

Scilab Manual for  
Digital Signal Processing  
by Ms E. Sangeetha Devi  
Electronics Engineering  
Periyar Maniammai University<sup>1</sup>

Solutions provided by  
Ms E.SANGEETHA DEVI  
Electronics Engineering  
PERIYAR MANIAMMAI UNIVERSITY, THANAJAVUR

May 28, 2026

<sup>1</sup>Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website <http://scilab.in>



# Contents

List of Scilab Solutions	3
1 GENERATION OF CONTINUOUS SIGNALS	6
2 GENERATION OF DISCRETE SIGNALS	15
3 GENERATION OF SINUSOIDAL SIGNALS	21
4 GENERATION OF COMPLEX EXPONENTIAL SIGNALS	23
5 EXPONENTIALLY GROWING & DECAYING SIGNAL	26
6 ADDITION OF TWO CONTINUOUS SIGNALS	30
7 ADDITION OF TWO DISCRETE SIGNALS	32
8 PROGRAM FOR SAMPLING PROCESS	35
9 PROGRAM FOR ALIASING PROCESS	38
10 LINEAR CONVOLUTION OF TWO SEQUENCES	40
11 CIRCULAR CONVOLUTION OF TWO SEQUENCES	42
12 DFT OF A SIGNAL	45
13 FFT OF A SIGNAL	49

# List of Experiments

Solution 1.1	sinewave . . . . .	6
Solution 1.2	cosine wave . . . . .	6
Solution 1.3	triangular wave . . . . .	10
Solution 1.4	signum function . . . . .	10
Solution 1.5	sinc function . . . . .	13
Solution 1.6	Exponential wave . . . . .	13
Solution 2.1	unit impulse signal . . . . .	15
Solution 2.2	unitstepsignal . . . . .	15
Solution 2.3	discreteexponentialwave . . . . .	19
Solution 2.4	unit ramp . . . . .	19
Solution 3.1	Generation of sinusoidal signals . . . . .	21
Solution 4.1	generation of complex exponential signals . . . . .	23
Solution 5.1	Exponentially Growing signals . . . . .	26
Solution 5.2	Exponentially decaying signals . . . . .	27
Solution 6.1	Addition of two continuous signals . . . . .	30
Solution 7.1	Addition of two discrete signals . . . . .	32
Solution 8.1	sampling . . . . .	35
Solution 9.1	Aliasing process . . . . .	38
Solution 10.1	linear convolution of two sequences . . . . .	40
Solution 11.1	circular convolution . . . . .	42
Solution 12.1	DFT . . . . .	45
Solution 13.1	fast fourier transform . . . . .	49

# List of Figures

1.1	sinewave . . . . .	7
1.2	cosine wave . . . . .	8
1.3	triangular wave . . . . .	9
1.4	signum function . . . . .	11
1.5	sinc function . . . . .	12
1.6	Exponential wave . . . . .	14
2.1	unit impulse signal . . . . .	16
2.2	unitstepsignal . . . . .	17
2.3	discreteexponentialwave . . . . .	18
2.4	unit ramp . . . . .	20
3.1	Generation of sinusoidal signals . . . . .	22
4.1	generation of complex exponential signals . . . . .	24
5.1	Exponentially Growing signals . . . . .	27
5.2	Exponentially decaying signals . . . . .	28
6.1	Addition of two continuous signals . . . . .	31
7.1	Addition of two discrete signals . . . . .	33
8.1	sampling . . . . .	36
9.1	Aliasing process . . . . .	39
10.1	linear convolution of two sequences . . . . .	41
11.1	circular convolution . . . . .	43
12.1	DFT . . . . .	46

13.1 fast fourier transform . . . . .	50
---------------------------------------	----

# Experiment: 1

## GENERATION OF CONTINUOUS SIGNALS

Scilab code Solution 1.1 sinewave

```
1  clc;
2  clf;
3  clear all;
4  //Caption:generation of sine wave
5  f=0.2;
6  t=0:0.1:10;
7  x=sin(2*%pi*t*f);
8  plot(t,x);
9  title('sine wave');
10 xlabel('t');
11 ylabel('x');
```

