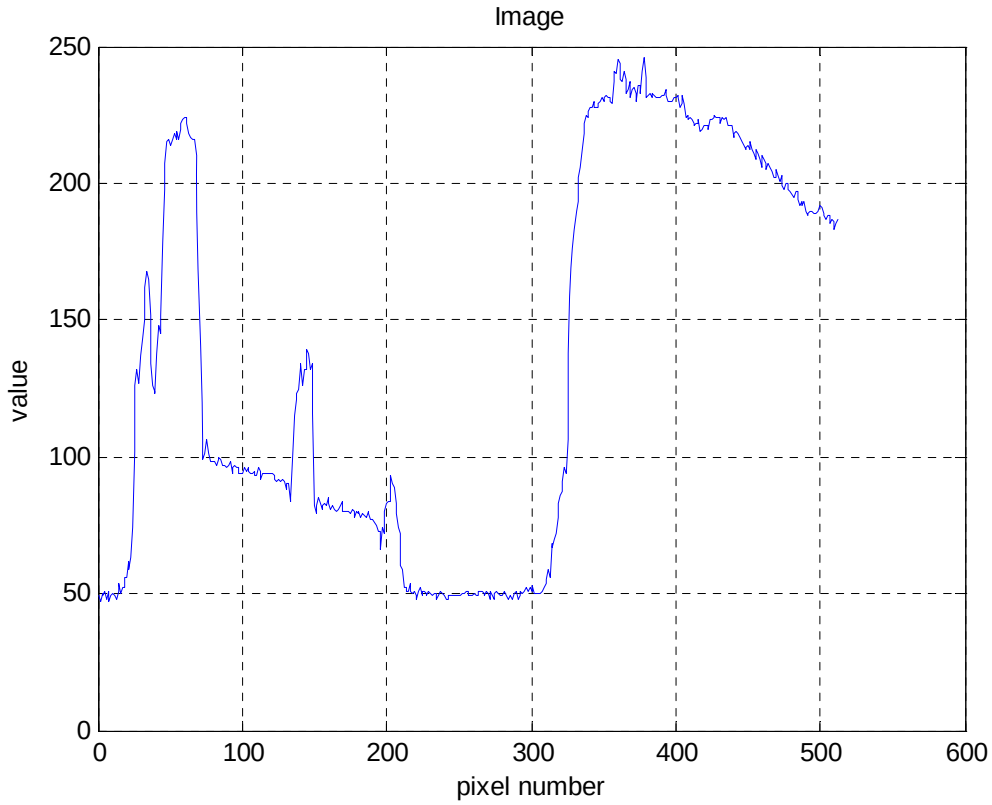
18-799F Algebraic Theory of Signal Processing
Spring 2007
Solutions: Assignment 7

(a)



(b)

```
norm(double(hw_image))^2

ans =

  1.1936e+007
```

(c)

DCT, type II :

```
function res = orthog_DCT2(N,X)
    % ORTHOG_DCT2(N,X)
    % Computes the orthogonal DCT-2 on input signal X

    res = zeros(N,1);

    % Compute the first coefficient separately, since it differs from
others by a
    % factor of 1/sqrt(2)
    for m=0:N-1
        res(1) = res(1) + X(m+1) / sqrt(N);
    end;

    % Compute all other coefficients
    for k=1:N-1
        for m=0:N-1
            res(k+1) = res(k+1) + X(m+1)*cos( k*(2*m+1)*pi/(2*N) ) *
sqrt(2/N);
        end;
    end;
end
```

Inverse DCT, type II:

```
function res = inv_orthog_DCT2(N,X)
    % INV_ORTHOG_DCT2(N,X)
    % Computes the inverse of the orthogonal DCT-2 on input signal X

    res = zeros(N,1);

    for k=0:N-1
        res(k+1) = res(k+1) + X(1) / sqrt(N);
        for m=1:N-1
            res(k+1) = res(k+1) + X(m+1)*cos( m*(2*k+1)*pi/(2*N) ) *
sqrt(2/N);
        end;
    end;
end
```

