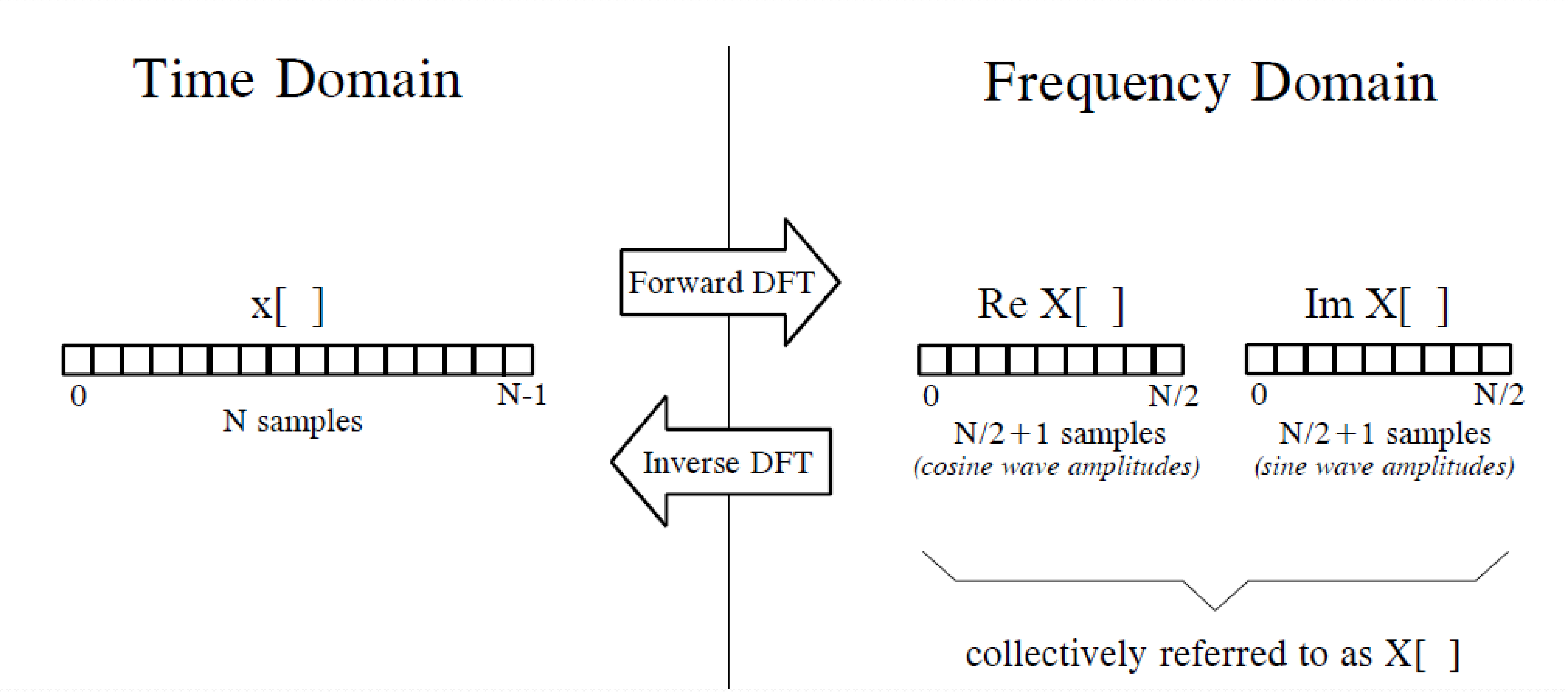
Finite or infinite

- Sine/cosine waves are infinite
- In DSP we have finite signals
- Finite signals cannot be decomposed to infinite parts!!
- What can we do?
  - Pad by zeros to infinity
    - Use DTFT (by the end of this course)
  - Assume the signal is periodic with period N
    - Use DFT (easier)

A point to remember

- When using DFT we assume that the signal we decompose is infinite and PERIODIC and that the period is N

Discrete Fourier Transform



Usually N is a power of 2 (to use FFT)