---

Scilab code Solution 1.2 cosine wave

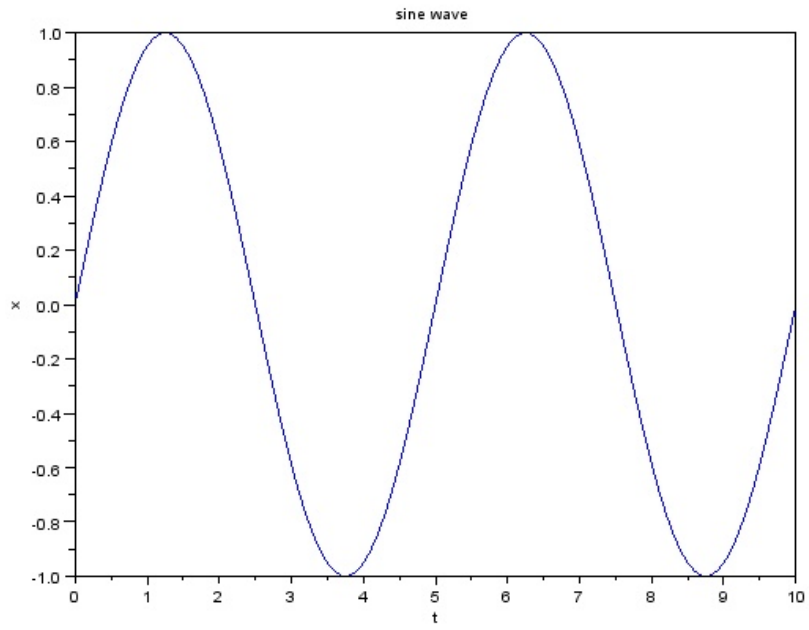


Figure 1.1: sinewave

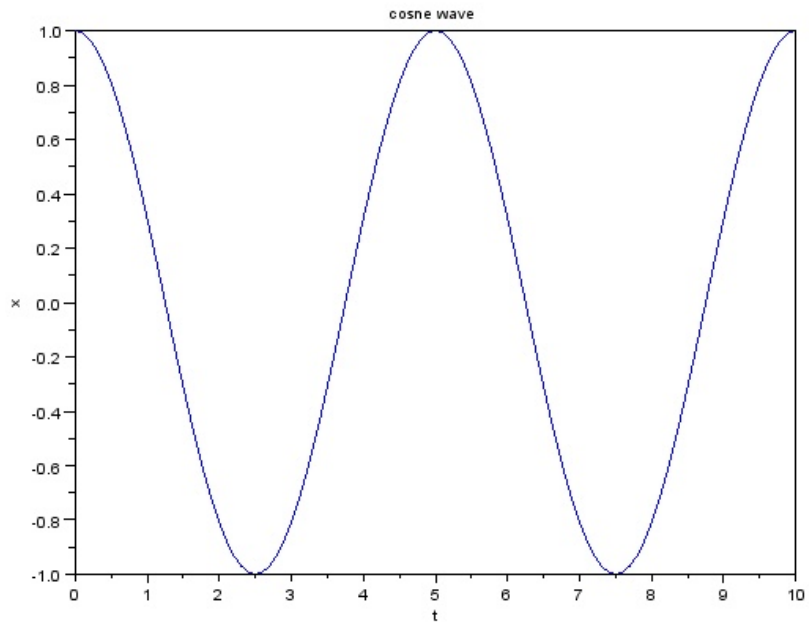


Figure 1.2: cosine wave

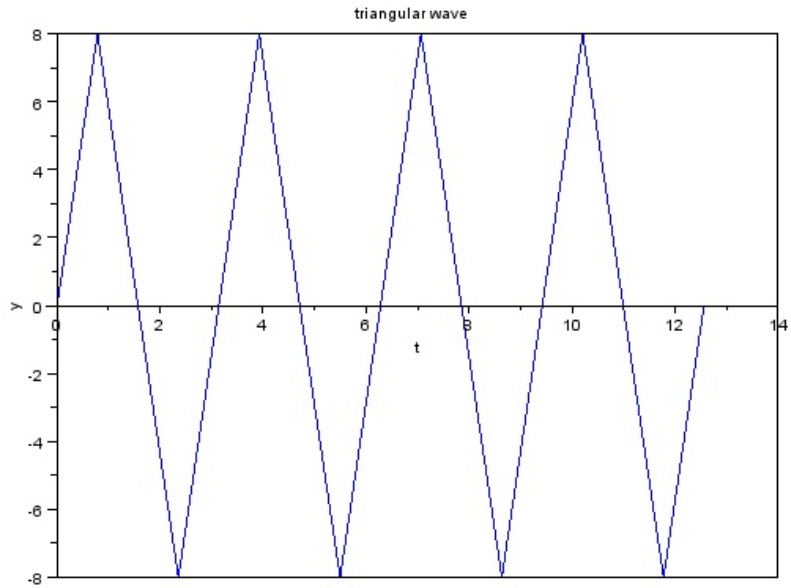


Figure 1.3: triangular wave

```

1  clc;
2  clf;
3  clear all;
4  //Caption:generation of cosine wave
5  f=0.2;
6  t=0:0.1:10;
7  x=cos(2*%pi*t*f);
8  plot(t,x);
9  title('cosne wave');
10 xlabel('t');
11 ylabel('x');

```

---

### Scilab code Solution 1.3 triangular wave

```
1 clc;
2 clf;
3 clear all;
4 //Caption:generation of triangular wave
5 a=8;
6 t=0:(%pi/4):(4*%pi);
7 y=a*sin(2*t);
8 a=gca();
9 a.x_location="middle"
10 plot(t,y);
11 title('triangular wave');
12 xlabel('t');
13 ylabel('y');
```

---

### Scilab code Solution 1.4 signum function

```
1 clc;
2 clf;
3 clear all;
4 //Caption:signum function
5 t =-5:0.1:5
6 a=gca();
7 a.x_location="middle"
8 x=sign(t);
9 b =gca();
10 b.y_location="middle"
11 plot(t,x);
12 title('signum function');
```

---

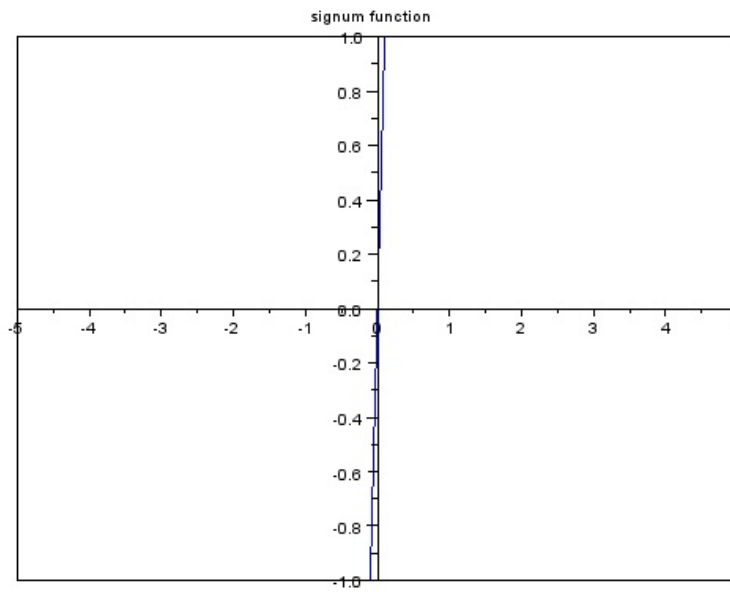


Figure 1.4: signum function

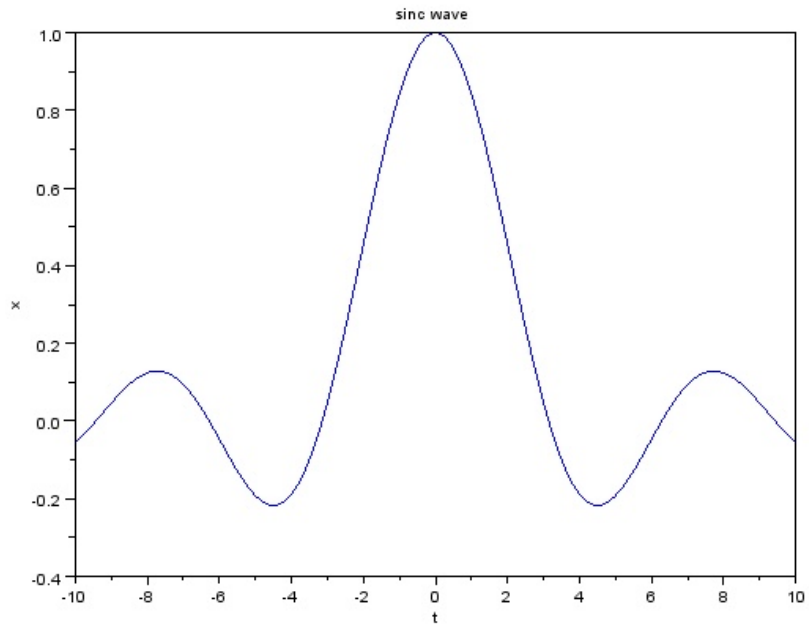


Figure 1.5: sinc function

### Scilab code Solution 1.5 sinc function

```
1 clc;
2 clf;
3 clear all;
4 //Caption:sinc function
5 t=-10:0.1:10
6 x=sinc(t);
7 plot(t,x);
8 title('sinc wave');
9 xlabel('t');
10 ylabel('x');
```

---

### Scilab code Solution 1.6 Exponential wave

```
1 clc;
2 clf;
3 clear all;
4 //Caption:generation of exponential wave
5 t=-2:0.1:2;
6 x=exp(t);
7 plot(t,x);
8 title('exponential wave');
9 xlabel('t');
10 ylabel('x');
```

