1. Consider a multiresolution analysis and the two-scale equation for $\varphi(t)$ given in (4.2.8). Assume that $\{\varphi(t - n)\}$ is an orthonormal basis for $V_0$. Prove that

(a) $g_0[n] = \sqrt{2} \langle \varphi(2t - n), \varphi(t) \rangle$

(b) $\|g_0[n]\| = 1$.

2. In a multiresolution analysis with a scaling function $\varphi(t)$ satisfying orthonormality to its integer shifts, consider the two-scale equation (4.2.8). Assume further $0 < |\Phi(0)| < \infty$ and that $\Phi(\omega)$ is continuous in $\omega = 0$.

(a) Show that $\sum_n g_0[n] = \sqrt{2}$.

(b) Show that $\sum_n g_0[2n] = \sum_n g_0[2n + 1]$.

3. Two-Scale Equation
   Consider two scaling functions $\phi_1(t)$ and $\phi_2(t)$ each satisfying a two-scale equation
   $$\phi_i(t) = 2 \sum_{k=0}^{N_i} c_i[k] \phi_i(2t - k), \quad i = 1, 2$$
   with Fourier transforms
   $$\Phi_i(\omega) = C_i(e^{j\omega/2}) \Phi_i(\omega/2) = \prod_{k=1}^{\infty} C_i(e^{j\omega/2^k}), \quad i = 1, 2$$
   Consider $\phi_3(t) = \phi_1(t) \ast \phi_2(t)$ (i.e. convolution) and $\phi_4(t) = \phi_1(t) \cdot \phi_2(t)$ (i.e. multiplication).

   (a) Prove that $\phi_3(t)$ satisfies a two-scale equation.
   (b) Give the coefficients $c_3[k]$ involved in the two-scale equation of $\phi_3(t)$.
   (c) Show that $\phi_4(t)$ does not satisfy a two-scale equation in general (e.g. construct a simple counter example).

4. Matlab exercise - Spectral factorization and Iterated Filterbanks
   (a) In this problem we will consider a practical example of spectral factorization. The following Matlab function is meant to return the Daubechies minimum-phase filter $D_k$ for any $k$, following the procedure illustrated on pages 130–131 of the textbook. The code provided below computes the coefficients of the polynomial $R(z)$, following the notation of the book. Complete the daub function by implementing a spectral factorization algorithm for $R(z)$ and by computing the corresponding Daubechies filter’s coefficients. Compare your results for $k = 2$ and $k = 4$ to the filter values given on page 131 and 129 respectively; they should be identical. (Hint: consider the orthonormality requirement when it comes to normalizing the coefficients.)
function f = daub(k)

% Length of R(z), which is also the order of the filter
r = 2*k-1;

% Let's compute the coefficients of (1 + z^-1)^k * (1 + z)^k
% Remember that polynomial multiplication is equivalent to convolution
p = [1 1]; % This can represent both (1 + z^-1) and (1 + z)
for n=1:k-1
    p = conv(p, [1 1]);
end
pz = conv(p, p);

% Now we want to build a convolution matrix A so that AR , with
% R = [r(k) r(k-1) ... r(0) ... r(k)]', gives us the coefficients of
% the even powers of P(z).

% Time-reverse (not necessary due to symmetry) and zero-pad pz
pz = [zeros(1,r) pz zeros(1,r)];
l = length(pz)

A = zeros(r,r);
for n=1:r
    A(n,:) = pz(l-2*n-r+1:l-2*n);
end

% Now we solve AR = B, where B is zero for all odd powers except for z^0
B = zeros(r,1);
B((r+1)/2) = 1;
R = A\B;

(b) Next we will explore graphically some properties of iterated filterbanks.
Write an m-function iter(f,N) which returns the N-th iteration of the filter specified by the vector f; this would be the impulse response of the lower branch of an iterated filterbank of depth N. Plot the 8th iteration for the length-4 Daubechies filter D2 (you can compute the taps with the completed daub.m function from part (a)). Plot also the 8th iteration of the length-8 Smith and Barnwell filter of page 129 in the book. Compare the behavior of the two filters with respect to iteration.