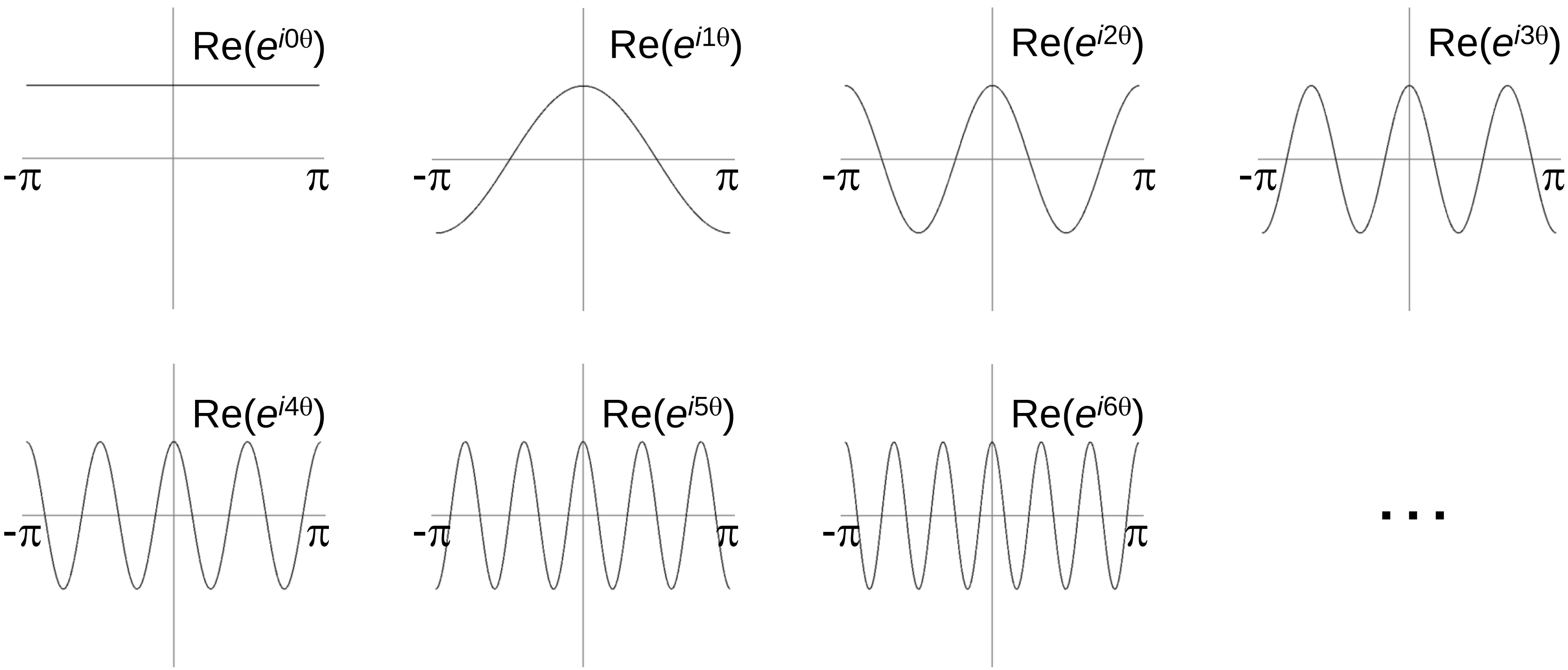
Physically Based Rendering (600.657)

Sampling

Fourier Analysis

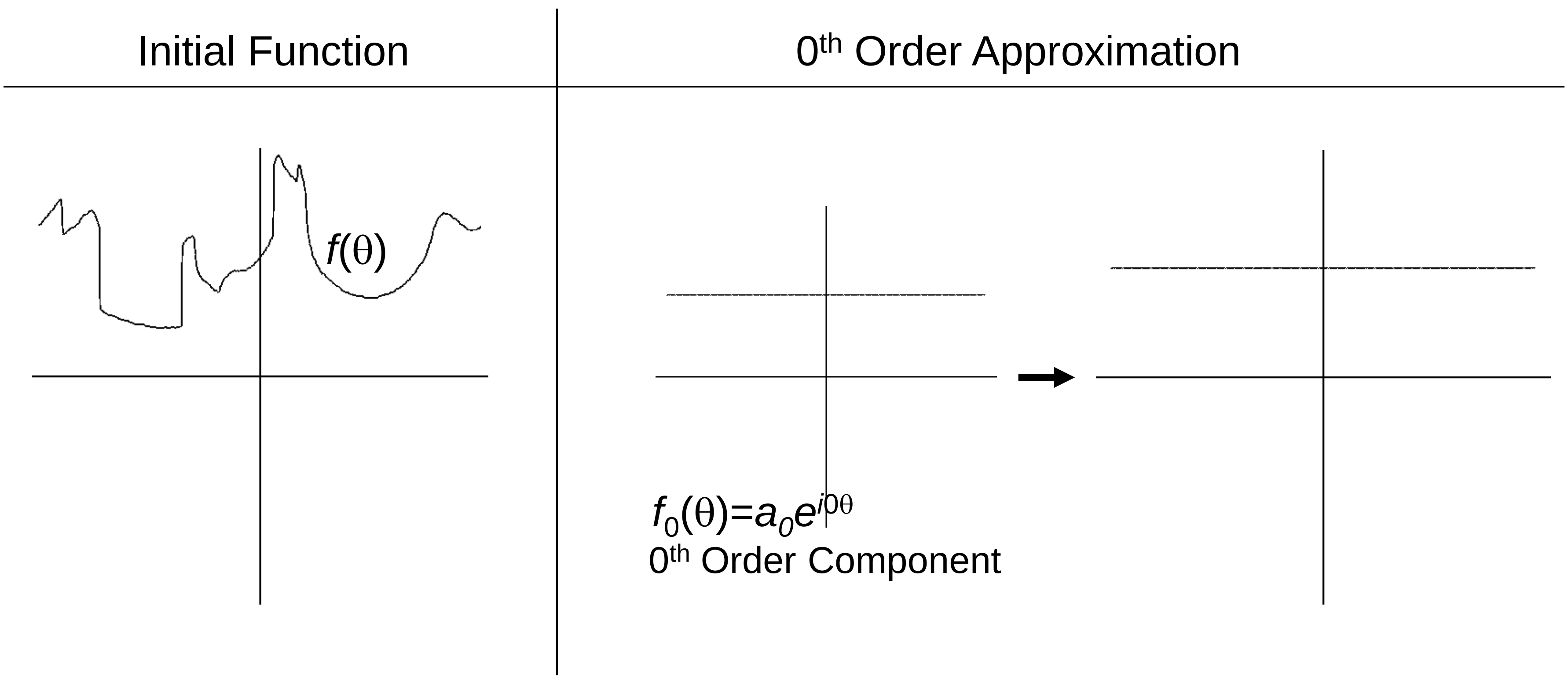
- Fourier analysis provides a way for expressing (or approximating) any signal as a sum of complex exponentials.



The Building Blocks for the Fourier Decomposition

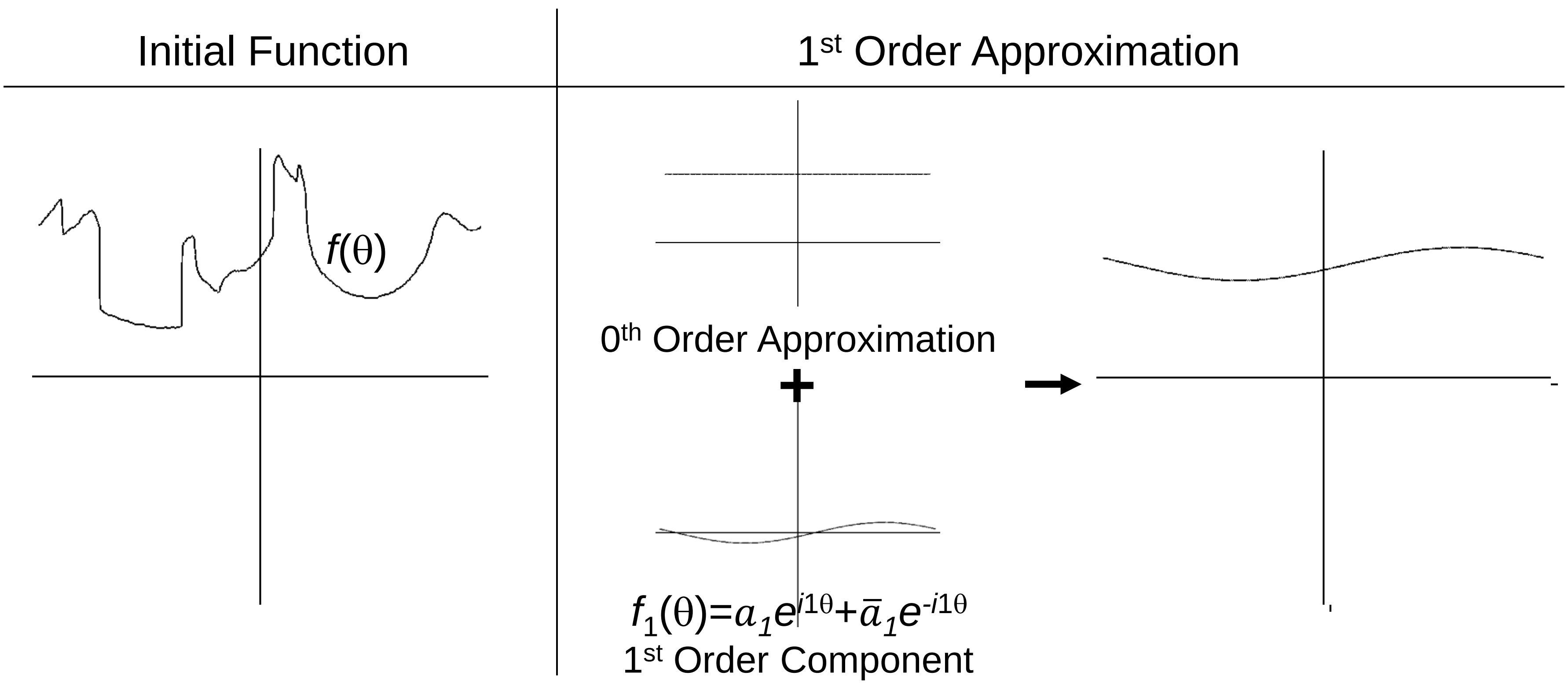
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



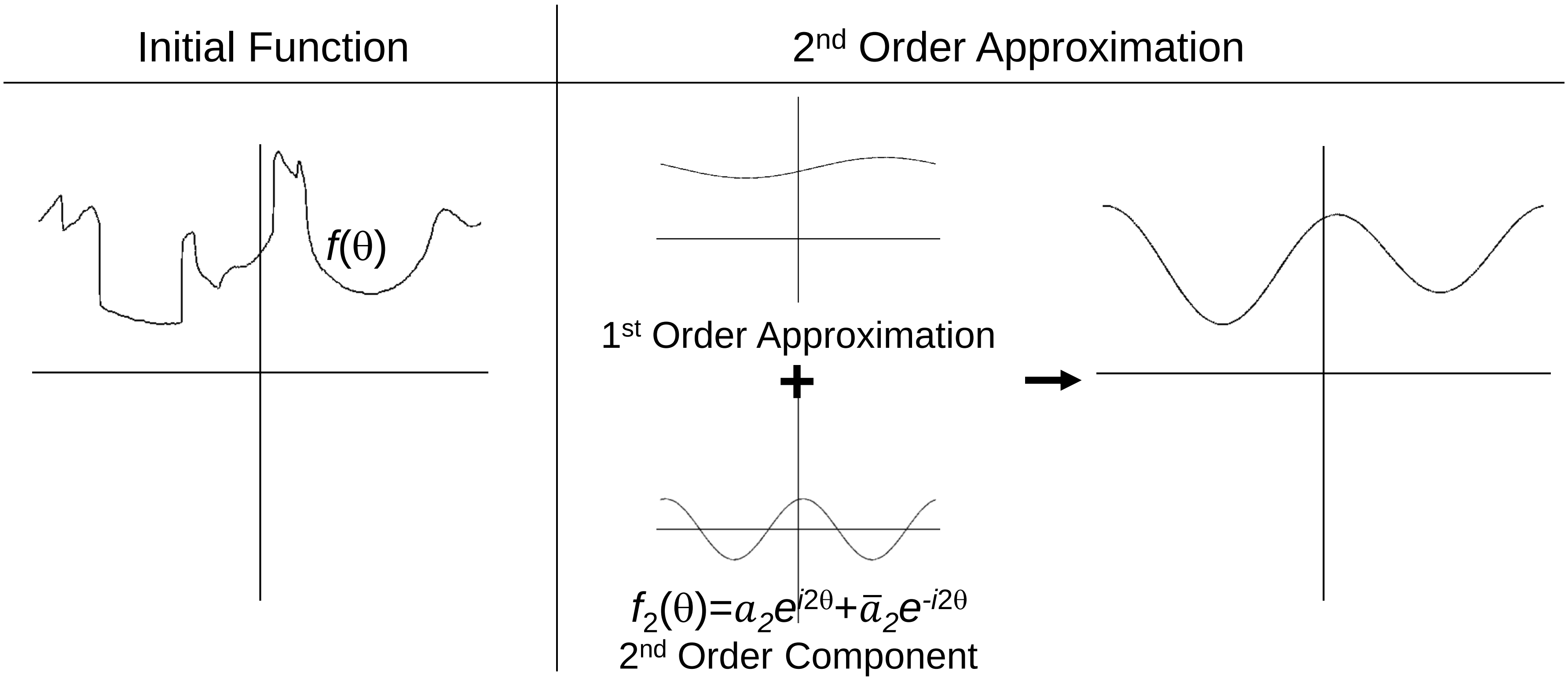
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



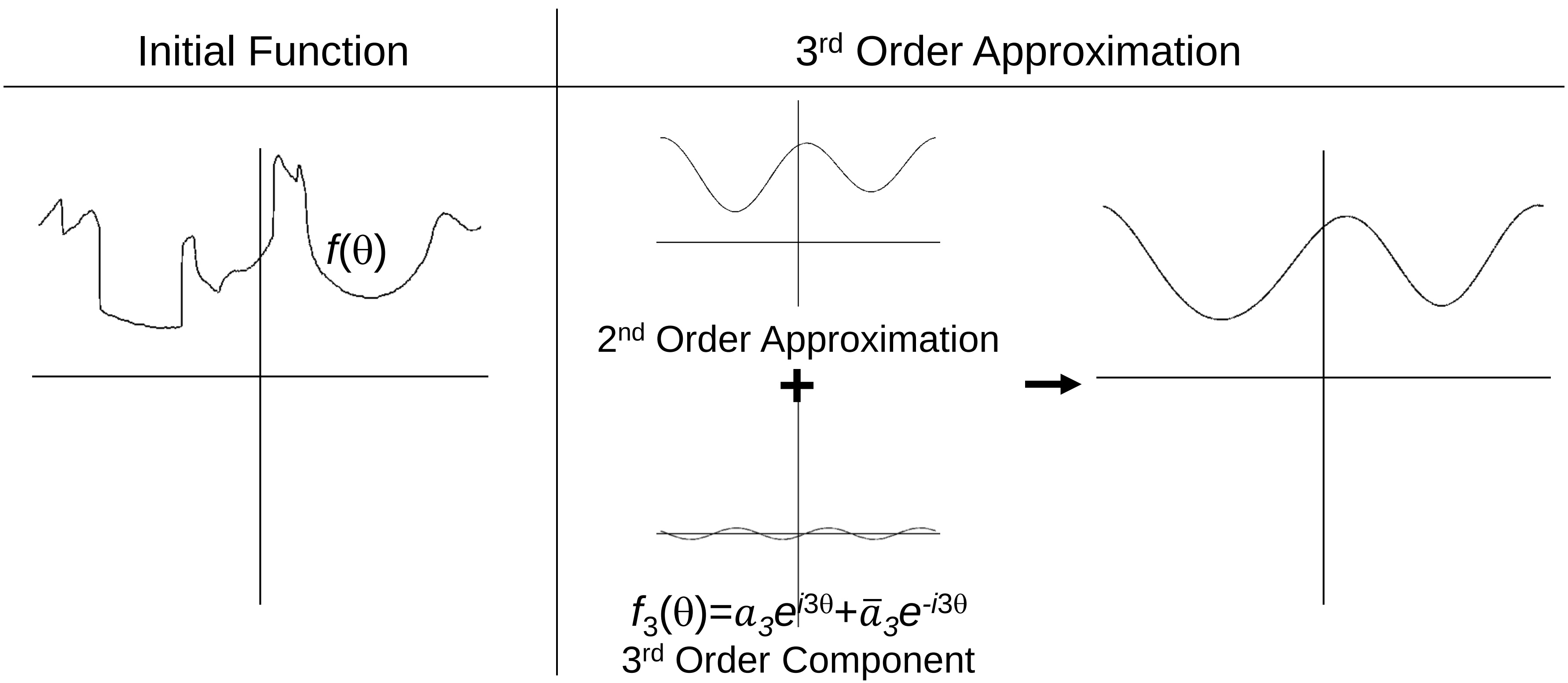
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



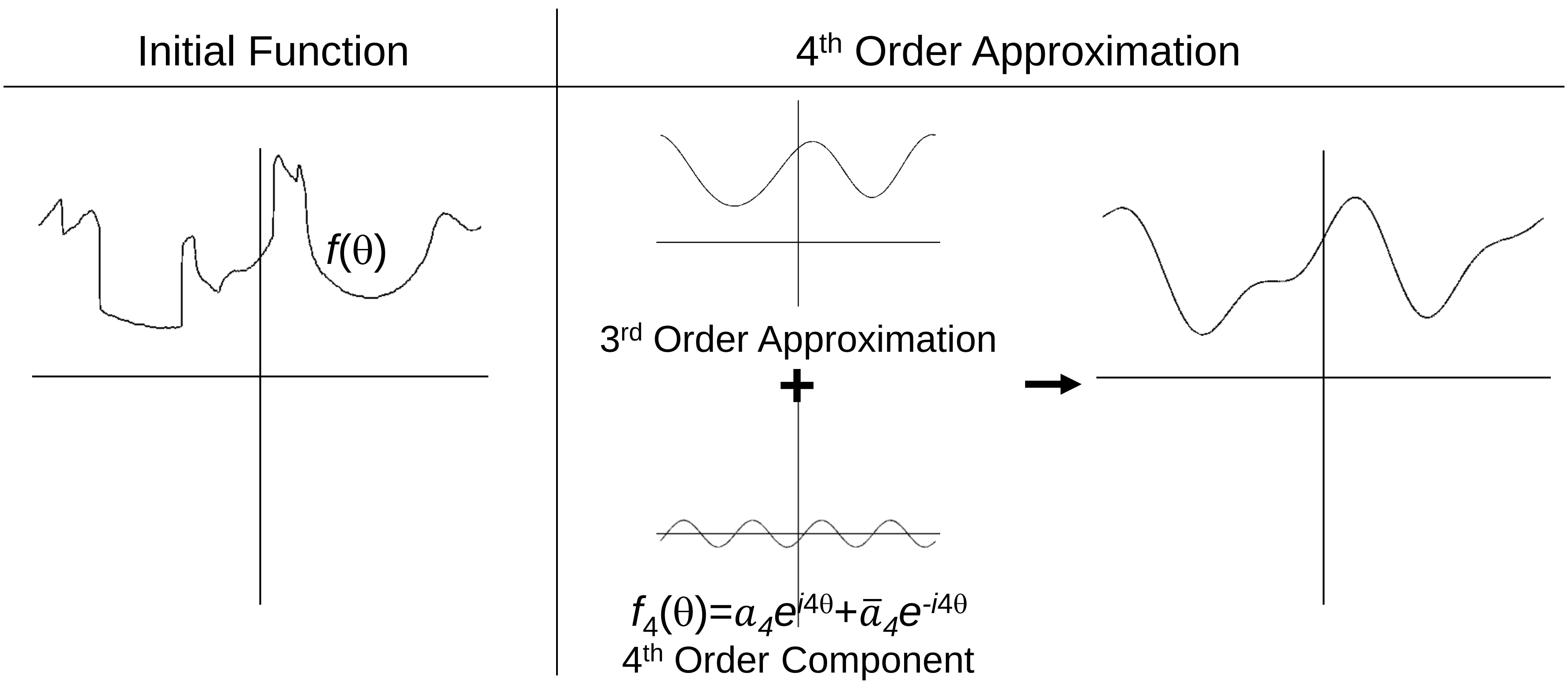
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



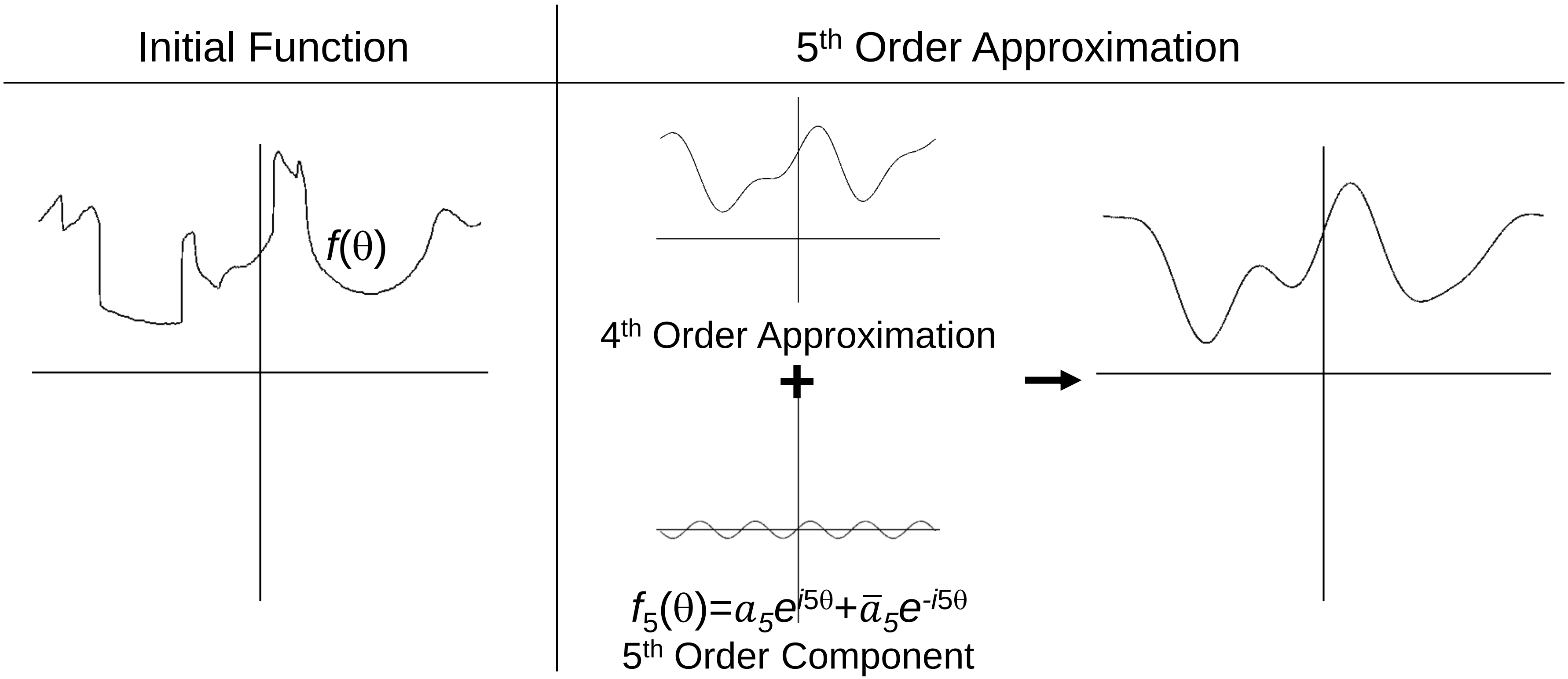
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



