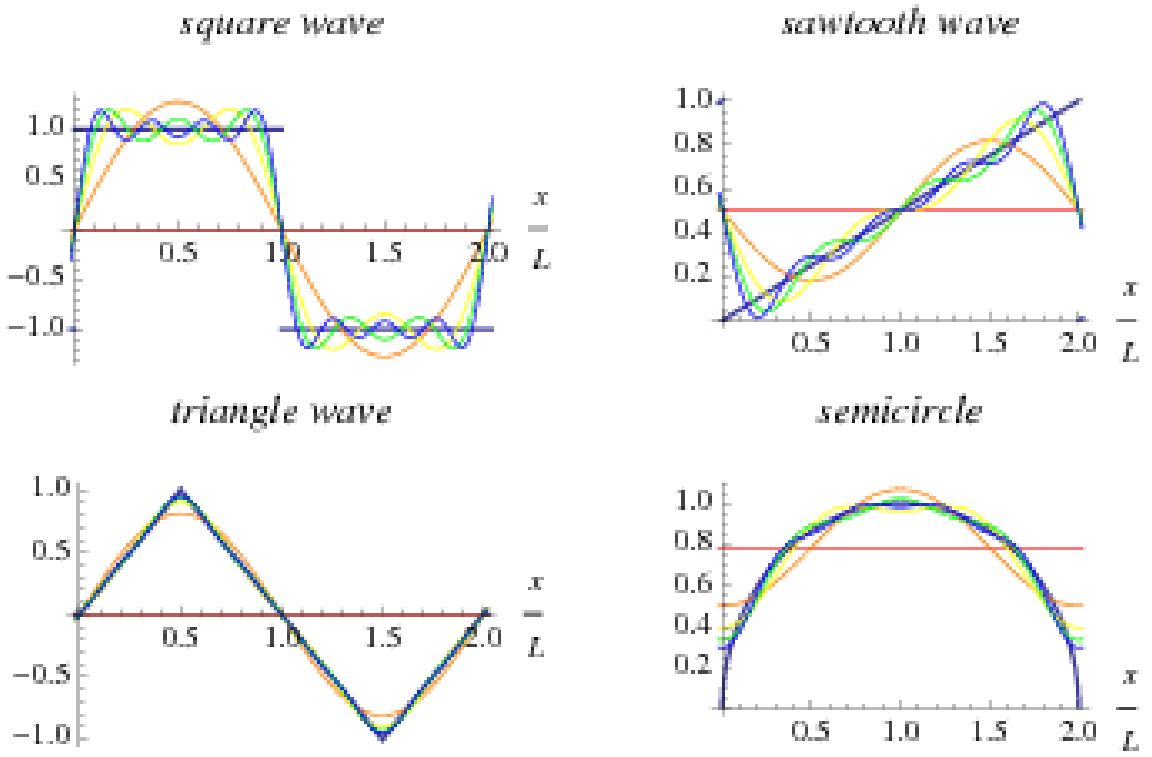
The Fourier transform

Fourier series



A Fourier series is an expansion of a periodic function in terms of an infinite sum of sines and cosines. Fourier series make use of the orthogonality of the sine and cosine functions.

Mathematically we can express the Fourier series as:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos(n\,x) + \sum_{n=1}^{\infty} b_n \sin(n\,x)$$

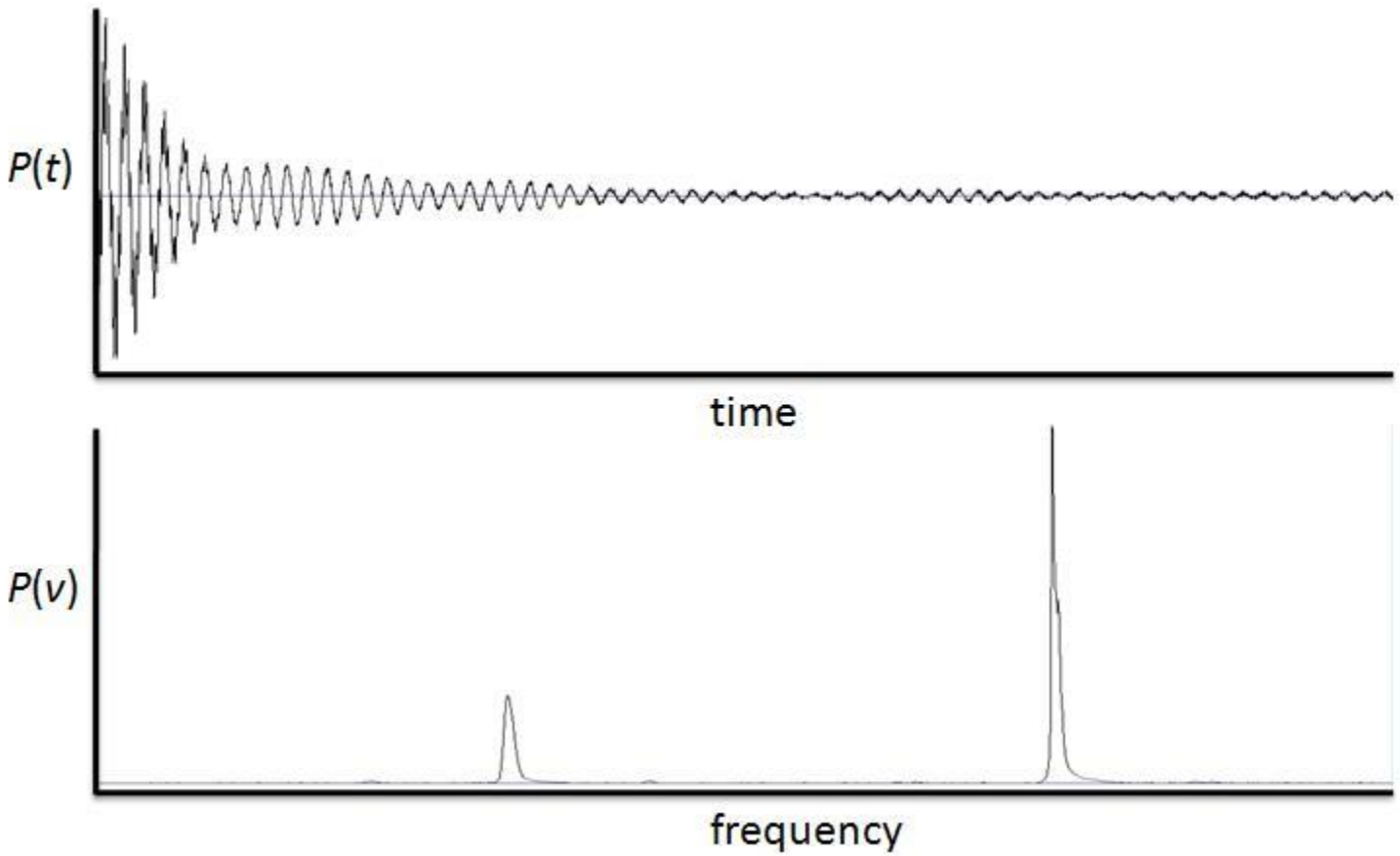
$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(n\,x)dx$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(n\,x)dx$$

