Answers

Answer Ex. 1.4. 1: Setting p_{ref}=0.00002 Pa and p=100 000 Pa in the decibel expression we get

$$20\log_{10}\left(\frac{p}{p_{\text{ref}}}\right) = 20\log_{10}\left(\frac{100000}{0.00002}\right) = 20\log_{10}\left(\frac{10^5}{2\times 10^{-5}}\right)$$
$$= 20\log_{10}\left(\frac{10^{10}}{2}\right) = 20\,(10 - \log_{10} 2) \approx 194\text{db}.$$

Answer Ex. 1.4. 3: The important thing to note here is that there are two oscillations present in Figure 1.1(b): One slow oscillation with a higher amplitude, and one faster oscillation, with a lower amplitude. We see that there are 10 periods of the smaller oscillation within one period of the larger oscillation, so that we should be able to reconstruct the figure by using frequencies where one is 10 times the other, such as 440Hz and 4400Hz. Also, we see from the figure that the amplitude of the larger oscillation is close to 1, and close to 0.3 for the smaller oscillation. A good choice therefore seems to be $a = 1, b = 0.3$.

Answer Ex. 1.4. 4: The code can look like this:

```
function playpuresound(f)
  fs=44100;
  t=0:(1/fs):3;
  sd=sin(2*pi*f*t);
  playerobj=audioplayer(sd,fs);
  playblocking(playerobj)
```

Answer Ex. 1.4. 5: The code can look like this:

```
function playsquare(T)
  % Play a square wave with period T over 3 seconds
```

```
  fs=44100;
  samplesperperiod=round(fs*T);
  oneperiod=[ones(1,round(samplesperperiod/2)) ...
             -ones(1,round(samplesperperiod/2))];
  allsamples=zeros(1,floor(3/T)*length(oneperiod));
  for k=1:floor(3/T)
    allsamples(((k-1)*length(oneperiod)+1):k*length(oneperiod))=oneperiod;
  end
  playerobj=audioplayer(allsamples,fs);
  playblocking(playerobj)
```

```
function playtriangle(T)
  % Play a triangle wave with period T over 3 seconds
  fs=44100;
  samplesperperiod=round(fs*T);
  oneperiod=[linspace(-1,1,round(samplesperperiod/2)) ...
           linspace(1,-1,round(samplesperperiod/2))];
  allsamples=zeros(1,floor(3/T)*length(oneperiod));
  for k=1:floor(3/T)
    allsamples(((k-1)*length(oneperiod)+1):k*length(oneperiod))=oneperiod;
  end
  playerobj=audioplayer(allsamples,fs);
  playblocking(playerobj)
```

Answer Ex. 1.4. 6: The code can look like this:

```
function playdifferentfs()
  [S fs]=wavread('castanets.wav');
  playerobj=audioplayer(S,fs);
  playblocking(playerobj);
  playerobj=audioplayer(S,2*fs);
  playblocking(playerobj);
  playerobj=audioplayer(S,fs/2);
  playblocking(playerobj);
```

```
function playreverse()
  [S fs]=wavread('castanets.wav');
  sz=size(S,1);
  playerobj=audioplayer(S(sz:(-1):1,:),fs);
  playblocking(playerobj);
```

Answer Ex. 1.4. 7: The code can look like this:

```
function playnoise(c)
  [S fs]=wavread('castanets.wav');
  sz=size(S,1);
  newS=S+c*(2*rand(sz,2)-1);
  newS=newS/max(max(abs(newS)));
  playerobj=audioplayer(newS,fs);
  playblocking(playerobj);
```

Answer Ex. 1.4. 8: The code can look like this:

```
function playwithecho(c,d)
  [S fs]=wavread('castanets.wav');
  sz=size(S,1);
  newS=S((d+1):sz,:)-0.5*S(1:(sz-d),:);
  newS=newS/max(max(abs(newS)));
  playerobj=audioplayer(newS,fs);
  playblocking(playerobj);
```

Answer Ex. 1.4. 9: The code can look like this:

```
function reducebass(k)
  c=[1/2 1/2];
  for z=1:(2*k-1)
    c=conv(c,[1/2 1/2]);
  end
  c=(-1).^(0:(2*k)).*c;
  [S fs]=wavread('castanets.wav');
  N=size(S,1);

  y=zeros(N,2);
  y(1:k,:)=S(1:k,:);
  for t=(k+1):(N-k)
    for j=1:(2*k+1)
      y(t,:)=y(t,:)+c(j)*S(t+k+1-j,:);
    end
  end
  y((N-k+1):N,:)=S((N-k+1):N,:);
  y=y/max(max(abs(y)));

  playerobj=audioplayer(y,fs);
  playblocking(playerobj);
```

```
function reducetreble(k)
  c=[1/2 1/2];
  for z=1:(2*k-1)
    c=conv(c,[1/2 1/2]);
  end
  [S fs]=wavread('castanets.wav');
  N=size(S,1);

  y=zeros(N,2);
  y(1:k,:)=S(1:k,:);
  for t=(k+1):(N-k)
    for j=1:(2*k+1)
      y(t,:)=y(t,:)+c(j)*S(t+k+1-j,:);
    end
  end
  y((N-k+1):N,:)=S((N-k+1):N,:);

  playerobj=audioplayer(y,fs);
  playblocking(playerobj);
```

Answer Ex. 2.1. 1: The function $f(t) = \frac{1}{\sqrt{t}} = t^{-1/2}$ can be used since it has the properties

$$\begin{aligned}
\int_0^T f(t)dt &= \lim_{x\to 0+} \int_x^T t^{-1/2}dt = \lim_{x\to 0+} \left[2t^{1/2}\right]_x^T \\
&= \lim_{x\to 0+} (2T^{1/2} - 2x^{1/2}) = 2T^{1/2} \\
\int_0^T f(t)^2 dt &= \lim_{x\to 0+} \int_x^T t^{-1}dt = \lim_{x\to 0+} [\ln t]_x^T \\
&= \ln T - \lim_{x\to 0+} \ln x = \infty.
\end{aligned}$$

Answer Ex. 2.1. 4: For $f(t) = t$ we get that $a_0 = \frac{1}{T}\int_0^T t dt = \frac{T}{2}$. We also get

$$\begin{aligned}
a_n &= \frac{2}{T}\int_0^T t\cos(2\pi nt/T)dt \\
&= \frac{2}{T}\left(\left[\frac{T}{2\pi n}t\sin(2\pi nt/T)\right]_0^T - \frac{T}{2\pi n}\int_0^T \sin(2\pi nt/T)dt\right) = 0 \\
b_n &= \frac{2}{T}\int_0^T t\sin(2\pi nt/T)dt \\
&= \frac{2}{T}\left(\left[-\frac{T}{2\pi n}t\cos(2\pi nt/T)\right]_0^T + \frac{T}{2\pi n}\int_0^T \cos(2\pi nt/T)dt\right) = -\frac{T}{\pi n}.
\end{aligned}$$

The Fourier series is thus

$$\frac{T}{2} - \sum_{n\geq 1}\frac{T}{\pi n}\sin(2\pi nt/T).$$

Note that this is almost a sine series, since it has a constant term, but no other cosine terms. If we had subtracted $T/2$ we would have obtained a function which is antisymmetric, and thus a pure sine series.
For $f(t) = t^2$ we get that $a_0 = \frac{1}{T}\int_0^T t^2 dt = \frac{T^2}{3}$. We also get

$$\begin{aligned}
a_n &= \frac{2}{T}\int_0^T t^2\cos(2\pi nt/T)dt \\
&= \frac{2}{T}\left(\left[\frac{T}{2\pi n}t^2\sin(2\pi nt/T)\right]_0^T - \frac{T}{\pi n}\int_0^T t\sin(2\pi nt/T)dt\right) \\
&= \left(-\frac{T}{\pi n}\right)\left(-\frac{T}{\pi n}\right) = \frac{T^2}{\pi^2 n^2} \\
b_n &= \frac{2}{T}\int_0^T t^2\sin(2\pi nt/T)dt \\
&= \frac{2}{T}\left(\left[-\frac{T}{2\pi n}t^2\cos(2\pi nt/T)\right]_0^T + \frac{T}{\pi n}\int_0^T t\cos(2\pi nt/T)dt\right) \\
&= -\frac{T^2}{\pi n}.
\end{aligned}$$

Here we see that we could use the expressions for the Fourier coefficients of $f(t) = t$ to save some work. The Fourier series is thus

$$\frac{T^2}{3} + \sum_{n\geq 1}\left(\frac{T^2}{\pi^2 n^2}\cos(2\pi nt/T) - \frac{T^2}{\pi n}\sin(2\pi nt/T)\right).$$

For $f(t) = t^3$ we get that $a_0 = \frac{1}{T}\int_0^T t^3 dt = \frac{T^3}{4}$. We also get

$$
\begin{aligned}
a_n &= \frac{2}{T}\int_0^T t^3 \cos(2\pi nt/T)dt \\
&= \frac{2}{T}\left(\left[\frac{T}{2\pi n}t^3 \sin(2\pi nt/T)\right]_0^T - \frac{3T}{2\pi n}\int_0^T t^2 \sin(2\pi nt/T)dt\right) \\
&= \left(-\frac{3T}{2\pi n}\right)\left(-\frac{T^2}{\pi n}\right) = \frac{3T^3}{2\pi^2 n^2} \\
b_n &= \frac{2}{T}\int_0^T t^3 \sin(2\pi nt/T)dt \\
&= \frac{2}{T}\left(\left[-\frac{T}{2\pi n}t^3 \cos(2\pi nt/T)\right]_0^T + \frac{3T}{2\pi n}\int_0^T t^2 \cos(2\pi nt/T)dt\right) \\
&= -\frac{T^3}{\pi n} + \frac{3T}{2\pi n}\frac{T^2}{\pi^2 n^2} = -\frac{T^3}{\pi n} + \frac{3T^3}{2\pi^3 n^3}.
\end{aligned}
$$

Also here we saved some work, by reusing the expressions for the Fourier coefficients of $f(t) = t^2$. The Fourier series is thus

$$
\frac{T^3}{4} + \sum_{n\geq 1}\left(\frac{3T^3}{2\pi^2 n^2}\cos(2\pi nt/T) + \left(-\frac{T^3}{\pi n} + \frac{3T^3}{2\pi^3 n^3}\right)\sin(2\pi nt/T)\right).
$$

We see that all three Fourier series converge slowly. This is connected to the fact that none of the functions are continuous at the borders of the periods.

Answer Ex. 2.1. 5: Let us define $a_{n,k}, b_{n,k}$ as the Fourier coefficients of t^k. When $k > 0$ and $n > 0$, integration by parts gives us the following difference equations:

$$
\begin{aligned}
a_{n,k} &= \frac{2}{T}\int_0^T t^k \cos(2\pi nt/T)dt \\
&= \frac{2}{T}\left(\left[\frac{T}{2\pi n}t^k \sin(2\pi nt/T)\right]_0^T - \frac{kT}{2\pi n}\int_0^T t^{k-1} \sin(2\pi nt/T)dt\right) \\
&= -\frac{kT}{2\pi n}b_{n,k-1} \\
b_{n,k} &= \frac{2}{T}\int_0^T t^k \sin(2\pi nt/T)dt \\
&= \frac{2}{T}\left(\left[-\frac{T}{2\pi n}t^k \cos(2\pi nt/T)\right]_0^T + \frac{kT}{2\pi n}\int_0^T t^{k-1} \cos(2\pi nt/T)dt\right) \\
&= -\frac{T^k}{\pi n} + \frac{kT}{2\pi n}a_{n,k-1}.
\end{aligned}
$$

When $n > 0$, these can be used to express $a_{n,k}, b_{n,k}$ in terms of $a_{n,0}, b_{n,0}$, for which we clearly have $a_{n,0} = b_{n,0} = 0$. For $n = 0$ we have that $a_{0,k} = \frac{T^k}{k+1}$ for all k. The following program computes $a_{n,k}, b_{n,k}$ recursively when $n > 0$.

```
function [ank,bnk]=findfouriercoeffs(n,k,T)
  ank=0; bnk=0;
  if k>0
    [ankprev,bnkprev]=findfouriercoeffs(n,k-1,T)
    ank=-k*T*bnkprev/(2*pi*n);
    bnk=-T^k/(pi*n) + k*T*ankprev/(2*pi*n);
  end
```

Answer Ex. 2.1. 7: The code can look like this:

```
function playsquaretrunk(T,N)
  fs=44100;
  t=0:(1/fs):3;
  sd=zeros(1,length(t));
  n=1;
  while n<=N
    sd = sd + (4/(n*pi))*sin(2*pi*n*t/T);
    n=n+2;
  end
  playerobj=audioplayer(sd,fs);
  playblocking(playerobj)
```

```
function playtriangletrunk(T,N)
  fs=44100;
  t=0:(1/fs):3;
  sd=zeros(1,length(t));
  n=1;
  while n<=N
    sd = sd - (8/(n^2*pi^2))*cos(2*pi*n*t/T);
    n=n+2;
  end
  playerobj=audioplayer(sd,fs);
  playblocking(playerobj)
```

Answer Ex. 2.2. 1: For $n_1 \neq n_2$ we have that

$$\begin{aligned}\langle e^{2\pi i n_1 t/T}, e^{2\pi i n_2 t/T}\rangle &= \frac{1}{T}\int_0^T e^{2\pi i n_1 t/T} e^{-2\pi i n_2 t/T} dt = \frac{1}{T}\int_0^T e^{2\pi i(n_1-n_2)t/T} dt \\ &= \left[\frac{T}{2\pi i(n_1-n_2)} e^{2\pi i(n_1-n_2)t/T}\right]_0^T \\ &= \frac{T}{2\pi i(n_1-n_2)} - \frac{T}{2\pi i(n_1-n_2)} = 0.\end{aligned}$$

When $n_1 = n_2$ the integrand computes to 1, so that $\|e^{2\pi i n t/T}\| = 1$.

Answer Ex. 2.2. 5: We obtain that

$$\begin{aligned} y_n &= \frac{1}{T}\int_0^{T/2} e^{-2\pi i n t/T} dt - \frac{1}{T}\int_{T/2}^{T} e^{-2\pi i n t/T} dt \\ &= -\frac{1}{T}\left[\frac{T}{2\pi i n} e^{-2\pi i n t/T}\right]_0^{T/2} + \frac{1}{T}\left[\frac{T}{2\pi i n} e^{-2\pi i n t/T}\right]_{T/2}^{T} \\ &= \frac{1}{2\pi i n}\left(-e^{-\pi i n} + 1 + 1 - e^{-\pi i n} +\right) \\ &= \frac{1}{\pi i n}\left(1 - e^{-\pi i n}\right) = \begin{cases} 0, & \text{if } n \text{ is even;} \\ 2/(\pi i n), & \text{if } n \text{ is odd.}\end{cases}. \end{aligned}$$

Instead using Theorem 2.11 together with the coefficients $b_n = \frac{2(1-\cos(n\pi))}{n\pi}$ we computed in Example 2.6, we obtain

$$y_n = \frac{1}{2}(a_n - i b_n) = -\frac{1}{2} i \begin{cases} 0, & \text{if } n \text{ is even;} \\ 4/(n\pi), & \text{if } n \text{ is odd.}\end{cases} = \begin{cases} 0, & \text{if } n \text{ is even;} \\ 2/(\pi i n), & \text{if } n \text{ is odd.}\end{cases}$$

when $n > 0$. The case $n < 0$ follows similarly.

Answer Ex. 2.2. 7: For $f(t) = t$ we get

$$\begin{aligned} y_n &= \frac{1}{T}\int_0^T t e^{-2\pi i n t/T} dt = \frac{1}{T}\left(\left[-\frac{T}{2\pi i n} t e^{-2\pi i n t/T}\right]_0^T + \int_0^T \frac{T}{2\pi i n} e^{-2\pi i n t/T} dt\right) \\ &= -\frac{T}{2\pi i n} = \frac{T}{2\pi n} i. \end{aligned}$$

From Exercise 4 we had $b_n = -\frac{T}{\pi n}$, for which Theorem 2.11 gives $y_n = \frac{T}{2\pi n} i$ for $n > 0$, which coincides with the expression we obtained. The case $n < 0$ follows similarly.