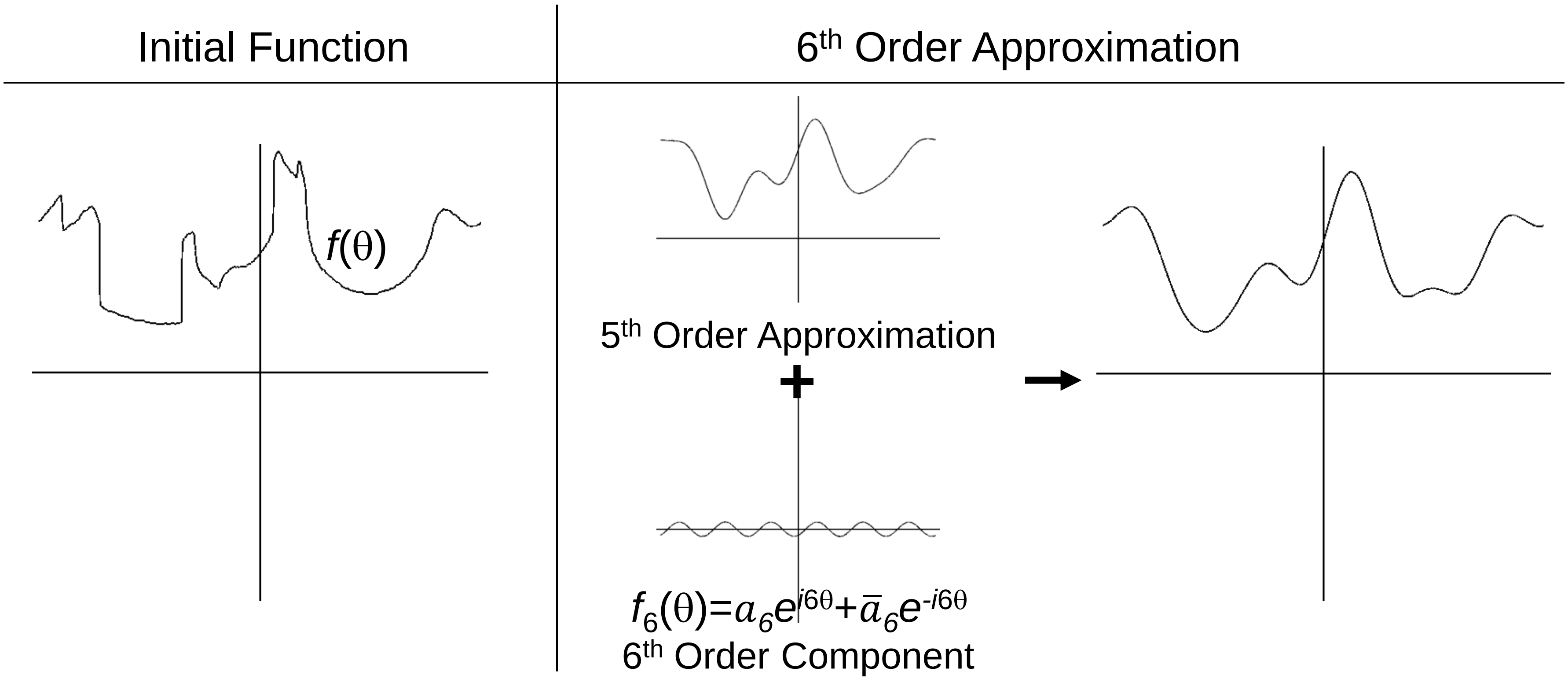
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



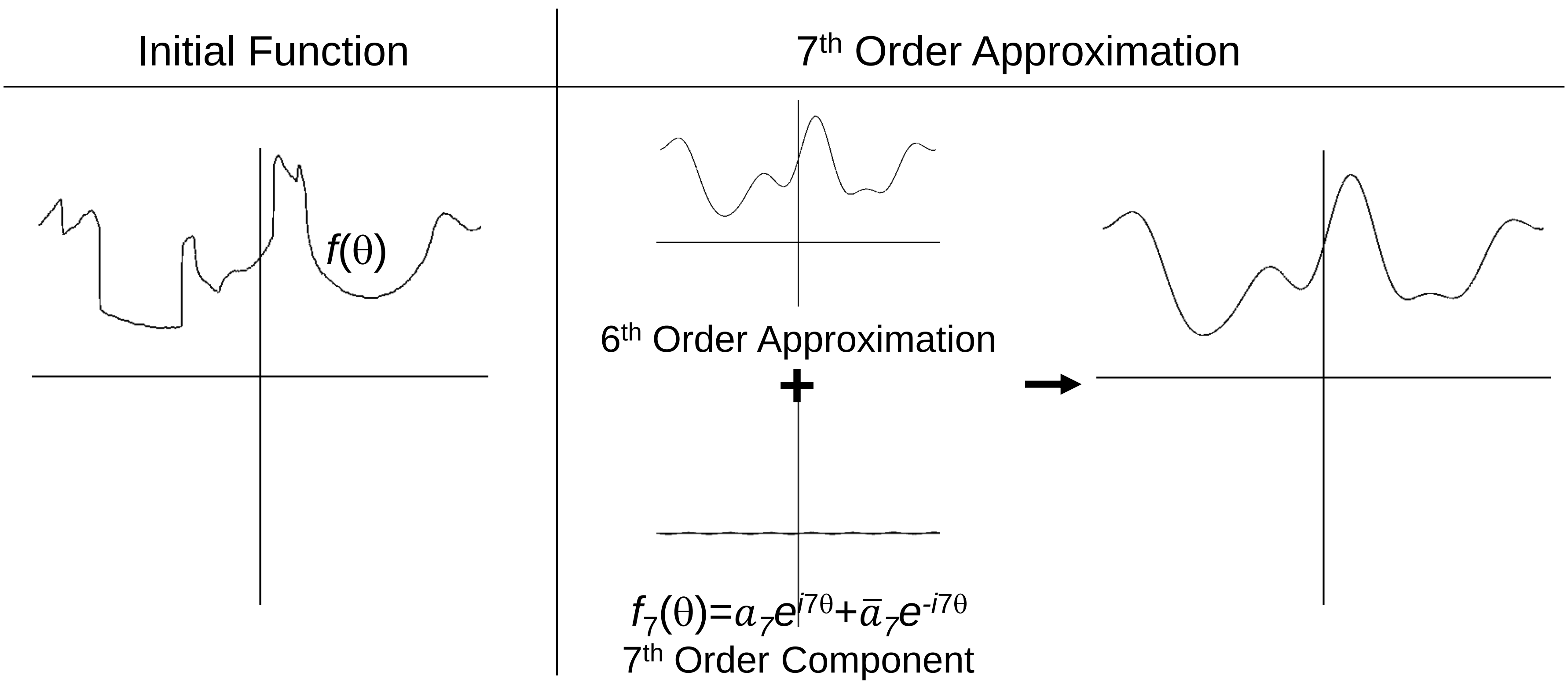
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



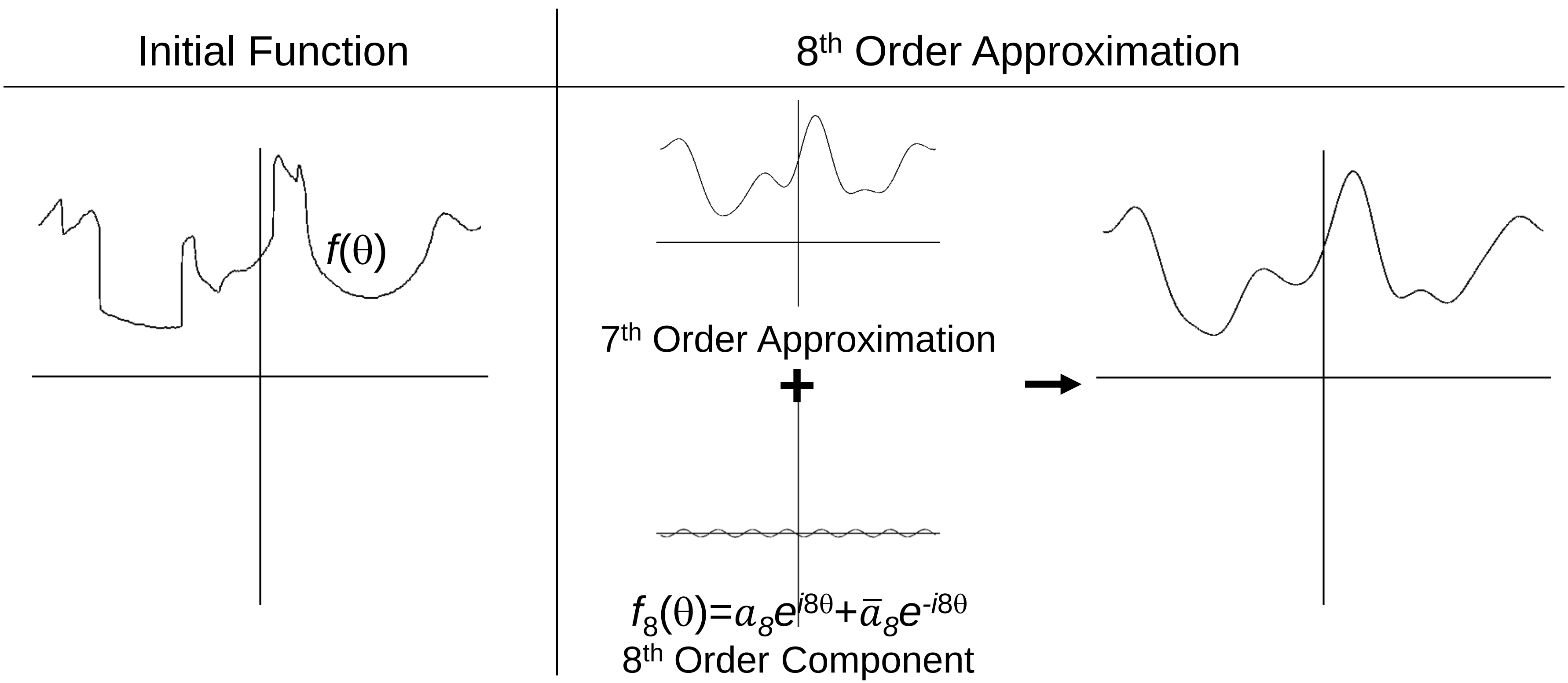
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



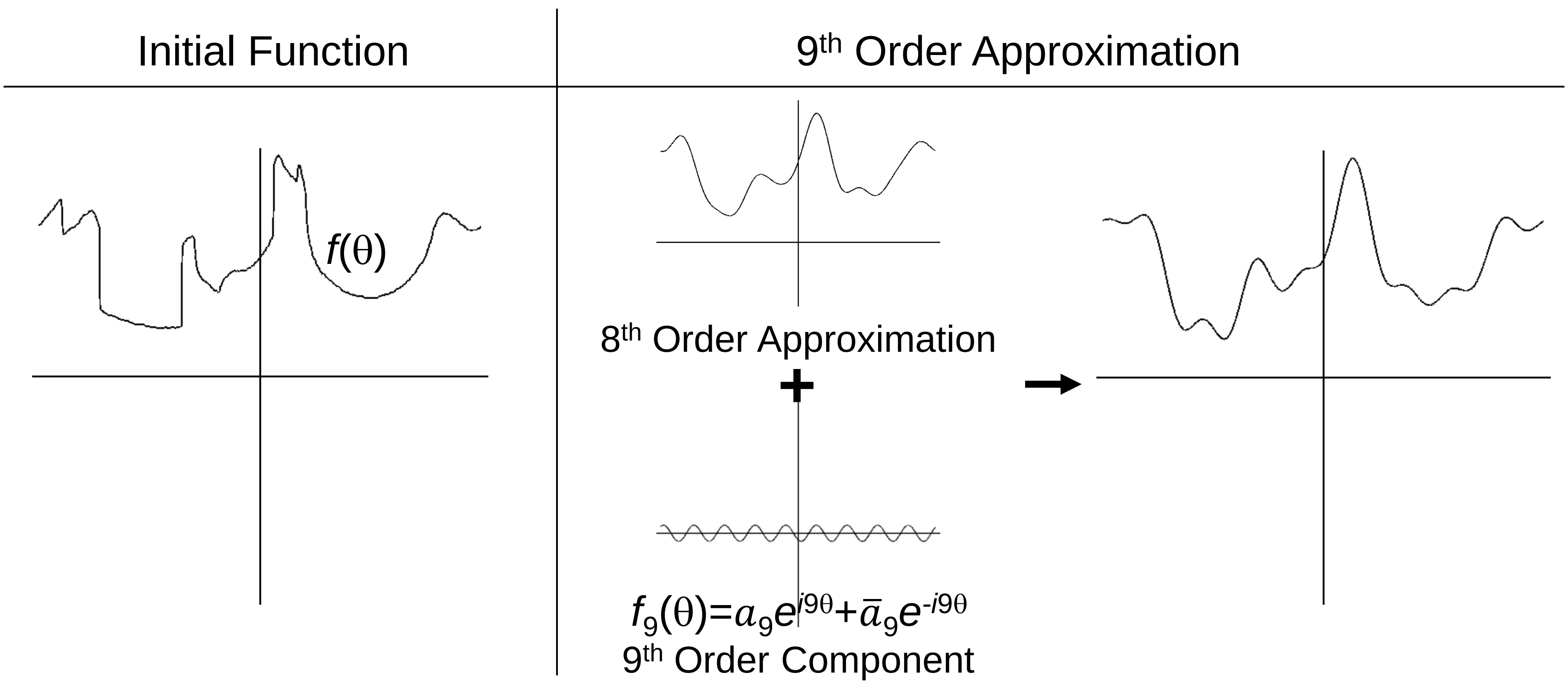
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



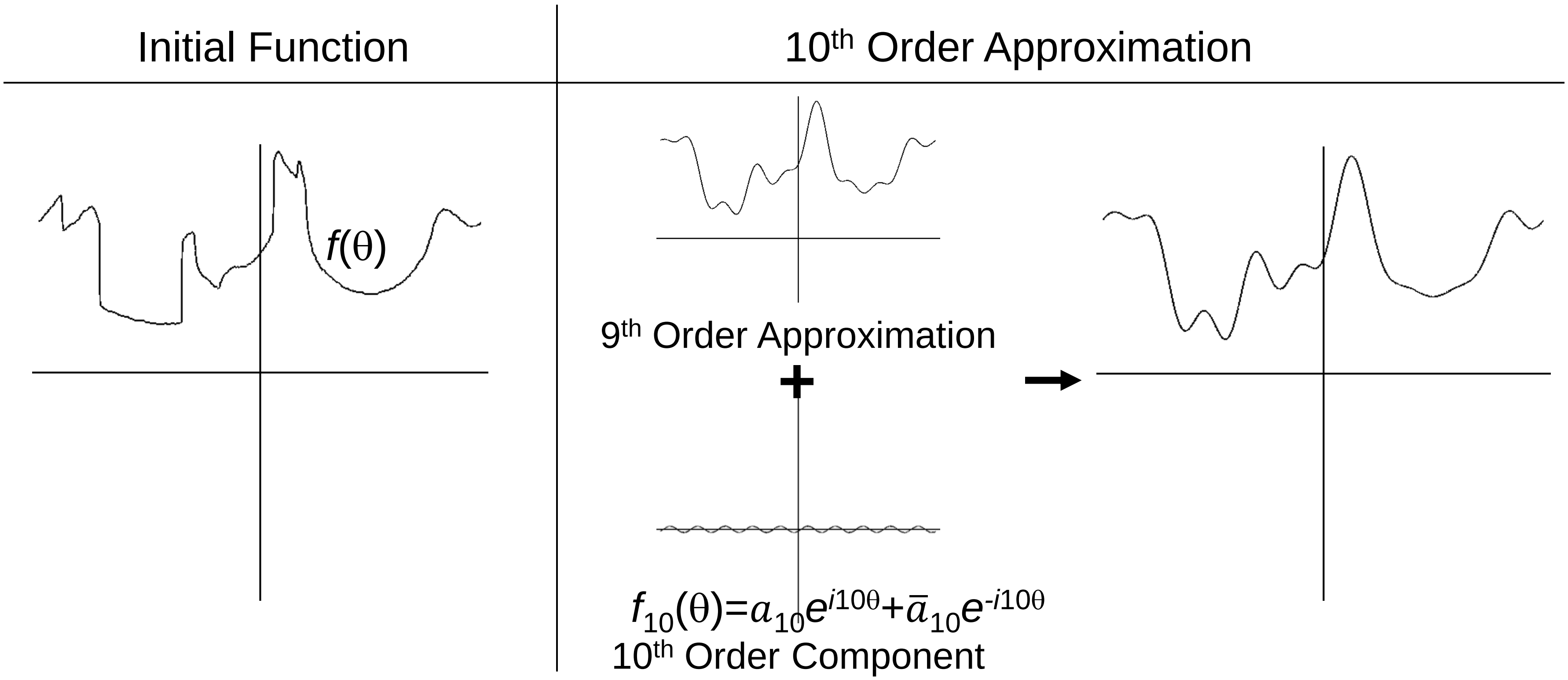
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



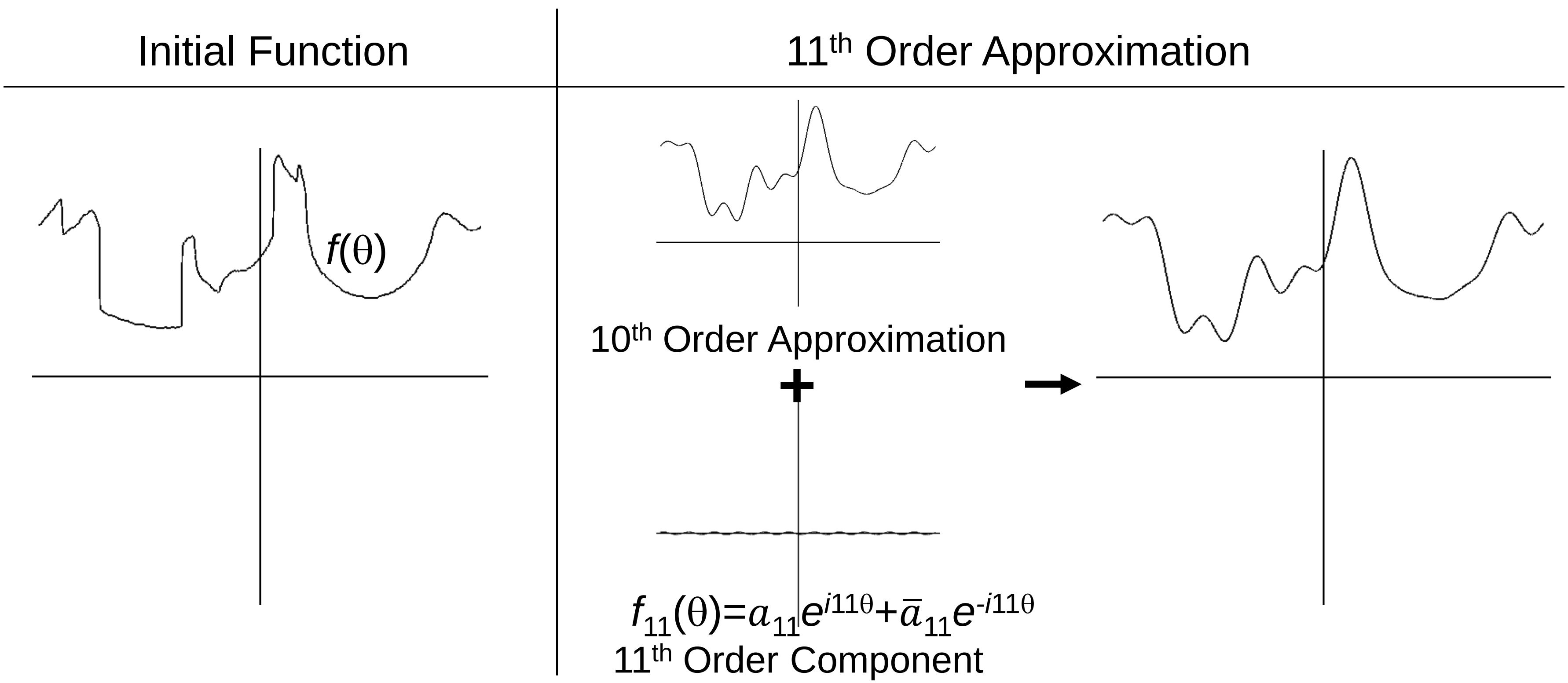
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



