Scilab Textbook Companion for Schaums Outlines Signals And Systems by H. P. Hsu[1]

Created by
Puneetha Ramachandra
B.Tech (pursuing)
Electronics Engineering
NIT, Surathkal
College Teacher
NA
Cross-Checked by
K. Suryanarayan, IITB

May 17, 2016

[1] Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Schaums Outlines Signals And Systems

Author: H. P. Hsu

Publisher: Tata McGraw Hill

Edition: 3

Year: 2004

ISBN: 0-07-030641-9

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

List of Figures

Chapter 1

Signals and Systems

Scilab code Exa 1.1 shifting and scaling of continuous time signal

```
//shifting and scaling
//example 1.1
clear;
clc;
close;
t = 0:1/100:4;
for i = 1:length(t)
  x(i) = (3/4)*t(i) ;
end
for i = length(t)+1:2*length(t)
x(i) = 0;
end
figure
a=gca();
t1=0:1/100:8;
plot(t1,x(1:$-1))
xtitle('x(t)')
figure
a=gca();
t2=t1+2;
plot(t2,x(1:$-1))
```

```
xtitle('x(t-2)')
a.y_location='origin'
figure
a=gca();
t3=0:1/200:4;
plot(t3,x(1:$-1))
xtitle('x(2t)')
figure
a=gca();
t4=0:1/50:16;
plot(t4,x(1:$-1))
xtitle('x(t/2)')
figure
a=gca();
t5=-8:1/100:0;
plot(t5,x($:-1:2))
xtitle('x(-t)')
a.y_location = "right";
```

Scilab code Exa 1.2 shifting and scaling of discrete time signal

```
//example 1.2
//shifting and scaling discrete signals
clear ;
clc;
close;
t=-2:6;
x(1:3)=0;
for i = 3:(length(t)-3)
  x(i) =i-3;
end
```

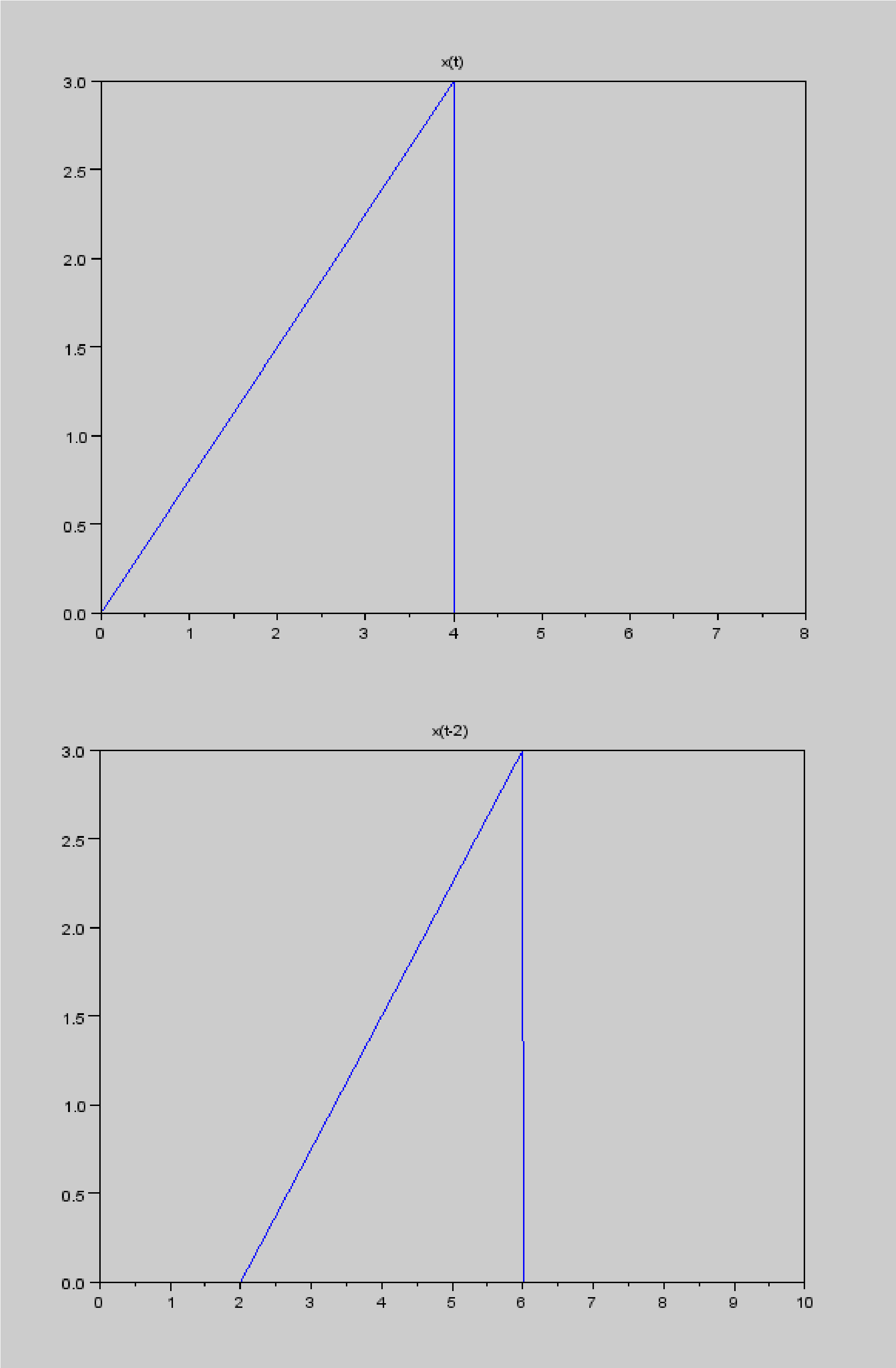
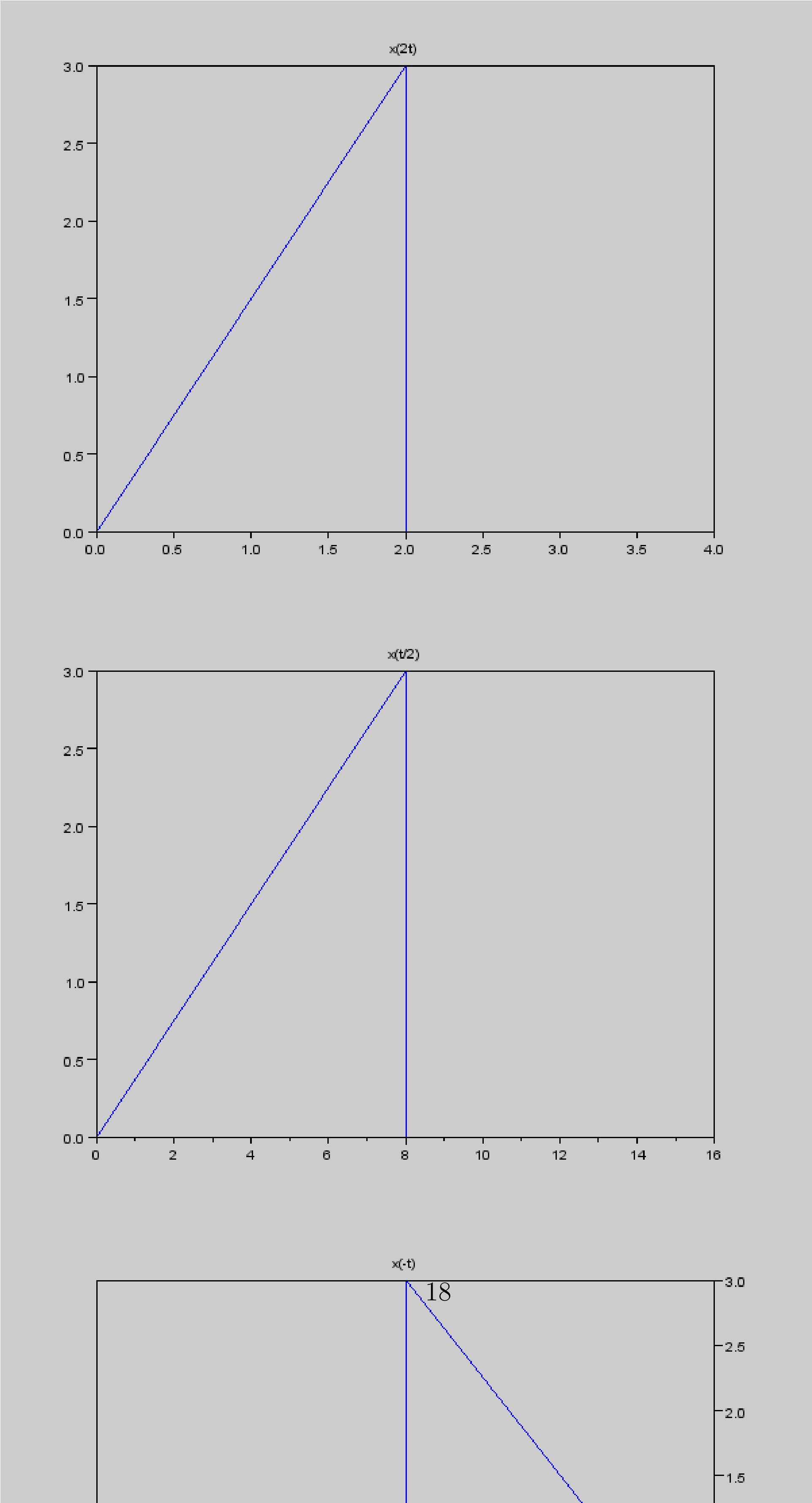


Figure 1.1: shifting and scaling of continuous time signal


x(2t)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x(t/2)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
10
12
14
16
x(-t)
3.0
2.5
2.0
1.5

```
x(i+1)=x(i);
x(i+2:9)=0;
figure
a=gca();
plot2d3(t,x)
plot(t,x,'r.')
xtitle('x[n]')
t1=t+2;
figure
a=gca();
plot2d3(t1,x)
plot(t1,x,'r.')
xtitle('x[n-2]')
a.thickness=2;
t2=-1:1/2:3;
figure
a=gca()
plot2d3(ceil(t2),x)
plot(ceil(t2),x,'r.')
xtitle('x[2n]')
a.thickness=2;
t3=-6:2;
figure
a=gca();
plot2d3(t3,x($:-1:1))
plot(t3,x($:-1:1),'r.')
xtitle('x[-n]')
a.y_location='right';
a.thickness=2;
t4=t3+2;
figure
a=gca();
plot2d3(t4,x($:-1:1))
plot(t4,x($:-1:1),'r.')
xtitle('x[-n+2]')
a.y_location='right';
a.thickness=2;
```

Scilab code Exa 1.3 sampling of continuous time signal

```
//example_1.3
// sampling of continuos function
clear;
clc;
close;
t=-1:1/100:1;
for i=1:length(t)
    x(i)=1-abs(t(i))
end
figure
a=gca();
plot2d(t,x)
xtitle('x(t)')
a.y_location='middle'
figure
a=gca();
for i=1:length(t)
    if t(i)<0 then
        t1(i)=ceil(t(i)*4)
    else
        t1(i)=floor(t(i)*4)
    end
end
plot2d3(ceil(t1),x)
xtitle('x[n]=x[n/4]')
figure
a=gca();
for i=1:length(t)
```

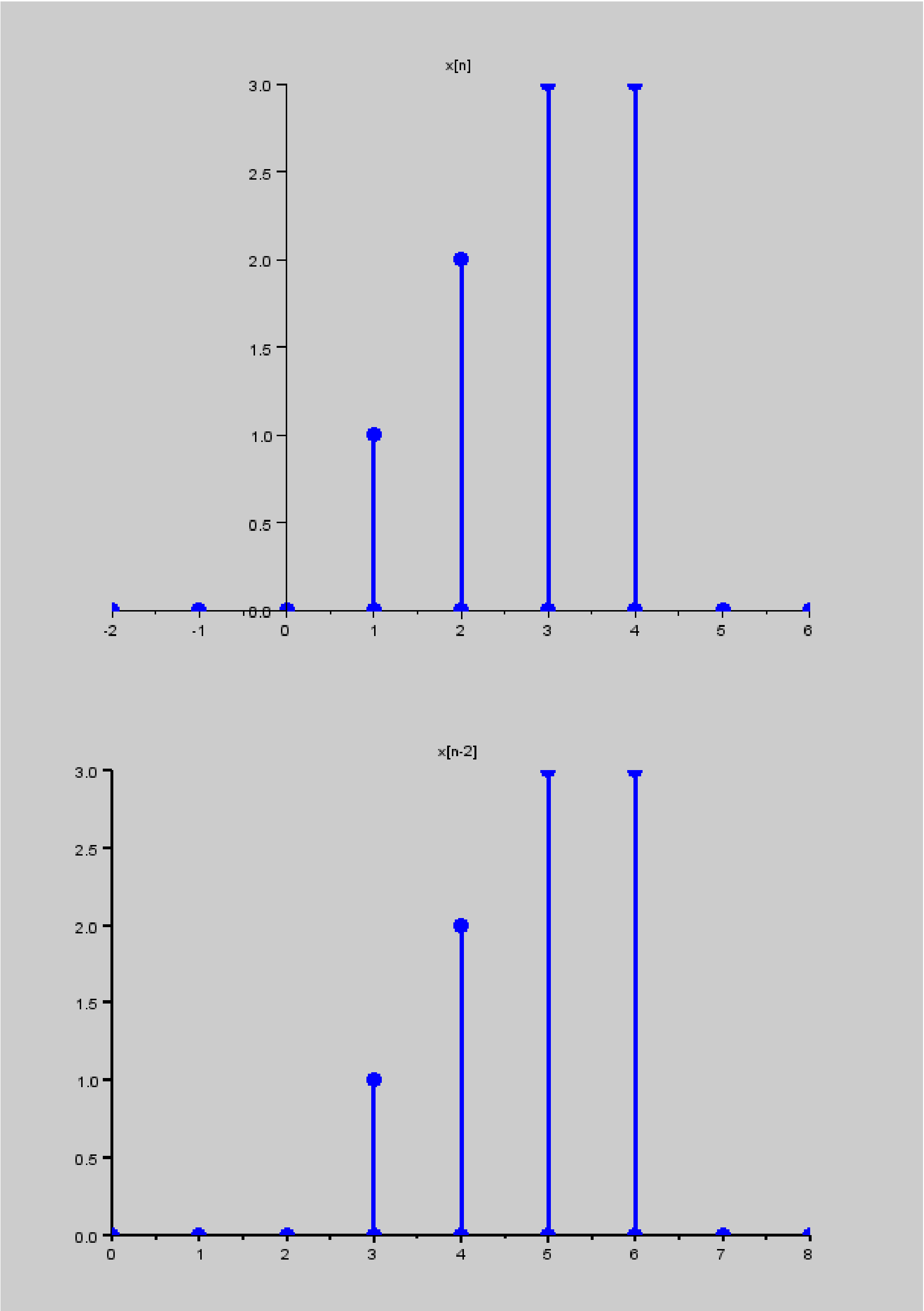
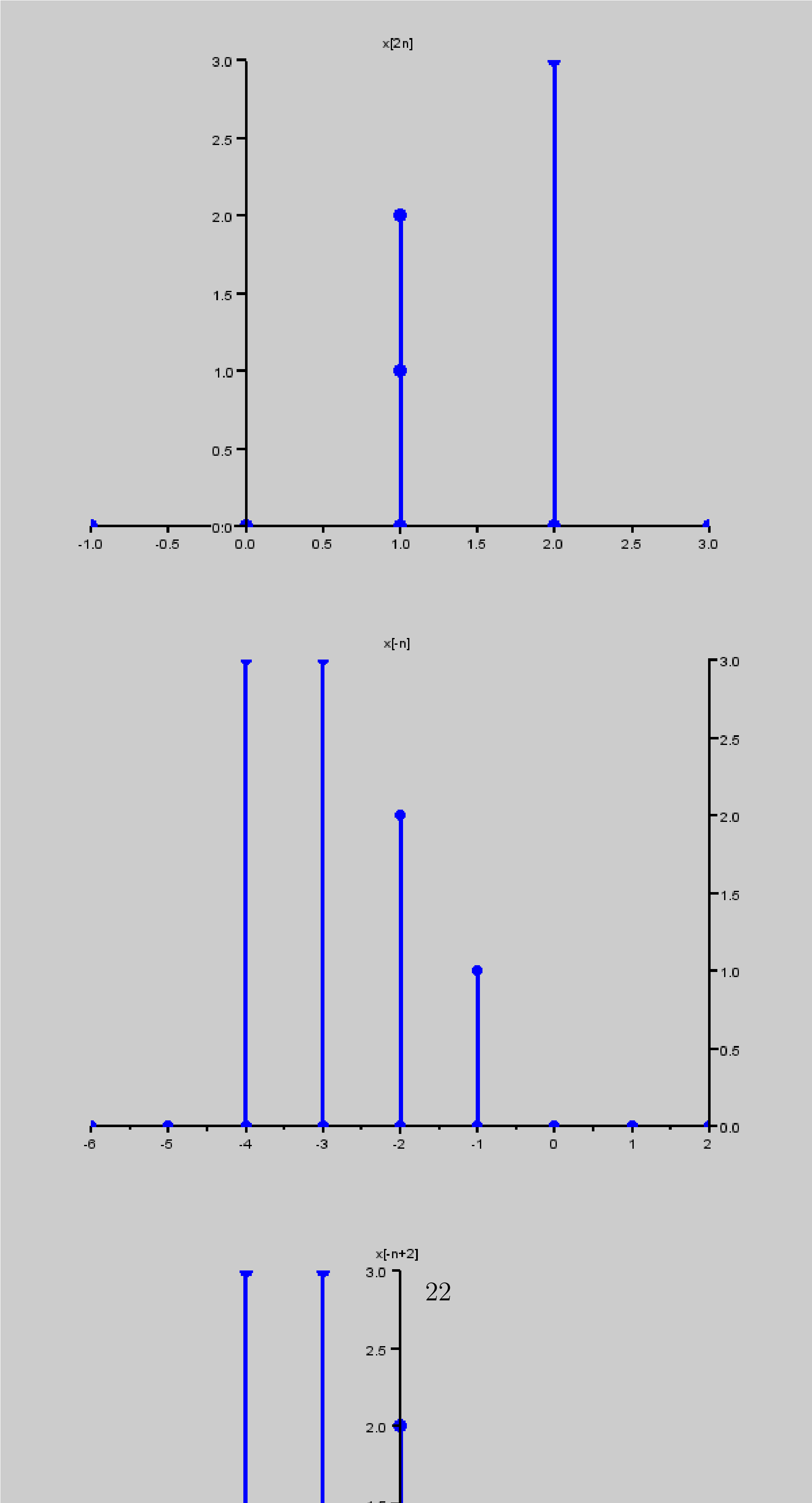


Figure 1.3: shifting and scaling of discrete time signal


x[2n]
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x[-n]
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-6
-5
-4
-3
-2
-1
0
1
2
x[-n+2]
3.0
2.5
2.0

```
    if t(i)<0 then
        t2(i)=ceil(t(i)*2)
    else
        t2(i)=floor(t(i)*2)
    end
end
plot2d3(ceil(t2),x)
xtitle('x[n]=x[n/2]')
figure
a=gca();
for i=1:length(t)
    if t(i)<0 then
        t3(i)=ceil(t(i))
    else
        t3(i)=floor(t(i))
    end
end
plot2d3(ceil(t3),x)
xtitle('x[n]')
```

Scilab code Exa 1.4 discrete time signal

```
//ex_4 combining two discrete signals
clear;
clc;
close;
t1=-2:7
t2=-3:4
x1=[0 0 0 1 2 3 0 0 2 2 0];
x2=[0 -2 -2 2 2 0 -2 0 0 0 0];
t3=min(t1(1),t2(1)):max(t1(length(t1)),t2(length(t2)
```

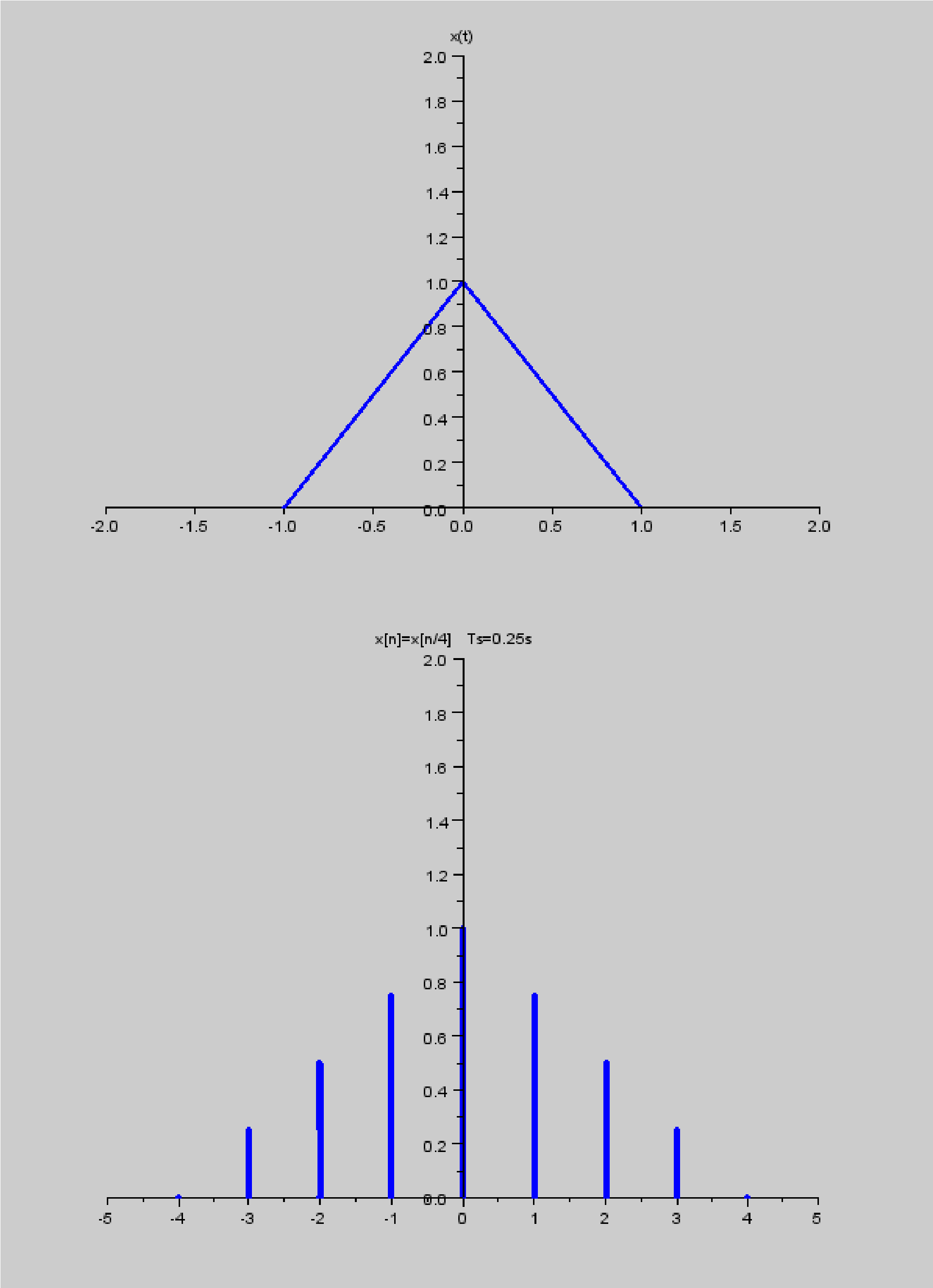


Figure 1.5: sampling of continuous time signal

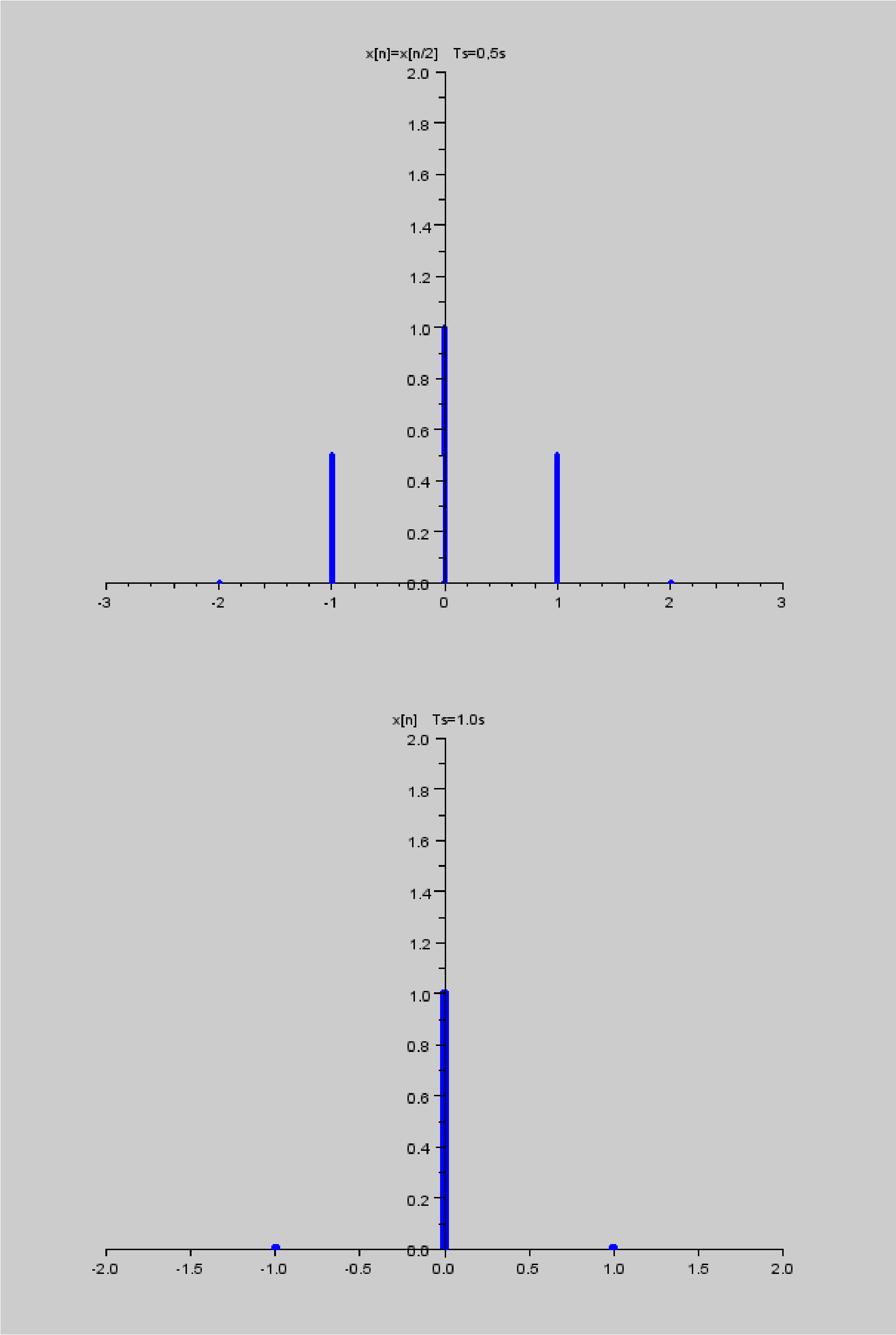


Figure 1.6: sampling of continuous time signal

```
   ));
figure
a=gca();
plot2d3(t3,x1)
plot(t3,x1,'r.')
xtitle('x1[n]')
figure
a=gca();
plot2d3(t3,x2)
plot(t3,x2,'r.')
xtitle('x2[n]')
a.x_location='middle'
figure
a=gca();
plot2d3(t3,x1+x2)
plot(t3,x1+x2,'r.')
xtitle('y1[n]=x1[n]+x2[n]')
a.x_location='origin'
figure
a=gca();
plot2d3(t3,2.*x1)
plot(t3,2.*x1,'r.')
xtitle('y2[n]=2 * x1[n]')
figure
a=gca();
plot2d3(t3,x1.*x2)
plot(t3,x1.*x2,'r.')
xtitle('y2[n]=x2[n] * x1[n]')
```

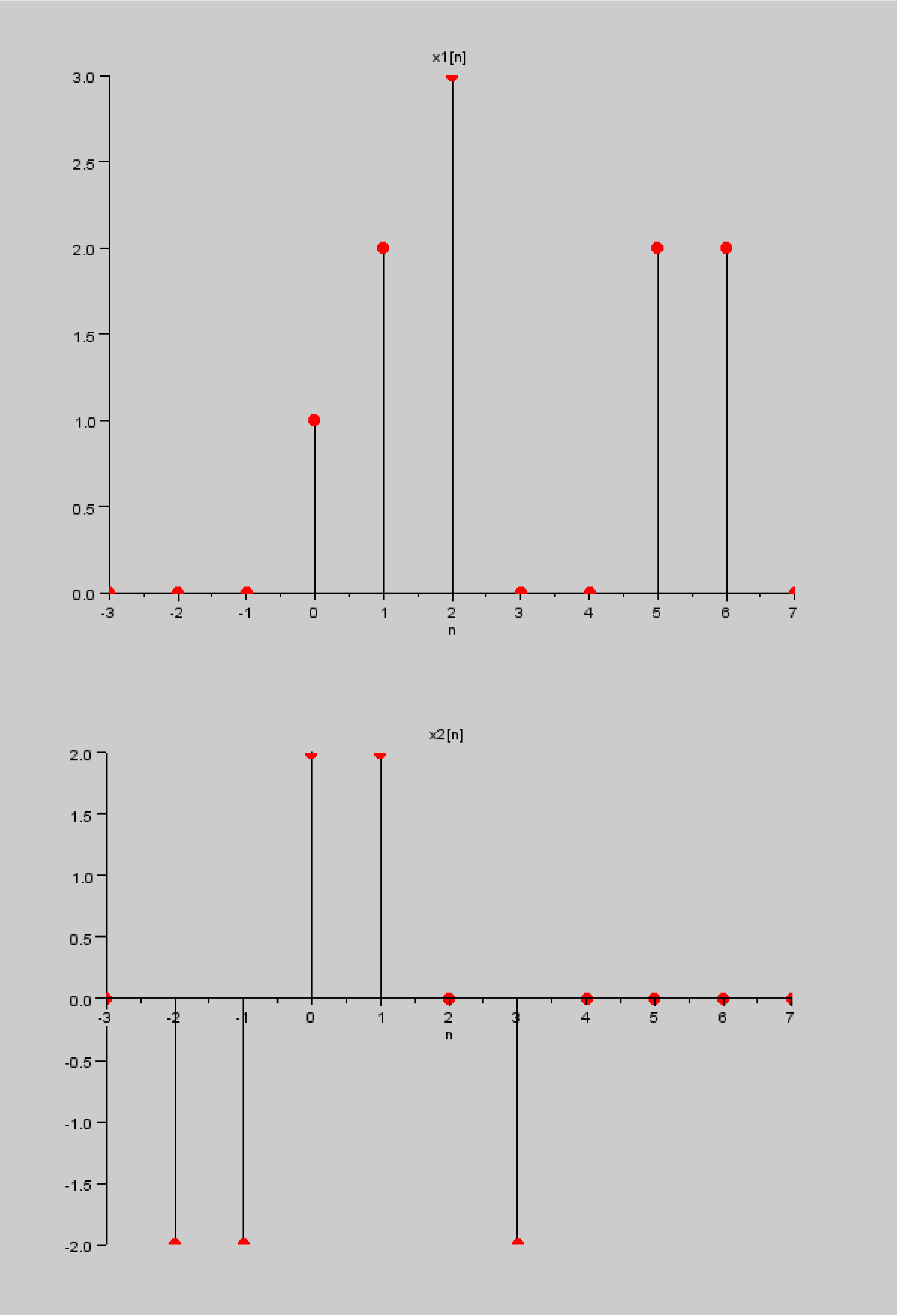
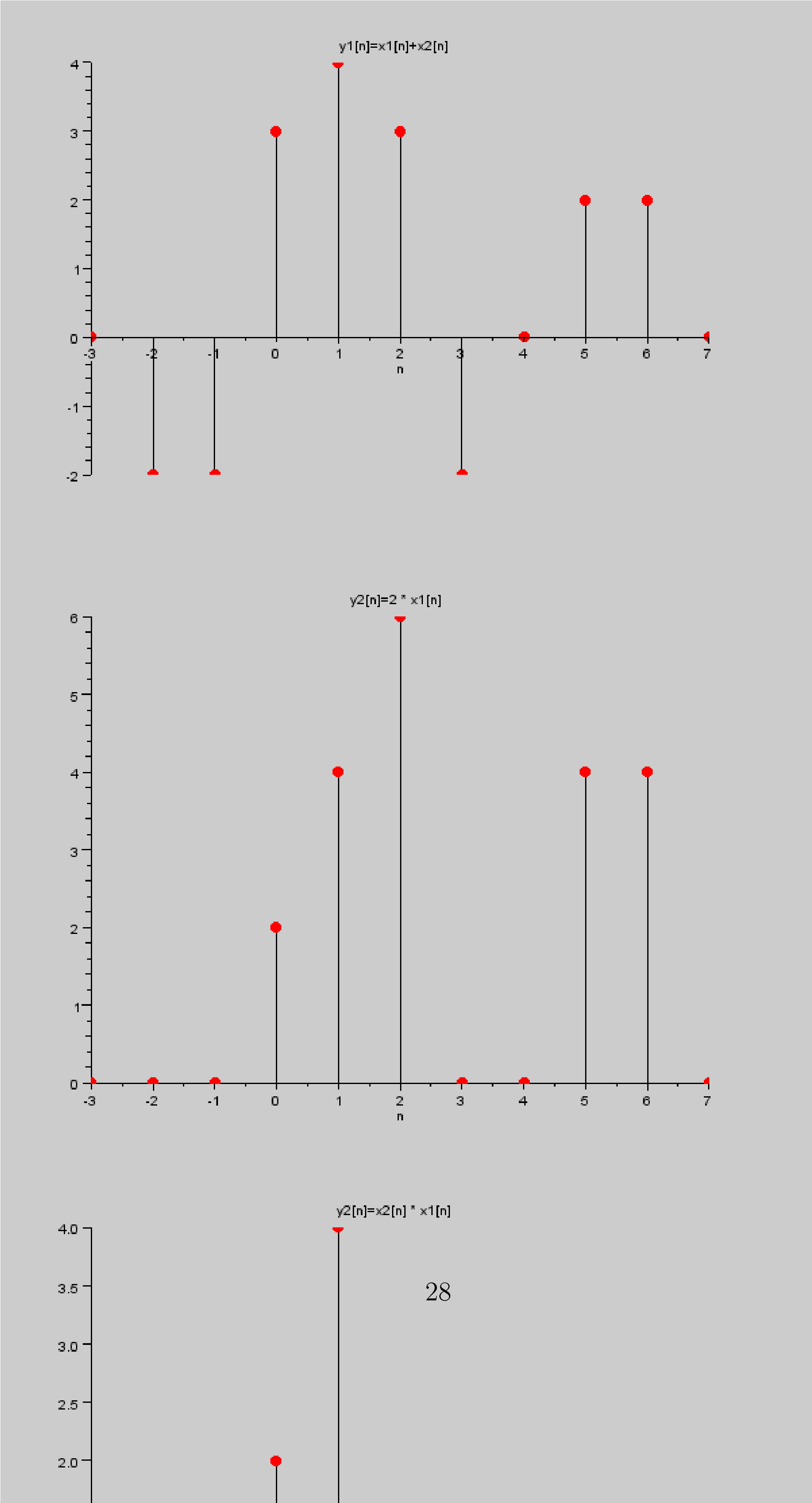
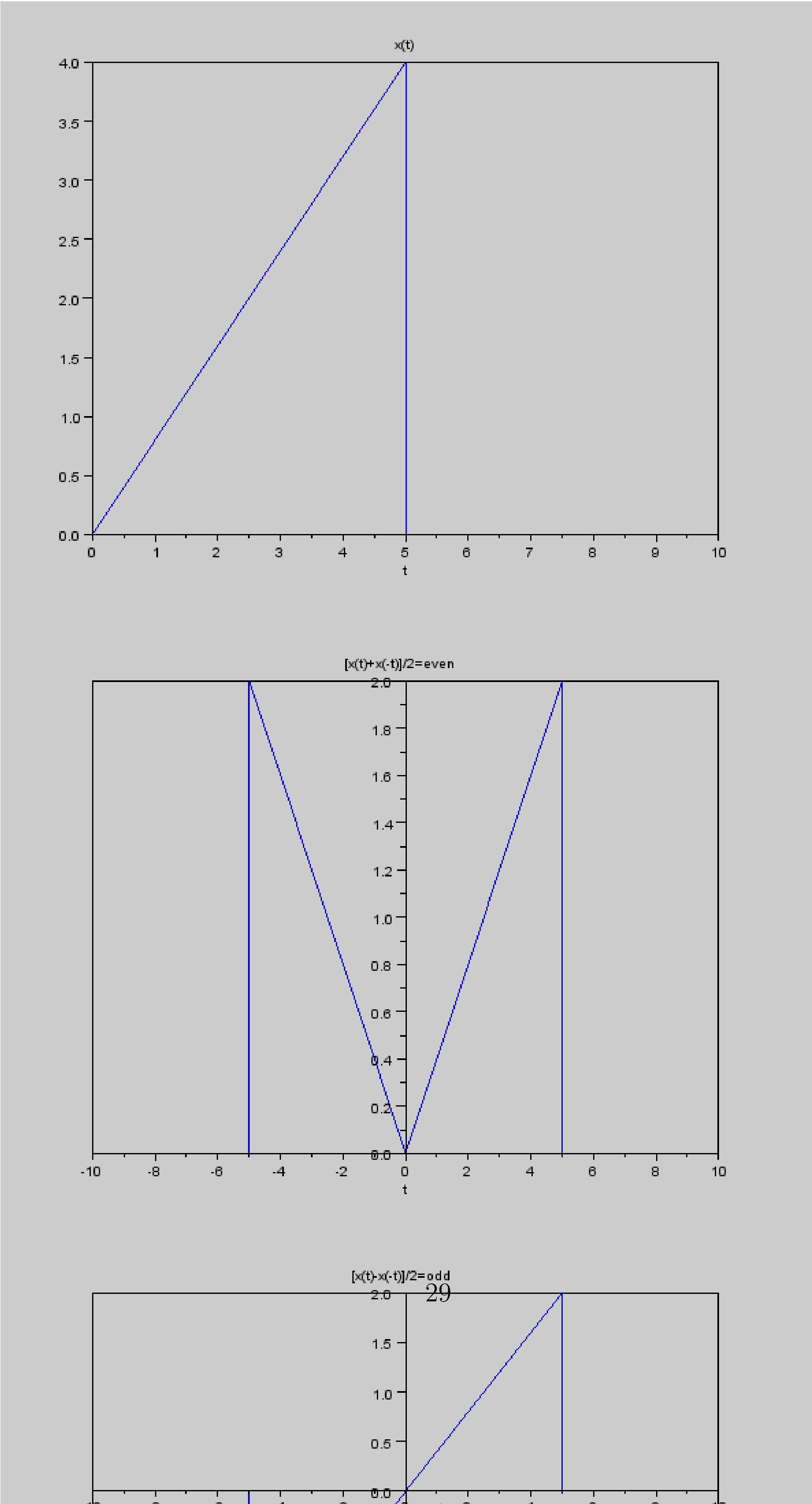