For $f(t) = t^2$ we get

$$y_n = \frac{1}{T}\int_0^T t^2 e^{-2\pi int/T}dt = \frac{1}{T}\left(\left[-\frac{T}{2\pi in}t^2 e^{-2\pi int/T}\right]_0^T + 2\int_0^T \frac{T}{2\pi in}te^{-2\pi int/T}dt\right)$$
$$= -\frac{T^2}{2\pi in} + \frac{T^2}{2\pi^2 n^2} = \frac{T^2}{2\pi^2 n^2} + \frac{T^2}{2\pi n}i.$$

From Exercise 4 we had $a_n = \frac{T^2}{\pi^2 n^2}$ and $b_n = -\frac{T^2}{\pi n}$, for which Theorem 2.11 gives $y_n = \frac{1}{2}\left(\frac{T^2}{\pi^2 n^2} + i\frac{T^2}{\pi n}\right)$ for $n > 0$, which also is seen to coincide with what we obtained. The case $n < 0$ follows similarly.
For $f(t) = t^3$ we get

$$y_n = \frac{1}{T}\int_0^T t^3 e^{-2\pi int/T}dt = \frac{1}{T}\left(\left[-\frac{T}{2\pi in}t^3 e^{-2\pi int/T}\right]_0^T + 3\int_0^T \frac{T}{2\pi in}t^2 e^{-2\pi int/T}dt\right)$$
$$= -\frac{T^3}{2\pi in} + 3\frac{T}{2\pi in}(\frac{T^2}{2\pi^2 n^2} + \frac{T^2}{2\pi n}i) = 3\frac{T^3}{4\pi^2 n^2} + \left(\frac{T^3}{2\pi n} - 3\frac{T^3}{4\pi^3 n^3}\right)i =$$

From Exercise 4 we had $a_n = \frac{3T^3}{2\pi^2 n^2}$ and $b_n = -\frac{T^3}{\pi n} + \frac{3T^3}{2\pi^3 n^3}$ for which Theorem 2.11 gives

$$y_n = \frac{1}{2}\left(\frac{3T^3}{2\pi^2 n^2} + i\left(\frac{T^3}{\pi n} - \frac{3T^3}{2\pi^3 n^3}\right)\right) = \frac{3T^3}{4\pi^2 n^2} + \left(\frac{T^3}{2\pi n} - \frac{3T^3}{4\pi^3 n^3}\right)i$$

for $n > 0$, which also is seen to coincide with what we obtained. The case $n < 0$ follows similarly.

Answer Ex. 2.2. 8: If f is symmetric about 0 we have that $b_n = 0$. Theorem 2.11 then gives that $y_n = \frac{1}{2}a_n$, which is real. The same theorem gives that that $y_{-n} = \frac{1}{2}a_n = y_n$. This proves 1.
If f is antisymmetric about 0 we have that $a_n = 0$. Theorem 2.11 then gives that $y_n = -\frac{1}{2}b_n$, which is purely imaginary. The same theorem gives that that $y_{-n} = \frac{1}{2}b_n = -y_n$. This proves 2.
When $y_n = y_{-n}$ we can write

$$y_{-n}e^{2\pi i(-n)t/T} + y_n e^{2\pi int/T} = y_n(e^{2\pi int/T} + e^{-2\pi int/T}) = 2y_n\cos(2\pi nt/T)$$

This is clearly symmetric, but then also $\sum_{n=-N}^{N} y_n e^{2\pi int/T}$ is symmetric since it is a sum of symmetric functions. This proves 3.
When $y_n = -y_{-n}$ we can write

$$y_{-n}e^{2\pi i(-n)t/T} + y_n e^{2\pi int/T} = y_n(-e^{2\pi int/T} + e^{2\pi int/T}) = 2iy_n\sin(2\pi nt/T)$$

This is clearly antisymmetric, but then also $\sum_{n=-N}^{N} y_n e^{2\pi int/T}$ is antisymmetric since it is a sum of antisymmetric functions, and since $y_0 = 0$. This proves 4.

Answer Ex. 2.4. 1: We obtain that

$$\begin{aligned}
y_n &= \frac{1}{T}\int_{-T/4}^{T/4} e^{-2\pi int/T}dt - \frac{1}{T}\int_{-T/2}^{-T/4} e^{-2\pi int/T}dt - \frac{1}{T}\int_{T/4}^{T/2} e^{-2\pi int/T}dt \\
&= -\left[\frac{1}{2\pi in}e^{-2\pi int/T}\right]_{-T/4}^{T/4} + \left[\frac{1}{2\pi in}e^{-2\pi int/T}\right]_{-T/2}^{-T/4} + \left[\frac{1}{2\pi in}e^{-2\pi int/T}\right]_{T/4}^{T/2} \\
&= \frac{1}{2\pi in}\left(-e^{-\pi in/2} + e^{\pi in/2} + e^{\pi in/2} - e^{\pi in} + e^{-\pi in} - e^{-\pi in/2}\right) \\
&= \frac{1}{\pi n}\left(2\sin(\pi n/2) - \sin(\pi n)\right) = \frac{2}{\pi n}\sin(\pi n/2).
\end{aligned}$$

The square wave defined in this exercise can be obtained by delaying our original square wave with $-T/4$. Using 3. in Theorem 2.18 with $d = -T/4$ on the complex Fourier coefficients $y_n = \begin{cases} 0, & \text{if } n \text{ is even;} \\ 2/(\pi in), & \text{if } n \text{ is odd.} \end{cases}$ which we obtained for the square wave in Exercise 2.2.5, we obtain the Fourier coefficients

$$\begin{aligned}
e^{2\pi in(T/4)/T}\begin{cases} 0, & \text{if } n \text{ is even;} \\ 2/(\pi in), & \text{if } n \text{ is odd.} \end{cases} &= \begin{cases} 0, & \text{if } n \text{ is even;} \\ \frac{2i\sin(\pi n/2)}{\pi in}, & \text{if } n \text{ is odd.} \end{cases} \\
&= \begin{cases} 0, & \text{if } n \text{ is even;} \\ \frac{2}{\pi n}\sin(\pi n/2), & \text{if } n \text{ is odd.} \end{cases}.
\end{aligned}$$

This verifies the result.

Answer Ex. 2.4. 2: Since the real Fourier series of the square wave is

$$\sum_{n\geq 1, n \text{ odd}} \frac{4}{\pi n}\sin(2\pi nt/T),$$

Theorem 2.11 gives us that the complex Fourier coefficients are $y_n = -\frac{1}{2}i\frac{4}{\pi n} = -\frac{2i}{\pi n}$, and $y_{-n} = \frac{1}{2}i\frac{4}{\pi n} = \frac{2i}{\pi n}$ for $n > 0$. This means that $y_n = -\frac{2i}{\pi n}$ for all n, so that the complex Fourier series of the square wave is

$$-\sum_{n \text{ odd}} \frac{2i}{\pi n}e^{2\pi int/T}.$$

Using property 4 in Theorem 2.18 we get that the $e^{-2\pi i4t/T}$ (i.e. set $d = -4$) times the square wave has its n'th Fourier coefficient equal to $-\frac{2i}{\pi(n+4)}$. Using linearity, this means that $2ie^{-2\pi i4t/T}$ times the square wave has its n'th Fourier coefficient equal to $\frac{4}{\pi(n+4)}$. We thus have that the function

$$f(t) = \begin{cases} 2ie^{-2\pi i4t/T} & , 0 \leq t < T/2 \\ -2ie^{-2\pi i4t/T} & , T/2 \leq t < T \end{cases}$$

has the desired Fourier series.

Answer Ex. 3.2. 1: As in Example 3.9 we get

$$F_4 \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

$$= \frac{1}{2} \begin{pmatrix} 2+3+4+5 \\ 2-3i-4+5i \\ 2-3+4-5 \\ 2+3i-4-5i \end{pmatrix} = \begin{pmatrix} 7 \\ -1+i \\ -1 \\ -1-i \end{pmatrix}.$$

Answer Ex. 3.2. 2: For $N = 6$ the entries are on the form $\frac{1}{\sqrt{6}} e^{2\pi ink/6} = \frac{1}{\sqrt{6}} e^{\pi ink/3}$. This means that the entries in the Fourier matrix are the numbers $\frac{1}{\sqrt{6}} e^{\pi i/3} = \frac{1}{\sqrt{6}}(1/2 + i\sqrt{3}/2)$, $\frac{1}{\sqrt{6}} e^{2\pi i/3} = \frac{1}{\sqrt{6}}(-1/2 + i\sqrt{3}/2)$, and so on. The matrix is thus

$$F_6 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1/2+i\sqrt{3}/2 & -1/2+i\sqrt{3}/2 & -1 & -1/2-i\sqrt{3}/2 & 1/2-i\sqrt{2}/2 \\ 1 & -1/2+i\sqrt{3}/2 & -1/2-i\sqrt{3}/2 & 1 & -1/2+i\sqrt{3}/2 & -1/2-i\sqrt{3}/2 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1/2-i\sqrt{3}/2 & -1/2+i\sqrt{3}/2 & 1 & -1/2-i\sqrt{3}/2 & -1/2+i\sqrt{3}/2 \\ 1 & 1/2-i\sqrt{2}/2 & -1/2-i\sqrt{3}/2 & -1 & -1/2+i\sqrt{3}/2 & 1/2+i\sqrt{3}/2 \end{pmatrix}$$

The cases $N = 8$ and $N = 12$ follow similarly, but are even more tedious. For $N = 8$ the entries are $\frac{1}{\sqrt{8}} e^{\pi ink/4}$, which can be expressed exactly since we can express exactly any sines and cosines of a multiple of $\pi/4$. For $N = 12$ we get the base angle $\pi/6$, for which we also have exact values for sines and cosines for all multiples.

Answer Ex. 3.2. 3: We get

$$y_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} c^k e^{-2\pi ink/N} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} (ce^{-2\pi in/N})^k$$

$$= \frac{1}{\sqrt{N}} \frac{1-(ce^{-2\pi in/N})^N}{1-ce^{-2\pi in/N}} = \frac{1}{\sqrt{N}} \frac{1-c^N}{1-ce^{-2\pi in/N}}.$$

Answer Ex. 3.2. 5: The code can look like this

```
function x=IDFTImpl(y)
  N=length(y);
  FN=zeros(N);
```

```
for k=1:N
  FN(k,:)=exp(2*pi*1i*(k-1)*(0:(N-1))/N)/sqrt(N);
end
x=FN*y;
```

Answer Ex. 3.3. 1: We have that $\lambda_S(\omega) = \frac{1}{2}(1 + \cos\omega)$. This clearly has the maximum point $(0, 1)$, and the minimum point $(\pi, 0)$.

Answer Ex. 3.3. 2: We have that $|\lambda_T(\omega)| = \frac{1}{2}(1 - \cos\omega)$. This clearly has the maximum point $(\pi, 1)$, and the minimum point $(0, 0)$. The connection between the frequency responses is that $\lambda_T(\omega) = \lambda_S(\omega + \pi)$.

Answer Ex. 3.3. 3: The sum of two digital filters is again a digital filter, and the first column in the sum can be obtained by summing the first columns in the two matrices. This means that the filter coefficients in $\frac{1}{2}(S_1 + S_2)$ can be obtained by summing the filter coefficients of S_1 and S_2, and we obtain

$$\frac{1}{2}\left(\{\underline{1}, 0, \ldots, 0, c\} + \{\underline{1}, 0, \ldots, 0, -c\}\right) = \{\underline{1}\}.$$

This means that $\frac{1}{2}(S_1 + S_2) = I$, since I is the unique filter with $\boldsymbol{e}_0$ as first column. The interpretation in terms of echos is that the echo from S_2 cancels that from S_1.

Answer Ex. 3.3. 4: The matrix for time reversal is the matrix

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

This is not a circulant Toeplitz matrix, since all diagonals assume the values 0 and 1, so that they are not constant on each diagonal. Time reversal is thus not a digital filter.
Let S denote time reversal. Clearly $S\boldsymbol{e}_1 = \boldsymbol{e}_{N-2}$. If S was time-invariant we would have that $S\boldsymbol{e}_0 = \boldsymbol{e}_{N-3}$, where we have delayed the input and output. But this clearly is not the case, since by definition $S\boldsymbol{e}_0 = \boldsymbol{e}_{N-1}$.

Answer Ex. 3.3. 5: The matrix for the operation which keeps every second component is

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix},$$

where 0 and 1 are repeated in alternating order along the main diagonal. Since the matrix is not constant on the main diagonal, it is not a circulant Toeplitz matrix, and hence not a filter.

Answer Ex. 3.3. 13: . The eigenvalues of S are $1, 5, 9$, and are found by computing a DFT of the first column (and multiplying by $\sqrt{N} = 2$). The eigenvectors are the Fourier basis vectors. 1 has multiplicity 2. Matlab uses some numeric algorithm to find the eigenvectors. However, eigenvectors may not be unique, so you have no control on which eigenvectors Matlab actually selects. In particular, here the eigenspace for $\lambda = 1$ has dimension 2, so that any linear combination of the two eigenvectors from this eigenspace also is an eigenvector. Here it seems that Matlab has chosen a linear combination which is different from a Fourier basis vector.

Answer Ex. 3.3. 14: Her we have that $s_0 = t_0 = 3$, $s_1 = t_1 = 4$, $s_2 = t_2 = 5$, and $s_3 = t_3 = 6$ (first formula), and $s_{N-2} = t_{-2} = 1$, $s_{N-1} = t_{-1} = 2$ (second formula). This means that the matrix of S is

$$S = \frac{1}{4}\begin{pmatrix} 3 & 2 & 1 & 0 & 0 & 6 & 5 & 4 \\ 4 & 3 & 2 & 1 & 0 & 0 & 6 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & 0 & 6 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 0 \\ 0 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & 0 & 6 & 5 & 4 & 3 & 2 \\ 2 & 1 & 0 & 0 & 6 & 5 & 4 & 3 \end{pmatrix}$$

The frequency response is

$$\lambda_S(\omega) = e^{2i\omega} + 2e^{i\omega} + 3 + 4e^{-i\omega} + 5e^{-2i\omega} + 6e^{-3i\omega}.$$

Answer Ex. 3.3. 15: Here we have that $t_{-1} = 1/4$, $t_0 = 1/4$, $t_1 = 1/4$, and $t_2 = 1/4$. We now get that $s_0 = t_0 = 1/4$, $s_1 = t_1 = 1/4$, and $s_2 = t_2 = 1/4$ (first formula), and $s_{N-1} = s_7 = t_{-1} = 1/4$ (second formula). This means that the matrix of S is

$$S = \frac{1}{4}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The frequency response is

$$\lambda_S(\omega) = \frac{1}{4}(e^{i\omega} + 1 + e^{-i\omega} + e^{-2i\omega}) = \frac{e^{i\omega}(1 - e^{-4i\omega})}{4(1 - e^{-i\omega})} = e^{-i\omega/2}\frac{\sin(2\omega)}{\sin(\omega/2)}$$

Answer Ex. 3.3. 16: The filter coefficients are $t_0 = s_0 = 1$, $t_1 = s_1 = 1$ (first formula), and $t_{-1} = s_{N-1} = 1$, $t_{-2} = s_{N-2} = 1$, $t_{-3} = s_{N-3} = 1$ (second formula). All other t_k are zero. This means that the filter can be written as $\{1, 1, 1, \underline{1}, 1\}$, using our compact notation.

Answer Ex. 3.3. 17: The frequency response is

$$\sum_{s=0}^{k} c^s e^{-is\omega} = \frac{1 - c^{k+1}e^{-i(k+1)\omega}}{1 - ce^{-i\omega}} = .$$

It is straightforward to compute the limit as $\omega \to 0$ as $c^k(k+1)$. This means that as we increase k or c, this limit also increases. The value of k also dictates oscillations in the frequency response, since the numerator oscillates fastest. When $c = 1$, k dictates how often the frequency response hits 0.

Answer Ex. 3.3. 18: If we write $S_1 = F_N^H D_1 F_N$ and $S_2 = F_N^H D_2 F_N$ we get

$$S_1 + S_2 = F_N^H(D_1 + D_2)F_N \qquad S_1S_2 = F_N^H D_1 F_N F_N^H D_2 F_N = F_N^H D_1 D_2 F_N$$

This means that the eigenvalues of $S_1 + S_2$ are the sum of the eigenvalues of S_1 and S_2, and the eigenvalues of S_1S_2 are the product of the eigenvalues of S_1 and S_2. The actual eigenvalues which are added and multiplied are dictated by the index of the frequency response, i.e. $\lambda_{S_1S_2,n} = \lambda_{S_1,n}\lambda_{S_2,n}$, and $\lambda_{S_1+S_2,n} = \lambda_{S_1,n} + \lambda_{S_2,n}$. In general there is no reason to believe that there is a formula for the eigenvalues for the sum or product of two matrices, based on eigenvalues of the individual matrices. However, when the same argument as for filters holds in all cases where the eigenvectors are equal.