Nuclear magnetic resonance
FT-NMR

Classic example: Free induction decay in NMR



Lorentzian broadening

A Lorentzian is the Fourier-transform of an exponential function. To see this we begin with the definition of a Fourier transform.

$$L(\omega) = \frac{1}{\pi}\int_0^\infty f(t)e^{i\omega t - \Gamma t}dt$$

We then substitute in $e^{-\Gamma t}$ where $\Gamma = 1/T_2$ is the relaxation time of the excited state. The precise model for the excited state dynamics depends on the system. If we consider magnetic resonance, we will include magnetic inhomogeneity in the linewidth and so T_2 is used. However, for optical transitions the homogeneous line width may include only the excited state life time, or natural life time, T_1.

The Fourier transform

This expression includes the appropriate normalization constant for a Lorentzian. In the general case of a Lorentzian centered about ω_o, we have

$$\frac{1}{\pi}\int_0^\infty e^{i(\omega-\omega_o)t-\Gamma t}dt = \frac{1}{\pi}\frac{1}{i(\omega-\omega_o)-\Gamma}$$

$$= \frac{1}{\pi}\frac{1}{i(\omega-\omega_o)-\Gamma}\left(\frac{-i(\omega-\omega_o)-\Gamma}{-i(\omega-\omega_o)-\Gamma}\right)$$

Absorption and dispersion

Out-of-phase
Absorption

In-phase
Dispersion

$$= \frac{1}{\pi}\frac{\Gamma}{(\omega-\omega_o)^2+\Gamma^2} - \frac{i}{\pi}\frac{(\omega-\omega_o)}{(\omega-\omega_o)^2+\Gamma^2}$$

The resulting function is complex. One can think of a complex function as resulting from in-phase and out-of-phase terms. An in-phase term leads to dispersion of the light and and out-of-phase term leads to absorption.

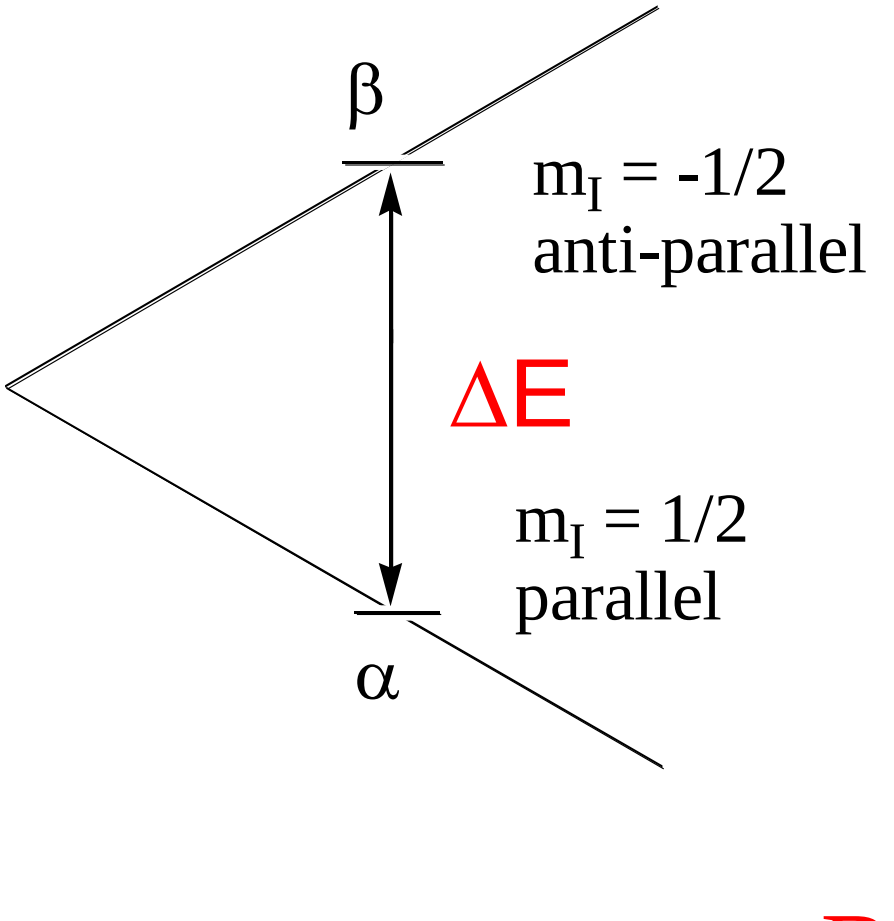
The NMR line shape is Lorentzian

Note that this is a normalized function so that the integral of $\Lambda(\omega)$ from $-\infty$ to ∞ is equal to one. Notice that the analogy with NMR is evident in the fact that our excited state decay function is a sinusoid times an exponential. The real part has the appearance

