## 基本概念

**傅立叶级数：**一个周期信号，可以由多个（有限或者无限）正弦和余弦信号叠加得到；即周期信号可以表示为离散频率点上的累加；其频谱是离散的

**傅立叶变换：**非周期信号，应用傅立叶级数，对周期T求无穷大极限，累加变成了积分；非周期信号可以表示在无限连续频率点上的积分（累加）；其频谱是连续的

**离散傅立叶级数：**周期为N的离散周期序列，可以由N个独立的谐波叠加得到

**离散傅立叶变换：**非周期的序列，可以表示为傅立叶积分

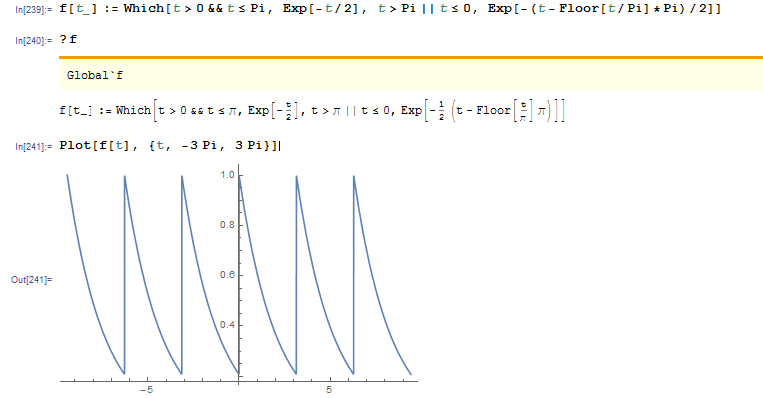
## 相关函数小结

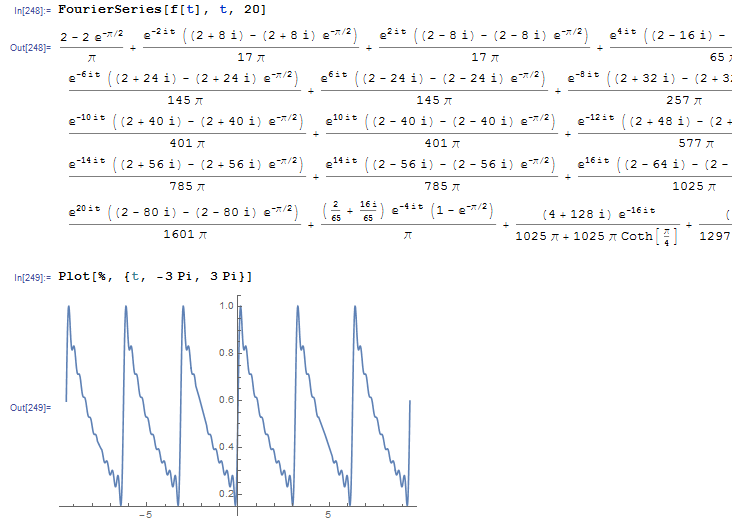


<http://reference.wolfram.com/language/guide/FourierAnalysis.html.zh>

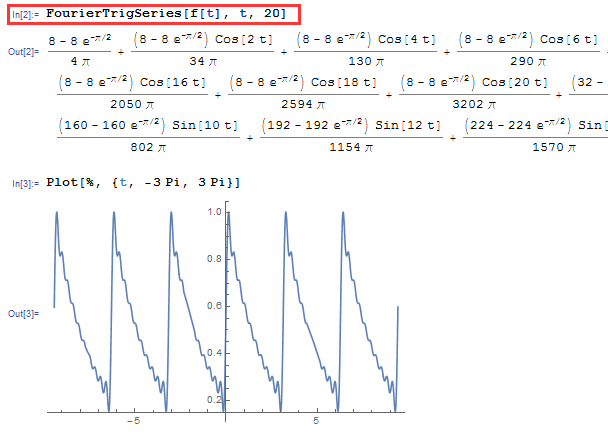
