

Below are **clean, fully-organised MATLAB codes ONLY**, with **clear question numbers**, for all **Exercise Problems** from **Week-10 (DFT / Fourier Series)** based on your PDF .

 ****EXERCISE 1 – Fourier Series Reconstruction (Use given Dr)****

```
``matlab
%% EXERCISE 1 – Fourier Series Expanded Signal (Using computed Dr)
No = 32;
n = 0:No-1;
xn = [ones(1,5) zeros(1,23) ones(1,4)];

% Compute Fourier Coefficients Dr
for r = 0:No-1
    Dr(r+1) = sum(xn .* exp(-1j*r*(2*pi/No)*n))/No;
end

% Reconstruct signal using full sum
x_rec = zeros(1,No);
for r = 0:No-1
    x_rec = x_rec + Dr(r+1) * exp(1j*r*(2*pi/No)*n);
end

% Plot reconstructed signal
figure;
subplot(3,1,1); stem(n,real(x_rec),'filled'); title('Reconstructed x[n]');

% Plot at least 5 individual FS terms
figure;
for k = 1:5
    subplot(5,1,k);
    term = Dr(k+1) * exp(1j*k*(2*pi/No)*n);
    stem(n,real(term),'filled');
    title(['FS Term r = ' num2str(k)]);
end
``
```

 ****EXERCISE 2 – Reconstruct Signal from Given Dr Table****

Given coefficients from PDF ($r = -5$ to $+5$):

```
``matlab
%% EXERCISE 2 – Reconstruct signal from given Dr table
No = 32;
n = 0:No-1;

% Given Dr for r = -5 to +5
Dr_small = [...
    -1j*0.8720 -1j*0.4330 +1j*0.3330 +1j*0.1440 -1j*0.2950 ...
    0 ...
    +1j*0.2950 -1j*0.1440 -1j*0.3330 +1j*0.4330 +1j*0.8720];

r_vals = -5:5;

% Reconstruct
x_rec = zeros(1,No);
```

```

for k = 1:length(r_vals)
    r = r_vals(k);
    x_rec = x_rec + Dr_small(k)*exp(1j*r*(2*pi/No)*n);
end

```

```

figure; stem(n,real(x_rec),'filled');
title('Reconstructed Signal from Given Dr');

```

```

% Print one full period values
disp('x[n] for one full period ='); disp(real(x_rec));
```

```

---

#  \*\*EXERCISE 3 – Verify Periodicity of Signals\*\*

```

```matlab
%% EXERCISE 3 – Check periodicity by plots

```

```

n = 0:200;

```

```

x1 = 3*sin(pi*n) + 2*sin(3*n);
x2 = 2 + 5*sin(4*pi*n) + 4*cos(7*pi*n);
x3 = 2*sin(3*n) + 7*cos(pi*n);
x4 = 7*cos(pi*n) + 5*sin(2*pi*n);
x5 = 3*cos((2*pi/2)*n) + 5*cos(2*pi*n);
x6 = (15/4)*sin(3*n) + cos(n);
x7 = sin((5*pi/2)*n) + (6/5)*cos(3*n) + 3*sin((pi/7)*n + pi/6);

```

```

figure;
subplot(4,2,1); stem(n,x1); title('x1');
subplot(4,2,2); stem(n,x2); title('x2');
subplot(4,2,3); stem(n,x3); title('x3');
subplot(4,2,4); stem(n,x4); title('x4');
subplot(4,2,5); stem(n,x5); title('x5');
subplot(4,2,6); stem(n,x6); title('x6');
subplot(4,2,7); stem(n,x7); title('x7');
```

```

---

#  \*\*EXERCISE 4 – DTFS of Given Periodic Signals\*\*

```

(a) **x[n] = 4cos(2.4πn) + 2sin(3.2πn)**

```

```

```matlab
%% EXERCISE 4(a)
No = 32; n = 0:No-1;
x = 4*cos(2.4*pi*n) + 2*sin(3.2*pi*n);

```

```

for r = 0:No-1
    Dr(r+1) = sum(x .* exp(-1j*r*(2*pi/No)*n))/No;
end

```

```

figure;
stem(0:31,abs(Dr)); title('|Dr|');
figure;
stem(0:31,angle(Dr)); title('Phase Dr');
```

```

---

### (b)  $x[n] = \cos(2.2\pi n) + \cos(3.3\pi n)$