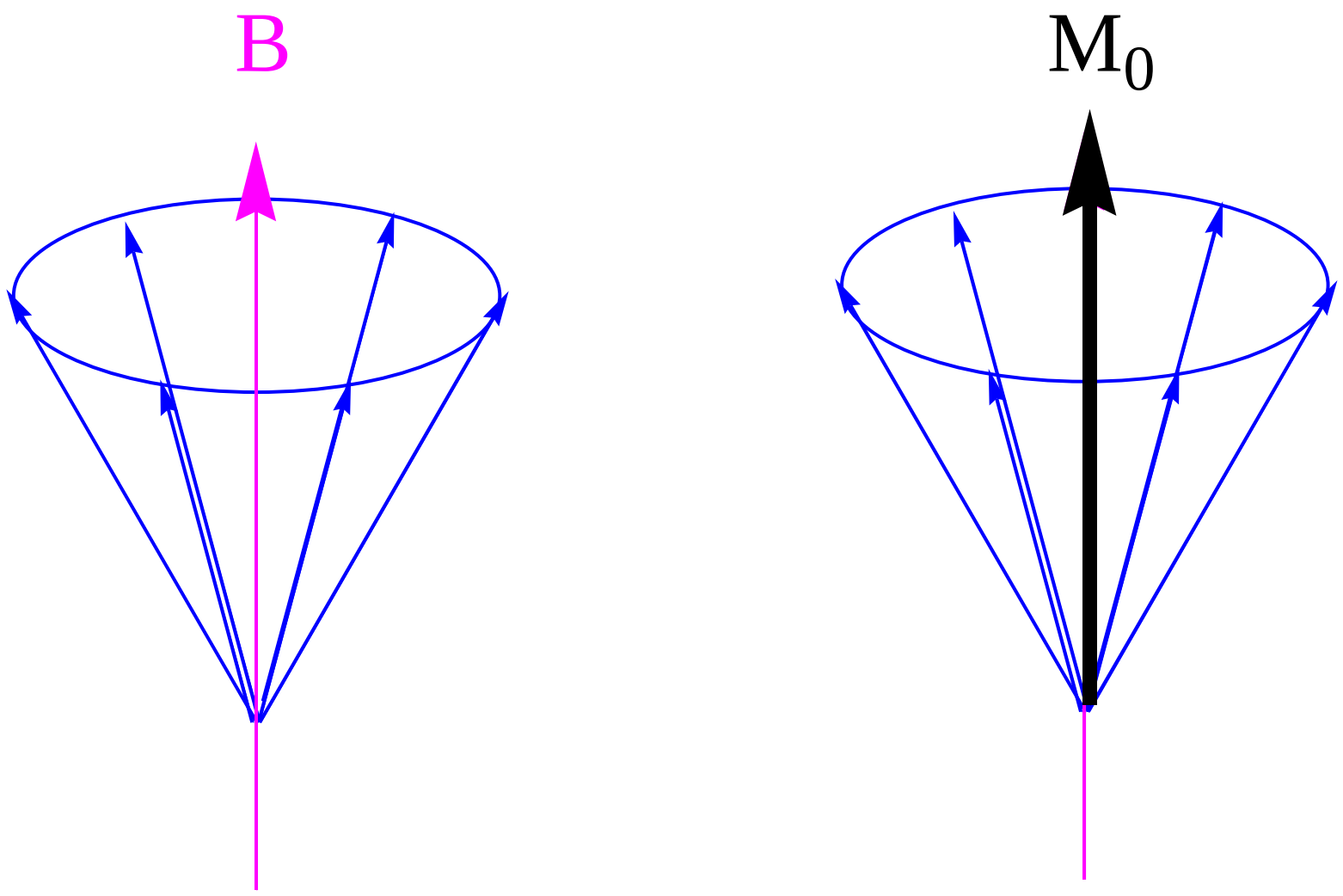
$$\Lambda(\omega) = \frac{\Gamma}{\pi\left(\Gamma^2 + \left(\omega - \omega_0\right)^2\right)}$$

This is what is observed in NMR. Note that we have added a factor of π for normalization. We can say that the Fourier transform of an exponential is a Lorentzian.

