
LECTURE 6 BASIC IMAGE RECONSTRUCTION

Lecture Notes by Assaf Tal

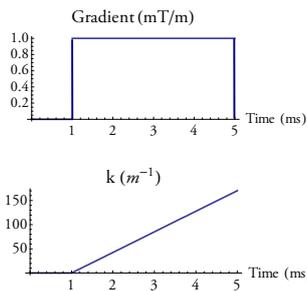
SAMPLING K-SPACE

We Can "Take A Walk" In k-Space By Varying The Gradient

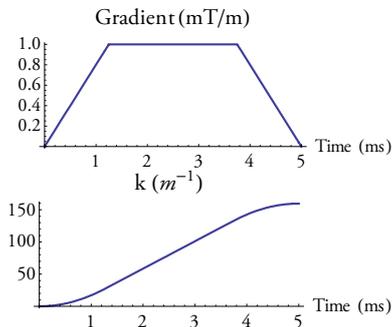
We manipulate $k(t)$ by changing the gradient. According to its definition,

$$k(t) = \gamma \int_0^t G(t') dt'$$

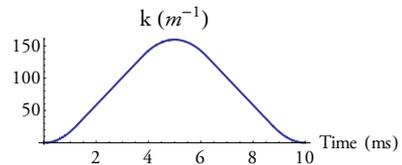
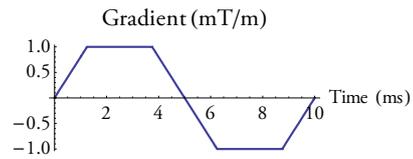
it is equal (per-component) to the total area underneath the gradient. In 1D, for example:



Of course, a realistic gradient would need non-zero ramp up and down times:



Since we can apply negative gradient values we can also "rewind" k back to the origin:



As another example, consider the following 2D gradient waveform function:

$$G_x(t) = G_0 \cdot \frac{t}{T} \cdot \cos(\omega_g t)$$

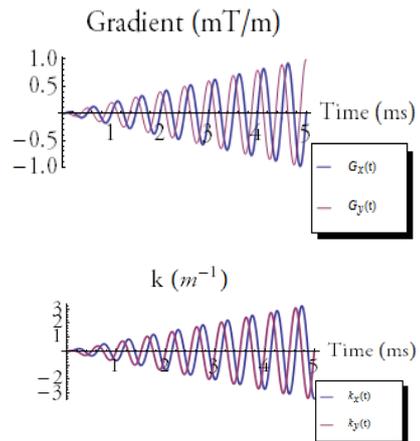
$$G_y(t) = G_0 \cdot \frac{t}{T} \cdot \cos(\omega_g t)$$

with $T = 5$ ms, $G_0 = 1$ mT/m and $\omega_g = 2\pi \cdot 2$ Hz. We can find $k(t)$ easily by integrating:

$$k_x = \gamma \int_0^t G_x(t') dt'$$

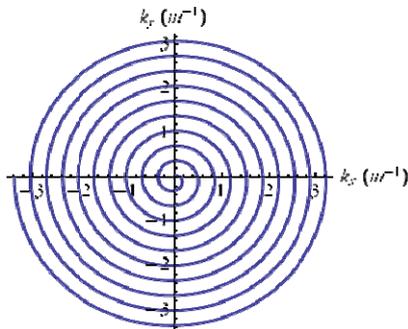
$$k_y = \gamma \int_0^t G_y(t') dt'$$

We will not actually carry out the integration analytically as it is not particularly interesting. Instead, I've used a plotting software package to plot both G_x , G_y and k_x , k_y as a function of time:



We can think of $k(t)$ as tracing a "path" in the k_x - k_y plane, by plotting it parametrically (that is,

plotting the position of the vector $\mathbf{k}(t)$ as a function of the variable t :

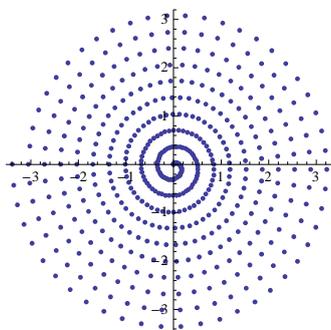


In effect, we are sampling data along this trajectory in k-space (that is, we are measuring $s(\mathbf{k})$ along the trajectory).