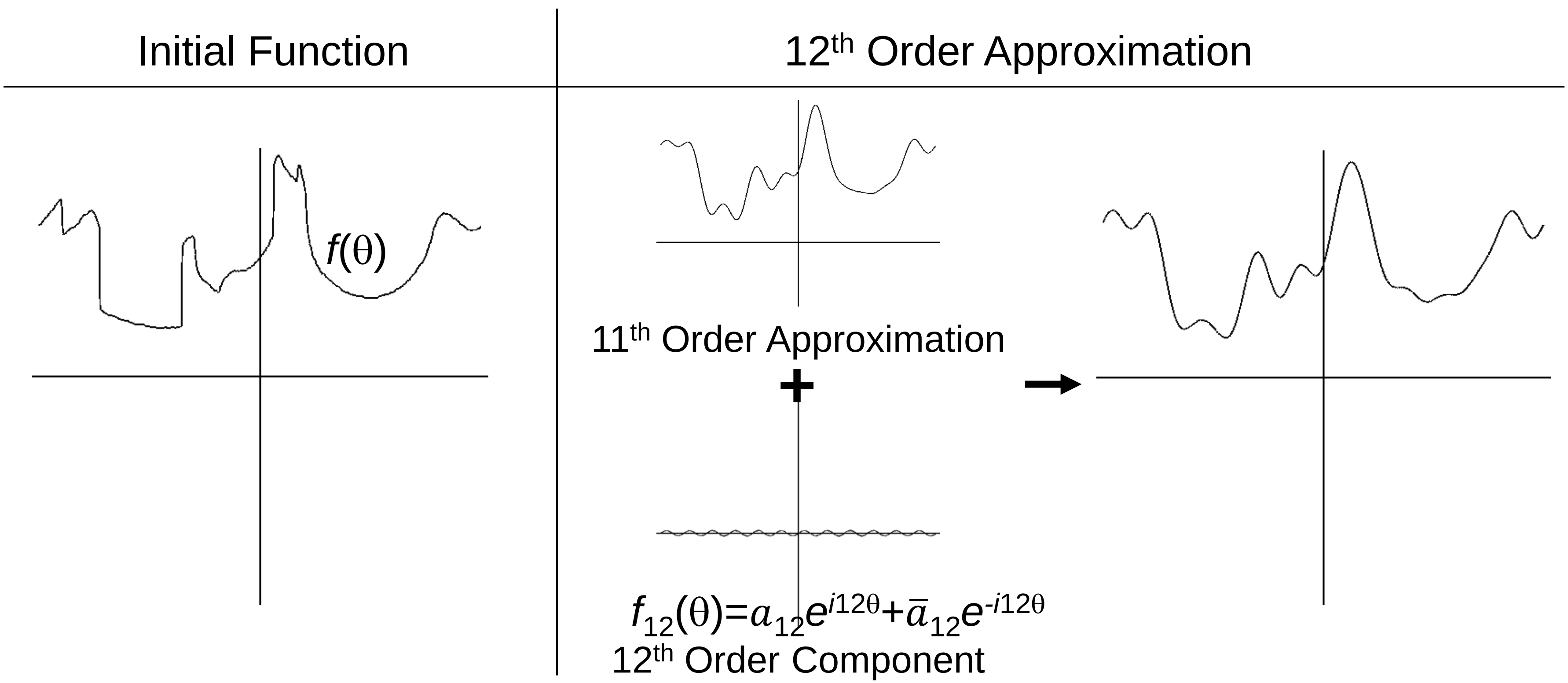
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



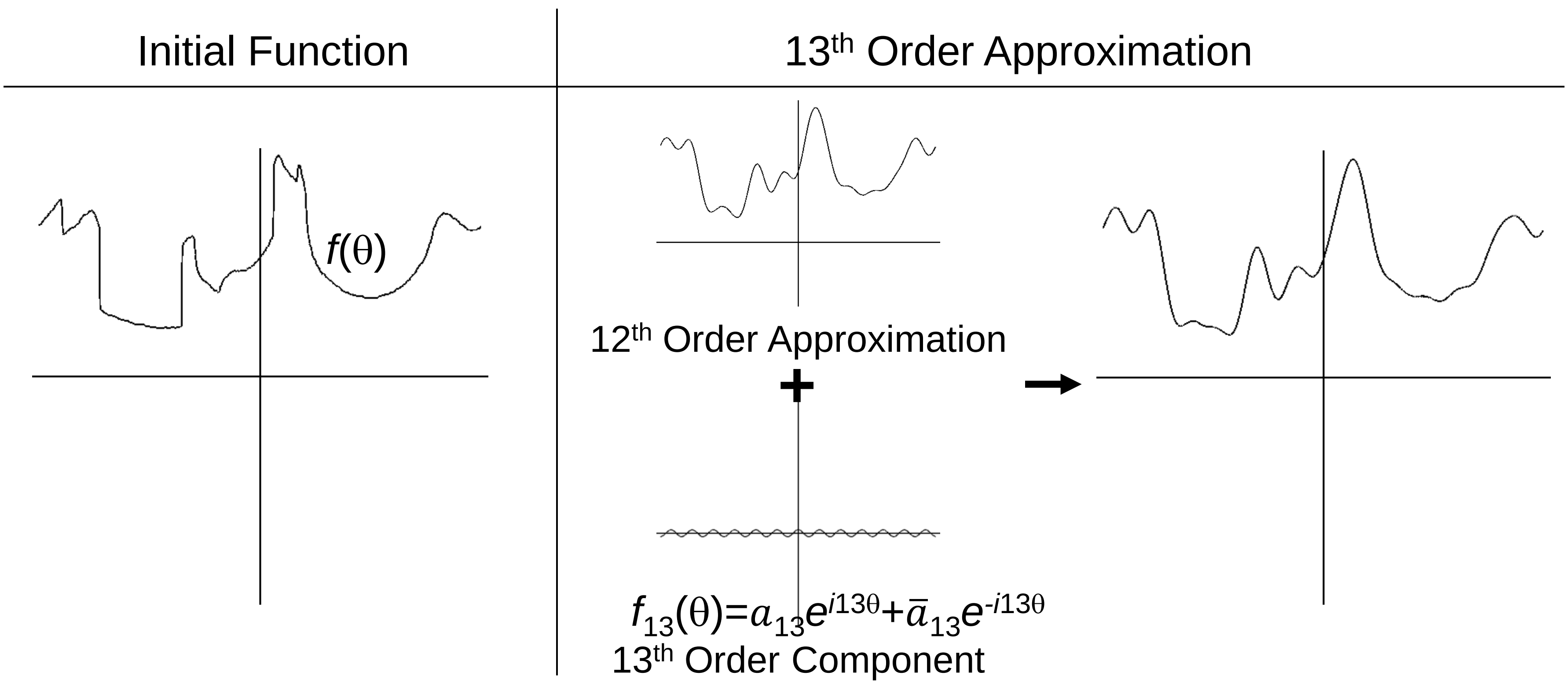
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



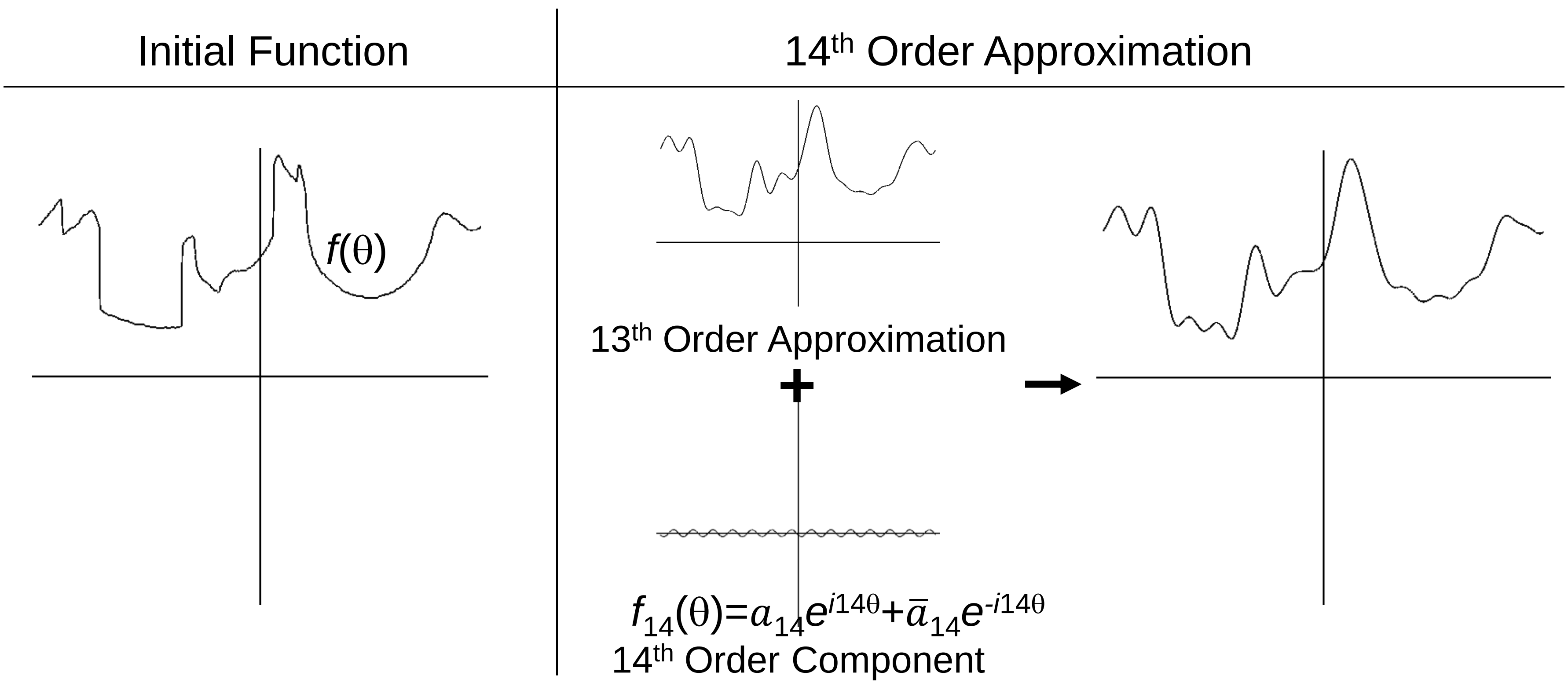
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



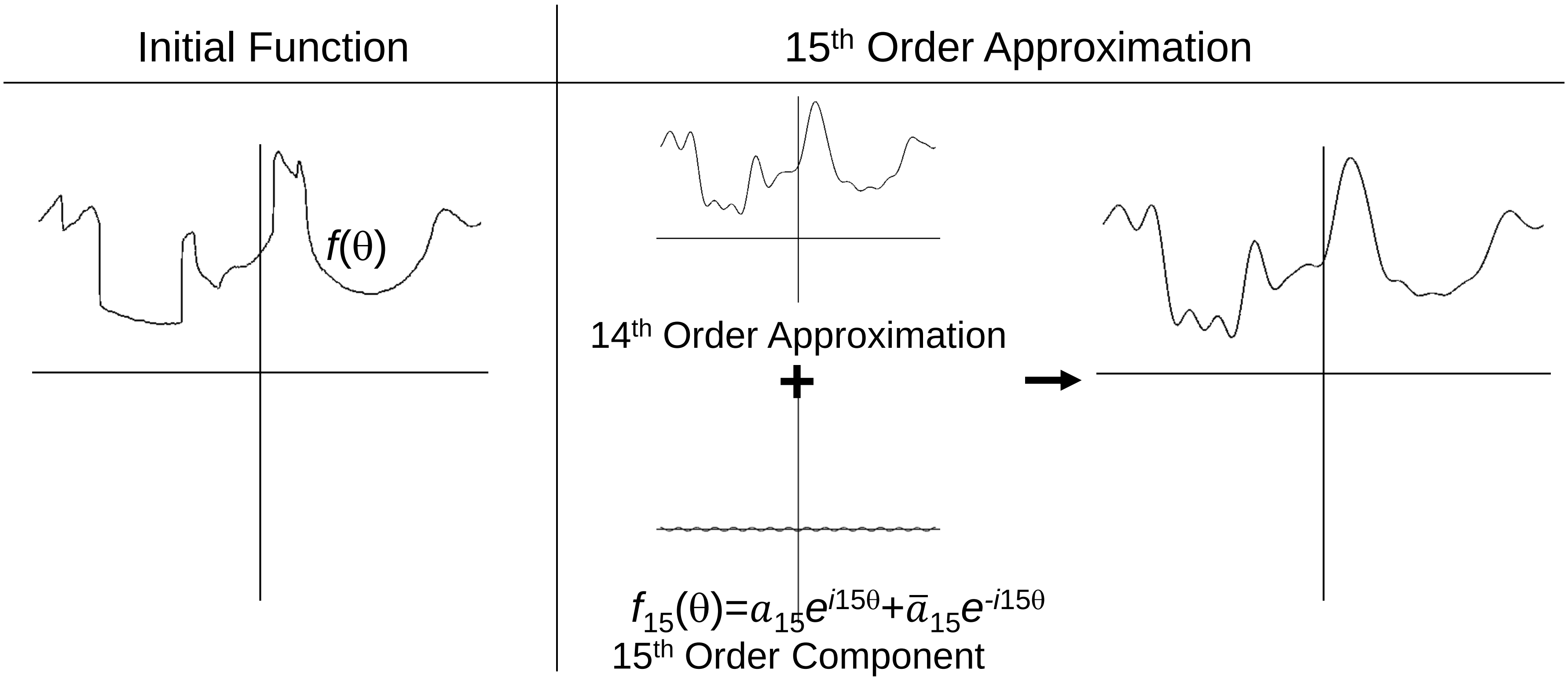
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



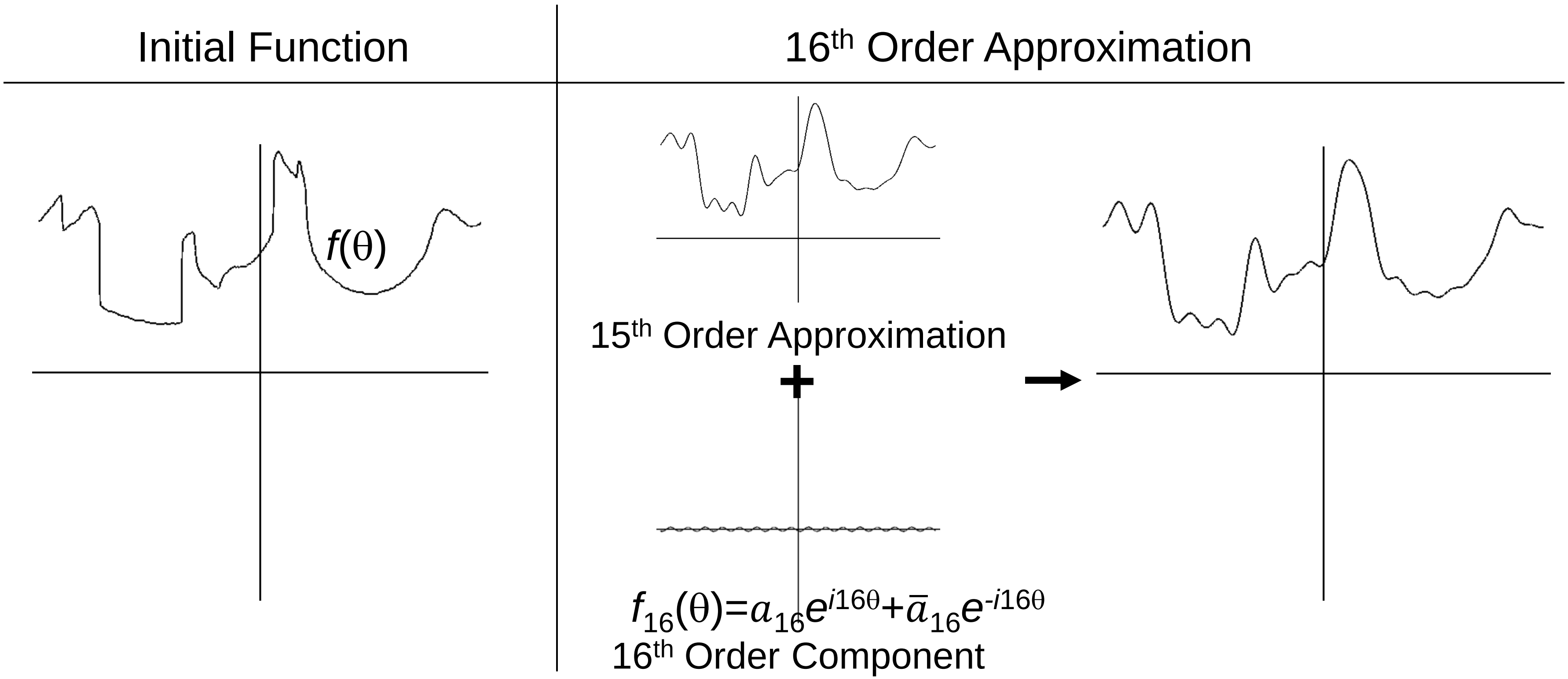
Fourier Analysis

- As higher frequency components are added to the approximation, finer details are captured.



Fourier Analysis

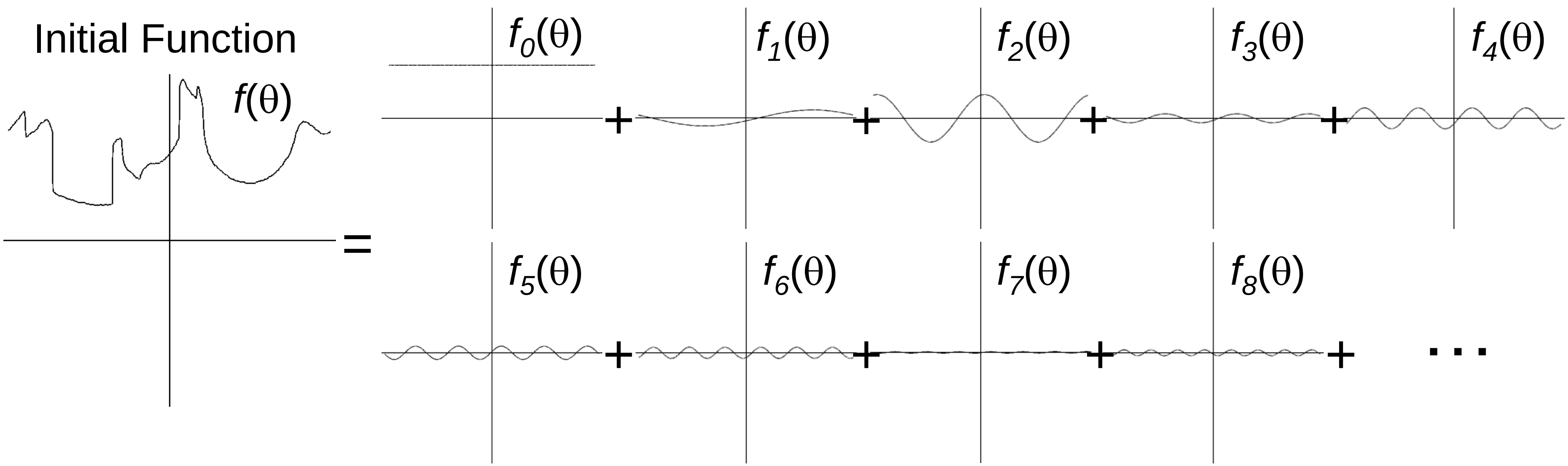
- As higher frequency components are added to the approximation, finer details are captured.



Fourier Analysis

- Combining all of the frequency components together, we get the initial function.

$$f(\theta) = \sum_{k=-\infty}^{\infty} f_k(\theta) = \sum_{k=-\infty}^{\infty} a_k \frac{e^{ik\theta}}{\sqrt{2\pi}}$$



Fourier Analysis

- Combining all of the frequency components together, we get the initial function.

$$f(\theta) = \sum_{k=-\infty}^{\infty} f_k(\theta) = \sum_{k=-\infty}^{\infty} a_k \frac{e^{ik\theta}}{\sqrt{2\pi}}$$

- To get the a_k, use the fact that the functions $e^{ik\theta}$ form an orthonormal basis and take the dot-product:

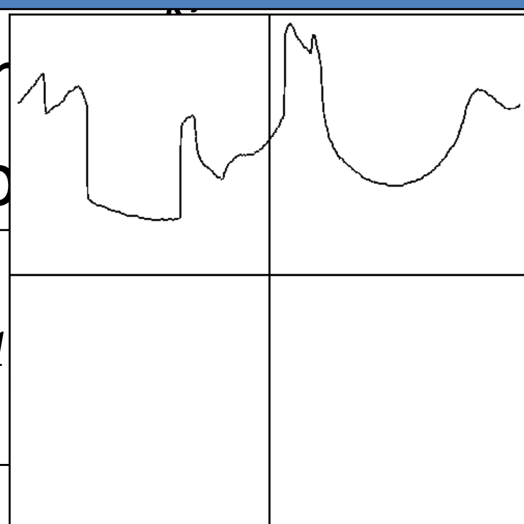
$$a_k = \left\langle f(\theta), \frac{e^{ik\theta}}{\sqrt{2\pi}} \right\rangle = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta$$

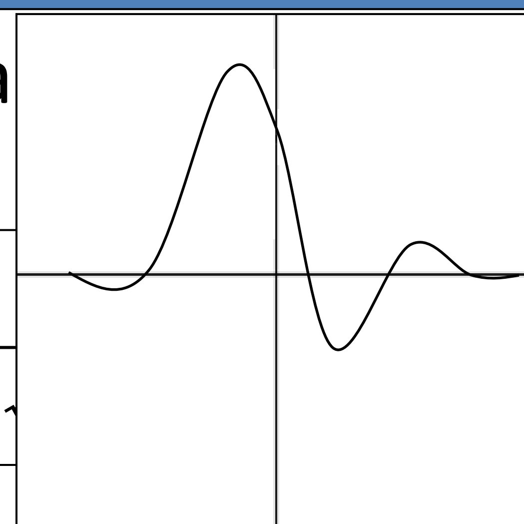
Fourier Analysis

- Combining all of the frequency components together, we get the initial function.

The mapping from the function *f* to its Fourier Coefficients is called the *Fourier Transform*.