## 傅立叶级数的例子

对于一个周期信号e^(-t/2)做傅立叶级数展开





其实还有其他形式：

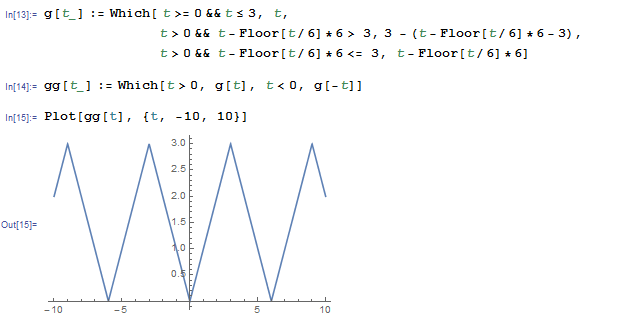


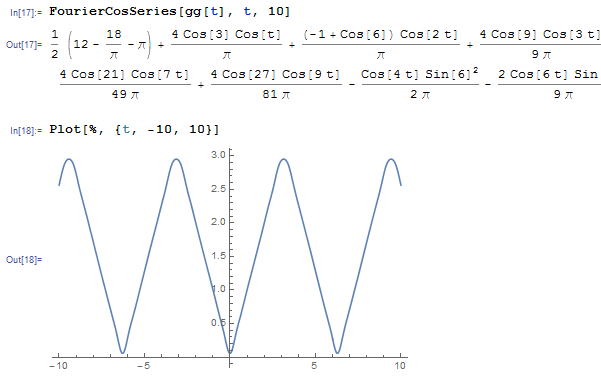
对于偶函数，只有cos分量，可以用FourierCosSeries[]来展开，

对于奇函数，只有sin分量，可以用FourierSinSeries[]来展开；

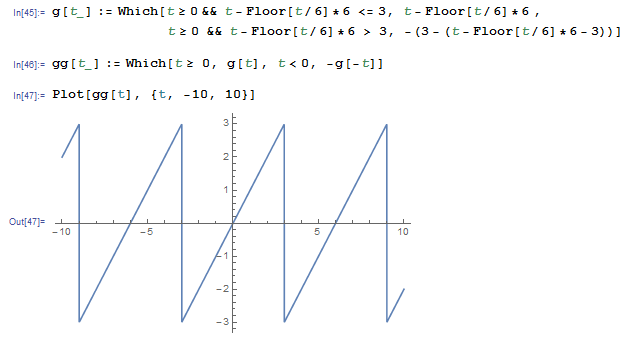
他们不是求傅立叶计数的简洁型，因为他们不能灵活处理相位。任何傅立叶级数简洁型是可以只含有cos或者sin的。

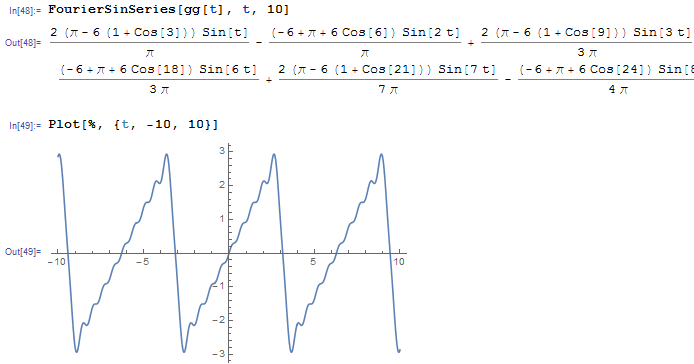
例如下面gg这个偶函数：





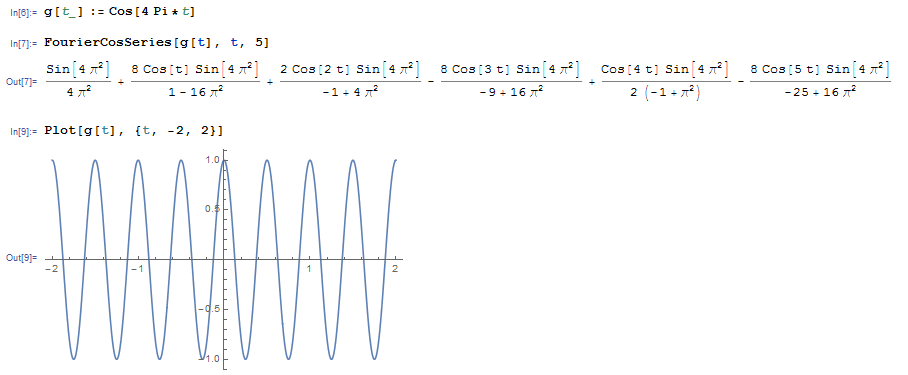
又例如gg这个奇函数:





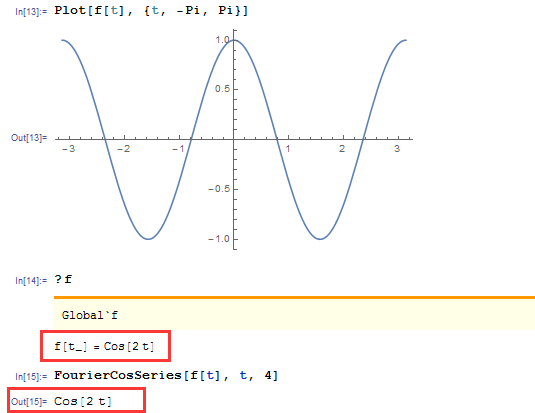
## 注意这个参数

对cos(4Pi\*t)做傅立叶级数展开



疑问：这个函数的T0=1/2，f0=2，w0=4Pi，那Cos这些谐波应该是Cos(n\* 4Pi\*t)的形式（教科书上这么说的：n乘以w0乘以t），上面的展开结果为什么不是？

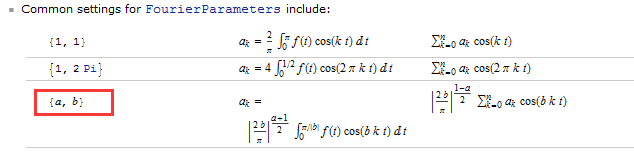
下面这个函数就分解得很好：



原来是工具使用的问题，需要加个参数来修改mathematica的惯例

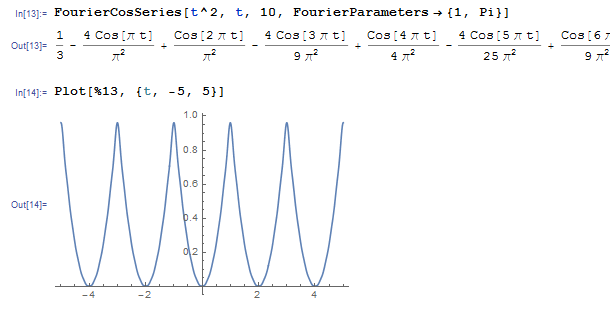


这个参数影响的是傅立叶级数的公式，例如对FourierCosSeries[]函数的影响是：



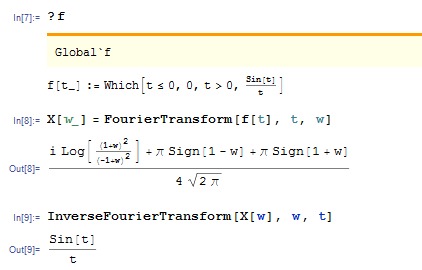
又例如，将周期为2的x(t)=t^2这样一个周期偶函数分解傅立叶级数，就可以通过FourierParameters这个选项来调整积分范围



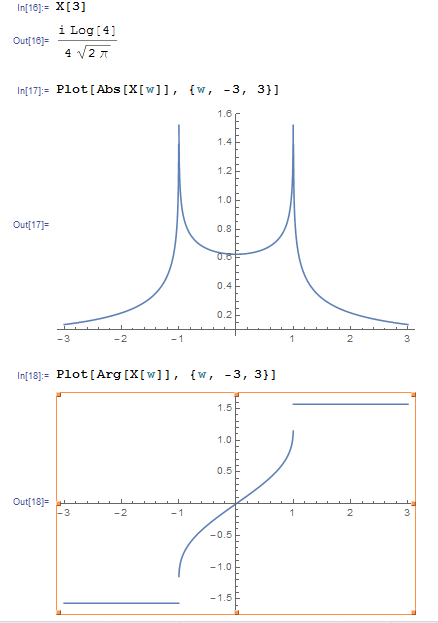




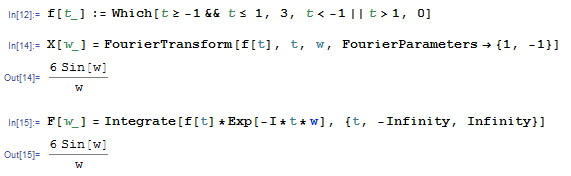
## 傅立叶变换和逆变换的例子



X[w]可能是一个复变函数，也可能是实函数，但w是实变量。下图分别给出了模和幅角的图形，（可能有更好的绘图方式）



Mathematica里FourierTransform[]函数内部使用的计算X(w)的公式，和书里的不太一样，可通过FourierParameter参数调整来达到一致：



