**Basic Problems**

16. (a) Proof:

$$x[n] = \frac{1}{N}\sum_{n=0}^{N-1} X[k]W_N^{-kn}, \quad n = 0, 1, \ldots, N-1 \tag{8.2}$$

$$\begin{aligned} x[n] &= \frac{1}{N}\sum_{n=0}^{N-1} X[k]W_N^{-kn} = \frac{1}{N}\mathrm{j}\left(\sum_{n=0}^{N-1}(-\mathrm{j}X[k])W_N^{-kn}\right) \\ &= \frac{1}{N}\mathrm{j}\left\{\sum_{k=0}^{N-1}(\mathrm{j}X^*[k])W_N^{kn}\right\}^* \end{aligned}$$

(b) tba.

(c) MATLAB function:

```
function x = idft_0816(X,N)
% Compute idft using fft function according to (8.90)
X = X(:).';
Nx = length(X);
if nargin == 1
    N = Nx;
elseif N <= Nx
    X = X(1:N);
else
    X = [X zeros(1,N-Nx)];
end
x = fft(conj(X)*j);
x = conj(x)*j/N;
```

17. Solution:
Direct computation:

$$(a + \mathrm{j}b)(c + \mathrm{j}d) = ac + \mathrm{j}bc + \mathrm{j}ad - bd = (ac - bd) + \mathrm{j}(bc + ad)$$

which contains four real multiplications and two real additions.
If we define

$$\begin{aligned} k_1 &= c \cdot (a + b) \\ k_2 &= a \cdot (d - c) \\ k_3 &= b \cdot (c + d) \end{aligned}$$

Hence, we can conclude that

$$ac - bd = k_1 - k_3, \qquad bc + ad = k_1 + k_2$$

which contains 3 real multiplications and 5 real additions.

18. Solution:
The resulting trend in the computational complexity of recursive DFT computations is much more efficient and close to linear than direct computation.

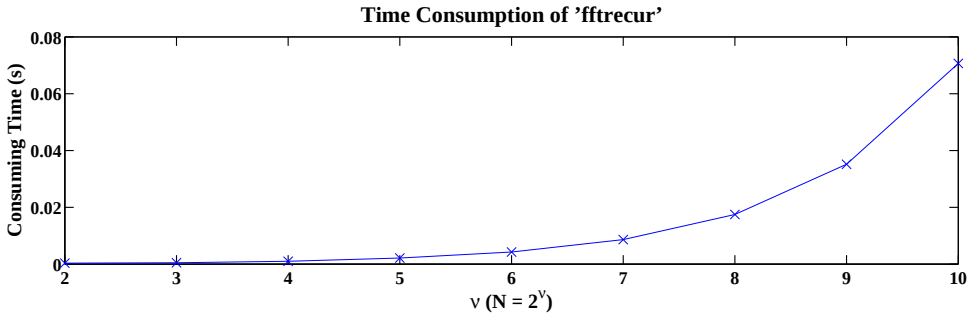


FIGURE 8.2: Plot of computation time for the `fftrecur` function for $N = 2^{\nu}$ where $2 \leq \nu \leq 10$.

MATLAB script:

```
% P0818: Investigate time consumption using fftrecur
close all; clc
nu = 2:10;
N = 2.^nu;
Ni = length(N);
t = zeros(1,Ni);
for ii = 1:Ni
    x = randn(1,N(ii)) + j*randn(1,N(ii));
    tic
    X = fftrecur(x);
    t(ii) = toc;
end
% Plot:
hfa = figconfg('P0818a','long');
plot(nu,t,'x-','markersize',12)
xlabel('\nu (N = 2^{\nu})','fontsize',LFS)
```