Our Sampling Of k-Space Is Limited By Both The Total (1) Acquisition Time And (2) Sampling Rate

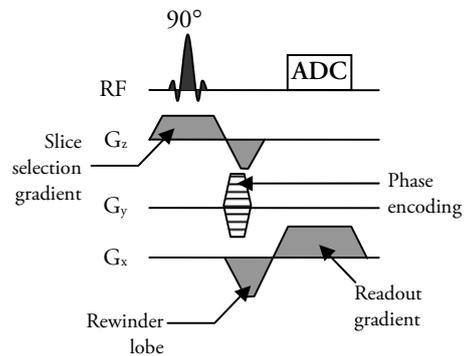
In theory, $s(\mathbf{k})$ and $f(\mathbf{r})$ are related via a CFT. However, we cannot measure $s(\mathbf{k})$ at every point in k-space. It should be quite clear to the reader at this point that traversing k-space by varying the gradients takes time, and that we do not have infinite acquisition time, seeing as the magnetization decays in the xy-plane with a time constant T_2 . Hence, the extent to which we can cover k-space is limited.

There is another factor limiting our coverage of k-space, which is the sampling rate of the analog to digital converter (ADC). The ADC samples at a constant rate in time steps Δt called the **dwell time**. Thus we do not measure data along a continuous \mathbf{k} variable but only at discrete points, i.e. a discrete trajectory:

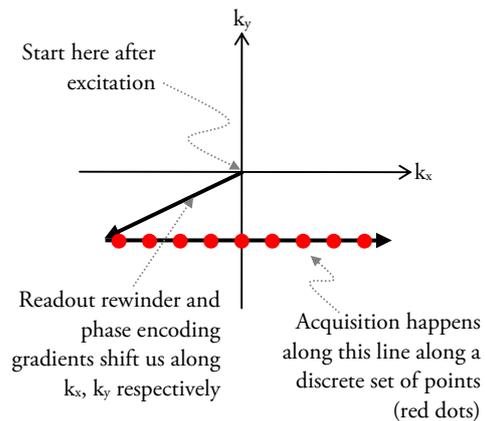


Phase And Frequency Encoding Are Often Combined To Read Out A Cartesian Data Set In k-Space

We have previously outlined a 2D pulse sequence which combines phase and frequency encoding, but have not delved into its workings. We will take a closer look at it now, with a small modification: we will insert a **rewinder gradient** before the readout gradient, for a reason that will become clear in a moment:



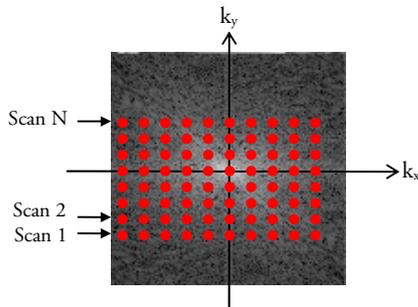
The above sequence is repeated multiple times, each time incrementing the phase encoding gradient from some initial negative value to some final positive value. A particular scan and its corresponding dataset in k-space would look like this:



Thus the purpose of the **rewinder gradient** is to shift us back along the readout axis (k_x) so we read out data symmetrically in k-space.

Repeating the sequence multiple times, varying the phase encoding gradient each time, yields a 2D data set of values of $s(\mathbf{k})$ sampled on a cartesian

grid of k points. Each line in k -space will result from an excitation-rewinding-acquisition block:



In the next section we will examine the problem of reconstructing the image $f(\mathbf{r})$ from this discrete cartesian sampled dataset of $s(\mathbf{k})$.