Figure 1.7: discrete time signal


y1[n]=x1[n]+x2[n]
4
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3
4
5
6
7
n
y2[n]=2 * x1[n]
6
5
4
3
2
1
0
-3
-2
-1
0
1
2
3
4
5
6
7
n
y2[n]=x2[n] * x1[n]
4.0
3.5
3.0
2.5
2.0

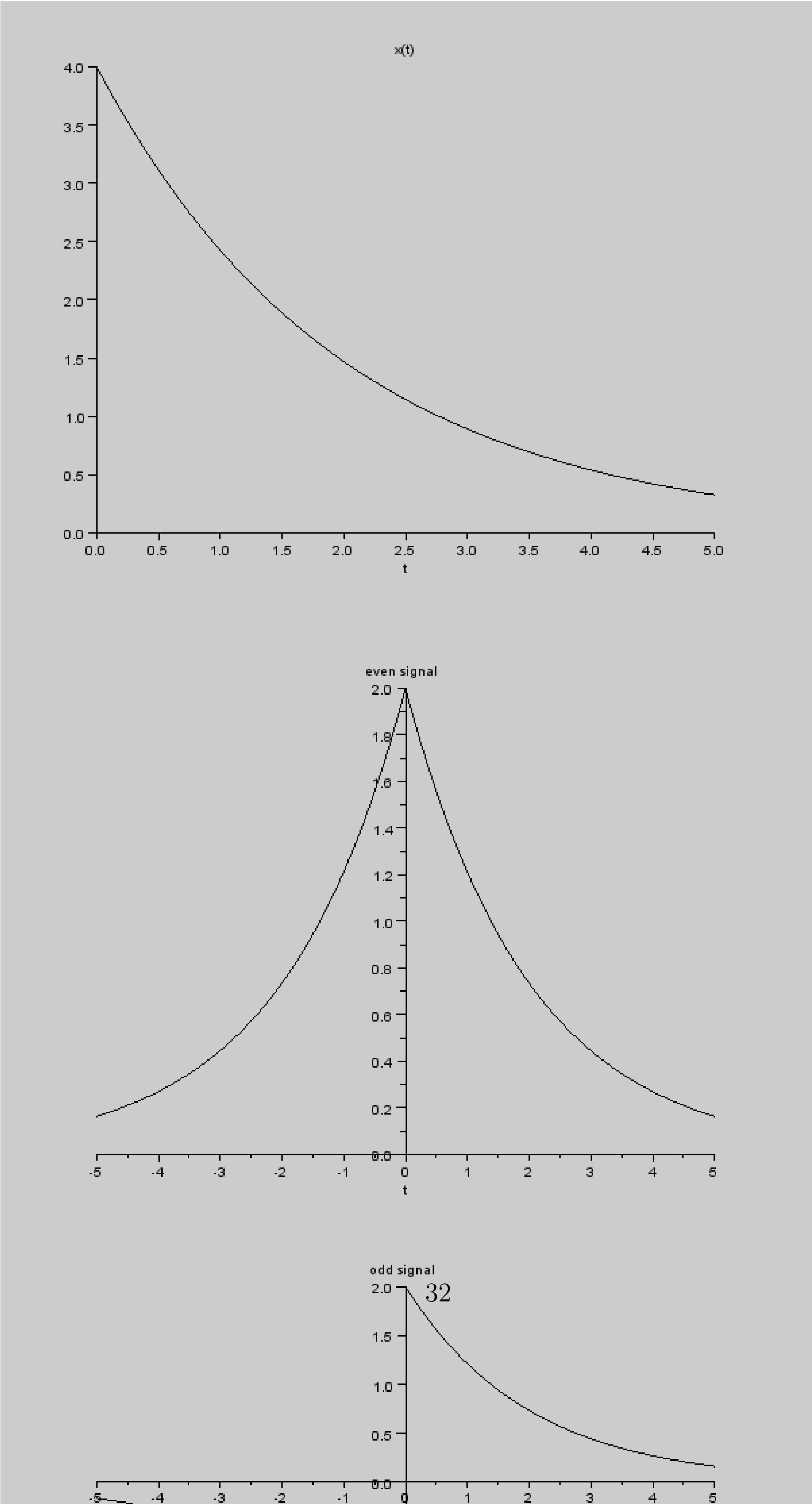

x(t)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
7
8
9
10
t
[x(t)+x(-t)]/2=even
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10
t
[x(t)-x(-t)]/2=odd
2.0
1.5
1.0
0.5
0.0

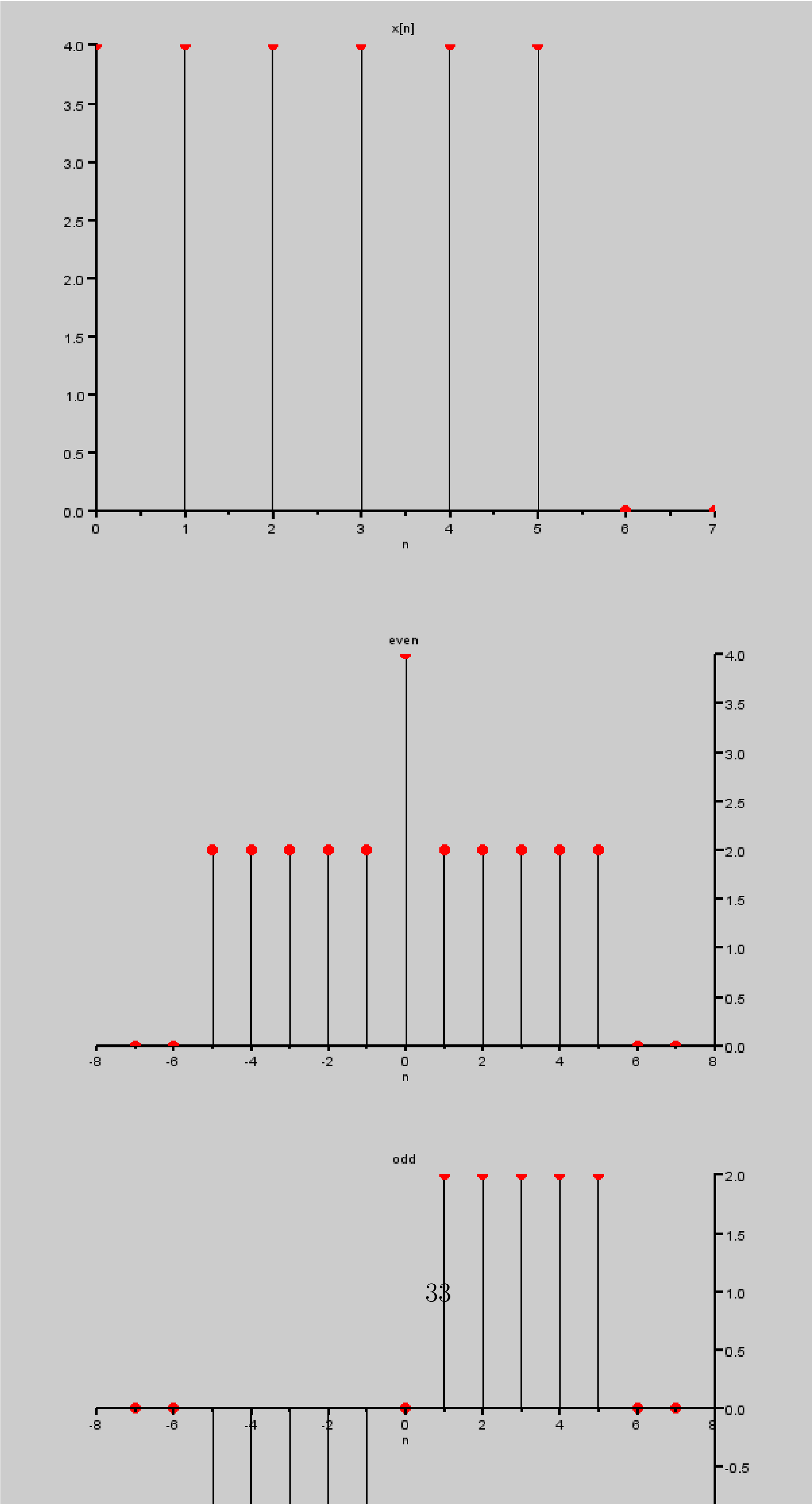
Scilab code Exa 1.5.a even and odd components

```
//ex_5 even and odd signals of x(t)
clear ;
clear x;
clear t;
clc;
close;
t = 0:1/100:5;
for i = 1:length(t)
  x(i) = (4/5)*t(i) ;
end
for i = length(t)+1:2*length(t)
x(i) = 0;
end
figure
a=gca();
t1=0:1/100:10;
plot(t1,x(1:$-1))
xtitle('x(t)')
figure
a=gca();
t3=0:1/100:10;
plot(t3,x(1:$-1)/2)
xtitle('[x(t)+x(-t)]/2=even')
t2=-10:1/100:0;
plot(t2,x($:-1:2)/2)
a.y_location='origin'
figure
a=gca();
t4=0:1/100:10;
plot(t3,x(1:$-1)/2)
xtitle('[x(t)-x(-t)]/2=odd')
t5=-10:1/100:0;
plot(t5,-x($:-1:2)/2)
a.y_location='origin'
a.x_location='origin'
```

Scilab code Exa 1.5.b even and odd components

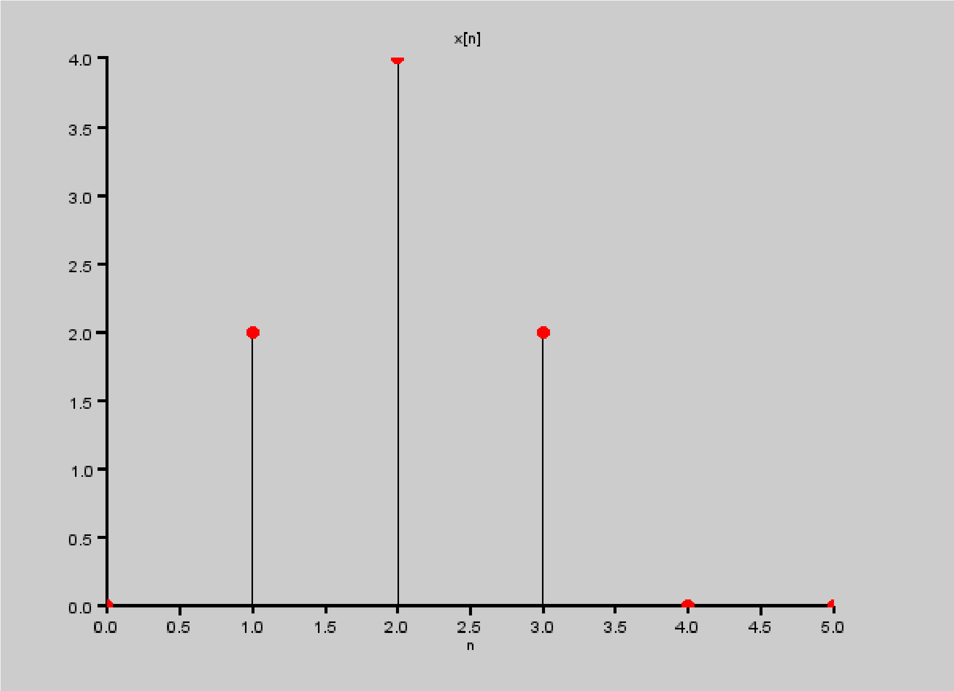
```
//ex_5 even and odd signals of x(t)
clear;
clc;
close;
t = 0:1/100:5;
x=4*exp(-0.5.*t)
figure
a=gca();
xtitle('x(t)')
plot2d(t,x)
figure
a=gca();
xtitle('even signal')
plot2d(t,x/2)
t1=-5:1/100:0;
plot2d(t1,x($:-1:1)/2)
a.y_location='origin'
figure
a=gca();
xtitle('odd signal')
plot2d(t,x/2)
t1=-5:1/100:0;
plot2d(t1,-x($:-1:1)/2)
a.y_location='origin'
a.x_location='origin'
```

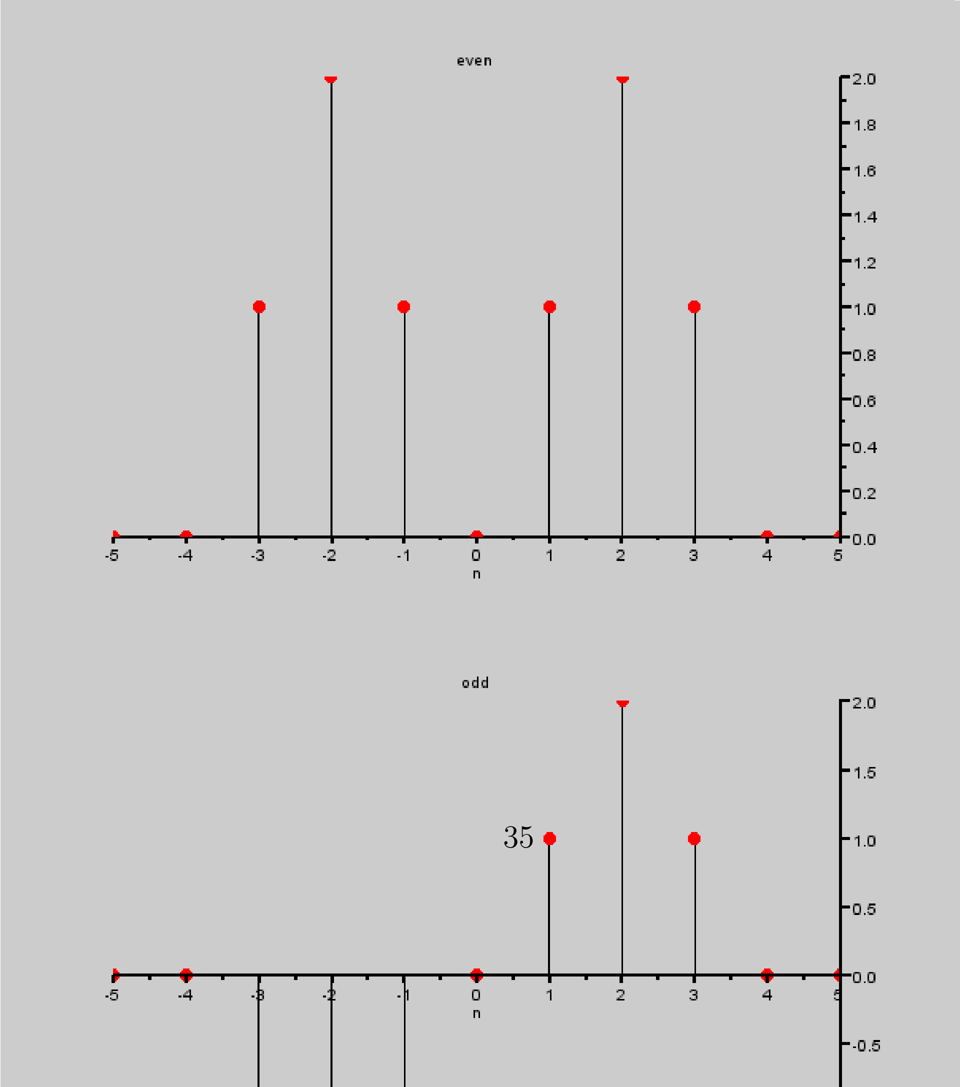

x(t)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
t
even signal
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-5 -4 -3 -2 -1 0 1 2 3 4 5
t
odd signal
2.0
1.5
1.0
0.5
0.0
-5 -4 -3 -2 -1 0 1 2 3 4 5


x[n]
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
7
n
even
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-8
-6
-4
-2
0
2
4
6
8
n
odd
2.0
1.5
1.0
0.5
0.0
-0.5
-8
-6
-4
-2
0
2
4
6
8
n

Scilab code Exa 1.5.c even and odd components

```
//ex_5 even and odd signals of x(t)
clear;
clc;
close;
t=0:7;
x=[4 4 4 4 4 4 0 0];
figure
a=gca();
plot2d3(t,x)
plot(t,x,'r.')
xtitle('x[n]')
a.thickness=2;
t1=-7:0;
figure
a=gca();
t2=-7:7;
y=[x($:-1:2)./2  x(1)  x(2:8)./2 ]
plot2d3(t2,y)
plot(t2,y,'r.')
xtitle('even')
a.y_location='right';
a.thickness=2;
figure
a=gca();
z=[-x($:-1:2)./2  0  x(2:8)./2 ]
plot2d3(t2,z)
plot(t2,z,'r.')
xtitle('odd')
a.y_location='right';
a.x_location='origin';
a.thickness=2;
```


x[n]
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
n


even
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-5
-4
-3
-2
-1
0
1
2
3
4
5
n
odd
2.0
1.5
1.0
0.5
0.0
-0.5
-5
-4
-3
-2
-1
0
1
2
3
4
5
n

Scilab code Exa 1.5.d even and odd components

```
//ex_5 d even and odd signals of x(t)
clear;
clc;
close;
t=0:5;
x=[ 0 2 4 2 0 0];
figure
a=gca();
plot2d3(t,x)
plot(t,x,'r.')
xtitle('x[n]','n')
a.thickness=2;
figure
a=gca();
t2=-5:5 ;
y=[ x($:-1:2)./2  x(1)  x(2:6)./2 ]
plot2d3(t2,y)
plot(t2,y,'r.')
xtitle('even','n')
a.y_location='right';
a.thickness=2;
figure
a=gca();
z=[ -x($:-1:2)./2  0  x(2:6)./2  ]
plot2d3(t2,z)
plot(t2,z,'r.')
xtitle('odd','n')
a.y_location='right';
a.x_location='origin';
a.thickness=2;
```

Scilab code Exa 1.6 even and odd components

```
//ex_6 even and odd signal of e^jt
clear;
clc;
close;
t = 0:1/100:5;
x=exp(%i.*t);
y=exp(-%i.*t);
even=x./2+y./2;
odd=x./2-y./2;
figure
a=gca();
plot2d(t,even)
a.x_location='origin'
xtitle('even','t')
figure
a=gca();
plot2d(t,odd./%i)
a.x_location='origin'
xtitle('odd','t')
```

Scilab code Exa 1.9 periodicity of exponential signal

```
//ex_9 to show that e^iwt is periodic with T=2*pi/W0
clear;
clc;
close;
t=0:1/100:10;
w0=1;
T=2*%pi/w0;
x=exp(%i*w0*t);
y=exp(%i*w0*(t+T));
```

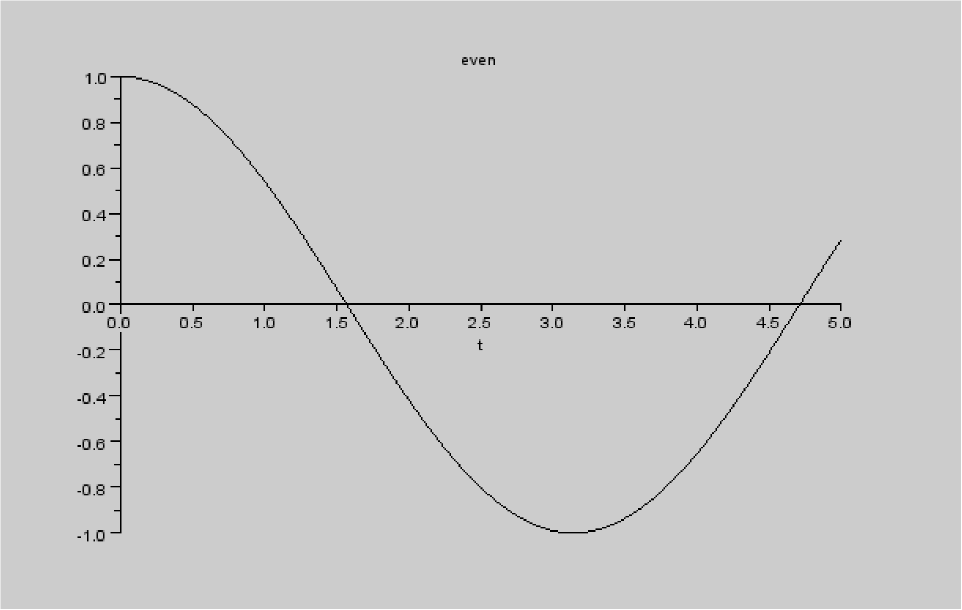


Figure 1.13: even and odd components

```
if ceil(x)==ceil(y) then
    disp('e^iwt is periodic with T=2*pi/W0')
else
    disp('nonperiodic')
end
```

Scilab code Exa 1.10 periodicity of sinusoidal signal

```
clear;
clc;
close;
t=0:1/100:10;
w=1;
theta=%pi/3;
T=2*%pi/w;
x=cos(t*w+theta);
```

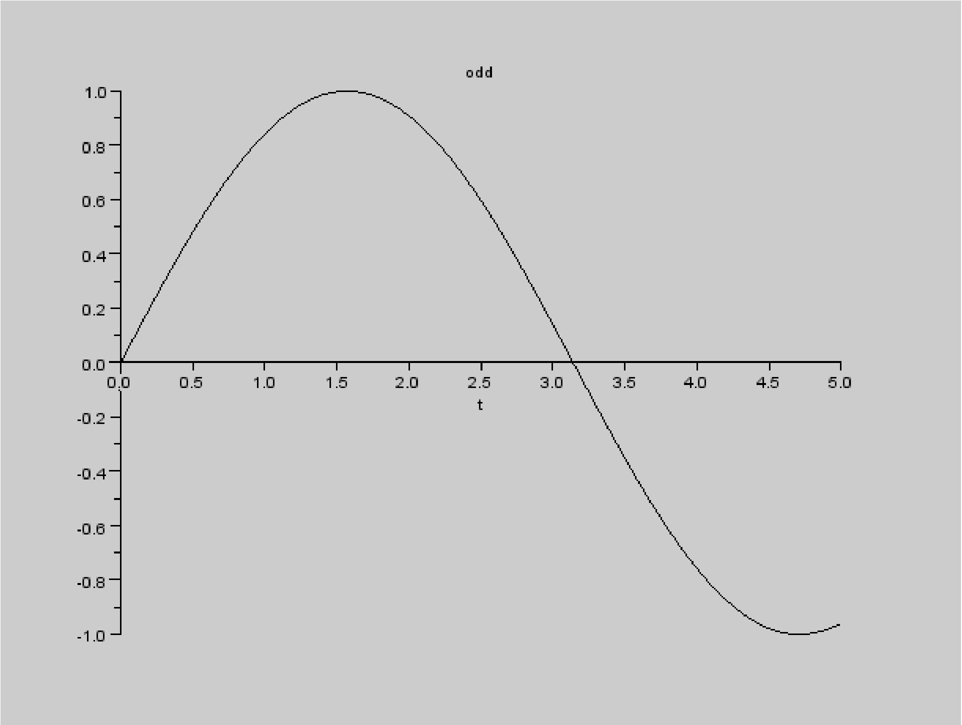


Figure 1.14: even and odd components

```
y=cos((t+T)*w+theta);
if ceil(x)==ceil(y) then
    disp('cos(wo*t+theta) is periodic with T=2*pi/W0')
else
    disp('nonperiodic')
end
```

Scilab code Exa 1.11 periodicity of exponential sequence

```
clear;
clc;
close;
n=0:100;
w0=1;
N=2*%pi/w0;
x=exp(%i*w0*n);
y=exp(%i*w0*(n+N));
if ceil(x)==ceil(y) then
    disp('e^iwn is periodic with N=2*pi/W0')
else
    disp('nonperiodic')
end
```

Scilab code Exa 1.16 fundamental period

```
//ex_16 to check if a function is periodic
clear;
clc;
close;
//x(t)=cos(t+pi/4)
w=1
t=2*%pi;//t=2*pi/w
```

```
disp(t,'a')
//x(t)=sin(2pi*t/3)
w=2*%pi/3;
t=2*%pi/w;//check if t is rational
if t==ceil(t) then
    disp(t,'b');
else
    disp('non periodic','b')
end
//x(t)=cos(pi*t/3)+sin(pi*t/4)
w1=%pi/3;
w2=%pi/4;
t1=2*%pi/w1;
t2=2*%pi/w2;
t=lcm([t1 t2/2]);

if t==ceil(t) then
    disp(t,'c');
else
    disp('non periodic')
end
//x(t)=cos(t)+sin(sqrt(2)*t)
w1=1;
w2=sqrt(2);
t1=2*%pi/w1;
t2=2*%pi/w2;
t=lcm([t1 t2]);
if t==ceil(t) then
    disp(t,'d');
else
    disp('non periodic')
end
//x(t)=(sin(t))^2=(1-cos(2*t))/2
w=2;
t=2*%pi/w;
disp(t,'e')
//x(t)=e^(%i*(%pi/2)*t-1)
w=%pi/2;
```

```
t=2*%pi/w;
disp(t,'f')
//x[n]=e^(%i*(%pi/4))
w=%pi/4;
N=2*%pi/w;
disp(N,'g')
//x[n]=cos(1*n/4)
w=1/4;
N=2*%pi/w;
if N==ceil(N) then
    disp(N,'h');
else
    disp('non periodic','h')
end
//x[n]=cos(%pi*n/3)+sin(%pi*n/4)
w1=%pi/3;
w2=%pi/4;
N1=2*%pi/w1;
N2=2*%pi/w2;
N=lcm([N1 N2/2]);
if N==ceil(N) then
    disp(N,'i');
else
    disp('non periodic','i')
end
//x[n]=(cos(%pi*n/8))^2=(1+cos(%pi*n/4))/2
w=%pi/4;
N=2*%pi/w;
disp(N,'j')
```

Scilab code Exa 1.21 unit step signal

```
//ex_21  to check if u(-t)={1 for t<0 and 0 for t>0}
clear;
clc;
```

```
close;
t=-10:1/100:10;
for i = 1:(length(t))
    if t(i)<0  then
        x(i)=0
    else
        x(i)=1
    end
end
figure
a=gca();
plot(t,x)
xtitle('u(t)','t')
a.data_bounds=[-10,-1;10,2];
figure
a=gca();
plot(t,x($:-1:1))
xtitle('u(-t)','t')
a.data_bounds=[-10,-1;10,2];
```

Scilab code Exa 1.22 continuous time signal

```
//ex_22 product of x(t) and unit step function
clear x;
clear t;
clear;
clear y;
clc;
close;
t=-1:1/100:2
for i=1:(length(t))
```

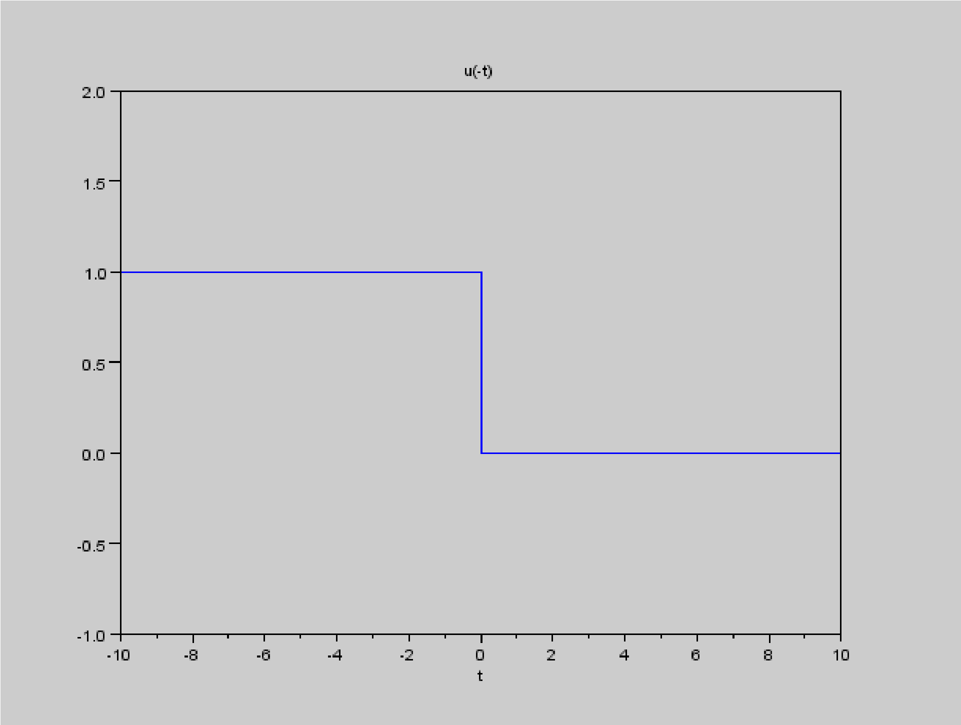


Figure 1.15: unit step signal

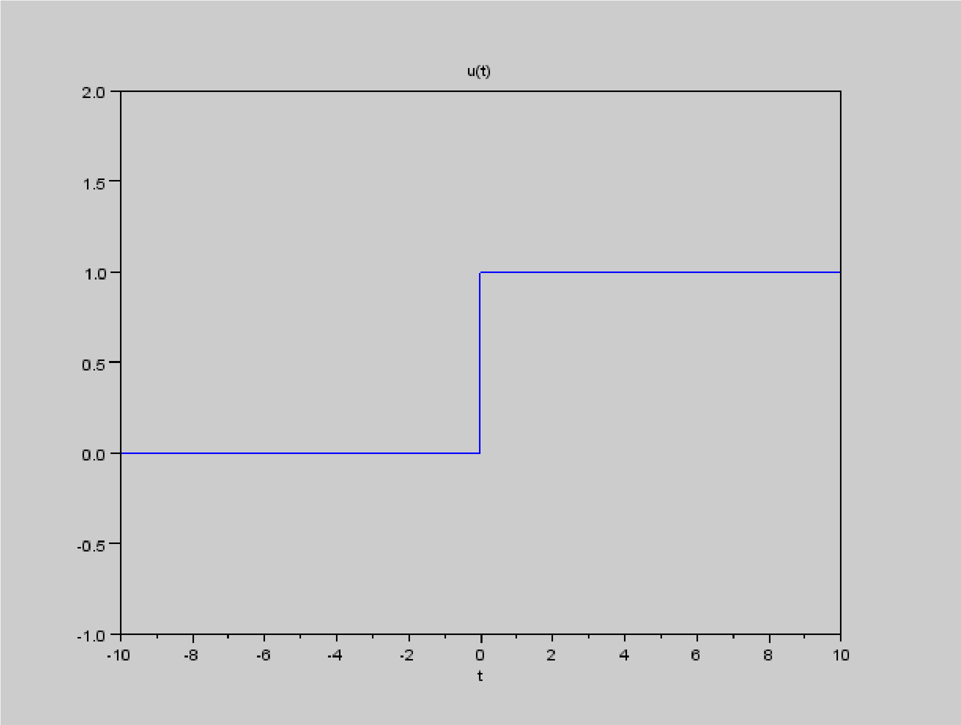


Figure 1.16: unit step signal

```
    if t(i)<0 then
        x(i)=t(i)+1;
    elseif t(i)<1
        x(i)=1
    else
        x(i)=2
    end
end
figure
a=gca();
plot2d(t,x)
a.y_location='origin'
xtitle('x(t)','t')
//a.x(t)*u(1-t)
for i = 1:(length(t))
    if t(i)<1  then
        u1(i)=1
    else
        u1(i)=0
    end
end
y=x.*u1
figure
a=gca();
plot2d(t,y)
a.y_location='origin'
xtitle('x(t)*u(1-t)','t')
for i = 1:(length(t))
    if t(i)<1 & t(i)>0 then
        u2(i)=1
    else
        u2(i)=0
    end
end
y=x.*u2;
figure
a=gca();
plot2d(t,y)
```

```
a.y_location='origin'
xtitle('x(t)*u(t-1)','t')
for i = 1:(length(t))
    if t(i)==3/2 then
        z(i)=x(i)
    else
        z(i)=0
    end
end
figure
a=gca();
plot2d3(t,z)
poly_1=a.children.children;
poly_1.thickness=3;
a.y_location='origin'
xtitle('x(t)*delta(t-3/2)','t')
```

Scilab code Exa 1.23 discrete time signal

```
//ex_23 product of discrete signal and unit step function
clear;
clc;
close;
t=-3:3;
x=[3 2 1 0 1 2 3];
//u[1-n]
for i=1:length(t);
    if t(i)<=1 then
```