Notation

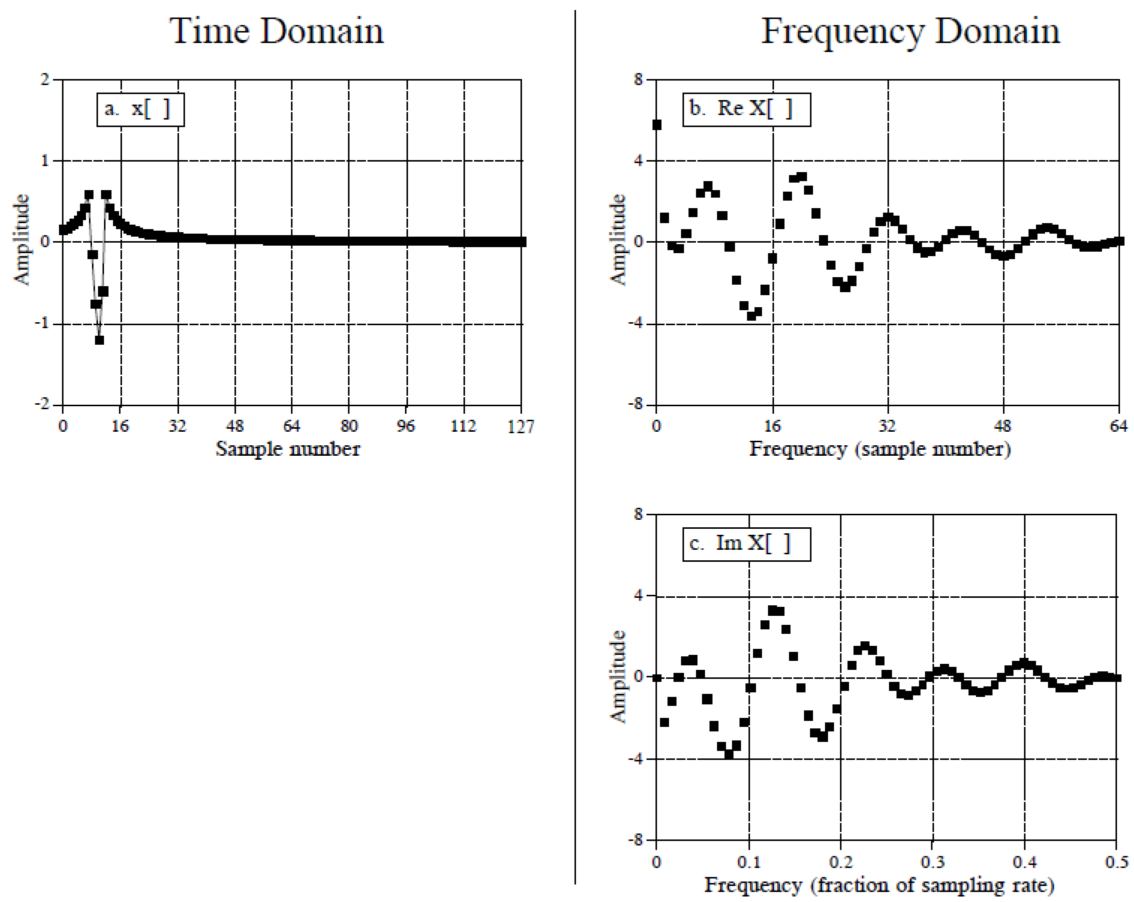
- Time Domain Signal:
  - Lower case letters (e.g. x,y,z)
- Complex Frequency Domain Signal:
  - Upper case letters (e.g. X, Y, Z)
- Real part of the frequency domain signal:
  - *Re*X, *Re*Y, *Re*Z
- Imaginary part of the frequency domain signal:
  - *Im*X, *Im*Y, *Im*Z