THE DISCRETE FOURIER TRANSFORM

Discrete Sampling Leads To Aliasing and Blurring

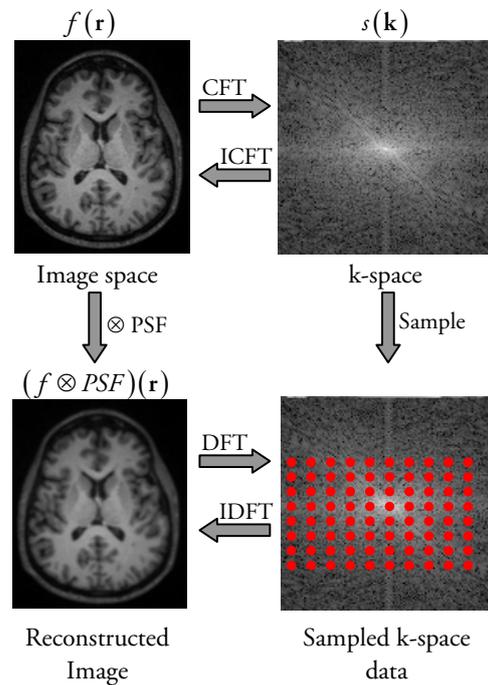
In the remainder of this lecture we will concern ourselves with two questions:

1. How do we reconstruct our image from the discretely sampled data?
2. What effect will this have on the image? That is, how will our reconstructed image compare to the “true” image $f(\mathbf{r})$?

The answers to these two questions will be:

1. Reconstruction takes place via a **Discrete Fourier Transform** (DFT).
2. The sampling will yield an image that is obtained by convolving the true image with a point spread function and sampling it on a discrete grid. The convolution will lead to two effect: the first will be **blurring** of the original image, and the second will be **aliasing** (that is, the appearance of “copies” of the original image which may or may not overlap with the original image, depending on how we choose our imaging parameters).

Graphically:



Statement Of The Problem

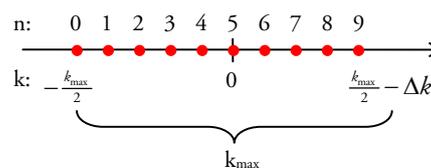
Let us confine ourselves to a one dimensional problem. Given $f(x)$, we form its continuous Fourier transform:

$$\begin{cases} s(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx & \text{(CFT)} \\ f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} s(k) e^{2\pi i k x} dk & \text{(ICFT)}. \end{cases}$$

We sample $s(k)$ on a discrete grid of N points, starting from some initial value $-k_{\max}/2$ and advancing in steps $\Delta k = k_{\max}/N$:

$$k_n = -\frac{k_{\max}}{2} + n \cdot \Delta k \quad (n = 0, 1, \dots, N-1)$$

The N points k_n in this particular sampling pattern are not symmetrically placed around 0: there is a last point missing at $k_{\max}/2$. However, if you think in terms of *intervals*, it is symmetrically placed. Here is an example with $N=10$:



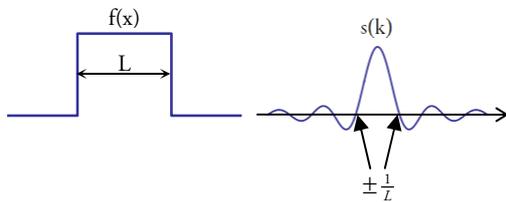
We now pose the question: given the set $s(k_n)$, how do we recover $f(x)$? It would be instructive to take a particular example. Suppose our “image” $f(x)$ is an uninteresting boxcar:

$$f(x) = \begin{cases} 1 & |x| \leq \frac{L}{2} \\ 0 & \text{elsewhere} \end{cases}$$

It is actually feasible to compute its CFT analytically:

$$\begin{aligned} s(k) &= \int_{-L/2}^{L/2} e^{-2\pi i k x} dx \\ &= \frac{e^{-2\pi i k x}}{-2\pi i k} \Big|_{x=-L/2}^{x=L/2} \\ &= \frac{1}{\pi k} \frac{e^{2\pi i k L/2} - e^{-2\pi i k L/2}}{2i} \\ &= \frac{\sin(\pi k L)}{\pi k} = L \cdot \text{sinc}(\pi k L) \end{aligned}$$

where we have used $\text{sinc}(x) = \frac{\sin(x)}{x}$ and $\sin(x) = \frac{e^{ix} - e^{-ix}}{2i}$. The two functions are plotted next to each other below:



This highlights a general feature of Fourier transforms, namely that the width of the function in one domain is approximately equal to the inverse of its transform in the Fourier domain.