---

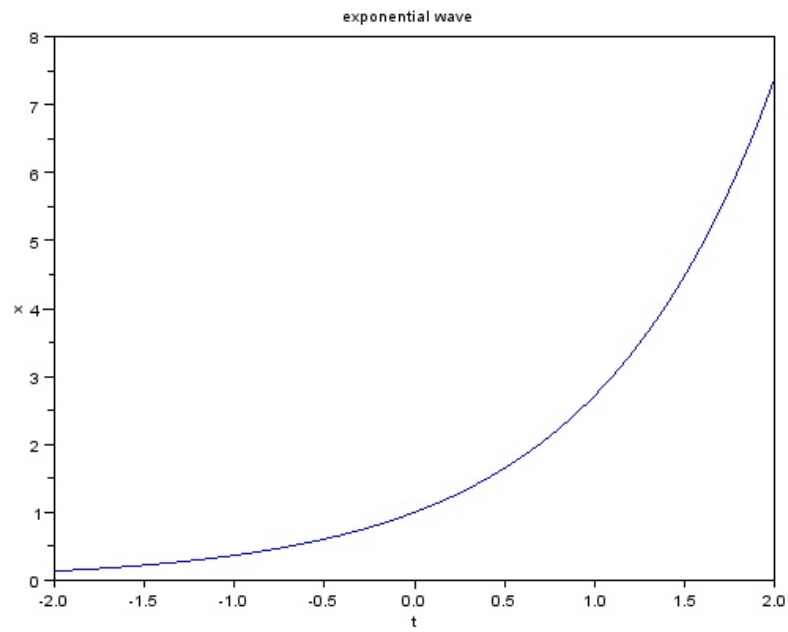


Figure 1.6: Exponential wave

## Experiment: 2

# GENERATION OF DISCRETE SIGNALS

Scilab code Solution 2.1 unit impulse signal

```
1  clc;
2  clf;
3  clear all;
4  //unit impulse
5  L=5;
6  n=-L:L;
7  x=[zeros(1,L),ones(1,1),zeros(1,L)];
8  a=gca();
9  a.y_location="middle"
10 plot2d3(n,x);
11 title('unit impulse');
```

---

Scilab code Solution 2.2 unitstepsignal

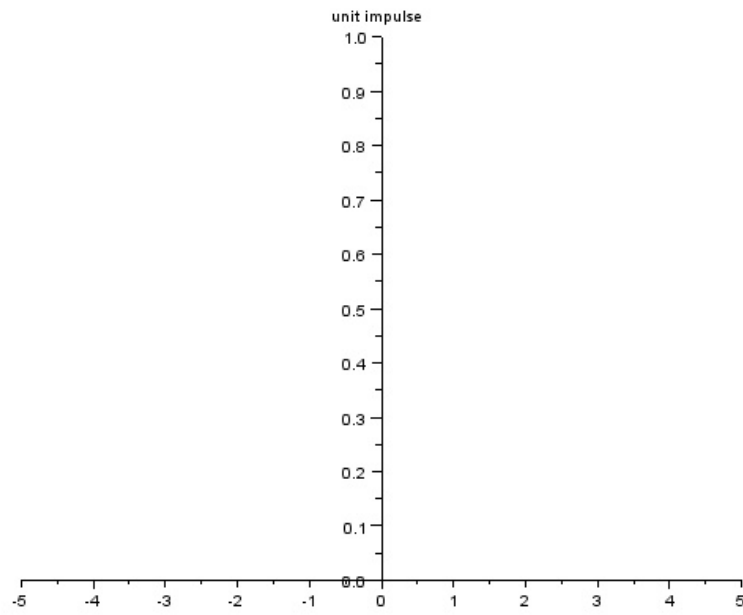


Figure 2.1: unit impulse signal

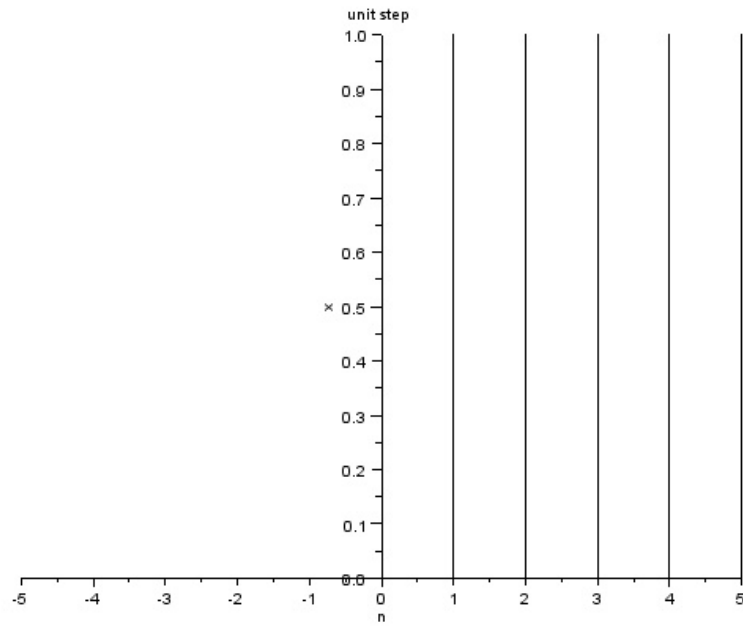


Figure 2.2: unitstepsignal

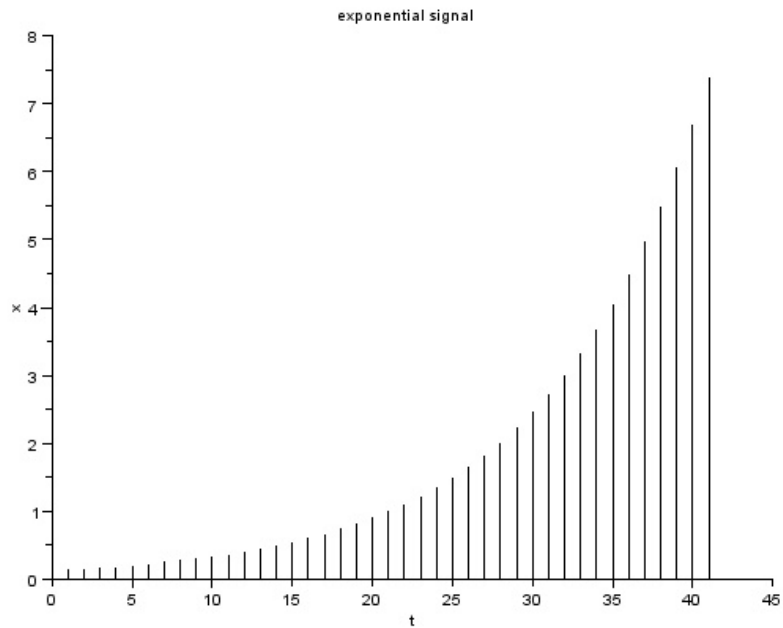


Figure 2.3: discreteexponentialwave

```

1  clc;
2  clf;
3  clear all;
4  L=5;
5  n=-L:L;
6  x=[zeros(1,L),ones(1,L+1)];
7  a=gca();
8  a.y_location="middle";
9  plot2d3(n,x);
10 title('unit step');
11 xlabel('n');
12 ylabel('x');

```

---

### Scilab code Solution 2.3 discreteexponentialwave

```
1 //unit exponential
2 clc;
3 clf;
4 clear all;
5 a=1;
6 x=exp(a*t);
7 plot2d3(x);
8 title('exponential signal');
9 xlabel('t');
10 ylabel('x');
```

---

### Scilab code Solution 2.4 unit ramp

```
1 //unit ramp
2 clc;
3 clf;
4 clear all;
5 L=5;
6 n=-L:L;
7 x=[zeros(1,L),0:L];
8 a =gca();
9 a.y_location='middle';
10 plot2d3(n,x);
11 xtitle('unit ramp signal');
12 xlabel('—>n');
13 ylabel('—>x(n)');
```