Example DFT

Time Domain : 0→N

Frequency Domain:
k : 0→N/2
f: 0→ 0.5
ω: 0→π

f is a fraction of f_s



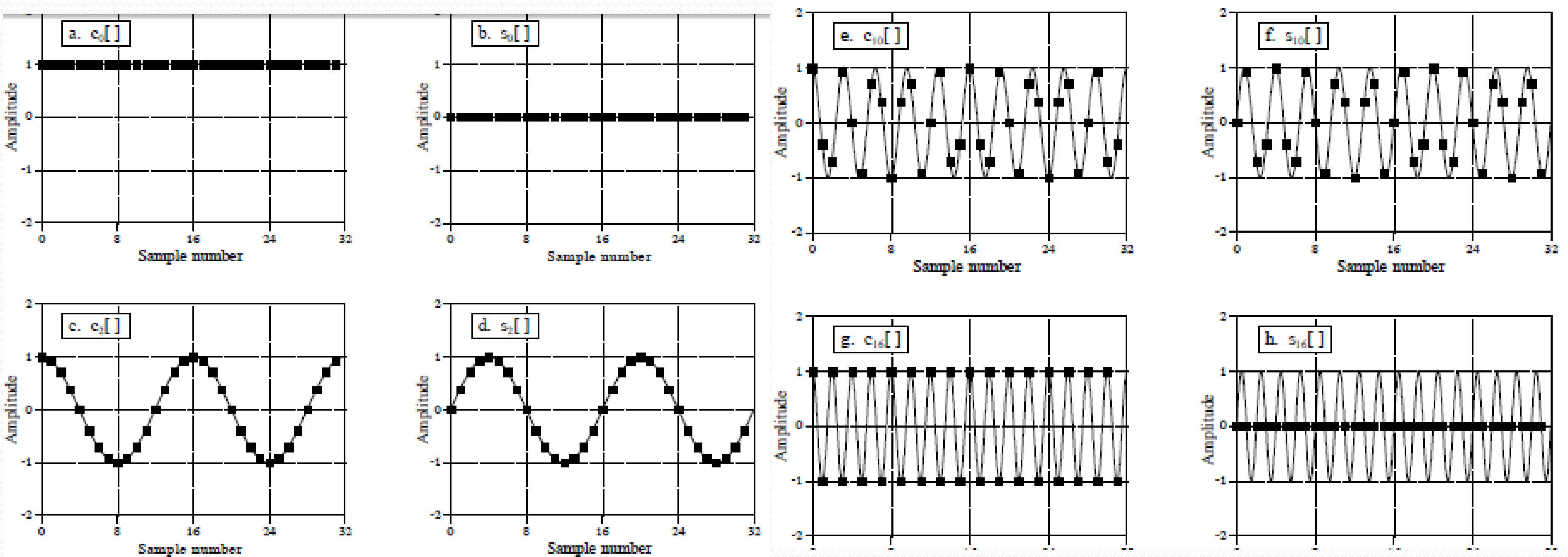
How to use the three notations

- $x[n] = \cos(2\pi kn/N)$
- $x[n] = \cos(2\pi f\ n)$
- $x[n] = \cos(\omega\ n)$

- This means:
  - $f = k/N$
  - $\omega = 2\pi f$

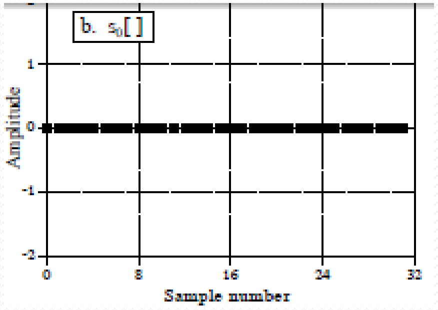
DFT basis functions

- $c_k[n] = \cos(2\pi kn/N)$
- $s_k[n] = \sin(2\pi kn/N)$

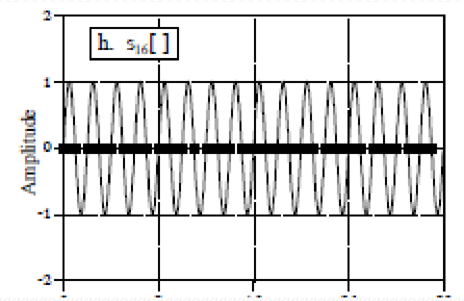


A puzzle for you

- Input is N points
- Output is 2*(N/2+1) = N**+2**
- Where did the extra two points come from???



- Solution
  - ImX[0]=ImX[N/2]=0
- Why?
  - They represent a signal of all zeros that cannot affect the time domain



Synthesis Equation

- From Frequency domain to Time domain

$$x[i] = \sum_{k=0}^{N/2} Re\bar{X}[k]\cos(2\pi ki/N) + \sum_{k=0}^{N/2} Im\bar{X}[k]\sin(2\pi ki/N)$$

$$Re\bar{X}[k] = \frac{ReX[k]}{N/2}$$

$$Im\bar{X}[k] = -\frac{ImX[k]}{N/2}$$

except for two special cases:

$$Re\bar{X}[0] = \frac{ReX[0]}{N}$$

$$Re\bar{X}[N/2] = \frac{ReX[N/2]}{N}$$

Calculating Inverse DFT

```
100 'THE INVERSE DISCRETE FOURIER TRANSFORM
110 'The time domain signal, held in XX[ ], is calculated from the frequency domain signals,
120 'held in REX[ ] and IMX[ ].
140 DIM XX[511]                'XX[ ]  holds the time domain signal
150 DIM REX[256]               'REX[ ] holds the real part of the frequency domain
160 DIM IMX[256]               'IMX[ ] holds the imaginary part of the frequency domain
170 '
180 PI = 3.14159265            'Set the constant, PI
190 N% = 512                   'N% is the number of points in XX[ ]
200 '
210 GOSUB XXXX                 'Mythical subroutine to load data into REX[ ] and IMX[ ]
```