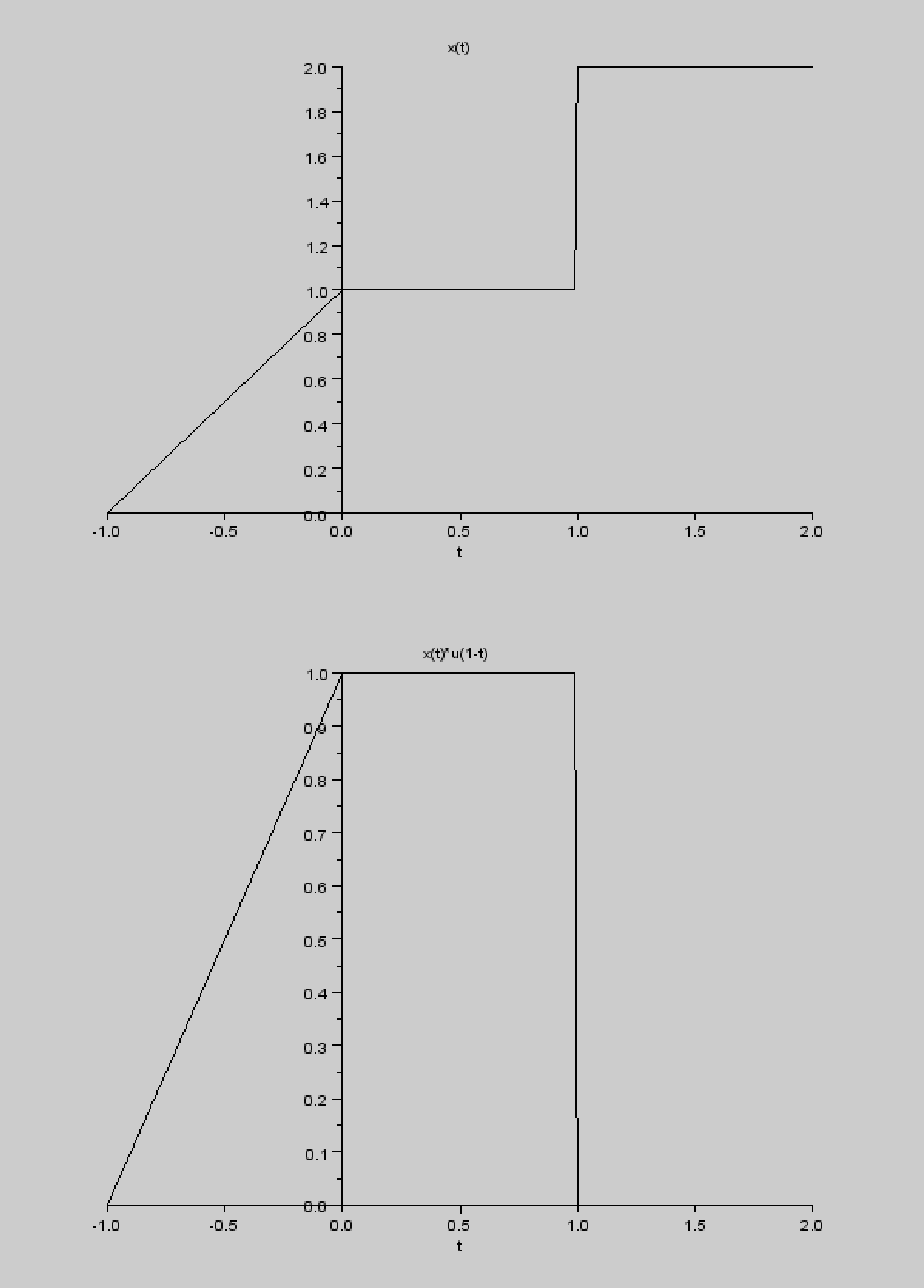


Figure 1.17: continuous time signal

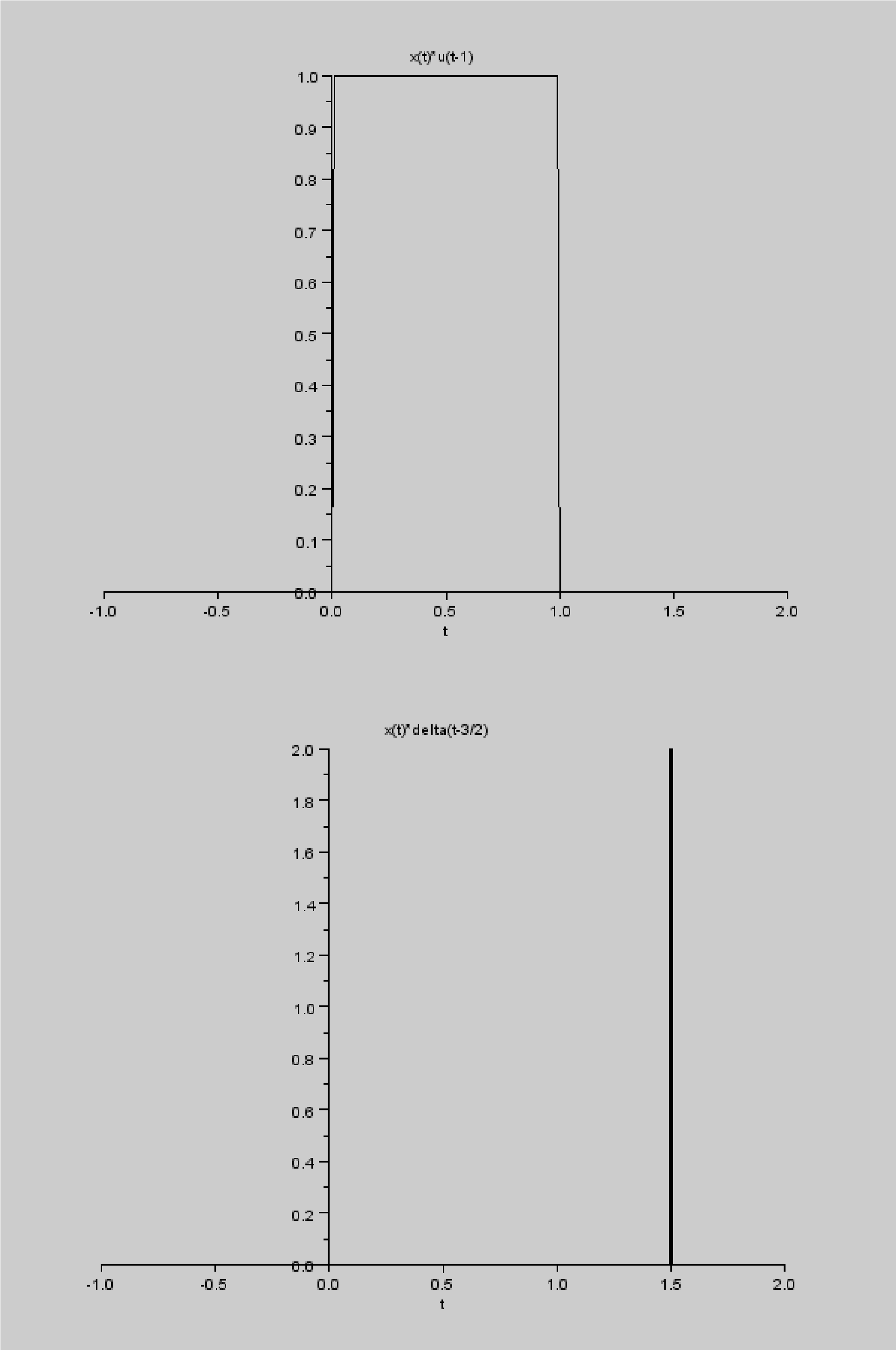
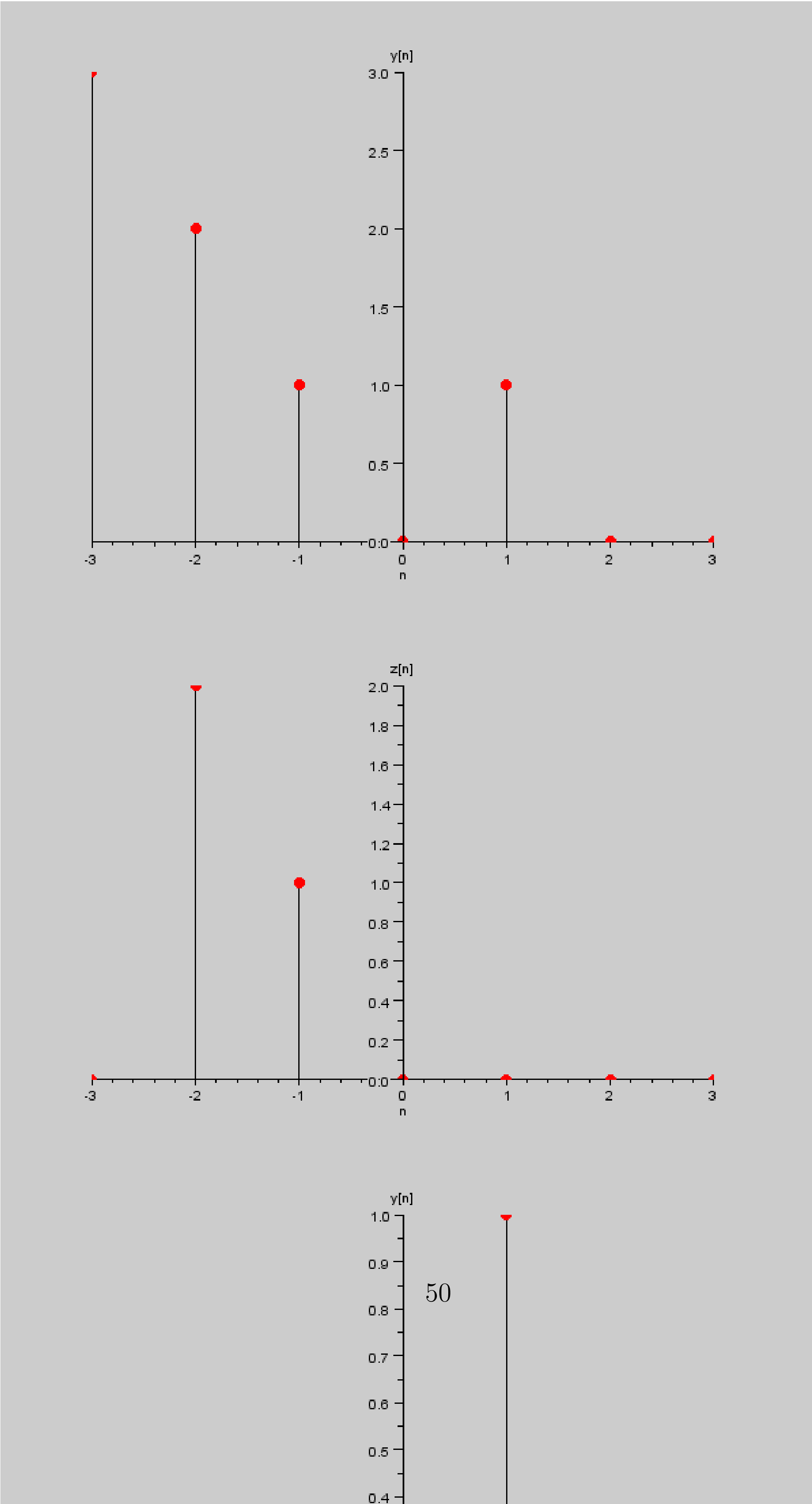


Figure 1.18: continuous time signal


y[n]
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-3
-2
-1
0
1
2
3
n
z[n]
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-3
-2
-1
0
1
2
3
n
y[n]
1.0
0.9
0.8
0.7
0.6
0.5
0.4

```
        u1(i)=1;
    else
        u1(i)=0;
    end
end
y=x.*u1';
figure
a=gca();
plot2d3(t,y)
plot(t,y,'r.')
xtitle('y[n]','n')
a.y_location='origin'
//u[n+2]-u[n]
for i=1:length(t);
    if t(i)<1 & t(i)>=-2 then
        u2(i)=1;
    else
        u2(i)=0;
    end
end
z=x.*u2';
figure
a=gca();
plot2d3(t,z)
plot(t,z,'r.')
xtitle('z[n]','n')
a.y_location='origin'
//$[n-1]
for i=1:length(t);
    if t(i)==1 then
        del(i)=1;
    else
        del(i)=0;
    end
end
p=x.*del';
figure
a=gca();
```

```
plot2d3(t,p)
plot(t,p,'r.')
xtitle('y[n]','n')
a.y_location='origin'
```

Scilab code Exa 1.31 first derivative of the signals

```
clear;
close;
clc;
t=-10:0.1:10;
a=2;
//x(t)=u(t)-u(t-a)
x=[zeros(1,find(t==0)-1) ones(1,find(t==a)-find(t==0)+1) zeros(1,length(t)-find(t==a))];
subplot(2,1,1)
plot(t,x)
xtitle('x(t)','t')
subplot(2,1,2)
plot2d3(t(1:$-1),diff(x))
xtitle('diff(x(t))','t')
//x(t)=t*(u(t)-u(t-a))
xb=t.*x;
figure
subplot(2,1,1)
plot(t,xb)
xtitle('x(t)','t')
subplot(2,1,2)
plot2d(t(1:$-1),diff(xb))
xtitle('diff(x(t))','t')
//x(t)=sgn(t)
x=[-ones(1,find(t==0)-1) ones(1,length(t)-find(t==0)+1)];
figure
subplot(2,1,1)
```

```
plot(t,x)
xtitle('x(t)','t')
subplot(2,1,2)
plot2d(t(1:$-1),diff(x))
xtitle('diff(x(t))','t')
```

Scilab code Exa 1.35 linearity

```
//ex_35 to check if a system is linear or non-linear
clear;
clc;
close;
x1=2;
x2=3;
y1=x1*x1;
y2=x2*x2;
y=y1+y2;
z=(x1+x2)*(x1+x2);
if z==y then
    disp('the system  is linear')
else
    disp("the system is nonlinear")
end
```

Scilab code Exa 1.36 memoryless causal stable system

```
// ex_36 check if y[n] = x[n-1] memoryless ,causal , linear , time variant
clear;
```

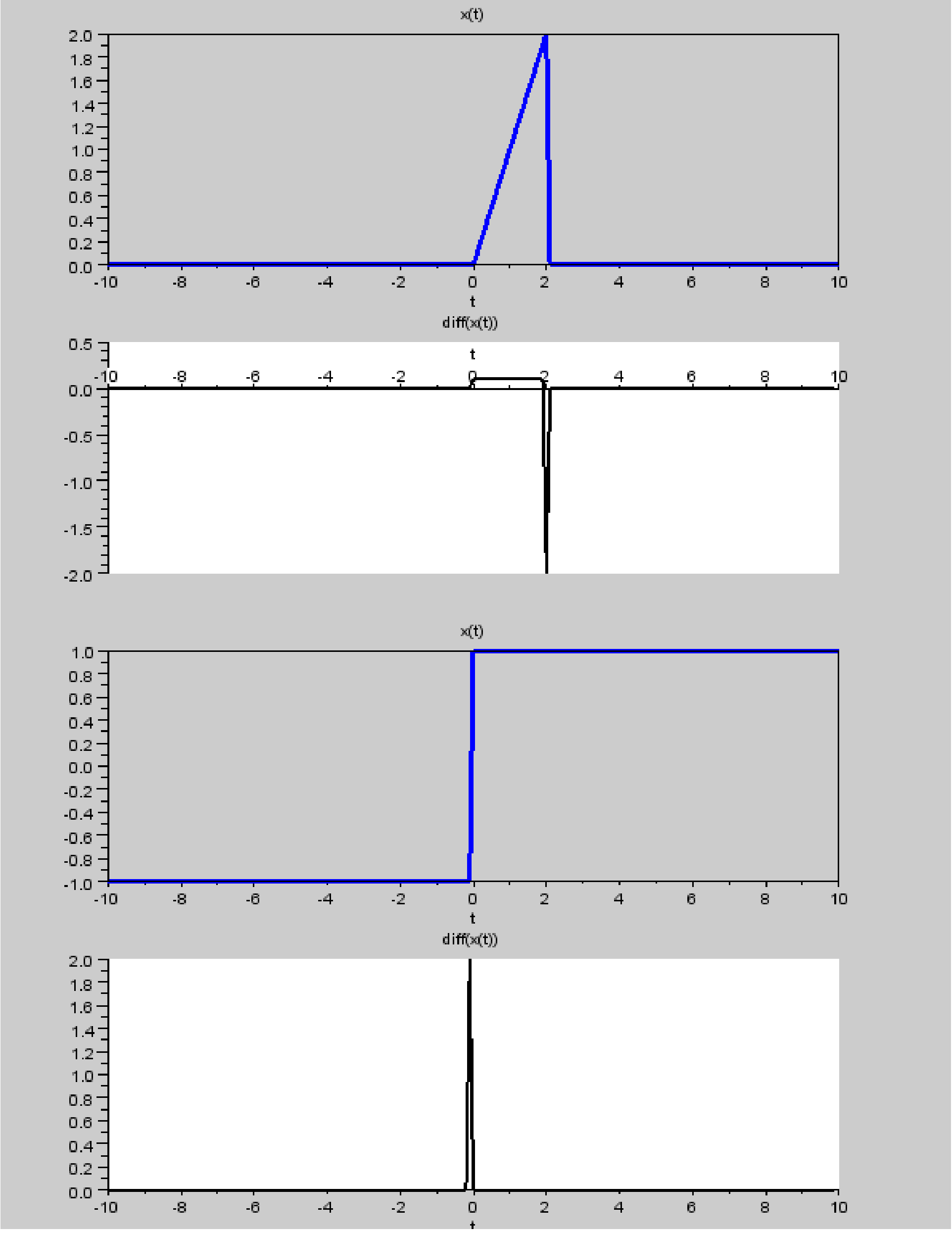


Figure 1.20: first derivative of the signals

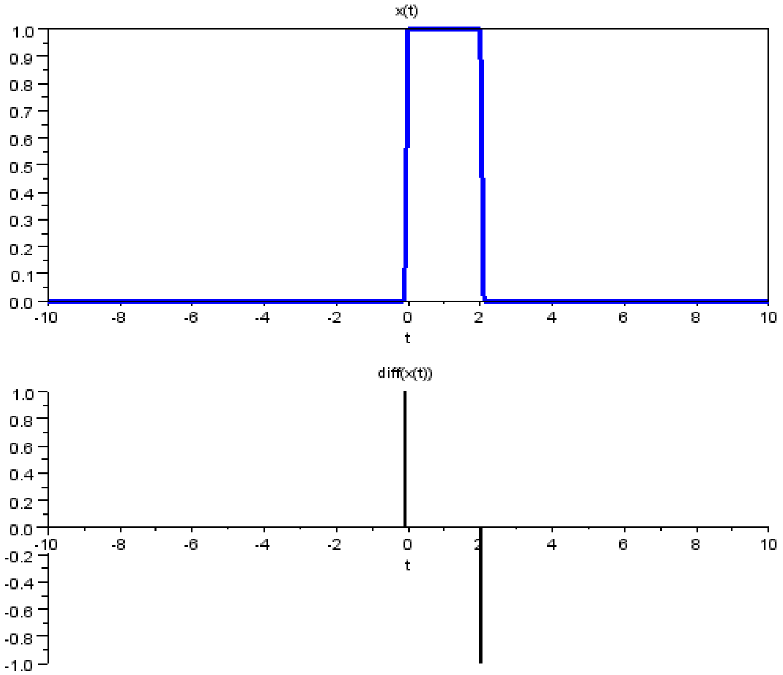


Figure 1.21: first derivative of the signals

```
clc;
s = 2;//shift
T = 20; //lenght of signal
x(1)=1;
for n = 2:T
  x(n) = n;
  y(n) = x(n-1);
end
if y(2)==x(2) then
    disp("memoryless")
else
    disp("not memoryless")
end
//causal if it does'nt depend on future
if y(2)==x(2) | y(2)==x(1) then
    disp('causal')
else
    disp('non casual')
end
x1=x;
y1=y;
x2(1)=2;
for n = 2:T
  x2(n) = 2;
  y2(n) = x2(n-1);
end
z=y1+y2;
for n = 2:T
  y3(n) = (x2(n-1)+x1(n-1));
end
if z==y3 then
    disp('linear')
else
    disp("nonlinear")
end
Ip = x(T-s);
Op = y(T-s);
if(Ip == Op)
```

```
  disp(' Time In-variant system');
else
  disp('Time Variant system');
end
Max =20;
dd=1;
for n=2:T
    if y(n)>Max then
        dd=0
    end
end
if dd==0
  disp('unstable')
 else
  disp('stable');
end
```

Scilab code Exa 1.38 memoryless causal time invariant system

```
// ex_38 check if y[n] = n.x[n] is memoryless, causal, linear, time-varaint
clear;
clc;
s = 2;//shift
T = 20; //lenght of signal
for n = 1:T
  x(n) = n;
  y(n) = n*x(n);
end
if y(1)==x(1) then
    disp("memoryless and causal")
else
    disp("noncausal")
end
x1=x;
```

```
y1=y;
for n = 1:T
  x2(n) = 2;
  y2(n) = n*x2(n);
end
z=y1+y2;
for n = 1:T
  y3(n) = n*(x2(n)+x1(n));
end
if z==y3 then
    disp('linear')
else
    disp("nonlinear")
end
Ip = x(T-s);
Op = y(T-s);
if(Ip == Op)
  disp(' Time In-variant system');
else
  disp('Time Variant system');
end
Max = 50;
S = 0;
for n=1:T
  S = S+y(n);
end
if (S >Max)
  disp('unstable')
 else
  disp('stable');
end
```

Scilab code Exa 1.39 time invariancy

```
//ex_39check if y[n]=x[k*n] is time invariant
```

```
clear;
clc;
s = 2;//shift
x=[1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9];
T=length(x);
k=3;
for n=1:T/k
    y(n)=x(k*n);
end
T=5;
Ip = x(T-s);
Op = y(T-s);
if(Ip == Op)
  disp(' Time In-variant system');
else
  disp('Time Variant system');
end
```

Scilab code Exa 1.41 linearity

```
clear;
close;
clc;
n=-2:4;
x1=[0 0 0 0 2 0 0];
y1=[0 0 0 0 1 2 0];
x2=[0 1 0 0 0 0 0];
y2=[0 2 1 0 0 0 0];
x3=[0 0 0 1 2 0 0];
y3=[0 0 0 2 3 1 0];
subplot(3,2,1);plot2d3(n,x1);plot(n,x1,'r.');xtitle('x1')
subplot(3,2,2);plot2d3(n,y1);plot(n,y1,'r.');xtitle('y1')
subplot(3,2,3);plot2d3(n,x2);plot(n,x2,'r.');xtitle(
'x2')
subplot(3,2,4);plot2d3(n,y2);plot(n,y2,'r.');xtitle('y2')
subplot(3,2,5);plot2d3(n,x3);plot(n,x3,'r.');xtitle('x3')
subplot(3,2,6);plot2d3(n,y3);plot(n,y3,'r.');xtitle('y3')
disp("it can be seen that x3[n]=x1[n]+x2[n-2] therefore for linear system         y3[n]=y1[n]+y2[n-2]")
figure
subplot(4,1,1);plot2d3(n,y1);plot(n,y1,'r.');xtitle('y1')
subplot(4,1,2);plot2d3(n+2,y2);plot(n+2,y2,'r.');xtitle('y2[n-2]')
subplot(4,1,3);plot2d3(n,y1+[0 0 y2(1:find(n+2==4))]);plot(n,y1+[0 0 y2(1:find(n+2==4))],'r.');xtitle('y1[n]+y2[n-2]')
subplot(4,1,4);plot2d3(n,y3);plot(n,y3,'r.');xtitle('y3[n]')
disp("from the figure y3[n]<>y1[n]+y2[n-2] therefore the system is not linear")
```

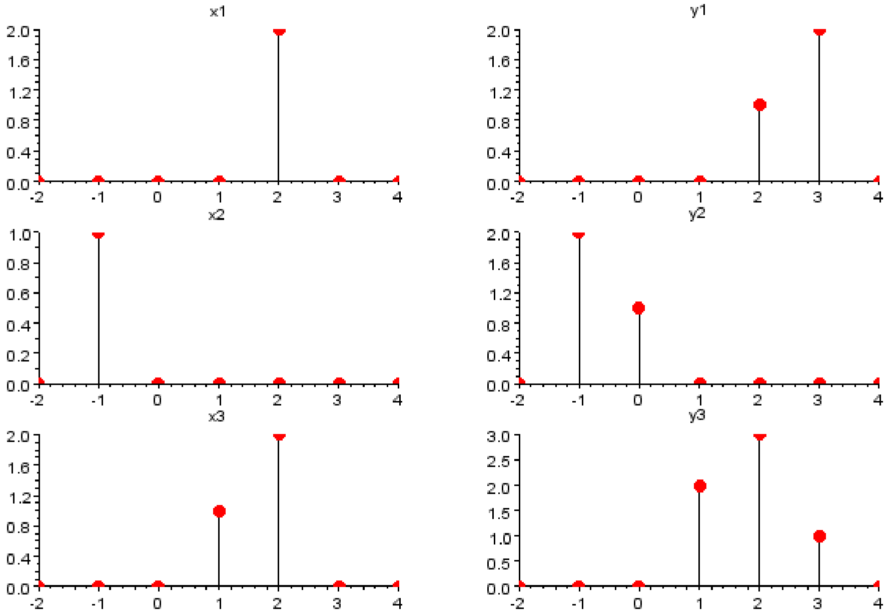


Figure 1.22: linearity

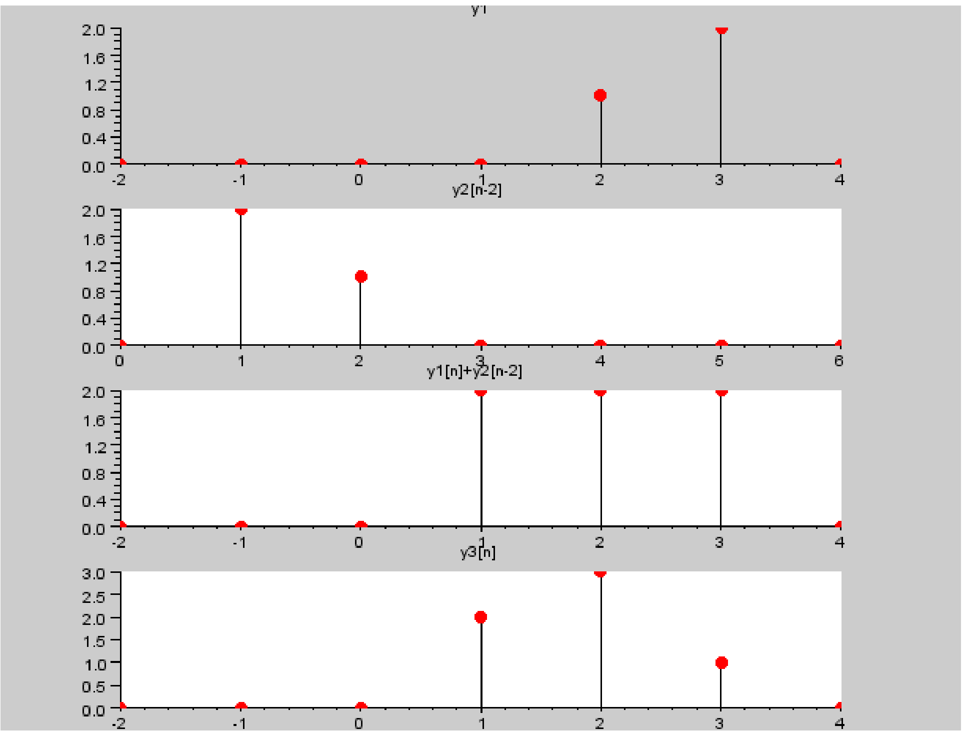


Figure 1.23: linearity

Chapter 2

Linear Time Invariant Systems

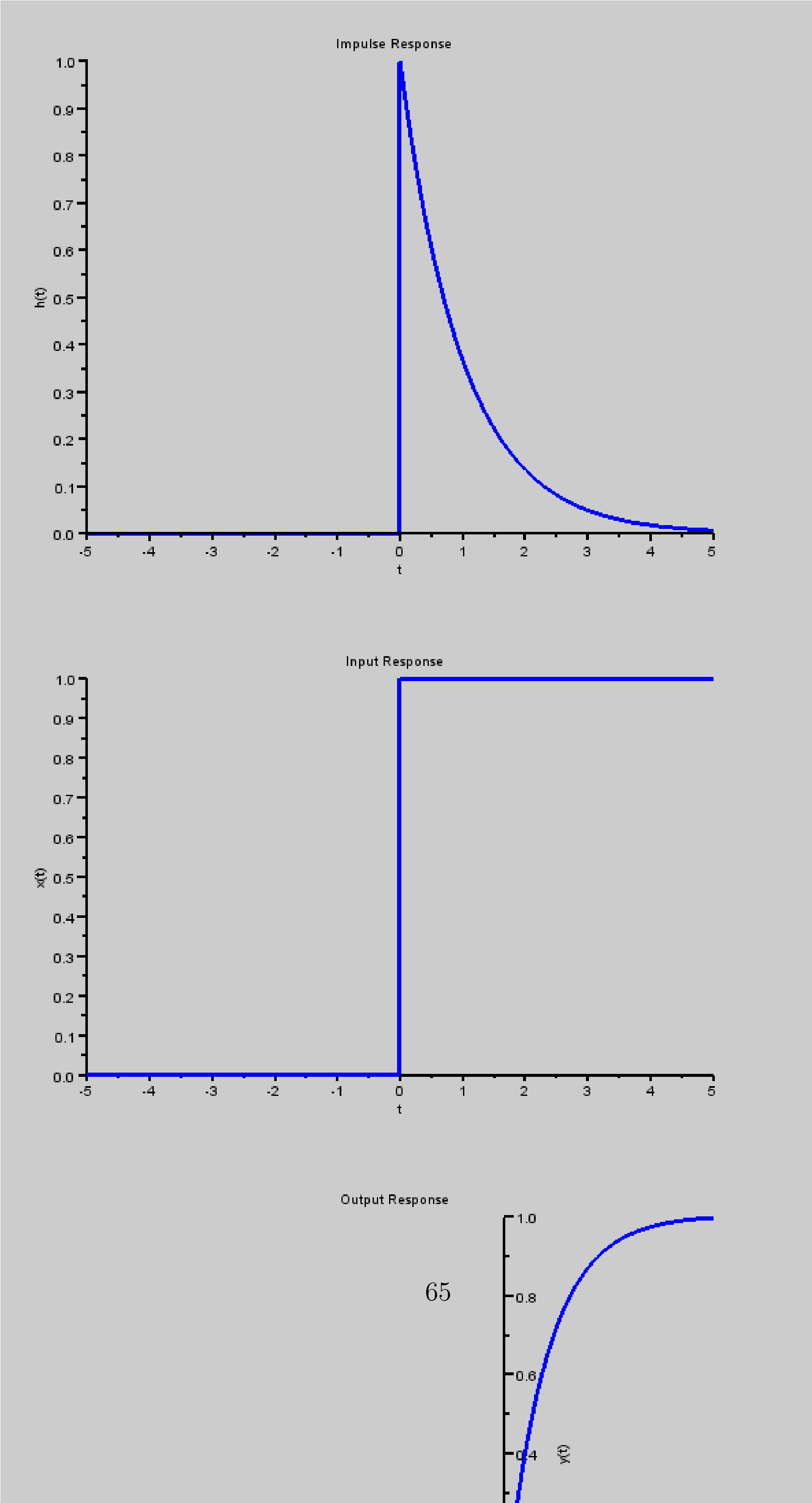
Scilab code Exa 2.4 output response

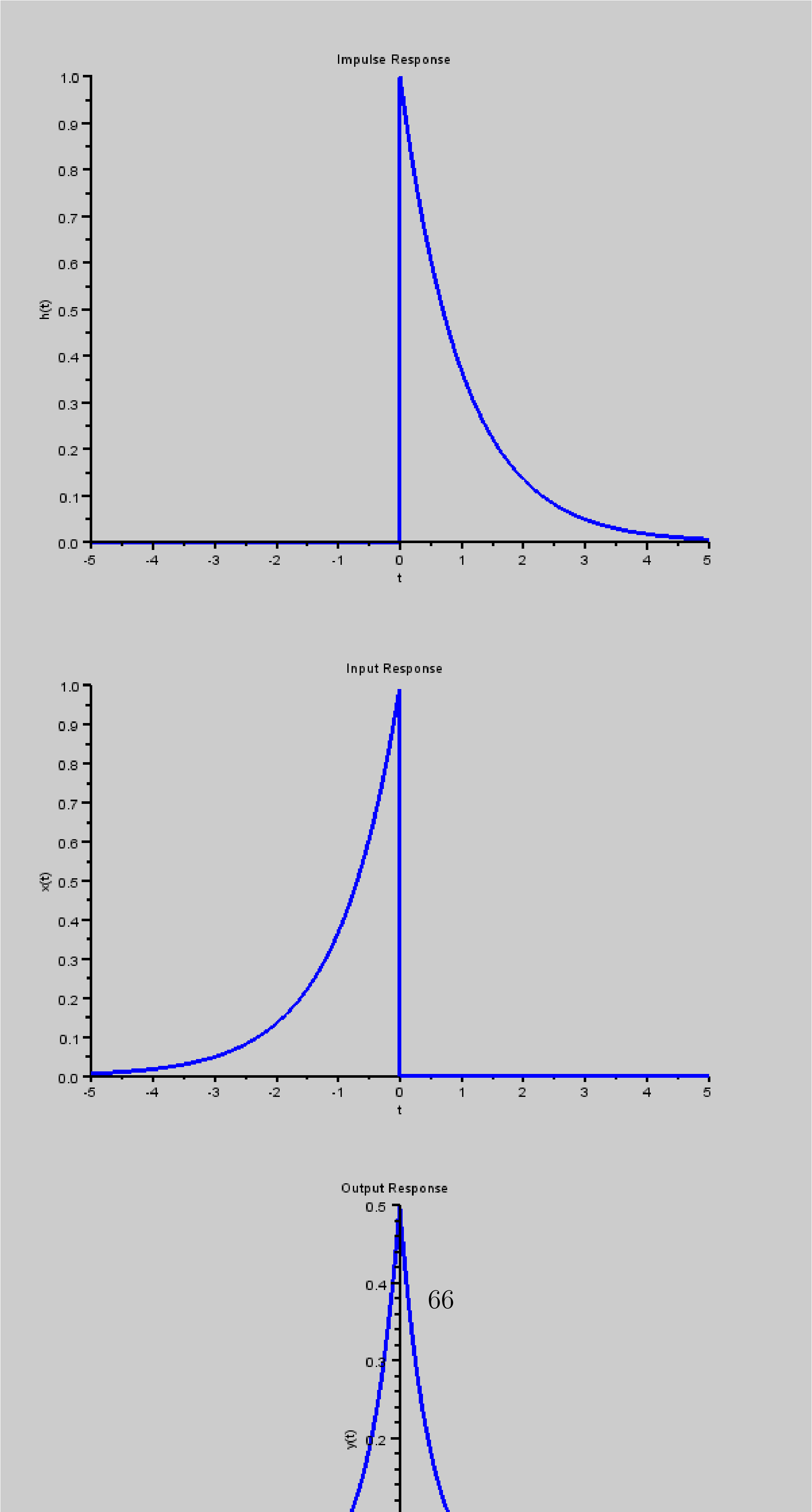
```
//Example 2.4:Convolution Integral
clear;
close;
clc;
t =-5:1/100:5;
for i=1:length(t)
    if t(i)<0 then
        h(i)=0;
        x(i)=0;
    else
        h(i)=exp(-t(i));
        x(i)=1;
    end
end
y = convol(x,h)./100;
figure
a=gca();
a.x_location=”origin”;
plot2d(t,h)
xtitle(’Impulse Response’,’t’,’h(t)’);
a.thickness = 2;
```

```
figure
a=gca();
plot2d(t,x)
xtitle('Input Response','t','x(t)');
a.thickness = 2;
figure
a=gca();
t1=-10:1/100:10;
a.y_location="origin";
a.x_location="origin";
d=find(t1==5);
plot2d(t1(1:d),y(1:d))
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Scilab code Exa 2.5 output response

```
//Example 2.4:Convolution Integral
clear;
close;
clc;
t =-5:1/100:5;
for i=1:length(t)
    if t(i)<0 then
        h(i)=0;
        x(i)=exp(t(i));
    else
        h(i)=exp(-t(i));
        x(i)=0;
    end
end
```

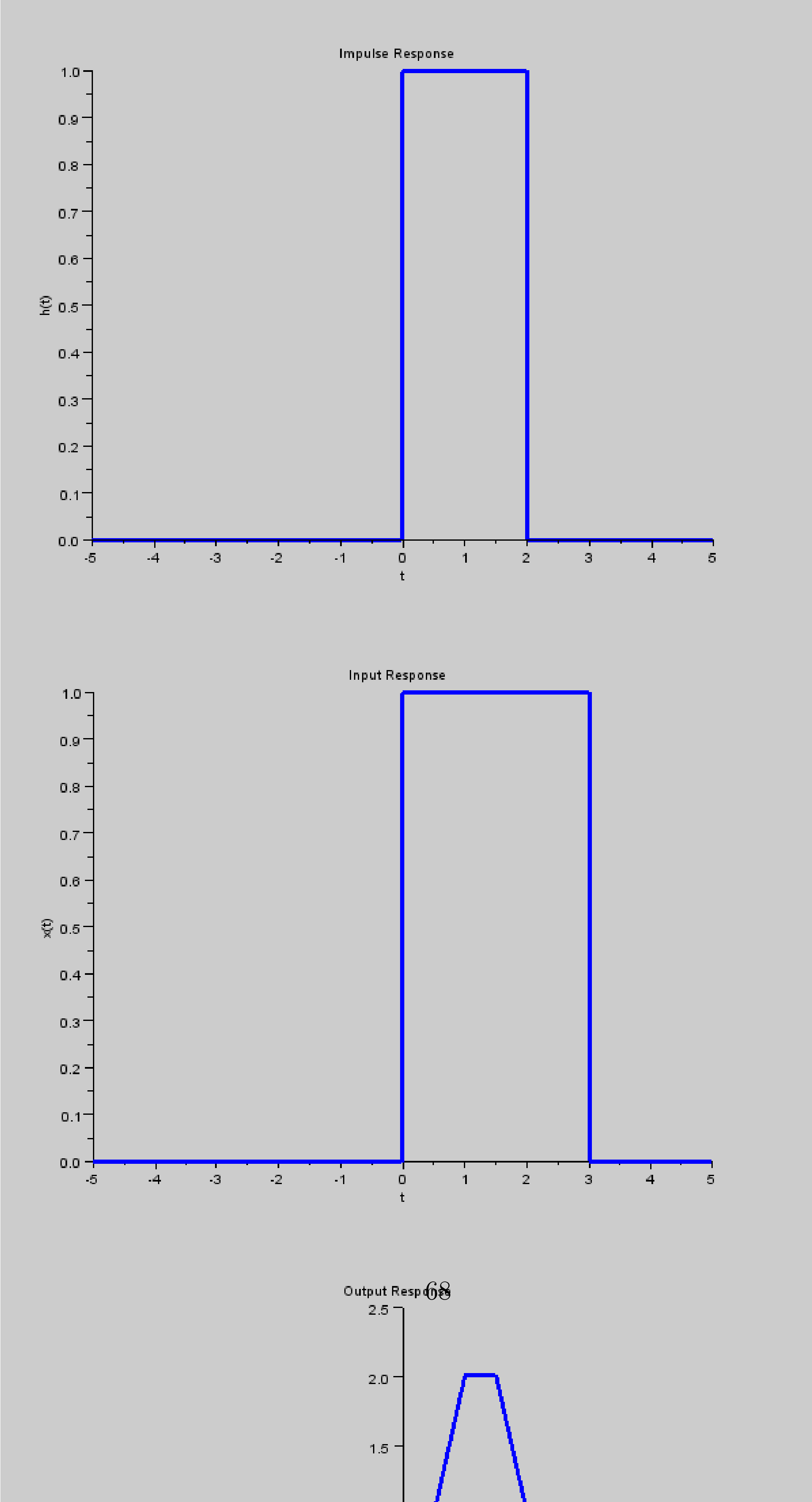

Impulse Response
h(t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-5
-4
-3
-2
-1
0
1
2
3
4
5
t
Input Response
x(t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-5
-4
-3
-2
-1
0
1
2
3
4
5
t
Output Response
1.0
0.8
0.6
0.4
y(t)


Impulse Response
h(t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-5
-4
-3
-2
-1
0
1
2
3
4
5
t
Input Response
x(t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-5
-4
-3
-2
-1
0
1
2
3
4
5
t
Output Response
y(t)
0.5
0.4
0.3
0.2

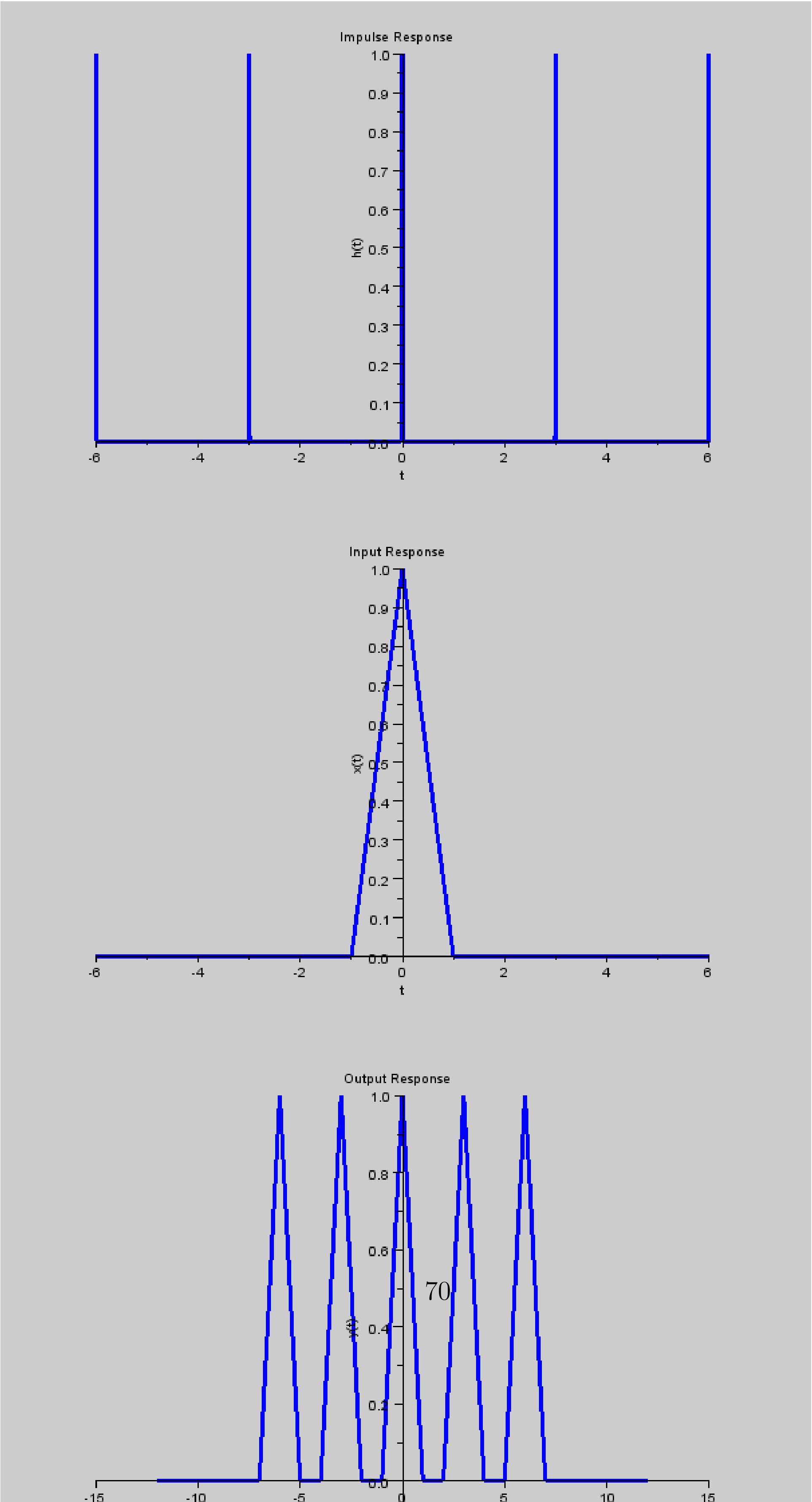
```
y = convol(x,h)./100;
figure
a=gca();
a.x_location="origin";
plot2d(t,h)
xtitle('Impulse Response','t','h(t)');
a.thickness = 2;
figure
a=gca();
plot2d(t,x)
xtitle('Input Response','t','x(t)');
a.thickness = 2;
figure
a=gca();
t1=-10:1/100:10;
a.y_location="origin";
plot2d(t1,y)
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Scilab code Exa 2.6 convolution

```
//Example 2.6:Convolution Integral
clear;
close;
clc;
t =-5:1/100:5;
for i=1:length(t)
    if t(i)<0 then
        h(i)=0;
        x(i)=0;
    elseif t(i)<=2
        h(i)=1;
```