Answer Ex. 3.4. 2: We first see that $d_{0,3} = \sqrt{\frac{1}{3}}$ and $d_{k,3} = \sqrt{\frac{2}{3}}$ for $k = 1, 2$. We also have that

$$\cos\left(2\pi\frac{n}{2N}\left(k+\frac{1}{2}\right)\right) = \cos\left(\pi\frac{n}{3}\left(k+\frac{1}{2}\right)\right),$$

so that the DCT matrix can be written as

$$\begin{aligned}
D_3 &= \begin{pmatrix} \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} \\ \sqrt{\frac{2}{3}}\cos\left(\frac{\pi}{3}\frac{1}{2}\right) & \sqrt{\frac{2}{3}}\cos\left(\frac{\pi}{3}\frac{3}{2}\right) & \sqrt{\frac{2}{3}}\cos\left(\frac{\pi}{3}\frac{5}{2}\right) \\ \sqrt{\frac{2}{3}}\cos\left(\frac{2\pi}{3}\frac{1}{2}\right) & \sqrt{\frac{2}{3}}\cos\left(\frac{2\pi}{3}\frac{3}{2}\right) & \sqrt{\frac{2}{3}}\cos\left(\frac{2\pi}{3}\frac{5}{2}\right) \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} \\ \sqrt{\frac{2}{3}}\cos(\pi/6) & \sqrt{\frac{2}{3}}\cos(\pi/2) & \sqrt{\frac{2}{3}}\cos(5\pi/6) \\ \sqrt{\frac{2}{3}}\cos(\pi/3) & \sqrt{\frac{2}{3}}\cos(\pi) & \sqrt{\frac{2}{3}}\cos(5\pi/3) \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} \\ \sqrt{\frac{2}{3}}(\sqrt{3}/2 + i/2) & 0 & \sqrt{\frac{2}{3}}(-\sqrt{3}/2 + i/2) \\ \sqrt{\frac{2}{3}}(1/2 + \sqrt{3}i/2) & -\sqrt{\frac{2}{3}} & \sqrt{\frac{2}{3}}(1/2 - \sqrt{3}i/2) \end{pmatrix}
\end{aligned}$$

Answer Ex. 3.4. 4: The code can look like this:

```
function y=filterT(t,x)
  N=length(x);
  y=zeros(length(x),1);
  E=length(t)-1;

  n=0;
  while n<E
      y(n+1)= t(1)*x(n+1);
      for k=1:n
          y(n+1) = y(n+1) + t(k+1)*(x(n+k+1)+x(n-k+1));
      end
      for k=(n+1):E
          y(n+1) = y(n+1) + t(k+1)*(x(n+k+1)+x(n-k+N+1));
      end
      n=n+1;
  end
  while n<(N-E)
      y(n+1)= t(1)*x(n+1);
      for k=1:E
          y(n+1) = y(n+1)+ t(k+1)*(x(n+k+1)+x(n-k+1));
```

```
    end
    n=n+1;
  end
  while n<N
    y(n+1) = t(1)*x(n+1);
    for k=1:(N-1-n)
        y(n+1) = y(n+1) + t(k+1)*(x(n+k+1)+x(n-k+1));
    end
    for k=(N-1-n+1):E
        y(n+1) = y(n+1) + t(k+1)*(x(n+k-N+1)+x(n-k+1));
    end
    n=n+1;
  end
```

Answer Ex. 4.1. 1: By inserting $N = 2^r$ and $x_r = M^{2^r}$ in $M_N = 2M_{N/2}+2N$ we get first $x_r = 2x_{r-1}+2\cdot 2^r$. Inserting $r+1$ for r we get $x_{r+1}-2x_r = 4\cdot 2^r$. The homogeneous equation $x_{r+1} - 2x_r = 0$ has the general solution $(x_h)_r = C2^r$. For a particular solution to the equation $x_{r+1} - 2x_r = 4 \cdot 2^r$, we should try $(x_p)_r = Ar2^r$ (since 2 is a root in the homogeneous equation), and we get that $A = 2$, so that $(x_p)_r = 2r2^r$, and the general solution to the difference equation is $x_r = 2r2^r + C2^r$. This means that

$$M_N = M_{2^r} = 2r2^r + C2^r = 2N \log_2 N + CN = O(2N \log_2 N),$$

since the first terms dominates in this expression, in particular, it does not matter what C is (although we can find C from $x_0 = 0$, since a DFT for $N = 1$ requires no multiplications).

Answer Ex. 4.1. 2: When we compute $e^{-2\pi in/N}$, we do some multiplications in the exponent. These are not counted because the multilication do not depend on $\boldsymbol{x}$, and may therefore be precomputed. We also have a multiplication with $\frac{1}{\sqrt{2}}$. These are typically not counted because one often defines a DFT so that this multiplication is absorbed in the definition.

Answer Ex. 4.1. 4: From the formula we see that the first third of the Fourier cofficients can be written

$$y_n = \frac{1}{\sqrt{3}} \left(F_{N/3}\boldsymbol{x}_1 + D_{N/3}F_{N/3}\boldsymbol{x}_2 + D_{N/3}^2 F_{N/3}\boldsymbol{x}_3 \right).$$

where $D_{N/3}$ is defined in the same way as $D_{N/2}$, but as a $(N/3)\times(N/3)$-matrix, and where $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ denotes the splitting of $\boldsymbol{x}$ into vectors for the corresponding indices. The second third of the Fourier cofficients can be written

$$\begin{aligned}
y_{N/3+n} &= \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} x_k e^{-2\pi i(N/3+n)k/N} \\
&= \frac{1}{\sqrt{N}}\sum_{k=0}^{N/3-1} x_{3k} e^{-2\pi i(N/3+n)3k/N} + \frac{1}{\sqrt{N}}\sum_{k=0}^{N/3-1} x_{3k+1} e^{-2\pi i(N/3+n)(3k+1)/N} \\
&+ \frac{1}{\sqrt{N}}\sum_{k=0}^{N/3-1} x_{3k+2} e^{-2\pi i(N/3+n)(3k+2)/N} \\
&= \frac{1}{\sqrt{N}}\sum_{k=0}^{N/3-1} x_{3k} e^{-2\pi i n 3k/N} + \frac{1}{\sqrt{N}} e^{-2\pi i(N/3+n)/N}\sum_{k=0}^{N/3-1} x_{3k+1} e^{-2\pi i n 3k/N} \\
&+ \frac{1}{\sqrt{N}} e^{-2\pi i(N/3+n)2/N}\sum_{k=0}^{N/3-1} x_{3k+2} e^{-2\pi i n 3k/N} \\
&= \frac{1}{\sqrt{3}}\left(F_{N/3}\boldsymbol{x}_1 + e^{-2\pi i/3} D_{N/3}F_{N/3}\boldsymbol{x}_2 + e^{-2\pi i 2/3} D_{N/3}^2 F_{N/3}\boldsymbol{x}_3\right).
\end{aligned}$$

The third part of the Fourier coefficients can be written

$$\begin{aligned}
y_{2N/3+n} &= \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} x_k e^{-2\pi i(2N/3+n)k/N} \\
&= \frac{1}{\sqrt{N}}\sum_{k=0}^{N/3-1} x_{3k} e^{-2\pi i(2N/3+n)3k/N} + \frac{1}{\sqrt{N}}\sum_{k=0}^{N/3-1} x_{3k+1} e^{-2\pi i(2N/3+n)(3k+1)/N} \\
&+ \frac{1}{\sqrt{N}}\sum_{k=0}^{N/3-1} x_{3k+2} e^{-2\pi i(2N/3+n)(3k+2)/N} \\
&= \frac{1}{\sqrt{N}}\sum_{k=0}^{N/3-1} x_{3k} e^{-2\pi i n 3k/N} + \frac{1}{\sqrt{N}} e^{-2\pi i(2N/3+n)/N}\sum_{k=0}^{N/3-1} x_{3k+1} e^{-2\pi i n 3k/N} \\
&+ \frac{1}{\sqrt{N}} e^{-2\pi i(2N/3+n)2/N}\sum_{k=0}^{N/3-1} x_{3k+2} e^{-2\pi i n 3k/N} \\
&= \frac{1}{\sqrt{3}}\left(F_{N/3}\boldsymbol{x}_1 + e^{-2\pi i 2/3} D_{N/3}F_{N/3}\boldsymbol{x}_2 + e^{-2\pi i 4/3} D_{N/3}^2 F_{N/3}\boldsymbol{x}_3\right) \\
&= \frac{1}{\sqrt{3}}\left(F_{N/3}\boldsymbol{x}_1 + e^{-2\pi i 2/3} D_{N/3}F_{N/3}\boldsymbol{x}_2 + D_{N/3}^2 F_{N/3}\boldsymbol{x}_3\right)
\end{aligned}$$

We get a similar factorization as in Theorem 4.4, but with the block matrix replaced by

$$\frac{1}{\sqrt{3}}\left(\begin{array}{c|c|c}
F_{N/3} & D_{N/3}F_{N/3} & D_{N/3}^2 F_{N/3} \\ \hline
F_{N/3} & e^{-2\pi i/3} D_{N/3}F_{N/3} & e^{-2\pi i 2/3} D_{N/3}^2 F_{N/3} \\ \hline
F_{N/3} & e^{-2\pi i 2/3} D_{N/3}F_{N/3} & D_{N/3}^2 F_{N/3}
\end{array}\right).$$

We see that $M_N = 3M_{N/3} + 2N$ when we count complex multiplications, so that $M_N = 3M_{N/3} + 8N$ when we count real multiplications. We get a difference equation of the form $x_{r+1} = 3x_r + 24 \cdot 3^r$. A particular solution to this is $(x_p)_r = 8r3^r$. Solving as above we get $M_N = O(8N \log_3 N)$. $\log_3 N$ can be written on the form $c \log_2 N$ for a constant c, this is on the form $O(c \log_2 N)$ for some c.
It is clear that this procedure can be developed also for numbers divisible by 5, 7, and so on (the number of blocks in the block matrix increase, though). In particular, we can develop a procedure for any factorization into prime numbers.

Answer Ex. 5.2. 1: We have that $f(t) = \sum_{n=0}^{N-1} c_n \phi_{0,n}$, where c_n are the coordinates of f in the basis $\{\phi_{0,0}, \phi_{0,1}, \ldots, \phi_{0,N-1}\}$. We now get that

$$f(k) = \sum_{n=0}^{N-1} c_n \phi_{0,n}(k) = c_k,$$

since $\phi_{0,n}(k) = 0$ when $n \neq k$. This shows that $(f(0), f(1), \ldots . f(N-1))$ are the coordinates of f.

Answer Ex. 5.2. 2: We have that

$$\mathrm{proj}_{V_0}(f) = \sum_{n=0}^{N-1} \left(\int_0^N f(t)\phi_{0,n}(t)dt \right) \phi_{0,n} = \sum_{n=0}^{N-1} \left(\int_n^{n+1} f(t)dt \right) \phi_{0,n},$$

where we have used the orthogonal decomposition formula. Note also that, if $f(t) \in V_1$, and $f_{n,1}$ is the value f attains on $[n, n+1/2)$, and $f_{n,2}$ is the value f attains on $[n+1/2, n+1)$, we have that

$$\begin{aligned}
\mathrm{proj}_{V_0}(f) &= \sum_{n=0}^{N-1} \left(\int_n^{n+1} f(t)dt \right) \phi_{0,n}(t) \\
&= \sum_{n=0}^{N-1} \left(\frac{1}{2} f_{n,1} + \frac{1}{2} f_{n,2} \right) \phi_{0,n}(t) = \sum_{n=0}^{N-1} \frac{f_{n,1} + f_{n,2}}{2} \phi_{0,n}(t),
\end{aligned}$$

which is the function which is $(f_{n,1} + f_{n,2})/2$ on $[n, n+1)$. This proves the first part of Proposition 5.13.

Answer Ex. 5.2. 3: We have that

$$\begin{aligned}
\|f - \mathrm{proj}_{V_0}(f)\|^2 &= \langle f - \mathrm{proj}_{V_0}(f), f - \mathrm{proj}_{V_0}(f) \rangle \\
&= \langle f, f \rangle - 2\langle f, \mathrm{proj}_{V_0}(f) \rangle + \langle \mathrm{proj}_{V_0}(f), \mathrm{proj}_{V_0}(f) \rangle
\end{aligned}$$

Now, note that

$$\langle \mathrm{proj}_{V_0}(f), \mathrm{proj}_{V_0}(f) \rangle = \sum_{n=0}^{N-1} \left(\int_n^{n+1} f(t)dt \right)^2$$

from what we just showed in Exercise 2 (use that the $\phi_{0,n}$ are orthonormal). This means that the above can be written

$$
\begin{aligned}
&= \langle f, f\rangle - 2\sum_{n=0}^{N-1}\int_0^N \left(\int_n^{n+1} f(s)ds\right)\phi_{0,n}(t)f(t)dt + \sum_{n=0}^{N-1}\left(\int_n^{n+1} f(t)dt\right)^2 \\
&= \langle f, f\rangle - 2\sum_{n=0}^{N-1}\int_n^{n+1} \left(\int_n^{n+1} f(s)ds\right)f(t)dt + \sum_{n=0}^{N-1}\left(\int_n^{n+1} f(t)dt\right)^2 \\
&= \langle f, f\rangle - 2\sum_{n=0}^{N-1}\left(\int_n^{n+1} f(t)dt\right)^2 + \sum_{n=0}^{N-1}\left(\int_n^{n+1} f(t)dt\right)^2 \\
&= \langle f, f\rangle - \sum_{n=0}^{N-1}\left(\int_n^{n+1} f(t)dt\right)^2 .
\end{aligned}
$$

Answer Ex. 5.2. 4: Since $\phi \in V_0$ we must have that $T(\phi) = \phi$. Since ψ is in the orthogonal complement of V_0 in V_1 we must have that $T(\psi) = 0$. The columns in the matrix of T are $[T(\phi_{0,0})]_{\mathcal{C}_1}$, $[T(\psi_{0,0})]_{\mathcal{C}_1}$, $[T(\phi_{0,1})]_{\mathcal{C}_1}$, $[T(\psi_{0,1})]_{\mathcal{C}_1}$, and so on, which is $[\phi_{0,0}]_{\mathcal{C}_1}$, $[0]_{\mathcal{C}_1}$, $[\phi_{0,1}]_{\mathcal{C}_1}$, $[0]_{\mathcal{C}_1}$, and so on, which is $\boldsymbol{e}_0$, $\mathbf{0}$, $\boldsymbol{e}_2$, $\mathbf{0}$, and so on. It follows that the matrix of T relative to $\mathcal{C}_1$ is given by the diagonal matrix where 1 and 0 are repeated alternatingly on the diagonal, N times (i.e. 1 at the even indices, 0 at the odd indices). (c) follows in the same way.