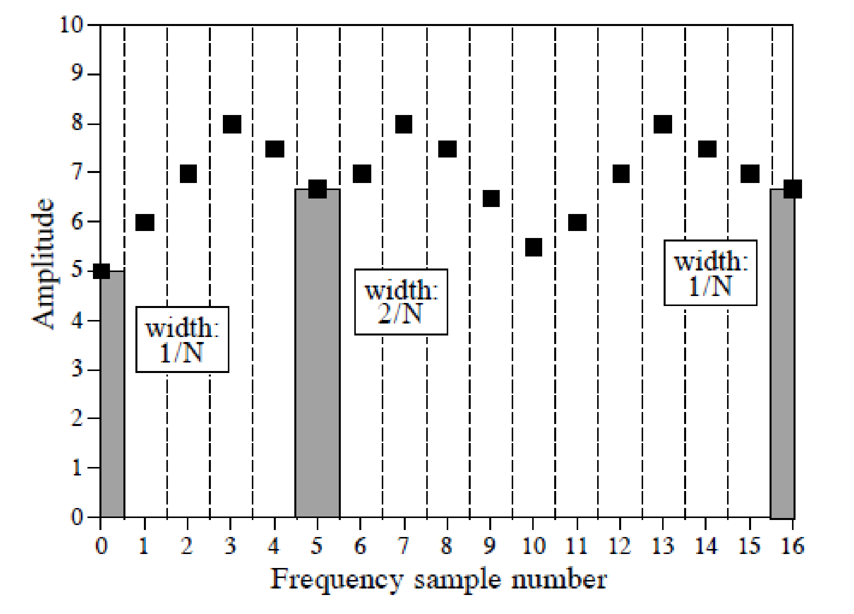
```
250 FOR K% = 0 TO 256
260   REX[K%] =  REX[K%] / (N%/2)
270   IMX[K%]  = -IMX[K%] / (N%/2)
280 NEXT K%
290 '
300 REX[0] = REX[0] / 2
310 REX[256] = REX[256] / 2
320 '
330 '
340 FOR I% = 0 TO 511          'Zero XX[ ] so it can be used as an accumulator
350   XX[I%] = 0
360 NEXT I%
```

```
420 FOR K% = 0 TO 256              'K% loops through each sample
430   FOR I% = 0 TO 511   'I% loops through each sample in XX[ ]
440     '
450     XX[I%] = XX[I%] + REX[K%] * COS(2*PI*K%*I%/N%)
460     XX[I%] = XX[I%] + IMX[K%] * SIN(2*PI*K%*I%/N%)
470     '
480   NEXT I%
490 NEXT K%
500 '
510 END
```

```
420 FOR I% = 0 TO 511              'I% loops through each sample in XX[ ]
430   FOR K% = 0 TO 256            'K% loops through each sample in REX[ ] and IMX[ ]
440     '
450     XX[I%] = XX[I%] + REX[K%] * COS(2*PI*K%*I%/N%)
460     XX[I%] = XX[I%] + IMX[K%] * SIN(2*PI*K%*I%/N%)
470     '
480   NEXT K%
490 NEXT I%
500 '
510 END
```

Why the 2/N, 1/N factors

- Frequency domain signals in DFT are defined as spectral density
- Spectral Density: How much signal (amplitude) exists per unit bandwidth
- Total bandwidth of discrete signals = N/2 (Nyquist)
- Bandwidth of every point is 2/N except first and last



Forward DFT

- Three solutions
  - N equations in N variables
  - Correlation
  - Fast Fourier Transform

DFT by N equations

$$x[i] = \sum_{k=0}^{N/2} Re\bar{X}[k]\cos(2\pi ki/N) + \sum_{k=0}^{N/2} Im\bar{X}[k]\sin(2\pi ki/N)$$

- Each value of *i* gives one equation.
- Remember that ImX[0]=ImX[N/2]=0
- We need N more equations
- Hence, each of ReX and ImX will be N/2+1 as expected
- All equations must be *linearly independent*

DFT by correlation

- Find the correlation between the basis function and the signal
- The average of this correlation is the required amplitude.