21. (a) Proof:

$$\begin{aligned}
y[Ln] &= \frac{1}{LN}\sum_{k=0}^{LN-1} Y[k]W_{LN}^{-k(LN)} \\
&= \frac{1}{LN}\left(\sum_{k=0}^{k_0-1} X[k]W_{LN}^{-k(Ln)} + \sum_{k=LN-k_0+1}^{LN-1} X[k+N-LN]W_{LN}^{-k(Ln)}\right) \\
&= \frac{1}{LN}\left(\sum_{k=0}^{k_0-1} X[k]W_{N}^{-kn} + \sum_{k=LN-k_0+1}^{LN-1} X[k+N-LN]W_{N}^{-kn}\right) \\
&= \frac{1}{LN}\left(\sum_{k=0}^{k_0-1} X[k]W_{N}^{-kn} + \sum_{k=N-k_0+1}^{N-1} X[k]W_{N}^{-kn}\right) \\
&= \frac{1}{L}\left(\frac{1}{N}\sum_{k=0}^{N-1} X[k]W_{N}^{-kn}\right) \\
&= \frac{1}{L}x[n]
\end{aligned}$$

(b) Solution:

$$y[k] = [2,0,0,0,0,0,0,2]$$

$$\begin{aligned}
y[n] &= \frac{1}{8}\sum_{k=0}^{7} Y[k]W_8^{nk} = \frac{1}{8}(2W_8^0 + 2W_8^{7n}) \\
&= \frac{1}{4}(1+W_8^{7n})
\end{aligned}$$

22. Solution:

$$\begin{cases} a[n] = x[2n], & n = 0,1,\ldots,\frac{N}{2}-1 \\ b[n] = x[2n+1], & n = 0,1,\ldots,\frac{N}{2}-1 \end{cases} \tag{8.19}$$

$$\begin{cases} X[k] = A[k] + W_N^k B[k], & n = 0,1,\ldots,\frac{N}{2}-1 \\ X[k+N/2] = A[k] - W_N^k B[k], & n = 0,1,\ldots,\frac{N}{2}-1 \end{cases} \tag{8.23}$$

If we mistakenly assign $a[n] = x[2n+1]$ and $b[n] = x[2n]$, we can recover the DFT $X[k]$ as:

$$\begin{aligned}
X'[k] &= \frac{X[k] - X[k+\frac{N}{2}]}{2}W_N^{-k} + \frac{X[k] + X[k+\frac{N}{2}]}{2}W_N^k \\
X'[k+\frac{N}{2}] &= \frac{X[k] - X[k+\frac{N}{2}]}{2}W_N^{-k} - \frac{X[k] + X[k+\frac{N}{2}]}{2}W_N^k
\end{aligned}$$

(b) MATLAB function:

```
function X = fftalt8(x)
if length(x)~=8
    error('bad input, illegal length')
end
N = 8;
s = x;
w = exp(-j*2*pi/N).^(0:N-1);
% Stage I:
temp = s;
s(1:4) = temp(1:4)+temp(5:8);
s(5:8) = temp(1:4)-temp(5:8);
% Stage II:
temp = s;
s(1:2) = temp(1:2)+temp(3:4);
s(3:4) = temp(5:6)+temp(7:8)*w(3);
s(5:6) = temp(1:2)-temp(3:4);
s(7:8) = temp(5:6)-temp(7:8)*w(3);
% Stage III:
temp = s;
s(1:4) = temp(1:2:end)+temp(2:2:end).*w(1:4);
s(5:8) = temp(1:2:end)-temp(2:2:end).*w(1:4);

X = s;
```

(c) Solution:
The coding complexity of the above function is much larger than that of the `fftditr2` function since the equations are not recursive.