```
``matlab
%% EXERCISE 4(b)
No = 32; n = 0:No-1;
x = cos(2.2*pi*n) + cos(3.3*pi*n);

for r = 0:No-1
 Dr(r+1) = sum(x .* exp(-1j*r*(2*pi/No)*n))/No;
end

figure; stem(abs(Dr)); title('|Dr|');
figure; stem(angle(Dr)); title('Phase Dr');
```

---

### (c)  $x[n] = 2\cos(3.2\pi(n-3))$

```
``matlab
%% EXERCISE 4(c)
No = 32; n = 0:No-1;
x = 2*cos(3.2*pi*(n-3));

for r = 0:No-1
 Dr(r+1) = sum(x .* exp(-1j*r*(2*pi/No)*n))/No;
end

figure; stem(abs(Dr)); title('|Dr|');
figure; stem(angle(Dr)); title('Phase Dr');
```

---

### (d)  $x[n] = a^n, 0 \leq n \leq 4, a = 0.707, No = 5$

```
``matlab
%% EXERCISE 4(d)
a = 0.707; No = 5; n = 0:No-1;
x = a.^n;

for r = 0:No-1
 Dr(r+1) = sum(x .* exp(-1j*r*(2*pi/No)*n))/No;
end

figure; stem(abs(Dr)); title('|Dr|');
figure; stem(angle(Dr)); title('Phase Dr');
```

---

#  \*\*EXERCISES – Fourier Transform Section\*\*

---

# EXERCISE 1 – Reconstruct  $x[n]$  from FT

```

```matlab
%% FT EXERCISE 1 – Reconstruct x[n]
n = 0:4;
x = 0.7071.^n;

h = 0.01;
Omega = -pi:h:pi;

% Compute X(e^{j\Omega})
for k = 1:length(Omega)
    X(k) = sum(x .* exp(-1j*Omega(k)*n));
end

% Inverse FT
x_rec = zeros(1,5);
for m = 1:5
    x_rec(m) = (h/(2*pi))*sum( X .* exp(1j*Omega*(m-1)) );
end

disp('Reconstructed x[n] ='); disp(real(x_rec))
```

```

---

# EXERCISE 2 – Zero-State Response using FT

```

```matlab
%% FT EXERCISE 2 – Zero-state response via FT
n = 0:4;
x = 0.7071.^n;

h = 0.01;
Omega = -pi:h:pi;

% Input spectrum
for k = 1:length(Omega)
    X(k) = sum(x .* exp(-1j*Omega(k)*n));
end

% System H(e^{j\Omega})
H = 1 ./ (exp(1j*Omega) - 0.0591);

Y = X .* H;

% Inverse FT
y = zeros(1,5);
for m = 1:5
    y(m) = (h/(2*pi))*sum( Y .* exp(1j*Omega*(m-1)) );
end

figure; stem(real(y)); title('Zero-State Response y[n]');
```

```

---

# EXERCISE 5 – FT of  $\gamma^n u[n]$ ,  $\gamma = -0.8$

```

```matlab
%% FT EXERCISE 5 – Fourier Transform of  $\gamma^n u[n]$ 
gamma = -0.8;

```

```

n = 0:200;
x = gamma.^n;

h = 0.01; Omega = -pi:h:pi;

for k = 1:length(Omega)
    X(k) = sum(x .* exp(-1j*Omega(k)*n));
end

figure; plot(Omega,abs(X)); title('|X(e^{j\Omega})|');
figure; plot(Omega,angle(X)); title('Phase of X(e^{j\Omega})');

```

If you want, I can also generate:

- ☒ Code for **P9-16, P9-22, P9-26, P9-27**
- ☒ Clean plots
- ☒ Combined master script

Just tell me ***"Give remaining exercise codes"***.

```

``matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% WEEK 10 – FOURIER / DFT : REMAINING EXERCISE CODES
% (From "Week10DFT.pdf") :contentReference[oaicite:0]{index=0}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% -----
%% FT EXERCISE 3 – PROPERTY EFFECTS ON FOURIER SPECTRUM
%% (Example with a base rectangular energy signal)
%% -----

clear; close all; clc;

% ----- Base signal: simple rectangular pulse x[n] -----
n = -10:10;
x = double((n >= -3) & (n <= 3)); % rectangle of width 7, amplitude 1

% Frequency grid
h = 0.01;
Omega = -pi:h:pi;

% FT of base signal
for k = 1:length(Omega)
    X(k) = sum(x .* exp(-1j*Omega(k)*n));
end

% ----- (a) Scalar addition: y = x + c -----
c = 2;
yA = x + c;
for k = 1:length(Omega)
    YA(k) = sum(yA .* exp(-1j*Omega(k)*n));
end

% ----- (b) Scalar multiplication: y = a*x -----
a = 3;
yB = a*x;
for k = 1:length(Omega)
    YB(k) = sum(yB .* exp(-1j*Omega(k)*n));
end

% ----- (c) Conjugation: y = conj(x) -----
yC = conj(x);
for k = 1:length(Omega)
    YC(k) = sum(yC .* exp(-1j*Omega(k)*n));
end

% ----- (d) Addition of two signals: y = x + z -----
z = double((n >= 0) & (n <= 6)); % another rectangle for example
yD = x + z;
for k = 1:length(Omega)
    YD(k) = sum(yD .* exp(-1j*Omega(k)*n));
end

% ----- (e) Time shifting: y[n] = x[n - n0] -----
n0 = 3;
yE = double(((n-n0) >= -3) & ((n-n0) <= 3));
for k = 1:length(Omega)
    YE(k) = sum(yE .* exp(-1j*Omega(k)*n));
end