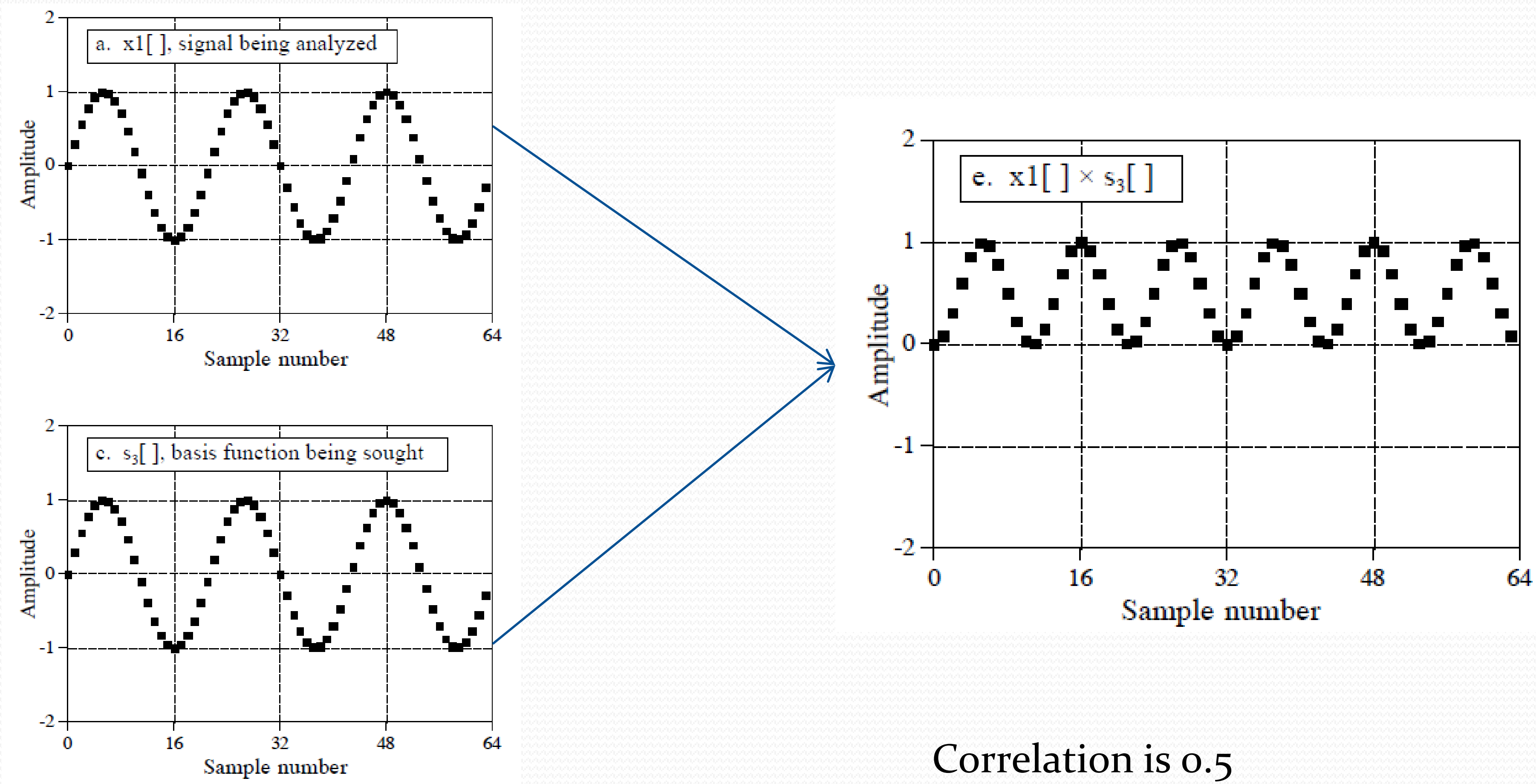
- For this to work all basis functions must have zero correlation.

- Sins and Cosines of different frequency have zero correlation

$$ReX[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi k i / N)$$

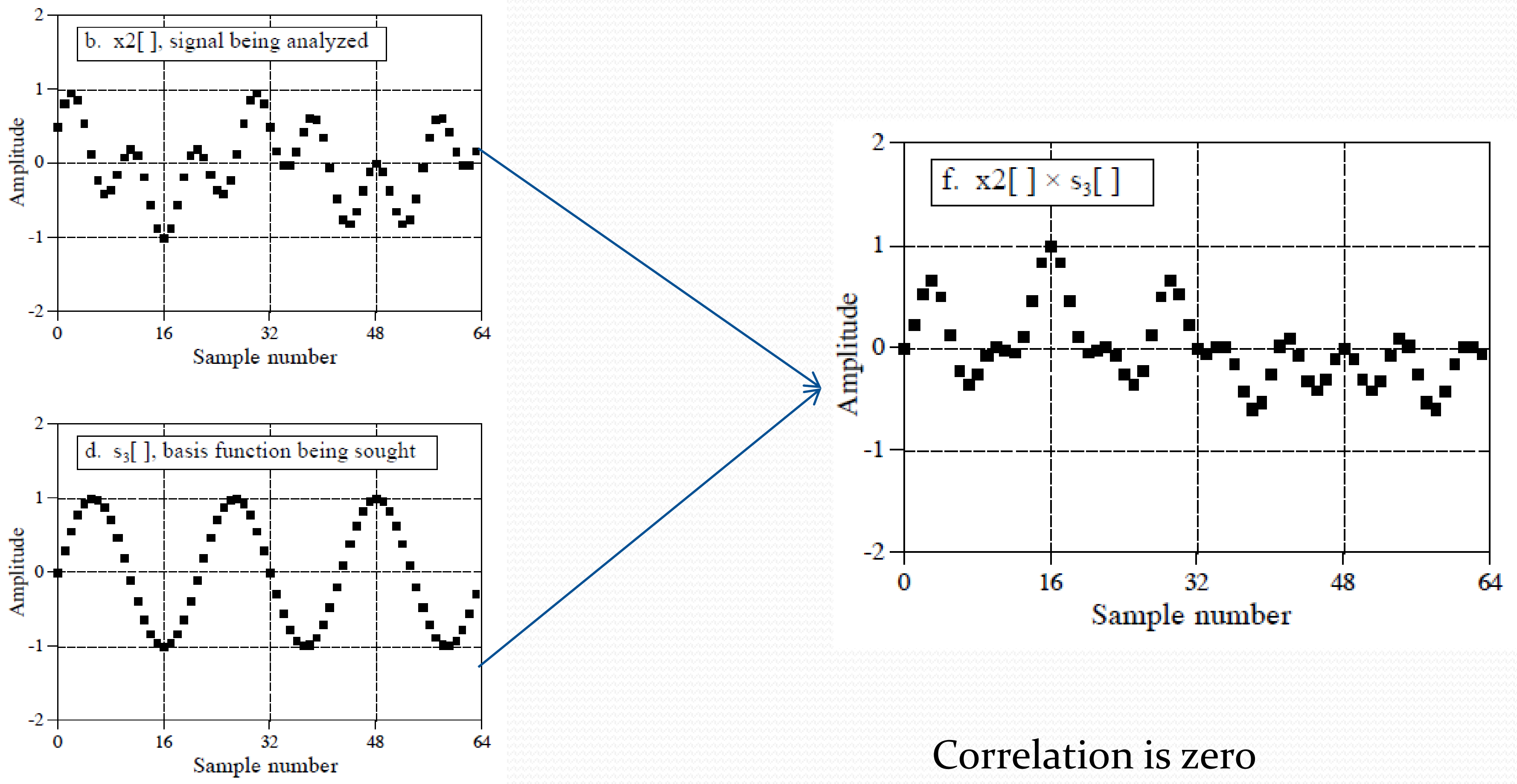
$$ImX[k] = -\sum_{i=0}^{N-1} x[i] \sin(2\pi k i / N)$$