Impulse Response
h(t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-5
-4
-3
-2
-1
0
1
2
3
4
5
t
Input Response
x(t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-5
-4
-3
-2
-1
0
1
2
3
4
5
t
Output Response
2.5
2.0
1.5

```
        x(i)=1;
    elseif t(i)<=3
        h(i)=0;
        x(i)=1;
    else
        h(i)=0;
        x(i)=0;
    end
end
y = convol(x,h)./100;
figure
a=gca();
a.x_location=”origin”;
plot2d(t,h)
xtitle(’Impulse Response’,’t’,’h(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
plot2d(t,x)
xtitle(’Input Response’,’t’,’x(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
t1=-10:1/100:10;
a.y_location=”origin”;
a.x_location=”origin”;
plot2d(t1,y)
xtitle(’Output Response’,’t’,’y(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```


Impulse Response
h(t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-6
-4
-2
0
2
4
6
t
Input Response
x(t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-6
-4
-2
0
2
4
6
t
Output Response
y(t)
1.0
0.8
0.6
0.4
0.2
0.0
-15
-10
-5
0
5
10
15

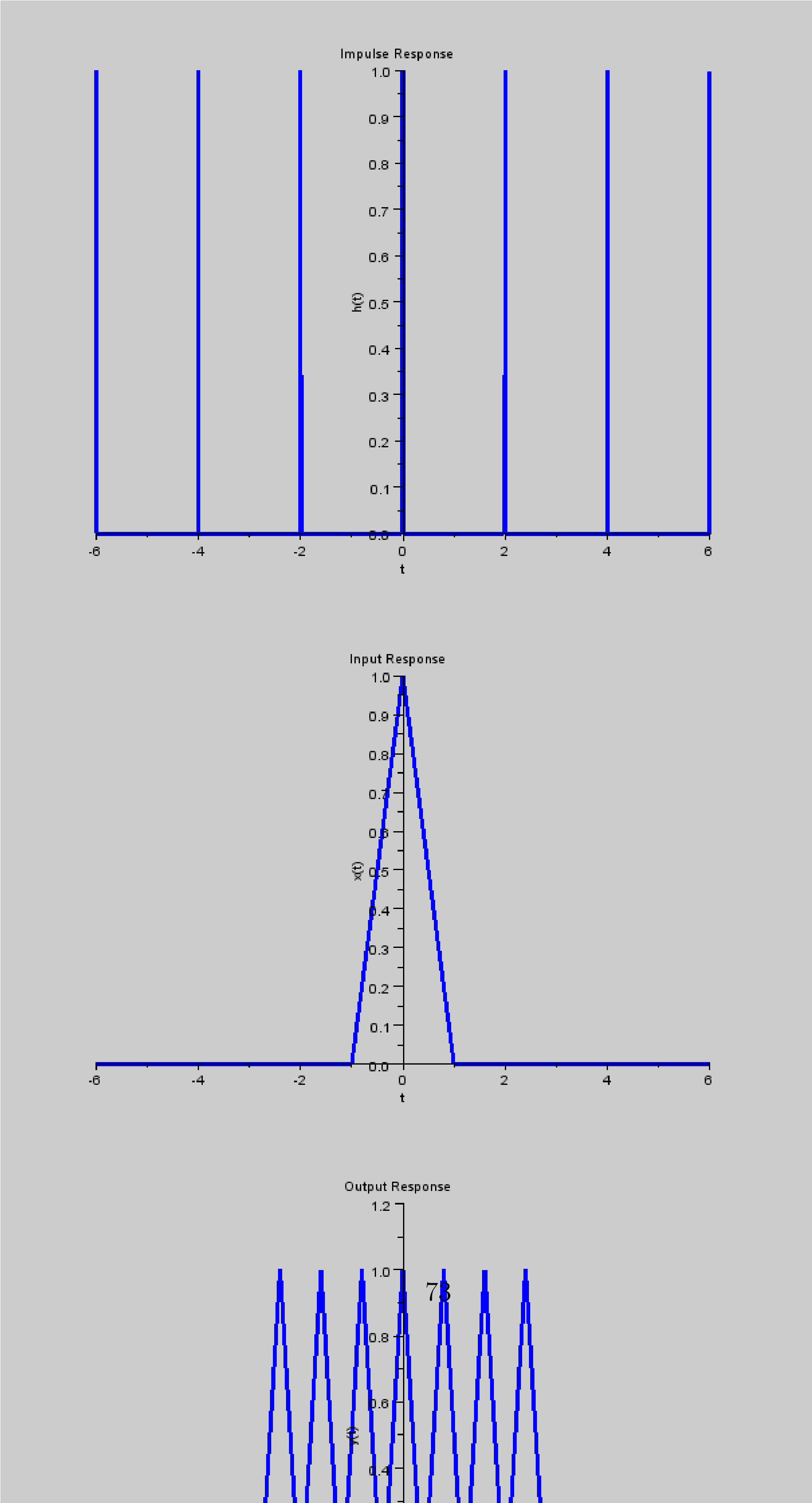
Scilab code Exa 2.7.a convolution of two rectangular pulse

```
//Example 2.7:Convolution Integral
clear;
close;
clc;
t =-6:1/100:6;
for i=1:length(t)
    if modulo(t(i),3)==0 then
        h(i)=1;
    else
        h(i)=0;
    end
    if t(i)<-1 then
        x(i)=0;
    elseif t(i)<0
        x(i)=1+t(i);
    elseif t(i)<1
        x(i)=1-t(i);
    else
        x(i)=0;
    end
end
y = convol(x,h);
figure
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d(t,h)
xtitle('Impulse Response','t','h(t)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d(t,x)
xtitle('Input Response','t','x(t)');
```

```
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
t1=-12:1/100:12;
a.y_location=”origin”;
a.x_location=”origin”;
plot2d(t1,y)
xtitle(’Output Response’,’t’,’y(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

Scilab code Exa 2.7.b convolution of two rectangular pulse

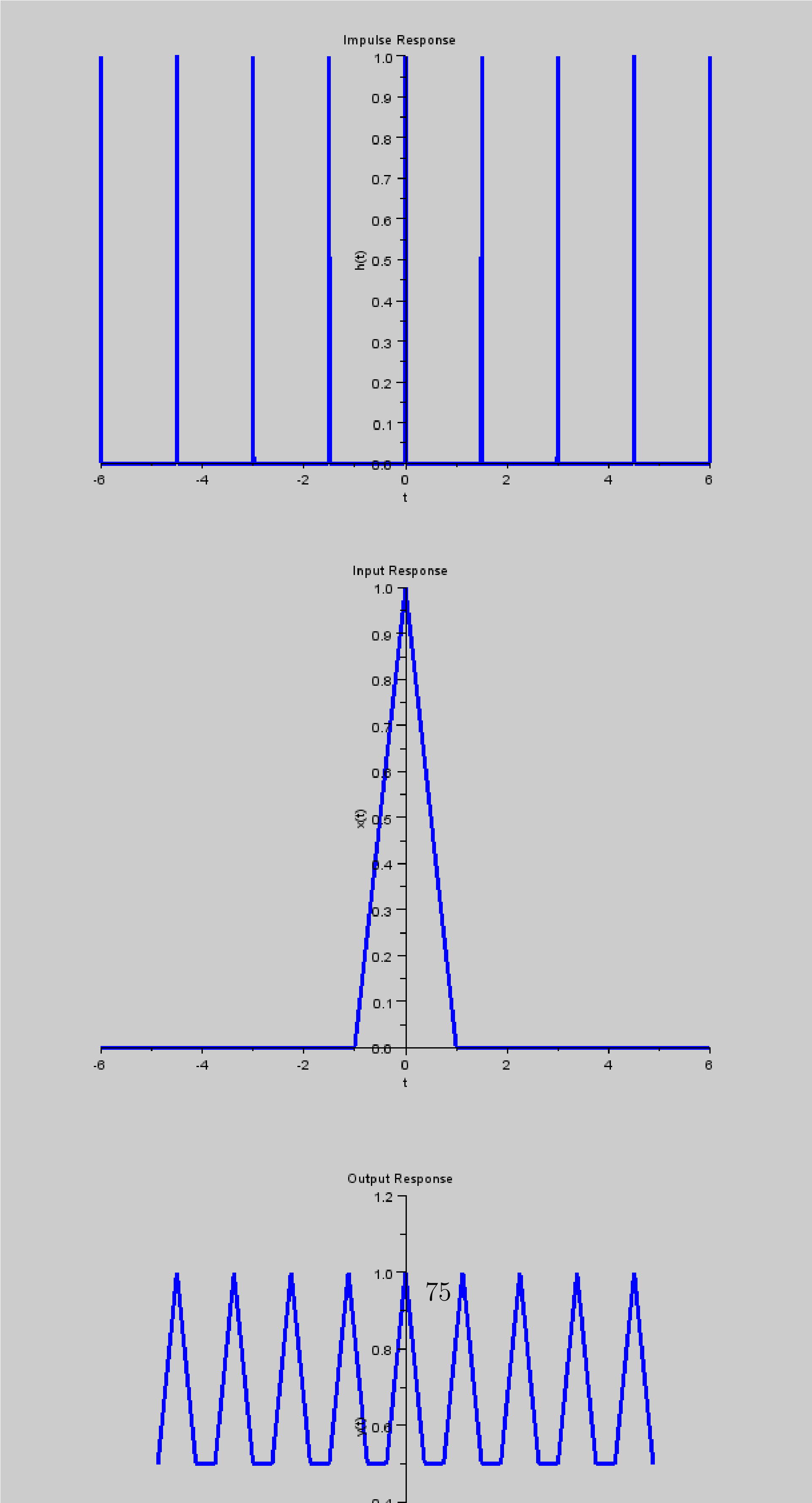
```
//Example 2.7:Convolution Integral
clear;
close;
clc;
t =-6:1/100:6;
for i=1:length(t)
    if modulo(t(i),2)==0 then
        h(i)=1;
    else
        h(i)=0;
    end
    if t(i)<-1 then
        x(i)=0;
    elseif t(i)<0
        x(i)=1+t(i);
    elseif t(i)<1
        x(i)=1-t(i);
    else
        x(i)=0;
```


Impulse Response
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
h(t)
-6
-4
-2
0
2
4
6
t
Input Response
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
x(t)
-6
-4
-2
0
2
4
6
t
Output Response
1.2
1.0
0.8
0.6
0.4
y(t)

```
    end
end
y = convol(x,h);
figure
a=gca();
a.x_location=”origin”;
a.y_location=”origin”;
plot2d(t,h)
xtitle(’Impulse Response’,’t’,’h(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
a.x_location=”origin”;
a.y_location=”origin”;
plot2d(t,x)
xtitle(’Input Response’,’t’,’x(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
t1=-12:1/100:12;
a.y_location=”origin”;
a.x_location=”origin”;
plot2d(t1,y)
xtitle(’Output Response’,’t’,’y(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

Scilab code Exa 2.7.c convolution of two rectangular pulse

```
//Example 2.7:Convolution Integral
clear;
```


Impulse Response
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
h(t)
-6
-4
-2
0
2
4
6
t
Input Response
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
x(t)
-6
-4
-2
0
2
4
6
t
Output Response
1.2
1.0
0.8
0.6
0.4
y(t)

```
close;
clc;
t =-6:1/100:6;
for i=1:length(t)
    if modulo(t(i),1.5)==0 then
        h(i)=1;
    else
        h(i)=0;
    end
    if t(i)<-1 then
        x(i)=0;
    elseif t(i)<0
        x(i)=1+t(i);
    elseif t(i)<1
        x(i)=1-t(i);
    else
        x(i)=0;
    end
end
y = convol(x,h);
figure
a=gca();
a.x_location=”origin”;
a.y_location=”origin”;
plot2d(t,h)
xtitle(’Impulse Response’,’t’,’h(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
a.x_location=”origin”;
a.y_location=”origin”;
plot2d(t,x)
xtitle(’Input Response’,’t’,’x(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
```

```
t1=-12:1/100:12;
a.y_location="origin";
a.x_location="origin";
b=find(t1==6.5);
c=find(t1==-6.5);
plot2d(t1(c:b),y(c:b))
xtitle('Output Response','t','y(t)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

Scilab code Exa 2.8 periodic convolution

```
//Example 2.8:Convolution Integral
clear;
close;
clc;
t =-4*10:1/100:4*10;
t2=-4:1/100:0;
for i=1:length(t2)
    if t2(i)<-2  then
        x(i)=1;
     else
        x(i)=0;
    end
end
fac=ceil(length(t)/length(t2));
s=[];
for i=1:fac;
   s=[s ;x];
end
y = convol(s,s)./2000;
figure
a=gca();
a.x_location="origin";
a.y_location="origin";
```

```
b=find(t==8);
c=find(t==-8);
plot2d(t(c:b),s(c:b))
xtitle(’Input Response’,’t’,’x(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
t1=-8*10:1/100:8*10;
a.y_location=”origin”;
a.x_location=”origin”;
b=find(t1==8);
c=find(t1==-8);
plot2d(t1(c:b),y(c:b))
xtitle(’Output Response’,’t’,’y(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

Scilab code Exa 2.9 output response

```
//Example 2.9:Convolution Integral
clear;
close;
clc;
t =-2:1/100:2;
for i=1:length(t)
    if t(i)<-1|t(i)>1 then
        x(i)=0;
    else
        x(i)=1;
    end
```

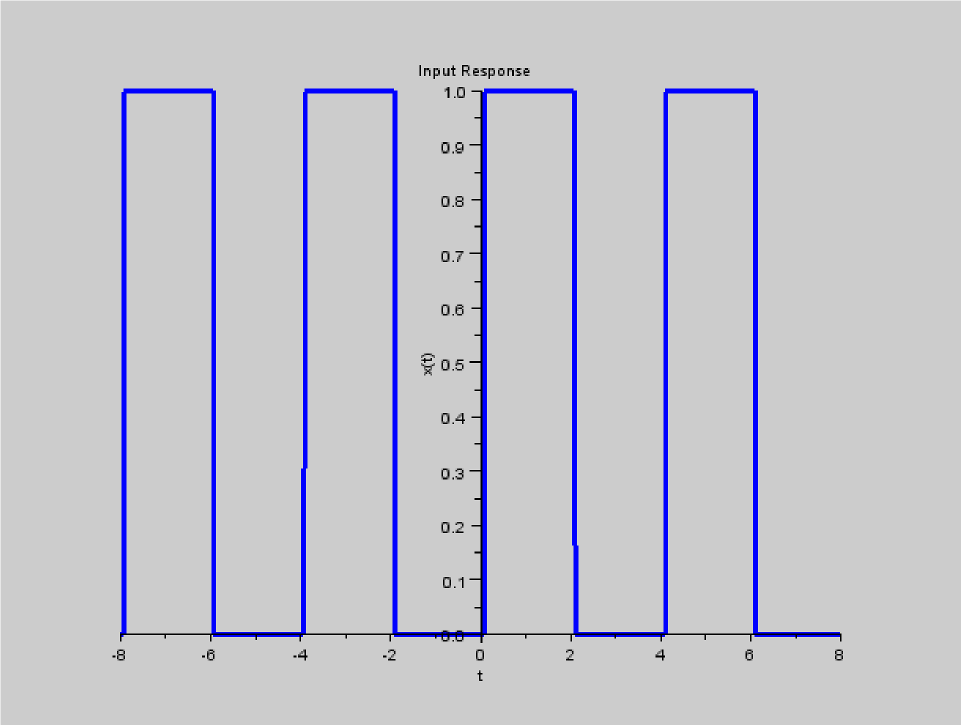


Figure 2.7: periodic convolution

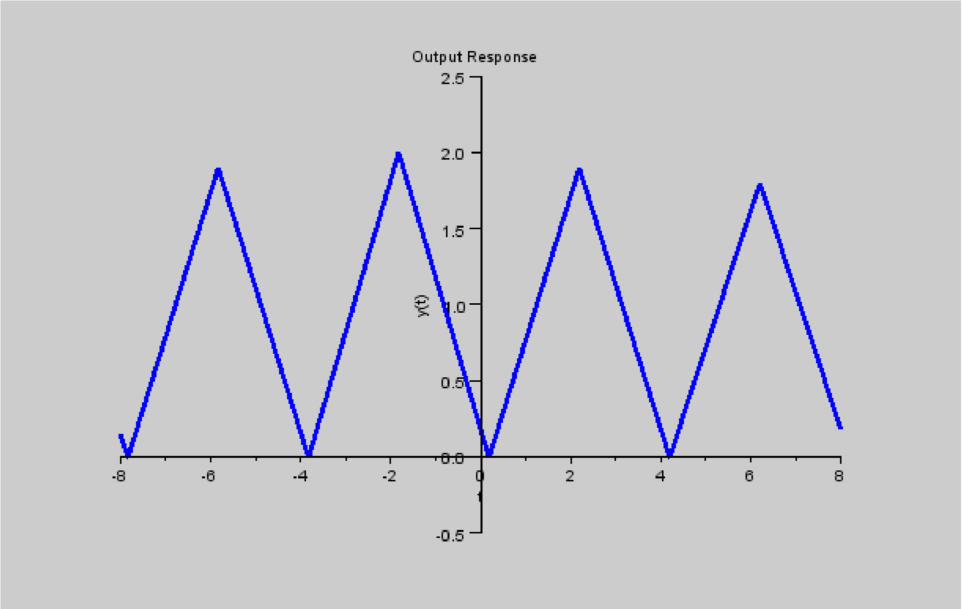


Figure 2.8: periodic convolution

```
end
ty=-2:1/100:4;
for i=1:length(ty)
    if ty(i)<-1|ty(i)>3 then
        y(i)=0;
    elseif ty(i)<1
        y(i)=1+ty(i);
    else
        y(i)=3-ty(i);
    end
end
figure
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d(t,x)
xtitle('Input Response','t','x(t)');
a.children.children.thickness = 3;
```

```
a.children.children.foreground= 2;
figure
a=gca();
a.y_location="origin";
a.x_location="origin";
plot2d(ty,y)
xtitle('Output Response','t','y(t)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
//since it is a time invariant system for x(t-2) o/p is y(t-2)
ty1=ty+2;
figure
a=gca();
a.y_location="origin";
a.x_location="origin";
plot2d(ty1,y)
xtitle('Output Response','t','y(t-2)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
//since the system is linear ,for x(t)/2 o/p is y(t)/2
figure
a=gca();
a.y_location="origin";
a.x_location="origin";
plot2d(ty,y./2)
xtitle('Output Response','t','.5*y(t)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

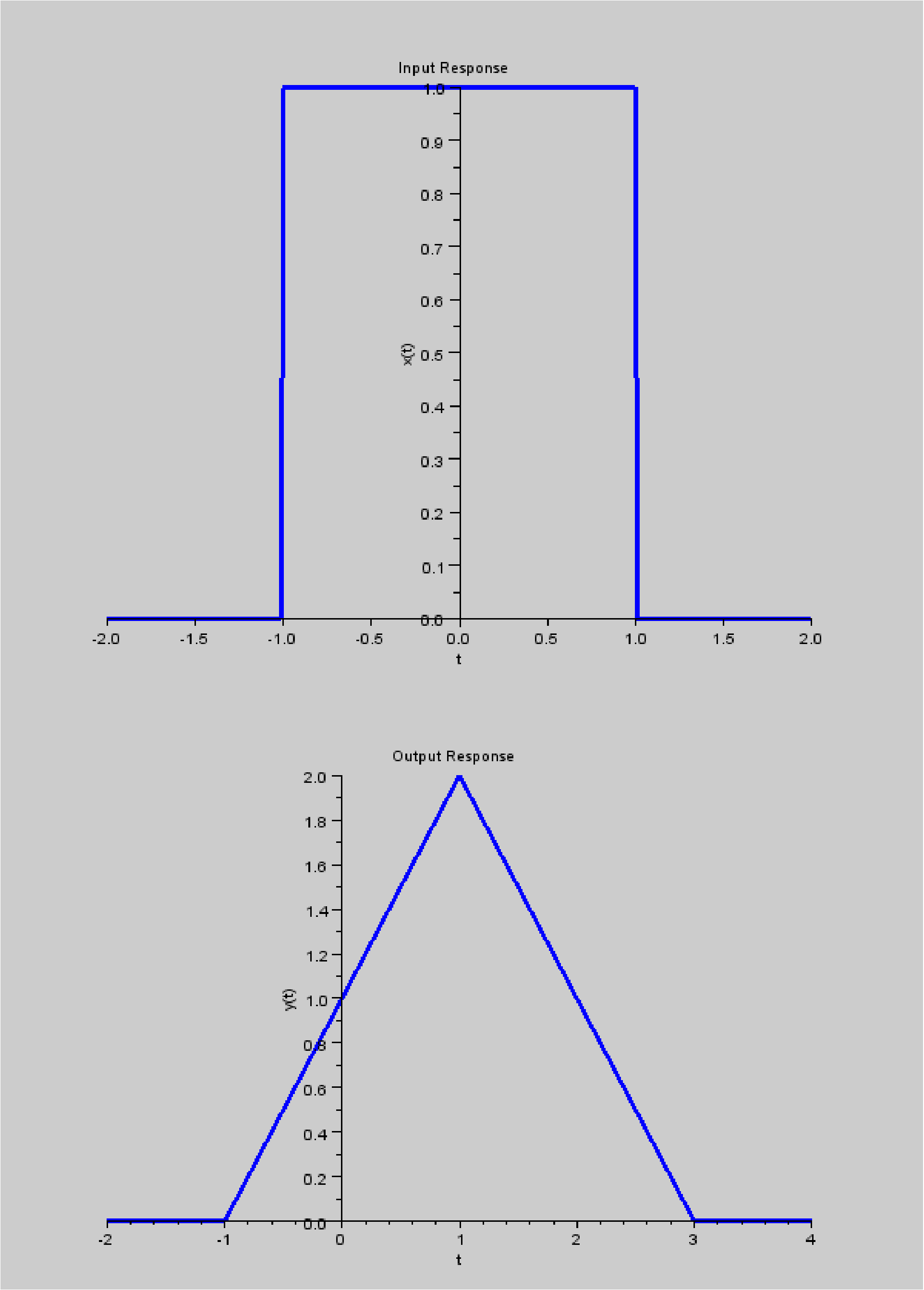


Figure 2.9: output response

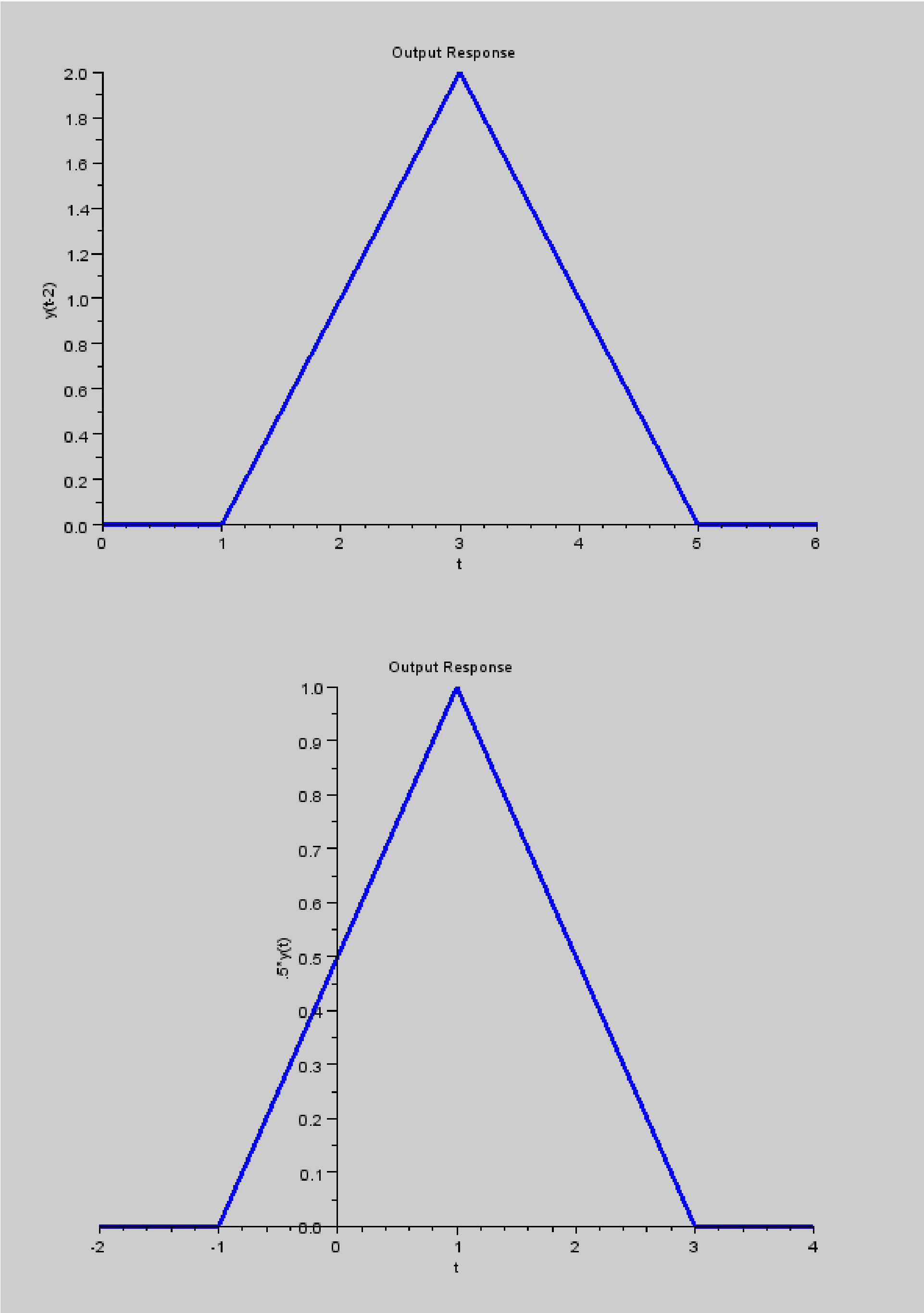
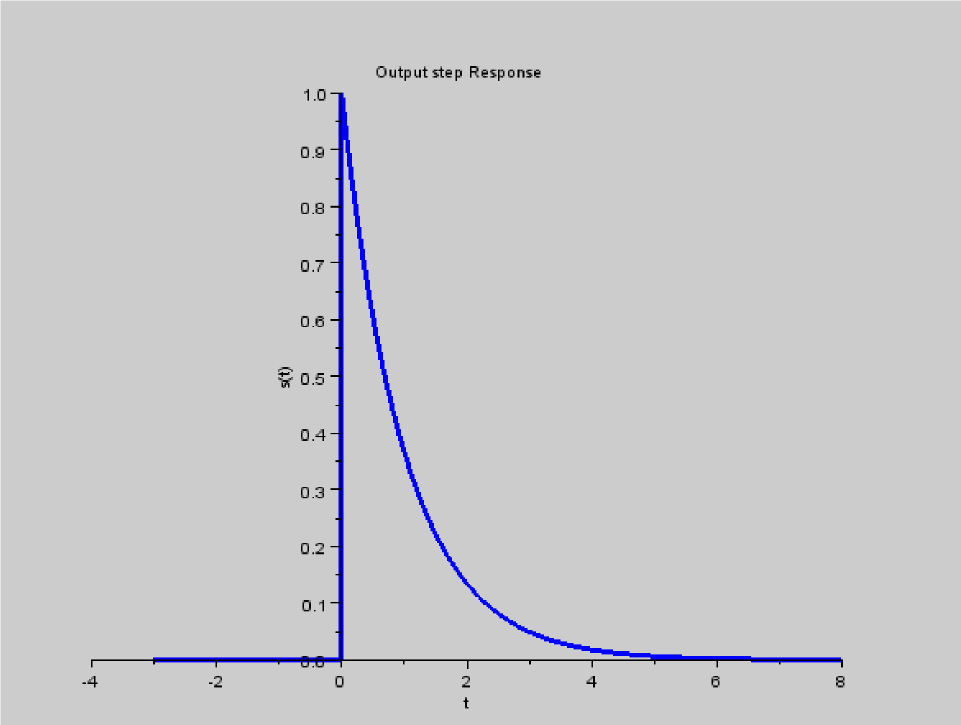
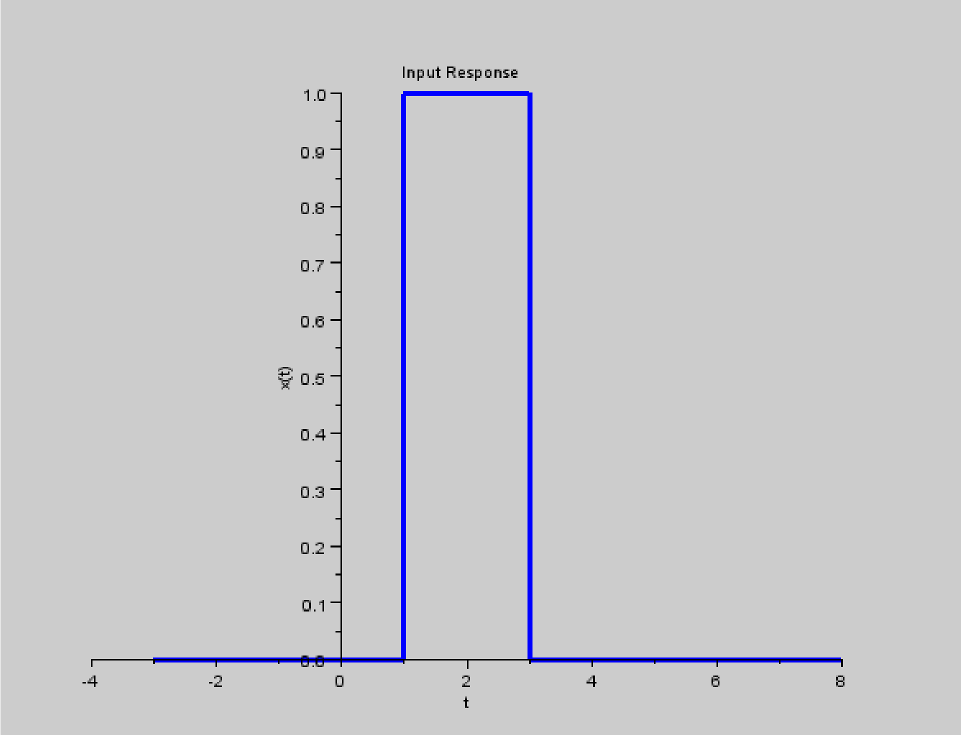
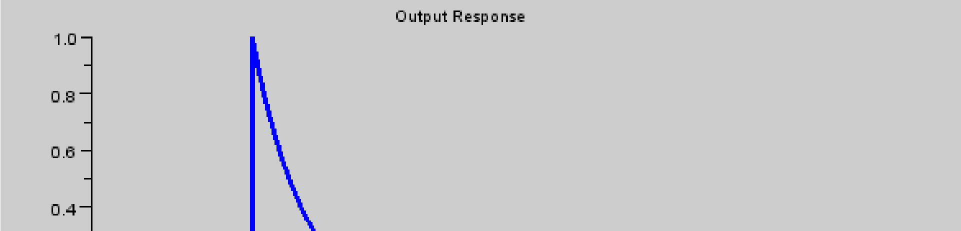


Figure 2.10: output response


Output step Response
1.0
0.9
0.8
0.7
0.6
s(t) 0.5
0.4
0.3
0.2
0.1
0.0
-4
-2
0
2
4
6
8
t


Input Response
1.0
0.9
0.8
0.7
0.6
x(t) 0.5
0.4
0.3
0.2
0.1
0.0
-4
-2
0
2
4
6
8
t


Output Response
1.0
0.8
0.6
0.4

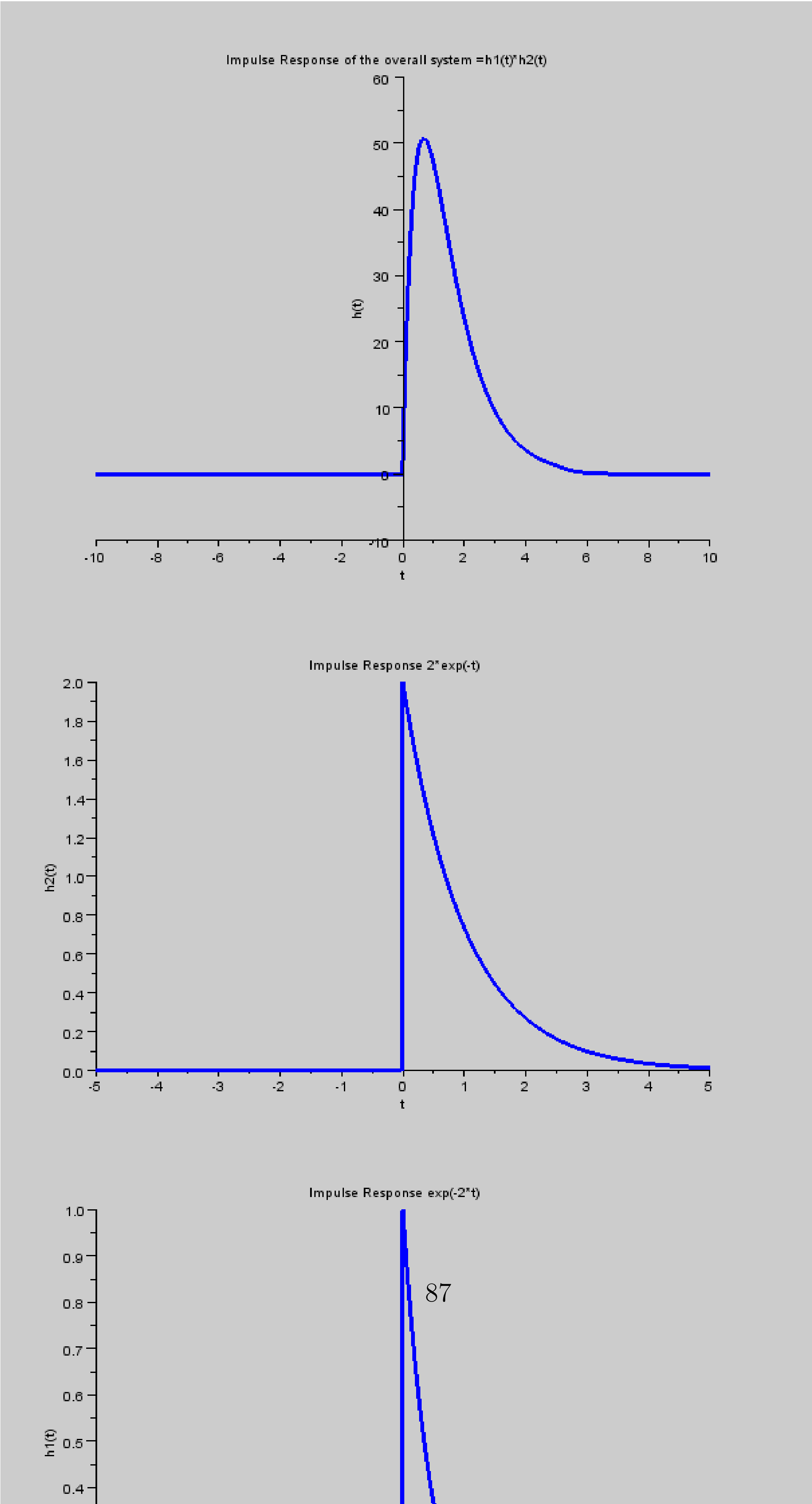
Scilab code Exa 2.10 output response

```
//Example 2.10:Convolution Integral
clear;
close;
clc;
t =-3:1/100:8;
s=[];
ss=[];
for i=1:length(t)
    if t(i)<1|t(i)>3 then
        x(i)=0;
    else
        x(i)=1;
    end
    if t(i)<0 then
        s(i)=0;
    else
       s(i)=exp(-t(i));
    end
end
figure
a=gca();
a.y_location=”origin”;
a.x_location=”origin”;
plot2d(t,s)
xtitle(’Output step Response’,’t’,’s(t)’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
a.x_location=”origin”;
a.y_location=”origin”;
plot2d(t,x)
```

```
xtitle('Input Response','t','x(t)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
t1=t+1;
t2=t+3;
s=s';
tt=min(min(t1,t2)):1/100:max(max(t1,t2));
ee=zeros(tt);
x=find(tt==-2);
y=find(tt==0);
z=find(tt==9);
for i=1:1:length(tt)
    if i<y then
        ee(i)=s(i);
    elseif i<z
        ee(i)=s(i)-s(i-y+1);
    else
        ee(i)=-s(i-y+1);
    end
end
figure
a=gca();
a.y_location="left";
a.x_location="origin";
plot2d(tt,ee)
xtitle('Output Response','t','s(t-1)-s(t-3)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

Scilab code Exa 2.14.a cascaded system