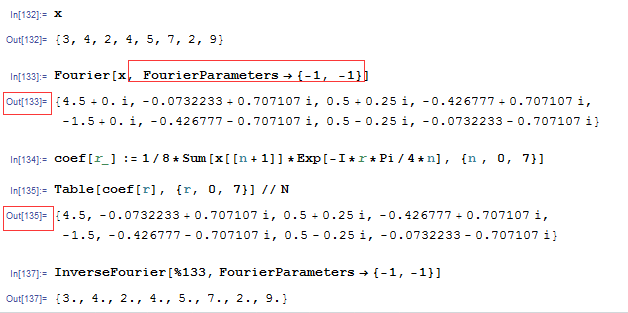
积分做傅立叶逆变换的时候，注意通过Assumptions告知MAM：t是个实数



## 离散傅立叶变换相关函数的理解：

### Fourier[]、InverseFourier[]

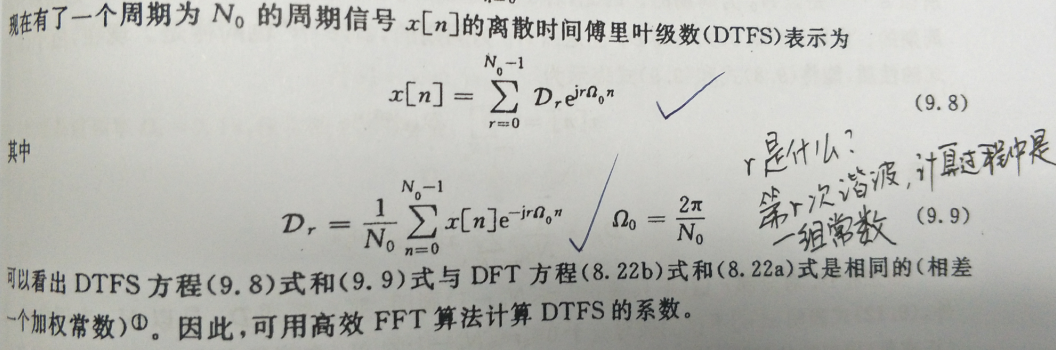
欲计算周期信号序列的离散时间傅立叶级数，可以使用Fourier[]函数来计算这些系数。下面的代码验证：该函数计算出来的系数，和教科书上的公式计算的一样：



怎么理解Fourier[]这个函数的输出呢，是一组复数，实部是cos的系数，虚部是sin的系数， 而三角函数的0.4Pi这个系数是由2Pi/N得来的，N这里等于5：