---

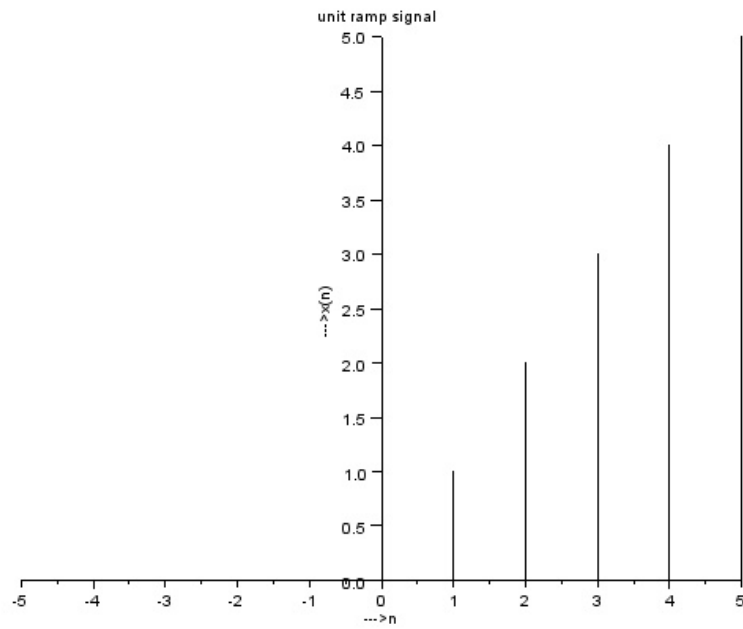


Figure 2.4: unit ramp

## Experiment: 3

# GENERATION OF SINUSOIDAL SIGNALS

Scilab code Solution 3.1 Generation of sinusoidal signals

```
1  clc;
2  clear all;
3  tic;
4  t=0:.01:%pi;
5  //generation of sine signals
6  y1=sin(t);
7  y2=sin(3*t)/3;
8  y3=sin(5*t)/5;
9  y4=sin(7*t)/7;
10 y5=sin(9*t)/9;
11 y = sin(t) + sin(3*t)/3 + sin(5*t)/5 + sin(7*t)/7 +
    sin(9*t)/9;
12 plot(t,y,t,y1,t,y2,t,y3,t,y4,t,y5);
13 legend('y','y1','y2','y3','y4','y5');
14 title('generation of sum of sinusoidal signals');
15 xgrid(1);
16 ylabel('—> Amplitude');
```

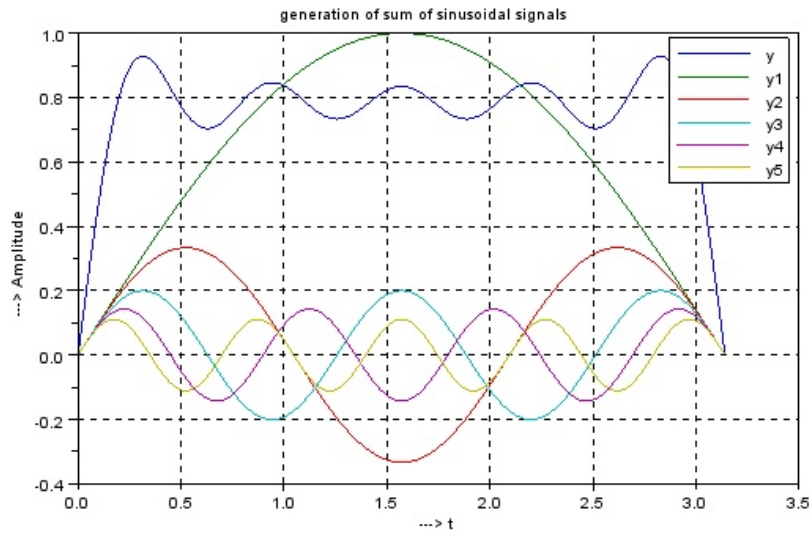


Figure 3.1: Generation of sinusoidal signals

```
17 xlabel('—> t');  
18 toc;
```

---

## Experiment: 4

# GENERATION OF COMPLEX EXPONENTIAL SIGNALS

Scilab code Solution 4.1 generation of complex exponential signals

```
1 //program for generation of complex exponential
  signals
2 clc;
3 clf;
4 clear all;
5 n=-10:0.1:10;
6 a=0.8;
7 b=-0.8;
8 c=1.25;
9 d=-1.25;
10 x1=a^n;
11 subplot(2,2,1);
12 plot2d3(n,x1);
13 xlabel('n');
14 ylabel('amplitude');
```

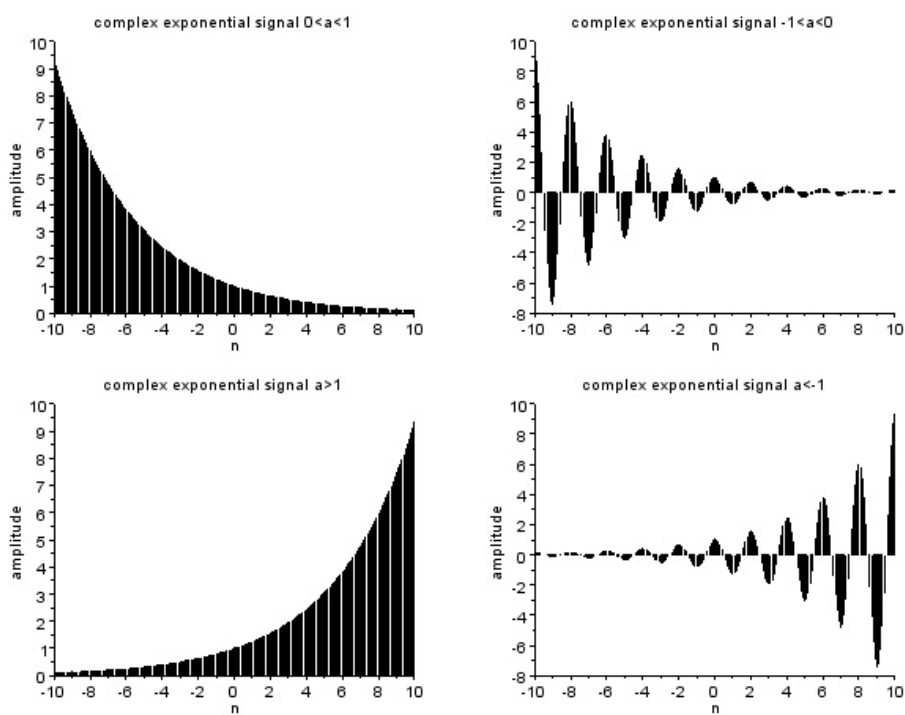


Figure 4.1: generation of complex exponential signals

```
15 title('complex exponential signal  $0 < a < 1$ ');
16 x2=b^n;
17 subplot(2,2,2);
18 plot2d3(n,x2);
19 xlabel('n');
20 ylabel('amplitude');
21 title('complex exponential signal  $-1 < a < 0$ ');
22 x3=c^n;
23 subplot(2,2,3);
24 plot2d3(n,x3);
25 xlabel('n');
26 ylabel('amplitude');
27 title('complex exponential signal  $a > 1$ ');
28 x4=d^n;
29 subplot(2,2,4);
30 plot2d3(n,x4);
31 xlabel('n');
32 ylabel('amplitude');
33 title('complex exponential signal  $a < -1$ ');
```

---

## Experiment: 5

# EXPONENTIALLY GROWING & DECAYING SIGNAL

Scilab code Solution 5.1 Exponentially Growing signals

```
1 //Exponentially decaying signal
2 clc;
3 clf;
4 clear all;
5 disp('Exponentially Decaying signal');
6 N=2;
7 a= 0.1;
8 n=0:0.1:N;
9 x=a.^-n;
10 disp('Exponentially growing signal');
11 disp(x);
12 plot2d3(n,x);
13 xlabel('Time');
14 ylabel('Amplitude');
15 title('Exponentially growing Signal Response');
```