DFT by Correlation Example



Correlation is 0.5

DFT by Correlation Example 2



Correlation is zero

Calculating DFT

$$ReX[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi k i / N)$$

$$ImX[k] = -\sum_{i=0}^{N-1} x[i] \sin(2\pi k i / N)$$

```
100 'THE DISCRETE FOURIER TRANSFORM
110 'The frequency domain signals, held in REX[ ] and IMX[ ], are calculated from
120 'the time domain signal, held in XX[ ].
130 '
140 DIM XX[511]                   'XX[ ]  holds the time domain signal
150 DIM REX[256]                  'REX[ ] holds the real part of the frequency domain
160 DIM IMX[256]                  'IMX[ ] holds the imaginary part of the frequency domain
170 '
180 PI = 3.14159265               'Set the constant, PI
190 N% = 512                      'N% is the number of points in XX[ ]
200 '
210 GOSUB XXXX                    'Mythical subroutine to load data into XX[ ]
220 '
230 '
240 FOR K% = 0 TO 256   'Zero REX[ ] & IMX[ ] so they can be used as accumulators
250   REX[K%] = 0
260   IMX[K%] = 0
270 NEXT K%
280 '
290 '                             'Correlate XX[ ] with the cosine and sine waves, Eq. 8-4
300 '
310 FOR K% = 0 TO 256             'K% loops through each sample in REX[ ] and IMX[ ]
320   FOR I% = 0 TO 511   'I% loops through each sample in XX[ ]
330    '
340    REX[K%] = REX[K%] + XX[I%] * COS(2*PI*K%*I%/N%)
350    IMX[K%] = IMX[K%]  - XX[I%] * SIN(2*PI*K%*I%/N%)
360    '
370   NEXT I%
380 NEXT K%
390 '
400 END
```

Duality

$$ReX[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi k i / N)$$

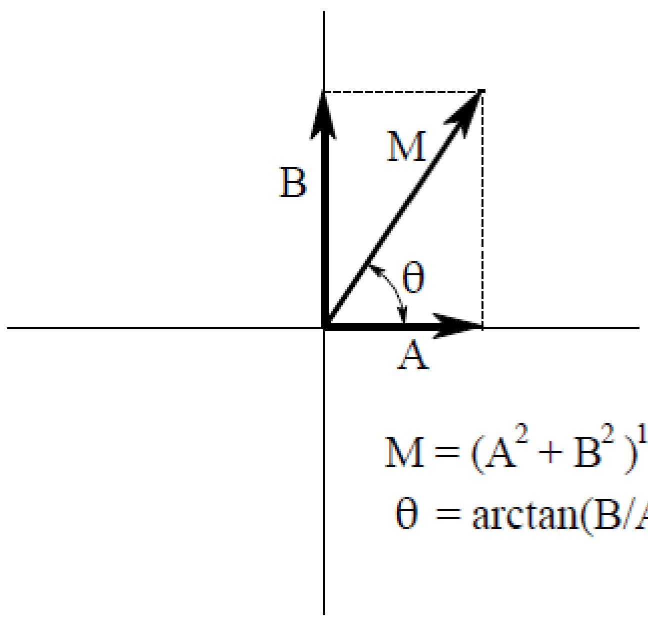
$$ImX[k] = -\sum_{i=0}^{N-1} x[i] \sin(2\pi k i / N)$$