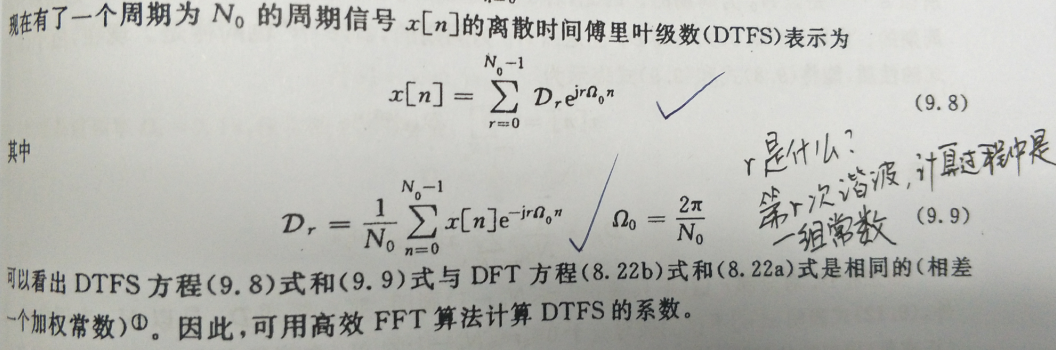


教科书上的公式如下：

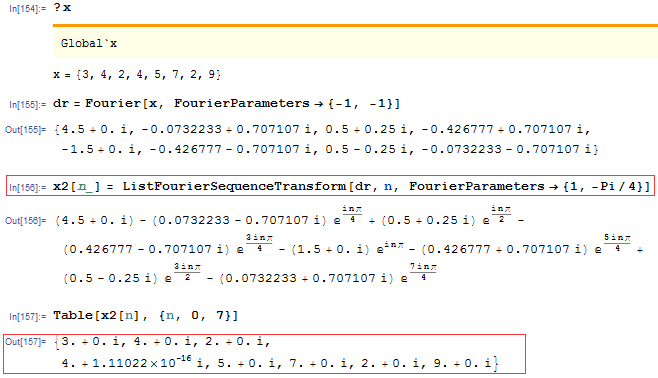


### ListFourierSequenceTransform[]

ListFourierSequenceTransform[]这个函数让人费解，似乎它的功能就是：在已知一个序列的情况下，将这个序列当作系数Dr，生成表达式9.8

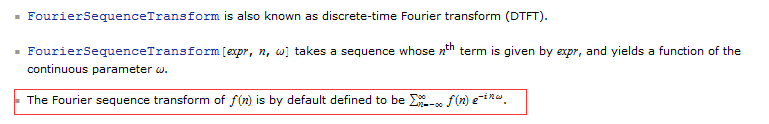


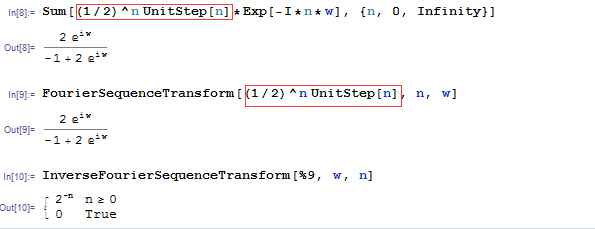
验证如下：



### FourierSequenceTransform[]

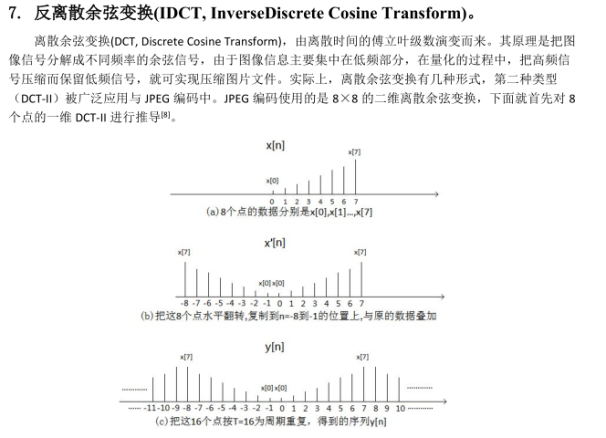
FourierSequenceTransform[]函数用来求非周期序列（离散）的傅立叶积分的X[w]的

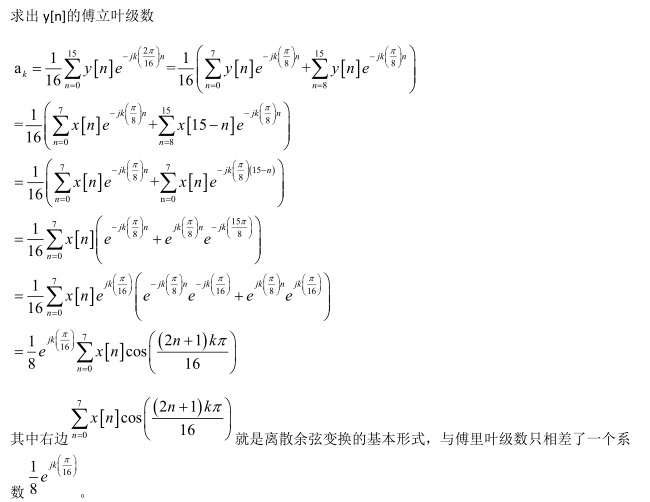


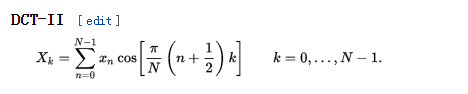


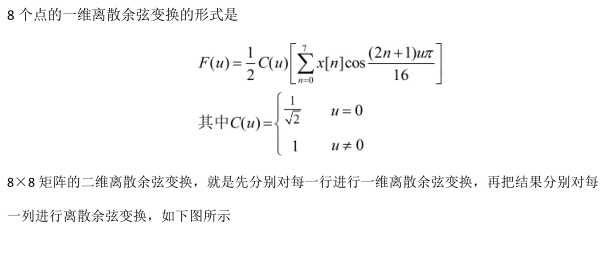
对应的有个函数InverseFourierSequenceTransform[]做反变换