- To g… …ons $e^{ik\theta}$ forr… …ha… …the dot-pro…

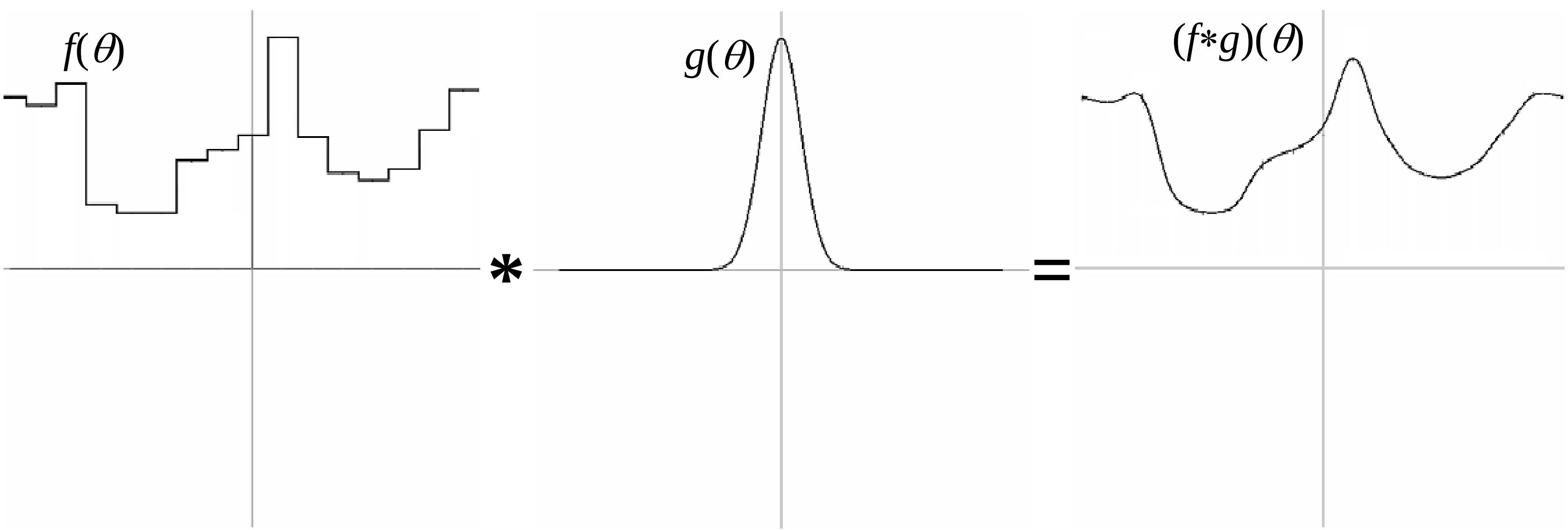




a … $= -$ … θ

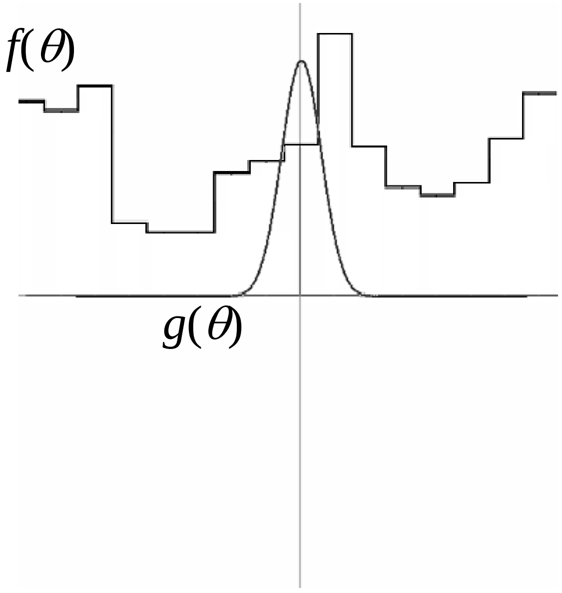
Convolution

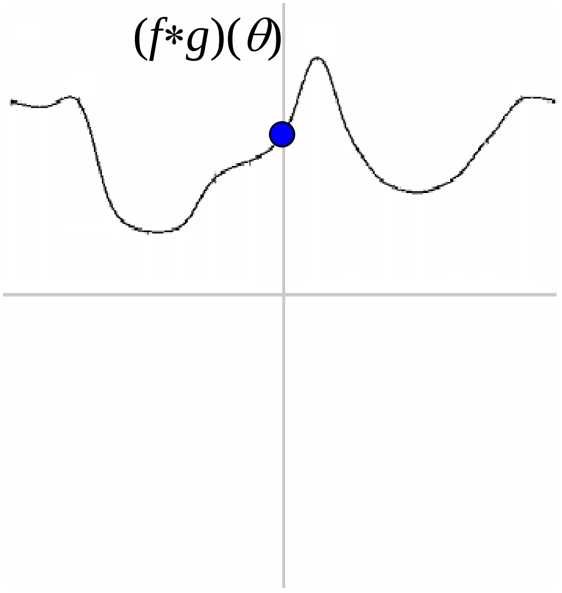
- To convolve functions *f* and *g*, we resample the function *f* using the weights given by (the reflection of) *g*.



Convolution

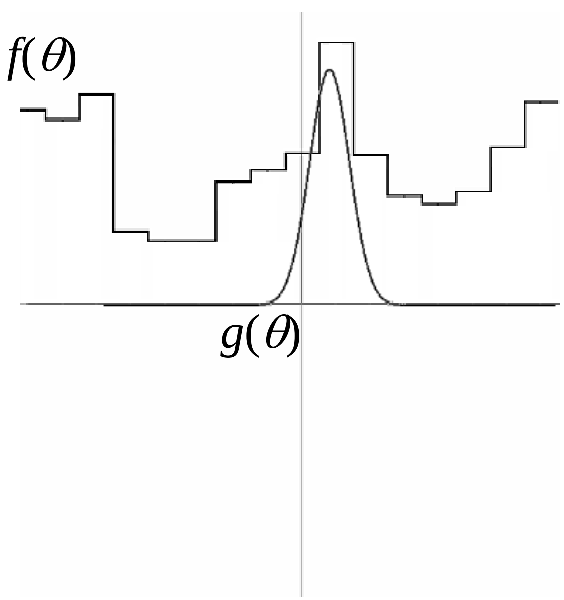
- To convolve functions *f* and *g*, we resample the function *f* using the weights given by (the reflection of) *g*.

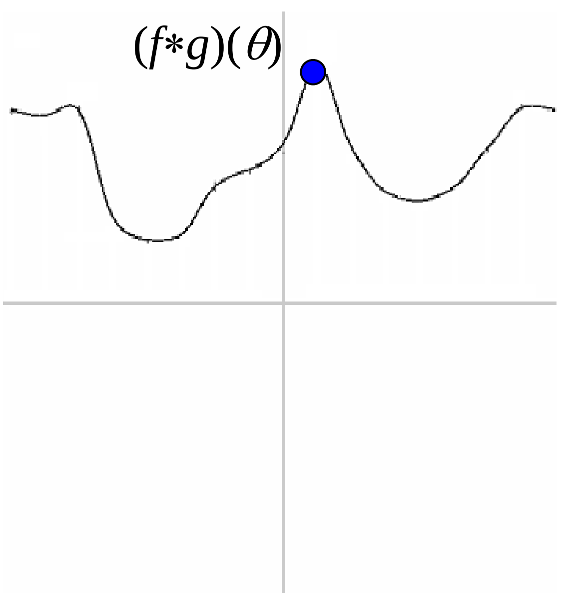




Convolution

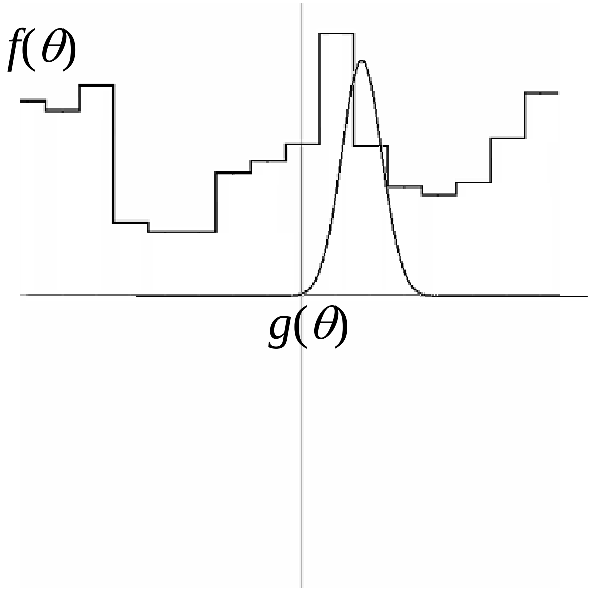
- To convolve functions *f* and *g*, we resample the function *f* using the weights given by (the reflection of) *g*.

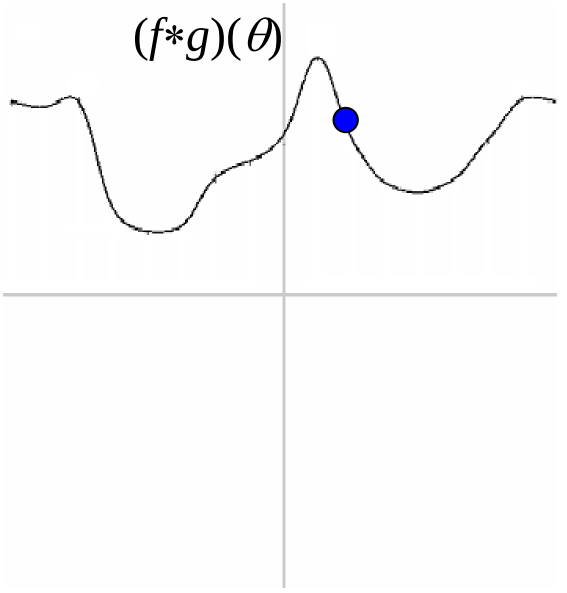




Convolution

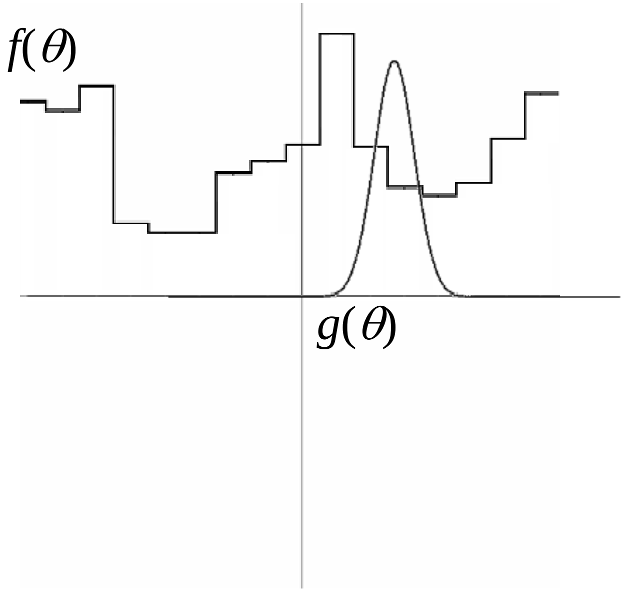
- To convolve functions *f* and *g*, we resample the function *f* using the weights given by (the reflection of) *g*.

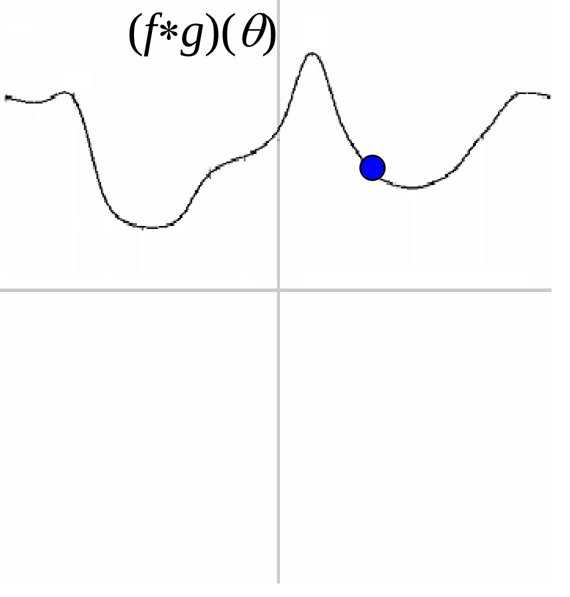




Convolution

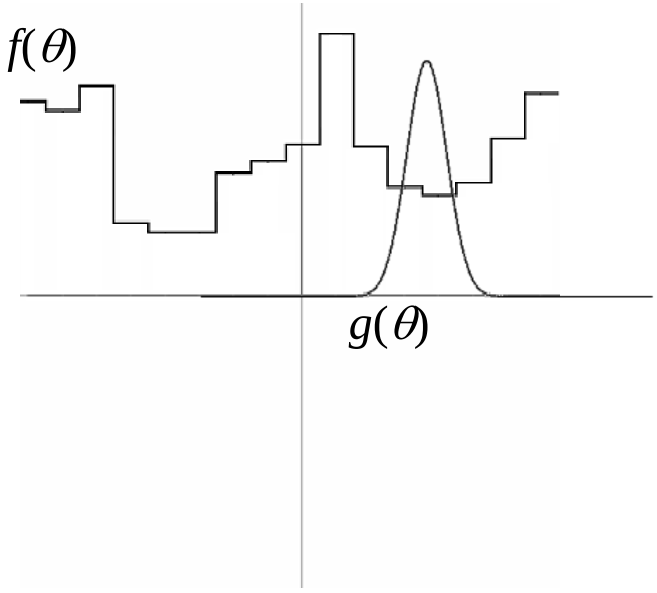
- To convolve functions *f* and *g*, we resample the function *f* using the weights given by (the reflection of) *g*.

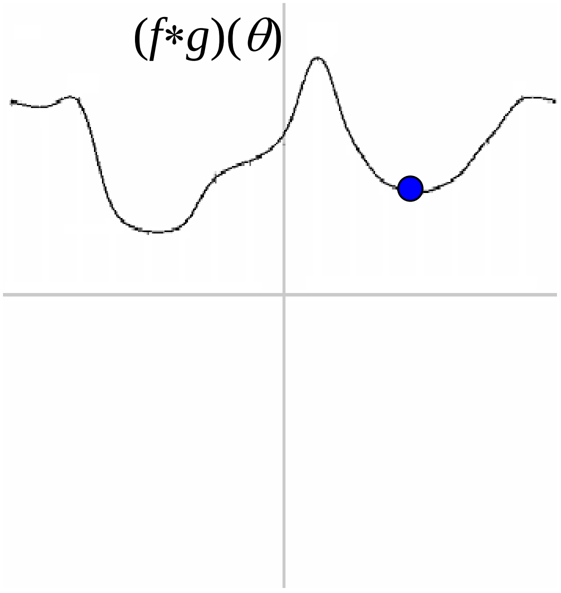




Convolution

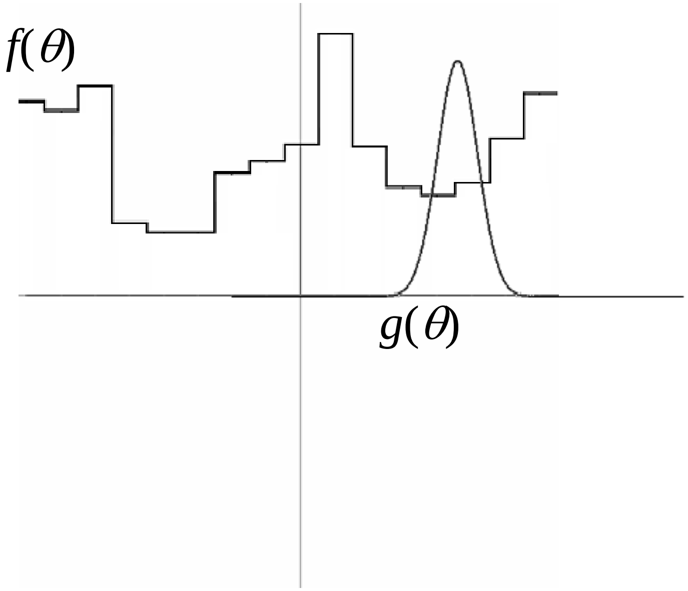
- To convolve functions *f* and *g*, we resample the function *f* using the weights given by (the reflection of) *g*.

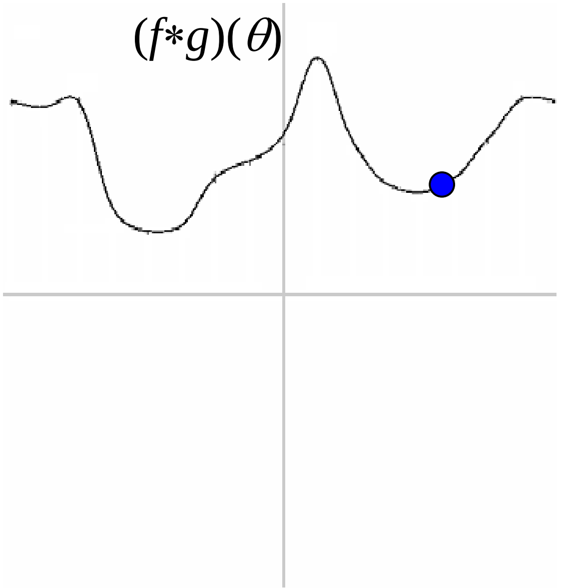




Convolution

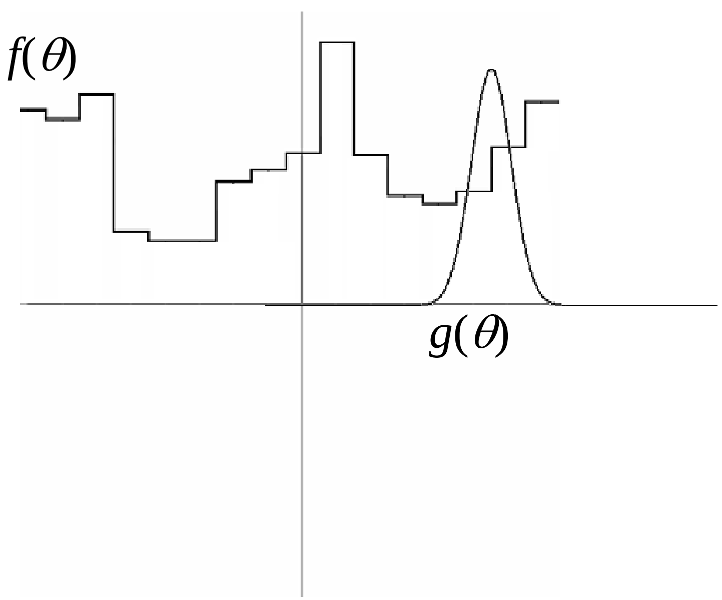
- To convolve functions *f* and *g*, we resample the function *f* using the weights given by (the reflection of) *g*.

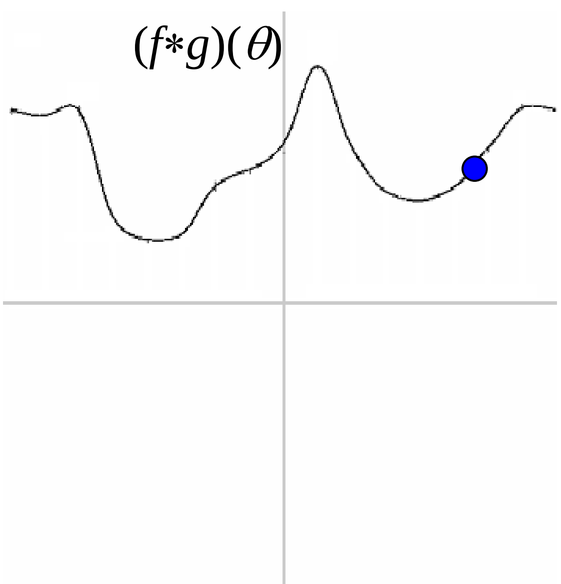




Convolution

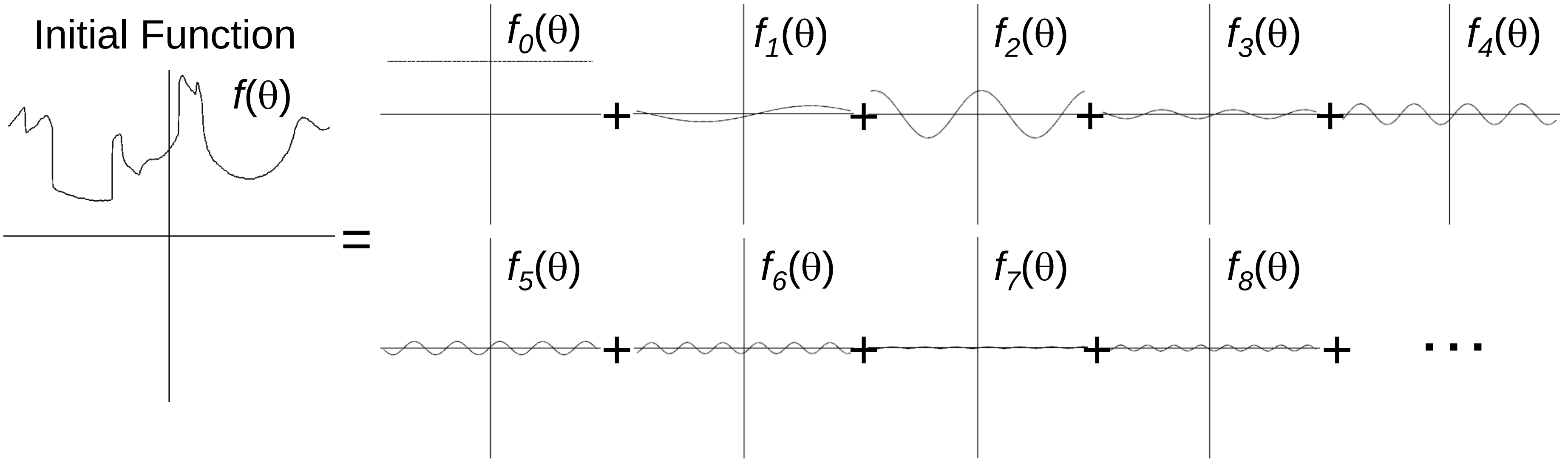
- To convolve functions *f* and *g*, we resample the function *f* using the weights given by (the reflection of) *g*.





Fourier Analysis

Q: What so special about the complex exponentials?