From The Continuous To The Discrete Fourier Transform (DFT)

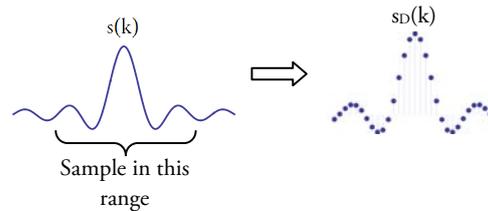
We would like to be able to compute $f(x)$ by inverting the continuous Fourier transform:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} s(k) e^{2\pi i k x} dk.$$

However, we only know $s(k)$ on our discretely sampled grid. Intuitively, let us form the function:

$$s_D(k) = \sum_{n=0}^{N-1} s(k_n) \delta(k - k_n).$$

The function $\delta(k)$ is known as a Dirac delta function, and you can think of it as a very sharply peaked function around $k=0$ with unit area. $s_D(k)$ is the sampled version of $s(k)$:



We hope that, if we sample “enough” points, we should be able to approximate the continuous Fourier integral by a discrete sum. This motivates us to form the discrete Fourier transform and its inverse:

Given a discrete set of (possibly complex) points s_0, s_1, \dots, s_{N-1} , we define their discrete Fourier transform (DFT) as the set of points $\hat{s}_0, \hat{s}_1, \dots, \hat{s}_{N-1}$:

$$\hat{s}_m = \sum_{n=0}^{N-1} s_n e^{\frac{2\pi i n m}{N}} \quad (\text{DFT})$$

Then it is possible to prove that, given $\hat{s}_0, \hat{s}_1, \dots, \hat{s}_{N-1}$, the numbers s_0, s_1, \dots, s_{N-1} can be recovered by computing the inverse DFT:

$$s_n = \frac{1}{N} \sum_{m=0}^{N-1} \hat{s}_m e^{-\frac{2\pi i n m}{N}} \quad (\text{IDFT})$$

Compare this to the CFT and its inverse:

$$\begin{cases} s(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx & (\text{CFT}) \\ f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} s(k) e^{2\pi i k x} dk & (\text{ICFT}). \end{cases}$$

Graphically, we have:

The Point Spread Function Of Cartesian Sampling Is The “Dirichlet Kernel”

We now calculate the IDFT of the sampled data points we’ve acquired in k-space:

$$\begin{aligned}\hat{s}_m &\equiv \hat{s}(k_m) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k_m x} dx \\ &= e^{\pi i k_{\max} x} \int_{-\infty}^{\infty} f(x) e^{-2\pi i m \Delta k x} dx, \quad (m = 0, 1, \dots, N-1)\end{aligned}$$

Now how we call our points \hat{s}_n instead of s_n to emphasize that we’re going to use an IDFT. We get, upon substitution and interchange of the summation and integration signs:

$$\begin{aligned}s_n &= \frac{1}{N} \sum_{m=0}^{N-1} \hat{s}_m e^{\frac{2\pi i n m}{N}} \\ &= \frac{e^{\pi i k_{\max} x}}{N} \int_{-\infty}^{\infty} f(x) \left(\sum_{m=0}^{N-1} e^{\frac{2\pi i n m}{N}} e^{-2\pi i m \Delta k x} \right) dx\end{aligned}$$

Since $e^a e^b = e^{a+b}$, we simplify as follows:

$$\begin{aligned}s_n &= \frac{1}{N} \sum_{m=0}^{N-1} \hat{s}_m e^{\frac{2\pi i n m}{N}} \\ &= \frac{e^{\pi i n}}{N} \int_{-\infty}^{\infty} f(x) \left(e^{\pi i k_{\max} \left[x - \frac{n}{N \Delta k} \right]} \sum_{m=0}^{N-1} e^{-2\pi i m \Delta k \left[x - \frac{n}{N \Delta k} \right]} \right) dx\end{aligned}$$