```
//Example 2.14:Convolution Integral
clear;
```



Impulse Response of the overall system =h1(t)*h2(t)
60
50
40
30
20
10
0
-10
h(t)
-10
-8
-6
-4
-2
0
2
4
6
8
10
t
Impulse Response 2*exp(-t)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
h2(t)
-5
-4
-3
-2
-1
0
1
2
3
4
5
t
Impulse Response exp(-2*t)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
h1(t)

```
close;
clc;
t =-5:1/100:5;
for i=1:length(t)
    if t(i)<0 then
        h1(i)=0;
        h2(i)=0;
    else
        h1(i)=exp(-2.*t(i));
        h2(i)=2*exp(-t(i));
    end
end
h=convol(h1,h2);
figure
a=gca();
a.x_location="origin";
plot2d(t,h1)
xtitle('Impulse Response exp(-2*t)','t','h1(t)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
plot2d(t,h2)
xtitle('Impulse Response 2*exp(-t)','t','h2(t)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
t1=-10:1/100:10;
a.y_location="origin";
plot2d(t1,h)
xtitle('Impulse Response of the overall system =h1(t)*h2(t)','t','h(t)');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

Scilab code Exa 2.14.b BIBO stability

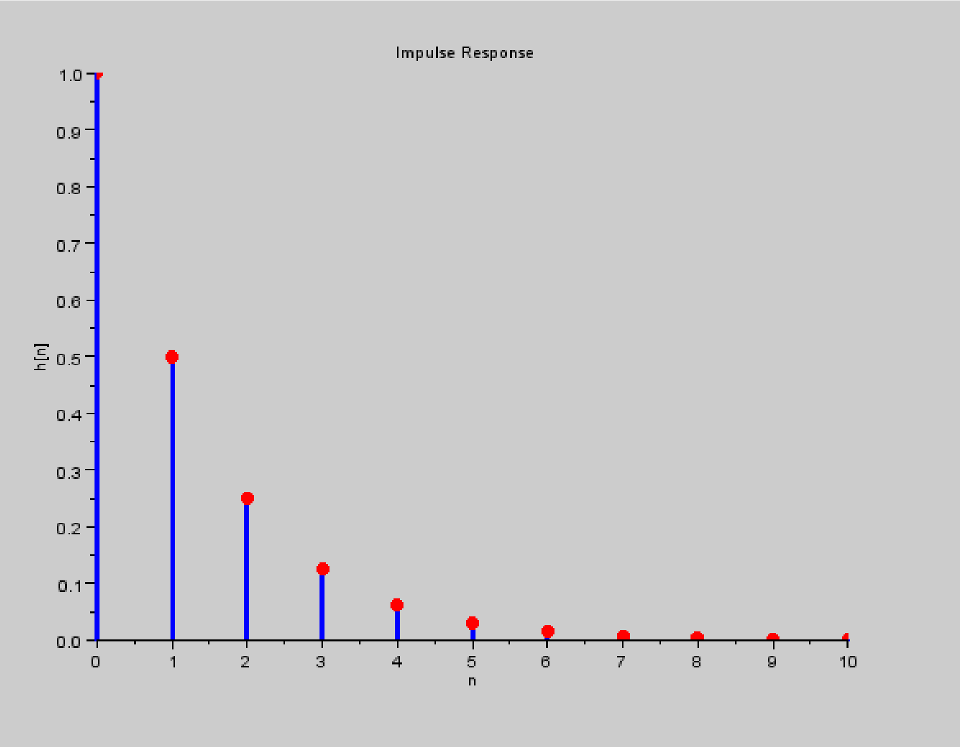
```
syms x y
y=integ(exp(-x)-exp(-2*x),x,0,1000000000)
disp("which is less than inf",y,"system  is bibo stable as y=");
```

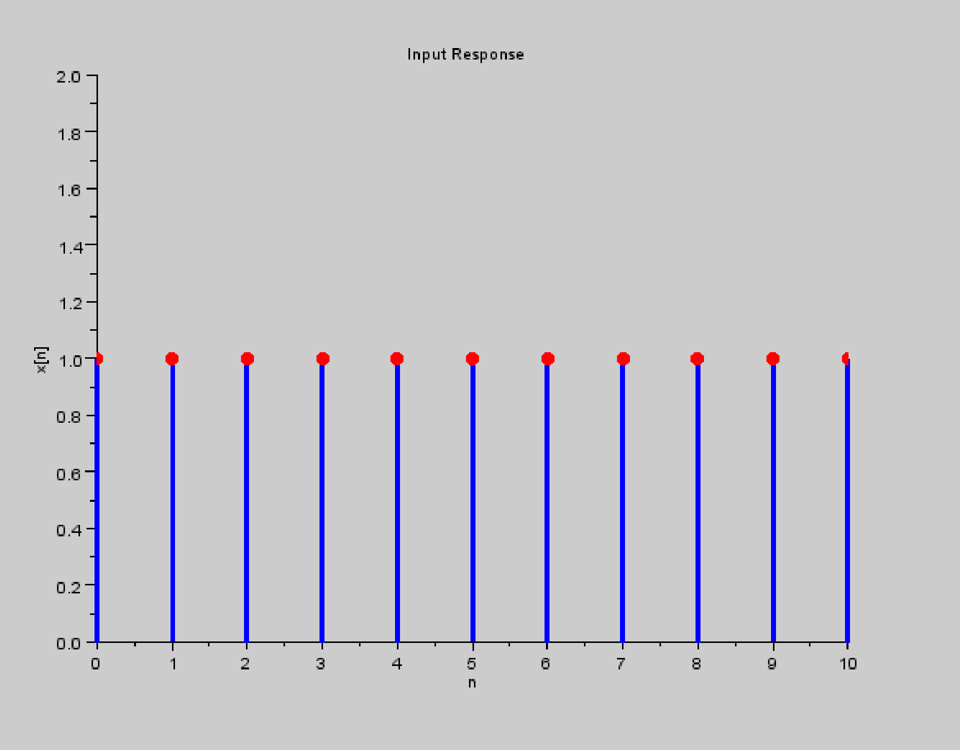
Scilab code Exa 2.15 eigenfunction of the system

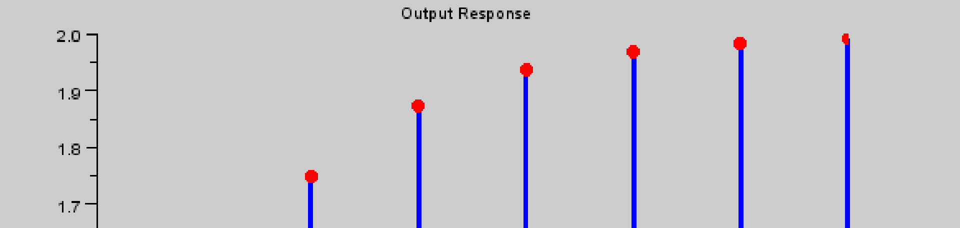
```
syms s T t
y=integ(exp(-(t-T))*exp(s*T),T,-%inf,t)
x=exp(s*t)
lamda=y/x//eigen value
disp(lamda,"b)  lamda=")
lamda_=laplace(exp(-t))
disp(lamda_,"c)  lamda=")
```

Scilab code Exa 2.16 eigen value of the system

```
syms s t T tou
y=T^-1*integ(exp(s*tou),tou,t-T/2,t+T/2)
x=exp(s*t)
lamda=y/x
disp(lamda,"lamda=")
```


Impulse Response
h[n]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
8
9
10
n


Input Response
x[n]
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
9
10
n

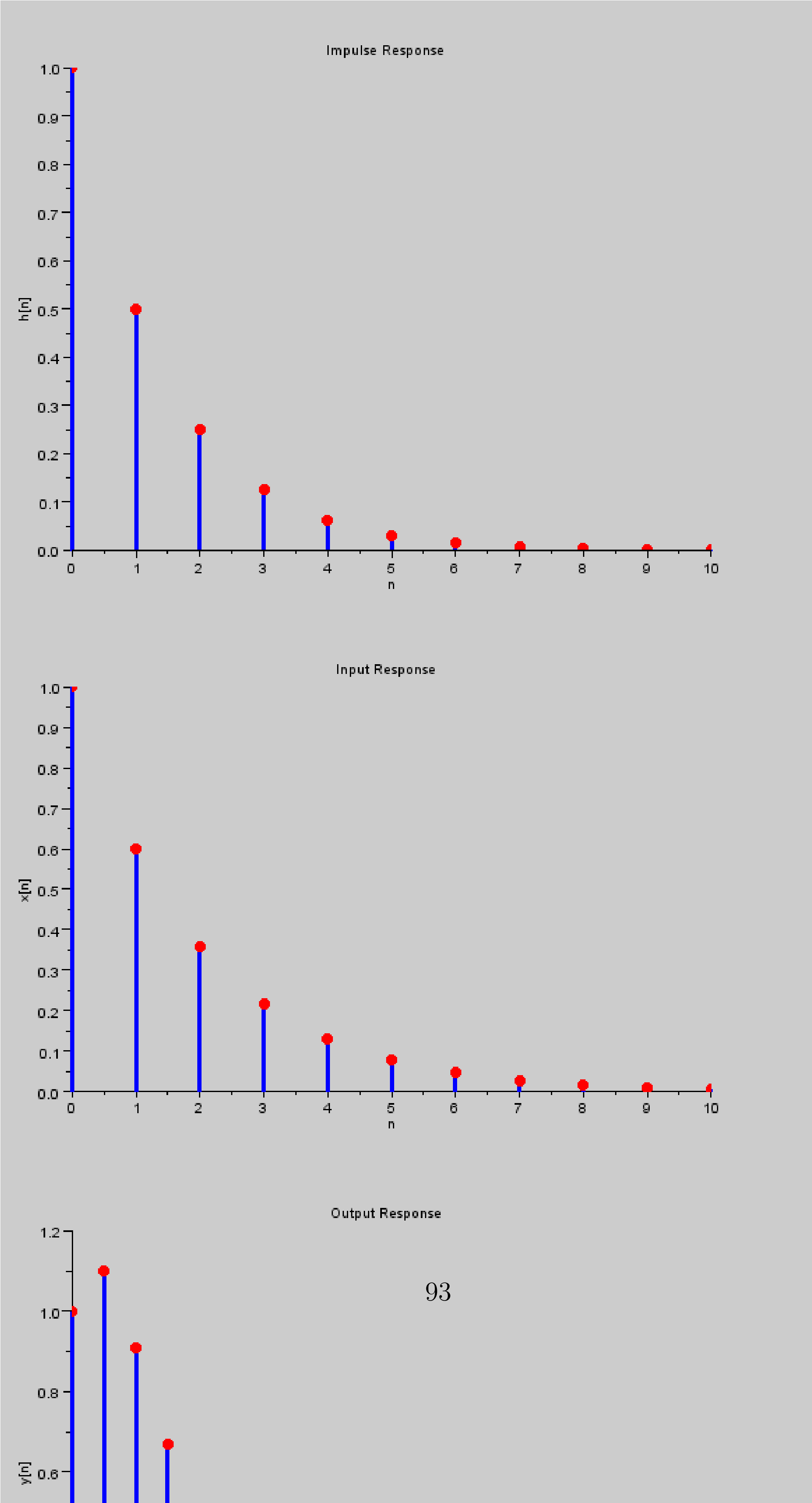

Output Response
2.0
1.9
1.8
1.7

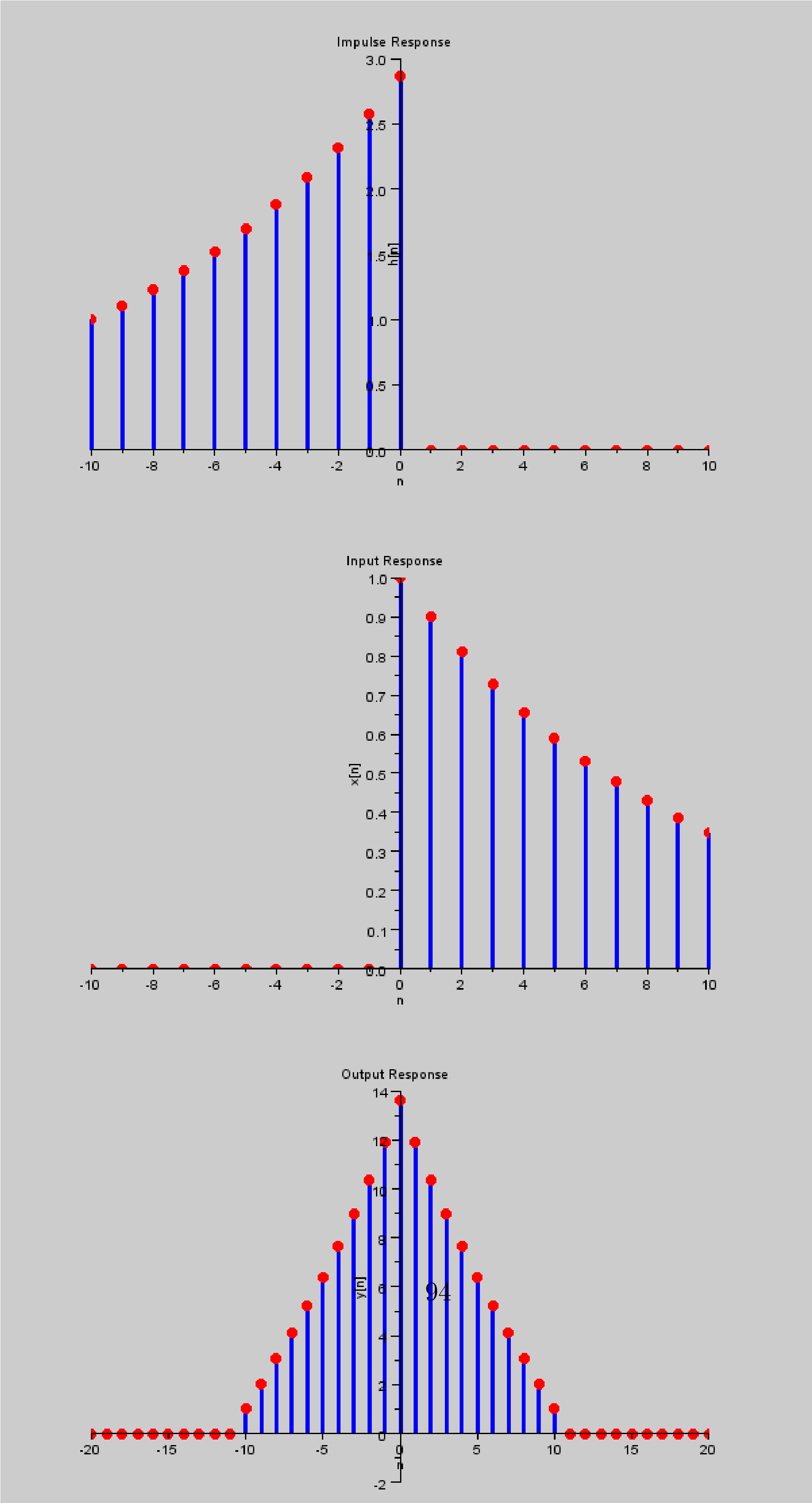
Scilab code Exa 2.28 output response of a discrete time system

```
clear;
clc;
n=0:10;
alpha=.5;
x=ones(n);
h=alpha^n;
y=convol(x,h);
figure
a=gca();
a.x_location="origin";
plot2d3(n,h)
plot(n,h,'r.')
xtitle('Impulse Response','n','h[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
plot2d3(n,x)
plot(n,x,'r.')
xtitle('Input Response','n','x[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
b=gca();
N=0:20;
a=find(N==7)
plot2d3(N(1:a),y(1:a))
plot(N(1:a),y(1:a),'r.')
xtitle('Output Response','n','y[n]');
b.children.children.thickness = 3;
b.children.children.foreground= 2;
```

Scilab code Exa 2.29.a convolution of discrete signals

```
clear;
clc;
n=0:10;
alpha=.5;
betaa=.6;
x=betaa^n;
h=alpha^n;
y=convol(x,h);
figure
a=gca();
a.x_location="origin";
plot2d3(n,h)
plot(n,h,'r.')
xtitle('Impulse Response','n','h[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
plot2d3(n,x)
plot(n,x,'r.')
xtitle('Input Response','n','x[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
N=0:20;
plot2d3(N,y)
plot(N,y,'r.')
xtitle('Output Response','n','y[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```


Impulse Response
h[n]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
8
9
10
n
Input Response
x[n]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
8
9
10
n
Output Response
y[n]
1.2
1.0
0.8
0.6


Impulse Response
3.0
2.5
2.0
1.5
1.0
0.5
0.0
h[n]
-10
-8
-6
-4
-2
0
2
4
6
8
10
n
Input Response
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
x[n]
-10
-8
-6
-4
-2
0
2
4
6
8
10
n
Output Response
14
12
10
8
6
4
2
0
-2
y[n]
-20
-15
-10
-5
0
5
10
15
20
n

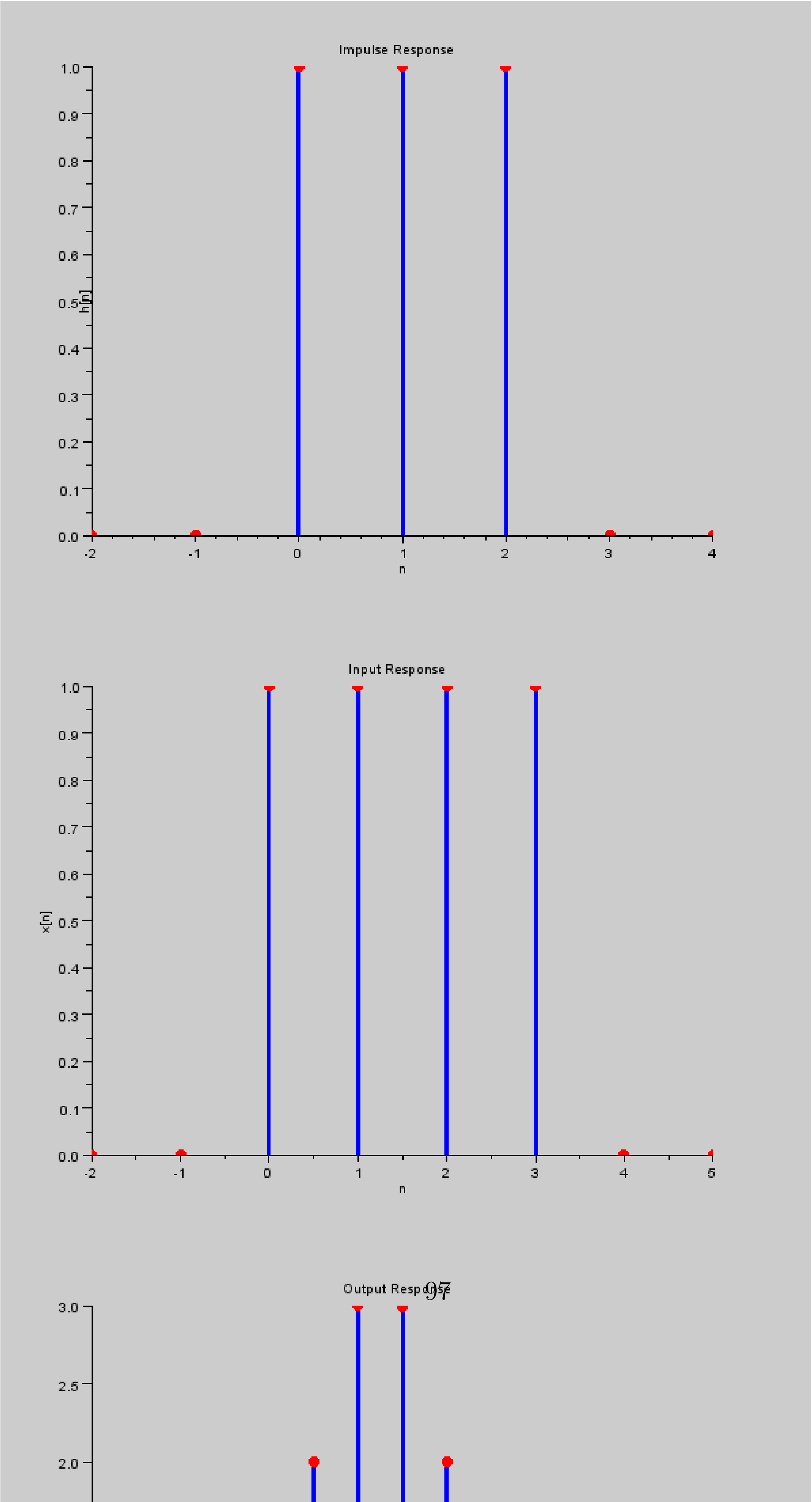
Scilab code Exa 2.29.b convolution of discrete signals

```
clear;
clc;
n=0:10;
alpha=.9;
x=[ zeros(1,length(n)-1) alpha^n];
h=[alpha^-n];
h=[h zeros(1,length(n)-1)];
y=convol(x,h);
figure
a=gca();
a.x_location=”origin”;
n=-10:10;
plot2d3(n,h)
plot(n,h,’r.’)
xtitle(’Impulse Response’,’n’,’h[n]’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
a.y_location=”origin”;
figure
a=gca();
plot2d3(n,x)
plot(n,x,’r.’)
a.y_location=”origin”;
a.x_location=”origin”;
xtitle(’Input Response’,’n’,’x[n]’);
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
N=-20:20;
plot2d3(N,y)
plot(N,y,’r.’)
a.x_location=”origin”;
a.y_location=”origin”;
xtitle(’Output Response’,’n’,’y[n]’);
a.children.children.thickness = 3;
```

```
a.children.children.foreground= 2;
```

Scilab code Exa 2.30 convolution of discrete signals

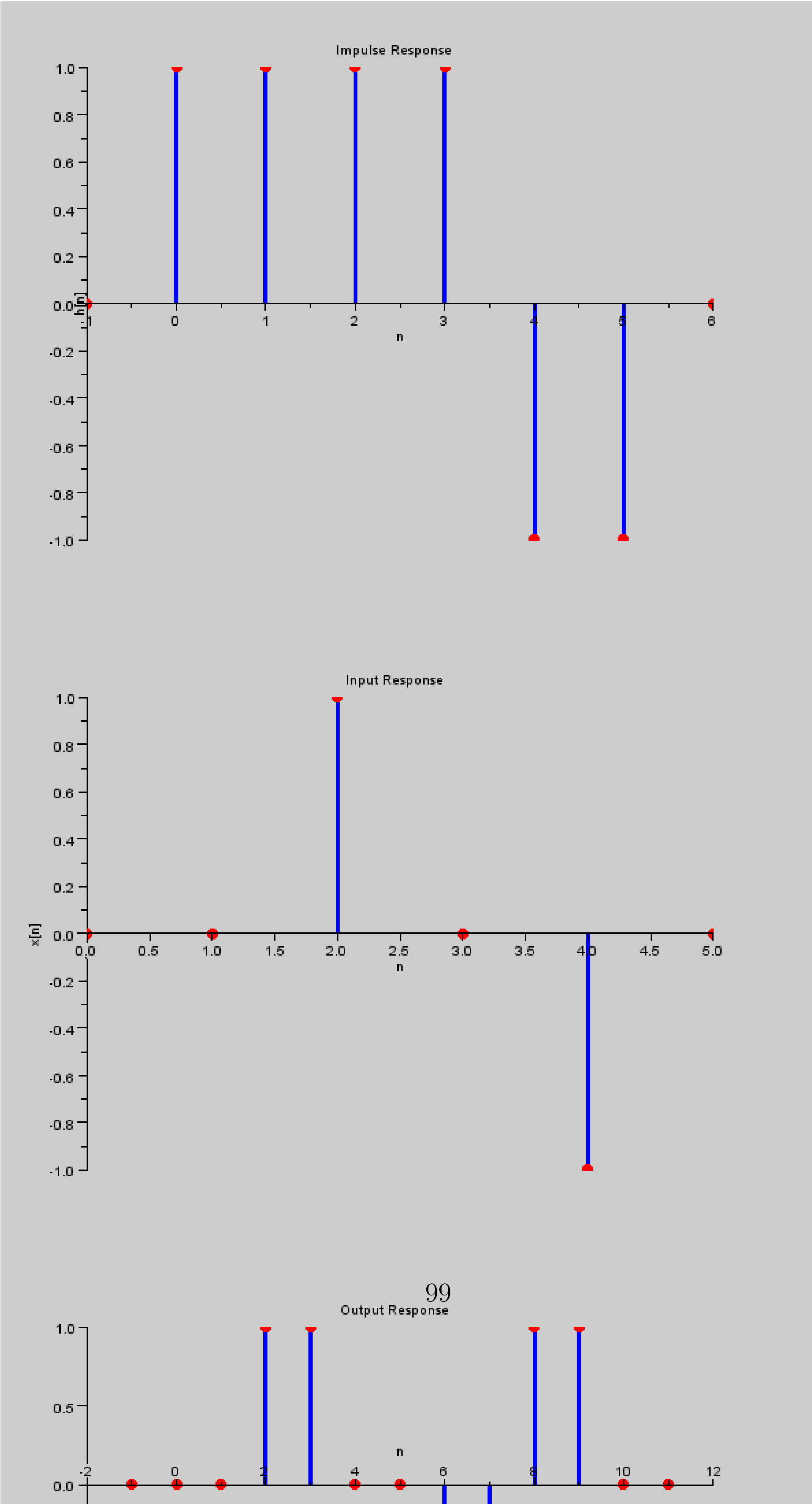
```
clear;
clc;
x=[0 0 1 1 1 1 0 0];
h=[0 0 1 1 1 0 0];
nx=-2:length(x)-3;
nh=-2:length(h)-3;
y=convol(x,h);
ny=min(nx)+min(nh):max(nx)+max(nh);
figure
a=gca();
a.x_location="origin";
plot2d3(nh,h)
plot(nh,h,'r.')
xtitle('Impulse Response','n','h[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
a.y_location="left";
figure
a=gca();
plot2d3(nx,x)
plot(nx,x,'r.')
a.y_location="left";
a.x_location="origin";
xtitle('Input Response','n','x[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
plot2d3(ny,y)
```


Impulse Response
h[n]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-2
-1
0
1
2
3
4
n
Input Response
x[n]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-2
-1
0
1
2
3
4
5
n
Output Response
3.0
2.5
2.0

```
plot(ny,y,'r.')
a.x_location="origin";
a.y_location="left";
xtitle('Output Response','n','y[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

Scilab code Exa 2.34 output response without using convolution

```
clear;
clc;
h=[0 1 1 1 1 -1 -1 0];
x=[0 0 1 0 -1 0];
nx=0:length(x)-1;
nh=-1:length(h)-2;
//y=convol(x,h);
ny=min(nx)+min(nh):max(nx)+max(nh);
//or x[n]=delta[n-2]-delta[n-4] therefore y[n]=h[n-2]-h[n-4]
n1=nh+2;
n2=nh+4;
ny=min(nx)+min(nh):max(nx)+max(nh);
j=1;
k=1;
h2=zeros(ny);
h4=h2;
a=find(ny==n1(1))
for j=1:length(nh)
    h2(a+j-1)=h(j)
        end
a=find(ny==n2(1))
for j=1:length(nh)
    h4(a+j-1)=h(j)
```


Impulse Response
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
h[n]
-1
0
1
2
3
4
5
6
n
Input Response
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
x[n]
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
n
Output Response
1.0
0.5
0.0
n
-2
0
2
4
6
8
10
12

```
        end
y=h2-h4;
figure
a=gca();
a.x_location="origin";
plot2d3(nh,h)
plot(nh,h,'r.')
xtitle('Impulse Response','n','h[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
a.y_location="left";
figure
a=gca();
plot2d3(nx,x)
plot(nx,x,'r.')
a.y_location="left";
a.x_location="origin";
xtitle('Input Response','n','x[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
figure
a=gca();
plot2d3(ny,y)
plot(ny,y,'r.')
a.x_location="origin";
a.y_location="left";
xtitle('Output Response','n','y[n]');
a.children.children.thickness = 3;
a.children.children.foreground= 2;
```

Scilab code Exa 2.36 causality

```
clear;
clc;
n=-5:5;
```

```
for i=1:length(n)
    if(n(i)>=-1)
        h(i)=2^-(n(i)+1);
     else
         h(i)=0;
     end
 end
causal=%t;
for i=1:length(n)
    if n(i)<0 & h(i)~=0 then
    causal=%f;
    end
end
disp(causal,"the statement that the system is causal is");
```

Scilab code Exa 2.38 BIBO stability and causality

```
clear;
clc;
n=-5:5;
alpha=.6;
for i=1:length(n)
    if(n(i)>=0)
        h(i)=alpha^n(i);
     else
         h(i)=0;
     end
 end
causal=%t;
for i=1:length(n)
    if n(i)<0 & h(i)~=0 then
    causal=%f;
    end
end
```

```
disp(causal,"the statement that the system is causal is");
n=0:100000;
for i=1:length(n)
    if(n(i)>=0)
        h(i)=alpha^n(i);
     else
         h(i)=0;
     end
 end
 bibo=sum(h);
 if (bibo<%inf) then
     disp("system is bibo stable");
 else
    disp("system is not stable");
 end
```

Chapter 3

Laplace transform and continuous time LTI systems

Scilab code Exa 3.1.a laplace transform

```
syms t a y
y= -laplace(-exp(-a*t),t)
```

Scilab code Exa 3.1.b laplace transform

```
syms t a y
y= -laplace(exp(a*t),t)
```

Scilab code Exa 3.3 laplace transform

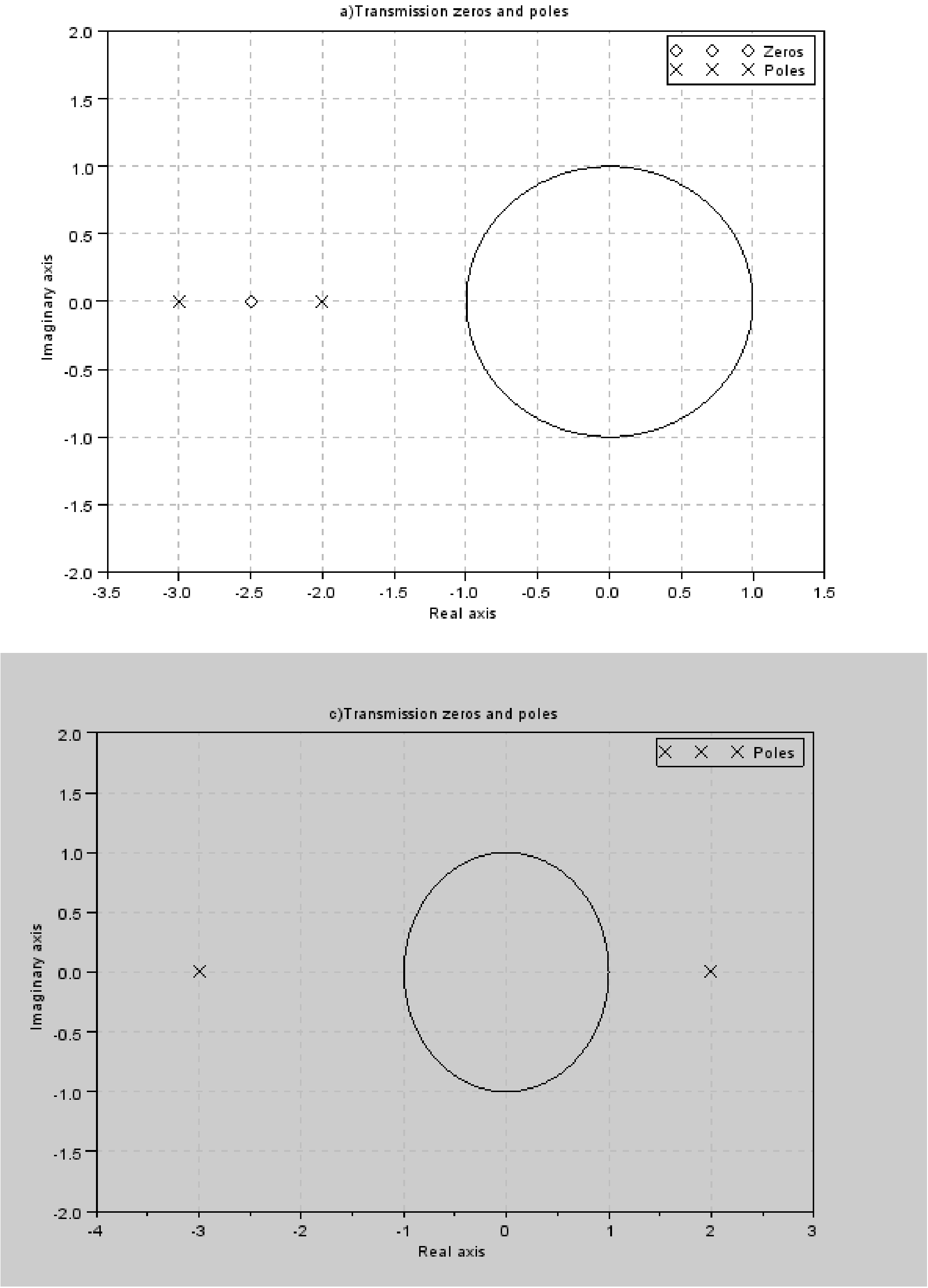
```
syms t s a T
y= integ(exp(-a*t-s*t),t,0,T)
```

Scilab code Exa 3.5 pole zero plot

```
clc
syms t s
s1=%s;
x=laplace(exp(-2*t)+exp(-3*t),t,s)
y=laplace(exp(-3*t)-%e^(2*t))
z=laplace(%e^(2*t)-%e^(-3*t))
disp(z,y,x,"laplace transform of a b c is")
x=1/(s1+2)+1/(s1+3);
plzr(x)
y=1/(s1+3)-1/(s1-2);
figure
plzr(y)
z=1/(s1-2)-1/(s1+3);
figure
plzr(z)
disp("there is no region of convergence for c hence no transform exists")
```

Scilab code Exa 3.6 ROC and pole zero plot

```
syms t s a
y=laplace(%e^(-a*t)-%e^(a*t))
disp(y,"X(s)=")
s1=%s;
//a>0
a=2;
```


a)Transmission zeros and poles

c)Transmission zeros and poles

Figure 3.1: pole zero plot

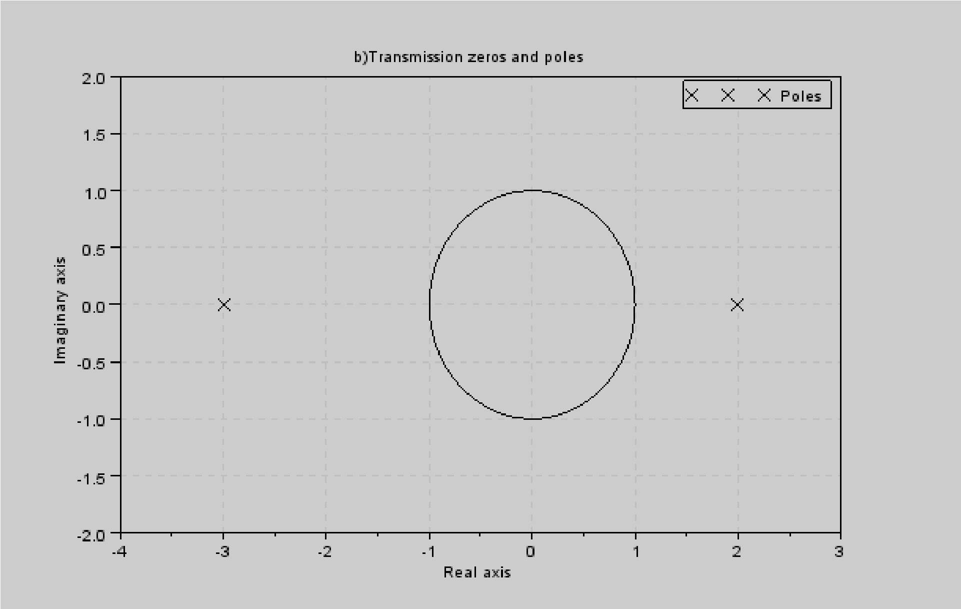


Figure 3.2: pole zero plot

```
t=-5:0.1:5;
x=%e^(-a*abs(t));
subplot(2,1,1)
plot(t,x)
subplot(2,1,2)
x=1/(s1+a)-1/(s1-a);
plzr(x)
//a<0
a=-0.5;
t=-5:0.1:5;
x=%e^(-a*abs(t));
figure
subplot(2,1,1)
plot(t,x)
subplot(2,1,2)
x=1/(s1+a)-1/(s1-a);
plzr(x)
disp(”there is no region of convergence when a<0
```