Fourier Analysis

A: Translating a complex exponential is the same as multiplying it:

$$e^{ik(\theta-\theta_0)} = e^{-ik\theta_0} \cdot e^{ik\theta}$$

Fourier Analysis

A: Translating a complex exponential is the same as multiplying it:

$$e^{ik(\theta-\theta_0)} = e^{-ik\theta_0} \cdot e^{ik\theta}$$

$\Rightarrow$Convolution in the spatial domain is multiplication in the frequency domain.

$$f(\theta) = \sum_{k=-\infty}^{\infty} a_k \frac{e^{ik\theta}}{\sqrt{2\pi}} \quad \text{and} \quad g(\theta) = \sum_{k=-\infty}^{\infty} b_k \frac{e^{ik\theta}}{\sqrt{2\pi}}$$

$$\Downarrow$$

$$(f * g)(\theta) = \sum_{k=-\infty}^{\infty} a_k b_k \frac{e^{ik\theta}}{\sqrt{2\pi}}$$

Fourier Analysis

A: Translating a complex exponential is the same as multiplying it:

$$e^{ik(\theta-\theta_0)} = e^{-ik\theta_0} \cdot e^{ik\theta}$$

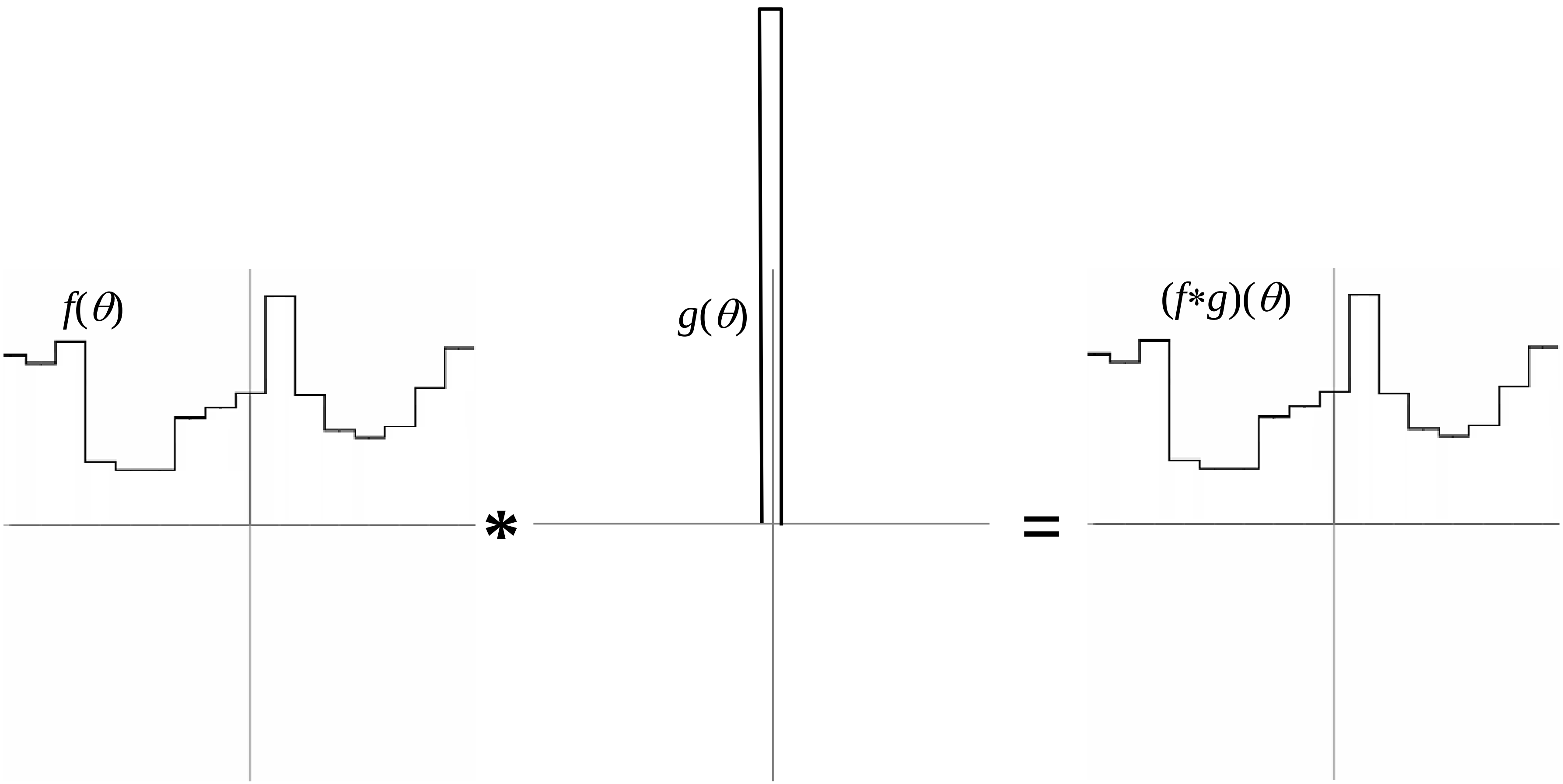
$\Rightarrow$Convolution in the spatial domain is multiplication in the frequency domain.

$\Rightarrow$Because the Fourier Transform is (almost) its own inverse, multiplication in the spatial domain is convolution in the frequency.

$$\left(f \cdot g\right)(\theta) = \sum_{k=-\infty}^{\infty} \left(a * b\right)_k \frac{e^{ik\theta}}{\sqrt{2\pi}}$$

Convolution

If *g* is a delta function (infinitely narrow, unit area), convolving with *g* preserves the signal.

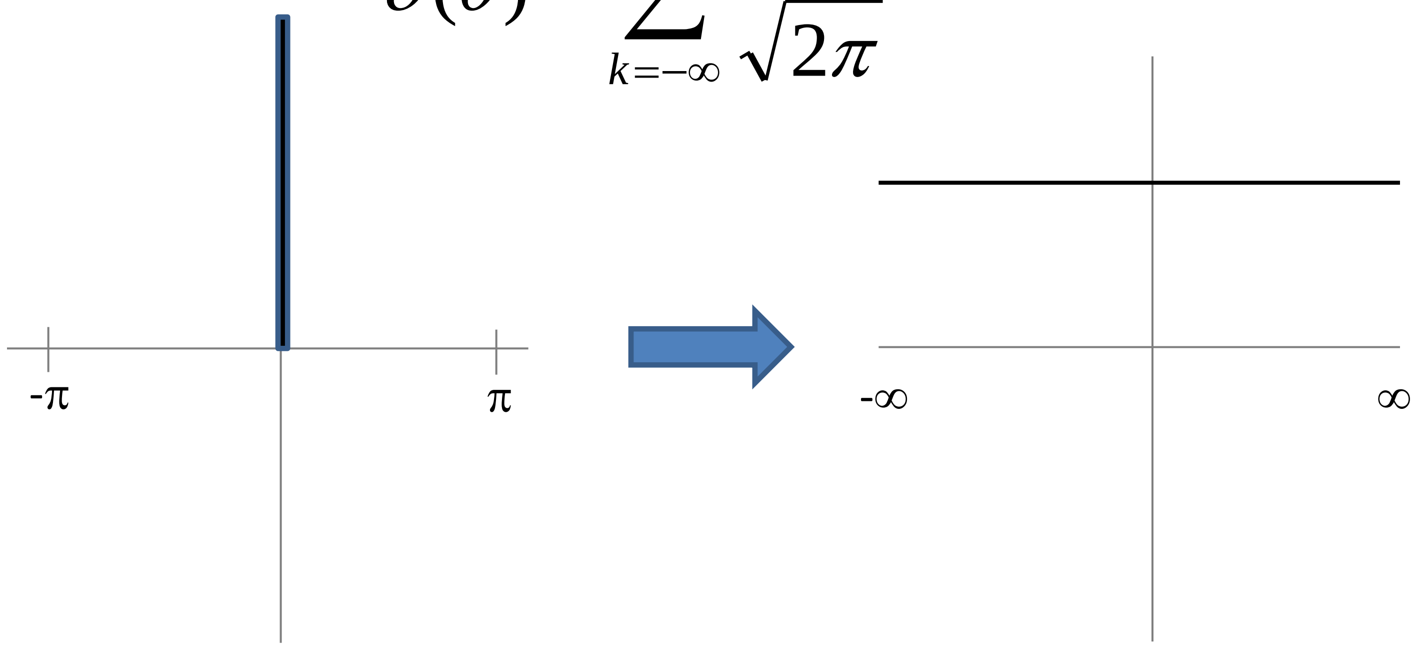


Convolution

If *g* is a delta function (infinitely narrow, unit area), convolving with *g* preserves the signal.

⇒The Fourier coefficients of the delta function are all ones:

$$\delta(\theta) = \sum_{k=-\infty}^{\infty} \frac{e^{ik\theta}}{\sqrt{2\pi}}$$



Convolution

If *g* is a delta function (infinitely narrow, unit area), convolving with *g* preserves the signal.

⇒The Fourier coefficients of the delta function are all ones:

$$\delta(\theta)=\sum_{k=-\infty}^{\infty}\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

⇒ The Fourier coefficients of a translated delta function are:

$$\delta(\theta-\theta_0)=\sum_{k=-\infty}^{\infty}e^{-ik\theta_0}\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

Impulse Trains

⇒The Fourier coefficients of the sum of two evenly spaced delta function are:

$$\delta(\theta+\pi)+\delta(\theta)=\sum_{k=-\infty}^{\infty}\frac{e^{ik\theta}}{\sqrt{2\pi}}+\sum_{k=-\infty}^{\infty}e^{ik\pi}\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

Impulse Trains

$\Rightarrow$The Fourier coefficients of the sum of two evenly spaced delta function are:

$$\delta(\theta+\pi)+\delta(\theta)=\sum_{k=-\infty}^{\infty}\frac{e^{ik\theta}}{\sqrt{2\pi}}+\sum_{k=-\infty}^{\infty}e^{ik\pi}\,\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

$$=\sum_{k=-\infty}^{\infty}\left(1+e^{ik\pi}\right)\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

Impulse Trains

$\Rightarrow$The Fourier coefficients of the sum of two evenly spaced delta function are:

$$\delta(\theta+\pi)+\delta(\theta)=\sum_{k=-\infty}^{\infty}\frac{e^{ik\theta}}{\sqrt{2\pi}}+\sum_{k=-\infty}^{\infty}e^{ik\pi}\,\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

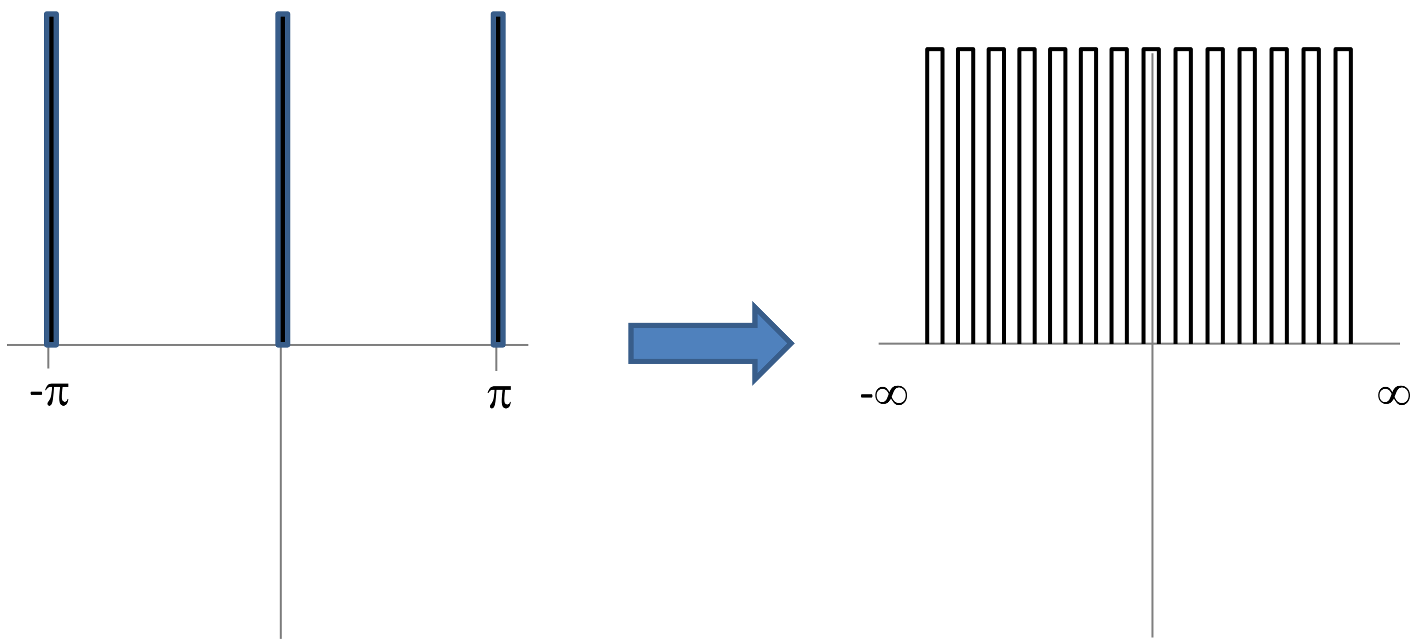
$$=\sum_{k=-\infty}^{\infty}\left(1+e^{ik\pi}\right)\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

$$=\sum_{k=-\infty}^{\infty}\left\{\begin{matrix}2 & \text{if } k \text{ is even}\\ 0 & \text{if } k \text{ is odd}\end{matrix}\right\}\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

Impulse Trains

⇒The Fourier coefficients of the sum of two evenly spaced delta function are:

$$\delta(\theta+\pi)+\delta(\theta)=\sum_{k=-\infty}^{\infty}\begin{Bmatrix}2 & \text{if } k \text{ is even} \\ 0 & \text{if } k \text{ is even}\end{Bmatrix}\frac{e^{ik\theta}}{\sqrt{2\pi}}$$



Impulse Trains

$\Rightarrow$The Fourier coefficients of the sum of *N* evenly spaced delta function are:

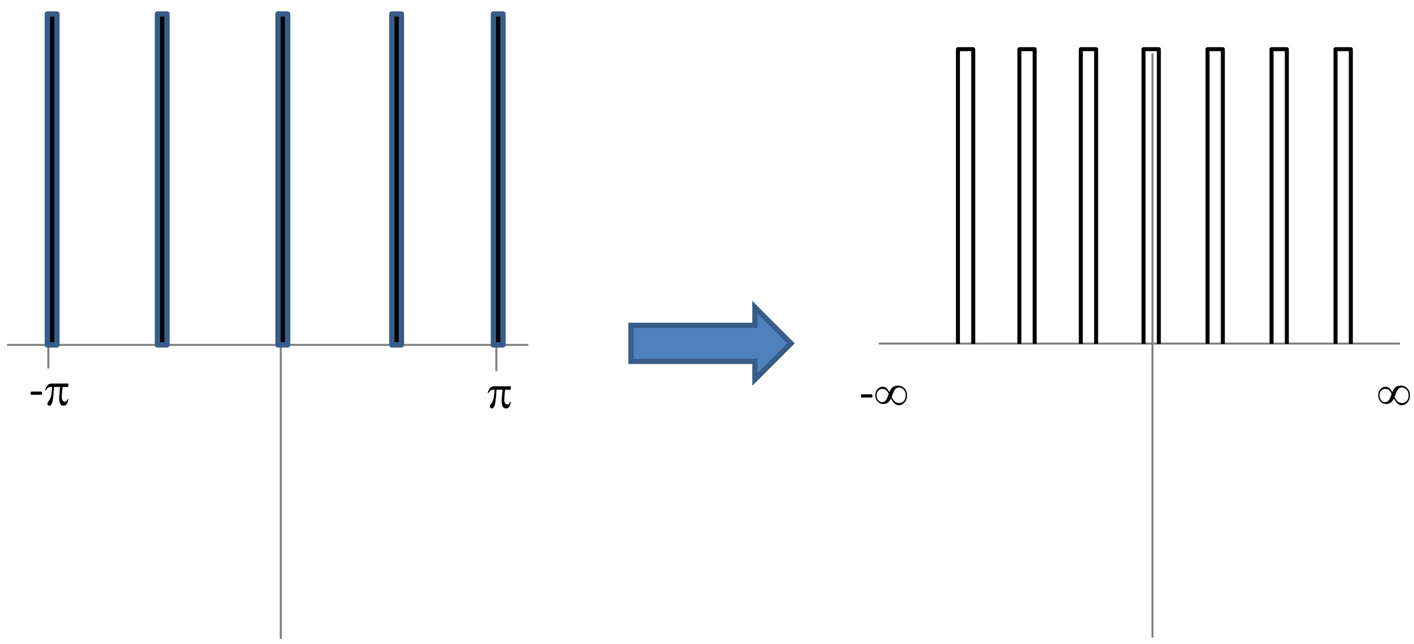
$$\sum_{j=0}^{N-1}\delta\left(\theta+\pi-\frac{2\pi j}{N}\right)=\sum_{k=-\infty}^{\infty}e^{ik\pi}\left(\sum_{j=0}^{N-1}e^{-ik\frac{2\pi j}{N}}\right)\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

$$=(-1)^N\sum_{k=-\infty}^{\infty}\begin{Bmatrix}N & \text{if } N \text{ divides } k\\ 0 & \text{otherwise}\end{Bmatrix}\frac{e^{ik\theta}}{\sqrt{2\pi}}$$

Impulse Trains

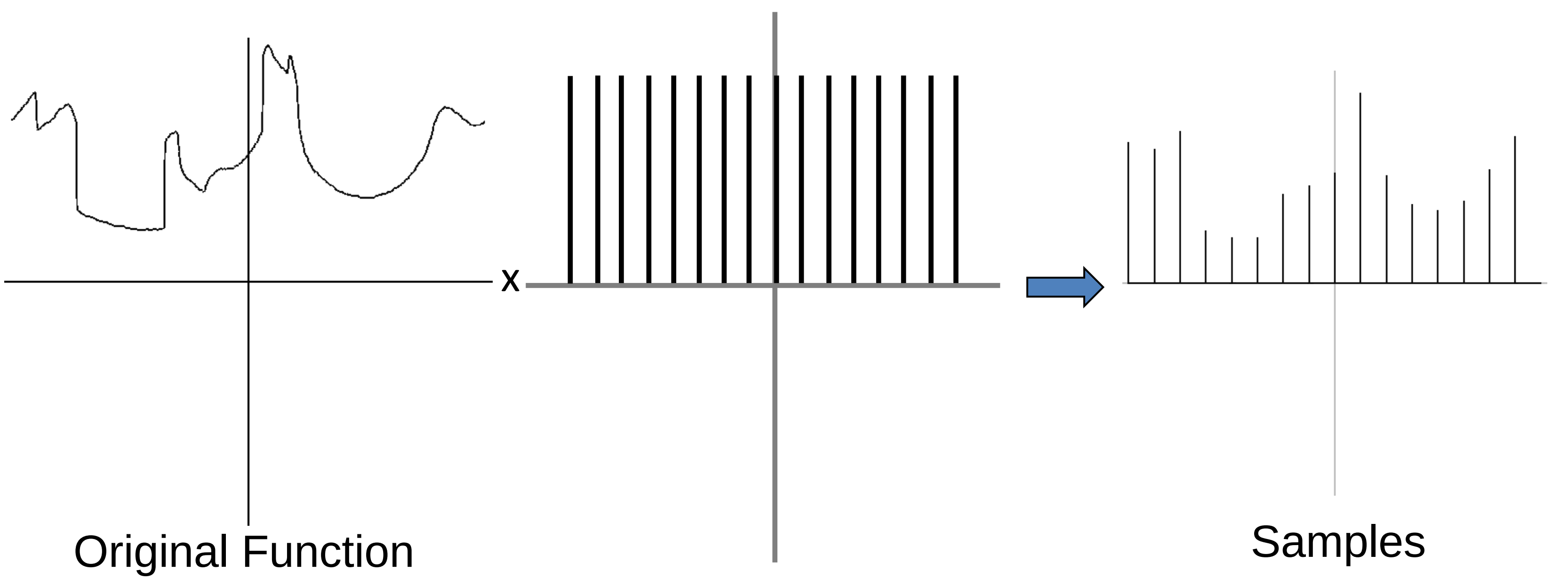
⇒The Fourier coefficients of the sum of *N* evenly spaced delta function are:

$$\sum_{j=0}^{N-1} \delta\left(\theta + \pi - \frac{2\pi j}{N}\right) = (-1)^N \sum_{k=-\infty}^{\infty} \left\{\begin{matrix} N & \text{if } N \text{ divides } k \\ 0 & \text{otherwise} \end{matrix}\right\} \frac{e^{ik\theta}}{\sqrt{2\pi}}$$



Impulse Trains

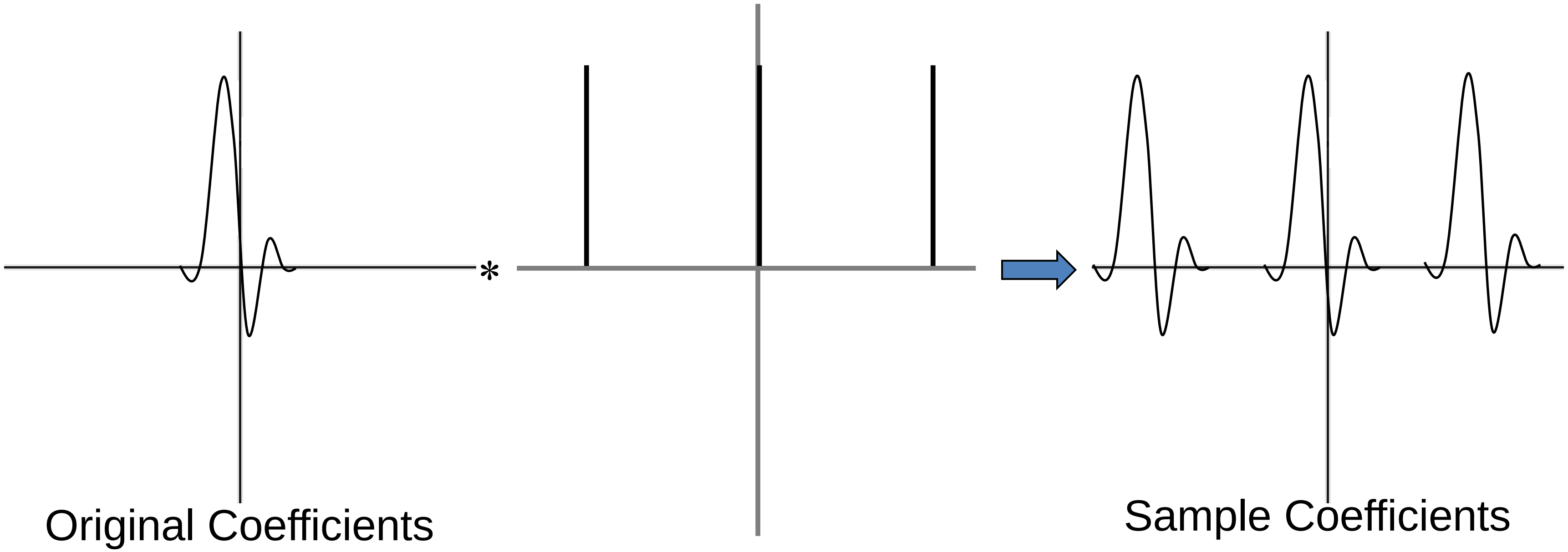
We can express the sampling of a signal as multiplication of the signal by an impulse train.



Impulse Trains

We can express the sampling of a signal as multiplication of the signal by an impulse train.