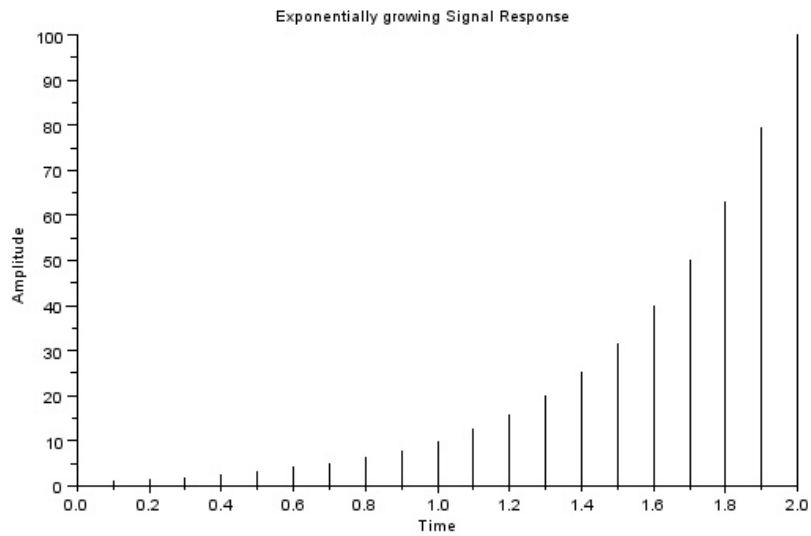


Figure 5.1: Exponentially Growing signals

---

### Scilab code Solution 5.2 Exponentially decaying signals

```
1
2 clc;
3 clf;
4 clear all;
5 disp('Exponentially Decaying signal');
6 N=5;
7 a= 0.1;
8 n=0:0.1:N;
9 x=a.^n;
10 disp('Exponentially decaying signal');
11 disp(x);
```

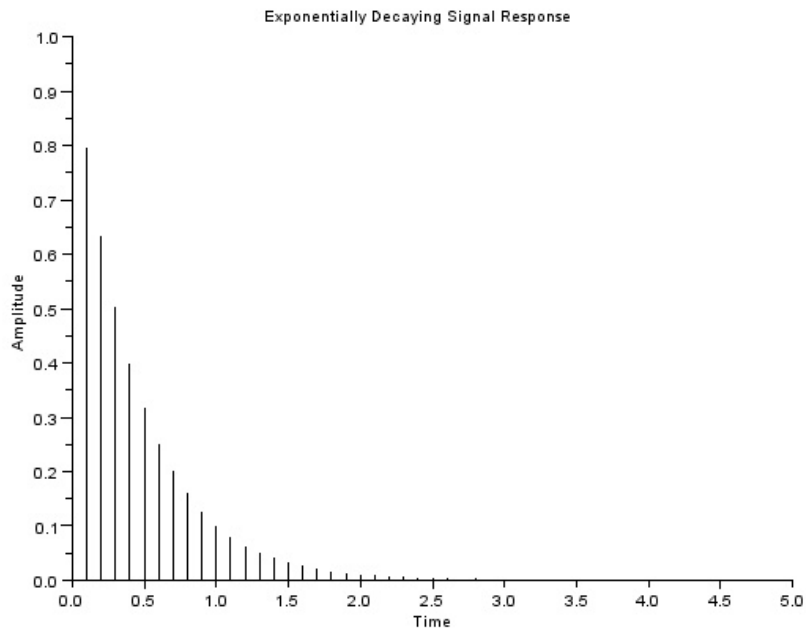


Figure 5.2: Exponentially decaying signals

```
12 plot2d3(n,x);
13 xlabel('Time');
14 ylabel('Amplitude');
15 title('Exponentially Decaying Signal Response');
```

---

# Experiment: 6

## ADDITION OF TWO CONTINUOUS SIGNALS

Scilab code Solution 6.1 Addition of two continuous signals

```
1 //program for addition of two continuous signals
2 clc;
3 clf;
4 clear all;
5 t=0:0.1:30
6 x1=sin(2*%pi*t/9);
7 x2=sin(2*%pi*t/2);
8 x3=x1+x2;
9 subplot(1,3,1);
10 plot(t,x1);
11 title('sinewave1');
12 xlabel('t');
13 ylabel('x1(t)');
14 subplot(1,3,2);
15 plot(t,x2);
16 title('sinewave2');
17 xlabel('t');
```

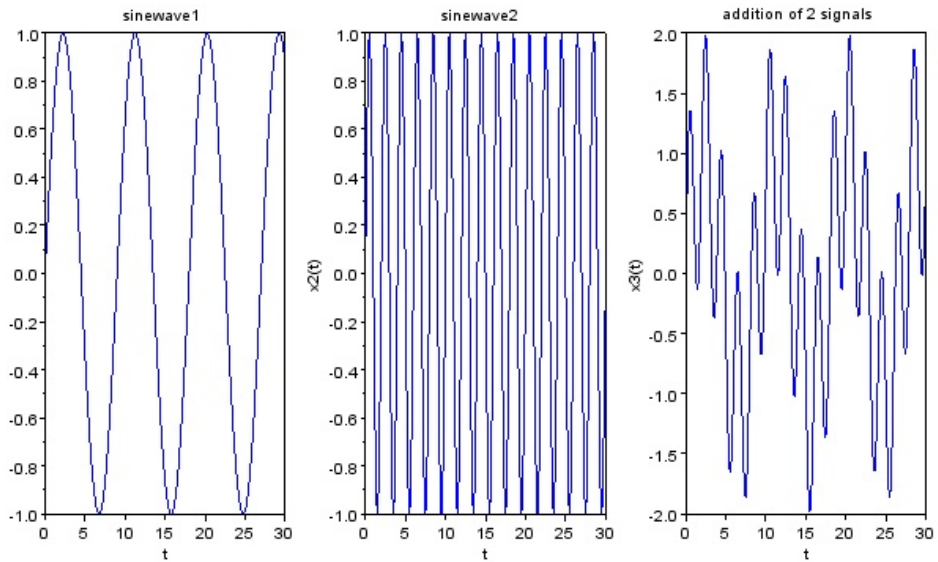


Figure 6.1: Addition of two continuous signals

```
18 ylabel('x2(t)');  
19 subplot(1,3,3);  
20 plot(t,x3);  
21 title('addition of 2 signals');  
22 xlabel('t');  
23 ylabel('x3(t)');
```

---

# Experiment: 7

## ADDITION OF TWO DISCRETE SIGNALS

Scilab code Solution 7.1 Addition of two discrete signals

```
1 //program for addition of two discrete signals
2 N=5;
3 n=0:0.1:N-1;
4 x1=sin(1*%pi*n);
5 x2=sin(2*%pi*n);
6 x3=x1+x2;
7 figure(1);
8 subplot(1,3,1);
9 plot2d3(n,x1);
10 title('sine wave1')
11 xlabel('n');
12 ylabel('x1(n)');
13 subplot(1,3,2)
14 plot2d3(n,x2);
15 title('sinewave2')
16 xlabel('n');
17 ylabel('x2(n)');
```

