This is the algorithm that calculates the DNL and INL errors. It uses samples that are generated by two small routines in the beginning of the code. These routines are called:make_sinewaveforunipolar or make_sinewaveforbipolar (they generate the sinewave) and adc_transferunipolar or adc_transferbipolar (they digitize the sinewave).There is a routine called histo(bits,sample) that calculates the number of occurences per bin.

The formulas used in the calculations of DNL and INL are shown below:

$$DNL(n)=\frac{h(n)_{ACTUAL}}{p(n)M_T} \quad (1)$$

and:

$$INL_j=\sum_{n=1}^{j} DNL_j \quad (2)$$

where h(n) is the number of occurrences per bin in the output histogram and p(n) is the probability density of the sine wave. MT is the number of samples.

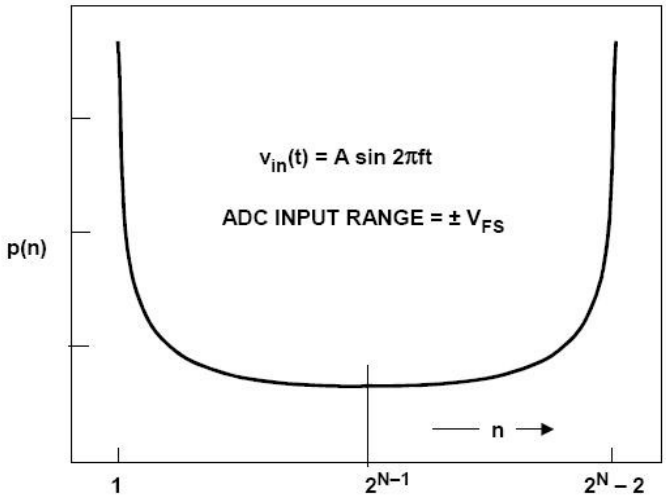
There are some notes that must to be mentioned:First of all,the number of samples MT should be very big;much bigger than the number of possible digital values that the ADC can represent. The input sine wave should be a pure sine wave without any distortion or noise. This is because of the fact that the DNL and INL errors come from the inside imperfections of the ADC and not from the input signal. In case someone wants to add some distortion on the input sine wave of frequency f,the distortion should be like that:

distortion= a1 sin 2π(2.222 f) t + a2 sin 2π(3.33333 f) t+...

where a1, a2 , … are small compared with the amplitude of the pure sine wave.

Also,the number of samples should always guarantee an integer number of sine wave periods. Also,the sine wave' s frequency and the sampling frequency should not be correlated in any way in order to receive as many different digital values ass possible,for the histogram to be reliable.

The main structure of the algorithm is the following:a histogram is made out of the digitized samples in order measure the number of occurrences per bin,meaning the number of times that a specific digital value was observed. Also, another vector is used, in which the theoretical values of the probability density function per bin are appended. p(n) should always look like the figure below. Then I used the formulas (1) and (2) to calculate a vector of DNL values and an INL value



Someone shouldn't expect zero DNL and INL values although the input is a pure sine wave. The DNL and INL errors are calculated in LSBs. To convert INL into %FSR someone should divide each INL with the number of digital values that the ADC can represent which means to divide the INL error with 2^N where N is the number of bits.