But multiplication $\Rightarrow$ convolution

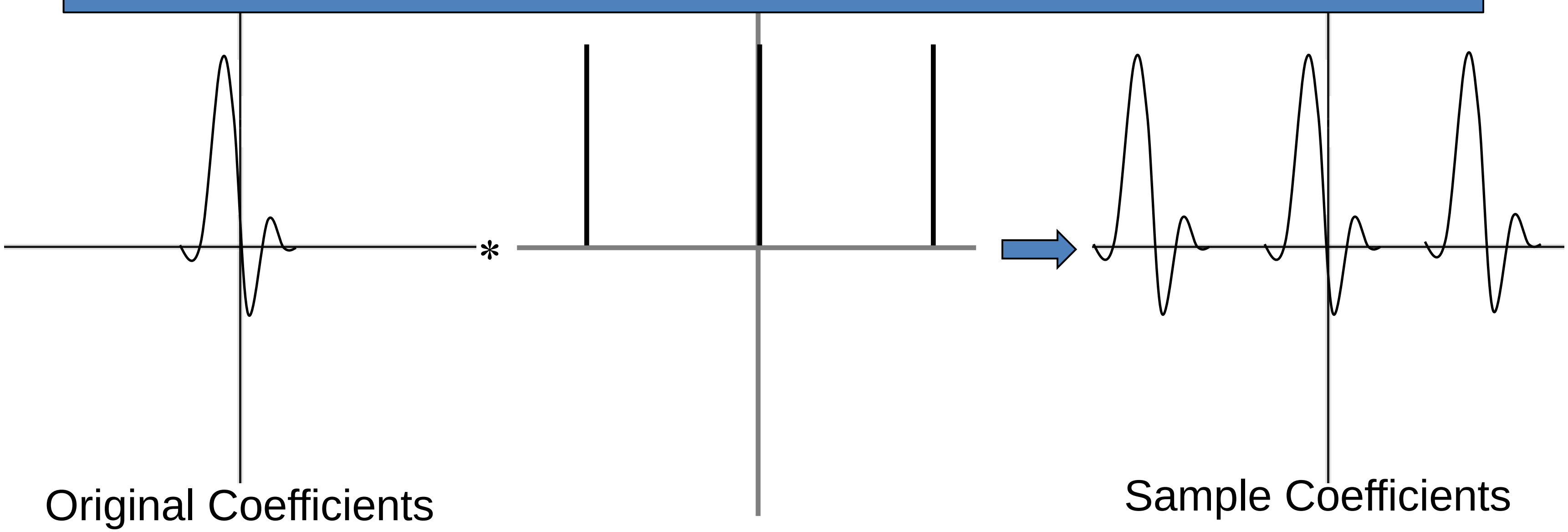


W
m
Bu

<u>Note</u>:

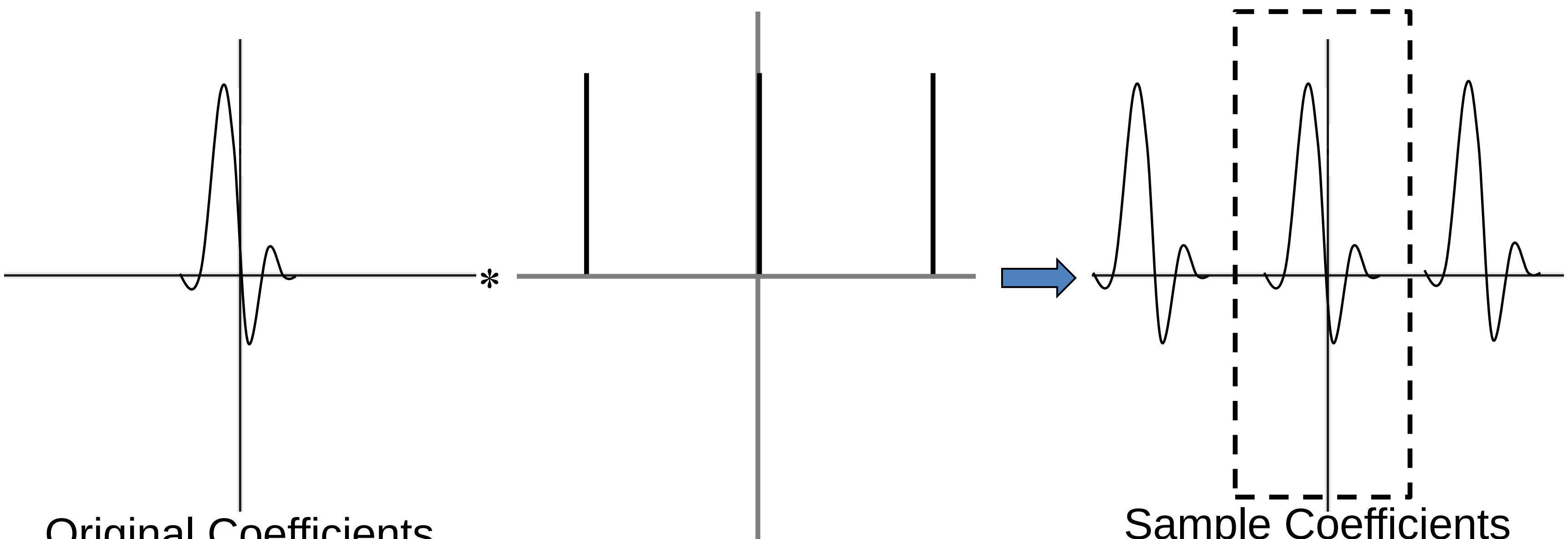
The sample coefficients are disjoint copies of the signal if:

1. The signal is band-limited (Fourier coefficients are zero beyond some point)
2. The sampling between impulses in the frequency domain is sufficiently far apart (sampling is fine enough).



Impulse Trains

If the sampling conditions are satisfied, we can reconstruct by convolving with a filter that pulls out the center of the spectrum.



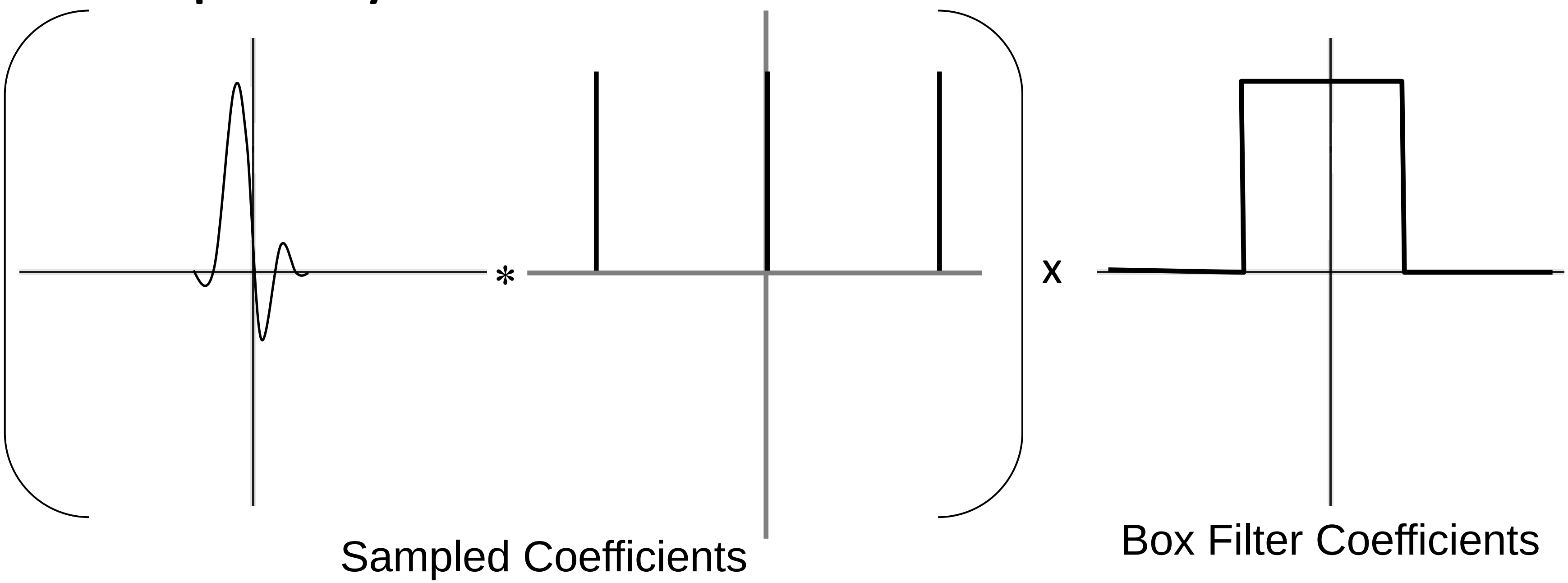
Original Coefficients

Sample Coefficients

Impulse Trains

If the sampling conditions are satisfied, we can reconstruct by convolving with a filter that pulls out the center of the spectrum.

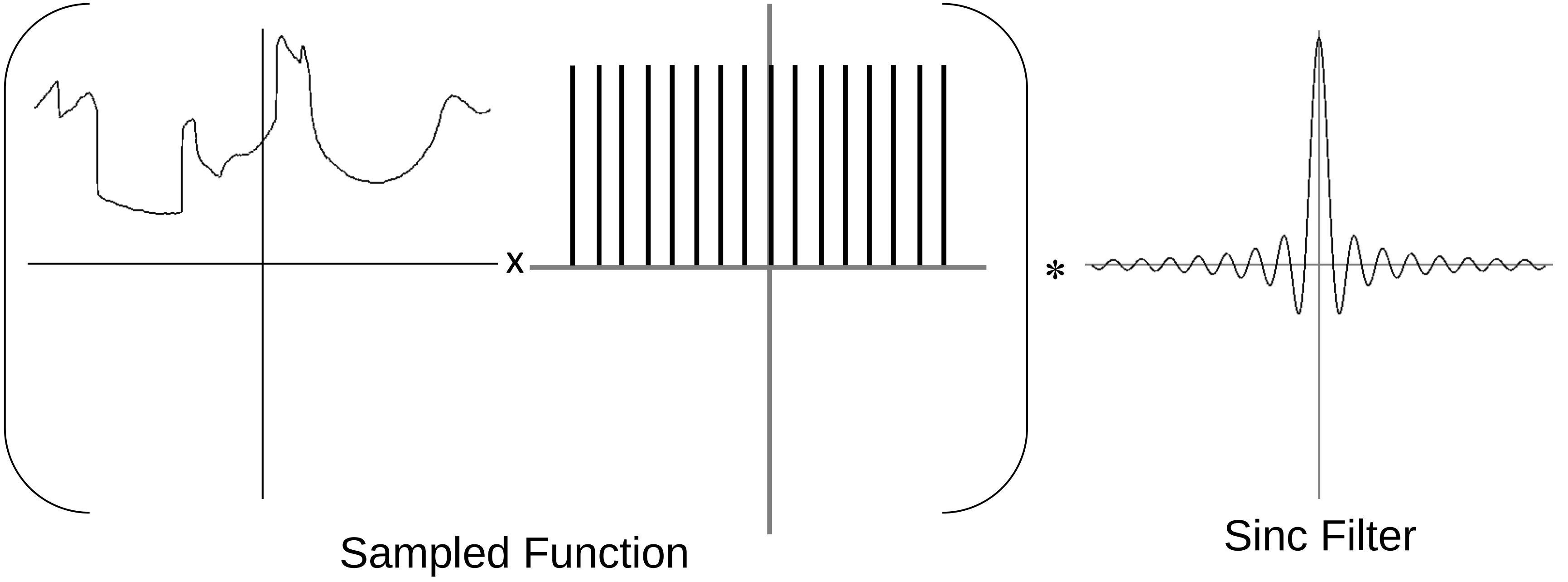
$\Rightarrow$ Want to multiply by a box filter in the frequency domain.



Impulse Trains

If the sampling conditions are satisfied, we can reconstruct by convolving with a filter that pulls out the center of the spectrum.

$\Rightarrow$ Want to multiply by a box filter in the frequency domain $\Leftrightarrow$ Convolve with a sinc.



Sampling / Reconstruction

We are assuming that the input signal is band-limited and the sampling is fine enough.

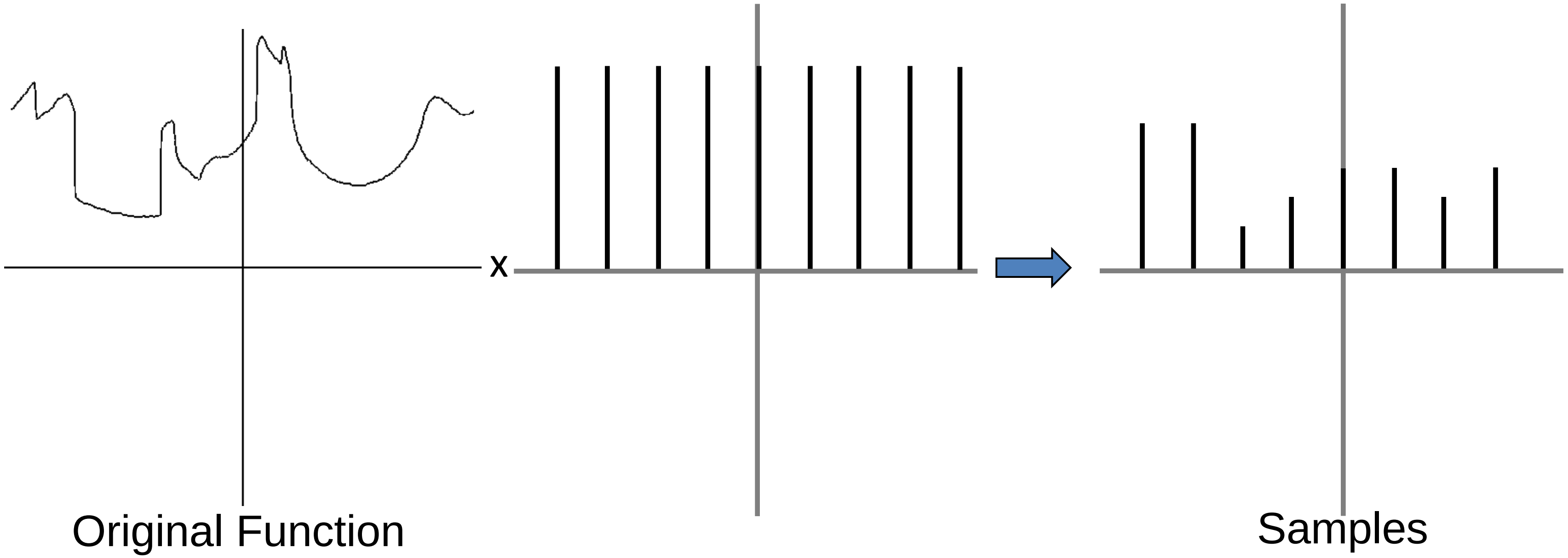
In practice, this assumption is false:

- The signal is not band-limited (occluding contours, sharp shadow boundaries, etc.)
- We are limited in the extent to which we can sample.

Sampling / Reconstruction

We are assuming that the input signal is band-limited and the sampling is fine enough.

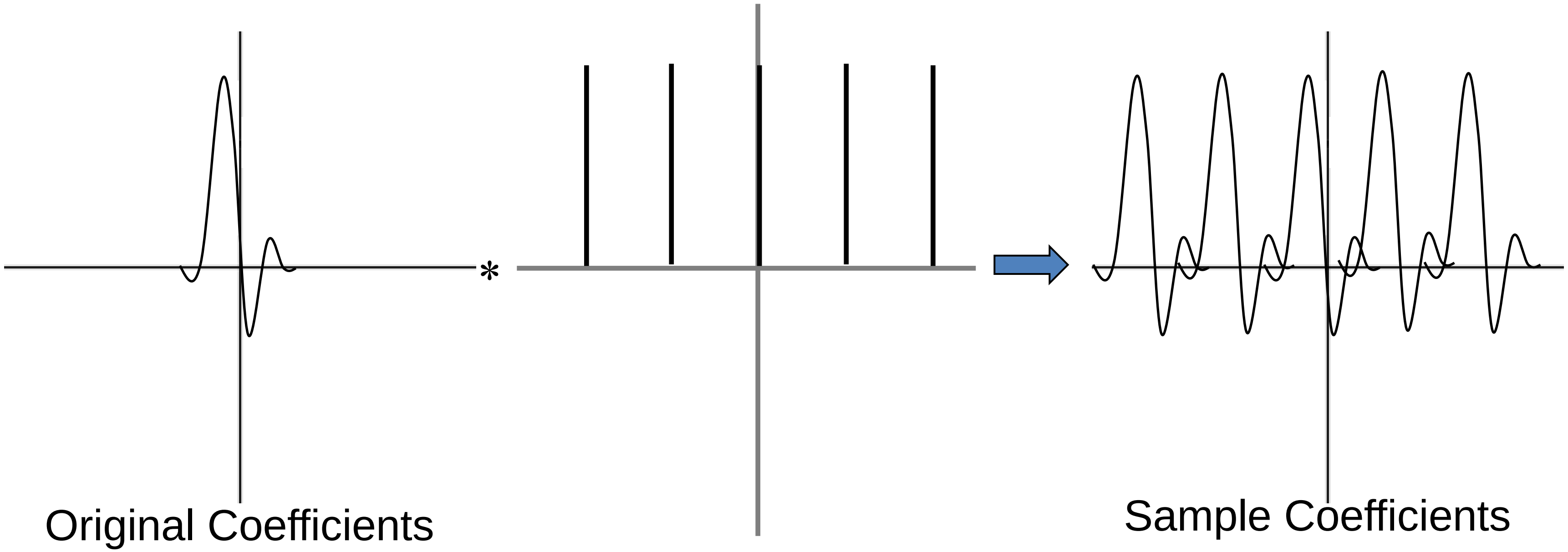
In practice, this assumption is false.



Sampling / Reconstruction

We are assuming that the input signal is band-limited and the sampling is fine enough.

In practice, this assumption is false.

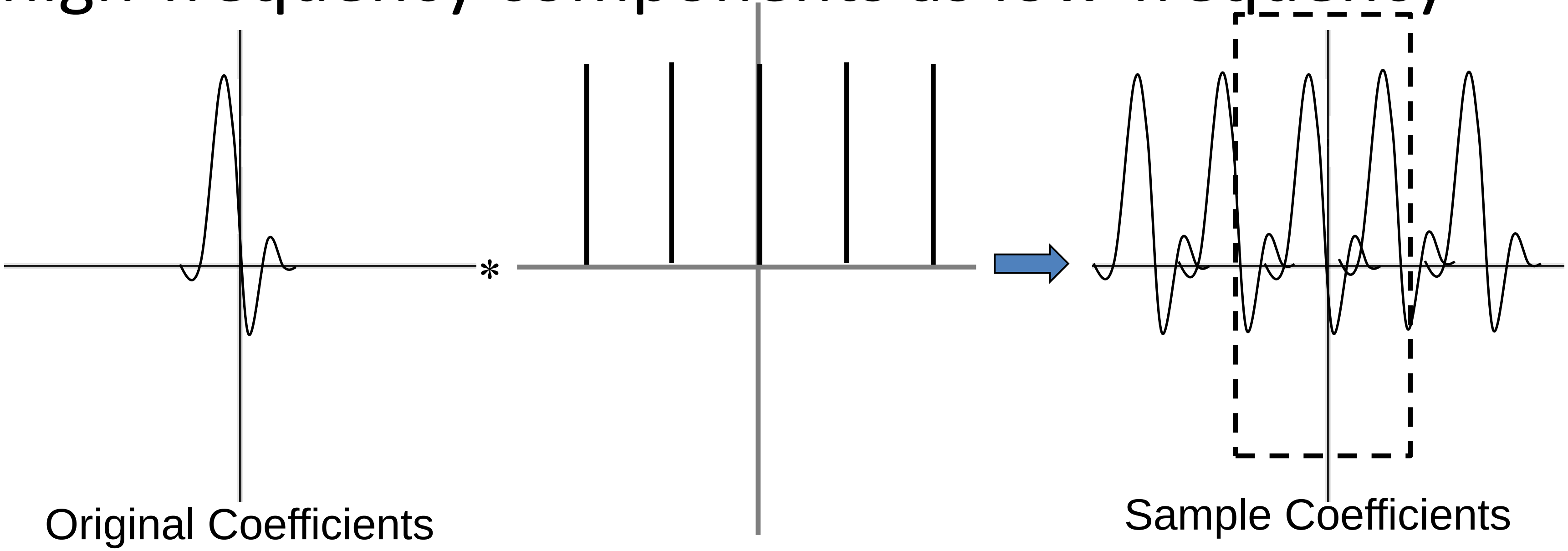


Sampling / Reconstruction

We are assuming that the input signal is band-limited and the sampling is fine enough.

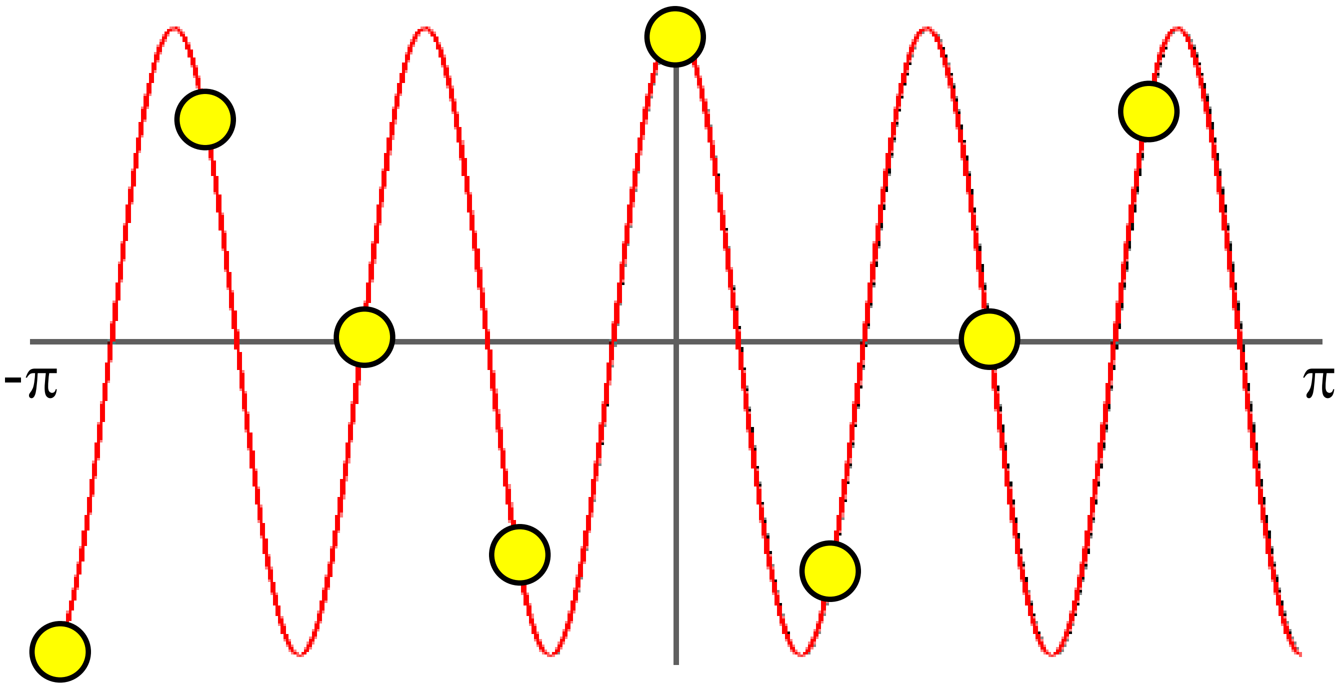
In practice, this assumption is false.