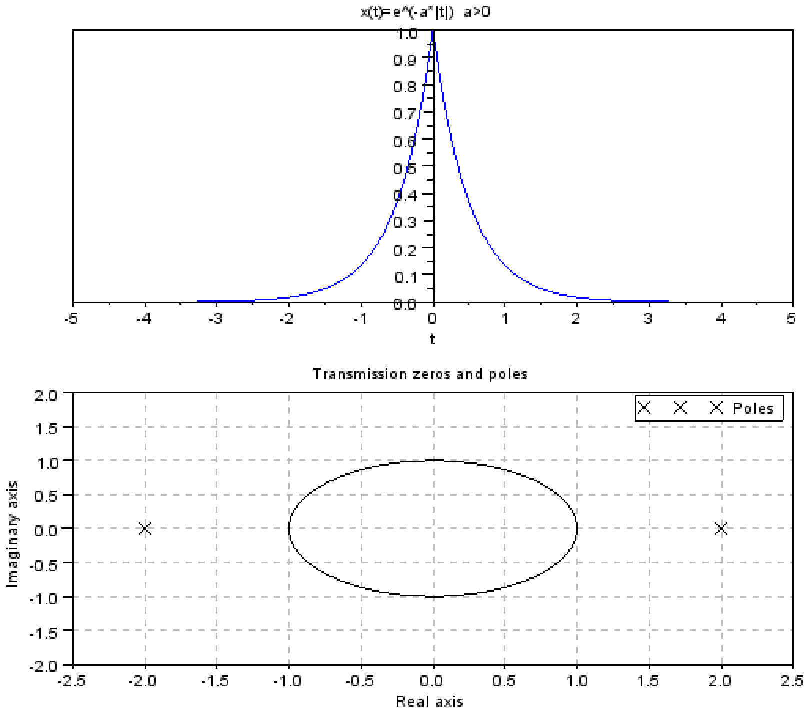


Figure 3.3: ROC and pole zero plot

```
   hence no transform exists for a<0")
```

Scilab code Exa 3.13 derivative and shifting property

```
syms t s a w
// given u(t)<-->1/s
//delta(t)=diff(u(t))
u=laplace(1);
d=s*u
```

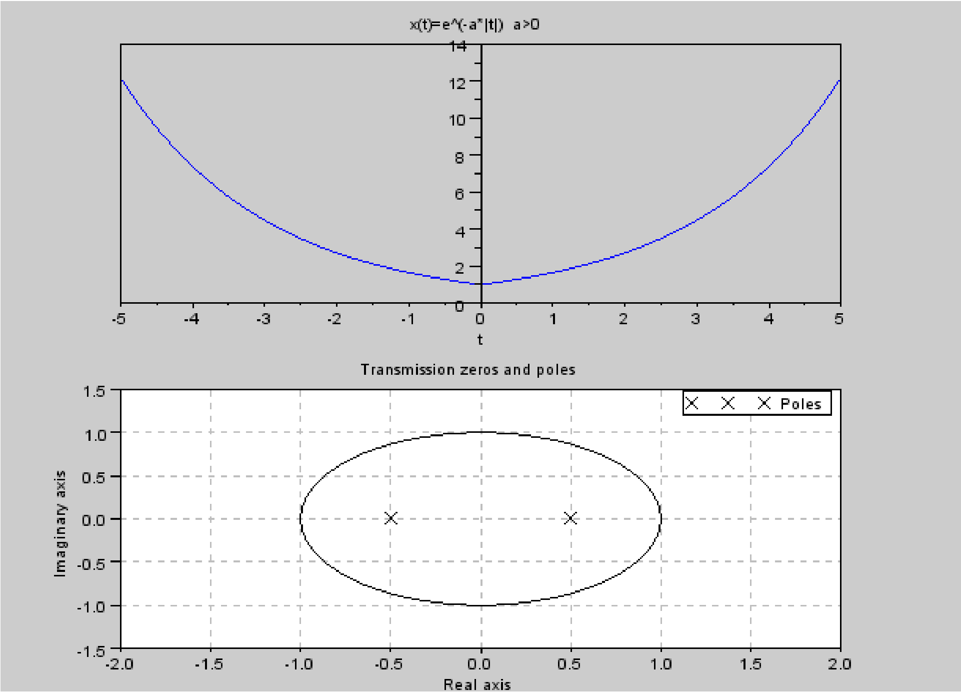


Figure 3.4: ROC and pole zero plot

```
disp(d,"delta(t) <-->")
d1=s*d
disp(d1,"diff(delta(t)) <-->")
tu=-diff(u,s)
disp(tu,"t*u(t) <-->")
eu=laplace(%e^-(a*t))
disp(eu,"e^-a*t*u(t) <-->")
teu=-diff(eu,s)
disp(teu,"t*e^-a*t*u(t) <-->")
cu=laplace(cos(w*t))
disp(cu,"cos(w0*t)*u(t) <-->")
ecu=laplace(%e^(-a*t)*cos(w*t))
disp(tu,"e^(-a*t)*cos(w0*t)*u(t) <-->")
```

Scilab code Exa 3.16 inverse laplace transform

```
clear;
syms t s
x=1/(s+1)
f1=ilaplace(x)
disp(f1*'u(t)',"a) x(t)=")
y=-1/(s+1)
f2=ilaplace(y)
disp(f2*'u(-t)',"b) x(t)=")
z=s/(s^2+4)
f3=ilaplace(z)
disp(f3*'u(t)',"c) x(t)=")
zz=(s+1)/((s+1)^2+4)
f4=ilaplace(zz)
disp(f4*'u(t)',"d) x(t)=")
```

Scilab code Exa 3.17 inverse laplace transform

```
clear;
syms t s
x=(2*s+4)/(s^2+4*s+3)
f1=ilaplace(x)
disp(f1*'u(t)',"a) x(t)=")
y=-(x)
f2=ilaplace(y)
disp(f2*'u(-t)',"b) x(t)=")
q= %s
z=pfss((2*q+4)/(q^2+4*q+3))
f3=ilaplace(-z(1))
f4=ilaplace(z(2))
ff=f3+f4
disp(f3*'u(-t)'+f4*'u(t)',"c) x(t)=")
```

Scilab code Exa 3.18 inverse laplace transform

```
clear;
clc;
syms t s
x=(5*s+13)/(s*(s^2+4*s+13));
X=ilaplace(x);
disp(X*'u(t)',"x(t)=")
```

Scilab code Exa 3.19 inverse laplace transform

```
syms t s
x=ilaplace((s^2+2*s+5)/((s+3)*(s+5)^2))
disp(x*'u(t)',"x(t)=")
```

Scilab code Exa 3.20 inverse laplace transform by partial fractions

```
syms t s
s= %s
a1=pfss((2*s+1)/(s+2))
f1=ilaplace(a1(1))
fx=f1
disp(fx*'u(t)'+'2*delta(t)' ,"a) x(t)=")
a2=pfss((s^2+6*s+7)/(s^2+3*s+2))
f1=ilaplace(a2(1))
f2=ilaplace(a2(2))
fy=f1+f2
disp(fy*'u(t)'+'delta(t)' ,"b) x(t)=")
a3=pfss((s^3+2*s^2+6)/(s^2+3*s))
f1=ilaplace(a3(1))
f2=ilaplace(a3(2))
fz=f1+f2
disp(fz*'u(t)'+'-delta(t)+delta1(t)',"c) x(t)=")
```

Scilab code Exa 3.21 inverse laplace transform of time shifted signal

```
syms t s
s=%s
a=ilaplace(2/(s^2+4*s+3))
b=ilaplace(2*s/(s^2+4*s+3))
c=ilaplace(4/(s^2+4*s+3))
disp(a*'u(t)'+b*'u(t-2)'+c*'u(t-4)',"x(t)=")
```

Scilab code Exa 3.22 differentiation in s domain

```
syms t s a
x=ilaplace((s+a)^-2)
disp(x*'u(t)',"x(t)=")
```

Scilab code Exa 3.24 output response

```
syms t s a
H=laplace(%e^(-a*t))
X=laplace(-%e^(a*t))
Y=X*H
y=ilaplace(Y)
disp(y,"y(t)=")
```

Scilab code Exa 3.25 impulse response

```
syms t s
X=laplace(1)
Y=laplace(2*%e^(-3*t))
H=Y/X;
disp(H,"H(s)=")
s=%s;
h=2*s/(s+3);
hp=pfss(h);
h1=ilaplace(hp(1));
disp(h1*'u(t)'+'2*delta(t)',"h(t)=")
//part b)
X=laplace(%e^-t)
Y=X*H;
y=ilaplace(Y);
disp(y*'u(t)',"y(t)=")
```

Scilab code Exa 3.27 cascaded system transfer function

```
clear;
clc;
syms t s
H1=laplace(%e^(-2*t))
H2=laplace(2*%e^(-t))
H=H1*H2;
h=ilaplace(H)
disp(h*'u(t)',"h(t)=")
```

Scilab code Exa 3.28 first order differntial equation

```
clear;
clc;
syms t s a
H=1/(s+a);
h=ilaplace(H)
disp(h*'u(t)',"h(t)=")
```

Scilab code Exa 3.29 impulse response

```
clear;
clc;
syms t s1 a
s=%s
H=(s+1)/(s+2);
hp=pfss(H)
h1=ilaplace(hp(1))
disp(h1*'u(t)'+'delta(t)',"h(t)=")
```

Scilab code Exa 3.30 causality and stability

```
clear;
clc;
syms t s1 a
s=%s;
H=1/((s+2)*(s-1));
hp=pfss(H)
h1=ilaplace(hp(1))
h2=ilaplace(hp(2))
disp((h1+h2)*'u(t)',"when the system is causal h(t)=")
disp(h1*'u(-t)'+h2*'u(-t)',"when the system is stable h(t)=")
disp((h1+h2)*'u(-t)',"when the system is neither stable nor causal h(t)=")
```

Scilab code Exa 3.34 bilateral laplace transform

```
syms t s
X1=laplace(%e^(-2*t))
X2=laplace(exp(2*t))
X=X1-X2
disp(X,"bilateral transform of x(t)=")
```

Scilab code Exa 3.36 unilateral laplace transform

```
syms s t
u=integ(exp(-s*t),t,0,%inf)
delta=s*u-1;
disp(u,"unilateral transform of u(t)")
disp(delta,"unilateral transform of delta(t)")
```

Scilab code Exa 3.37 unilateral laplace transform method

```
clear;
clc;
syms n s a b y0 K t;
X=laplace(K*%e^(-b*t));
Y=y0/(s+a)+X/(s+a);
y=ilaplace(Y)
disp(y,"y(t)=")
```

Scilab code Exa 3.38 second order ODE

```
clear;
clc;
syms n s t;
//y(0)=2 y'(0)=1
X=laplace(%e^(-t));
Y=(X+2*s+11)/(s^2+5*s+6);
y=ilaplace(Y)
disp(y,"y(t)=")
```

Scilab code Exa 3.39 RC circuit

```
clear;
clc;
syms n s t R C V v0;
I=(V-v0)/(s*(R+1/(C*s)));
i=ilaplace(I);
disp(i*'u(t)',"i(t)=")
Vc=(V/(R*C*s)+v0)/(s+1/(R*C));
vc=ilaplace(Vc)
disp(vc,"vc(t)=")
```

Scilab code Exa 3.40 RC circuit response

```
clear;
clc;
syms n s t R C V v0;
I=(V-v0)/(s*(R+1/(C*s)));
i=ilaplace(I);
disp(i*’u(t)’,”i(t)=”)
Vc=(V/(R*C*s)+v0)/(s+1/(R*C));
vc=ilaplace(Vc)
disp(vc*’u(t)’,”vc(t)=”)
```

Scilab code Exa 3.41 RLC circuit

```
clear;
clc;
syms s t n;
I=1/(s/2+2+20/s)
i=ilaplace(I)
disp(i,”i(t)=”)
```

Scilab code Exa 3.42 circuit analysis

```
clear;
clc;
s=%s;
I1=(s+1)/(s+1/4);
I1p=pfss(I1)
i1=ilaplace(I1p(1))
```

```
disp(i1*'u(t)'+'delta(t)',”i1(t)=”)
I2=(s-1/2)/(s+1/4);
I2p=pfss(I2)
i2=ilaplace(I2p(1))
disp(i2*'u(t)'+'delta(t)',”i2(t)=”)
```

Chapter 4

The z transform and discrete time LTI systems

Scilab code Exa 4.1.a z transform

```
clear;
clc;
syms n z a;
x=a^n;
X=symsum(-x*z^-n,n,-%inf,-1)
disp(X,”ans=”)
```

Scilab code Exa 4.1.b z transform

```
clear;
clc;
syms n z a;
x=a^-n;
X=symsum(x*z^-n,n,-%inf,-1)
disp(X,”ans=”)
```

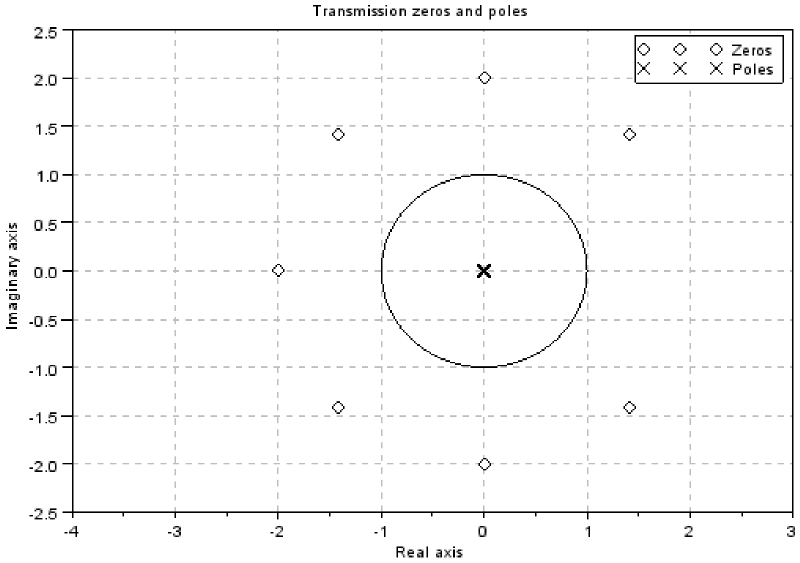


Figure 4.1: pole zero plot

Scilab code Exa 4.3 finite sequence z transform

```
clear;
 clc;
 syms n z X;
 x=[5 3 -2 0 4 -3]
 X=0;
 for i=-2:3;
     X=X+x(i+3)*z^-i
 end
```

Scilab code Exa 4.4 pole zero plot

```
clear;
clc;
syms N n z a;
x=a^n;
X=symsum(x*z^-n,n,0,N-1)
//pole zero map for N=8,a=2,
z=%s;
X1=%s;
X1=0;
for i=0:7
    X1=X1+(2*z^-1)^i
end
plzr(X1)
```

Scilab code Exa 4.6.a z transform and pole zero plot

```
clear;
clc;
syms n z;
x1=(1/2)^n;
X1=symsum(x1*(z^-n),n,0,%inf)
x2=3^-n;
X2=symsum(x2*(z^-n),n,0,%inf)
X=X1+X2
z=%s;
XX=%s;
XX=z/(z-.5)+z/(z-(1/3));
plzr(XX)
a=denom(XX)
b=roots(a)
i=1;
```

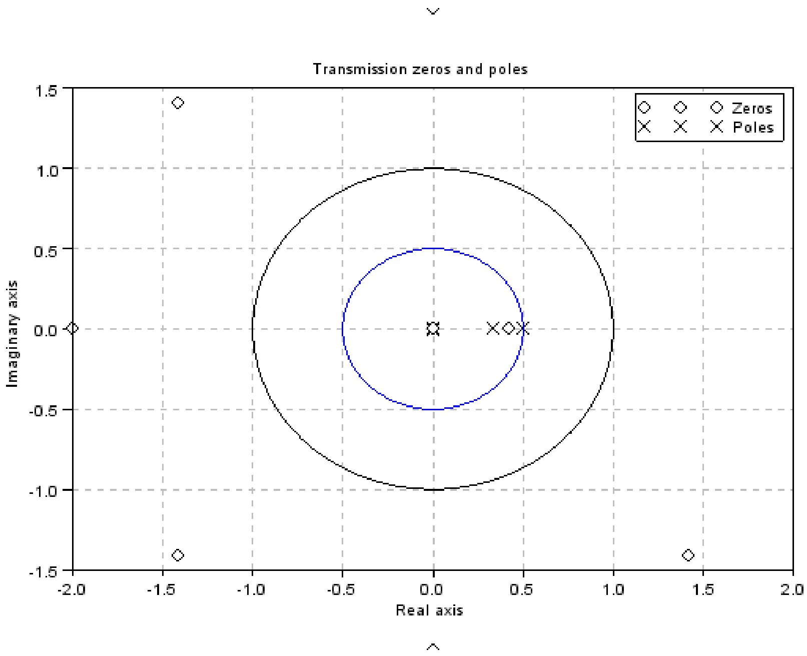


Figure 4.2: z transform and pole zero plot

```
for theta=0:1/50:360
    rx(i)=.5*cos(theta);
    ry(i)=.5*sin(theta);
    i=i+1;
end
plot(rx,ry)
// the region outside blue circle indicates roc
```

Scilab code Exa 4.6.b z transform and pole zero plot

```
clear;
clc;
syms n z;
```

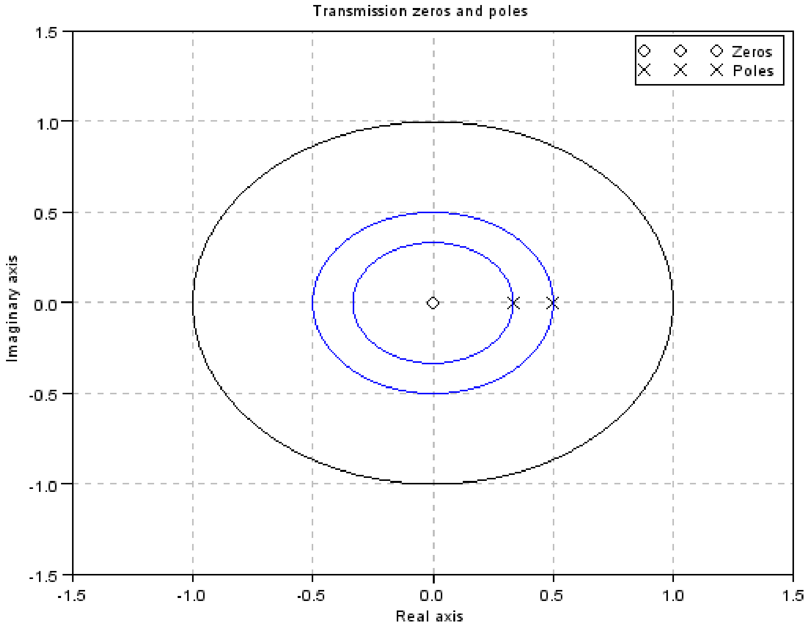


Figure 4.3: z transform and pole zero plot

```
x1=(1/2)^n;
X1=symsum(x1*(z^-n),n,-%inf,-1)
x2=3^-n;
X2=symsum(x2*(z^-n),n,0,%inf)
X=X1+X2
z=%s;
XX=%s;
XX=-z/(z-.5)+z/(z-(1/3));
plzr(XX)
a=denom(XX)
b=roots(a)
i=1;
for theta=0:1/50:360
    rx(i)=b(1)*cos(theta);
    ry(i)=b(1)*sin(theta);
    i=i+1;
end
plot(rx,ry)
i=1;
for theta=0:1/50:360
    rx(i)=b(2)*cos(theta);
    ry(i)=b(2)*sin(theta);
    i=i+1;
end
plot(rx,ry)
// the region between the blue circles indicates roc
```

Scilab code Exa 4.6.c z transform and pole zero plot

```
clear;
clc;
syms n z;
x1=(1/2)^n;
X1=symsum(x1*(z^-n),n,-%inf,-1)
x2=3^-n;
```

```
X2=symsum(x2*(z^-n),n,0,%inf)
//we see that the ROC of X1 and X2 donot overlap
   therefore X(z)does not exists
```

Scilab code Exa 4.7 pole zero plot

```
clear ;
clear x;
clear n;
clc;
//0<a<1
a=.7 ;
n=-10:10;
for i=1:length(n)
    if n(i)>0 then
        x(i)=a^n(i);
    else
        x(i)=a^-n(i);
    end
end
figure
a=gca();
a.x_location="origin";
xtitle('x[n] for a<1','n','x[n]');
a.thickness = 2;
plot2d3(n,x)
plot(n,x,'r.')
//a>1
a=1.3;
for i=1:length(n)
    if n(i)>0 then
        x(i)=a^n(i);
    else
        x(i)=a^-n(i);
    end
```

```
end
figure
a=gca();
a.x_location="origin";
xtitle('x[n] for a>1','n','x[n]');
a.thickness = 2;
plot2d3(n,x)
plot(n,x,'r.')
//|z|>a  then X(z)=z/(z-a) if |z|<1/a then X(z)=-z/(z-1/a)
z=%s;
a=.5;
xx=z/(z-a)-z/(z-(1/a));
figure
plzr(xx);
d=denom(xx);
r=roots(d);
i=1;
for theta=0:1/50:360
    rx(i)=r(1)*cos(theta);
    ry(i)=r(1)*sin(theta);
    i=i+1;
end
plot(rx,ry)
i=1;
for theta=0:1/50:360
    rx(i)=r(2)*cos(theta);
    ry(i)=r(2)*sin(theta);
    i=i+1;
end
plot(rx,ry)
//the region between the blue lines is the ROC
```

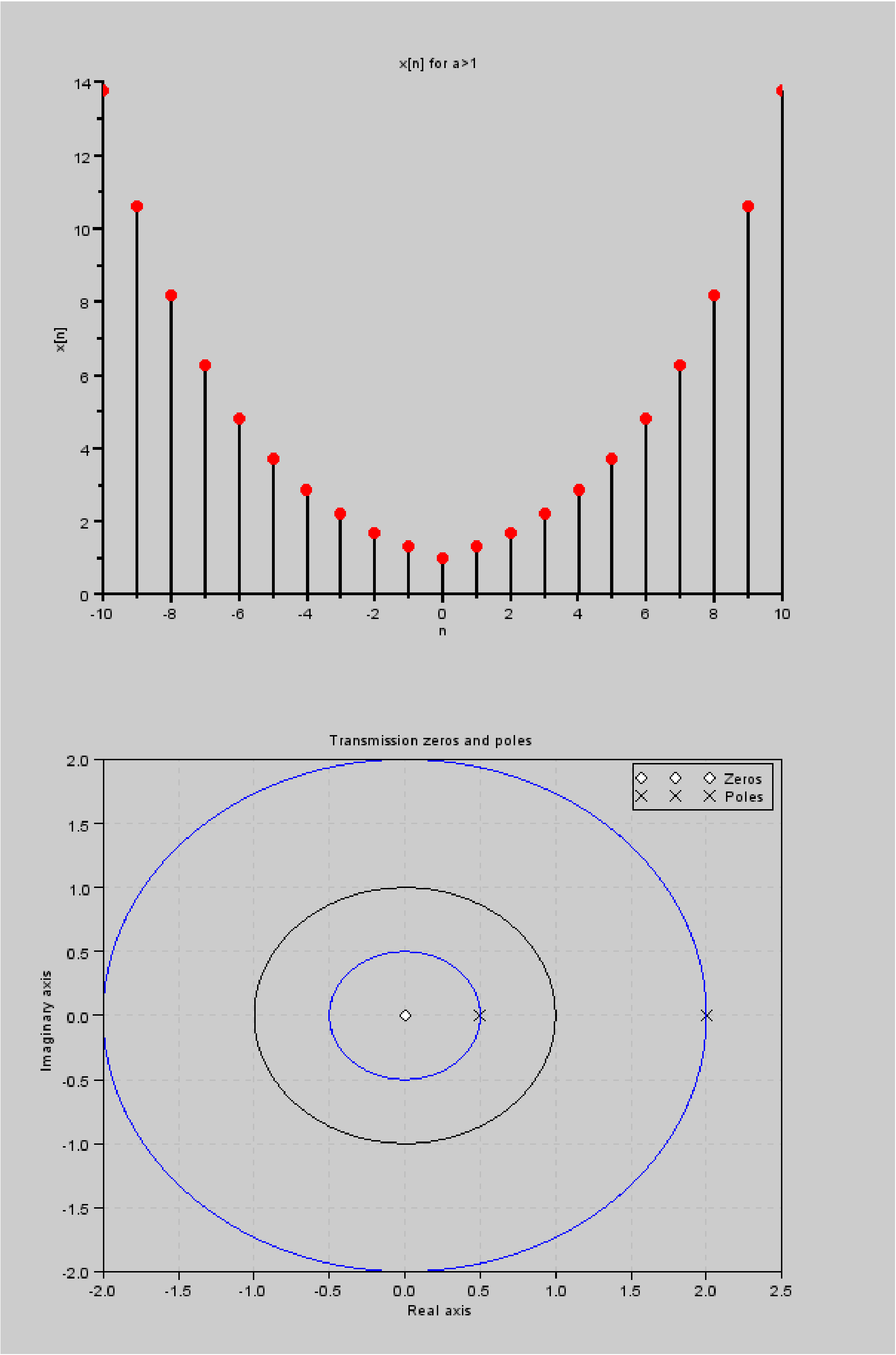


Figure 4.4: pole zero plot

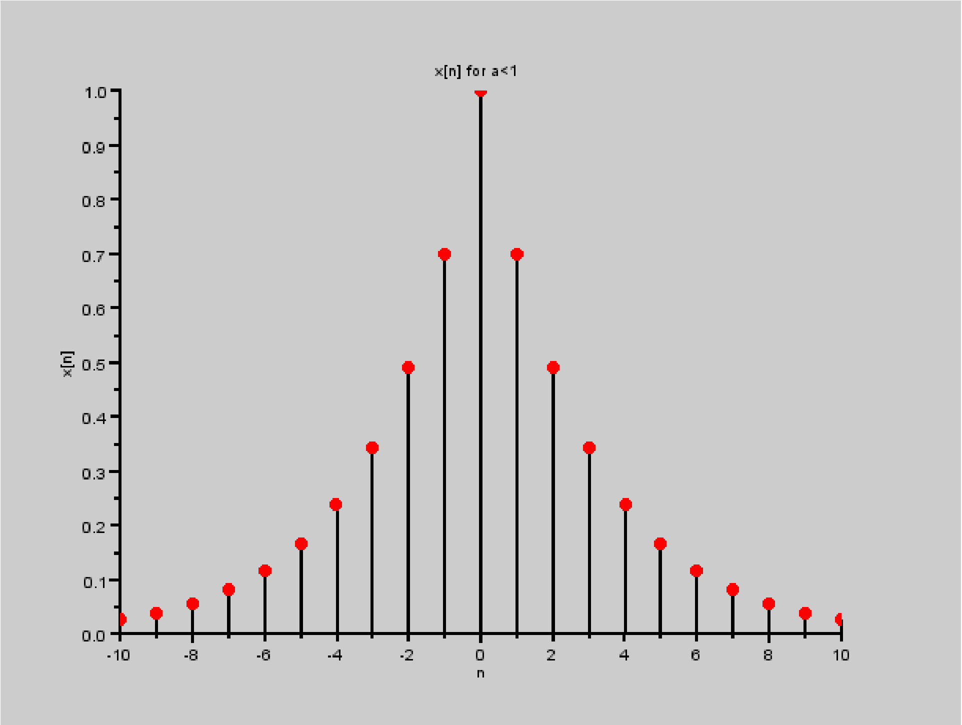


Figure 4.5: pole zero plot

Scilab code Exa 4.10 z transform

```
clc;
syms n z n0 a;
X=symsum(-1*z^-n,n,n0,%inf)
disp(X,"u[n-n0] <-->")
X2=symsum(z^-n,n,n0,n0);
disp(X2,"delta[n-n0] <-->")
X=symsum((a)^(n+1)*z^-n,n,-1,%inf)
disp(X,"a^(n+1)*u[n+1] <-->")
X=symsum(1*z^-n,n,-%inf,0)
disp(X,"u[-n] <-->")
X=symsum(a^-n*z^-n,n,-%inf,0)
disp(X,"a^-n*u[-n] <-->")
```

Scilab code Exa 4.12 differentiation property

```
clc;
syms n z a;
X=symsum(a^n*z^-n,n,0,%inf)
disp(X,"a^n*u[n] <-->")
//by differentiation of z property
Xn=-z*diff(X,z);
disp(Xn,"n*a^n*u[n] <-->")
Xa=diff(X,a);
disp(Xa,"n*a^(n-1)*u[n] <-->")
```

Scilab code Exa 4.15 inverse z transform

```
clear;
clc;
syms z;
X=[z^2 .5*z -5/2 z^-1];
n=-2:1;
a=size(X);
for i = 1:a(2)
    x(i)=X(i)*(z^n(i));
end
disp(x,'x[n]=')
```

Scilab code Exa 4.16 inverse z transform

```
clear;
clc;
z = %z;
a=5;
syms n z1;
X  =1/(1-a*z^-1);
X1 = denom(X);
zp = roots(X1);
X1 = 1/(1-a*z1^-1);
F = X1*(z1^(n-1))*(z1-zp(1));
ha = limit(F,z1,zp(1));
disp(ha*'u(n)','han]=')
a=.5
X  =1/(1-a*z^-1);
X1 = denom(X);
zp = roots(X1);
X1 = 1/(1-a*z1^-1);
F = X1*(z1^(n-1))*(z1-zp(1));
hb = limit(F,z1,zp(1));
disp(hb*'-u(-n-1)','hb[n]=')
```

Scilab code Exa 4.18.a power series expansion technique

```
z=%z;
x=ldiv(z,2*z^2-3*z+1,5)
for i=5:-1:1
    y(6-i)=x(i)*2^i;
end
mprintf('x[n]={ . . . . %d %d %d %d 0}',y(2),y(3),y(4),y(5))
```

Scilab code Exa 4.18.b power series expansion technique

```
z=%z;
x=ldiv(z,2*z^2-3*z+1,5);
mprintf('x[n]={0,%.2f,%.2f,%.2f,....}',x(1),x(2),x(3));
```

Scilab code Exa 4.19 inverse z transform

```
z = %z;
syms n z1;//To find out Inverse z transform z must be linear z = z1
X  =z*.5/((z-(1/2))*(z-1))
X1 = denom(X);
zp = roots(X1);
X1 = z1*.5/((z1-(1/2))*(z1-1))
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
x1 = limit(F1,z1,zp(1));
```

```
disp(x1,'x1[n]=')
x2 = limit(F2,z1,zp(2));
disp(x2,'x2[n]=')
x = x1+x2;
disp(x,'x[n]=')
//a) when |z|<.5
n=-10:0;
disp(-x*'u(-n-1)',"when |z|<1/2 x[n]=")
xn=-(1-2^-n);
mprintf('x[n]={....,%.2f,%.2f,%.2f,0}',xn($-3),xn($-2),xn($-1));
//b) when |z|>1
n=0:10;
disp(x*'u(n)',"when |z|>1 x[n]=")
xn=(1-2^-n);
mprintf('x[n]={%.2f,%.2f,%.2f...}',xn(1),xn(2),xn(3));
```

Scilab code Exa 4.20 inverse z transform

```
z = %z;
syms n z1;
X  =z/((z-1)*(z-2)^2);
X1 = denom(X);
zp = roots(X1);
X1 = z1/((z1-1)*(z1-2)^2);
F1 = X1/z1*(z1-zp(3))^2;
F2 = X1/z1*(z1-zp(1));
Y2 = limit(F1,z1,zp(3));
C1 = limit(F2,z1,zp(1));
F3=(X1/z1-(Y2*F1+C1*F2))*(z1-zp(3));
Y1 = limit(F3,z1,0);

Xa=z1/(z1-zp(1));
F2 = Xa*z1^(n-1)*(z1-zp(1));
```

```
x1=limit(F2,z1,zp(1));
Xb=z1/(z1-zp(3));
F1= Xb*z1^(n-1)*(z1-zp(3));
x2 =limit(F1,z1,zp(3));
//x3 is differntiation of x2 w.r.t a where a is x2=a^n
x3=n*2^(n-1);
x=C1*x1+Y1*x2+Y2*x3;
disp(x*'u(n)',"x[n]=");
```

Scilab code Exa 4.21 inverse z transform

```
z = %z;
syms n z1 ;
X   =1/((z-1)*(z-2));
//Xz=2z+1+X;
X1 = denom(X);
zp = roots(X1);
X1 = 1/((z1-1)*(z1-2));
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
x1 = limit(F1,z1,zp(1));
disp(x1,'x1[n]=')
x2 = limit(F2,z1,zp(2));
disp(x2,'x2[n]=')
x = x1+x2;
disp(x,'xt[n]=')
disp(x*'-u(-n-1)'+'2*delta(n+1)+3/4*delta(n)',"x[n]=")
```

Scilab code Exa 4.22 inverse z transform

```
clc;
```

```
z = %z;
syms n z1 ;
X  =3/(z-2);
X1 = denom(X);
zp = roots(X1);
X1 = 3/(z1-2);
F1 = X1*(z1^(n-1))*(z1-zp(1));
x1 = limit(F1,z1,zp(1));
disp(x1,'x1[n]=')
disp(x1*'u(n-1)',"x[n]=")
```

Scilab code Exa 4.23 inverse z transform

```
clc;
z = %z;
syms n z1 ;
X  =z/((z+1)*(z+3));
X1 = denom(X);
zp = roots(X1);
X1 = z1/((z1+1)*(z1+3));
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
x1 = limit(F1,z1,zp(1));
disp(x1,'x1[n]=')
x2 = limit(F2,z1,zp(2));
disp(x2,'x2[n]=')
xt = x1+x2;
disp(xt*'u(n)','xt[n]=')
//x[n]=2*xt[n-1]+xt[n-3]+3*xt[n-5];F1 = X1*(z1^(n-1))*(z1-zp(1));
F1 = X1*(z1^(n-2))*(z1-zp(1));
F2 = X1*(z1^(n-2))*(z1-zp(2));
x1 = limit(F1,z1,zp(1));
disp(x1,'x1[n]=')
x2 = limit(F2,z1,zp(2));
```

```
disp(x2,'x2[n]=')
xt1 = x1+x2;
F1 = X1*(z1^(n-4))*(z1-zp(1));
F2 = X1*(z1^(n-4))*(z1-zp(2));
x1 = limit(F1,z1,zp(1));
disp(x1,'x1[n]=')
x2 = limit(F2,z1,zp(2));
disp(x2,'x2[n]=')
xt3 = x1+x2;
F1 = X1*(z1^(n-6))*(z1-zp(1));
F2 = X1*(z1^(n-6))*(z1-zp(2));
x1 = limit(F1,z1,zp(1));
disp(x1,'x1[n]=')
x2 = limit(F2,z1,zp(2));
disp(x2,'x2[n]=')
xt5 = x1+x2;
disp(2*xt1*'u(n-1)'+xt3*'u(n-3)'+3*xt5*'u(n-5)',"x[n]")
```

Scilab code Exa 4.25 inverse z transform

```
clc;
syms n z a;
X=symsum(-1*z^-n,n,0,%inf)
disp(X,"u[n] <-->")
h=a^n;
H=symsum(h*z^-n,n,0,%inf)
disp(H,"H(z)=")
Y=X*H;
disp(Y,"Y(z)=")
Y=z^2/((z-1)*(z-a));
F1=Y*z^(n-1)*(z-a);
F2=Y*z^(n-1)*(z-1);
x1=limit(F1,z,a);
x2=limit(F2,z,1);
```

```
x=x1+x2;
disp(x*’u(n)’,”x[n]=”)
```

Scilab code Exa 4.26 output response

```
clc;
syms n z a b;
x=a^n;
X=symsum(x*z^-n,n,0,%inf)
disp(X,”a^n*u[n] <—>”)
h=b^n;
H=symsum(h*z^-n,n,0,%inf)
disp(H,”H(z)=”)
Y=X*H;
disp(Y,”Y(z)=”)
Y=z^2/((z-a)*(z-b));
F1=Y*(z-a)*z^(n-1);
F2=Y*(z-b)*z^(n-1);
x1=limit(F1,z,a);
x2=limit(F2,z,b);
x=x1+x2;
disp(x*’u(n)’,”x[n]=”)
```

Scilab code Exa 4.27 output response

```
clc;
x=[1 1 1 1];
h=[1 1 1];
Xz=0;
z=poly(0,”z”);
for i=1:length(x)
    Xz=Xz+x(i)*z^(1-i);
end
```

```
Hz=0;
for i=1:length(h)
    Hz=Hz+h(i)*z^(1-i);
end
Yz=Xz*Hz;
y=coeff(numer(Yz))
disp(y,”y[n]=”)
```

Scilab code Exa 4.28 impulse response

```
clc;
syms n z a;
X=symsum(1*z^-n,n,0,%inf)
disp(X,”u[n] <—>”)
y=a^n;
Y=symsum(y*z^-n,n,0,%inf)
disp(Y,”a^n*u[n] <—>”)
H=Y/X;
disp(H,”H(z)=”);
H=(z-1)/(z-a);
F1=H*z^(n-1)*(z-a);
h=limit(F1,z,a);
disp(h*'u(n)'+'1/a*delta(n)',”h[n]=”)
```