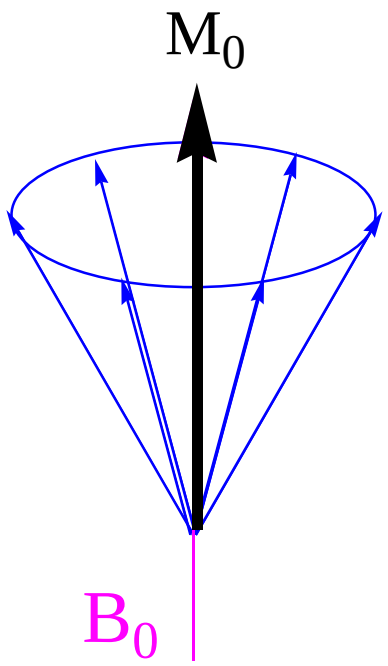
NMR in spin ½ system



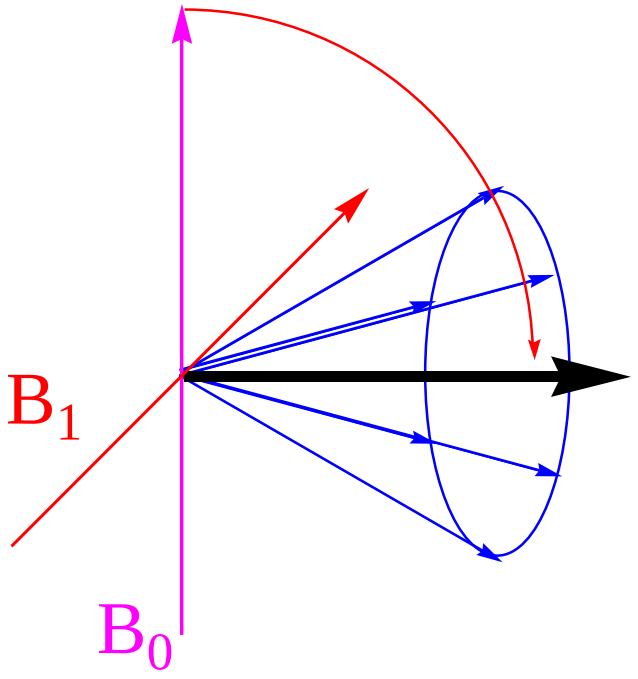
Definition of the magnetization vector



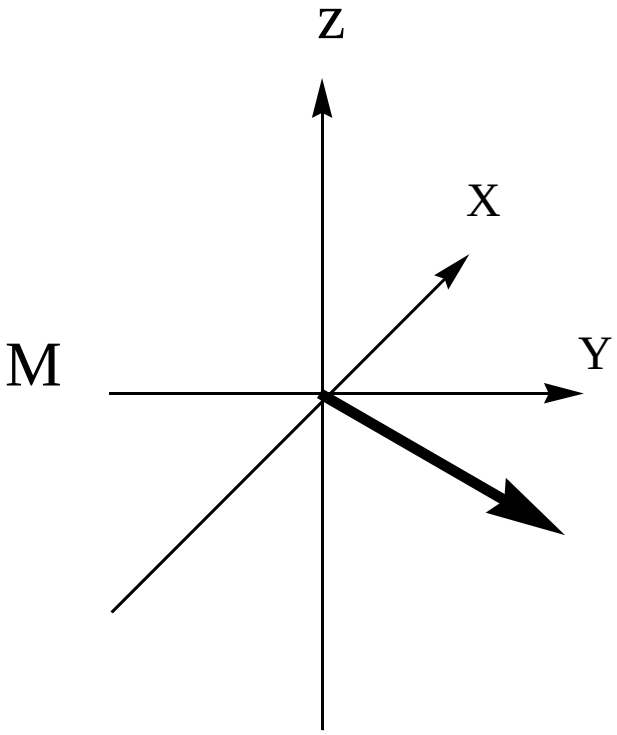
The π/2 pulse



Equilibrium

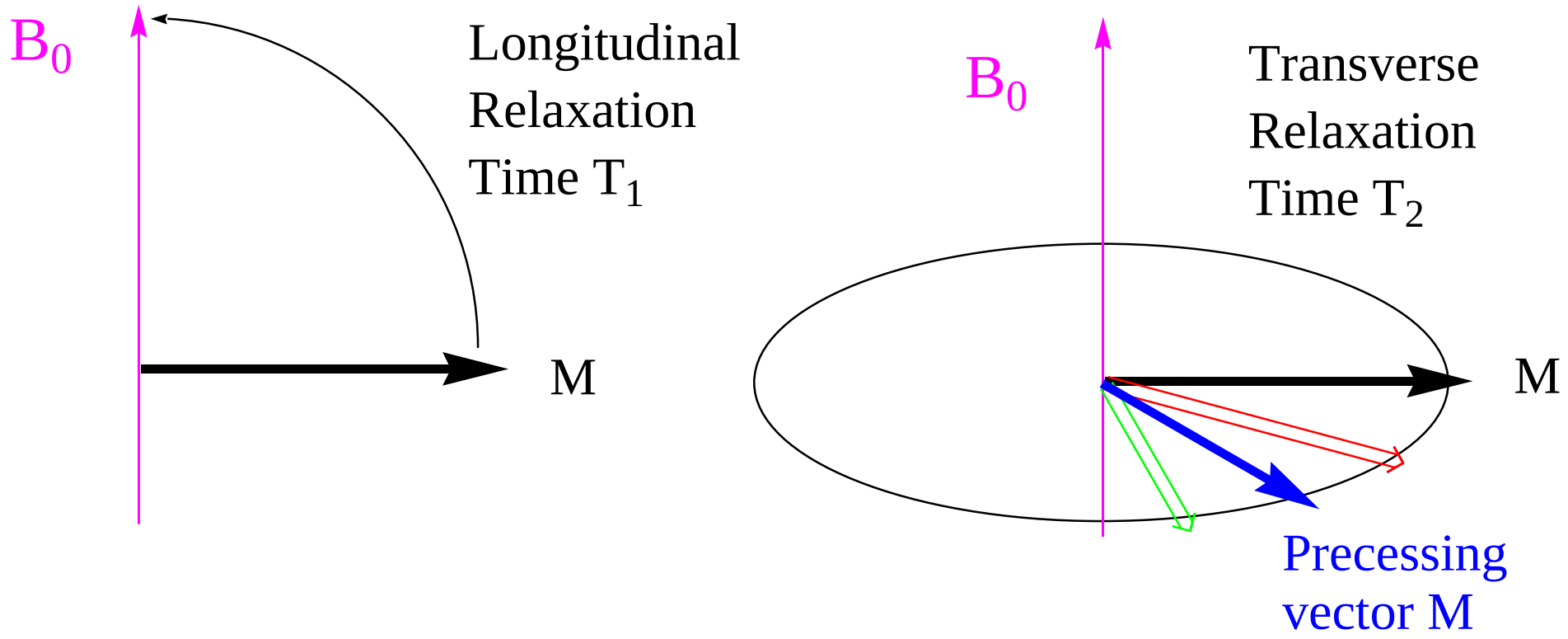


Effect of a π/2 pulse is to rotate M into the x,y plane



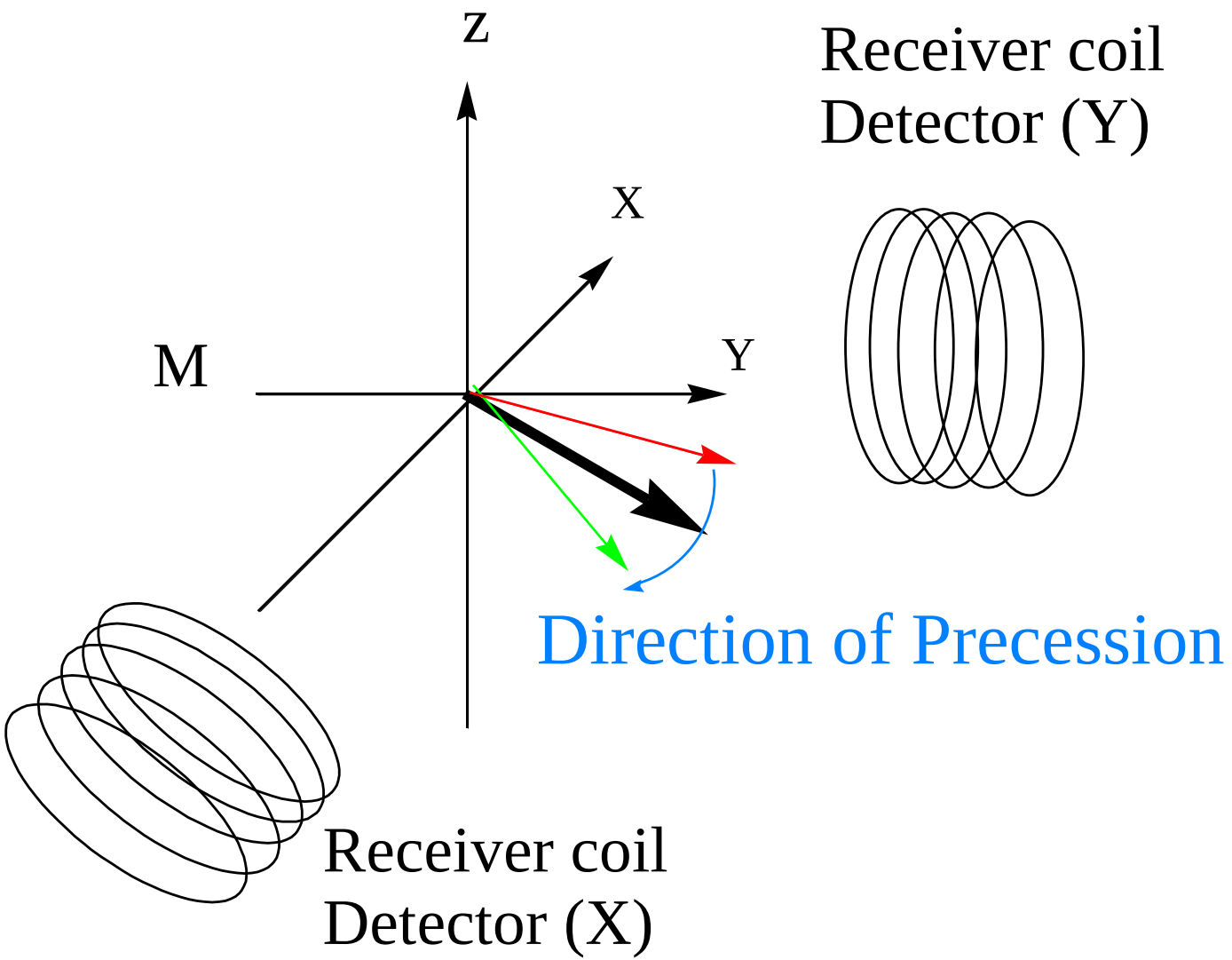
Precession in the x,y plane oscillating magnetic field free induction decay