- sine in the time domain → single point in frequency domain
- sine in the frequency domain → single point in time domain

- Convolution in time domain → multiplication in frequency domain
- Convolution in frequency domain → multiplication in time domain

Rectangular and Polar Notations

$$A\cos(x) + B\sin(x) = M\cos(x + \theta)$$



$$M = (A^2 + B^2)^{1/2}$$

$$\theta = \arctan(B/A)$$

Conversion Formulas

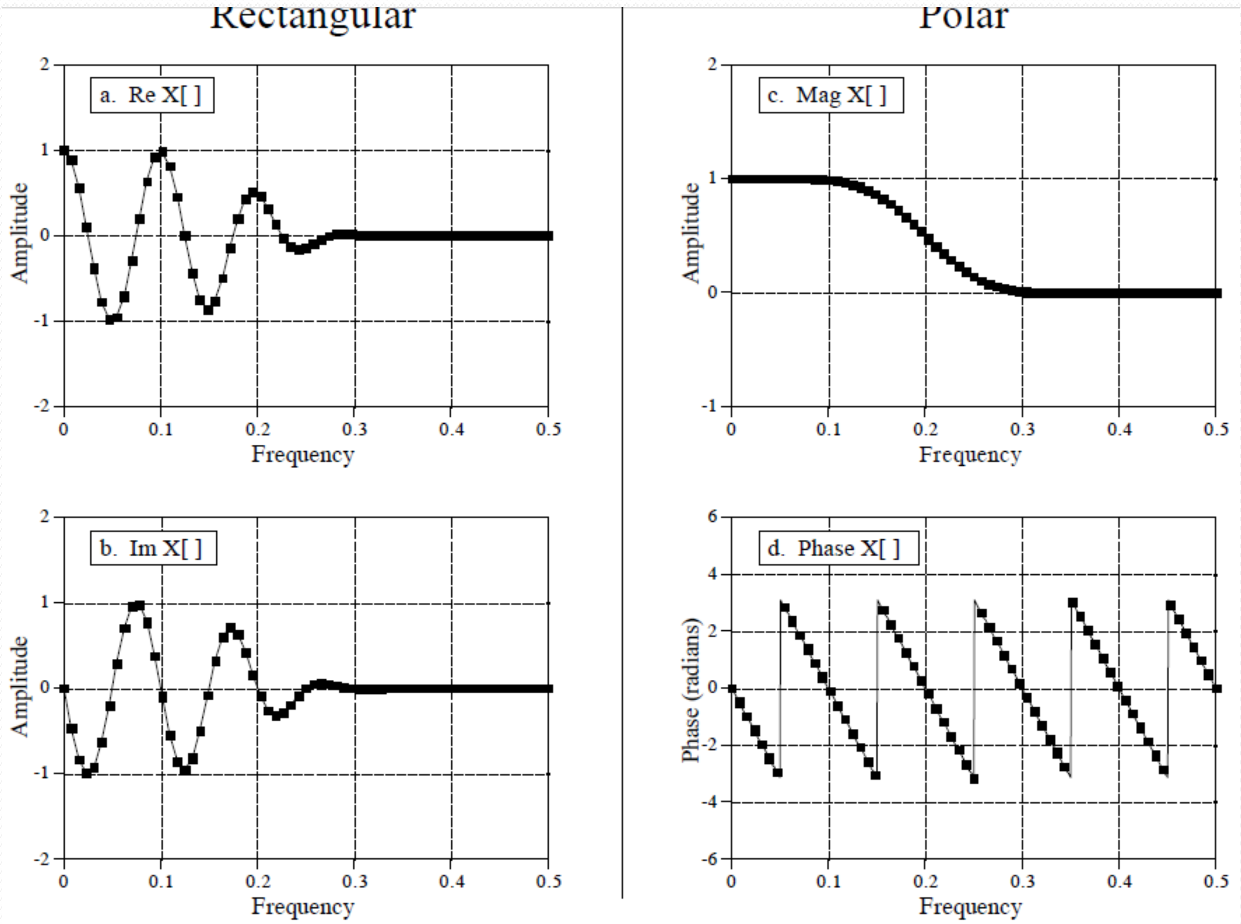
$$MagX[k] = (ReX[k]^2 + ImX[k]^2)^{1/2}$$

$$PhaseX[k] = \arctan\left(\frac{ImX[k]}{ReX[k]}\right)$$

$$ReX[k] = MagX[k] \cos(PhaseX[k])$$

$$ImX[k] = MagX[k] \sin(PhaseX[k])$$

Example



Rectangular
Polar
a. Re X[ ]
b. Im X[ ]
c. Mag X[ ]
d. Phase X[ ]
Amplitude
Phase (radians)
Frequency

When to use what?