Answer Ex. 5.2. 5: From lemma 5.9 it follows that

$$
\begin{aligned}
\text{proj}_{V_0}(\phi_{1,2n}) &= \phi_{0,n}/\sqrt{2} \\
\text{proj}_{V_0}(\phi_{1,2n+1}) &= \phi_{0,n}/\sqrt{2}
\end{aligned}
$$

This means that

$$
\begin{aligned}
[\text{proj}_{V_0}(\phi_{1,2n})]_{\boldsymbol{\phi}_0} &= \boldsymbol{e}_n/\sqrt{2} \\
[\text{proj}_{V_0}(\phi_{1,2n+1})]_{\boldsymbol{\phi}_0} &= \boldsymbol{e}_n/\sqrt{2}.
\end{aligned}
$$

These are the columns in the matrix for proj_{V_0} relative to the bases $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_0$. This matrix is thus

$$
\frac{1}{\sqrt{2}}\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1
\end{pmatrix}.
$$

Similarly, from lemma 5.12 it follows that

$$
\begin{aligned}
\text{proj}_{W_0}(\phi_{1,2n}) &= \psi_{0,n}/\sqrt{2} \\
\text{proj}_{W_0}(\phi_{1,2n+1}) &= -\psi_{0,n}/\sqrt{2}
\end{aligned}
$$

This means that

$$[\mathrm{proj}_{W_0}(\phi_{1,2n})]_{\boldsymbol{\psi}_0} = \boldsymbol{e}_n/\sqrt{2}$$
$$[\mathrm{proj}_{W_0}(\phi_{1,2n+1})]_{\boldsymbol{\psi}_0} = -\boldsymbol{e}_n/\sqrt{2}.$$

These are the columns in the matrix for proj_{W_0} relative to the bases $\boldsymbol{\phi}_1$ and $\boldsymbol{\psi}_0$. This matrix is thus

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{pmatrix}.$$

Answer Ex. 5.2. 6: The orthogonal decomposition theorem gives that

$$\begin{aligned}\mathrm{proj}_{W_0}(f) &= \sum_{n=0}^{N-1}\langle f,\psi_{0,n}\rangle\psi_{0,n}(t) = \sum_{n=0}^{N-1}\left(\int_0^N f(t)\psi_{0,n}(t)dt\right)\psi_{0,n}(t)\\ &= \sum_{n=0}^{N-1}\left(\int_n^{n+1} f(t)\psi_{0,n}(t)dt\right)\psi_{0,n}(t)\\ &= \sum_{n=0}^{N-1}\left(\int_n^{n+1/2} f(t)dt - \int_{n+1/2}^{n+1} f(t)dt\right)\psi_{0,n}(t),\end{aligned}$$

where we used that $\psi_{0,n}$ is nonzero only on $[n, n+1)$, and is 1 on $[n, n+1/2)$, and -1 on $[n+1/2, n+1)$. Note also that, if $f(t) \in V_1$, and $f_{n,1}$ is the value f attains on $[n, n+1/2)$, and $f_{n,2}$ is the value f attains on $[n+1/2, n+1)$, we have that

$$\begin{aligned}\mathrm{proj}_{W_0}(f) &= \sum_{n=0}^{N-1}\left(\int_n^{n+1/2} f(t)dt - \int_{n+1/2}^{n+1} f(t)dt\right)\psi_{0,n}(t)\\ &= \sum_{n=0}^{N-1}\left(\frac{1}{2}f_{n,1} - \frac{1}{2}f_{n,2}\right)\psi_{0,n}(t) = \sum_{n=0}^{N-1}\frac{f_{n,1}-f_{n,2}}{2}\psi_{0,n}(t),\end{aligned}$$

which is the function which is $(f_{n,1}-f_{n,2})/2$ on $[n, n+1/2)$, and $-(f_{n,1}-f_{n,2})/2$ on $[n+1/2, n+1)$. This proves the second part of Proposition 5.13.

Answer Ex. 5.3. 1: Since $\phi_{m,n} \in V_m$ we must have that $T(\phi_{m,n}) = \phi_{m,n}$. Since $\psi_{m,n}$ is in the orthogonal complement of V_m in V_{m+1} we must have that $T(\psi_{m,n}) = 0$. The columns in the matrix of T are $[T(\phi_{m,0})]_{\mathcal{C}_{m+1}}$, $[T(\psi_{m,0})]_{\mathcal{C}_{m+1}}$, $[T(\phi_{m,1})]_{\mathcal{C}_{m+1}}$, $[T(\psi_{m,1})]_{\mathcal{C}_{m+1}}$, and so on, which is $[\phi_{m,0}]_{\mathcal{C}_{m+1}}$, $[0]_{\mathcal{C}_{m+1}}$, $[\phi_{m,1}]_{\mathcal{C}_{m+1}}$, $[0]_{\mathcal{C}_{m+1}}$, and so on, which is $\boldsymbol{e}_0$, $\boldsymbol{0}$, $\boldsymbol{e}_2$, $\boldsymbol{0}$, and so on. It follows that the matrix of

T relative to $\mathcal{C}_{m+1}$ is given by the diagonal matrix where 1 and 0 are repeated alternatingly on the diagonal, $2^m N$ times (i.e. 1 at the even indices, 0 at the odd indices). (c) follows in the same way.

Answer Ex. 5.3. 2: If $f \in V_m$ we can write $f(t) = \sum_{n=0}^{2^m N-1} c_{m,n}\phi_{m,n}(t)$. We now get

$$\begin{aligned}
g(t) = f(2t) &= \sum_{n=0}^{2^m N-1} c_{m,n}\phi_{m,n}(2t) = \sum_{n=0}^{2^m N-1} c_{m,n}2^{m/2}\phi(2^m 2t - n) \\
&= \sum_{n=0}^{2^m N-1} c_{m,n}2^{-1/2}2^{(m+1)/2}\phi(2^{m+1}t - n) = \sum_{n=0}^{2^m N-1} c_{m,n}2^{-1/2}\phi_{m+1,n}(t).
\end{aligned}$$

This shows that $g \in V_{m+1}$. To prove the other way, assume that $g(t) = f(2t) \in V_{m+1}$. This means that we can write $g(t) = \sum_{n=0}^{2^{m+1}N-1} c_{m+1,n}\phi_{m+1,n}(t)$. We now have

$$\begin{aligned}
f(t) = g(t/2) &= \sum_{n=0}^{2^{m+1}N-1} c_{m+1,n}\phi_{m+1,n}(t/2) = \sum_{n=0}^{2^{m+1}N-1} c_{m+1,n}2^{(m+1)/2}\phi(2^m t - n) \\
&= \sum_{n=0}^{2^m N-1} c_{m+1,n}2^{(m+1)/2}\phi(2^m t - n) + \sum_{n=2^m N}^{2^{m+1}N-1} c_{m+1,n}2^{(m+1)/2}\phi(2^m t - n) \\
&= \sum_{n=0}^{2^m N-1} c_{m+1,n}2^{(m+1)/2}\phi(2^m t - n) + \sum_{n=0}^{2^m N-1} c_{m+1,n+2^m N}2^{(m+1)/2}\phi(2^m t - n - 2^m N) \\
&= \sum_{n=0}^{2^m N-1} c_{m+1,n}2^{(m+1)/2}\phi(2^m t - n) + \sum_{n=0}^{2^m N-1} c_{m+1,n+2^m N}2^{(m+1)/2}\phi(2^m t - n) \\
&= \sum_{n=0}^{2^m N-1} (c_{m+1,n} + c_{m+1,n+2^m N})2^{1/2}2^{m/2}\phi(2^m t - n) \\
&= \sum_{n=0}^{2^m N-1} (c_{m+1,n} + c_{m+1,n+2^m N})2^{1/2}\phi_{m,n}(t) \in V_m
\end{aligned}$$

The thing which made this a bit difficult was that the range of the n-indices here was outside $[0, 2^m N - 1]$ (which describe the legal indices in the basis V_m), so that we had to use the periodicty of ϕ.

Answer Ex. 5.3. 3: By definition, $[T_1]_{\mathcal{B}_1} \oplus [T_2]_{\mathcal{B}_2} \oplus \cdots \oplus [T_n]_{\mathcal{B}_n}$ is a block matrix where the blocks on the diagonal are the matrices $[T_1]_{\mathcal{B}_1}$, $[T_2]_{\mathcal{B}_2}$, and so on. If $\boldsymbol{b}_i$ are the basis vectors in $\mathcal{B}_i$, the columns in $[T_i]_{\mathcal{B}_i}$ are $[T(\boldsymbol{b}_j)]_{\mathcal{B}_i}$. This means that $[T_1]_{\mathcal{B}_1} \oplus [T_2]_{\mathcal{B}_2} \oplus \cdots \oplus [T_n]_{\mathcal{B}_n}$ has $[T(\boldsymbol{b}_j)]_{\mathcal{B}_i}$ in the j'th block, and $\mathbf{0}$ elsewhere. This means that we can write it as

$$\mathbf{0} \oplus \cdots \mathbf{0} \oplus [T(\boldsymbol{b}_j)]_{\mathcal{B}_i} \oplus \mathbf{0} \cdots \mathbf{0}.$$

On the other hand, $[T_1 \oplus T_2 \oplus \ldots \oplus T_n]_{\mathcal{B}_1 \oplus \mathcal{B}_2 \oplus \ldots \oplus \mathcal{B}_n}$ is a matrix of the same size, and the corresponding column to that of the above is

$$\begin{aligned}
&[(T_1 \oplus T_2 \oplus \ldots \oplus T_n)(\mathbf{0} \oplus \cdots \mathbf{0} \oplus \boldsymbol{b}_j \oplus \mathbf{0} \cdots \mathbf{0})]_{\mathcal{B}_1 \oplus \mathcal{B}_2 \oplus \ldots \oplus \mathcal{B}_n} \\
&= [\mathbf{0} \oplus \cdots \mathbf{0} \oplus T(\boldsymbol{b}_j) \oplus \mathbf{0} \cdots \mathbf{0}]_{\mathcal{B}_1 \oplus \mathcal{B}_2 \oplus \ldots \oplus \mathcal{B}_n} \\
&= \mathbf{0} \oplus \cdots \mathbf{0} \oplus [T(\boldsymbol{b}_j)]_{\mathcal{B}_i} \oplus \mathbf{0} \cdots \mathbf{0}.
\end{aligned}$$

Here $\boldsymbol{b}_j$ occurs as the i'th summand. This is clearly the same as what we computed for the right hand side above.

Answer Ex. 5.3. 4: Assume that λ is an eigenvalue common to both T_1 and T_2. Then there exists a vector $\boldsymbol{v}_1$ so that $T_1\boldsymbol{v}_1 = \lambda\boldsymbol{v}_1$, and a vector $\boldsymbol{v}_2$ so that $T_2\boldsymbol{v}_2 = \lambda\boldsymbol{v}_2$. We now have that

$$\begin{aligned}
(T_1 \oplus T_2)(\boldsymbol{v}_1 \oplus \boldsymbol{v}_2) &= \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{pmatrix} \\
&= \begin{pmatrix} T_1\boldsymbol{v}_1 \\ T_2\boldsymbol{v}_2 \end{pmatrix} = \begin{pmatrix} \lambda\boldsymbol{v}_1 \\ \lambda\boldsymbol{v}_2 \end{pmatrix} \\
&= \lambda \begin{pmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{pmatrix} = \lambda(\boldsymbol{v}_1 \oplus \boldsymbol{v}_2).
\end{aligned}$$

This shows that λ is an eigenvalue for λ also, and that $\boldsymbol{v}_1 \oplus \boldsymbol{v}_2$ is a corresponding eigenvector.

Answer Ex. 5.3. 5: We have that

$$\begin{aligned}
(A \oplus B)(A^{-1} \oplus B^{-1}) &= \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} A^{-1} & 0 \\ 0 & B^{-1} \end{pmatrix} \\
&= \begin{pmatrix} AA^{-1} & 0 \\ 0 & BB^{-1} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} = I
\end{aligned}$$

where we have multiplied as block matrices. This proves that $A \oplus B$ is invertible, and states what the inverse is.

Answer Ex. 5.3. 6: We have that

$$(A \oplus B)(C \oplus D) = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} C & 0 \\ 0 & D \end{pmatrix} = \begin{pmatrix} AC & 0 \\ 0 & BD \end{pmatrix} = (AC) \oplus (BD)$$

where we again have multiplied as block matrices.

Answer Ex. 5.3. 8: The following code achieves this:

```
[S,fs]=wavread('castanets.wav');
```

```
newx=DWTHaarImpl(S(1:2^17,1),2);
plot(0:(2^17-1),newx(1:2^17,1))
axis([0 2^17 -1 1]);
```

The values from V_0 corresponds to the first $1/4$ values in the plot, the values from W_0 corresponds to the next $1/4$ values in the plot, while the values from W_1 correspond to the last $1/2$ of the values in the plot.

Answer Ex. 5.3. 9: The following code achieves the task

```
function playDWTlower(m)
  [S fs]=wavread('../castanets');
  newx=DWTHaarImpl(S(1:2^17,1),m);
  len=length(newx);
  newx((len/2^m+1):length(newx))=zeros(length(newx)-len/2^m,1);
  newx=IDWTHaarImpl(newx,m);
  playerobj=audioplayer(newx,fs);
  playblocking(playerobj);
```

For $m = 2$ we clearly hear a degradation in the sound. For $m = 4$ and above most of the sound is unrecognizable. There is no reason to believe that sound samples returned by the function lie in $[-1, 1]$. you can check this by printing the maximum value in the returned array on screen inside this method.

Answer Ex. 5.3. 11: The following code can be used

```
function playDWTlowerdifference(m)
  [S fs]=wavread('../castanets');
  newx=DWTHaarImpl(S(1:2^17,1),m);
  len=length(newx);
  newx(1:(len/2^m))=zeros(len/2^m,1);
  newx=IDWTHaarImpl(newx,m);
  playerobj=audioplayer(newx,fs);
  playblocking(playerobj);
```

Answer Ex. 5.3. 12: Note first that, similarly to the computation in Exercise 5.2.6, we have that

$$\int_0^N f(t)\psi_{m,n}(t)dt = 2^{m/2}\left(\int_{n2^{-m}}^{(n+1/2)2^{-m}} f(t)dt - \int_{(n+1/2)2^{-m}}^{(n+1)2^{-m}} f(t)dt\right).$$

With $f(t) = 1 - 2|1/2 - t/N|$ we have two possibilities: when $n < N2^{m-1}$ we have that $[n2^{-m},(n+1)2^{-m}) \subset [0,N/2]$, so that $f(t)=2t/N$, and we get

$$\begin{aligned} w_{m,n} &= 2^{m/2}\left(\int_{n2^{-m}}^{(n+1/2)2^{-m}} 2t/N dt - \int_{(n+1/2)2^{-m}}^{(n+1)2^{-m}} 2t/N dt\right) \\ &= 2^{m/2}[t^2/N]_{n2^{-m}}^{(n+1/2)2^{-m}} - 2^{m/2}[t^2/N]_{(n+1/2)2^{-m}}^{(n+1)2^{-m}} \\ &= \frac{2^{-3m/2}}{N}\left(2(n+1/2)^2-n^2-(n+1)^2\right) = -\frac{2^{-3m/2-1}}{N}. \end{aligned}$$

When $n \geq N2^{m-1}$ we have that $f(t)=2-2t/N$, and using that $\int_0^N \psi_{m,n}(t)dt = 0$ we must get that $w_{m,n} = \frac{2^{-3m/2-1}}{N}$.
For $f(t)=1/2+\cos(2\pi t/N)/2$, note first that this has the same coefficients as $\cos(2\pi t/N)/2$, since $\int_0^N \psi_{m,n}(t)dt = 0$. We now get

$$\begin{aligned} w_{m,n} &= 2^{m/2}\left(\int_{n2^{-m}}^{(n+1/2)2^{-m}} \cos(2\pi t/N)/2dt - \int_{(n+1/2)2^{-m}}^{(n+1)2^{-m}} \cos(2\pi t/N)/2dt\right) \\ &= 2^{m/2}[N\sin(2\pi t/N)/(4\pi)]_{n2^{-m}}^{(n+1/2)2^{-m}} - 2^{m/2}[N\sin(2\pi t/N)/(4\pi)]_{(n+1/2)2^{-m}}^{(n+1)2^{-m}} \\ &= \frac{2^{m/2-2}N}{\pi}\left(2\sin(2\pi(n+1/2)2^{-m}/N) - \sin(2\pi n2^{-m}/N) - \sin(2\pi(n+1)2^{-m}/N)\right). \end{aligned}$$

There seems to be no more possibilities for simplification here.

Answer Ex. 5.3. 13: We get

$$\begin{aligned} w_{m,n} &= 2^{m/2}\left(\int_{n2^{-m}}^{(n+1/2)2^{-m}} (t/N)^k dt - \int_{(n+1/2)2^{-m}}^{(n+1)2^{-m}} (t/N)^k dt\right) \\ &= 2^{m/2}[t^{k+1}/((k+1)N^k)]_{n2^{-m}}^{(n+1/2)2^{-m}} - 2^{m/2}[t^{k+1}/((k+1)N^k)]_{(n+1/2)2^{-m}}^{(n+1)2^{-m}} \\ &= \frac{2^{-m(k+1/2)}}{(k+1)N^k}\left(2(n+1/2)^{k+1} - n^{k+1} - (n+1)^{k+1}\right). \end{aligned}$$

The leading term n^{k+1} will here cancel, but the others will not, so there is no room for further simplification here.