Precession in the x,y plane



Detection of the NMR signal

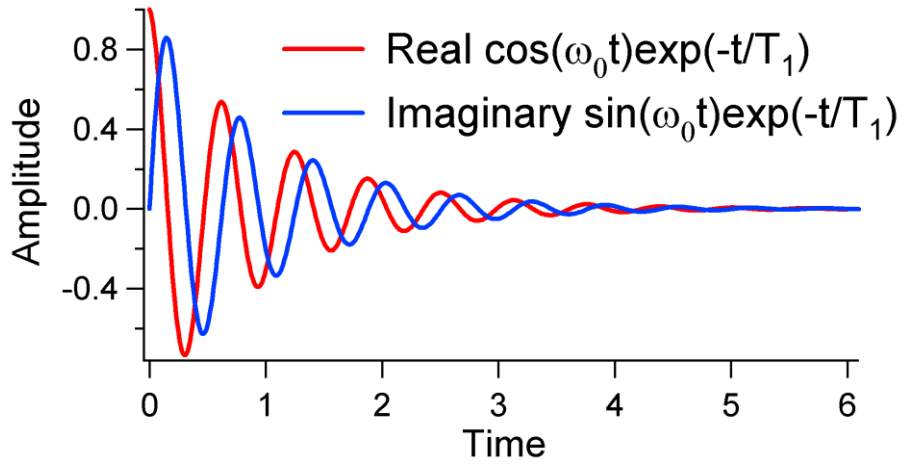
Rotating M vector



The Free Induction Decay

One example of a common Fourier transform is the free induction decay in NMR. In this instance there are both real and imaginary parts. This is because when the magnetization is rotated into the x-y plane it is detected by Coils along both x and y (out-of-phase and in-phase).

$$L(\omega) = \int_{-\infty}^{\infty} exp\{-i(\omega - \omega_0)t - \Gamma t\}$$



Relaxation times T_1 and T_2

$$FID(t) = e^{i\omega t - t/T_2}$$

$$FID(t) = \cos(\omega t)e^{-t/T_2} + isin(\omega t)e^{-t/T_2}$$

$$\frac{1}{T_2} = \frac{1}{2T_1} + \frac{1}{T_2^*}$$

Lifetime broadening

The time energy uncertainty principle is

$$\Delta\nu T_1 \geq \frac{1}{2\pi}$$

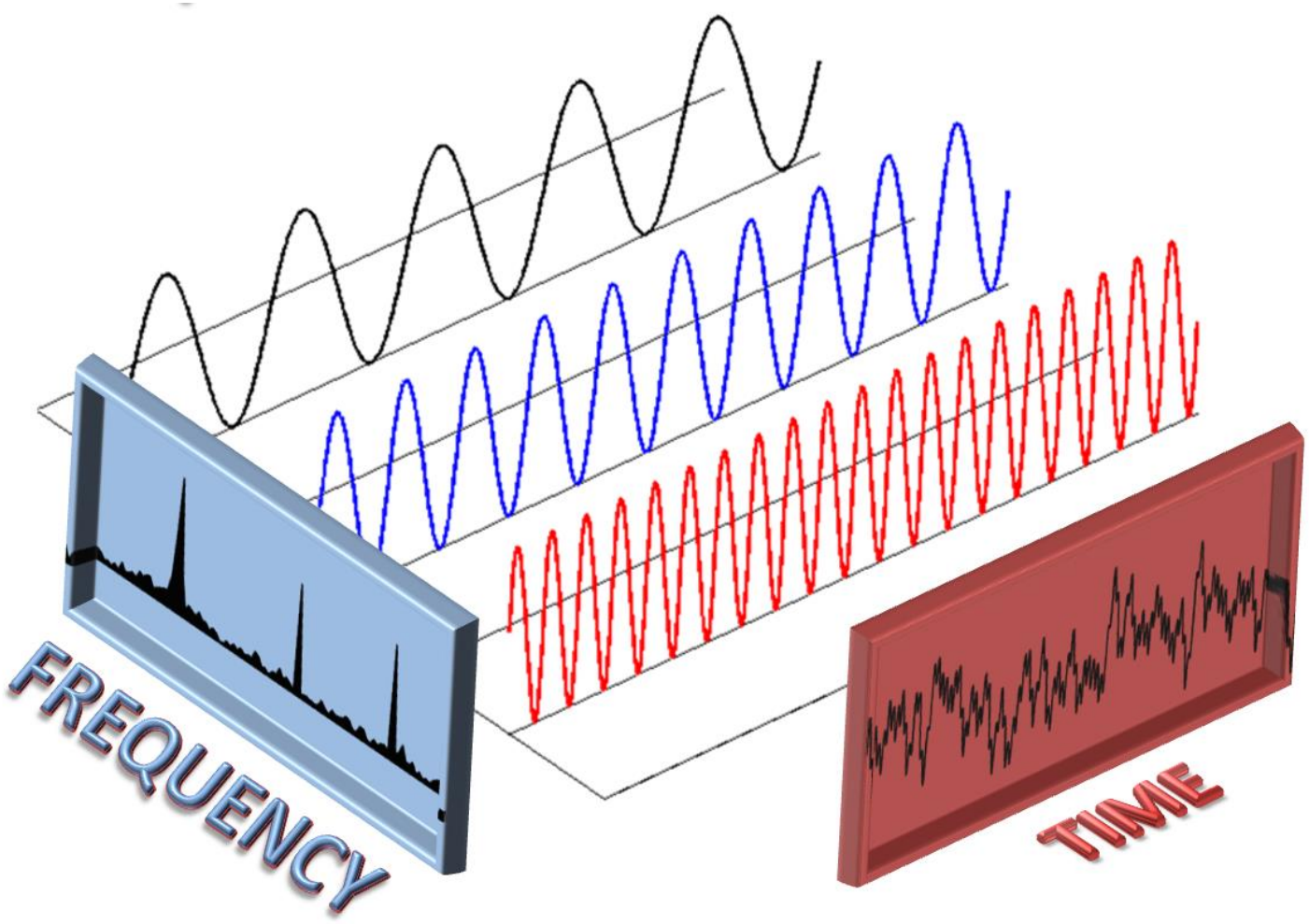
This expression can be written in terms of the units of cm^{-1} and ps.