DCT, type IV:

```
function res = orthog_DCT4(N,X)
    % ORTHOG_DCT4(N,X)
    % Computes the orthogonal DCT-4 on input signal X

    res = zeros(N,1);

    for k=1:N-1
        for m=0:N-1
```

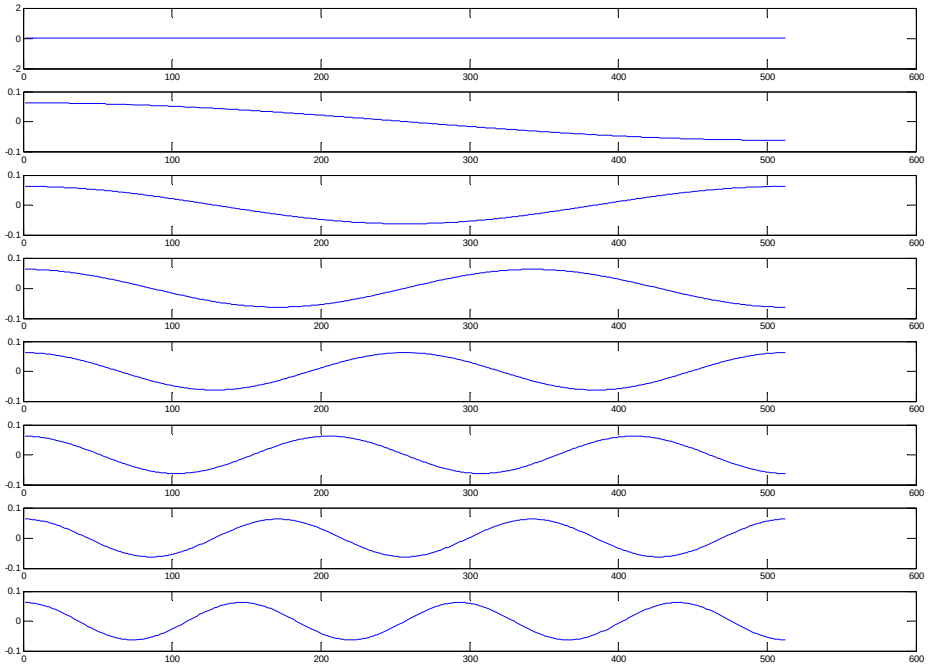
```
            res(k+1) = res(k+1) + X(m+1)*cos( (2*k+1)*(2*m+1)*pi/(4*N)
) * sqrt(2/N);
        end;
    end
end
```

Inverse DCT, type IV:

```
function res = inv_orthog_DCT4(N,X)
    % INV_ORTHOG_DCT4(N,X)
    % Computes the inverse of orthogonal DCT-4 on input signal X

    res = zeros(N,1);

    for k=0:N-1
        for m=0:N-1
            res(k+1) = res(k+1) + X(m+1)*cos( (2*k+1)*(2*m+1)*pi/(4*N)
) * sqrt(2/N);
        end;
    end
end
```

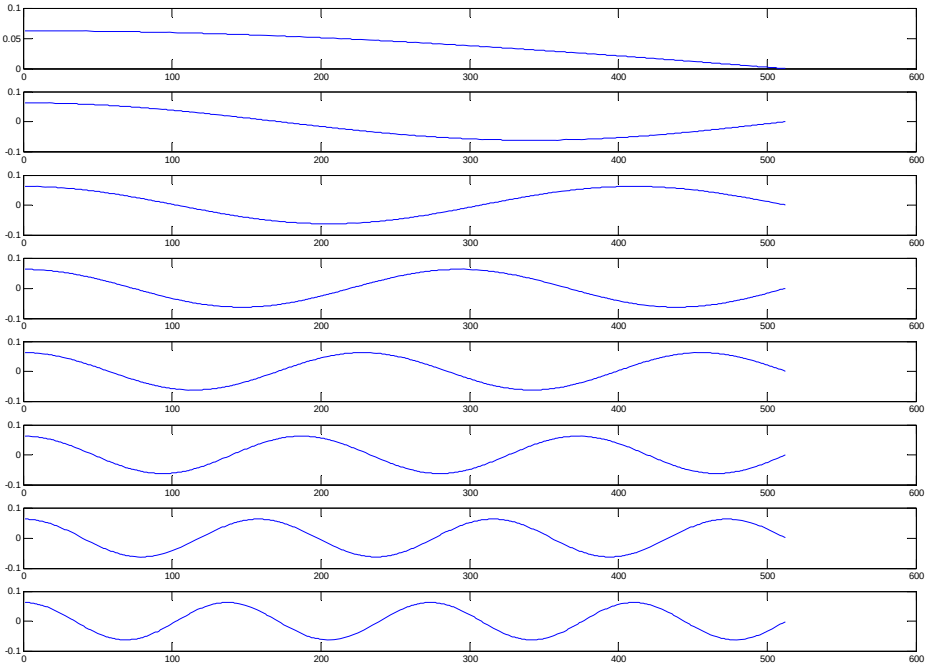
Eight lowest pure frequencies for DCT-2:

```
N = 512 ;
figure ;

 for k = 1:8
    e_k = zeros(N,1);
    e_k(k) = 1;
    f_k = inv_orthog_DCT2(N,e_k) ;
    subplot(8,1,k);
    plot(1:512, f_k);
    hold on;
end;
```