Answer Ex. 5.4. 1: Let us write $f(t)=\sum_{n=0}^{N-1} c_n\phi_{0,n}(t)$. If k is an integer we have that

$$f(k) = \sum_{n=0}^{N-1} c_n\phi_{0,n}(k) = \sum_{n=0}^{N-1} c_n\phi(k-n).$$

Clearly the only integer for which $\phi(s) \neq 0$ is $s=0$ (since $\phi(0)=1$), so that the only n which contributes in the sum is $n=k$. This means that $f(k)=c_k$, so that $[f]_{\phi_0} = (f(0), f(1), \dots, f(N-1))$.

Answer Ex. 5.4. 2: We have that

$$\begin{aligned}\langle \phi_{0,n}, \phi_{0,n} \rangle &= \int_{n-1}^{n+1} (1-|t-n|)^2 dt \\ &= \int_{n-1}^{n+1} \left(1 - 2|t-n| + (t-n)^2\right) dt \\ &= 2 - 2 + \left[\frac{1}{3}(t-n)^3\right]_{n-1}^{n+1} = \frac{2}{3}.\end{aligned}$$

We also have

$$\begin{aligned}\langle \phi_{0,n}, \phi_{0,n+1} \rangle &= \int_{n}^{n+1} (1-(t-n))(1+(t-n-1))dt = \int_0^1 (1-u)(1+u-1)du \\ &= \int_0^1 (t-t^2)dt = \frac{1}{2} - \frac{1}{3} = \frac{1}{6}.\end{aligned}$$

Finally, the supports of $\phi_{0,n}$ and $\phi_{0,n\pm k}$ are disjoint for $k > 1$, so that we must have $\langle \phi_{0,n}, \phi_{0,n\pm k} \rangle = 0$ in that case.

Answer Ex. 5.4. 3: We have that

$$\chi_{[-1/2,1/2)} * \chi_{[-1/2,1/2)}(x) = \int_{-\infty}^{\infty} \chi_{[-1/2,1/2)}(t)\chi_{[-1/2,1/2)}(x-t)dt.$$

The integrand here is 1 when $-1/2 < t < 1/2$ and $-1/2 < x - t < 1/2$, or in other words when $\max(-1/2, -1/2 + x) < t < \min(1/2, 1/2 + x)$ (else it is 0). When $x > 0$ this happens when $-1/2 + x < t < 1/2$, and when $x < 0$ this happens when $-1/2 < t < 1/2 + x$. This means that

$$\chi_{[-1/2,1/2)} * \chi_{[-1/2,1/2)}(x) = \begin{cases} \int_{-1/2+x}^{1/2} dt = 1 - x & , x > 0 \\ \int_{-1/2}^{1/2+x} dt = 1 + x & , x < 0. \end{cases}$$

But this is by definition ϕ.

Answer Ex. 5.5. 1: a. The function $\hat{\psi}$ is a sum of the functions $\psi = \phi_{1,1}$, ϕ, and $\phi_{0,1}$ (i.e. we have set $n = 0$ in Equation (5.54)). All these are continuous and piecewise linear, and we can write

$$\phi_{1,1}(t) = \begin{cases} 2t & 0 \le t < 1/2 \\ 2-2t & 1/2 \le t < 1 \\ 0 & \text{elsewhere} \end{cases}$$

$$\phi(t)(t) = \begin{cases} 1+t & -1 \le t < 0 \\ 1-t & 0 \le t < 1 \\ 0 & \text{elsewhere} \end{cases}$$

$$\phi_{0,1}(t) = \begin{cases} t & 0 \le t < 1 \\ 2-t & 1 \le t < 2 \\ 0 & \text{elsewhere} \end{cases}.$$

It follows that $\hat{\psi}(t) = \phi_{1,1}(t) - \alpha\phi(t) - \beta\phi_{1,1}$ is piecewise linear, and linear on the segments $[-1, 0]$, $[0, 1/2]$, $[1/2, 1]$, $[1, 2]$.

On the segment $[-1, 0]$ only the function ϕ is seen to be nonzero, and since $\phi(t) = 1 + t$ here, we have that $\hat{\psi}(t) = -\alpha(1 + t) = -\alpha - \alpha t$ here.

On the segment $[0, 1/2]$ all three functions are nonzero, and

$$\begin{aligned} \phi_{1,1}(t) &= 2t \\ \phi(t)(t) &= 1 - t \\ \phi_{0,1}(t) &= t \end{aligned}$$

on this interval. This means that $\hat{\psi}(t) = 2t - \alpha(1 - t) - \beta t = (2 + \alpha - \beta)t - \alpha$ on $[0, 1/2]$.

On the segment $[0, 1/2]$ all three functions are nonzero, and

$$\begin{aligned} \phi_{1,1}(t) &= 2 - 2t \\ \phi(t)(t) &= 1 - t \\ \phi_{0,1}(t) &= t \end{aligned}$$

on this interval. This means that $\hat{\psi}(t) = 2-2t-\alpha(1-t)-\beta t = (\alpha-\beta-2)t-\alpha+2$ on $[1/2, 1]$.

On the segment $[1, 2]$ only the function $\phi_{0,1}$ is seen to be nonzero, and since $\phi_{0,1}(t) = 2-t$ here, we have that $\hat{\psi}(t) = -\beta(2-t) = \beta t - 2\beta$ here. For all other values of t, $\hat{\psi}$ is zero. This proves the formulas for $\hat{\psi}$ on the different intervals.

b. We can write

$$
\begin{aligned}
\int_0^N \hat{\psi}(t)dt &= \int_{-1}^2 \hat{\psi}(t)dt = \int_{-1}^0 \hat{\psi}(t)dt + \int_0^{1/2} \hat{\psi}(t)dt + \int_{1/2}^1 \hat{\psi}(t)dt + \int_1^2 \hat{\psi}(t)dt \\
&= \int_{-1}^0 (-\alpha - \alpha t)dt + \int_0^{1/2} (2+\alpha-\beta)t - \alpha)dt \\
&+ \int_{1/2}^1 ((\alpha-\beta-2)t - \alpha + 2)dt + \int_1^2 (\beta t - 2\beta)dt \\
&= \left[-\alpha t - \frac{1}{2}\alpha t^2\right]_{-1}^0 + \left[\frac{1}{2}(2+\alpha-\beta)t^2 - \alpha t\right]_0^{1/2} \\
&+ \left[\frac{1}{2}(\alpha-\beta-2)t^2 + (2-\alpha)t\right]_{1/2}^1 + \left[\frac{1}{2}\beta t^2 - 2\beta t\right]_1^2 \\
&= -\alpha + \frac{1}{2}\alpha + \frac{1}{8}(2+\alpha-\beta) - \frac{1}{2}\alpha + \frac{3}{8}(\alpha-\beta-2) + \frac{1}{2}(2-\alpha) + \frac{3}{2}\beta - 2\beta \\
&= \frac{1}{2} - \alpha - \beta,
\end{aligned}
$$

$\int_0^N t\hat{\psi}(t)dt$ is computed similarly, so that we in the end arrive at $\frac{1}{4} - \beta$.
c. The equation system

$$
\begin{aligned}
\frac{1}{2} - \alpha - \beta &= 0 \\
\frac{1}{4} - \beta &= 0
\end{aligned}
$$

has the unique solution $\alpha = \beta = \frac{1}{4}$, which we already have found.

Answer Ex. 5.5. 2: a. In order for ψ to have vanishing moments we must have that $\int \hat{\psi}(t)dt = \int t\hat{\psi}(t)dt = 0$ Substituting $\hat{\psi} = \psi - \alpha\phi_{0,0} - \beta\phi_{0,1}$ we see that, for $k = 0, 1$,

$$\int t^k \left(\alpha\phi_{0,0} + \beta\phi_{0,1}\right) dt = \int t^k \psi(t)dt.$$

The left hand side can here be written

$$
\begin{aligned}
\int t^k \left(\alpha\phi_{0,0} + \beta\phi_{0,1}\right) dt &= \alpha \int t^k \phi_{0,0} dt + \beta \int t^k \phi_{0,1}(t)dt \\
&= \alpha \int_{-1}^1 t^k (1 - |t|)dt + \beta \int_0^2 t^k (1 - |t-1|)dt = \alpha a_k + \beta b_k.
\end{aligned}
$$

The right hand side is

$$\int t^k \psi(t)dt = \int t^k \phi_{1,1}(t)dt = \int_0^1 (1 - 2|t - 1/2|)dt = e_k.$$

The following program sets up the corresponding equation systems, and solves it by finding α, β.

```
A=zeros(2);
b=zeros(2,1);
for k=0:1
    A(k+1,:) = [quad(@(t)t.^k.*(1-abs(t)),-1,1)...
                quad(@(t)t.^k.*(1-abs(t-1)),0,2)];
    b(k+1)=quad(@(t)t.^k.*(1-2*abs(t-1/2)),0,1);
end
A\b
```

b. Similarly to a., Equation (5.60) gives that

$$\int t^k \left(\alpha\phi_{0,0} + \beta\phi_{0,1} + \gamma\phi_{0,-1} + \delta\phi_{0,2}\right) dt = \int t^k \psi(t) dt.$$

The correspodning equation system is deduced exactly as in a. The following program sets up the corresponding equation systems, and solves it by finding $\alpha, \beta, \gamma, \delta$.

```
A=zeros(4);
b=zeros(4,1);
for k=0:3
    A(k+1,:) = [quad(@(t)t.^k.*(1-abs(t)),-1,1)...
                quad(@(t)t.^k.*(1-abs(t-1)),0,2)...
                quad(@(t)t.^k.*(1-abs(t+1)),-2,0)...
                quad(@(t)t.^k.*(1-abs(t-2)),1,3)];
    b(k+1)=quad(@(t)t.^k.*(1-2*abs(t-1/2)),0,1);
end
A\b
```

c. The function $\hat{\psi}$ now is supported on $[-2, 3]$, and can be plottes as follows:

```
t=linspace(-2,3,100);
plot(t, (t>=0).*(t<=1).*(1-2*abs(t-0.5)) ...
        -coeffs(1)*(t>=-1).*(t<=1).*(1-abs(t))...
        -coeffs(2)*(t>=0).*(t<=2).*(1-abs(t-1))...
        -coeffs(3)*(t>=-2).*(t<=0).*(1-abs(t+1))...
        -coeffs(4)*(t>=1).*(t<=3).*(1-abs(t-2)))
```

e. If we define

$$\hat{\psi} = \psi_{0,0} - \sum_{k=0}^{K} \left(\alpha_k \phi_{0,-k} - \beta_k \phi_{0,k+1}\right),$$

we have $2k$ unknowns. These can be determined if we require $2k$ vanishing moments.

Answer Ex. 5.6. 1: You can set for instance $H_0 = \{1/4, \underline{1/2}, 1/4\}$, and $H_1 = \{\underline{1}\}$ (when you write down the corresponding matrix you will see that $A_{0,1} = 1/2$, $A_{1,0} = 0$, so that the matrix is not symmetric)

Answer Ex. 5.6. 2: It turns out that this is wrong. In fact, the Haar wavelet is a counterexample!

Answer Ex. 5.6. 4: The following code can be used:

```
function xnew=DWTImpl(h0,h1,x,m)
  for mres=1:m
    len=length(x)/2^(mres-1);
    x(1:len)=rowsymmmratrans(h0,h1,x(1:len));

    % Reorganize the coefficients
    l=x(1:2:(len-1));
    h=x(2:2:len);
    x(1:len)=[l h];
  end
  xnew=x;
```

Answer Ex. 5.6. 5: The following code can be used:

```
function x=IDWTImpl(g0,g1,xnew,m)
  [a0,a1]=changecolumnrows(g0,g1);
  for mres=m:(-1):1
    len=length(xnew)/2^(mres-1);

    % Reorganize the coefficients first
    l=xnew(1:(len/2));
    h=xnew((len/2+1):len);
    xnew(1:2:(len-1))=l;
    xnew(2:2:len)=h;

    xnew(1:len)=rowsymmmratrans(a0,a1,xnew(1:len));
  end
  x=xnew;
```

Answer Ex. 5.6. 6: a. We have that $H_0 = \frac{1}{5}\{1, 1, \underline{1}, 1, 1\}$, and $H_1 = \frac{1}{3}\{-1, \underline{1}, -1\}$.

The frequency responses are

$$\begin{aligned}\lambda_{H_0}(\omega) &= \frac{1}{5}e^{2i\omega} + \frac{1}{5}e^{i\omega} + \frac{1}{5} + \frac{1}{5}e^{-i\omega}\frac{1}{5}e^{-2i\omega} \\ &= \frac{2}{5}\cos(2\omega) + \frac{2}{5}\cos\omega + \frac{1}{5} \\ \lambda_{H_1}(\omega) &= -\frac{1}{3}e^{i\omega} + \frac{1}{3} - \frac{1}{3}e^{-i\omega} = -\frac{2}{3}\cos\omega + \frac{1}{3}.\end{aligned}$$

Both filters are symmetric, and we have that `h0`$=(1/5, 1/5, 1/5)$, and `h1`$=(1/3, -1/3)$.
b. We have that $G_0 = \{1/4, \underline{1/2}, 1/4\}$, and $G_1 = \{1/16, -1/4, \underline{3/8}, -1/4, 1/16\}$.
The frequency responses are

$$\begin{aligned}\lambda_{G_0}(\omega) &= \frac{1}{4}e^{i\omega} + \frac{1}{2} + \frac{1}{4}e^{-i\omega} \\ &= \frac{1}{2}\cos(\omega) + \frac{1}{2} \\ \lambda_{G_1}(\omega) &= \frac{1}{16}e^{2i\omega} - \frac{1}{4}e^{i\omega} + \frac{3}{8} - \frac{1}{4}e^{-i\omega}\frac{1}{16}e^{-2i\omega} \\ &= \frac{1}{8}\cos(2\omega) - \frac{1}{2}\cos\omega + \frac{3}{8}.\end{aligned}$$

Both filters are symmetric, and we have that `g0`$=(1/2, 1/4)$, and `g1`$=(3/8, -1/4, 1/16)$.

Answer Ex. 5.6. 7: a. We have that $H_0 = \{1/16, 1/4, \underline{3/8}, 1/4, 1/16\}$, and $H_1 = \{-1/4, \underline{1/2}, -1/4\}$. The frequency responses are

$$\begin{aligned}\lambda_{H_0}(\omega) &= \frac{1}{16}e^{2i\omega} + \frac{1}{4}e^{i\omega} + \frac{3}{8} + \frac{1}{4}e^{-i\omega}\frac{1}{16}e^{-2i\omega} \\ &= \frac{1}{8}\cos(2\omega) + \frac{1}{2}\cos\omega + \frac{3}{8} \\ \lambda_{H_1}(\omega) &= -\frac{1}{4}e^{i\omega} + \frac{1}{2} - \frac{1}{4}e^{-i\omega} \\ &= -\frac{1}{2}\cos(\omega) + \frac{1}{2}.\end{aligned}$$

The two first rows in $P_{\mathcal{C}_1 \leftarrow \phi_1}$ are

$$\begin{pmatrix} 3/8 & 1/4 & 1/16 & 0 & \cdots & 1/16 & 1/4 \\ -1/4 & 1/2 & -1/4 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

The remaining rows are obtained by translating these in alternating order.
b. We have that $G_0 = \frac{1}{3}\{1, \underline{1}, 1\}$, and $G_1 = \frac{1}{5}\{1, -1, \underline{1}, -1, 1\}$. The frequency responses are

$$\begin{aligned}\lambda_{G_0}(\omega) &= \frac{1}{3}e^{i\omega} + \frac{1}{3} + \frac{1}{3}e^{-i\omega} = \frac{2}{3}\cos\omega + \frac{1}{3} \\ \lambda_{G_1}(\omega) &= \frac{1}{5}e^{2i\omega} - \frac{1}{5}e^{i\omega} + \frac{1}{5} - \frac{1}{5}e^{-i\omega}\frac{1}{5}e^{-2i\omega} \\ &= \frac{2}{5}\cos(2\omega) - \frac{2}{5}\cos\omega + \frac{1}{5}\end{aligned}$$

The two first columns in $P_{\phi_1 \leftarrow \mathcal{C}_1}$ are

$$\begin{pmatrix} 1/3 & -1/5 \\ 1/3 & 1/5 \\ 0 & -1/5 \\ 0 & 1/5 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1/3 & 1/5 \end{pmatrix}$$

The remaining columns are obtained by translating these in alternating order.