$$\Delta\tilde{\nu}(cm^{-1}) \geq \frac{1}{2\pi c T_1(s)}$$

$$\Delta\tilde{\nu} \geq \frac{5.32\ cm^{-1}ps}{T_1}$$

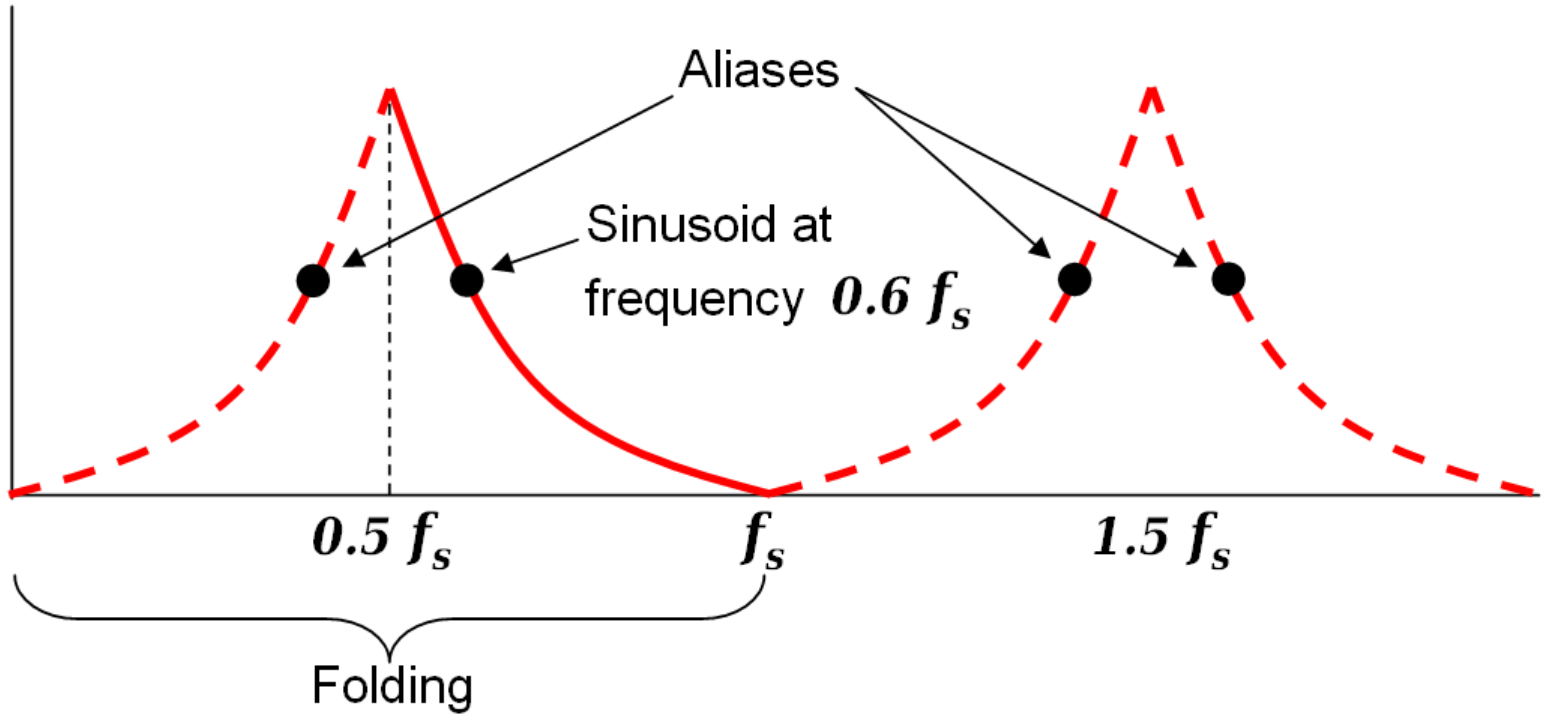
This derivation assumes that the life time is shorter than the dephasing time, which occurs for some systems. If the dephasing time, T_2 is shorter then T_2 should replace T_1 in the above equations.