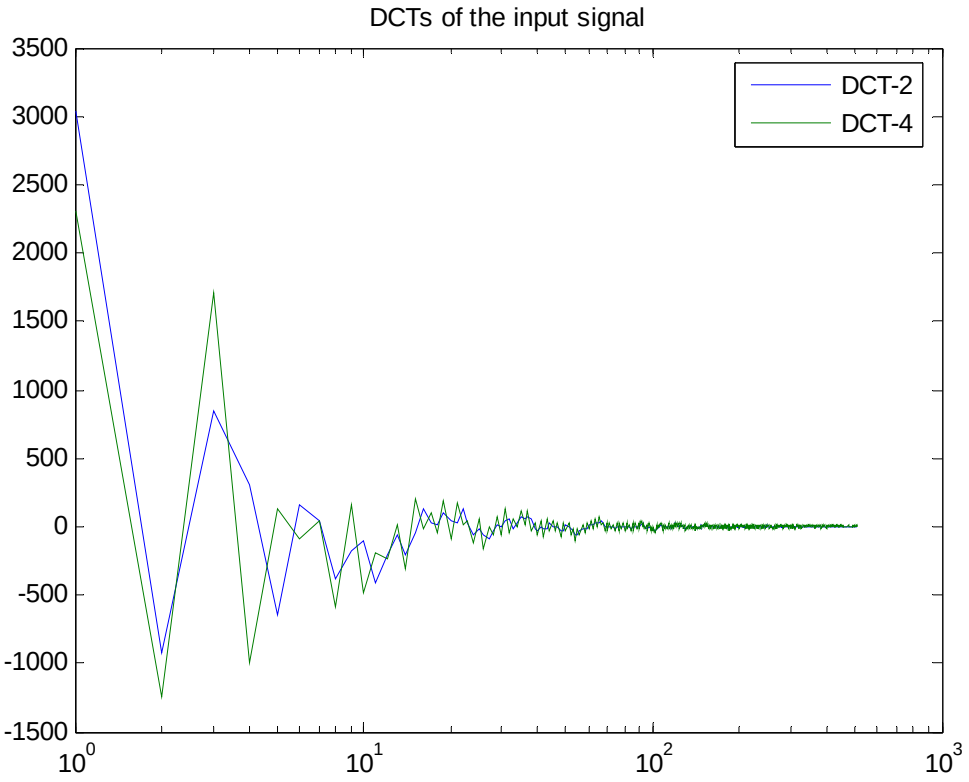
Eight lowest pure frequencies for DCT-4:

```
N = 512 ;
figure ;

for k = 1:8
    e_k = zeros(N,1);
    e_k(k) = 1;
    f_k = inv_orthog_DCT4(N,e_k) ;
    subplot(8,1,k);
    plot(1:512, f_k);
    hold on;
end;
```



(d)

```
N = 512 ;
load('image512.mat');

y2 = orthog_DCT2(N, double(hw_image));
y4 = orthog_DCT4(N, double(hw_image));
figure;
semilogx(1:N, y2, 1:N, y4);
legend('DCT-2', 'DCT-4');
title('DCTs of the input signal');

norm(y2)^2
norm(y4)^2
```



Energy of y2 is 11935840
Energy of y4 is 11935840

Since we apply an orthogonal transform to the input signal, its energy remains unchanged.

