

We use a fnite sum here merely for simplicity (fewer parameters to determine). An important aspect of the spectrum is that each frequency component ***ck*** can be manipulated separately: Instead of fnding the Fourier spectrum from a time-domain specifcation, let's construct it in the frequency domain by selecting the ***ck*** according to some rule that relates coefcient values to the alphabet. In defning this rule, we want to always create a real-valued signal x (t). Because of the Fourier spectrum's

properties (Property 4.1, p. 121), the spectrum must have conjugate symmetry. This requirement means that we can only assign positive-indexed coefcients (positive frequencies), with negative-indexed ones equaling the complex conjugate of the corresponding positive-indexed ones.

Assume we have ***N*** letters to encode: {***a***1,...,***a*N** }. One simple encoding rule could be to make a single Fourier coefcient be non-zero and all others zero for each letter. For example, if ***an*** occurs , we make ***c*n** = 1 and ***ck***=0,



. In this way, the ***n***th harmonic of the frequency



is used to represent a letter. Note that the **bandwidth** the range of frequencies required for the encoding equals



Another possibility is to consider the binary representation of the letter's index. For example, if the letter a13 occurs, converting 13 to its base 2 representation, we have **13 = 11012**. We can use the pattern of zeros and ones to represent directly which Fourier coefcients we "turn on" (set equal to one) and which we "turn of."

Compare the bandwidth required for the direct encoding scheme (one nonzero Fourier coefficient for each letter) to the binary number scheme. Compare the bandwidths for a 128-letter alphabet. Since both schemes represent information without loss we can determine the typed letter uniquely from the signal's spectrum both are viable. Which makes more efficient use of bandwidth and thus might be preferred?

**Exercise 4.6.1**