- Rectangular form is usually used for calculations
- Polar form is usually used for display
  - Sinusoidal fidelity means that the only changes possible to a sinusoidal are phase shifts and amplitude scaling
  - These are clear in the polar form

Conversion algorithm

```
100 'RECTANGULAR-TO-POLAR & POLAR-TO-RECTANGULAR CONVERSION
110 '
120 DIM REX[256]                    'REX[ ]     holds the real part
130 DIM IMX[256]                    'IMX[ ]     holds the imaginary part
140 DIM MAG[256]                    'MAG[ ]     holds the magnitude
150 DIM PHASE[256]                  'PHASE[ ]   holds the phase
160 '
170 PI = 3.14159265
180 '
190 GOSUB XXXX                      'Mythical subroutine to load data into REX[ ] and IMX[ ]
200 '
210 '
220 '                               'Rectangular-to-polar conversion, Eq. 8-6
230 FOR K% = 0 TO 256
240   MAG[K%] = SQR( REX[K%]^2 + IMX[K%]^2 )            'from Eq. 8-6
250   IF REX[K%] = 0 THEN REX[K%] = 1E-20               'prevent divide by 0 (nuisance 2)
260   PHASE[K%] = ATN( IMX[K%] / REX[K%] )              'from Eq. 8-6
270   '                                                 'correct the arctan (nuisance 3)
280   IF REX[K%] < 0 AND IMX[K%] <  0 THEN PHASE[K%] = PHASE[K%]  - PI
290   IF REX[K%] < 0 AND IMX[K%] >= 0 THEN PHASE[K%] = PHASE[K%] + PI
300 NEXT K%
310 '
320 '
330 '                               'Polar-to-rectangular conversion, Eq. 8-7
340 FOR K% = 0 TO 256
350   REX[K%] = MAG[K%] * COS( PHASE[K%] )
360   IMX[K%] = MAG[K%] *  SIN( PHASE[K%] )
370 NEXT K%
380 '
390 END
```

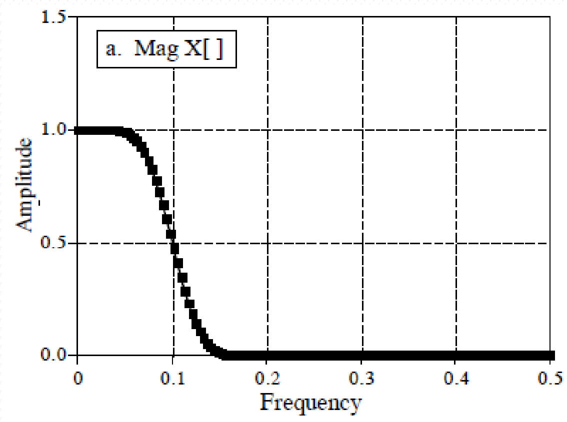
Notes on Polar form

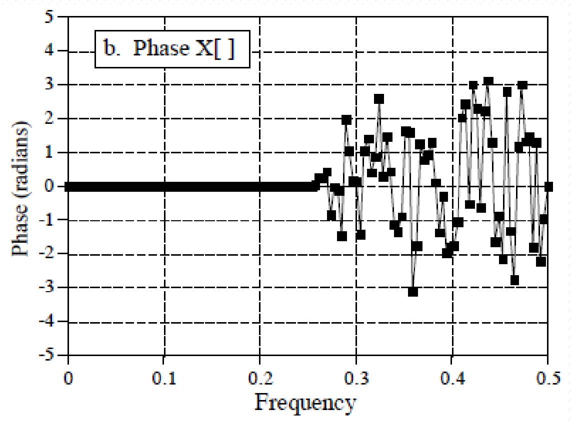
- As defined all phases are in radians not degrees
- Remember not to divide by zero when ReX[i]=0
- Calculating phase:

ReX	ImX	Correction
+	+	0
+	-	0
-	+	$+\pi$
-	-	$-\pi$

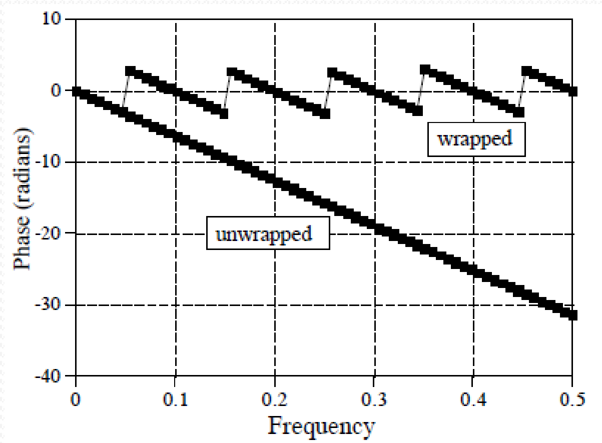
Notes on Polar form

- Very small amplitudes cause large noise in the phase
  - (-π→ π)





- Phase wrapping (2 π ambiguity)
  - Solution: unwrapping



Apparent discontinuity of phase