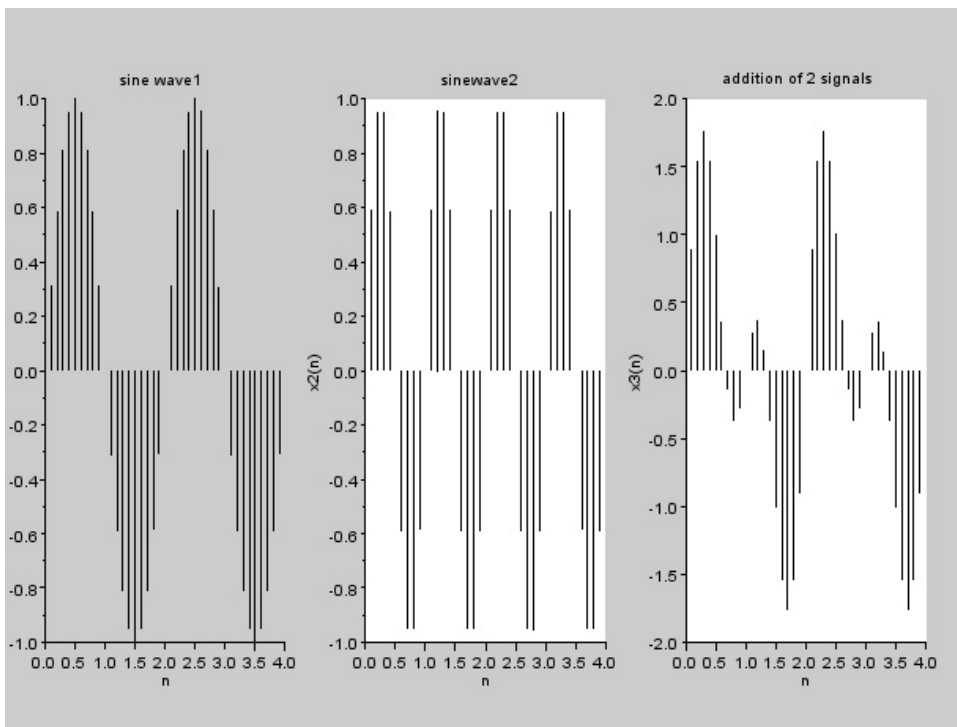


Figure 7.1: Addition of two discrete signals

```
18 subplot(1,3,3);
19 plot2d3(n,x3);
20 title('addition of 2 signals')
21 xlabel('n');
22 ylabel('x3(n)');
```

---

# Experiment: 8

## PROGRAM FOR SAMPLING PROCESS

Scilab code Solution 8.1 sampling

```
1 //program for sampling
2 clc;
3 clf;
4 clear all;
5 t=0:0.01:100;
6 fm=0.02;
7 x=cos(2*%pi*fm*t);
8 subplot(2,2,1);
9 plot(t,x);
10 title('continious time signal');
11 fs1=0.002;
12 n=0:1:50;
13 x1=cos(2*%pi*fm*n/fs1);
14 subplot(2,2,2);
15 plot2d3(n,x1);
16 title('time signal fs1<2fm');
17 fs2=0.04;
```

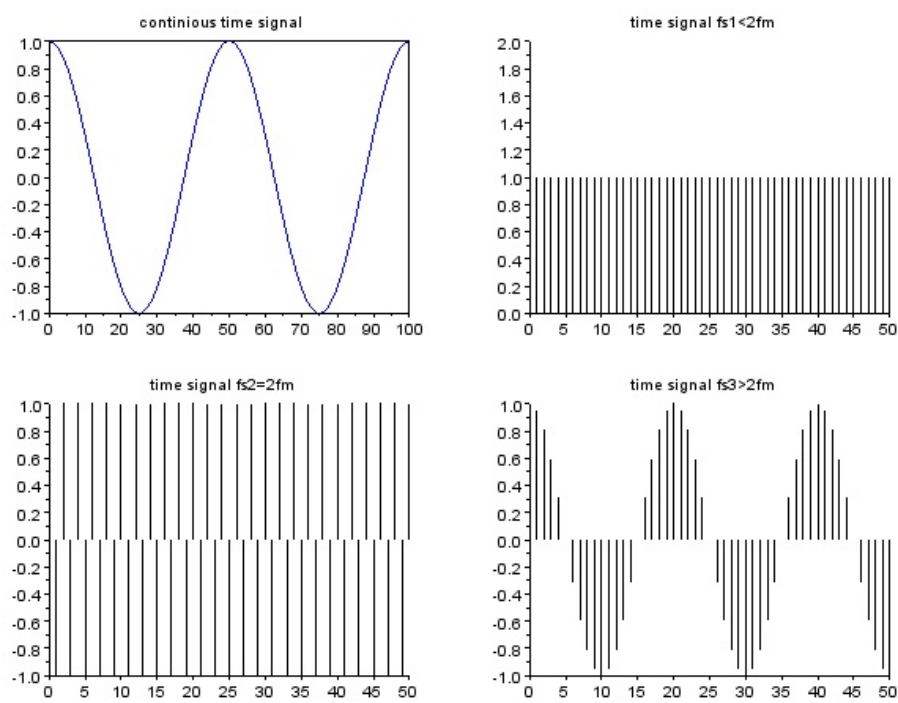


Figure 8.1: sampling

```
18 x2=cos(2*pi*fm*n/fs2);
19 subplot(2,2,3);
20 plot2d3(n,x2);
21 title('time signal fs2=2fm');
22 fs3=0.4;
23 x3=cos(2*pi*fm*n/fs3);
24 subplot(2,2,4);
25 plot2d3(n,x3);
26 title('time signal fs3>2fm');
```

---

## Experiment: 9

# PROGRAM FOR ALIASING PROCESS

Scilab code Solution 9.1 Aliasing process

```
1 //program for effect of aliasing
2 clc;
3 clf;
4 clear all;
5 fs=8000;
6 t=0:1/fs:0.01;
7 f=[1000,2000,3000,3500,4000,4500,8000,8500,9000];
8 i=1;
9 while i<7
10     y=sin(2*%pi*f(i)*t);
11     subplot(3,2,i);
12     plot(t,y);
13     xlabel('frequency');
14     ylabel('amplitude');
15     i=i+1;
16 end
```