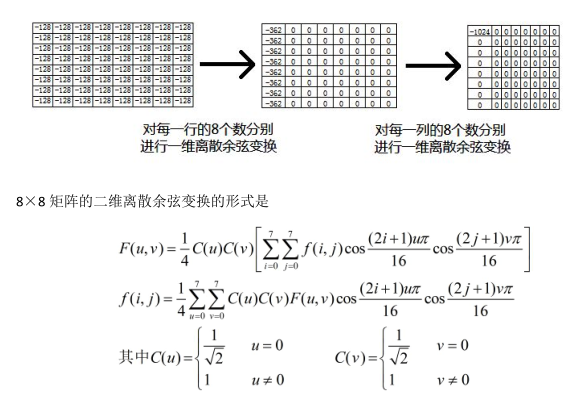
## DCT变换与离散傅立叶变换的关系



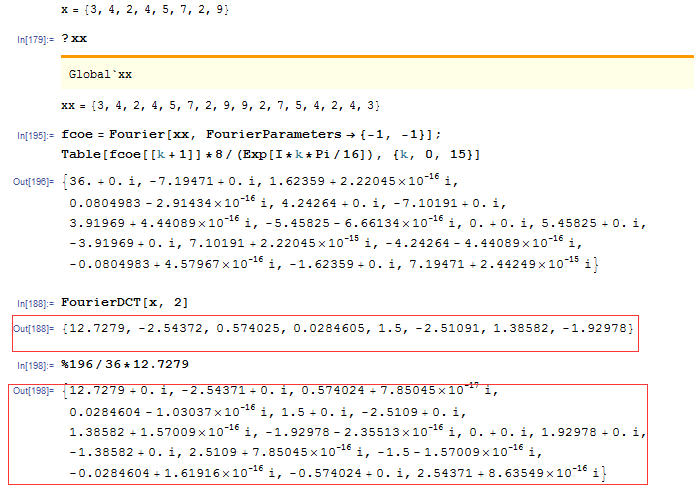






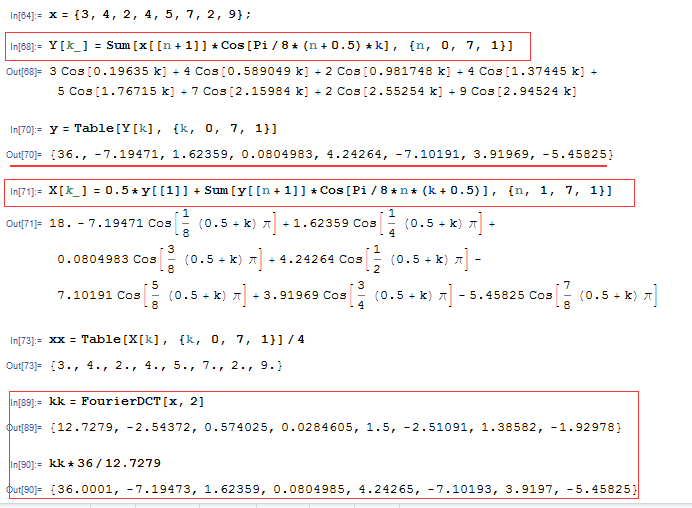


用一个数列验证上面的说法，红框里的值是相等的：



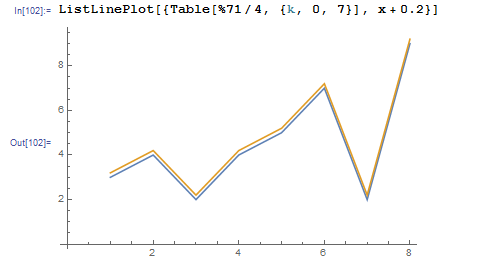
## 理解DCT

按照公式计算DCT和iDCT



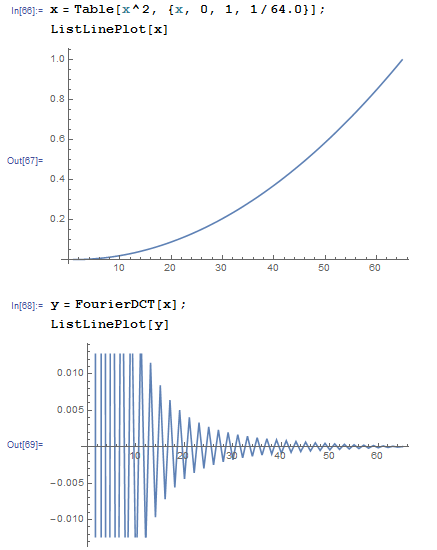
out[71]其实就可以这样来理解：原序列x，可以表示为一组cos函数相加的形式，这些cos函数的系数，就是DCT变换获得的系数，cos函数的参数包含序列的下标k

将两个数列画到同一个图表里来验证他们值是一致的，为了不要完全重合影响视觉，x加了一点偏移0.2：



## DCT变换和数据压缩

对离散值列表做DCT变换：



将频域的高频分量丢弃，然后逆变换到时域，与原来数据相比，失真很小