Multiplying with a box function, we pick up high-frequency components as low-frequency



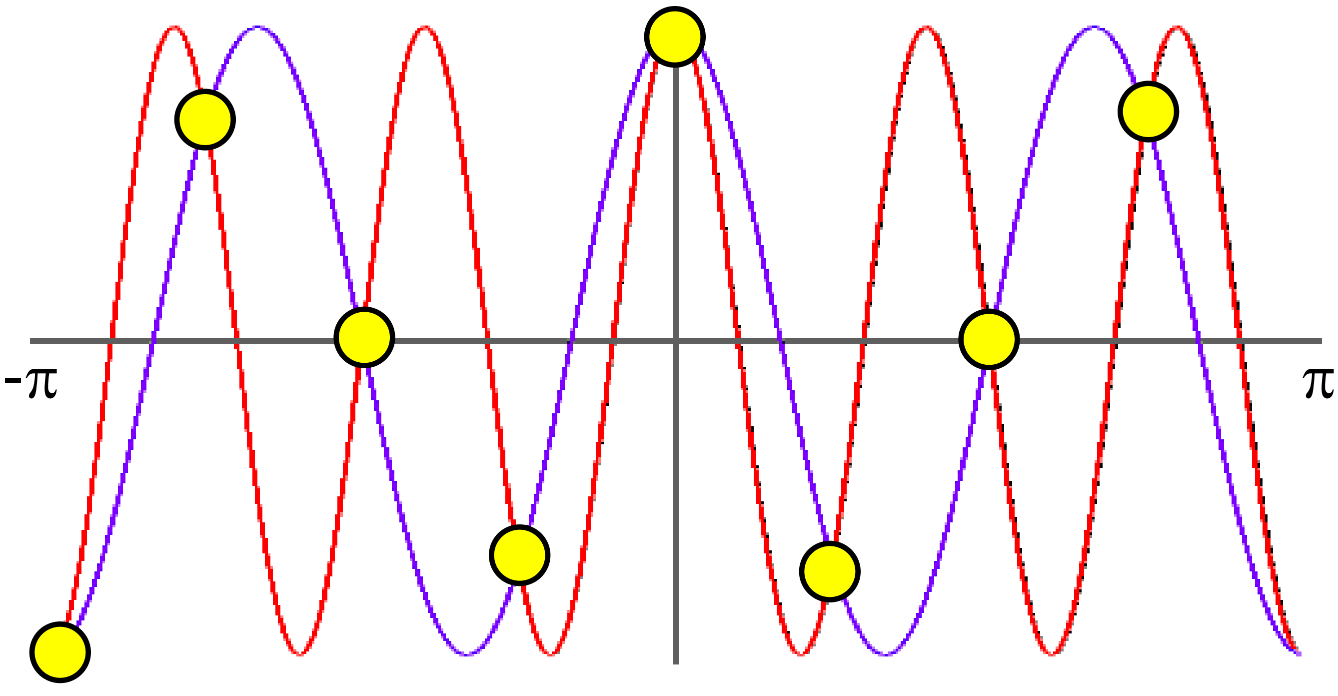
Aliasing

- When a high-frequency signal is sampled with insufficiently many samples, it will be perceived as a lower-frequency signal. This masking of higher frequencies as lower ones is referred to as **aliasing**.



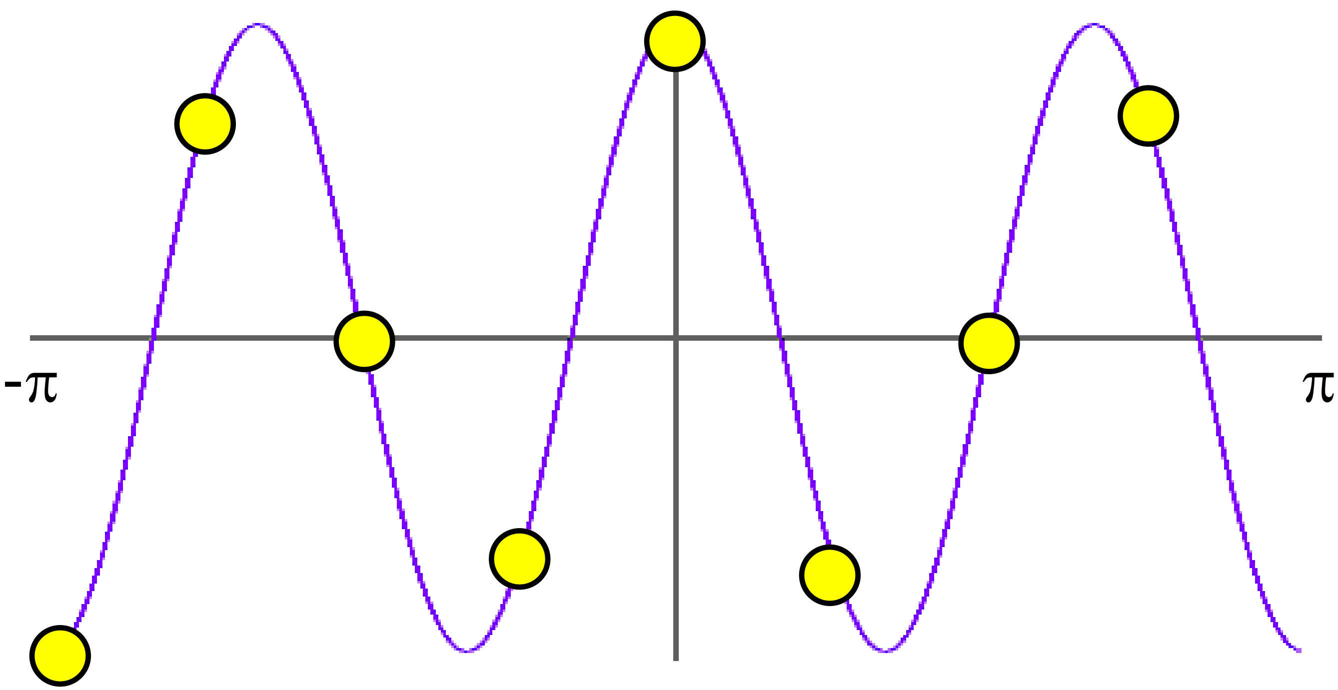
Aliasing

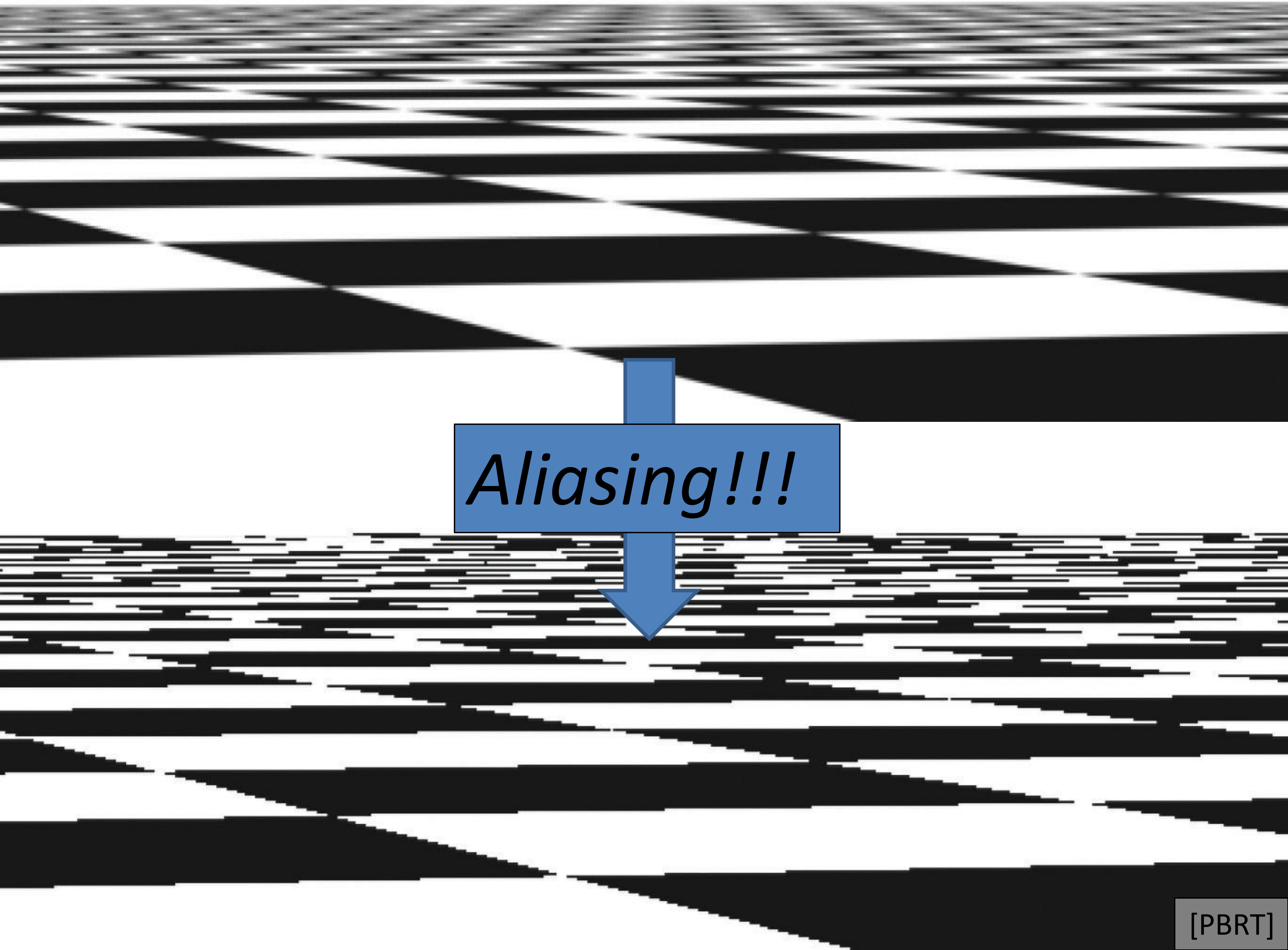
- When a high-frequency signal is sampled with insufficiently many samples, it will be perceived as a lower-frequency signal. This masking of higher frequencies as lower ones is referred to as **aliasing**.



Aliasing

- When a high-frequency signal is sampled with insufficiently many samples, it will be perceived as a lower-frequency signal. This masking of higher frequencies as lower ones is referred to as **aliasing**.




Aliasing!!!
[PBRT]

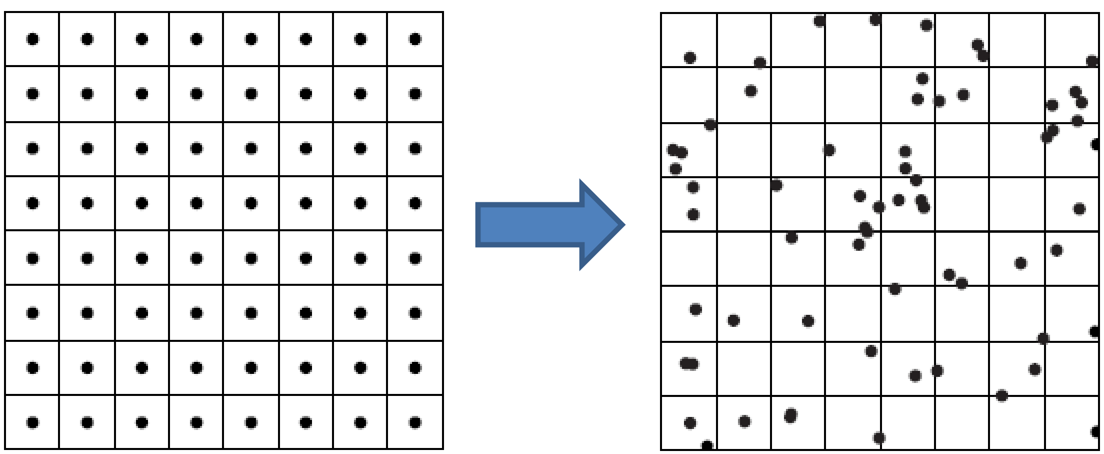
Sampling / Reconstruction

Since we can’t increase the sampling rate beyond the necessary (Nyquist) frequency, our samples are bound to contain high-frequency info.

Sampling / Reconstruction

Since we can't increase the sampling rate beyond the necessary (Nyquist) frequency, our samples are bound to contain high-frequency info.

Try to remove the effects of aliasing by randomizing the sampling positions.



Sampling / Reconstruction

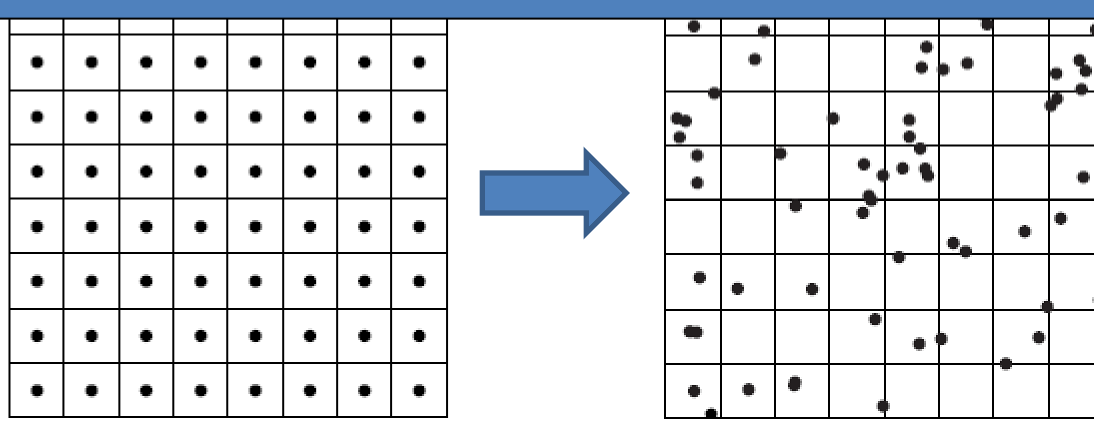
Since we can't increase the sampling rate beyond the necessary (Nyquist) frequency, our samples are bound to contain high-frequency info.

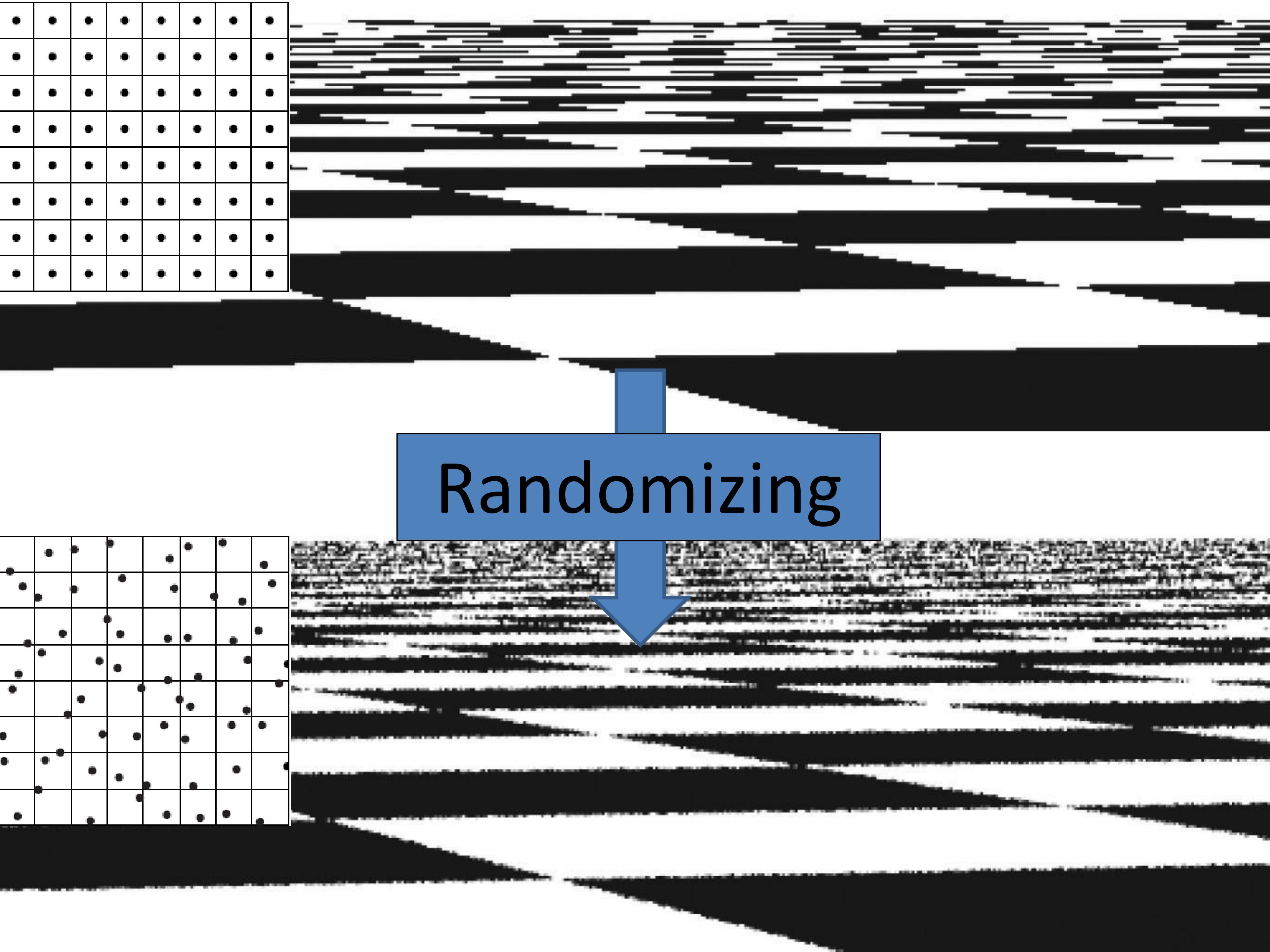
T

ra

We still sample the high frequency, but we now de-correlate the phase alignment.

Aliasing $\Rightarrow$ Noise

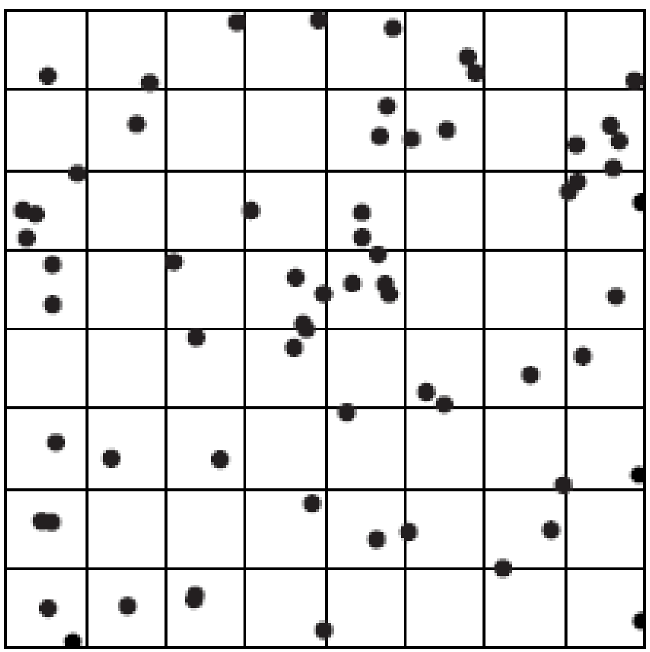


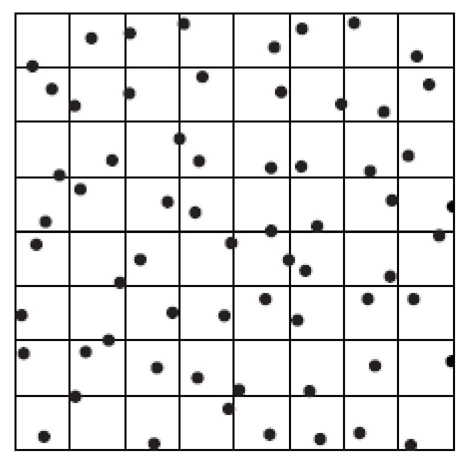

Randomizing

Random Sampling

Challenge:

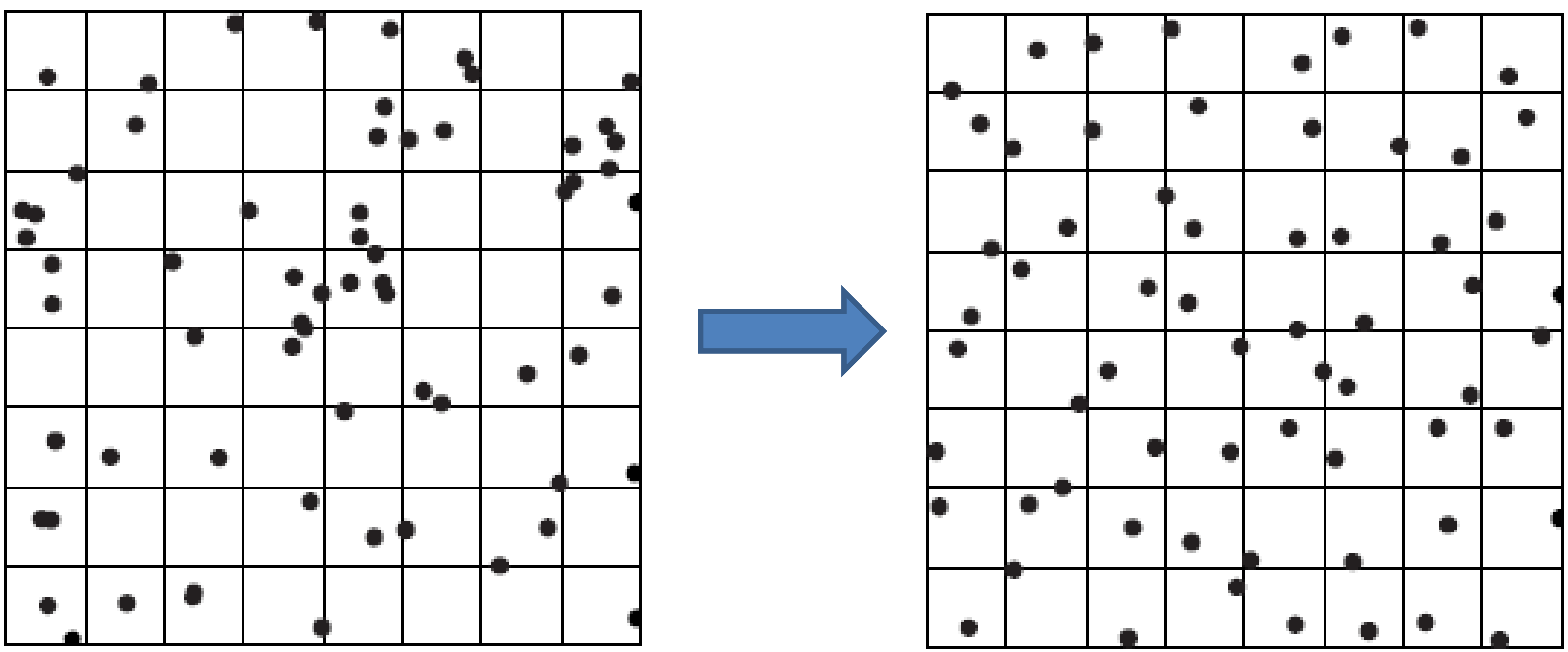
If we randomly sample the plane, we are likely to get some regions that are over-sampled and others that are under-sampled.