Scilab code Exa 4.29 impulse response and output

```
clc;
syms n z1 z;
z=%z;
X=symsum(1*z1^-n,n,0,%inf)
disp(X,”u[n] <—>”)
y=2*3^-n;
Y=symsum(y*z1^-n,n,0,%inf)
```

```
disp(Y,"Y(z)=")
H=Y/X;
disp(H,"H(z)=")
Hz=2*(z-1)/(z-1/3);
Hd=denom(Hz);
zp=roots(Hd);
Hz=2*(z1-1)/(z1-1/3);
F1 = Hz*(z1^(n-1))*(z1-zp(1));
x1 = limit(F1,z1,zp(1));
disp(x1,'x1[n]=')
x =x1;
disp(x*'u(n)'+'6*delta(n)','y[n]=');
disp("part b")
x=(1/2)^n;
X=symsum(x*z1^-n,n,0,%inf)
disp(X,"X(z)")
Yz=X*Hz;
disp(Yz,"Y(z)=")
Yz=2*z1*(z1-1)/((z1-1/2)*(z1-1/3));
F1 = Yz*(z1^(n-1))*(z1-1/2);
F2 = Yz*(z1^(n-1))*(z1-1/3);
y1 = limit(F1,z1,1/2);
disp(y1,'y1[n]=')
y2 = limit(F2,z1,1/3);
disp(y2,'y2[n]=')
y = y1+y2;
disp(y*'u(n)','y[n]=')
```

Scilab code Exa 4.31 impulse response

```
clear;
clc;
syms n z1 a b;
h=a^n;
H=symsum(h*z1^-n,n,0,%inf)
```

```
disp(H,"H(z)");
disp("since |z|>|a| z=%inf is included hence the system is causal")
disp("if |a|<1 then the ROC of H(z) contains the unit circle and hence the        system will be stable")
```

Scilab code Exa 4.32 impulse and step response

```
z = %z;
syms n z1;//To find out Inverse z transform z must be linear z = z1
X  =z^2/((z-(1/4))*(z-(1/2)))
X1 = denom(X);
zp = roots(X1);
X1 = z1^2/((z1-(1/4))*(z1-(1/2)))
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
h1 = limit(F1,z1,zp(1));
disp(h1,'h1[n]=')
h2 = limit(F2,z1,zp(2));
disp(h2,'h2[n]=')
h = h1+h2;
disp(h,'h[n]=')
//step response
Xz = z/(z-1);//u[n]
Y  = Xz*X;
Y1 = denom(Y);
zp = roots(Y1);
Y1 = z1^3/((z1-1)*(z1-(1/4))*(z1-(1/2)))
F1 = Y1*(z1^(n-1))*(z1-zp(1));
F2 = Y1*(z1^(n-1))*(z1-zp(2));
F3=  Y1*(z1^(n-1))*(z1-zp(3));
y1 = limit(F1,z1,zp(1));
disp(y1,'y1[n]=')
```

```
y2 = limit(F2,z1,zp(2));
disp(y2,'y2[n]=')
y3 = limit(F3,z1,zp(3));
disp(y3,'y3[n]=')
y = y1+y2+y3;
disp(y*'u(n)','y[n]=')
```

Scilab code Exa 4.35 unilateral z transform

```
//unilateral z-transform
clear;
clc;
syms n z a;
x=a^n;
X=symsum(x*z^-n,n,0,%inf)
disp(X,"ans=")
//x[n]=a^(n+1)*u[n+1]
x=a^(n+1);
X=symsum(x*z^-n,n,0,%inf)
disp(X,"ans=")
```

Scilab code Exa 4.37 unilateral z transform method

```
clc;
clear;
syms n z a b y1 K;
Yz=a*y1*(z/(z-a))+K*z^2/((z-b)*(z-a));
//y1=y[-1]
Y1=z/(z-a);
Y2=z^2/((z-b)*(z-a));
FY1=Y1*z^(n-1)*(z-a);
y1n= limit(FY1,z,a);
FY21 = Y2*(z^(n-1))*(z-a);
```

```
FY22 = Y2*(z^(n-1))*(z-b);
y21n = limit(FY21,z,a);
y22n = limit(FY22,z,b);
y=a*y1*y1n+K*(y22n+y21n);
disp(y*'u(n)',"y[n]=")
```

Scilab code Exa 4.38 difference equation

```
clear;
clc;
syms z n ;
x=(1/3)^n;
X=symsum(x*z^-n,n,0,%inf)
disp(X,"X(z)=")
Xz=z/(z-1/3);
Yz=z/(2*(z-1/2))+z^2/((z-1/3)*(z-1/2));
//y[-1]=1
Y1=z/(z-1/2);
Y2=z^2/((z-1/3)*(z-1/2));
FY1=Y1*z^(n-1)*(z-1/2);
y1n= limit(FY1,z,1/2);
FY21 = Y2*(z^(n-1))*(z-1/2);
FY22 = Y2*(z^(n-1))*(z-1/3);
y21n = limit(FY21,z,1/2);
y22n = limit(FY22,z,1/3);
y=(1/2)*y1n+(y22n+y21n);
disp(y*'u(n+1)',"part a) y[n]=")
//b)
x=(1/2)^n;
X=symsum(x*z^-n,n,0,%inf)
disp(X,"X(z)=")
Xz=z/(z-1/2);
Yz=z*(3*z^2-2*z+.5)/(3*(z-1)*(z-1/2)*(z-1/3));
//y[-1]=1 y[-2]=2
F1 = Yz*(z^(n-1))*(z-1);
```

```
F2 = Yz*(z^(n-1))*(z-1/2);
F3=  Yz*(z^(n-1))*(z-1/3);
y1 = limit(F1,z,1);
disp(y1,'y1[n]=')
y2 = limit(F2,z,1/2);
disp(y2,'y2[n]=')
y3 = limit(F3,z,1/3);
disp(y3,'y3[n]=')
y = y1+y2+y3;
disp(y*'u(n+2)','y[n]=')
```

Chapter 5

Fourier analysis of continuous time system and signals

Scilab code Exa 5.4.a fourier series representation

```
clear;
close;
clc;
T=1;
t=0:0.01:1;
w0=2*%pi*T;
x=cos(w0*t);
for k=-5:5
    cc(k+6,:)=exp(-%i*k*w0*t);
    ck(k+6)=x*cc(k+6,:)'/length(t);
    if abs(ck(k+6))<0.01 then
    ck(k+6)=0;
    end
end
```

Scilab code Exa 5.4.b fourier series representation

```
clear;
close;
clc;
T=1;
t=0:0.01:1;
w0=2*%pi*T;
x=sin(w0*t);
for k=-5:5
    cc(k+6,:)=exp(-%i*k*w0*t);
    ck(k+6)=x*cc(k+6,:)'/length(t);
    if abs(ck(k+6))<0.01 then
    ck(k+6)=0;
    end
end
```

Scilab code Exa 5.4.c fourier series representation

```
clear ;
clear cc;
close;
clc;
t=0:0.01:20;
x=cos(2*t+%pi/4);
w0=2*%pi*1;
for k=-5:5
    cc(k+6,:)=exp(-%i*k*2*t);
    ck(k+6)=x*cc(k+6,:)'/length(t);
    if abs(ck(k+6))<0.1 then
    ck(k+6)=0;
    end
end
```

Scilab code Exa 5.4.d fourier series representation

```
clear ;
close;
clc;
t=0:0.01:50;
x1=cos(4*t);
x2=sin(6*t);
w1=4;
w2=6;
w=int32([w1,w2]);
w0=gcd(w);
x=x1+x2;
for k=-5:5
    cc(k+6,:)=exp(-%i*k.*t*2);
    ck(k+6)=x*cc(k+6,:)'/length(t);
    if abs(ck(k+6))<0.1 then
    ck(k+6)=0;
    end
end
```

Scilab code Exa 5.4.e fourier series representation

```
clear ;
close;
clc;
t=0:0.01:50;
x=(sin(t))^2;
for k=-5:5
    cc(k+6,:)=exp(-%i*k*2*t);
    ck(k+6)=x*cc(k+6,:)'/length(t);
    if abs(ck(k+6))<0.1 then
    ck(k+6)=0;
    end
end
```

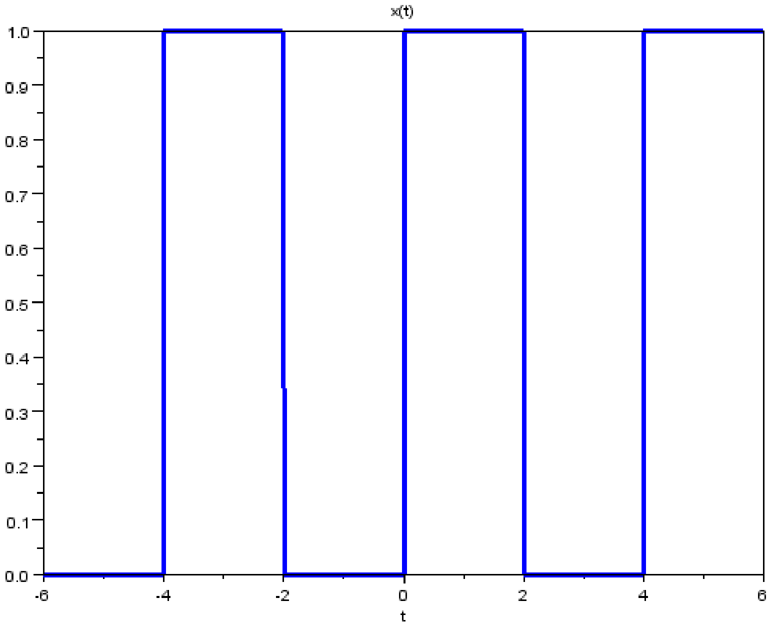


Figure 5.1: fourier series of a periodic square wave

Scilab code Exa 5.5 fourier series of a periodic square wave

```
clear;
close;
clc;
T0=2
t=-5.99:0.01:6;
t_temp=0.01:0.01:T0;
s=length(t)/length(t_temp);
x=[];
```

```
for i=1:s
    if modulo(i,2)==1  then
        x=[x zeros(1,length(t_temp))];
    else
        x=[x ones(1,length(t_temp))];
    end
end
a=gca();
plot(t,x,'r')
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
w0=%pi/2;
for k=-10:10
    cc(k+11,:)=exp(-%i*k*w0*t);
    ck(k+11)=x*cc(k+11,:)'/length(t);
    if abs(ck(k+11))<0.01 then
    ck(k+11)=0;
    else if real(ck(k+11))<0.1  then
    ck(k+11)=%i*imag(ck(k+11));
    end
    end
    if k==0 then
        c0=ck(k+11);
    end
end
//trigmometeric form
a0=2*c0;
a=2*real(ck);
b=2*imag(ck);
```

Scilab code Exa 5.6 fourier series of a periodic square wave

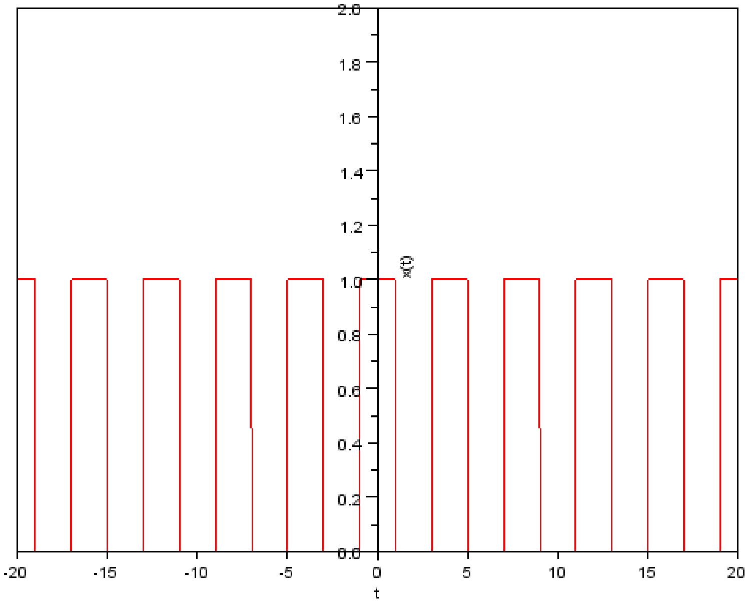


Figure 5.2: fourier series of a periodic square wave

```
clear ;
close;
clc;
T0=4;
t=-20.99:0.01:21;
t_temp=0.01:0.01:T0/2;
s=length(t)/length(t_temp);
x=[];
for i=1:s
    if modulo(i,2)==0  then
        x=[x zeros(1,length(t_temp))];
    else
        x=[x ones(1,length(t_temp))];
    end
end
a=gca();
plot(t,x,’r’)
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle(’x(t)’,’t’)
w0=%pi/2;
for k=-10:10
    cc(k+11,:)=exp(-%i*k*w0*t);
    ck(k+11)=x*cc(k+11,:)’/length(t);
    if abs(ck(k+11))<0.01 then
    ck(k+11)=0;
    else if imag(ck(k+11))<0.01  then
    ck(k+11)=real(ck(k+11));
    end
    end
    if k==0 then
        c0=ck(k+11);
    end
end
//trigmometeric form
a0=2*c0;
a=2*real(ck);
```

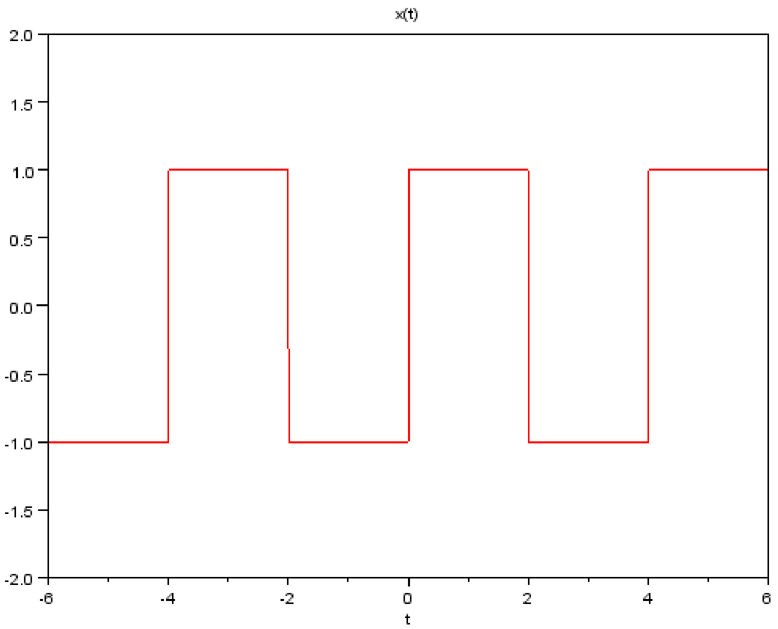


Figure 5.3: fourier series of a periodic square wave

```
b=2*imag(ck);
```

Scilab code Exa 5.7 fourier series of a periodic square wave

```
clear;
close;
clc;
T0=4;
t=-5.99:0.01:6;
t_temp=0.01:0.01:T0/2;
s=length(t)/length(t_temp);
```

```
x=[];
for i=1:s
    if modulo(i,2)==1  then
        x=[x -ones(1,length(t_temp))];
    else
        x=[x ones(1,length(t_temp))];
    end
end
plot(t,x,'r')
w0=%pi/2;
for k=-10:10
    cc(k+11,:)=exp(-%i*k*w0*t);
    ck(k+11)=x*cc(k+11,:)'/length(t);
    if abs(ck(k+11))<0.01 then
    ck(k+11)=0;
    else if real(ck(k+11))<0.1  then
    ck(k+11)=%i*imag(ck(k+11));
    end
    end
    if k==0 then
        c0=ck(k+11);
    end
end
//trigmometeric form
a0=2*c0;
a=2*real(ck);
b=2*imag(ck);
```

Scilab code Exa 5.8 fourier series of a periodic impulse train

```
clear;
close;
clc;
```

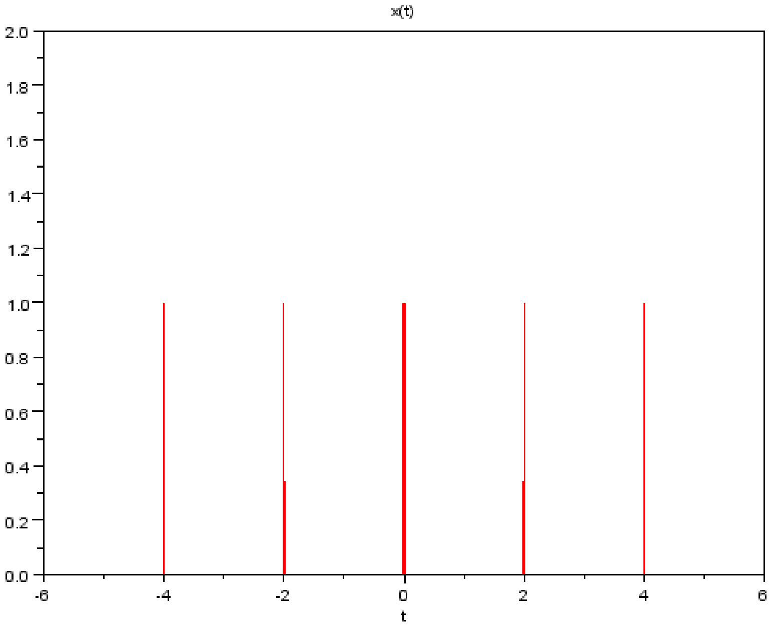


Figure 5.4: fourier series of a periodic impulse train

```
T0=2;
t=-6:0.01:6;
x=[];
for i=1:length(t)
    if modulo(t(i),T0)==0  then
        x(i)=1;
    else
        x(i)=0;
    end
end
plot(t,x,'r')
//by sifting property
ck=1/T0;
//trigmometeric form
a0=2*ck;
a=2*real(ck);
b=2*imag(ck);
```

Scilab code Exa 5.9 differentiation property of fourier series

```
clear;
close;
clc;
T0=4;
t=-5.99:0.01:6;
t_temp=0.01:0.01:T0/2;
s=length(t)/length(t_temp);
dx=[];
x=[];
for i=1:s
    if modulo(i,2)==1  then
        dx=[dx -ones(1,length(t_temp))];
        x=[x .5*t_temp($:-1:1)];
    else
        dx=[dx ones(1,length(t_temp))];
        x=[x .5*t_temp];
    end
end
figure
a=gca();
plot2d(t,x,8)
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
figure
a=gca();
plot(t,dx,'r')
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
w0=%pi/2;
for k=-10:10
    cc(k+11,:)=exp(-%i*k*w0*t);
    cdk(k+11)=dx*cc(k+11,:)'/length(t);
    if abs(cdk(k+11))<0.01 then
    cdk(k+11)=0;
    else if real(cdk(k+11))<0.1  then
    cdk(k+11)=-%i*imag(cdk(k+11));
    end
    end
    if k==0 then
        ck(k+11)=.5;
        c0=ck(k+11);
    else
       ck(k+11)=cdk(k+11)/(%i*k*w0);
    end
end
//trigmometeric form
a0=2*c0;
a=2*real(ck);
b=2*imag(ck);
```

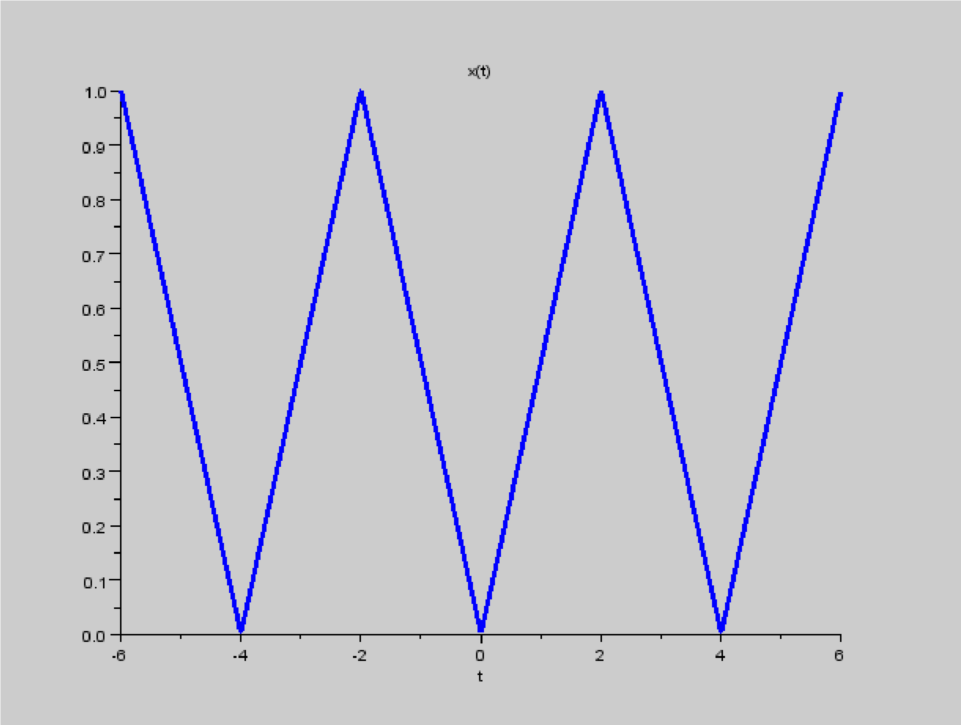


Figure 5.5: differentiation property of fourier series

Scilab code Exa 5.10 differentiation property of fourier series

```
clear;
close;
clc;
T0=4;
t=-7.99:0.01:8;
t_temp=0.01:0.01:T0;
```

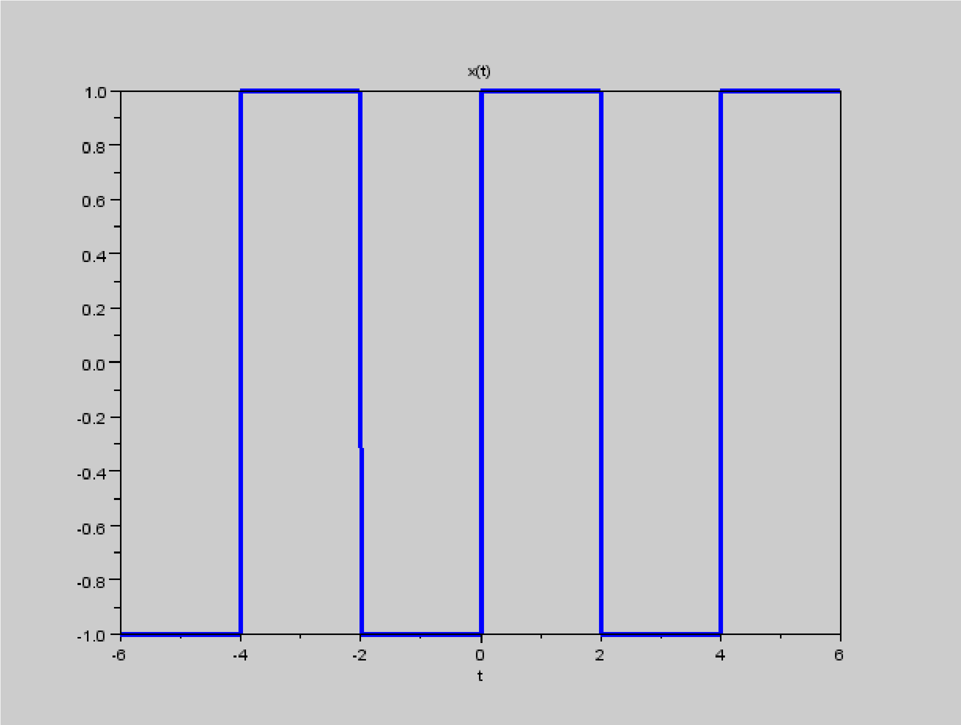


Figure 5.6: differentiation property of fourier series

```
s=length(t)/length(t_temp);
dx=[];
x=[];
for i=1:s
    x=[x 0.5*t_temp($:-1:1)];
end
for i=1:length(t)
    if modulo(t(i),T0)==0  then
        dx(i)=1;
    else
        dx(i)=0;
    end
end
dx1=-1/T0;
figure
a=gca();
plot2d(t,x,6)
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
figure
plot(t,dx,'r');
plot(t,dx1);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('diff(x(t))','t')
w0=%pi/2;
//by sifting property
cdk=1/T0;
for k=-10:10
    if k~=0 then
    ck(k+11)=cdk/(%i*k*w0);
 else
     ck(k+11)=.5;
     c0=ck(k+11);
    end
```

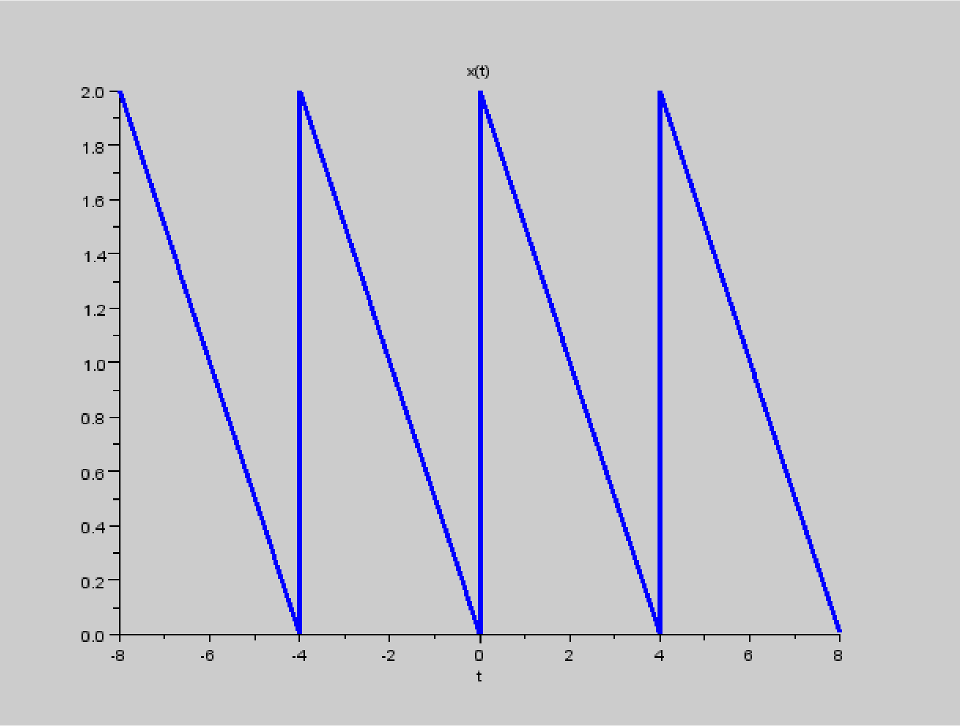


Figure 5.7: differentiation property of fourier series

```
end
a0=2*c0;
ak=2*real(ck);
bk=-2*imag(ck);
```

Scilab code Exa 5.11 magnitude spectra of a periodic square wave

```
clear ;
close;
```

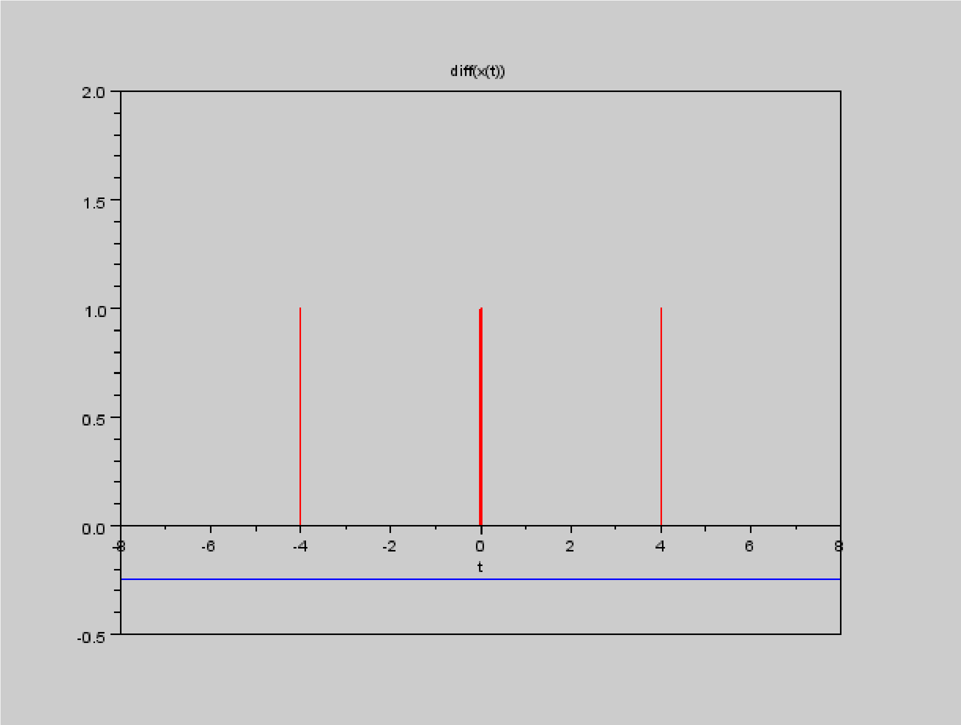


Figure 5.8: differentiation property of fourier series

```
clc;
T0=16;
t=-79.99:0.01:80;
t_temp=0.01:0.01:T0/4;
s=length(t)/length(t_temp);
x=[];
for i=1:s
    if modulo(i,4)==1  then
        x=[x ones(1,length(t_temp))];
    else
        x=[x zeros(1,length(t_temp))];
    end
end
a=gca();
plot(t,x,'r')
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
w0=%pi/8;
for k=-30:30;
    cc(k+31,:)=exp(-%i*k*w0*t);
    ck(k+31)=x*cc(k+31,:)'/length(t);
    if abs(ck(k+31))<0.01 then
    ck(k+31)=0;
    //else if real(ck(k+11))<0.1  then
    //ck(k+11)=%i*imag(ck(k+11));
    //end
    end
    if k==0 then
      c0=ck(k+31);
    end
end
k=-30:30;
figure
a=gca();
plot2d3(k,abs(ck));
poly1=a.children.children;
```

```
poly1.thickness=3;
poly1.foreground=2;
xtitle('|ck|','k')
//part b
t_temp=0.01:0.01:T0/8;
s=length(t)/length(t_temp);
x=[];
for i=1:s
    if modulo(i,8)==1  then
        x=[x ones(1,length(t_temp))];
    else
        x=[x zeros(1,length(t_temp))];
    end
end
figure
a=gca();
plot(t,x,'r')
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
w0=%pi/8;
for k=-30:30;
    cc(k+31,:)=exp(-%i*k*w0*t);
    ck(k+31)=x*cc(k+31,:)'/length(t);
    if abs(ck(k+31))<0.01 then
    ck(k+31)=0;
    //else if real(ck(k+11))<0.1  then
    //ck(k+11)=%i*imag(ck(k+11));
    //end
    end
    if k==0 then
      c0=ck(k+31);
    end
end
k=-30:30;
figure
a=gca();
```

```
plot2d3(k,abs(ck));
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('|ck|','k')
```

Scilab code Exa 5.19 fourier transform of a rectangular pulse

```
clear;
clc;
close;
A=1;
t0=1;
dt=.1;
t=-10:dt:10;
for i=1:length(t)
    if t(i)>-t0 & t(i)<t0 then
        x(i)=A;
    else
        x(i)=0;
    end
end
a=gca();
plot(t,x);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
//clf();
wmax=10;
w=-wmax:0.1:wmax;
```

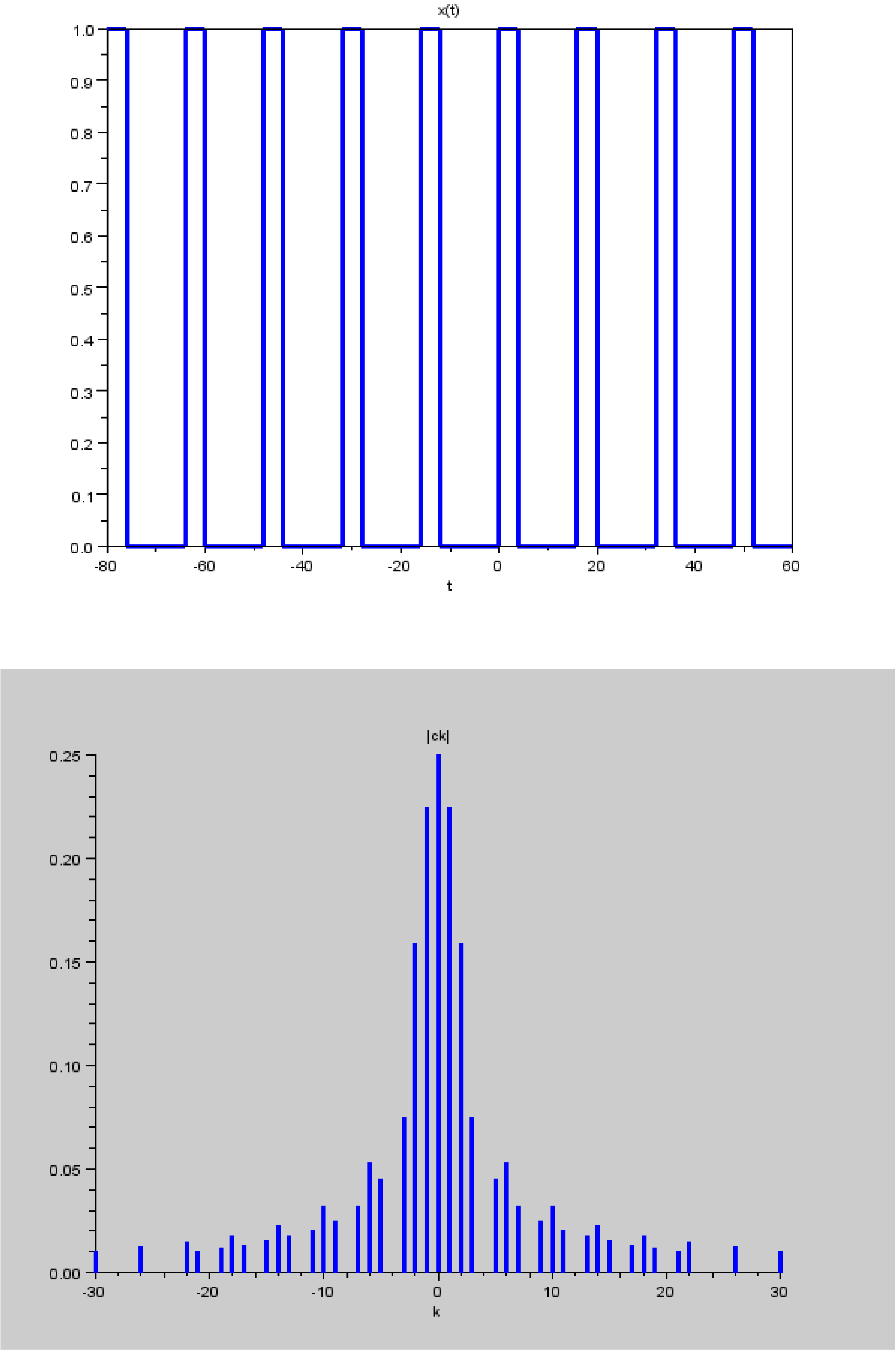


Figure 5.9: magnitude spectra of a periodic square wave

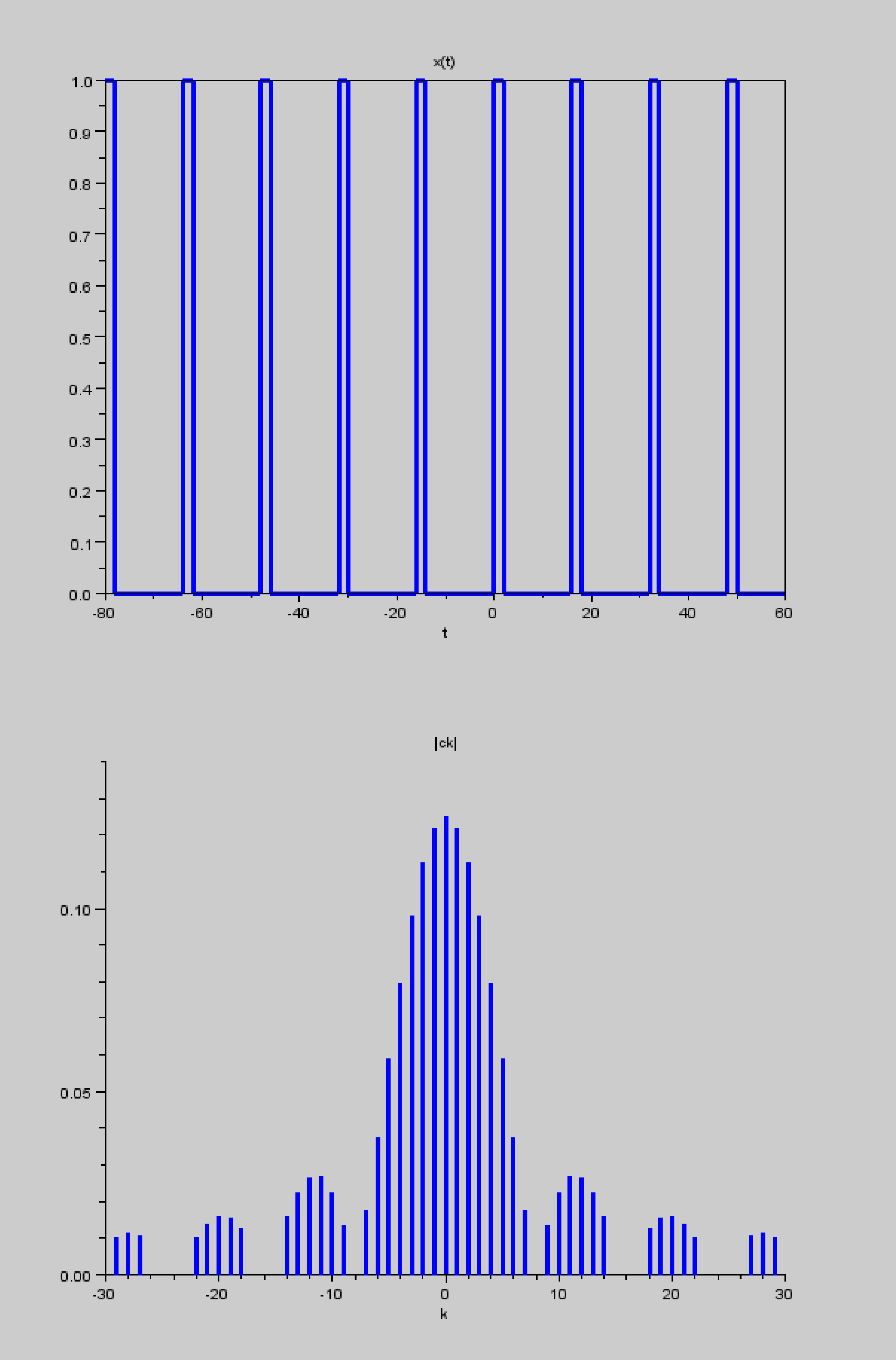


Figure 5.10: magnitude spectra of a periodic square wave

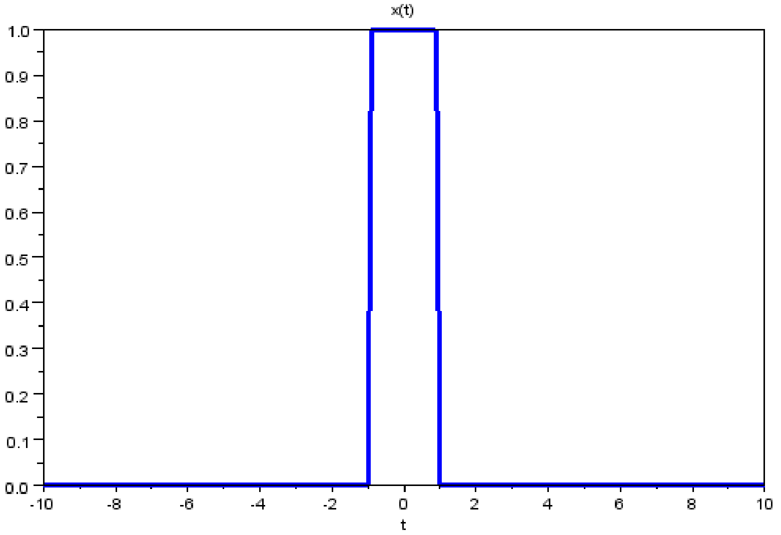


Figure 5.11: fourier transform of a rectangular pulse

```
Xw=x'*exp(-%i*(w'*t))*dt;
Xw_mag=(Xw);
Xw_mag=[Xw_mag($:-1:1) Xw_mag];
w=[w($:-1:1) w];
figure
a=gca();
plot2d(w,Xw_mag);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('|X(w)|','w')
```

