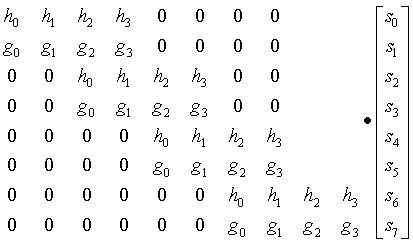
From http://www.bearcave.com/misl/misl\_tech/wavelets/daubechies/index.html

In the case of the forward transform, with a finite data set (as opposed to the mathematician's imaginary infinite data set), *i* will be incremented until it is equal to N-2. In the last iteration the inner product will be calculated from calculated from s[N-2], s[N-1], s[N] and s[N+1]. Since s[N] and s[N+1] don't exist (they are beyond the end of the array), this presents a problem. This is shown in the transform matrix below.

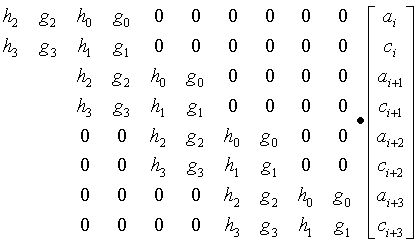
**Daubechies D4 forward transform matrix for an 8 element signal**



Note that this problem does not exist for the Haar wavelet, since it is calculated on only two elements, s[i] and s[i+1].

A similar problem exists in the case of the inverse transform. Here the inverse transform coefficients extend beyond the beginning of the data, where the first two inverse values are calculated from s[-2], s[-1], s[0] and s[1]. This is shown in the inverse transform matrix below.

**Daubechies D4 inverse transform matrix for an 8 element transform result**



Three methods for handling the edge problem:

1. Treat the data set as if it is periodic. The beginning of the data sequence repeats folling the end of the sequence (in the case of the forward transform) and the end of the data wraps around to the beginning (in the case of the inverse transform).
2. Treat the data set as if it is mirrored at the ends. This means that the data is reflected from each end, as if a mirror were held up to each end of the data sequence.
3. Gram-Schmidt orthogonalization. Gram-Schmidt orthoganalization calculates special scaling and wavelet functions that are applied at the start and end of the data set.

Zeros can also be used to fill in for the missing elements, but this can introduce significant error.