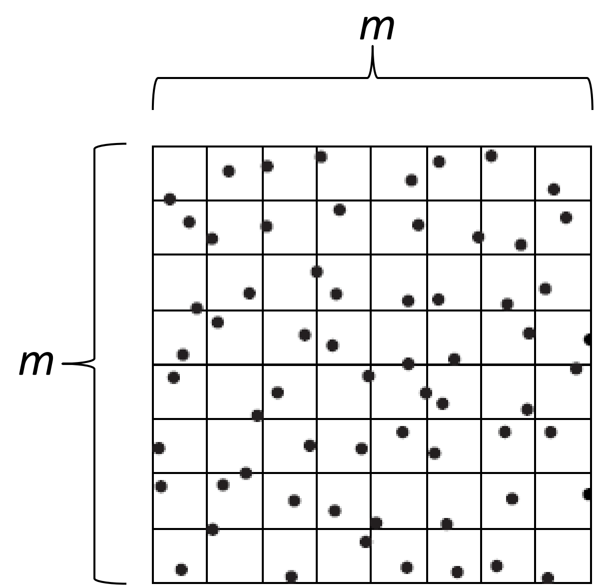
Stratified Sampling

Decompose domain into regular cells and choose a sample randomly from within each cell.



Stratified Sampling

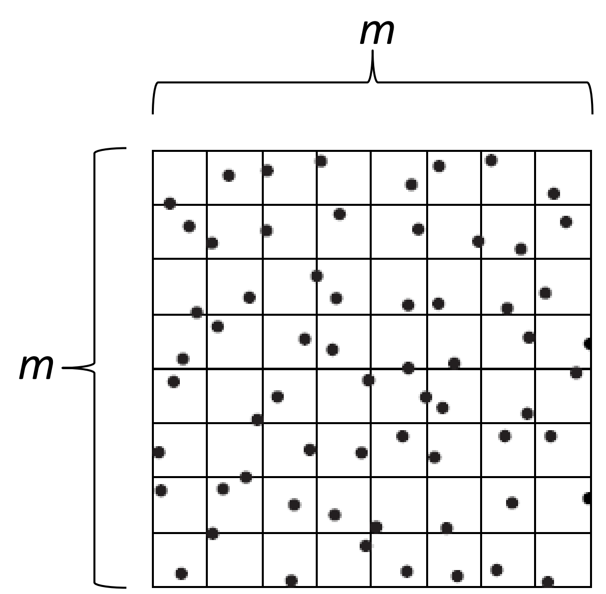
For a *d*-dimensional space, partitioning each dimension into *m* domains gives m^d samples.



Stratified Sampling

Limitation:

We would like *m* to be large to have stratified samples, but for large *d* we hit an impractically large number of samples.



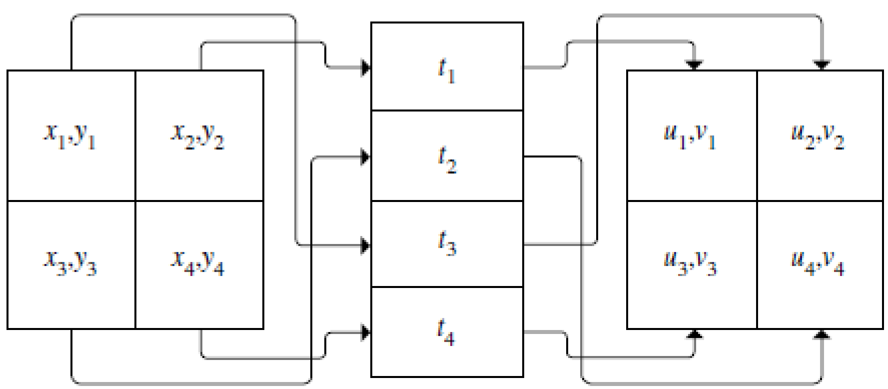
Stratified Sampling

Limitation:

We would like *m* to be large to have stratified samples, but for large *d* we hit an impractically large number of samples.

Approach:

Separately generate stratified samples for each dimension and then combine.



Stratified Sampling

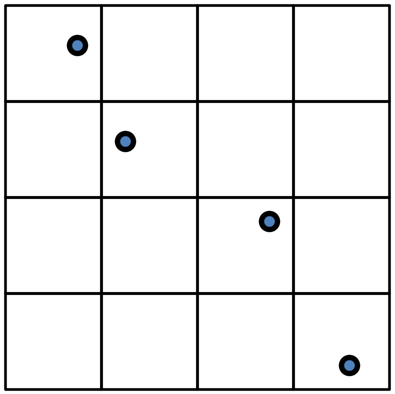
Limitation:

For dimensions $d>1$, the number of samples has to be factorizable into d (roughly equal) factors.

Latin Hypercube Sampling

To generate m (not m^2) samples in d dimensions:

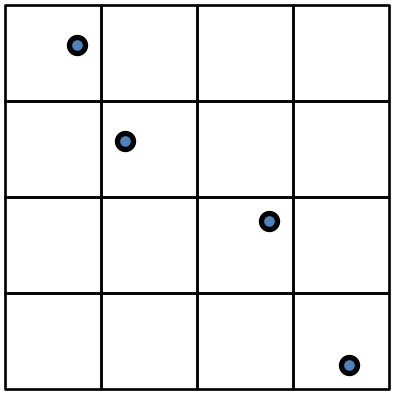
1. Partition each dimension into m parts and place a random sample in each diagonal cell.



Latin Hypercube Sampling

To generate m (not m^2) samples in d dimensions:

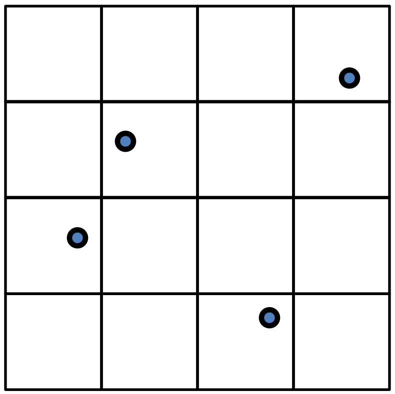
1. Partition each dimension into m parts and place a random sample in each diagonal cell. Each column/row has one sample.



Latin Hypercube Sampling

To generate m (not m^2) samples in d dimensions:

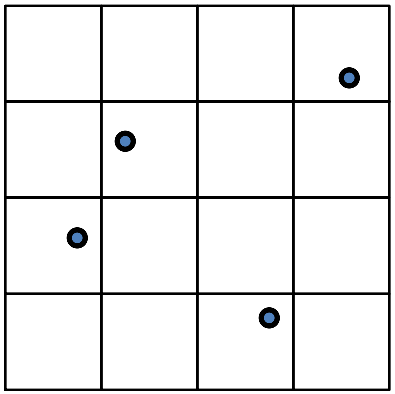
1. Partition each dimension into m parts and place a random sample in each diagonal cell. Each column/row has one sample.
2. Randomly permute along each dimension.



Latin Hypercube Sampling

To generate m (not m^2) samples in d dimensions:

1. Partition each dimension into m parts and place a random sample in each diagonal cell. Each column/row has one sample.
2. Randomly permute along each dimension. Each column/row still has one sample.



Other Sampling Methods

1. Low-discrepancy sampling
2. Best-candidate sampling
3. Adaptive Sampling

Other Sampling Methods

1. Low-discrepancy sampling
   – Find the sampling that minimizes the difference between the expect number of samples in a region and the actual number of samples.

Other Sampling Methods

1. Low-discrepancy sampling
   - Find the sampling that minimizes the difference between the expect number of samples in a region and the actual number of samples.

     For an integer written out in base *b* as:

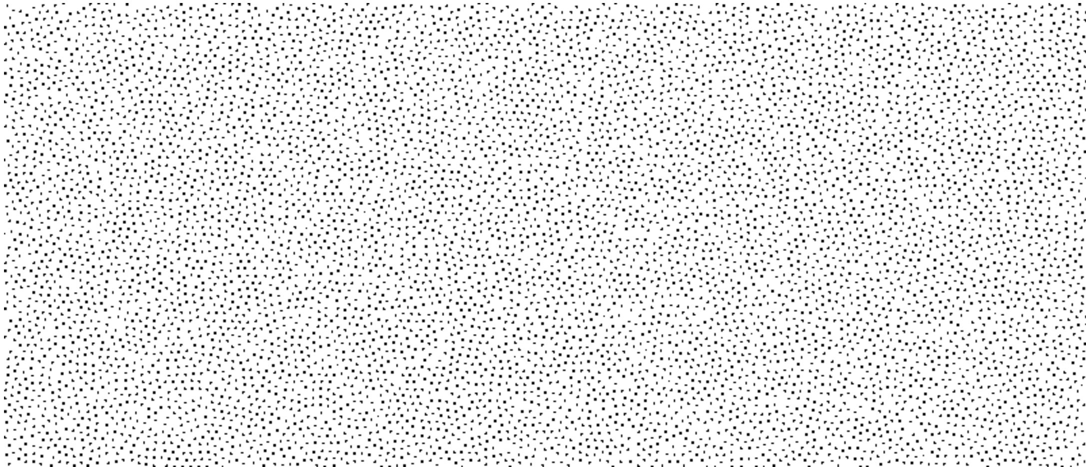
$$n = \sum_{k=0}^{\infty} a_k b^k \quad \text{with} \quad 0 \leq a_k < b$$

define the *radical inverse* to be the floating point value in the range[0,1) with:

$$\Phi_b(n) = \sum_{k=0}^{\infty} a_k b^{-1-k}$$

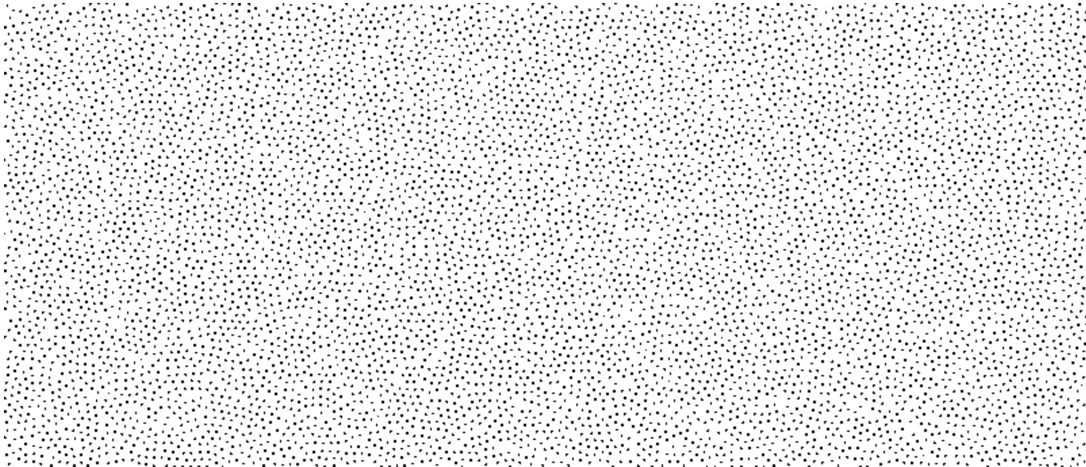
Other Sampling Methods

1. Low-discrepancy sampling
2. Best-candidate sampling
   – Generate samples that are guaranteed not to get too close.



Other Sampling Methods

1. Low-discrepancy sampling
2. Best-candidate sampling
   – Generate samples that are guaranteed not to get too close.
     Pre-compute and then tile.



Other Sampling Methods

1. Low-discrepancy sampling
2. Best-candidate sampling
3. Adaptive sampling
   – Evaluate the results from the samples and determine if more samples are required.

Other Sampling Methods

1. Low-discrepancy sampling
2. Best-candidate sampling
3. Adaptive sampling
   – Evaluate the results from the samples and determine if more samples are required.
     - Samples come from different geometry
     - Sample color variation is large.

Reconstruction

Note that in using random sampling, we are no longer sampling at the pixel-resolution.

Once we have the samples, we would like to reconstruct a function sampled at a fixed resolution (width x height).

Reconstruction

Note that in using random sampling, we are no longer sampling at the pixel-resolution so we have to:

1. Reconstruct a function from the samples.
2. Sample the function at the resolution of the image.

Reconstruction

Note that in using random sampling, we are no longer sampling at the pixel-resolution so we have to:

1. Reconstruct a function from the samples.
2. Sample the function at the resolution of the image.

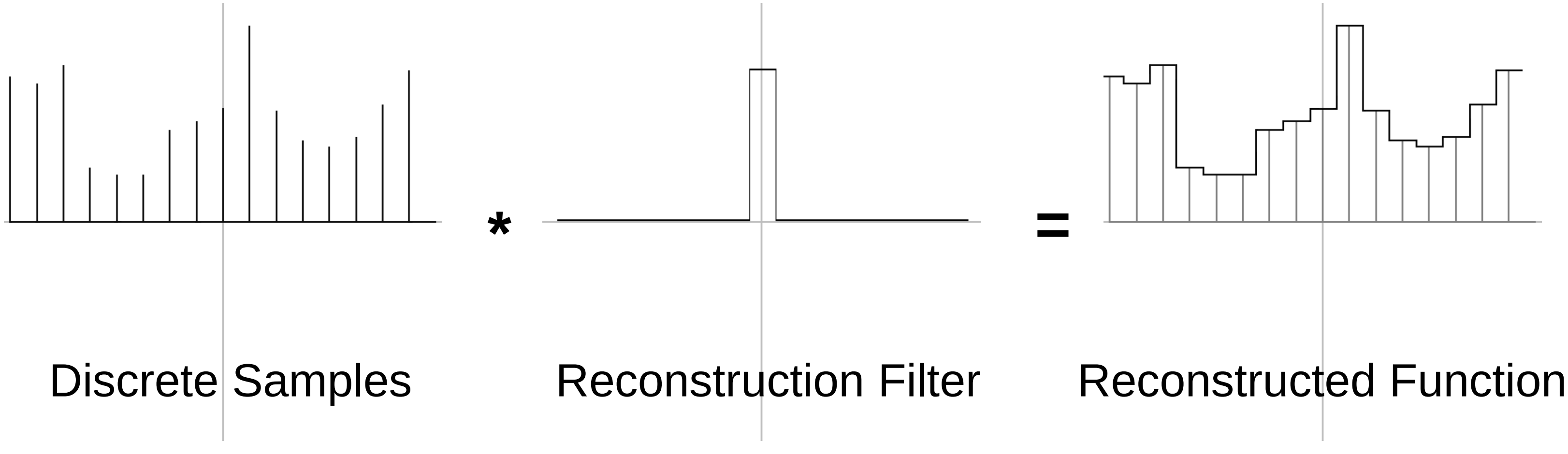
To avoid aliasing, we need to reconstruct with an appropriate (smoothed) filter.

Filter Options

Box-Filter:

✓ Can be implemented efficiently because the filter is non-zero in a very small region.

✘ Introduces high frequency content that will cause aliasing when sampled into an image.

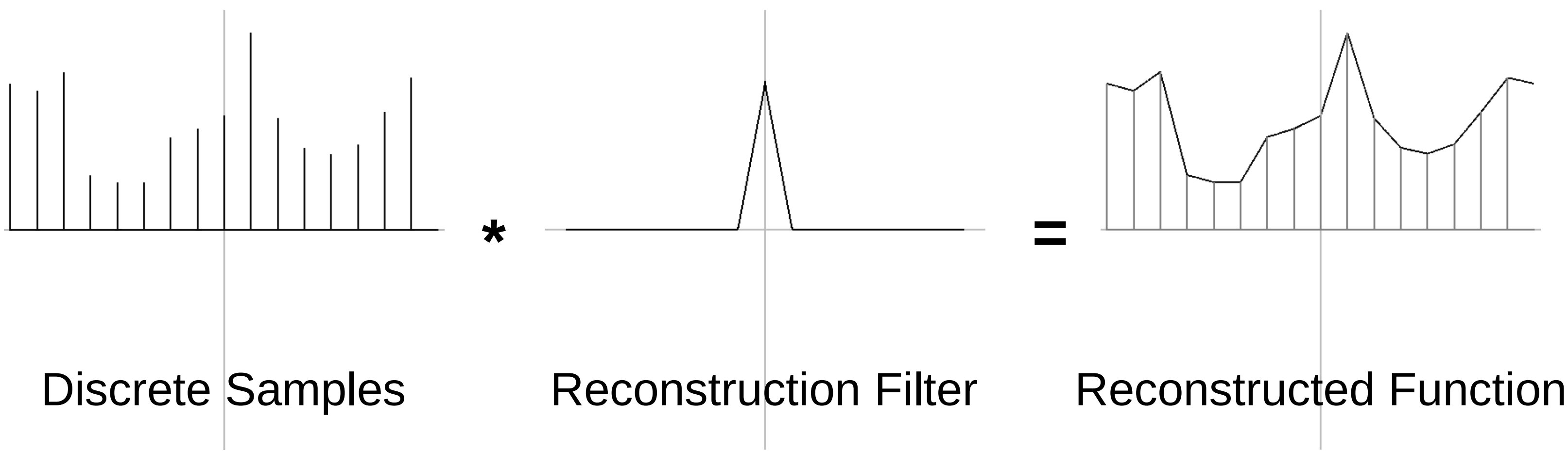


Filter Options

Hat-Filter:

✓ Can be implemented efficiently because the filter is non-zero in a very small region.

✘ Partially addresses the aliasing problem, but still introduces high frequency content that will cause aliasing when sampled into an image.

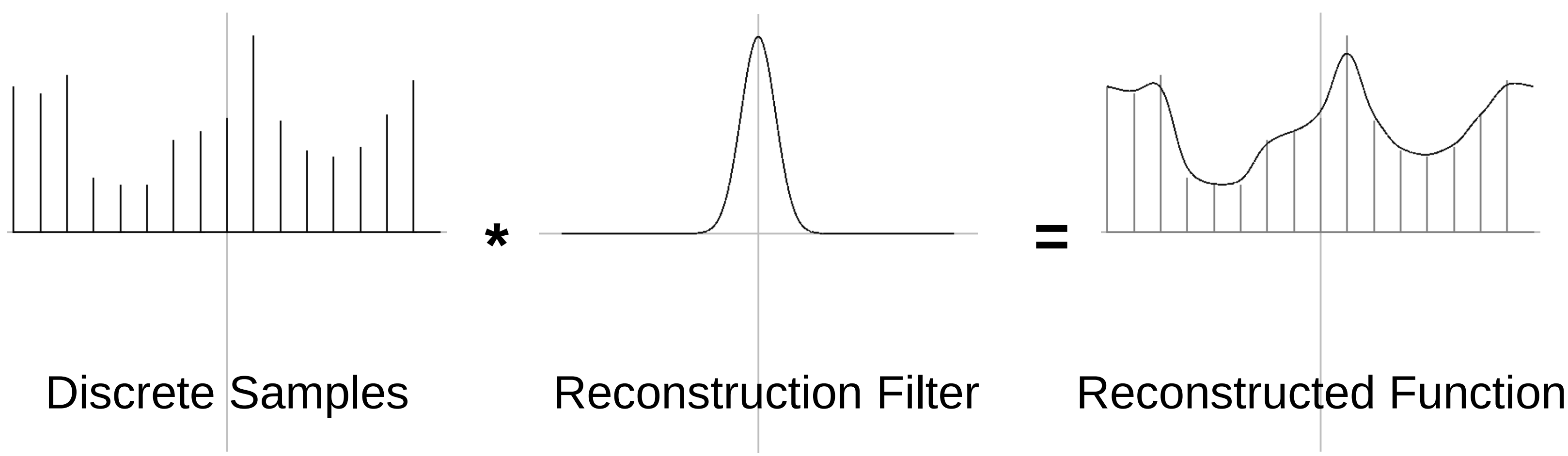


Filter Options

<u>(Clamped) Gaussian-Filter</u>:

✘ Is slow to implement because the filter is non-zero in a large region.

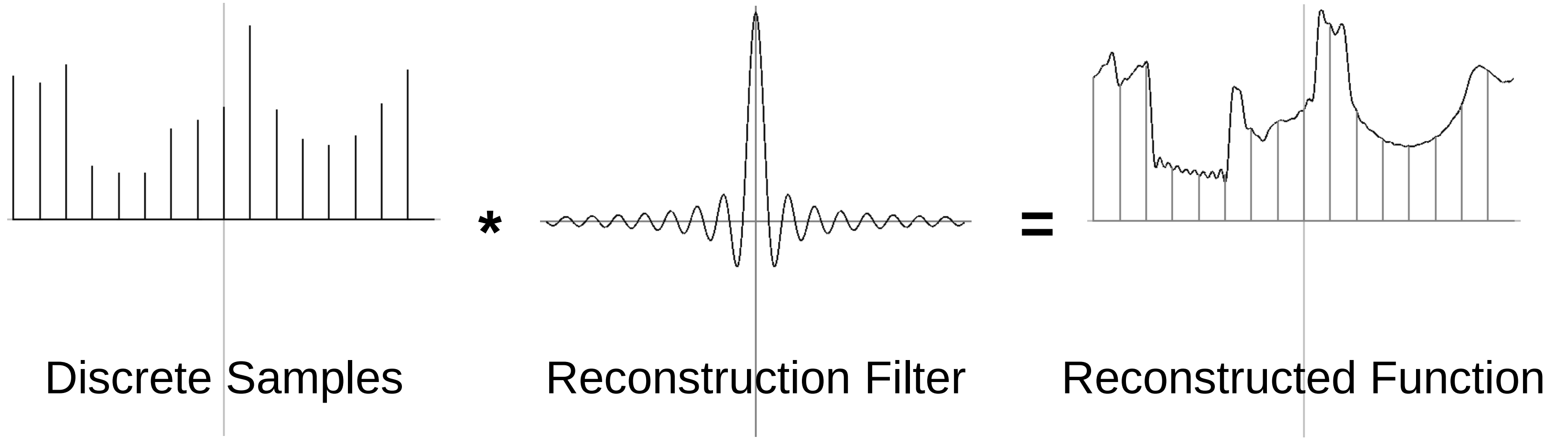
✓ Addresses the aliasing problem by killing off most of the high frequencies.



Filter Options

Sinc-Filter:

- ✘ Is slow to implement because the filter is non-zero in a large region.
- ✘ Assigns negative weights.
- ✘ Ringing at discontinuities.
- ✓ Addresses the aliasing problem by killing off the high frequencies.



Filter Options

Lanczos-Filter:

Modulates the first *k* lobes of the sinc filter with a cosine function.