Answer Ex. 5.6. 8: The following code can be used:

```
function playDWTfilterslower(m,h0,h1,g0,g1)
  [S fs]=wavread('../castanets');
  newx=DWTImpl(h0,h1,S(1:2^17,1),m);
  len=length(newx);
  newx((len/2^m+1):length(newx))=zeros(length(newx)-len/2^m,1);
  newx=IDWTImpl(g0,g1,newx,m);
  playerobj=audioplayer(newx,fs);
  playblocking(playerobj);
```

Answer Ex. 5.6. 9: The following code can be used:

```
function playDWTfilterslowerdifference(m,h0,h1,g0,g1)
  [S fs]=wavread('../castanets');
  newx=DWTImpl(h0,h1,S(1:2^17,1),m);
  len=length(newx);
  newx(1:(len/2^m))=zeros(len/2^m,1);
  newx=IDWTImpl(g0,g1,newx,m);
  playerobj=audioplayer(newx,fs);
  playblocking(playerobj);
```

After the replacements in the function `playDWTall`, we get code which looks like this

```
function playDWTalldifference(m)
disp('Haar wavelet');
playDWTlowerdifference(m);
disp('Wavelet for piecewise linear functions');
playDWTfilterslowerdifference(m,[sqrt(2)],...
    [sqrt(2) -1/sqrt(2)],...
    [1/sqrt(2) 1/(2*sqrt(2))],...
```

```
    [1/sqrt(2)]);
disp('Wavelet for piecewise linear functions, alternative  version');
playDWTfilterslowerdifference(m,[3/(2*sqrt(2)) 1/(2*sqrt(2)) -1/(4*sqrt(2))],...
   [sqrt(2) -1/sqrt(2)],...
   [1/sqrt(2) 1/(2*sqrt(2))],...
   [3/(4*sqrt(2)) -1/(4*sqrt(2)) -1/(8*sqrt(2))]);
```

Answer Ex. 5.6. 10: The code which can be used looks like this a.

```
newx=IDWTImpl([1/sqrt(2) 1/(2*sqrt(2))],[1/sqrt(2)],...
              [-coeffs(1); -coeffs(2); -coeffs(4); 0; 0; 0; 0; ...
               -coeffs(3); 1; 0; 0; 0; 0; 0; 0; 0],1);
```

b.

```
g1=newx(2:6)';
[g1(5:(-1):3) g1(2:5)] % compact filter notation
g0=[1/sqrt(2) 1/(2*sqrt(2))];
omega=linspace(0,2*pi,100);
plot(omega,g1(1)+g1(2)*2*cos(omega)+g1(3)*2*cos(2*omega)...
           +g1(4)*2*cos(3*omega)+g1(5)*2*cos(4*omega))
```

c.

```
alpha=1/(g0(1)*g1(1)+2*g0(2)*g1(2));
h0=alpha*(-1).^(0:(length(g1)-1)).*g1;
h1=alpha*(-1).^(0:(length(g0)-1)).*g0;
```

d.

```
for m=1:4
    m
    playDWTfilterslower(m,h0,h1,g0,g1);
    playDWTfilterslowerdifference(m,h0,h1,g0,g1);
end
```

Answer Ex. 7.1. 1: We have that

$$((I \otimes T)(\boldsymbol{x} \otimes \boldsymbol{y}))_{i,j} = x_i(T(\boldsymbol{y}))_j$$
$$= x_i \frac{1}{2}(y_{j+1} - y_{j-1}) = \frac{1}{2}x_i y_{j+1} - \frac{1}{2}x_i y_{j-1} = \frac{1}{2}(\boldsymbol{x} \otimes \boldsymbol{y})_{i,j+1} - \frac{1}{2}(\boldsymbol{x} \otimes \boldsymbol{y})_{i,j-1}.$$

The same type of equation holds if we replace $\boldsymbol{x} \otimes \boldsymbol{y}$ with any matrix X, and we see that this corresponds to placing the computational molecule

$$\frac{1}{2}\begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

over the image samples, which is seen to coincide with the molecule from Equation 6.6 in Example 6.18.

Answer Ex. 7.1. 2: In Example 7.9 we showed that $((T\otimes I)X)_{i,j} = \frac{1}{2}(X_{i+1,j} - \frac{1}{2}X_{i-1,j}$. Using the previous exercise we get that

$$\begin{aligned}((T\otimes T)X)_{i,j} &= ((T\otimes I)((I\otimes T)X))_{i,j} \\ &= \frac{1}{2}(((I\otimes T)X))_{i+1,j} - \frac{1}{2}((I\otimes T)X))_{i-1,j} \\ &= \frac{1}{2}\left(\frac{1}{2}X_{i+1,j+1} - \frac{1}{2}X_{i+1,j-1}\right) - \frac{1}{2}\left(\frac{1}{2}X_{i-1,j+1} - \frac{1}{2}X_{i-1,j-1}\right) \\ &= \frac{1}{4}X_{i+1,j+1} - \frac{1}{4}X_{i+1,j-1} - \frac{1}{4}X_{i-1,j+1} + \frac{1}{4}X_{i-1,j-1}.\end{aligned}$$

This is the same as placing the computational molecule

$$\frac{1}{4}\begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

over the image samples, which is seen to coincide with the molecule from Equation 6.9 in Example 6.18

Answer Ex. 7.1. 5: We have that

$$\begin{aligned}F(\alpha\boldsymbol{x}_1 + \beta\boldsymbol{x}_2, \boldsymbol{y}) &= (\alpha\boldsymbol{x}_1 + \beta\boldsymbol{x}_2)\otimes\boldsymbol{y} = (\alpha\boldsymbol{x}_1 + \beta\boldsymbol{x}_2)\boldsymbol{y}^T \\ &= \alpha\boldsymbol{x}_1\boldsymbol{y}^T + \beta\boldsymbol{x}_2\boldsymbol{y}^T = \alpha(\boldsymbol{x}_1\otimes\boldsymbol{y}) + \beta(\boldsymbol{x}\otimes\boldsymbol{y}) \\ &= \alpha F(\boldsymbol{x}_1, \boldsymbol{y}) + \beta F(\boldsymbol{x}_1, \boldsymbol{y}).\end{aligned}$$

The second statement follows similarly.

Answer Ex. 7.1. 6: Multiplicaton with the matrix

$$T = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}$$

reverses the elements in a vector. This means that

$$((T\otimes I)(\boldsymbol{x}\otimes\boldsymbol{y}))_{i,j} = ((T\boldsymbol{x})\otimes\boldsymbol{y})_{i,j} = (T\boldsymbol{x})_i y_j = x_{M-1-i}y_j = (\boldsymbol{x}\otimes\boldsymbol{y})_{M-1-i,j}.$$

This means that also $((T\otimes I)X)_{i,j} = X_{M-1-i,j}$ for all X, so that $T\otimes I$ reverses rows, and thus is a solution to a.. Similarly one shows that $I\otimes T$ reverses columns, and is thus a solution to b.. It turns out that it is impossible to find S and T so that transposing a matrix X corresponds to computing $(S \otimes T)X$. To see why, S and T would need to fulfill

$$(S \otimes T)(\boldsymbol{e}_i \otimes \boldsymbol{e}_j) = (S\boldsymbol{e}_i) \otimes (T\boldsymbol{e}_j) = \boldsymbol{e}_j \otimes \boldsymbol{e}_i,$$

since $\boldsymbol{e}_j \otimes \boldsymbol{e}_i$ is the transpose of $\boldsymbol{e}_i \otimes \boldsymbol{e}_j$. This would require that $S\boldsymbol{e}_i = \boldsymbol{e}_j$ for all i, j, which is impossible.

Answer Ex. 7.2. 1: The following code can be used:

```
function newX=FFT2Impl(X)
  for k=1:2
      for s=1:size(X,2)
          X(:,s)=FFTImpl(X(:,s));
      end
      X=X';
  end
  newX=X;
```

```
function newX=IFFT2Impl(X)
  for k=1:2
      for s=1:size(X,2)
          X(:,s)=IFFTImpl(X(:,s));
      end
      X=X';
  end
  newX=X;
```

```
function newX=DCT2Impl(X)
  for k=1:2
      for s=1:size(X,2)
          X(:,s)=DCTImpl(X(:,s));
      end
      X=X';
  end
  newX=X;
```

```
function newX=IDCT2Impl(X)
  for k=1:2
      for s=1:size(X,2)
          X(:,s)=IDCTImpl(X(:,s));
      end
      X=X';
  end
  newX=X;
```

Answer Ex. 7.2. 2: The following code can be used:

```
function newX=transform2jpeg(X)
  numblocksx=size(X,1)/8;
  numblocksy=size(X,2)/8;
  for bx=0:(numblocksx-1)
    for by=0:(numblocksy-1)
      X((1+8*bx):8*(bx+1),(1+8*by):8*(by+1))=...
          DCT2Impl(X((1+8*bx):8*(bx+1),(1+8*by):8*(by+1)));
    end
  end
  newX=X;
```

```
function X=transform2invjpeg(newX)
  numblocksx=size(newX,1)/8;
  numblocksy=size(newX,2)/8;
  for bx=0:(numblocksx-1)
    for by=0:(numblocksy-1)
      newX((1+8*bx):8*(bx+1),(1+8*by):8*(by+1))=...
          IDCT2Impl(newX((1+8*bx):8*(bx+1),(1+8*by):8*(by+1)));
    end
  end
  X=newX;
```

Answer Ex. 8.1. 1: The following code can be used:

```
function Xnew=DWT2HaarImpl(X,m)
  for mres=1:m
    l1=size(X,1)/2^(mres-1);
    l2=size(X,2)/2^(mres-1);
    for s=1:l2
      X(1:l1,s)=DWTHaarImpl(X(1:l1,s),1);
    end
    X=X';

    for s=1:l1
      X(1:l2,s)=DWTHaarImpl(X(1:l2,s),1);
    end
    X=X';
  end
  Xnew=X;
```

```
function X=IDWT2HaarImpl(Xnew,m)
  for mres=m:(-1):1
```

```
    l1=size(Xnew,1)/2^(mres-1);
    l2=size(Xnew,2)/2^(mres-1);
    for s=1:l2
      Xnew(1:l1,s)=IDWTHaarImpl(Xnew(1:l1,s),1);
    end
    Xnew=Xnew';

    for s=1:l1
      Xnew(1:l2,s)=IDWTHaarImpl(Xnew(1:l2,s),1);
    end
    Xnew=Xnew';
  end
  X=Xnew;
```

Answer Ex. 8.1. 2: The following code achieves the task

```
function showDWTlower(m)
  img=double(imread('mm.gif','gif'));
  newimg=DWT2HaarImpl(img,m);
  [l1,l2]=size(img);
  tokeep=newimg(1:(l1/(2^m)),1:(l2/(2^m)));
  newimg=zeros(size(newimg));
  newimg(1:(l1/(2^m)),1:(l2/(2^m)))=tokeep;
  newimg=IDWT2HaarImpl(newimg,m);
  imageview(abs(newimg));
```

There is no reason to believe that image samples returned by the function lie in $[0, 255]$. You can check this by printing the maximum value in the returned array on screen inside this method.

Answer Ex. 8.1. 3: The following code can be used

```
function showDWTlowerdifference(m)
  img=double(imread('mm.gif','gif'));
  newimg=DWT2HaarImpl(img,m);
  [l1,l2]=size(img);
  newimg(1:(l1/2^m),1:(l2/2^m))=zeros(l1/2^m,l1/2^m);
  newimg=IDWT2HaarImpl(newimg,m);
  imageview(abs(newimg));
```

Answer Ex. 8.1. 4: The following code can be used:

```
function Xnew=DWT2Impl(h0,h1,X,m)
```

```
  for mres=1:m
    l1=size(X,1)/2^(mres-1);
    l2=size(X,2)/2^(mres-1);
    for s=1:l2
      X(1:l1,s)=DWTImpl(h0,h1,X(1:l1,s),1);
    end
    X=X';

    for s=1:l1
      X(1:l2,s)=DWTImpl(h0,h1,X(1:l2,s),1);
    end
    X=X';
  end
  Xnew=X;
```

```
function X=IDWT2Impl(g0,g1,Xnew,m)
  for mres=m:(-1):1
    l1=size(Xnew,1)/2^(mres-1);
    l2=size(Xnew,2)/2^(mres-1);
    for s=1:l2
      Xnew(1:l1,s)=IDWTImpl(g0,g1,Xnew(1:l1,s),1);
    end
    Xnew=Xnew';

    for s=1:l1
      Xnew(1:l2,s)=IDWTImpl(g0,g1,Xnew(1:l2,s),1);
    end
    Xnew=Xnew';
  end
  X=Xnew;
```

Answer Ex. 8.1. 5: The following code can be used:

```
function showDWTfilterslower(m,h0,h1,g0,g1)
  img=double(imread('mm.gif','gif'));
  newimg=DWT2Impl(h0,h1,img,m);
  [l1,l2]=size(img);
  tokeep=newimg(1:(l1/(2^m)),1:(l2/(2^m)));
  newimg=zeros(size(newimg));
  newimg(1:(l1/(2^m)),1:(l2/(2^m)))=tokeep;
  newimg=IDWT2Impl(g0,g1,newimg,m);
  imageview(abs(newimg));
```

Answer Ex. 8.1. 6: The following code can be used:

```
function showDWTfilterslowerdifference(m,h0,h1,g0,g1)
  img=double(imread('mm.gif','gif'));
  newimg=DWT2Impl(h0,h1,img,m);
  [l1,l2]=size(img);
  newimg(1:(l1/2^m),1:(l2/2^m))=zeros(l1/2^m,l1/2^m);
  newimg=IDWT2Impl(g0,g1,newimg,m);
  imageview(abs(newimg));
```

After the replacements in the function `showDWTall`, we get code which looks like this

```
function showDWTalldifference(m)
disp('Haar wavelet');
showDWTlowerdifference(m);
disp('Wavelet for piecewise linear functions');
showDWTfilterslowerdifference(m,[sqrt(2)],...
    [sqrt(2) -1/sqrt(2)],...
    [1/sqrt(2) 1/(2*sqrt(2))],...
    [1/sqrt(2)]);
disp('Wavelet for piecewise linear functions, alternative  version');
showDWTfilterslowerdifference(m,[3/(2*sqrt(2)) 1/(2*sqrt(2)) -1/(4*sqrt(2))],...
   [sqrt(2) -1/sqrt(2)],...
   [1/sqrt(2) 1/(2*sqrt(2))],...
   [3/(4*sqrt(2)) -1/(4*sqrt(2)) -1/(8*sqrt(2))]);
```

Answer Ex. 9.3. 3: You can argue in many ways here: For instance the derivative of $f^2(x)$ is $2f(x)f'(x)$, so that extremal points of f are also extremal points of f^2.

Answer Ex. 9.3. 7: a. If $c_{i,j} = 0$ the function to be minimized is

$$\alpha \sum_{i \leq n} c_{ii} x_i^2 - \sum_{j=1}^{n} \mu_j x_j.$$

The gradient of this function is $2\alpha C\boldsymbol{x} - \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ is the vector with μ in all entries. Lagrange multipliers thus gives that $2\alpha C\boldsymbol{x} - \boldsymbol{\mu} = \boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ is the vector with λ in all entries. This gives that $x_i = \frac{\mu+\lambda}{2\alpha c_i}$. If $\sum x_i = 1$ we must have that $\frac{\mu+\lambda}{2\alpha} \sum \frac{1}{c_i} = 1$, so that $\lambda = -\mu + \frac{2\alpha}{\sum 1/c_i}$.
b. When $n = 2$, we have that $x_2 = 1 - x_1$, so that

$$\begin{aligned} f(x_1, x_2) &= \alpha c_{11} x_1^2 + \alpha c_{22} x_2^2 + \alpha(c_{12} + c_{21}) x_1 x_2 - \mu x_1 - \mu x_2 \\ &= \alpha c_{11} x_1^2 + \alpha c_{22} (1 - x_1)^2 + \alpha(c_{12} + c_{21}) x_1 (1 - x_1) - \mu x_1 - \mu(1 - x_1) \\ &= \alpha(c_{11} + c_{22} - c_{12} - c_{21}) x_1^2 + \alpha(-2c_{22} + c_{12} + c_{21}) x_1 + \alpha c_{22} - \mu \end{aligned}$$

The derivative of this is $2\alpha(c_{11}+c_{22}-c_{12}-c_{21})x_1+\alpha(-2c_{22}+c_{12}+c_{21})$, which is 0 when $x_1 = -\frac{-2c_{22}+c_{12}+c_{21}}{2(c_{11}+c_{22}-c_{12}-c_{21})}$. This is not dependent on α.