27. Proof:

$$\begin{cases} X[k] = A[k] + W_N^k B[k] \\ X[k+\frac{N}{2}] = A[k] - W_N^k B[k] \end{cases} \quad k = 0, 1, \ldots, \frac{N}{2} - 1. \tag{8.23}$$

$$X_{n_2}[k_1] = \sum_{n_1=0}^{N_1-1} x_{n_2}[n_1] W_{N_1}^{k_1 n_1} \tag{8.52}$$

$$x_{k_1}[n_2] \triangleq W_N^{k_1 n_2} X_{n_2}[k_1] \tag{8.54}$$

$$X[k_1 + N_1 k_2] = \sum_{n_2=0}^{N_2-1} x_{k_1}[n_2] W_{N_2}^{k_2 n_2} \tag{8.55}$$

If we have $N_1 = \frac{N}{2}$, and $N_2 = 2$, thus (8.55) can be written as

$$\begin{cases} X[k_1] = x_{k_1}[0] + x_{k_1}[1] \\ X[k_1 + \frac{N}{2}] = x_{k_1}[0] - x_{k_1}[1] \end{cases}$$

Then, (8.54) can be written as

$$\begin{cases} x_{k_1}[0] = X_0[k_1] \\ x_{k_1}[1] = W_N^{k_1} X_1[k_1] \end{cases} \qquad k_1 = 0, 1, \ldots, \frac{N}{2}$$

Hence, (8.52) can be written as

$$X_0[k_1] = \sum_{n_1=0}^{\frac{N}{2}-1} x_0[n_1] W_{\frac{N}{2}}^{k_1 n_1} = \sum_{n_1=0}^{\frac{N}{2}-1} x[2n_1] W_{\frac{N}{2}}^{k_1 n_1}$$
$$X_1[k_1] = \sum_{n_1=0}^{\frac{N}{2}-1} x_1[n_1] W_{\frac{N}{2}}^{k_1 n_1} = \sum_{n_1=0}^{\frac{N}{2}-1} x[2n_1+1] W_{\frac{N}{2}}^{k_1 n_1}$$

which is the same as the DIT-FFT algorithm of (8.23).

28. (a) Solution:

$$X[5k] = \sum_{n=0}^{2} \left( x[n] + x[n+3] + x[n+6] + x[n+9] + x[n+12] \right) W_3^{nk}$$
$$X[5k+1] = \sum_{n=0}^{2} \big( x[n] + x[n+3] W_{15}^{3} + x[n+6] W_{15}^{6} + x[n+9] W_{15}^{9} + x[n+12] W_{15}^{12} \big) W_{15}^{n} W_3^{nk}$$
$$X[5k+2] = \sum_{n=0}^{2} \big( x[n] + x[n+3] W_{15}^{6} + x[n+6] W_{15}^{12} + x[n+9] W_{15}^{3} + x[n+12] W_{15}^{9} \big) W_{15}^{2n} W_3^{nk}$$
$$X[5k+3] = \sum_{n=0}^{2} \big( x[n] + x[n+3] W_{15}^{9} + x[n+6] W_{15}^{3} + x[n+9] W_{15}^{12} + x[n+12] W_{15}^{6} \big) W_{15}^{3n} W_3^{nk}$$
$$X[5k+4] = \sum_{n=0}^{2} \big( x[n] + x[n+3] W_{15}^{12} + x[n+6] W_{15}^{9} + x[n+9] W_{15}^{6} + x[n+12] W_{15}^{3} \big) W_{15}^{4n} W_3^{nk}$$

```
K = max(M,N); n = 0:K; Wn2 = W.^(n.*n/2);
g = x.*exp(-1j*wL*nx).*Wn2(1:N);
nh = -(N-1):M-1; h = W.^(-nh.*nh/2);
L = M + N;
G = fft(g,L);
H = fft(h,L);
Y = G.*H;
y = ifft(Y);
X = y(N:N+M-1).*Wn2(1:M); w = wL:dw:wH;
```

32. (a) See plot below.