The function in parenthesis is just a function of x , shifted by an amount $n/\Delta k$ (depending on n). Let us **define**:

$$\begin{aligned}PSF(x) &= \frac{1}{N} e^{\pi i k_{\max} x} \sum_{m=0}^{N-1} e^{-2\pi i m \Delta k x} \\ &= \frac{1}{N} e^{\pi i k_{\max} x} \sum_{m=0}^{N-1} \left(e^{-2\pi i \Delta k x} \right)^m\end{aligned}$$

This will turn out to be the point spread function, since, using this definition:

$$s_n = e^{\pi i n} \int_{-\infty}^{\infty} f(x) PSF\left(x - \frac{n}{N \Delta k}\right) dx.$$

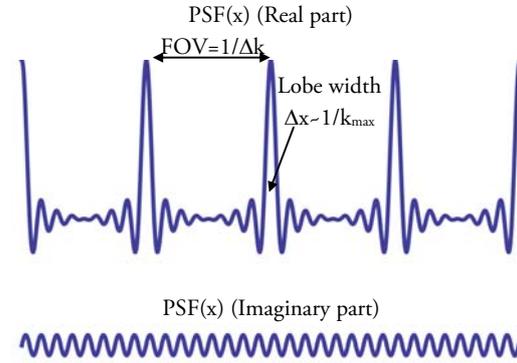
Calculating the summation in $PSF(x)$ is straightforward since it is nothing more than a geometrical series:

$$\sum_{m=0}^{N-1} a^m = \frac{1-a^N}{1-a}, \quad a \equiv e^{-2\pi i \Delta k x}.$$

Substituting and simplifying, we get:

$$PSF(x) = \frac{e^{\pi i x \Delta k} \sin(\pi \cdot k_{\max} \cdot x)}{\sin(\pi \cdot \Delta k \cdot x)} \quad (k_{\max} = N \cdot \Delta k)$$

This is known as the Dirichlet kernel. We plot it and note some crucial features:



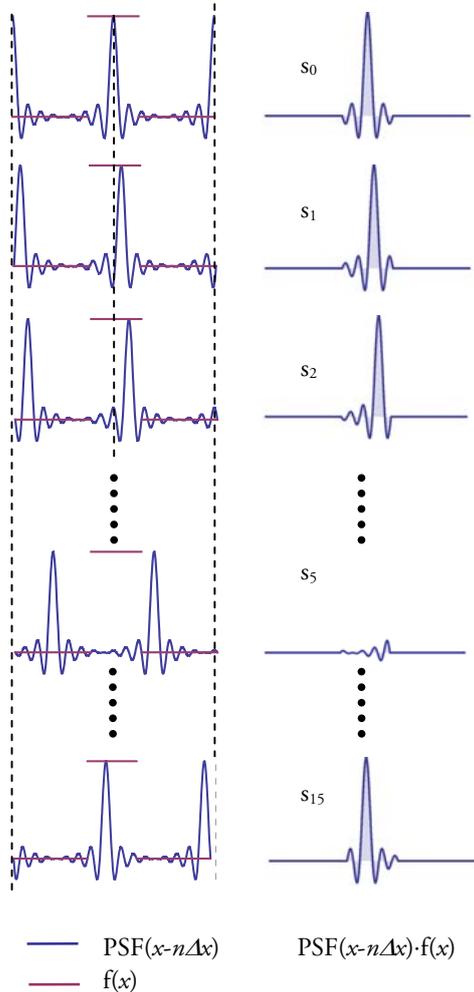
It is periodic with a period $1/\Delta k$, and its lobes have a width approximately given by $1/k_{\max} = 1/(N \Delta k)$, i.e. N times smaller than its periodicity. We define two quantities known as the Field Of View (FOV) and spatial resolution (Δx):

$$\begin{aligned}FOV &= \frac{1}{\Delta k} \\ \Delta x &= \frac{1}{k_{\max}} = \frac{1}{N \cdot \Delta k} = \frac{FOV}{N}\end{aligned}$$

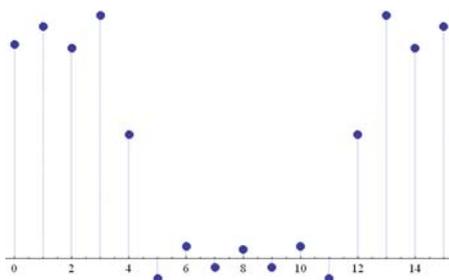
The Point Spread Function “Scans” The Image From $-FOV/2$ To $FOV/2$, In Steps Of Δx