```

```

end

% ----- (f) Time reversal (example of scaling):  $y[n] = x[-n]$  -----
yF = fliplr(x);
for k = 1:length(Omega)
    YF(k) = sum(yF .* exp(-1j*Omega(k)*n));
end

% ----- (g) Frequency shifting: multiply by  $e^{j\Omega_0 n}$  -----
Omega0 = pi/4;
yG = x .* exp(1j*Omega0*n);
for k = 1:length(Omega)
    YG(k) = sum(yG .* exp(-1j*Omega(k)*n));
end

% ----- (h) Time convolution:  $y = x (*) z$  -----
yH = conv(x,z);
nH = (n(1)+n(1)):(n(end)+n(end));
for k = 1:length(Omega)
    YH(k) = sum(yH .* exp(-1j*Omega(k)*nH));
end

% ----- (i) Frequency convolution (via product in time) -----
% Here we just show example: product in time => convolution in frequency.
w = double((n >= -1) & (n <= 1)); % small window
yl = x .* w;
for k = 1:length(Omega)
    Yl(k) = sum(yl .* exp(-1j*Omega(k)*n));
end

% Example plot: base  $|X|$  and some modified ones
figure;
subplot(3,1,1); plot(Omega,abs(X)); title('|X( $\Omega$ )| base');
subplot(3,1,2); plot(Omega,abs(YA)); title('|Y_A( $\Omega$ )| scalar add');
subplot(3,1,3); plot(Omega,abs(YE)); title('|Y_E( $\Omega$ )| time shift');

%% -----
%% FT EXERCISE 4 – INVERSE FT OF GIVEN SPECTRA (GENERIC TEMPLATE)
%% P9-15 (a) and (b): YOU MUST DEFINE  $X(\Omega)$  AS PER YOUR FIGURE
%% -----

clear; close all; clc;

% Frequency grid
h = 0.01;
Omega = -pi:h:pi;
Nrec = 41; % length of recovered sequence (example n = -20..20)
n = -20:20;

% ----- CASE (a) : define  $X_a(\Omega)$  from P9-15(a) -----
Xa = zeros(size(Omega));

% >>> FILL THIS DEFINITION from your textbook figure P9-15(a) <<<
% Example shape (RECTANGULAR, JUST A PLACEHOLDER!):
%  $X_a(\text{abs}(\Omega) \leq \pi/3) = 1$ ;

% Inverse FT for case (a)
xa = zeros(size(n));
for m = 1:length(n)

```

```

    xa(m) = (h/(2*pi))*sum( Xa .* exp(1j*Omega*n(m)) );
end

```

```

figure;
stem(n,real(xa),'filled');
title('x_a[n] from P9-15(a) (template)');

```

```

% ----- CASE (b) : define X_b(Ω) from P9-15(b) -----
Xb = zeros(size(Omega));

```

```

% >>> FILL THIS DEFINITION from your textbook figure P9-15(b) <<<
% Example placeholder:
% Xb(abs(Omega) <= pi/4) = (abs(Omega(abs(Omega)<=pi/4))/(pi/4));

```

```

xb = zeros(size(n));
for m = 1:length(n)
    xb(m) = (h/(2*pi))*sum( Xb .* exp(1j*Omega*n(m)) );
end

```

```

figure;
stem(n,real(xb),'filled');
title('x_b[n] from P9-15(b) (template)');

```

```

%% -----
%% FT EXERCISE 6 – FIRST-PRINCIPLE FT OF P9-16, P9-22, P9-26, P9-27
%% (You only need to EDIT the definitions of x_n for each figure)
%% -----

```

```

clear; close all; clc;
h = 0.01;
Omega = -pi:h:pi;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% P9-16 (a) -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Define n-range large enough to cover all non-zero samples shown
n = -10:10;
x_161 = zeros(size(n));

```

```

% >>> FILL x_161 according to Fig P9-16(a) <<<
% Example pattern (PLACEHOLDER ONLY):
% x_161(n==0) = 3;
% x_161(n==1) = 2; x_161(n==2) = 2; etc...