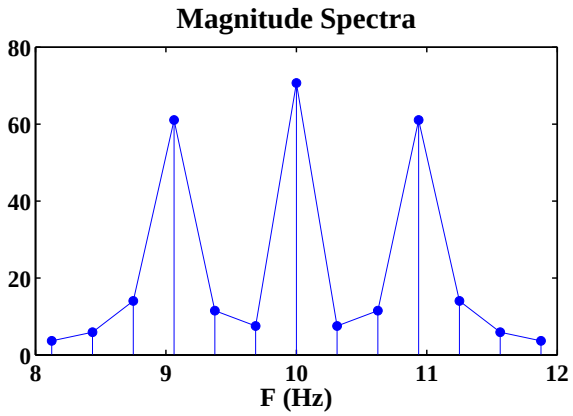


FIGURE 8.3: Magnitude spectra of 128-point FFT of $x[n]$ over $8 \leq F \leq 12$ Hz.

(b) See plot below.

(c) See plot below.

(d) Solution:
Using `cta` function has the smallest number of computations with a better display.

MATLAB script:

```
% P0832: Illustration of using cta algorithm
close all; clc
Fs = 40;
n = 0:127;
T = 1/Fs;
nT = n*T;
```

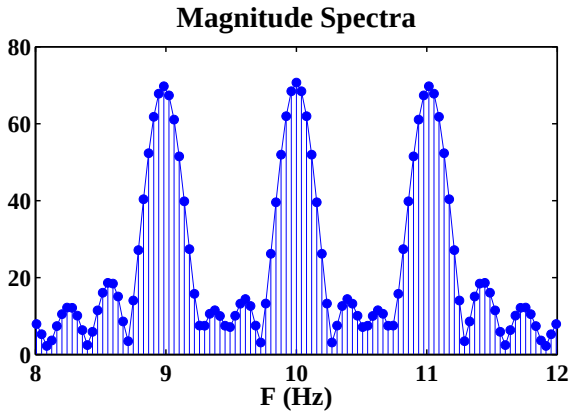


FIGURE 8.4: Magnitude spectra of 1024-point FFT of $x[n]$ over $8 \leq F \leq 12$ Hz.

```
dt = 0.01;
t = 0:dt:n(end)*T;
xc = cos(20*pi*t) + cos(18*pi*t) + cos(22*pi*t);
xn = cos(20*pi*nT) + cos(18*pi*nT) + cos(22*pi*nT);
L = 128; % Part (a)
% L = 1024; % Part (b)
k = 0:L-1;
X = fft(xn,L);
Fp = Fs*k/L;
indk = Fp >= 8 & Fp <= 12;
kp = k(indk);
%% Part (c):
[Xcta,w] = cta(xn,length(kp),8*2*pi*T,12*2*pi*T);
%% Plot:
hfa = figconfg('P0832a','small');
plot(Fp(indk),abs(X(indk))); hold on
stem(Fp(indk),abs(X(indk)),'filled')
xlim([8 12])
xlabel('F (Hz)','fontsize',LFS)
title('Magnitude Spectra','fontsize',TFS)

hfb = figconfg('P0832b','small');
plot(w*Fs/2/pi,abs(Xcta)); hold on
stem(w*Fs/2/pi,abs(Xcta),'filled')
```

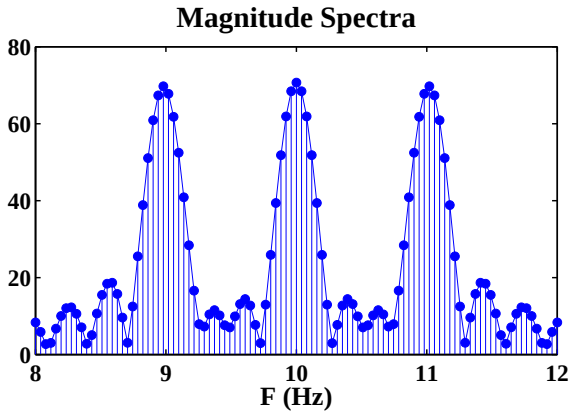


FIGURE 8.5: Magnitude spectra of DFT of $x[n]$ over $8 \leq F \leq 12$ Hz using `cta` function.

```
xlim([8 12])
xlabel('F (Hz)','fontsize',LFS)
title('Magnitude Spectra','fontsize',TFS)
```