To visualize the imaging process, I’ve plotted the boxcar function $f(x)$ with a width of unity, and have chosen the following sampling parameters: $FOV = 2$, $N = 16$ so $\Delta x = 0.125$ and $\Delta k = 0.5$. Since $N = 16$, we have 16 k-space samples $\hat{s}_0, \hat{s}_1, \dots, \hat{s}_{15}$ from which we construct 16 image coefficients s_0, s_1, \dots, s_{15} . The j^{th} coefficient is obtained by

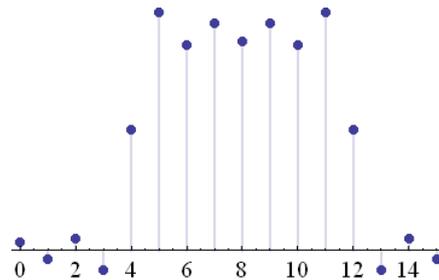
multiplying the true distribution of spins, $f(x)$, by $\text{PSF}(x-n\Delta x)$ and integrating (i.e. calculating the area). This is shown in the following diagram:



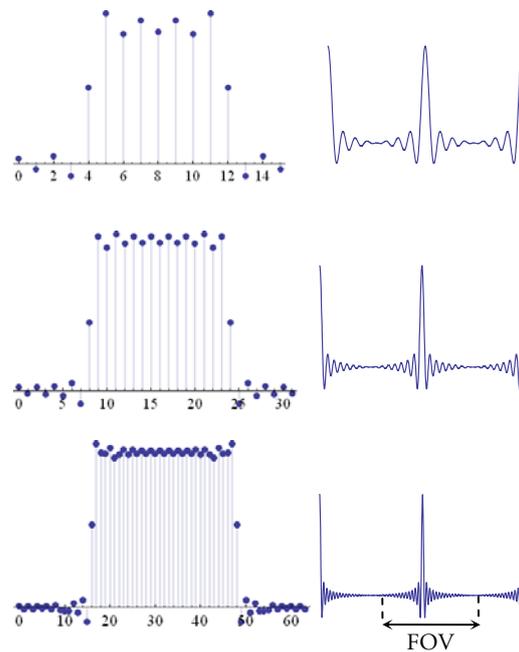
We see that the PSF “moves” in small steps of Δx . When the main lobe moves outside the spin distribution the area becomes negligible, and when it moves back in the signal grows back up. If we actually calculate s_n and plot the result, we obtain:



This looks like a “mirrored” version of $f(x)$ about $x=0$. This mirroring is the result of the PSF starting from the center of the image at $n=0$ and not from the far left edge. This implies that we need to “switch” the right and left parts of the image, which yields:



This is not a perfect image of the boxcar function but it’s not too bad. Here are the coefficients s_k when we increase N to 32, and then 64 (shown on the left) and the corresponding changes to the PSF on the right:

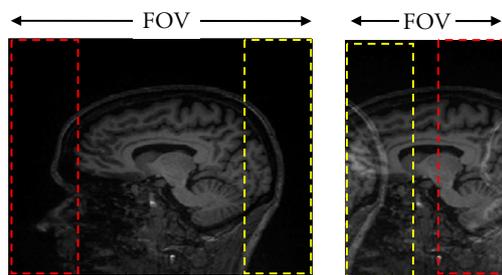


Aliasing Is Caused When The Steps In k -Space Are Not Small Enough

The Dirichlet kernel has a periodic structure with periodicity given by the Field of View:

$$PSF(x + FOV) = PSF\left(x + \frac{1}{\Delta k}\right) = PSF(x).$$

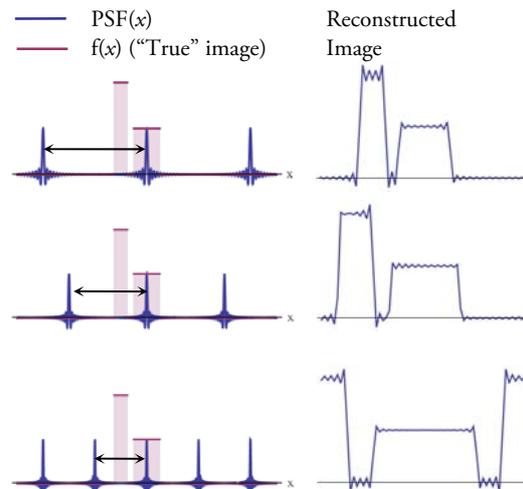
If the FOV is smaller than the imaged object then we will get **aliasing**. This describes a phenomena in which the distance between adjacent lobes, given by the FOV, becomes smaller than the object. Thus, as the PSF “scans” the object, non-central lobes will re-enter the image, effectively causing multiple copies of the object to appear in the reconstructed image. This has a very classic appearance in MRI, in which one side of the image “wraps” into the opposite side:



Aliasing is removed by keeping the FOV larger than the object:

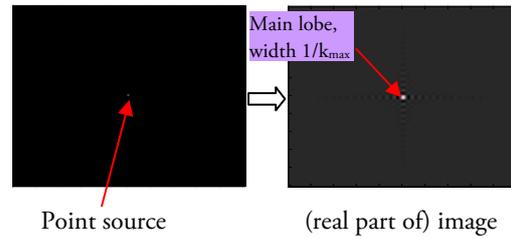
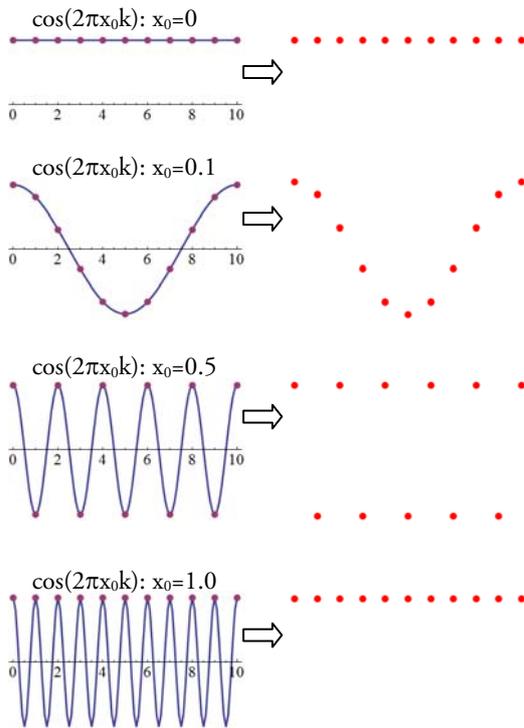
To avoid aliasing, keep $FOV = 1/\Delta k$ bigger than the object's dimensions.

This is illustrated in the following diagram, showing an object ($f(x)$), two boxcars of size 3 arbitrary units, and a point spread function with $FOV = 4.0, 3.0$ and 2.0 . In the last case aliasing will occur ($N=64$).

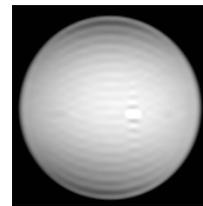


Another Way To Understanding Aliasing

Aliasing can also be understood from a different perspective: we are sampling the image in k-space, which is its Fourier space. This space represents the frequencies of the object. Once we only consider discrete frequencies in jumps of Δk , we lose the ability to tell apart positions x and $x+1/\Delta k$, as shown in the following diagram, in which $\Delta k=1$ (in arbitrary units) and $\cos(2\pi x_0 k)$ and its sampled version are both plotted for $x_0=0, 0.1, 0.5$ and 1.0 , showing that, for $x_0=0$ and 1 , the set of sampled points coincides and the two positions cannot be discerned:



The wiggly edges of the PSF are responsible for an artifact often referred to as Gibb's ringing, observed as "wiggles" upon transitioning from one intensity sharply to another:



Since the wiggles become smaller as $N \rightarrow \infty$ while keeping the FOV fixed, one way to reduce it (but not completely eliminate it) is to simply take more points.

Back to resolution: the rule of thumb is

To reduce blurring, keep $\Delta x = 1/k_{\max}$ as small as possible.