---

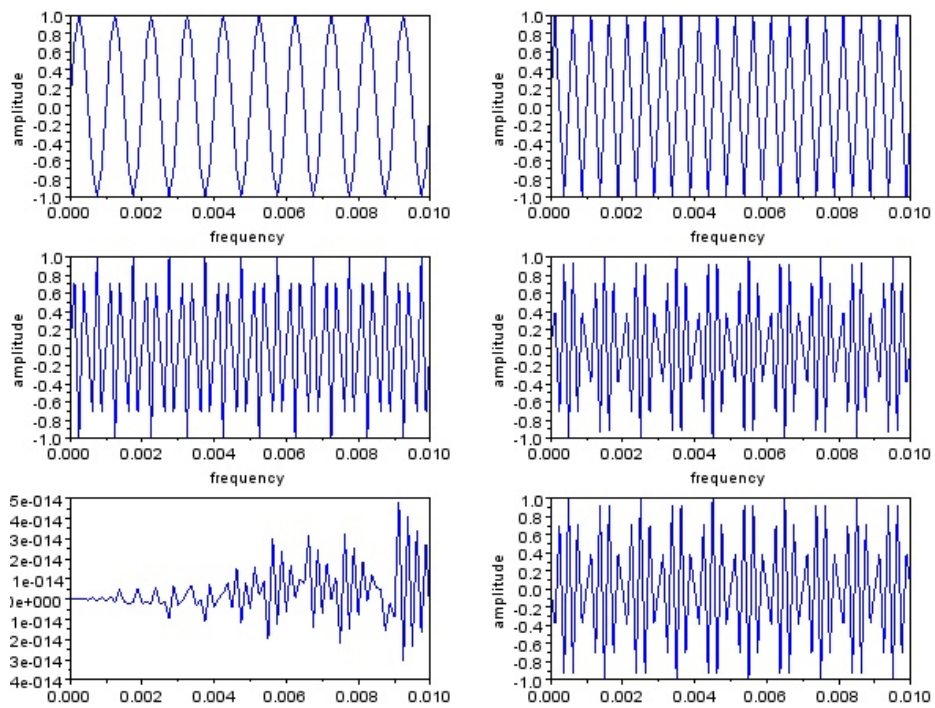


Figure 9.1: Aliasing process

## Experiment: 10

# LINEAR CONVOLUTION OF TWO SEQUENCES

Scilab code Solution 10.1 linear convolution of two sequences

```
1 //program for linear convolution
2 clc;
3 clf;
4 clear all;
5 x=input('Enter the first sequence:');
6 y=input('Enter the second Sequence:');
7 n=convol(x,y);
8 subplot(2,2,1);
9 plot2d3(x);
10 title('first sequence');
11 xlabel('n————>');
12 ylabel('amp————>');
13 subplot(2,2,2);
14 plot2d3(y);
15 title('second sequence');
16 xlabel('n————>');
17 ylabel('amp————>');
18 subplot(2,2,3);
19 plot2d3(n);
```

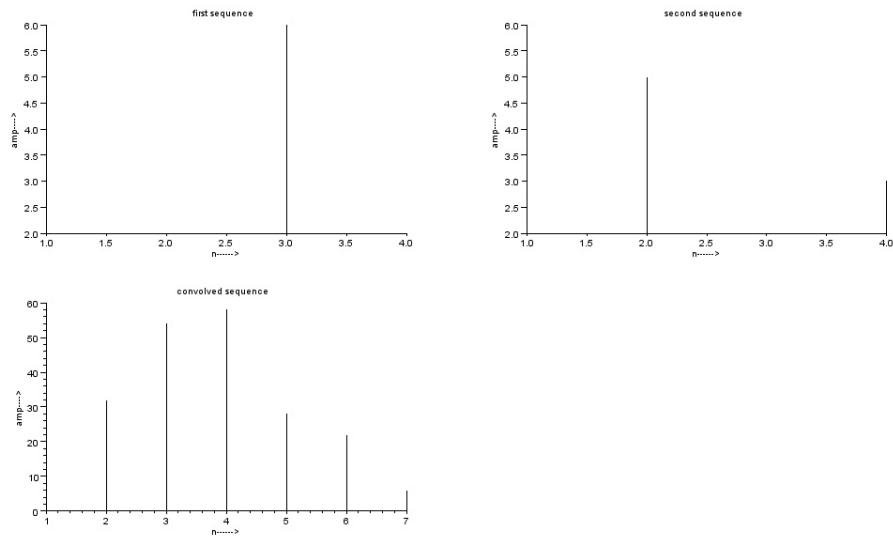


Figure 10.1: linear convolution of two sequences

```

20 title('convolved sequence');
21 xlabel('n————>');
22 ylabel('amp————>');
23 disp('The convolved sequence is');
24 disp(n);
25
26 //result:
27 //Enter the first sequence:[4 2 6 2]
28 //Enter the second Sequence:[6 5 2 3]
29 //
30 // The convolved sequence is
31 //
32 //      24.      32.      54.      58.      28.      22.      6.

```

---

## Experiment: 11

# CIRCULAR CONVOLUTION OF TWO SEQUENCES

Scilab code Solution 11.1 circular convolution

```
1 //program for circular convolution
2 clc;
3 clf;
4 clear all;
5 g=input('Enter the first sequence:');
6 h=input('Enter the second sequence:');
7 N1=length(g);
8 N2=length(h);
9 N=max(N1,N2);
10 N3=N1-N2;
11 if(N3>=0) then
12     h=[h,zeros(1,N3)];
13 else
14     g=[g,zeros(1,-N3)];
15 end
16 for n=1:N
17     y(n)=0;
```

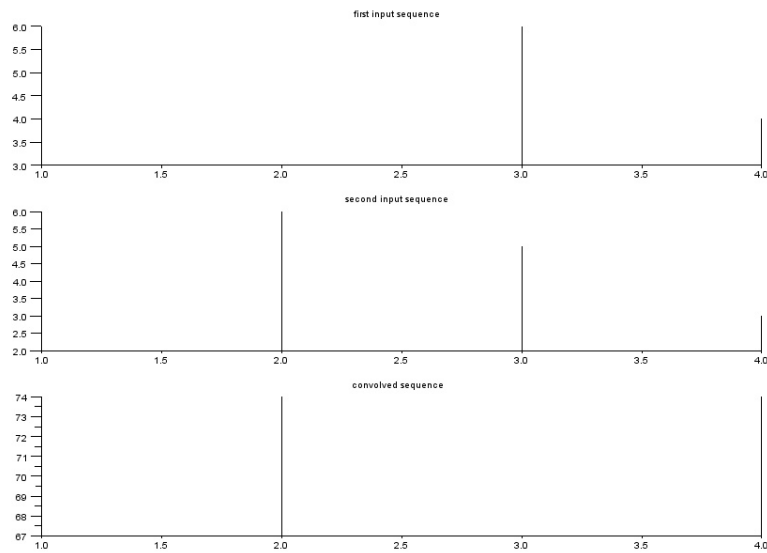


Figure 11.1: circular convolution

```

18 for i=1:N
19     j=n-i+1;
20     if(j<=0)
21         j=N+j;
22     end
23     y(n)=y(n)+g(i)*h(j);
24 end
25 end
26 disp('The resultant signal is:');
27 disp(y);
28 subplot(3,1,1);
29 plot2d3(g);
30 title('first input sequence');
31 subplot(3,1,2);
32 plot2d3(h);
33 title('second input sequence');
34 subplot(3,1,3);
35 plot2d3(y);
36 title('convolved sequence');
37

```

```
38
39
40 //result
41 //Enter the first sequence:[5 3 6 4]
42 //Enter the second sequence:[2 6 5 3]
43 //
44 // The resultant signal is:
45 //
46 //      73.      74.      67.      74.
```

---

# Experiment: 12

## DFT OF A SIGNAL

Scilab code Solution 12.1 DFT

```
1 //program for calculation of DFT of a signal
2 clc;
3 clf;
4 clear all;
5 N=input('Enter the value of N:');
6 x=input('enter input sequence');
7 for k=1:N
8     y(k)=0;
9     for n=1:N
10        y(k)=y(k)+x(n).*exp(-%i*2*%pi*(k-1)*(n-1)/N)
11        ;
12        A=real(y);
13        B=imag(y);
14    end;
15 end;
16 mag=abs(y);
17 x1=atan(imag(y),real(y));
18 phase=x1*(180/%pi);
19 disp('the resultant DFT sequence is');
```