Answer Ex. 9.3. 8: a. We have that $f(\boldsymbol{x}) = \sum_i q_i x_i$, so that $\frac{\partial f}{\partial x_i} = q_i$, so that $\nabla f(\boldsymbol{x}) = \boldsymbol{q}$. Clearly $\frac{\partial^2 f}{\partial x_i \partial x_j} = 0$, so that $\nabla^2 f(\boldsymbol{x}) = \mathbf{0}$.
b. Note that, in the text of the exercise, we should also have assumed that A is symmetric. We have that

$$f(\boldsymbol{x}) = \frac{1}{2}\sum_{i,j} x_i A_{ij} x_j = \frac{1}{2}\sum_i A_{ii}x_i^2 + \frac{1}{2}\sum_{i,j,i\neq j} x_i A_{ij} x_j,$$

so that

$$\begin{aligned}\frac{\partial f}{\partial x_i} &= A_{ii}x_i + \frac{1}{2}\sum_{j,j\neq i} x_j(A_{ij}+A_{ji}) = \frac{1}{2}\sum_j x_j(A_{ij}+A_{ji}) \\ &= \frac{1}{2}\sum_j (A_{ij} + (A^T)_{ij})x_j = (\frac{1}{2}(A+A^T)\boldsymbol{x})_i\end{aligned}$$

This gives $\nabla f = \frac{1}{2}(A+A^T)\boldsymbol{x}$. In particular, when A is symmetric, this gives $\nabla f = A\boldsymbol{x}$. Finally we get

$$\frac{\partial f}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_j}(\frac{1}{2}\sum_j (A_{ij}+(A^T)_{ij})x_j) = \frac{1}{2}\sum_j (A_{ij}+(A^T)_{ij}),$$

so that $\nabla^2 f = \frac{1}{2}(A+A^T)$. In particular, when A is symmetric we get $\nabla^2 f = A$.

Answer Ex. 9.3. 9: First note that $f(0,0) = 3$, and that $f(2,1) = 8$ We have that $\nabla f = (2x_1+3x_2, 3x_1-10x_2)$, and that $\nabla f(0,0) = (0,0)$, and $\nabla f(2,1) = (7,-4)$. The first order Taylor approximation at $(0,0)$ is thus

$$f(0,0) + \nabla f(0,0)^T(\boldsymbol{x} - (0,0)) = 3.$$

The first order Taylor approximation at $(2,1)$ is

$$\begin{aligned}f(2,1) + \nabla f(2,1)^T(\boldsymbol{x}-(2,1)) &= 8 + (7,-4)^T(x_1-2, x_2-1) \\ &= 8 + 7(x_1-2) - 4(x_2-1) = 7x_1 - 4x_2 - 2.\end{aligned}$$

Answer Ex. 9.3. 11: If A is positive definite then its eigenvalues λ_i are positive. The eigenvalues of A^{-1} are $1/\lambda_i$, which also are positive, so that A^{-1} also is positive definite.

Answer Ex. 10.3. 1: The function $f(x, y, z) = x^2 + y^2 - z$ is convex (the Hessian is positive semidefinite). The set in question can be written as the points where $f(x, y, z) \leq 0$, which is a sublevel set, and therefore convex.

Answer Ex. 10.3. 5: Write B in row echelon form, to see which are pivot variables. Express these variables in terms of the free variables, and replace the pivot variables in all the equations. $A\boldsymbol{x} \geq \boldsymbol{b}$ then takes the form $C\boldsymbol{x} \geq \boldsymbol{b}$ (where $\boldsymbol{x}$ now is a shorter vector), and this can be written as $-C\boldsymbol{x} \leq -\boldsymbol{b}$, which is on the new form with $H = -C$, $\boldsymbol{h} = -\boldsymbol{b}$. Note that this strategy rewrites the vector $\boldsymbol{c}$ to a shorter vector.

Answer Ex. 10.3. 6: Let $\boldsymbol{y} = \sum_{j=1}^{t} \lambda_j \boldsymbol{x}_j$ and $\boldsymbol{z} = \sum_{j=1}^{t} \mu_j \boldsymbol{x}_j$, where all $\lambda_j, \mu_j \geq 0$, and $\sum_{j=1}^{t} \lambda_j = 1$, $\sum_{j=1}^{t} \mu_j = 1$. For any $0 \leq \lambda \leq 1$ we have that

$$(1-\lambda)\boldsymbol{y} + \lambda\boldsymbol{z} = (1-\lambda)\sum_{j=1}^{t} \lambda_j \boldsymbol{x}_j + \lambda \sum_{j=1}^{t} \mu_j \boldsymbol{x}_j = \sum_{j=1}^{t} ((1-\lambda)\lambda_j + \lambda\mu_j)\boldsymbol{x}_j.$$

The sum of the coefficients here is

$$\sum_{j=1}^{t} ((1-\lambda)\lambda_j + \lambda\mu_j) = (1-\lambda)\sum_{j=1}^{t} \lambda_j + \lambda \sum_{j=1}^{t} \mu_j = 1 - \lambda + \lambda = 1,$$

so that C is a convex set.

Answer Ex. 10.3. 7: Follows from Proposition 10.3, since $f(x) = e^x$ is convex, and $\boldsymbol{H}(\boldsymbol{x}) = \sum_{j=1}^{n} x_j$ is affine.

Answer Ex. 10.3. 9: a. λf is convex if $\lambda \geq 0$, concave if $\lambda \leq 0$.
b. $\min\{f, g\}$ may be neither convex or concave, consider the functions $f(x) = x^2$, $g(x) = (x-1)^2$.
c. $|f|$ may be neither convex or concave, consider the function $f(x) = x^2 - 1$.

Answer Ex. 11.2. 1: $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ is equivalent to $\|\boldsymbol{F}(\boldsymbol{x})\|^2 = \sum_i F_i(\boldsymbol{x}) = 0$, where F_i are the component functions of $\boldsymbol{F}$. Solving $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ thus is equivalent to showing that 0 is the minimum value of $\sum_i F_i(\boldsymbol{x})$.

Answer Ex. 11.2. 2: Here we construct the function $f(x) = T(x) - x = x/2 - 3x^3/2$, which has derivative $f'(x) = 1/2 - 9x^2/2$. We can then run Newton's method as follows:

```
newtonmult(sqrt(5/3),@(x)(0.5*x-1.5*x^3),@(x)(0.5-4.5*x^2))
```

This converges to the zero we are looking for, which we easily compute as $x = \sqrt{1/3}$.

Answer Ex. 11.2. 4: a. The function $x \to \|A\boldsymbol{x}\|$ is continuous, and any continuous function achieves a supremum in a closed set (here $\|\boldsymbol{x}\| = 1$).
b. For $n = 2$, it is clear that the sublevel set is the square with corners $(1, 0)$, $(-1, 0)$, $(0, 1)$, $(0, -1)$. The function $f(\boldsymbol{x}) = \|A\boldsymbol{x}\|$ is the composition of a convex function and an affine function, so that it must be convex. If $\boldsymbol{x} \in \mathbb{R}^n$ and $\|\boldsymbol{x}\|_1 = 1$, we can write $\boldsymbol{x} = \sum_{i=1}^n \lambda_i \boldsymbol{v}_i$, where $0 \le \lambda_i \le 1$, $\sum_{i=1}^n \lambda_i = 1$, and $\boldsymbol{v}_i = \pm \boldsymbol{e}_i$ (i.e. it absorbs the sign of the ith component). If $\boldsymbol{w}$ is the vector among $\{\pm \boldsymbol{e}_j\}_j$ so that $f(\pm \boldsymbol{e_j}) \le f(\boldsymbol{w})$ for all j and all signs, Jensen's inequality (Theorem 10.4) gives

$$f(\boldsymbol{x}) = f\left(\sum_{i=1}^n \lambda_i \boldsymbol{v}_i\right) \le \sum_{i=1}^n \lambda_i f(\boldsymbol{v}_i) \le \sum_{i=1}^n \lambda_i f(\boldsymbol{w}) = f(\boldsymbol{w}),$$

so that f assumes its maximum in $\boldsymbol{w}$. For this particular f, if $\boldsymbol{w} = \pm \boldsymbol{e}_k$, the maximum is

$$\|A\boldsymbol{w}\|_1 = \| \pm \operatorname{col}_k A\| = \sum_{i=1}^n | \pm a_{ik}| = \sum_{i=1}^n |a_{ik}|.$$

It is now clear that $\|A\| = \sup_k \sum_{i=1}^n |a_{ik}|$.

Answer Ex. 11.2. 5: We are asked to find for which A we have that $\|A\boldsymbol{x}\|_1 < \|\boldsymbol{x}\|_1$ for any $\boldsymbol{x}$. From the previous exercise we know that this happens if and only if $\|A\| < 1$, i.e. when $\sum_{i=1}^n |a_{ik}| < 1$ for all k.

Answer Ex. 11.2. 6: You can write

```
newtonmult(x0,...
  @(x)([x(1)^2-x(1)/x(2)^3+cos(x(1))-1; 5*x(1)^4+2*x(1)^3-tan(x(1)*x(2)^8)-3]),...
  @(x)([2*x(1)-1/x(2)^3-sin(x(1))                          3*x(1)/x(2)^4; ...
        20*x(1)^3+6*x(1)^2-x(2)^8/(cos(x(1)*x(2)^8))^2      -8*x(1)*x(2)^7/(cos(x(1)*x(2)^8))^2])...
)
```

Answer Ex. 12.2. 4: The gradient of f is $\nabla f = (4 + 2x_1, 6 + 4x_2)$, and the Hessian matrix is $\nabla^2 f = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$, which is positive definite. The only stationary point is $(-2, -3/2)$, which is a minimum.
The gradient of g is $\nabla g = (4 + 2x_1, 6 - 4x_2)$, and the Hessian matrix is $\nabla^2 g =$

$\begin{pmatrix} 2 & 0 \\ 0 & -4 \end{pmatrix}$, which is indefinite. The only stationary point is $(-2, 3/2)$, which must be a saddle point.

Answer Ex. 12.2. 5: The gradient is $\nabla f = (-400x_1(x_2 - x_1^2) - 2(1 - x_1), 200(x_2 - x_1^2))$. The Hessian matrix is

$$\nabla^2 f = \begin{pmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}.$$

Clearly the only stationary point is $x = (1, 1)$, and we get that

$$\nabla^2 f(1,1) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}.$$

It is straightforward to check that this matrix is positive definite, so that $(1, 1)$ is a local minimum.

Answer Ex. 12.2. 6: The steepest descent method takes the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k),$$

where $\nabla f(\boldsymbol{x}_k) = A\boldsymbol{x}_k - \boldsymbol{b}$. We have that

$$\begin{aligned} f(\boldsymbol{x}_{k+1}) &= (1/2)\boldsymbol{x}_{k+1}^T A \boldsymbol{x}_{k+1} - \boldsymbol{b}^T \boldsymbol{x}_{k+1} \\ &= \frac{1}{2}\left(\nabla f(\boldsymbol{x}_k)^T A \nabla f(\boldsymbol{x}_k)\right)\alpha_k^2 - \frac{1}{2}\left(\boldsymbol{x}_k^T A \nabla f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T A \boldsymbol{x}_k\right)\alpha_k \\ &+ \frac{1}{2}\boldsymbol{x}_k^T A \boldsymbol{x}_k - \boldsymbol{b}^T(\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)) \\ &= \frac{1}{2}\left(\nabla f(\boldsymbol{x}_k)^T A \nabla f(\boldsymbol{x}_k)\right)\alpha_k^2 \\ &+ \left(\boldsymbol{b}^T \nabla f(\boldsymbol{x}_k) - \frac{1}{2}\left(\boldsymbol{x}_k^T A \nabla f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T A \boldsymbol{x}_k\right)\right)\alpha_k \\ &+ \frac{1}{2}\boldsymbol{x}_k^T A \boldsymbol{x}_k - \boldsymbol{b}^T \boldsymbol{x}_k. \end{aligned}$$

Now, since we claim that $\nabla f(\boldsymbol{x}_k)$ is an eigenvector, and that A is symmetric, we get that $A\nabla f(\boldsymbol{x}_k) = \lambda \nabla f(\boldsymbol{x}_k)$ and $\nabla f(\boldsymbol{x}_k)^T A = \lambda \nabla f(\boldsymbol{x}_k)^T$, where λ is the corresponding eigenvalue. This means that the above can be written

$$f(\boldsymbol{x}_{k+1}) = \frac{1}{2}\lambda \|\nabla f(\boldsymbol{x}_k)\|^2 \alpha_k^2 + \left(\boldsymbol{b}^T \nabla f(\boldsymbol{x}_k) - \boldsymbol{x}_k^T A \nabla f(\boldsymbol{x}_k)\right)\alpha_k + \frac{1}{2}\boldsymbol{x}_k^T A \boldsymbol{x}_k - \boldsymbol{b}^T \boldsymbol{x}_k$$

If we take the derivative of this w.r.t. α_k and set this to 0 we get

$$\begin{aligned} \alpha_k &= -\frac{\boldsymbol{b}^T \nabla f(\boldsymbol{x}_k) - \boldsymbol{x}_k^T A \nabla f(\boldsymbol{x}_k)}{\lambda \|\nabla f(\boldsymbol{x}_k)\|^2} = \frac{\left(\boldsymbol{x}_k^T A - \boldsymbol{b}^T\right)\nabla f(\boldsymbol{x}_k)}{\lambda \|\nabla f(\boldsymbol{x}_k)\|^2} \\ &= \frac{(A\boldsymbol{x}_k - \boldsymbol{b})^T \nabla f(\boldsymbol{x}_k)}{\lambda \|\nabla f(\boldsymbol{x}_k)\|^2} = \frac{\nabla f(\boldsymbol{x})^T \nabla f(\boldsymbol{x}_k)}{\lambda \|\nabla f(\boldsymbol{x}_k)\|^2} = \frac{1}{\lambda}. \end{aligned}$$

This means that $\alpha_k = \frac{1}{\lambda}$ is the step size we should use when we perform exact line search. We now compute that

$$\begin{aligned}\nabla f(\boldsymbol{x}_{k+1}) &= A\boldsymbol{x}_{k+1} - \boldsymbol{b} = A\left(\boldsymbol{x}_k - \frac{1}{\lambda}\nabla f(\boldsymbol{x}_k)\right) - \boldsymbol{b}\\ &= A\boldsymbol{x}_k - \frac{1}{\lambda}A\nabla f(\boldsymbol{x}_k) - \boldsymbol{b} = A\boldsymbol{x}_k - \nabla f(\boldsymbol{x}_k) - \boldsymbol{b}\\ &= \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_k) = \boldsymbol{0},\end{aligned}$$

which shows that the minimum is reached in one step.

Answer Ex. 12.2. 7: a. This is simply Exercise 9.3.8.
b. If $\nabla^2 f(\boldsymbol{x})$ is positive definite, its eigenvalues are positive, so that the determinant is positive, and that the matrix is invertible. $\boldsymbol{h} = -\nabla^2 f(\boldsymbol{x}_k)^{-1}\nabla f(\boldsymbol{x}_k)$ follows after multiplying with the inverse.

Answer Ex. 12.2. 8: Here we have said nothing about the step length, but we can implement this as in the function `newtonbacktrack` as follows:

```
function [xopt,numit]=steepestdescent(f,df,x0)
  epsilon=10^(-3);
  xopt=x0;
  maxit=100;
  for numit=1:maxit
      d=-df(xopt);
      eta=-df(xopt)'*d;
      if eta/2<epsilon
          break;
      end
      alpha=armijorule(f,df,xopt,d);
      xopt=xopt+alpha*d;
  end
```

The algorithm can be tested on the first function from Exercise 4 as follows:

```
f=@(x)(4*x(1)+6*x(2)+x(1)^2+2*x(2)^2);
df=@(x)([4+2*x(1);6+4*x(2)])
steepestdescent(f,df,[-1;-1])
```

Answer Ex. 12.2. 9: The function can be implemented as follows:

```
function alpha=armijorule(f,df,x,d)
  beta=0.2; s=0.5; sigma=10^(-3);
  m=0;
```

```
  while (f(x)-f(x+beta^m*s*d) < -sigma *beta^m*s *(df(x))'*d)
      m=m+1;
  end
  alpha = beta^m*s;
```

Answer Ex. 12.2. 10: The function can be implemented as follows:

```
function [xopt,numit]=newtonbacktrack(f,df,d2f,x0)
  epsilon=10^(-3);
  xopt=x0;
  maxit=100;
  for numit=1:maxit
      d=-d2f(xopt)\df(xopt);
      eta=-df(xopt)'*d;
      if eta/2<epsilon
          break;
      end
      alpha=armijorule(f,df,xopt,d);
      xopt=xopt+alpha*d;
  end
```

Answer Ex. 13.3. 3: This is the same as finding the minimum of $f(x_1,\ldots,x_n) = -x_1x_2\cdots x_n$. This boils down to the equations $-\prod_{i\neq j} x_i = 1$, since clearly the minimum is not attained when there are any active constraints. This implies that $x_1 = \ldots = x_n$, so that all $x_i = 1/n$. It is better to give a direct argument here that this must be a minimum, than to attempt to analyse the second order conditions for a minimum.

