coefficient	value	coefficient	value
$h(0)$	0.8526986790	$g(-1)$	0.7884856164
$h(\pm 1)$	0.3774028556	$g(-1\pm 1)$	−0.4180922732
$h(\pm 2)$	−0.1106244044	$g(-1\pm 2)$	−0.0406894176
$h(\pm 3)$	−0.0238494650	$g(-1\pm 3)$	0.0645388826
$h(\pm 4)$	0.0378284555		

Table 5.3: Nonzero coefficients of the 9,7-biorthogonal filters.

Likewise, the *9,7-biorthogonal filters* h, g given approximately in Table 5.3 are WS about 0 and -1. All their other coefficients are zero. Exact formulas for the coefficients may be found in Daubechies on page 279, Table 8.3.

Note that neither pair h, g of WS filters are orthogonal CQFs, as they are not related by conjugacy. They nevertheless allow perfect reconstruction, as their combined action is invertible. The filter transform that inverts the h, g combined filter transform is given by their actual conjugates $\tilde{h}, \tilde{g}$, defined by

$$\tilde{h}(k) \stackrel{\text{def}}{=} (-1)^k g(-1-k); \qquad \tilde{g}(k) \stackrel{\text{def}}{=} (-1)^{k+1} h(-1-k). \tag{5.74}$$

The conjugate of a symmetric filter is also symmetric:

Lemma 5.14 *Suppose $h = \{h(k)\}$ is a filter sequence, and $g = \{g(k)\}$ is its conjugate defined by $g(k) = (-1)^k \overline{h(2M-1-k)}$, then*

1. *h is WS about C if and only if g is WS about $2M - C - 1$;*
2. *h is WA about C if and only if g is WA about $2M - C - 1$;*
3. *h is HS about C if and only if g is HA about $2M - C - 2 + \frac{1}{2}$;*
4. *h is HA about C if and only if g is HS about $2M - C - 2 + \frac{1}{2}$.*

Proof: In the first case, compute

$$\begin{aligned}
\overline{g(k)} &= (-1)^k h(2M-1-k) \;=\; (-1)^k h(2C-2M+1+k) \\
&= (-1)^k h(2M-1-[4M-2C-2-k]) \\
&= (-1)^{4M-2C-2-k} h(2M-1-[4M-2C-2-k]) \\
&= \overline{g(4M-2C-2-k)}.
\end{aligned}$$

Modified slightly, the same argument proves the second case:

$$\overline{g(k)} = (-1)^k h(2M-1-k) = -(-1)^k h(2C-2M+1+k) = -\overline{g(4M-2C-2-k)}.$$

In the third case,

$$\begin{aligned}
\overline{g(k)} &= (-1)^k h(2M-1-k) \;=\; (-1)^k h(2C+1-2M+1+k) \\
&= (-1)^k h(2M-1-[4M-2C-3-k]) \\
&= -(-1)^{4M-2C-3-k} h(2M-1-[4M-2C-3-k]) \\
&= -\overline{g(4M-2C-3-k)}.
\end{aligned}$$

In the fourth case,

$$
\begin{aligned}
\overline{g(k)} &= (-1)^k h(2M-1-k) \;=\; -(-1)^k h(2C+1-2M+1+k) \\
&= -(-1)^k h(2M-1-[4M-2C-3-k]) \\
&= (-1)^{4M-2C-3-k} h(2M-1-[4M-2C-3-k]) \\
&= \overline{g(4M-2C-3-k)}.
\end{aligned}
$$

This also follows from the third case by exchanging h and g and then substituting $C \leftarrow 2M - C - 2 + \frac{1}{2}$. □

With symmetric filters and symmetric extension and periodization, a discrete wavelet transform may be performed on signals of arbitrary length.

We begin by separating the three kinds of lifting steps. The step that computes new values at odd multiples of the increment dq is called *prediction*:

Whole-Sample Symmetric Lifting: Prediction Step

```
wslpredict( u[], N, dq, coeff ):
[0] Let i = dq
[1] While i<N-2*dq, do [2] to [3]
[2]    Sum u[i] += coeff*(u[i-dq]+u[i+dq])
[3]    Increment i += 2*dq
[4] If i+dq<N, then sum u[i] += coeff*(u[i-dq]+u[i+dq])
[5] Else sum u[i] += 2*coeff*u[i-dq]
```

Step 4 handles the odd N/dq case. Step 5 handles even N/dq by whole-sample symmetric extension.

The step that computes new values at even multiples of dq is called *updating*:

Whole-Sample Symmetric Lifting: Updating Step

```
wslupdate( u[], N, dq, coeff ):
[0] Sum u[0] += 2*coeff*u[dq]
[1] Let i = 2*dq
[2] While i<N-2*dq, do [2] to [3]
[3]    Sum u[i] += coeff*(u[i-dq]+u[i+dq])
[4]    Increment i += 2*dq
[5] If i<N, then do [6] to [7]
[6]    If i+dq<N, then sum u[i] += coeff*(u[i-dq]+u[i+dq])
[7]    Else sum u[i] += 2*coeff*u[i-dq]
```

Steps 0 and 7 perform whole-sample symmetric extension. Step 7 is supplanted by step 6 if N/dq is even. Note that the inverse of `wslpredict(u,N,dq,coeff)` is `wslpredict(u,N,dq,-coeff)`, and the inverse of `wslupdate(u,N,dq,coeff)` is `wslupdate(u,N,dq,-coeff)`.