Of course, how far we can go out in k-space will depend on how long we have to sample and how strong our gradients are.

Another Way To Understand Blurring

Blurring can also be understood as follows: if we could sample all of k-space we could reconstruct our source image perfectly with an inverse continuous Fourier transform:

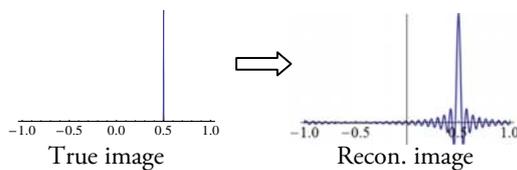
$$f(x) = ICFT[s(k)]$$

Putting aside for a moment the question of the discreteness of sampling, we look at the effect of the finiteness of our sampling extend. We're basically sampling $s(k)$ only in some interval $[-\frac{k_{\max}}{2}, \frac{k_{\max}}{2}]$. This is the same as fully sampling $s(k)$ times a windowing function $W(k)$, which is equal to 1 in $k \in [-\frac{k_{\max}}{2}, \frac{k_{\max}}{2}]$ and 0 outside the

Blurring Occurs When We Don't Go Far Enough In k-Space

The width of the Dirichlet kernel's main lobe sets the *resolution* of the image. Any point image $\delta(x - x_0)$ will be replaced by the PSF, centered at x_0 :

$$\delta(x - x_0) \rightarrow \int \delta(x' - x_0) PSF(x - x') dx' = PSF(x - x_0)$$



This means features will get "broadened" by the PSF's lobe's main width, which is approximately $\frac{1}{k_{\max}}$. This determines our real spatial resolution. The same effect can be seen in 2D and 3D cartesian sampling schemes as well. For example, in 2D ($FOV_x = FOV_y = 256$ mm, $N_x = N_y = 64$):

interval. What would we get if we were to apply an ICFT to that?

$$ICFT[s(k) \cdot W(k)] = ?$$

Fortunately, a well known theorem from Fourier theory¹ called the Convolution Theorem states that the Fourier of the product equals the convolution of the Fourier. That is:

$$\begin{aligned} ICFT[s(k) \cdot W(k)] \\ = ICFT[s(k)] \otimes ICFT[W(k)] \end{aligned}$$

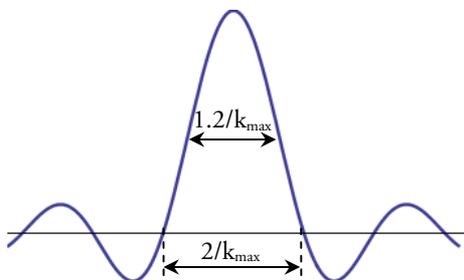
We already know that $f(x) = ICFT[s(k)]$, and we've also shown that the Fourier transform of a boxcar function $W(k)$ of width k_{max} is a sinc of width $\sim 1/k_{max}$. Thus,

$$\begin{aligned} ICFT[s(k) \cdot W(k)] \\ = (\text{image}) \otimes \left(\text{sinc of width } \frac{1}{k_{max}} \right) \end{aligned}$$

This in essence restates our previous result: features will get "broadened" by the sinc function's main width, which is approximately $\frac{1}{k_{max}}$.

Why The Nominal Resolution Tells Only "Half The Story"

Suppose we image in one dimension with the following parameters: FOV = 10 mm and N=10, meaning our **nominal resolution** is $\Delta x = \text{FOV}/N = 1$ mm. However, the PSF is not a perfect boxcar with a width of 1 mm. Rather, there is no unique way to define the "resolution" of the PSF, only to say it is approximately $1/k_{max}$. Some parameters of the main lobe are shown below:



¹ It's not difficult to prove. You should try it.

So the width at the base is actually **twice** the nominal resolution (Δx), and even the width at half the maximal amplitude is approximately $1.2 \cdot \Delta x$. It is best to keep in mind these two figures of merit and not the nominal resolution itself, although almost all published papers do not mention the PSF at all and quote solely the nominal resolution. Still, it is quite easy to visualize the true form of the PSF given the nominal resolution, at least for cartesian sampling.