To find the corresponding signal models, we apply theorem 17 (p. 41): it follows that we only need to adjust the C-transform Φ according to the values of β_2 and β_3.

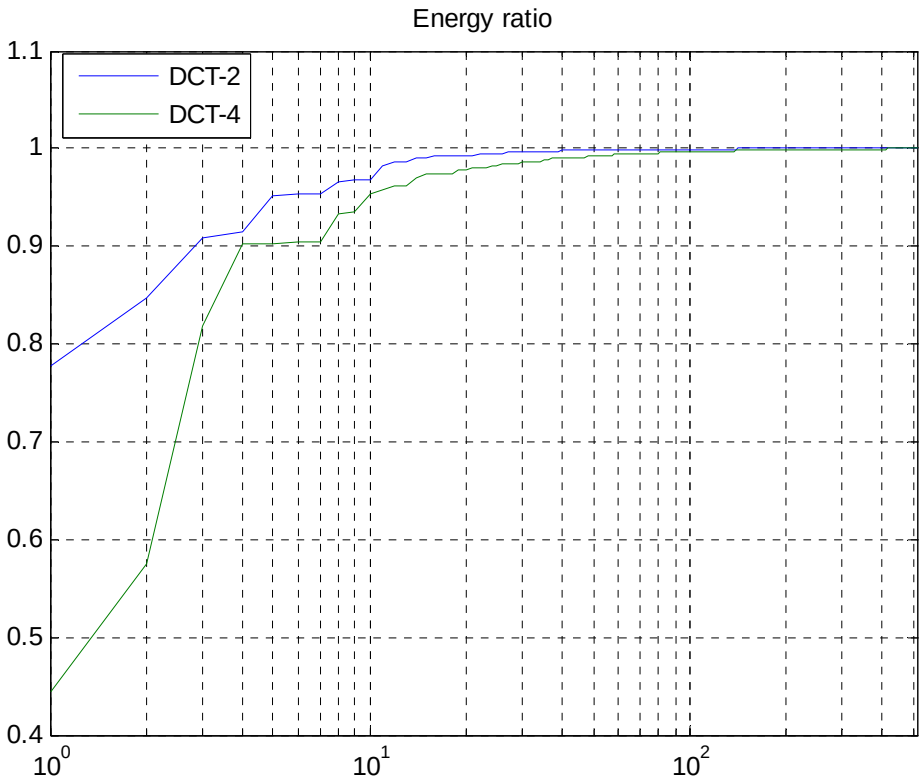
Orthogonal DCT-2:
$A = M = C[x]/\ 2(x-1)U_{n-1}(x)$
Since $\beta_2 = \beta_3 = 1$, we don't need to adjust the V-transform $\Phi = \Sigma s_k V_k$.

Orthogonal DCT-4:
$A = M = C[x]/\ 2T_n(x)$
Since $\beta_2 = \beta_3 = 1$, we don't need to adjust the V-transform $\Phi\ \Sigma s_k V_k$.

(e)

	$R_{t,k} \geq 0.5$	$R_{t,k} \geq 0.9$	$R_{t,k} \geq 0.99$
DCT-2	1	3	14
DCT-4	2	4	40

```
N = 512 ;
load('image512.mat');

y2 = orthog_DCT2(N, double(hw_image));
y4 = orthog_DCT4(N, double(hw_image));

R_2 = zeros(N,1);
R_4 = zeros(N,1);

for k = 1:N
    R_2(k) = norm(y2(1:k))^2/norm(y2)^2;
    R_4(k) = norm(y4(1:k))^2/norm(y4)^2;
end;

figure;
semilogx(1:N, R_2, 1:N, R_4);
legend('DCT-2', 'DCT-4');
title('Energy ratio');
grid on;
axis([0 N 0.4 1.1]);
```


Energy ratio
DCT-2
DCT-4
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
10^0
10^1
10^2