Conjugate variables

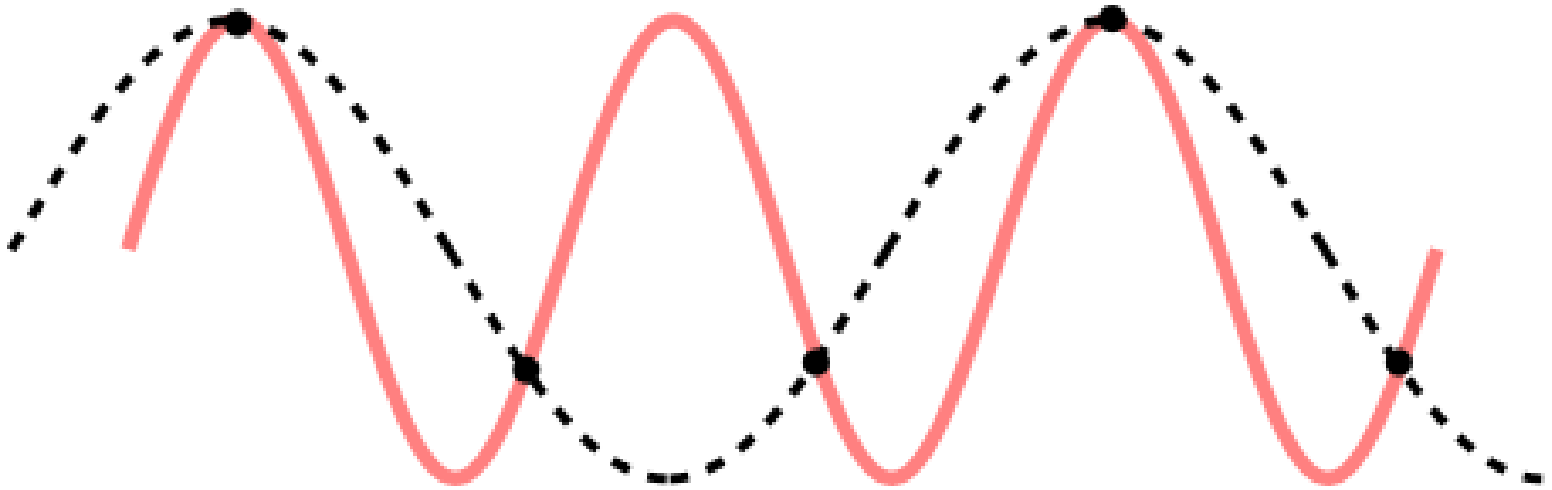


The Nyquist frequency

The Nyquist frequency



Visualizing folding: the data frequency exceeds the sampling frequency



Sampling frequency - - - - - - -

Data frequency ———

Aliasing

Referring again to Figure 1, undersampling of the sinusoid at 0.6 f_s is what allows there to be a lower-frequency *alias*, which is a different function that produces the same set of samples. That condition is usually described as *aliasing*. The mathematical algorithms that are typically used to recreate a continuous function from its samples will misinterpret the contributions of undersampled frequency components, which causes distortion. Samples of a pure 0.6 f_s sinusoid would produce a 0.4 f_s sinusoid instead. If the true frequency was 0.4 f_s, there would still be aliases at 0.6, 1.4, 1.6, etc. but the reconstructed frequency would be correct.

The Nyquist frequency

The **Nyquist frequency** is named after electronic engineer Harry Nyquist. We can think of it as the folding frequency meaning that if the sampling frequency is less than ½ of the Nyquist frequency we will see artifacts folded into the signal. In the example shown in the figure f_s is the sampling rate and 0.5 f_s is the corresponding Nyquist frequency. The black dot plotted at 0.6 f_s represents the amplitude and frequency of a sinusoidal function whose frequency is 60% of the sample-rate (f_s). The other three dots indicate the frequencies and amplitudes of three other sinusoids that would produce the same set of samples as the actual sinusoid that was sampled. The symmetry about 0.5 f_s is referred to as *folding*.

In order to recover all Fourier components of a periodic waveform, it is necessary to use a sampling rate at least twice the *highest* waveform frequency. The Nyquist frequency, or Nyquist limit, is the highest frequency that can be coded at a given sampling rate in order to be able to fully reconstruct the signal, i.e.,

$$f_{Nyquist} = \frac{1}{2}\,\nu$$

The Nyquist-Shannon theorem

Sampling is the process of converting a signal into a numeric sequence (a function of discrete time or space). This the process of digitization
The Nyquist-Shannon theorem states:

If a function $x(t)$ contains no frequencies higher than B counts per second, it is completely determined by giving its ordinates at a series of points spaced $1/(2B)$ seconds apart.

A sufficient sample-rate is at $2B$ samples/second or greater. Conversely, for a given sample rate f_s the bandlimit for perfect reconstruction is $B \leq f_s/2$. When the bandlimit is too high, the reconstruction exhibits imperfections known as aliasing.

Fast Fourier Transform

The need for a fast Fourier Transform

While NMR line shapes can be obtained in theory using an analytical function, try doing this if there are 50 nucleic oscillating in the sample.

There are many examples in science where we need FT Methods. We will use Fourier transform infrared in the class. In that method the signal is composed of the detector response at various different positions of the moving mirror in an interferometer. We must take the FT of the "interferogram" in order to obtain the optical response.

In powder X-ray diffraction the electron density is obtained as the FT of the measured intensities at various h,k,l (Miller) indices, corresponding to Bragg planes in the material.

The discrete Fourier transform

An FFT is the fastest known method to compute the Discrete Fourier Transfer (DFT). Let x_0, ..., x_{N-1} be complex numbers. The DFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}} \qquad k = 0, 1, \ldots . . N-1$$

Evaluating this definition directly requires O(N^2) operations: there are N outputs X_k, and each output requires a sum of N terms. An FFT is any method to compute the same results in O(N log N) operations.

The Cooley-Tukey algorithm

By far the most commonly used FFT is the Cooley-Tukey algorithm. This is a divide and conquer algorithm that recursively breaks down a DFT of any composite size $N = N_1 N_2$ into many smaller DFTs of sizes N_1 and N_2, along with O(N) multiplications by complex roots of unity traditionally called twiddle factors (after Gentleman and Sande, 1966).

This method (and the general idea of an FFT) was published by J.W. Cooley and J.W. Tukey in 1965, but it was later discovered that those two authors had independently re-invented an algorithm known to Gauss around 1805.

Divide and conquer

The best known use of the Cooley–Tukey algorithm is to divide the transform into two pieces of size $N/2$ at each step, and is therefore limited to power-of-two sizes, but any factorization can be used in general (as was known to both Gauss and Cooley/Tukey). These are called the **radix-2** and **mixed-radix** cases, respectively (and other variants such as the split-radix FFT have their own names as well). Although the basic idea is recursive, most traditional implementations rearrange the algorithm to avoid explicit recursion. Also, because the Cooley–Tukey algorithm breaks the DFT into smaller DFTs, it can be combined arbitrarily with any other algorithm for the DFT.