```

```

% FT via first principle
for k = 1:length(Omega)
    X_161(k) = sum(x_161 .* exp(-1j*Omega(k)*n));
end

```

```

% Verify by inverse FT
x_161_rec = zeros(size(n));
for m = 1:length(n)
    x_161_rec(m) = (h/(2*pi))*sum( X_161 .* exp(1j*Omega*n(m)) );
end

```

```

figure;

```

```
subplot(2,1,1); stem(n,x_161,'filled'); title('P9-16(a) x[n] (manual)');
subplot(2,1,2); stem(n,real(x_161_rec),'o'); title('Reconstructed x[n]');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ----- P9-16 (b), (c), (d) – similar pattern -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Copy, rename as x_162, x_163, x_164, and fill from your figure
% Example template for (b):
```

```
n = -10:10;
x_162 = zeros(size(n));

% >>> FILL x_162 from P9-16(b) <<<

for k = 1:length(Omega)
    X_162(k) = sum(x_162 .* exp(-1j*Omega(k)*n));
end

x_162_rec = zeros(size(n));
for m = 1:length(n)
    x_162_rec(m) = (h/(2*pi))*sum( X_162 .* exp(1j*Omega*n(m)) );
end

figure;
subplot(2,1,1); stem(n,x_162,'filled'); title('P9-16(b) x[n]');
subplot(2,1,2); stem(n,real(x_162_rec),'o'); title('Reconstructed x[n]');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ----- P9-22 (a)–(c) : same idea, different x[n] ranges -----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Example template for P9-22(c) (wide trapezoid in the figure):
```

```
n = -10:10;
x_922c = zeros(size(n));

% >>> FILL x_922c values from P9-22(c) figure <<<

for k = 1:length(Omega)
    X_922c(k) = sum(x_922c .* exp(-1j*Omega(k)*n));
end

x_922c_rec = zeros(size(n));
for m = 1:length(n)
    x_922c_rec(m) = (h/(2*pi))*sum( X_922c .* exp(1j*Omega*n(m)) );
end

figure;
subplot(2,1,1); stem(n,x_922c,'filled'); title('P9-22(c) x[n]');
subplot(2,1,2); stem(n,real(x_922c_rec),'o'); title('Reconstructed x[n]');
```

```

%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
% ----- P9-26 (a), (b) : impulses / pulses -----
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%

```

```

n = -20:20;
x_926a = zeros(size(n));

% >>> FILL x_926a from P9-26(a) <<<
% e.g. constant-height samples over some finite support

```

```

for k = 1:length(Omega)
    X_926a(k) = sum(x_926a .* exp(-1j*Omega(k)*n));
end

x_926a_rec = zeros(size(n));
for m = 1:length(n)
    x_926a_rec(m) = (h/(2*pi))*sum( X_926a .* exp(1j*Omega*n(m)) );
end

figure;
subplot(2,1,1); stem(n,x_926a,'filled'); title('P9-26(a) x[n]');
subplot(2,1,2); stem(n,real(x_926a_rec),'o'); title('Reconstructed x[n]');

```

```

%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
% ----- P9-27 : one-sided sequence (example) -----
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%

```

```

n = -10:10;
x_927 = zeros(size(n));

% >>> FILL x_927 from P9-27 figure <<<
% (Looks like one-sided non-zero region)

```

```

for k = 1:length(Omega)
    X_927(k) = sum(x_927 .* exp(-1j*Omega(k)*n));
end

x_927_rec = zeros(size(n));
for m = 1:length(n)
    x_927_rec(m) = (h/(2*pi))*sum( X_927 .* exp(1j*Omega*n(m)) );
end

figure;
subplot(2,1,1); stem(n,x_927,'filled'); title('P9-27 x[n]');
subplot(2,1,2); stem(n,real(x_927_rec),'o'); title('Reconstructed x[n]');
'''

```

These are all the **remaining exercise codes**; for P9-15, P9-16, P9-22, P9-26, P9-27 you just have to fill in the exact **$x[n]$** or **$X(\Omega)$** from your printed figures, the rest of the code is ready to run.