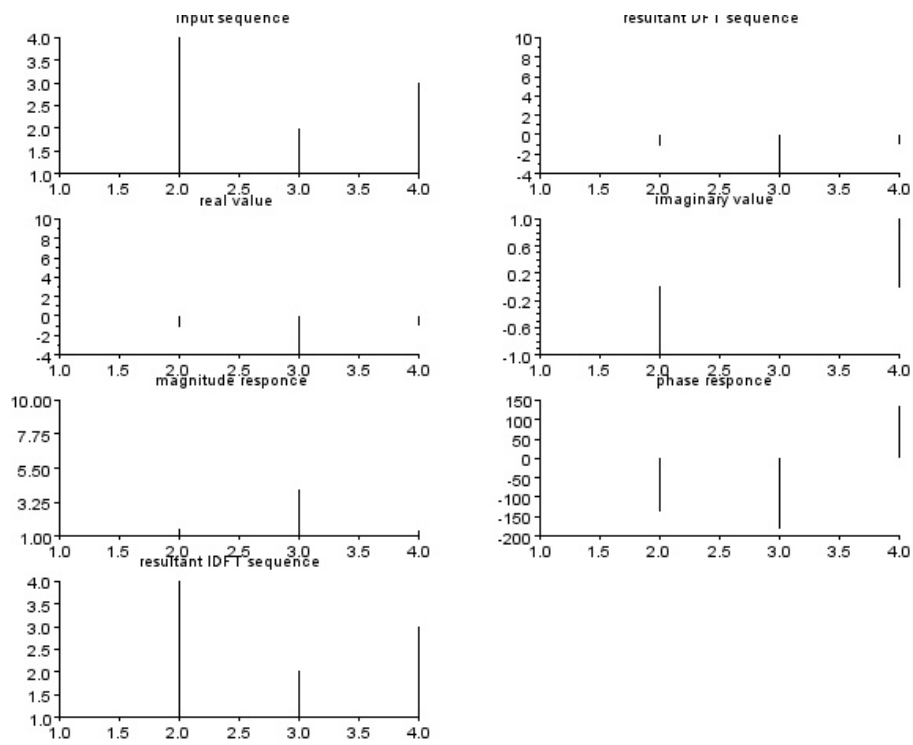


Figure 12.1: DFT

```

19 disp(y);
20 disp('the resultant real value is');
21 disp(A);
22 disp('the resultant imaginary value is');
23 disp(B);
24 disp('the magnitude response is');
25 disp(mag);
26 disp('the phase response is');
27 disp(phase);
28 for n=1:N
29     y(n)=0;
30     for k=1:N
31         y(n)=y(n)+(1/N)*(y(k).*exp(%i*2*%pi*(k-1)*(n
32             -1)/N));
33         C=real(x);
34     end;
35 end;
36 disp('the resultant IDFT sequence is');
37 disp(C);
38 subplot(4,2,1);
39 plot2d3(x);
40 title('input sequence');
41 subplot(4,2,2);
42 plot2d3(A);
43 title('resultant DFT sequence');
44 subplot(4,2,3);
45 plot2d3(A);
46 title('real value');
47 subplot(4,2,4);
48 plot2d3(B);
49 title('imaginary value');
50 subplot(4,2,5);
51 plot2d3(mag);
52 title('magnitude response');
53 subplot(4,2,6);
54 plot2d3(phase);
55 title('phase response');
56 subplot(4,2,7);

```

```

56 plot2d3(C);
57 title('resultant IDFT sequence');
58
59
60 //result
61 //Enter the value of N:4
62 //enter input sequence[1 4 2 3]
63
64 // the resultant DFT sequence is
65
66 //      10.  - 1. - i      - 4. - 1.102D-15i  - 1. + i
67
68 // the resultant real value is
69 //      10.  - 1.  - 4.  - 1.
70
71 // the resultant imaginary value is
72
73 //      0.  - 1.  - 1.102D-15      1.
74
75 // the magnitude response is
76
77 //      10.      1.4142136      4.      1.4142136
78
79 // the phase response is
80
81 //      0.  - 135.  - 180.      135.
82
83 // the resultant IDFT sequence is
84
85 //      1.      4.      2.      3.

```

---

# Experiment: 13

## FFT OF A SIGNAL

Scilab code Solution 13.1 fast fourier transform

```
1 //program for calculation of FFT of a signal
2 clc;
3 clf;
4 clear all;
5 N=input('Enter the value of N:');
6 x=input('enter input sequence ');
7 y=fft(x);
8 A=real(y);
9 B=imag(y);
10 mag=abs(y);
11 x1=atan(imag(y),real(y));
12 phase=x1*(180/%pi);
13 disp('the resultant FFT sequence is ');
14 disp(y);
15 disp('the magnitude response is ');
16 disp(mag);
17 disp('the phase response is ');
18 disp(phase);
19 z=ifft(y);
```

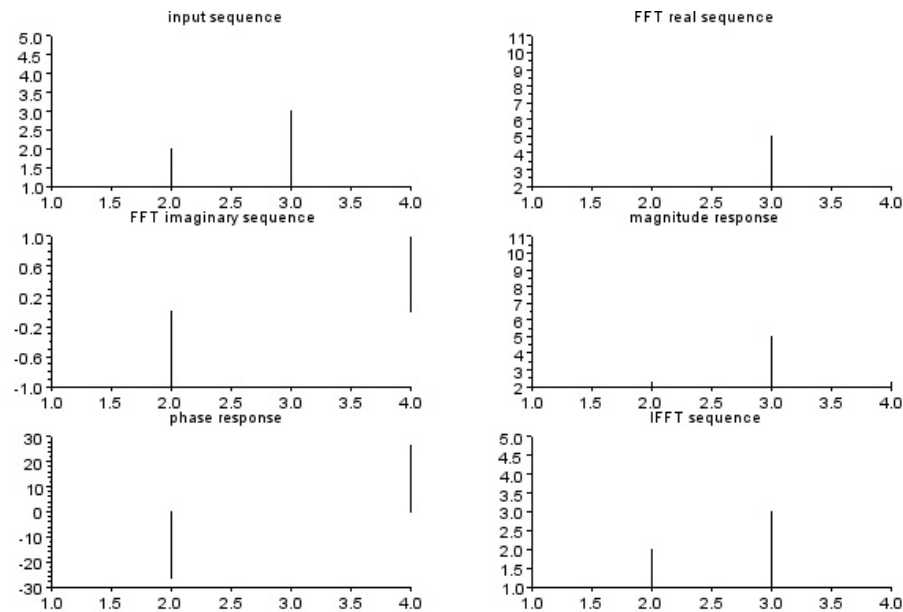


Figure 13.1: fast fourier transform

```

20 disp('the resultant IFFT sequence is ');
21 disp(z);
22 subplot(3,2,1);
23 plot2d3(x);
24 title('input sequence');
25 subplot(3,2,2);
26 plot2d3(A);
27 title('FFT real sequence');
28 subplot(3,2,3);
29 plot2d3(B);
30 title('FFT imaginary sequence');
31 subplot(3,2,4);
32 plot2d3(mag);
33 title('magnitude response');
34 subplot(3,2,5);
35 plot2d3(phase);
36 title('phase response');
37 subplot(3,2,6);
38 plot2d3(x);

```

```

39 title('IFFT sequence');
40
41
42 //result
43 //Enter the value of N:4
44 //enter input sequence[5 2 3 1]
45
46 //the resultant FFT sequence is
47
48 //      11.      2. - i      5.      2. + i
49
50 //the magnitude response is
51
52 //      11.      2.236068      5.      2.236068
53
54 //the phase response is
55
56 //0. - 26.565051      0.      26.565051
57
58 //the resultant IFFT sequence is
59
60 //      5.      2.      3.      1.

```

---