Even and odd terms

The Radix-2 DIT algorithm rearranges the DFT of the function x_n into two parts: a sum over the even-numbered indices n = 2m and a sum over the odd-numbered indices n = 2m + 1.

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{i2\pi(2m)k}{N}} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{i2\pi(2m+1)k}{N}}$$

A radix-2 decimation-in-time (DIT) FFT is the simplest and most common form of the Cooley–Tukey algorithm, although highly optimized Cooley–Tukey implementations typically use other forms of the algorithm as described below. Radix-2 DIT divides a DFT of size *N* into two interleaved DFTs (hence the name "radix-2") of size *N*/2 with each recursive stage.

Recursive approach

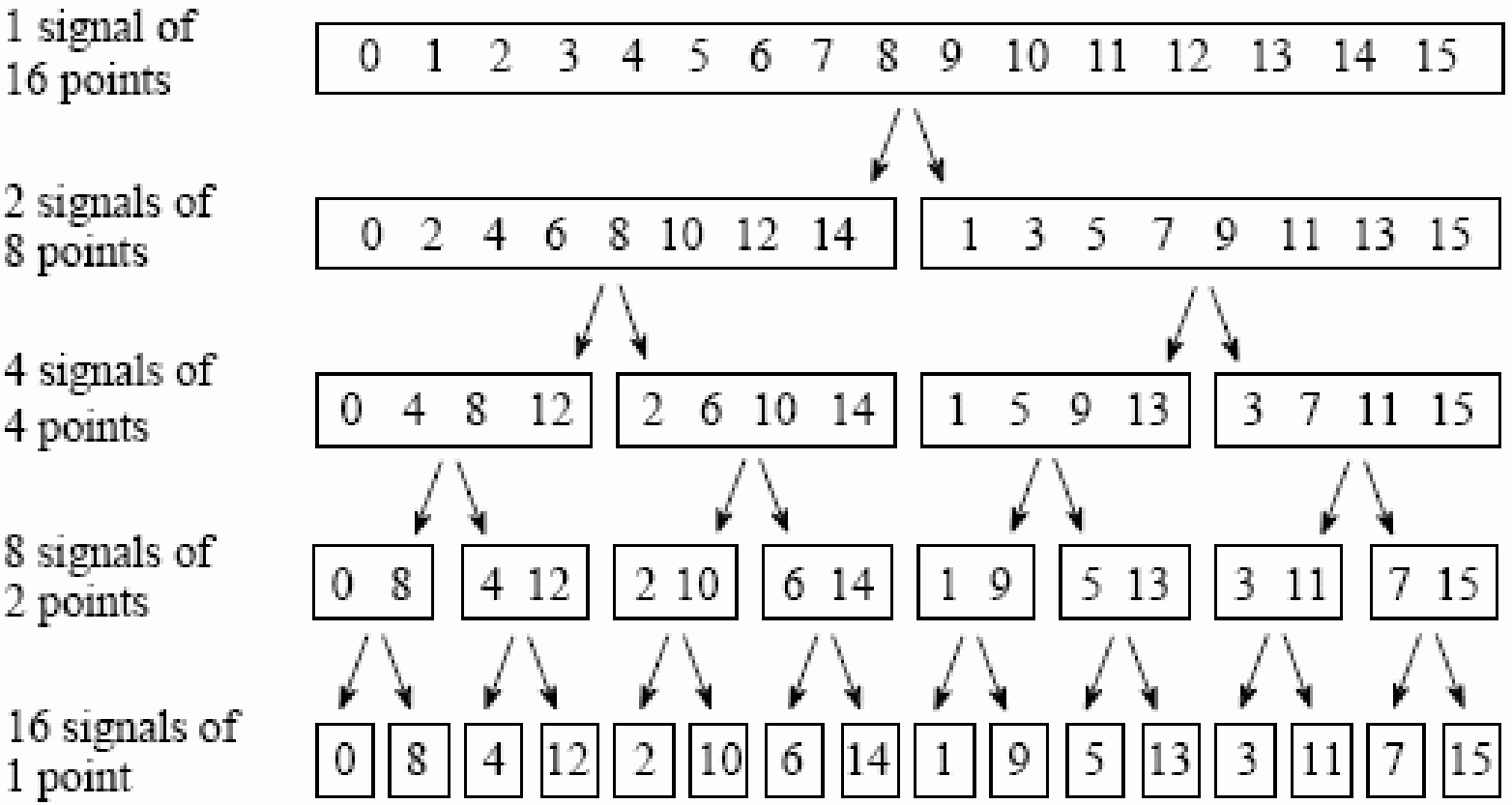
Radix-2 DIT first computes the DFTs of the even-indexed inputs $x_{2m} = x_0, x_2, \ldots x_{N-2}$ and of the odd-indexed inputs $x_{2m+1} = x_1, x_3, \ldots . x_{N-1}$, and then combines those two results to produce the DFT of the whole sequence. This idea can then be performed recursively to reduce the overall runtime to $O(N \log N)$. This simplified form assumes that N is a power of two. Since the number of sample points N can usually be chosen freely by the application, this is often not an important restriction.

The twiddle factor

One can factor a common multiplier $e^{-\frac{2\pi i}{N}k}$ out of the second sum, as shown in the equation below. It is then clear that the two sums are the DFT of the even-indexed part x_{2m} and the DFT of odd-indexed part x_{2m+1} of the function. Denote the DFT of the ***E***ven-indexed inputs x_{2m} by E_k and the DFT of the ***O***dd-indexed inputs x_{2m+1} by O_k and we obtain:

$$X_k = \underbrace{\sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk}}_{\text{DFT of even-indexed part of } x_m} + e^{-\frac{2\pi i}{N}k} \underbrace{\sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk}}_{\text{DFT of odd-indexed part of } x_m}$$

FFT decomposition



The Scientist and Engineer's Guide to Digital Signal Processing
By Steven W. Smith, Ph.D.

The FFT decomposition. An N point signal is decomposed into N signals each containing a single point. Each stage uses an *interlace decomposition*, separating the even and odd numbered samples.