Answer Ex. 13.3. 4: We can formulate the problem as finding the minimum of $f(x_1,x_2) = (x_1-a_1)^2 + (x_2-a_2)^2$ subject to the constraint $h_1(x_1,x_2) = x_1^2+x_2^2 = 1$. The minimum can be found geometrically by drawing a line which passes through $\boldsymbol{a}$ and the origin, and reading the intersection with the unit circle. This follows also from that ∇f is parallel to $\boldsymbol{x}-\boldsymbol{a}$, ∇h_1 is parallel to $\boldsymbol{x}$, and from that the KKT-conditions say that these should be parallel.

Answer Ex. 13.3. 6: We rewrite the constraint as $g_1(x_1,x_2) = x_1^2+x_2^2-1 = 0$, and get that $\nabla g_1(x_1,x_2) = (2x_1,2x_2)$. Clearly all points are regular, since $\nabla g_1(x_1,x_2) \neq 0$ whenever $g_1(x_1,x_2) = 0$. Since $\nabla f = (1,1)$ we get that the gradient of the Lagrangian is

$$\begin{pmatrix}1\\1\end{pmatrix} + \lambda\begin{pmatrix}2x_1\\2x_2\end{pmatrix} = \mathbf{0},$$

which gives that $x_1 = x_2$. This gives us the two possible feasible points $(1/\sqrt{2}, 1/\sqrt{2})$ and $(-1/\sqrt{2}, -1/\sqrt{2})$. For the first we see that $\lambda = -1/\sqrt{2}$, for the second we see that $\lambda = 1/\sqrt{2}$. The Hessian of the Lagrangian is $\lambda \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$. For the point $(1/\sqrt{2}, 1/\sqrt{2})$ this is negative definite since λ is negative, for the point $(-1/\sqrt{2}, -1/\sqrt{2})$ this is positive definite since λ is positive. From the second order conditions it follows that the minimum is attained in $(-1/\sqrt{2}, -1/\sqrt{2})$.
If we instead eliminated x_2 we must write $x_2 = -\sqrt{1-x_1^2}$ (since the positive square root gives a bigger value for f), so that we must minimize $f(x) = x - \sqrt{1-x^2}$ subject to the constraint $-1 \leq x \leq 1$. The derivative of this is $1 + \frac{x}{\sqrt{1-x^2}}$, which is zero when $x = -\frac{1}{\sqrt{2}}$, which we found above. We also could have found this by considering the two inequality constraints $-x-1 \leq 0$ and $x - 1 \leq 0$.
If the first one of these is active (i.e. $x = -1$), the KKT conditions say that $f'(-1) > 0$. However, this is not the case since $f'(x) \to -\infty$ when $x \to -1_+$. If the second constraint is active (i.e. $x = 1$), the KKT conditions say that $f'(1) < 0$. This is not the case since $f'(x) \to \infty$ when $x \to 1-$. When we have no active constraint, the problem boils down to setting the derivative to zero, in which case we get the solution we already have found.

Answer Ex. 13.3. 9: We define $h_1(x_1, \ldots, x_n) = x_1 + \ldots + x_n - 1$, and find that $\nabla h_1 = (1, 1, \ldots, 1)$. The stationary points are characterized by $\nabla f + \boldsymbol{\lambda}^T(1, 1, \ldots, 1) = \mathbf{0}$, which has a solution exactly when $\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} = \ldots = \frac{\partial f}{\partial x_n}$.

Answer Ex. 13.3. 10: We substitute $x_n = 1 - x_1 - \ldots - x_{n-1}$ in the expression for f, to turn the problem into one of minimizing a function in $n-1$ variables.

Answer Ex. 13.3. 11: The problem can be rewritten to the following minimation problem:

$$\min\{-\boldsymbol{x}^T A\boldsymbol{x} : g_1(\boldsymbol{x}) = \|\boldsymbol{x}\| - 1 = 0\}.$$

We have that $\nabla f(\boldsymbol{x}) = -A\boldsymbol{x}$, and $\nabla g_1(\boldsymbol{x}) = \frac{2\boldsymbol{x}}{2\|\boldsymbol{x}\|} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$. Clearly all points are regular, and we get that

$$\nabla f + \lambda \nabla g_1 = -A\boldsymbol{x} + \lambda \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} = \mathbf{0}.$$

Since we require that $\|\boldsymbol{x}\| = 1$ we get that $A\boldsymbol{x} = \lambda \boldsymbol{x}$. In other words, the optimal point $\boldsymbol{x}$ is an eigenvector of A, and the Lagrange multiplier is the corresponding eigenvalue.

Answer Ex. 13.3. 12: Define $f(x_1, x_2, x_3) = (1/2)(x_1^2+x_2^2+x_3^2)$ and $g_1(x_1, x_2, x_3) = x_1+x_2+x_3$. We have that $\nabla f = (x_1, x_2, x_3)$, $\nabla g_1 = (1, 1, 1)$. Clearly all points are regular points. If there are no active constraints, we must have that $\nabla f = \mathbf{0}$, so that $x_1 = x_2 = x_3 = 0$, which does not fulfill the constraint. If teh constraint is active we must have that $(x_1, x_2, x_3) + \mu(1, 1, 1) = \mathbf{0}$ for some $\mu \leq 0$, which is satisfied when $x_1 = x_2 = x_3 < 0$. Clearly we must have that $x_1 = x_2 = x_3 = -2$. The Hessian of $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ is easily computed to be positive definite, so that we have found a minimum.

Answer Ex. 13.3. 13: We need to minimze $f(x_1, x_2) = (x_1-3)^2+(x_2-5)^2+x_1x_2$ subject to the constraints

$$\begin{aligned}
g_1(x_1, x_2) &= -x_1 \leq 0 \\
g_2(x_1, x_2) &= -x_2 \leq 0 \\
g_3(x_1, x_2) &= x_1 - 1 \leq 0 \\
g_4(x_1, x_2) &= x_2 - 1 \leq 0.
\end{aligned}$$

We have that $\nabla f = (2(x_1 - 3) + x_2, 2(x_2 - 5) + x_1)$, and $\nabla g_1 = (-1, 0)$, $\nabla g_2 = (0, -1)$, $\nabla g_3 = (1, 0)$, $\nabla g_4 = (0, 1)$. If there are no active constraints the KKT conditions say that $\nabla f = \mathbf{0}$, so that

$$2x_1 + x_2 = 6x_1 + 2x_2 \qquad = 10$$

which gives that $x_1 = 2/3$ and $x_2 = 14/3$. This point does not satisfy the constraints, however.
Assume now that we have one active constraint. We have four possibilities in this case (and any solution will be regular). If the first constraint is active the KKT conditions say that

$$\begin{pmatrix} 2x_1 + x_2 - 6 \\ x_1 + 2x_2 - 10 \end{pmatrix} + \mu \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \mathbf{0}$$

Setting $x_1 = 0$ we get that $x_2 - 6 = \mu$ and $2x_2 - 10 = 0$, so that $x_2 = 5$, which does not satisfy the constraints.
If the second constraint is active we get similarly that $2x_1 - 6 = 0$, which also does not satisfy the constraints.
If the third constraint is active we get $2x_2 - 9 = 0$, which does not satisfy the constraints.
If the fourth constraint is active we get $2x_1 - 5 = 0$, which does not satisfy the constraint.
Assume that we have two active constraints. Also here there are four possibilities (and any solution will be regular):
$x_1 = x_2 = 0$: The KKT consitions say that $(-6, -10) + (-\mu_1, -\mu_2) = \mathbf{0}$, which is impossible since μ_1, μ_2 are positive.
$x_1 = 0, x_2 = 1$: The KKT conditions say that $(-5, -8) + (-\mu_1, \mu_4) = \mathbf{0}$, which also is impossible

$x_1 = 1, x_2 = 0$: The KKT conditions say that $(-4,-9) + (\mu_3, -\mu_2) = \mathbf{0}$, which also is impossible
$x_1 = x_2 = 1$: The KKT conditions say that $(-3,-7) + (\mu_3, \mu_4)$, which has a solution.
Clearly it is not possible to have more than two active constraints. The minimum point is therefore $(1,1)$.

Answer Ex. 13.3. 14: We can define $g_1(x_1,x_2) = x_1^2 + x_2^2 - 2$, so that the only constraint is $g_1(x_1,x_2) \leq 0$. We have that $\nabla g_1 = (2x_1, 2x_2)$, and this can be zero if and only if $x_1 = x_2 = 0$. However $g_1(0,0) = -2 < 0$, so that the equality is not active. This means that all points are regular for this problem.
We compute that $\nabla f = (1,1)$. If g_1 is not an active inequality, the KKT conditions say that $\nabla f = 0$, which is impossible. If g_1 is active, we get that

$$\nabla f(x_1,x_2) + \mu \nabla g_1(x_1,x_2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \mu \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

so that $1 = -2\mu x_1$ and $1 = -2\mu x_2$ for some $\mu \geq 0$. This is satisfied if $x_1 = x_2$ is negative. For g_1 to be active we must have that $x_1^2 + x_2^2 = 2$, which implies that $x_1 = x_2 = -1$. We have that $f(-1,-1) = -2$.

Answer Ex. 13.3. 15: We define $g_j(\boldsymbol{x}) = -x_j$ for $j = 1, \ldots, n$, and $g_{n+1}(\boldsymbol{x}) = \sum_{j=1}^n x_j - 1$. We have that $\nabla g_j = -\boldsymbol{e}_j$ for $1 \leq j \leq n$, and $\nabla g_{n+1} = (1,1,\ldots,1)$. If there are no active inequalities, we must have that $\nabla f(\boldsymbol{x}) = \mathbf{0}$. If the last constraint is not active we have that

$$\nabla f = \sum_{j \in A(x), j \leq n} \mu_j \boldsymbol{e}_j,$$

i.e. ∇f points into the cone spanned by $\boldsymbol{e}_j$, $j \in A(x)$. If the last constraint is active also , we see that

$$\nabla f = \sum_{j \notin A(x), j \leq n} -\mu_{n+1} \boldsymbol{e}_j \sum_{j \in A(x), j \leq n} (\mu_j - \mu_{n+1}) \boldsymbol{e}_j.$$

∇f is on this form whenever components outside the active set are equal and ≤ 0, and all are components are greater than or equal to this.

Answer Ex. 14.2. 1: The constraint $A\boldsymbol{x} = \boldsymbol{b}$ actually yields one constraint per row in A, and the gradient of the i'th constraint is the i'th row in A. This gives the following sum in the KKT conditions:

$$\sum_{i=1}^m \nabla g_i \lambda_i = \sum_{i=1}^m a_{i\cdot}^T \lambda_i == \sum_{i=1}^m (A^T)_{\cdot i} \lambda_i = A^T \lambda.$$

The gradient of $f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{h} + \frac{1}{2}\boldsymbol{h}^T \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{h}$ is $\nabla f(\boldsymbol{x}_k) + \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{h}$. The KKT conditions are thus $\nabla f(\boldsymbol{x}_k) + \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{h} + A^T\lambda = \boldsymbol{0}$ and $A\boldsymbol{h} = \boldsymbol{0}$. This can be written as the set of equations

$$\begin{aligned} \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{h} + A^T\lambda &= -\nabla f(\boldsymbol{x}_k) \\ A\boldsymbol{h} + 0\lambda &= \boldsymbol{0}, \end{aligned}$$

from which the stated equation system follows.

Answer Ex. 14.2. 2: a. We can set $g_1(x,y) = -x$, $g_2(x,y) = -y$, $A = \begin{pmatrix}-1 & 1\end{pmatrix}$, and $\boldsymbol{b} = \begin{pmatrix}1\end{pmatrix}$.
c. The barrier problem here is to minimize $x + y - \mu \ln x - \mu \ln y$ subject to the constraint $y - x = 1$. The gradient of the Lagrangian is $(1-\mu/x, 1-\mu/y) + A^T\lambda$. If this is $\boldsymbol{0}$ we must have that $1 - \mu/x = \mu/y - 1$, so that $2xy = \mu(x+y)$ The constraint gives that $y = x+1$, so that $2x(x+1) = \mu(2x+1)$. This can be written as $2x^2 + (2-\mu)x - \mu = 0$, which has the solution

$$x = \frac{-(2-\mu) \pm \sqrt{(2-\mu)^2 + 8\mu}}{4} = \frac{-(2-\mu) \pm (2+\mu)}{4},$$

which gives the two solutions $(x,y) = (\mu/2, \mu/2+1)$ and $(x,y) = (-1,0)$. Only the first solution here is within the domain of definition for f, so the barrier method obtains this minimum.
d. By inserting $y = x+1$ for the constraint we see that we need to minimize $g(x) = 2x + 1$ subject to $x \geq 0$, which clearly has a minimum for $x = 0$, and then $y = 1$. This gives the same minimum as in c.
e. The KKT conditions takes one of the following forms:

- If there are no active inequalities:

  $$\nabla f + A^T\lambda = (1,1) + \lambda(-1,1) = \boldsymbol{0},$$

  which has no solutions.

- The first inequality is active (i.e. $x = 0$):

  $$\nabla f + A^T\lambda + \mu_1 \nabla g_1 = (1,1) + \lambda(-1,1) + \mu_1(-1,0) = (1-\lambda-\mu_1, 1+\lambda) = \boldsymbol{0},$$

  which gives that $\lambda = -1$ and $\mu_1 = \mu_1 = 2$. When $x = 0$ the equaility constraint gives that $y = 1$, so that $(0,1)$ satisfies the KKT conditions.

- The second inequality is active (i.e $y = 0$): The first constraint then gives that $x = -1$, which does not give a feasible point.

In conclusion, $(0,1)$ is the only point which satisfies the KKT conditions. If we attempt the second order test, we will see that it is inconclusive, since the

Hessian of the Lagrangian is zero. To prove that $(1,0)$ must be a minimum, you can argue that f is very large outside any rectangle, so that it must have a minimum on this rectangle (the rectangle is a closed and bounded set).
f. With the barrier method we obtained the solution $\boldsymbol{x}(\mu) = (\mu/2, \mu/2+1)$. Since this converges to $(0,1)$ as $\mu \to 0$, the central path converges to the solution we have found.

Answer Ex. 14.2. 3: You can use the following code:

```
IPBopt(@(x)(x(1)+x(2)),@(x)(-x(1)),@(x)(-x(2)),...
       @(x)([1;1]),@(x)([-1;0]),@(x)([0;-1]),...
       @(x)(zeros(2)),@(x)(zeros(2)),@(x)(zeros(2)),...
       [-1 1],1,[4;5])
```

Answer Ex. 14.2. 4: Here we have that $\nabla f = 2x$, $\nabla g_1 = -1$, $\nabla g_2 = 1$. If there are no active constraints the KKT conditions say that $2x = 0$, so that $x = 0$, which is outside the domain of definition for f.
If the first constraint is active we get that $2x - \mu_1 = 4 - \mu_1 = 0$, so that $\mu_1 = 4$. This is a candidate for the minimum (clearly the second order conditions for a minimum is fulfilled here as well, since the Hessian of the Lagrangian is 2).
If the second constraint is active we get that $2x + \mu_2 = 4 + \mu_2 = 0$, so that $\mu_2 = -4$, so that this gives no candidate for a solution.
It is impossible for both constraints to be active at the same time, so $x = 2$ is the unique minimum.

Answer Ex. 14.2. 6: You can use the following code:

```
IPBopt2(@(x)((x-3).^2),@(x)(2-x),@(x)(x-4),...
       @(x)(2*(x-3)),@(x)(-1),@(x)(1),...
       @(x)(2),@(x)(0),@(x)(0),3.5)
```