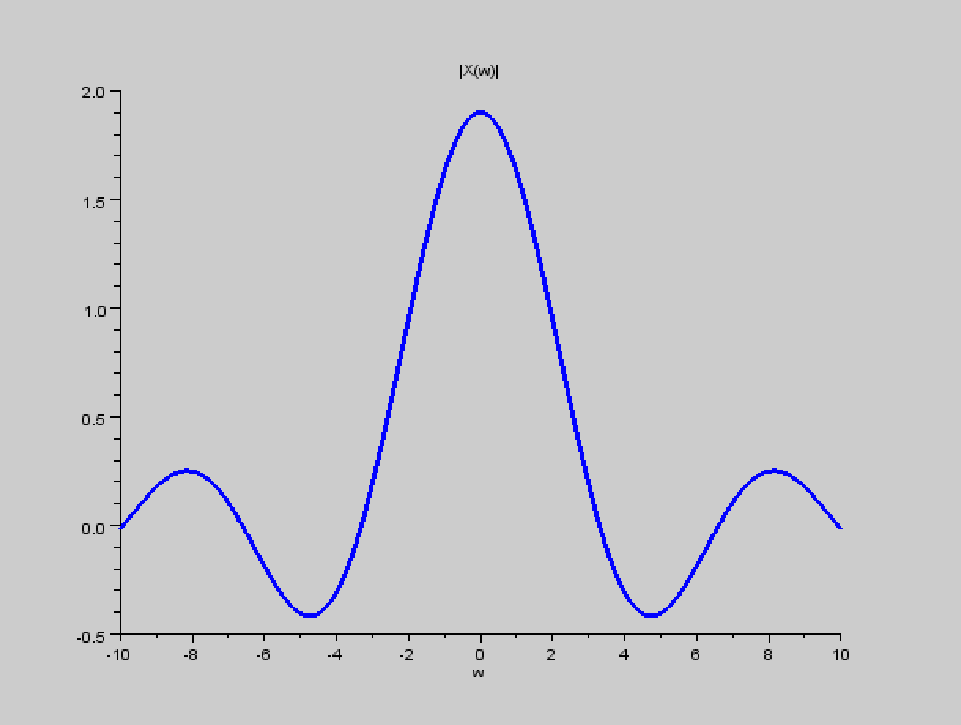


Figure 5.12: fourier transform of a rectangular pulse

Scilab code Exa 5.20 fourier transform of a sinc function

```
clear;
clc;
close;
A=1;
t0=1;
dt=.1;
t=-10:dt:10;
for i=1:length(t)
    if t(i)==0 then
        x(i)=A/%pi;
    else
        x(i)=sin(A*t(i))./(%pi*t(i));
    end
end
a=gca();
plot(t,x);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
//clf();
wmax=10;
w=-wmax:0.1:wmax;
Xw=x'*exp(-%i*(w'*t))*dt;
Xw_mag=(Xw);
//Xw_mag=[Xw_mag($:-1:1) Xw_mag];
figure
a=gca();
plot2d(w,Xw_mag);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('|X(w)|','w')
```

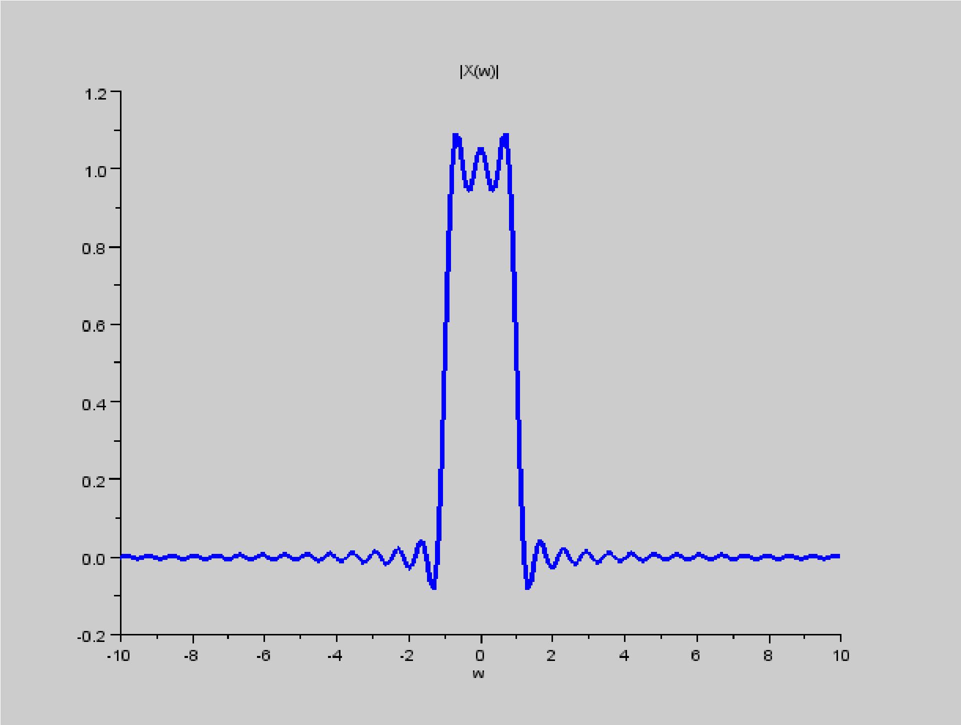


Figure 5.13: fourier transform of a sinc function

Scilab code Exa 5.21 fourier transform

```
clear;
clc;
close;
a=1;
dt=.01;
t=0:dt:10;
```

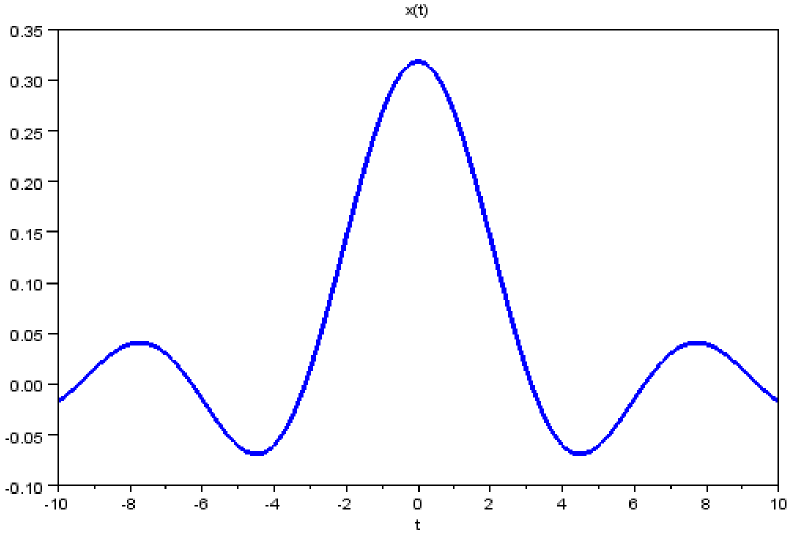


Figure 5.14: fourier transform of a sinc function

```
x=exp(-a*t);
y=[x($:-1:1) x];
t1=-10:dt:10.01;
a=gca();
plot(t1,y);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
//clf();
//stacksize(10000,100000)
wmax=10;
w=0:0.01:wmax;
Xw=x*exp(-%i*(w'*t))*dt;
Xw_mag=(Xw);
Xw_mag=[Xw_mag($:-1:1) Xw_mag];
w=[-w($:-1:1) w];
figure
```

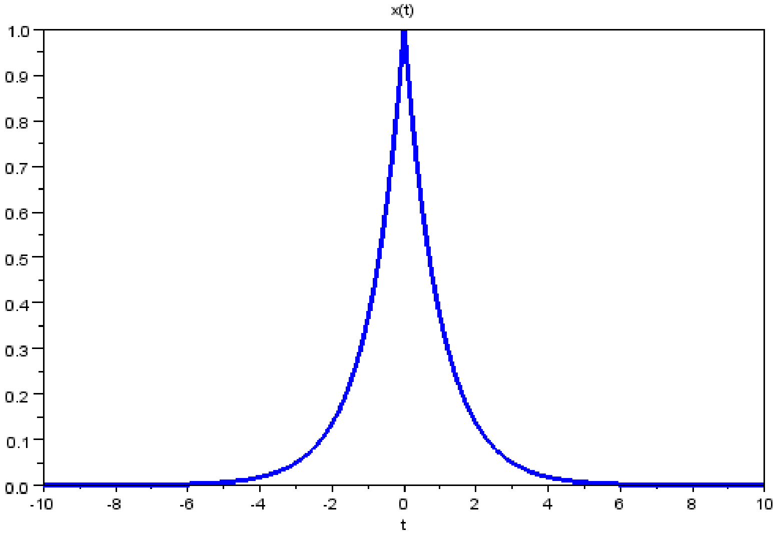


Figure 5.15: fourier transform

```
a=gca();
plot2d(w,Xw_mag);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
```

Scilab code Exa 5.22 fourier transform

```
clear;
clc;
close;
```

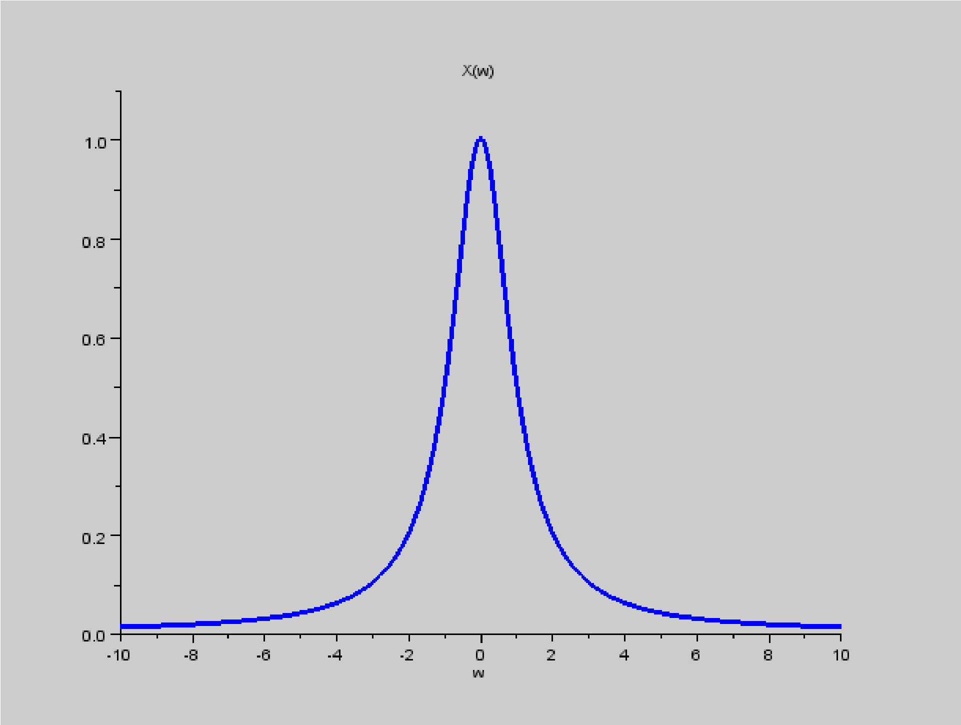


Figure 5.16: fourier transform

```
a=1.5;
dt=.1;
t=0:dt:10;
s=size(t);
x=ones(1,s(2))./(a^2+t^2);
y=[x($:-1:1) x];
t1=-10:dt:10.1;
a=gca();
plot(t1,y);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')

//clf();
wmax=10;
w=0:0.1:wmax;
Xw=x*exp(-%i*(w'*t))*dt;
Xw_mag=(Xw);
Xw_mag=[Xw_mag($:-1:1) Xw_mag];
w=[-w($:-1:1) w];
figure
a=gca()
plot2d(w,Xw_mag);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
```

Scilab code Exa 5.23.a fourier transform

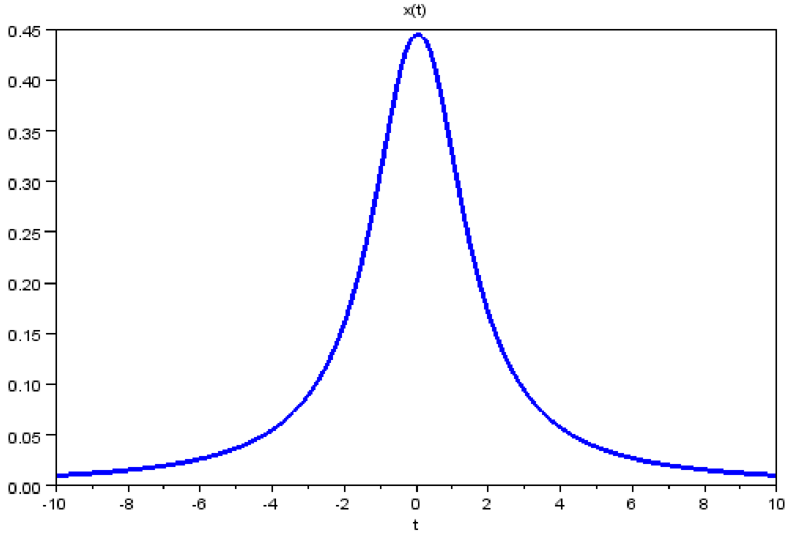


Figure 5.17: fourier transform

```
clear;
clc;
close;
dt=.1
t=-10:dt:10;
for i=1:length(t)
    if t(i)==0 then
        x(i)=1;
    else
        x(i)=0;
    end
end
a=gca();
plot2d3(t,x);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
```

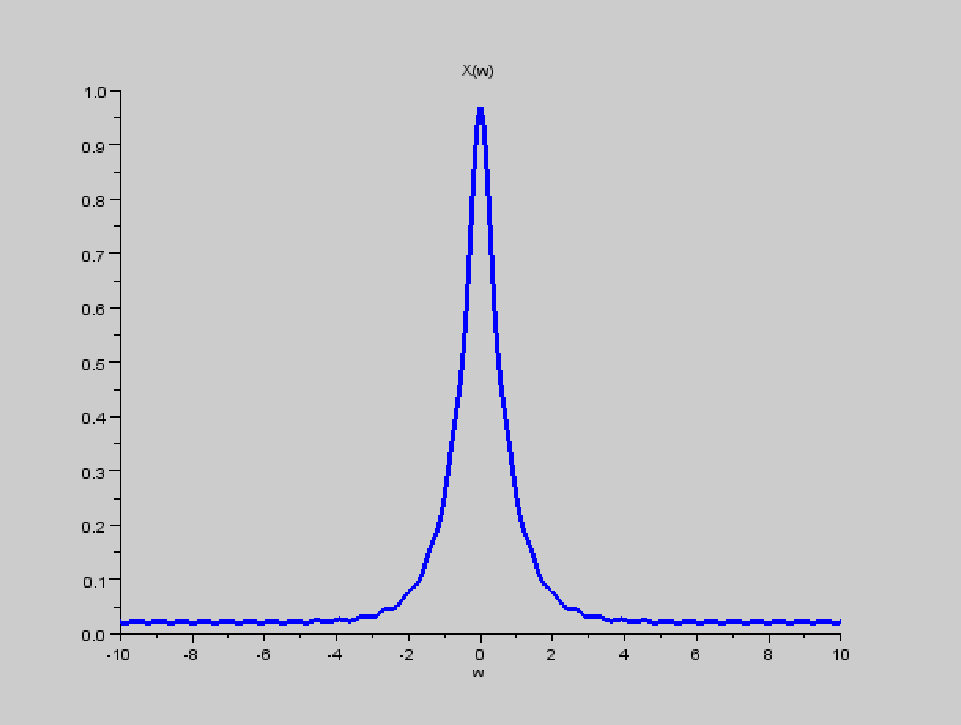


Figure 5.18: fourier transform

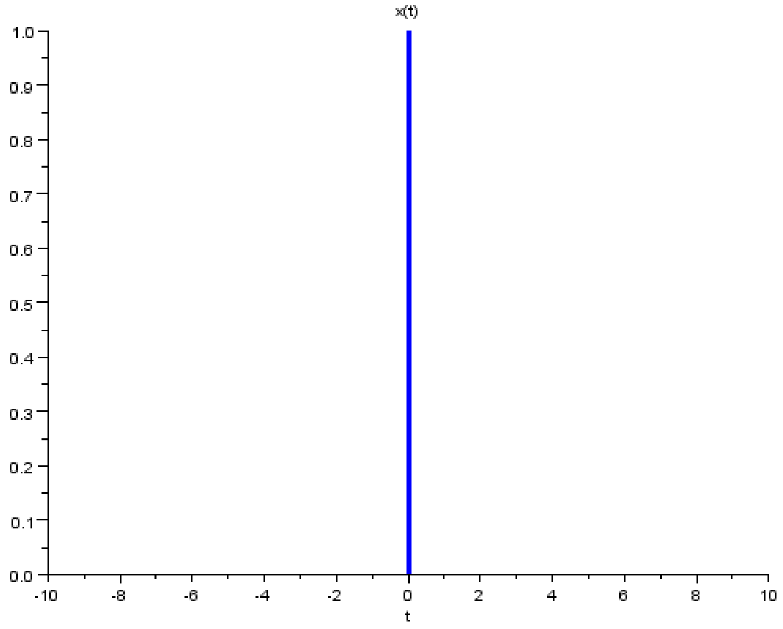


Figure 5.19: fourier transform

```
wmax=10;
w=-wmax:0.1:wmax;
Xw=x'*exp(-%i*(w'*t))*dt;
Xw_mag=(Xw);
figure
a=gca();
plot2d(w,Xw_mag);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
```

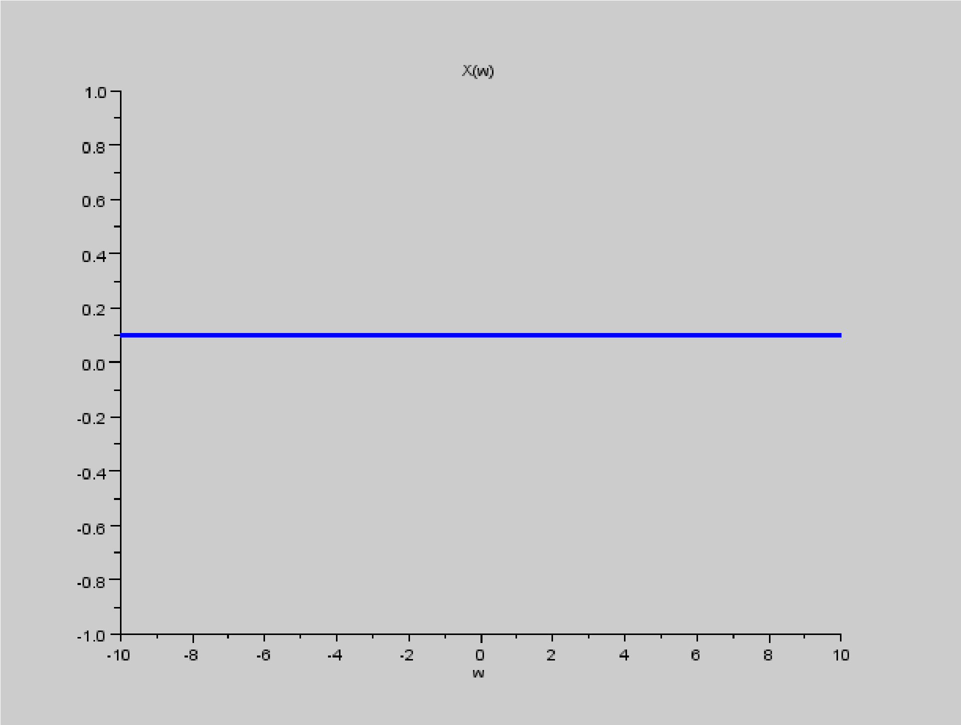


Figure 5.20: fourier transform

Scilab code Exa 5.23.b fourier transform

```
//frequency shifting
clear;
clc;
close;
dt=.1;
w0=1;//positive number
t=-10:dt:10;
x=exp(%i*t*w0);
a=gca();
plot(t,x);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
wmax=10;
w=-wmax:0.1:wmax;
Xw=x*exp(-%i*(w'*t))*dt;
figure
a=gca();
plot2d(w,round(Xw/10));
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
```

Scilab code Exa 5.23.c fourier transform

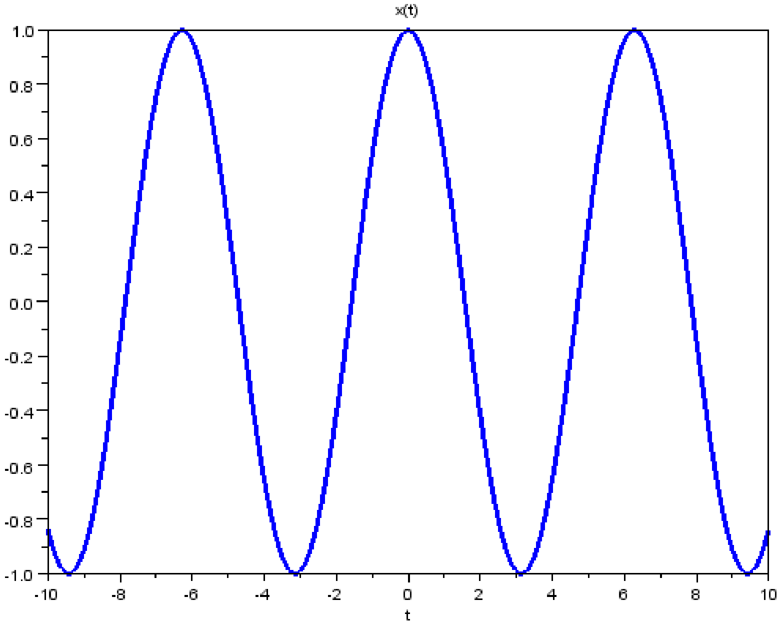


Figure 5.21: fourier transform

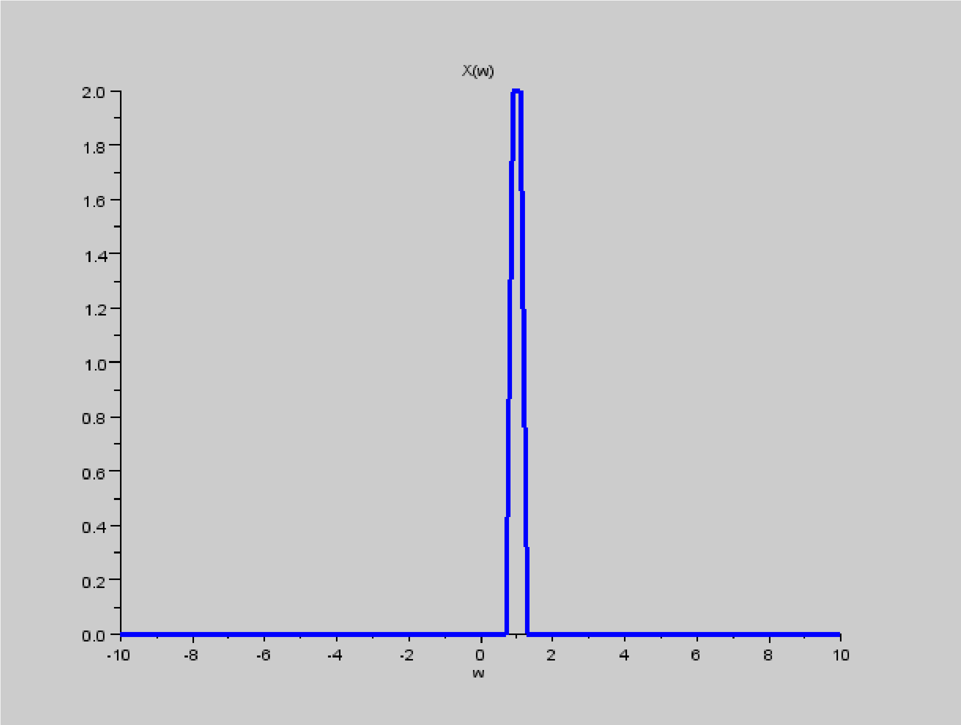


Figure 5.22: fourier transform

```
//frequency shifting
clear;
clc;
close;
dt=.1;
w0=1;//positive number
t=-10:dt:10;
x=exp(-%i*t*w0);
a=gca();
plot(t,x);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
wmax=10;
w=-wmax:0.1:wmax;
Xw=x*exp(-%i*(w'*t))*dt;
figure
a=gca();
plot2d3(w,round(Xw/10));
poly1=a.children.children;
poly1.thickness=1;
poly1.foreground=2;
xtitle('X(w)','w')
plot(w,round(Xw/10),'b.')
```

Scilab code Exa 5.23.d fourier transform

```
//frequency shifting
clear;
clc;
```

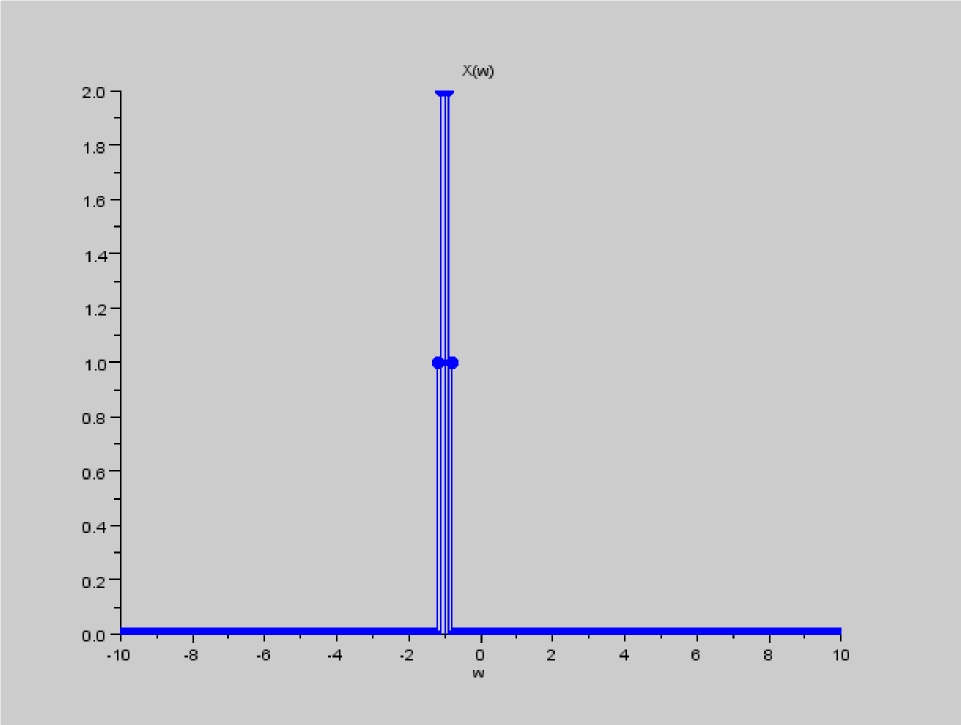


Figure 5.23: fourier transform

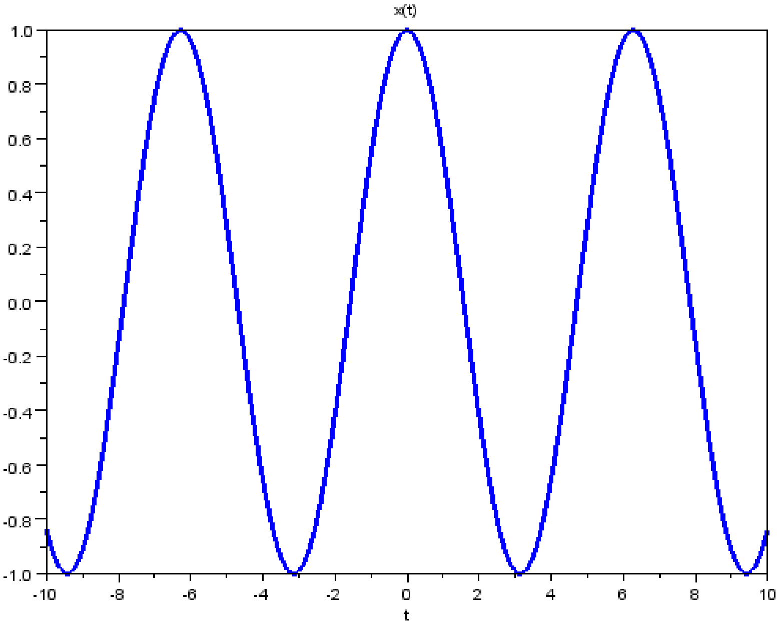


Figure 5.24: fourier transform

```
close;
dt=.1;
w0=2;//positive number
t=-10:dt:10;
x=cos(w0*t);//or 1/2*(exp(-j*w*t)+exp(j*w*t))
a=gca();
plot(t,x);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
wmax=10;
w=-wmax:0.1:wmax;
Xw=x*exp(-%i*(w'*t))*dt;
figure
a=gca();
plot2d3(w,round(abs(Xw/10)));
poly1=a.children.children;
poly1.thickness=2;
poly1.foreground=2;
xtitle('X(w)','w')
```

Scilab code Exa 5.23.e fourier transform

```
//frequency shifting
clear;
clc;
close;
dt=.1;
w0=.8;//positive number
t=-10:dt:10;
```

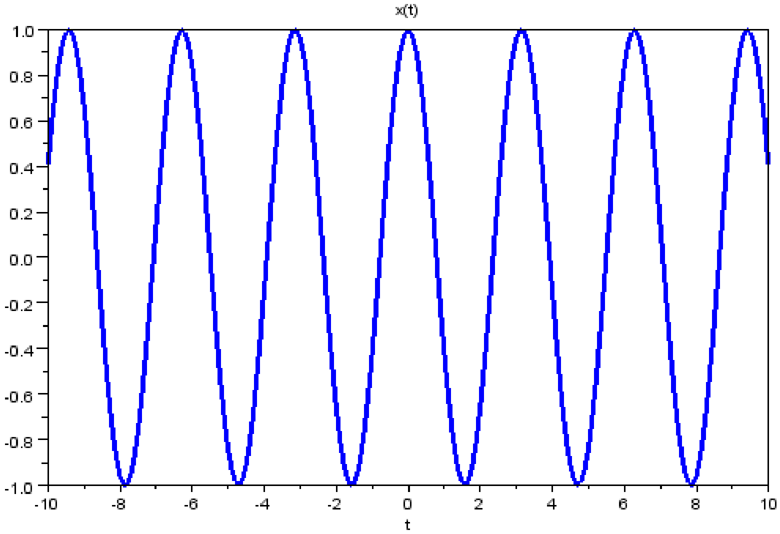


Figure 5.25: fourier transform

```
x=sin(w0*t);//or 1/2*(exp(-j*w*t)+exp(j*w*t))
plot(t,x);
a=gca()
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
wmax=10;
w=-wmax:0.1:wmax;
Xw=x*exp(-%i*(w'*t))*dt;
figure
a=gca()
plot2d3(w,round(abs(Xw/10)));
poly1=a.children.children;
poly1.thickness=2;
poly1.foreground=2;
xtitle('X(w)','w')
```

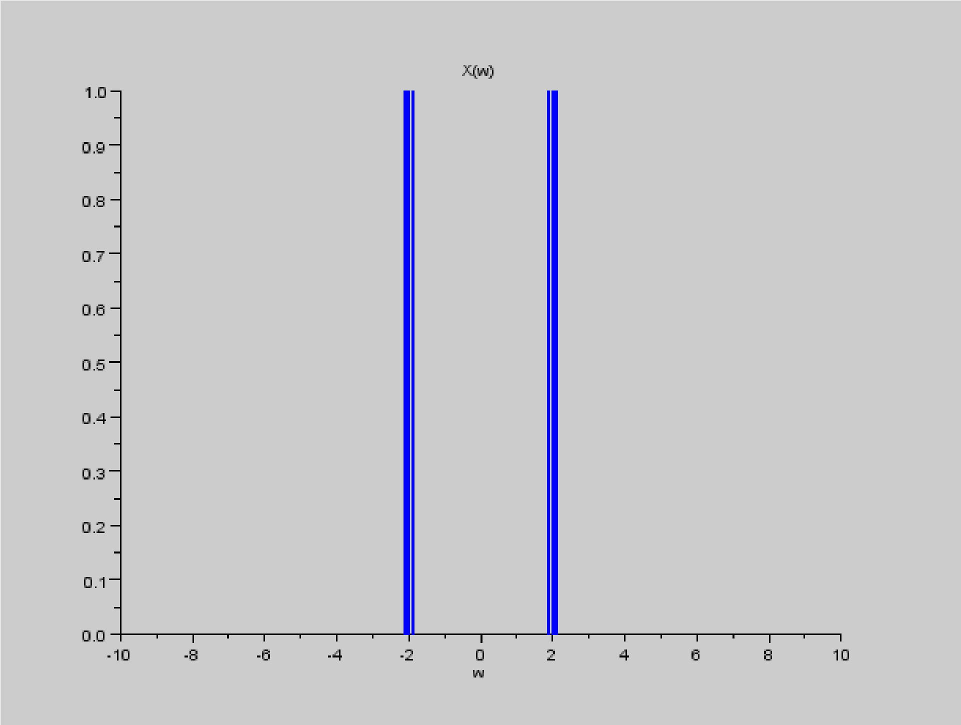


Figure 5.26: fourier transform

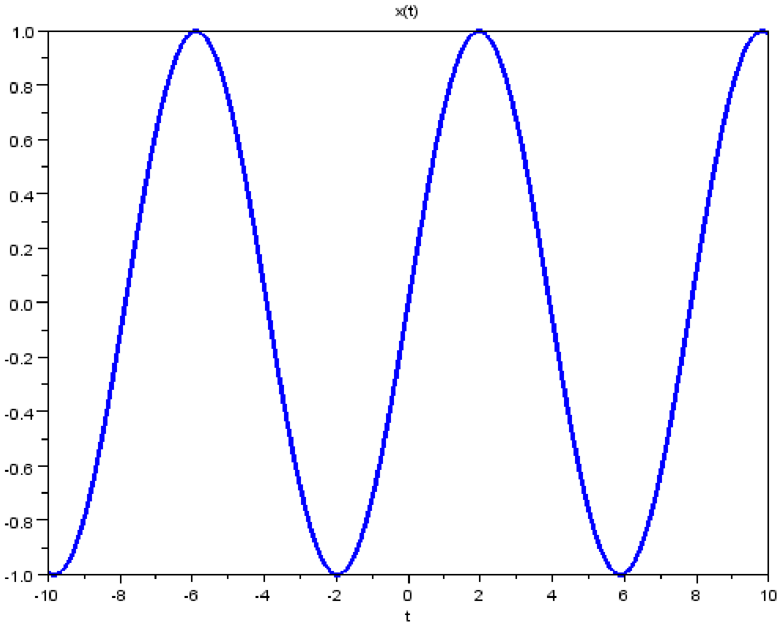


Figure 5.27: fourier transform

Scilab code Exa 5.25 fourier transform of a periodic impulse train

```
clear;
clc;
close;
dt=.1;
t0=1;//positive number
t=-10:dt:10;
```

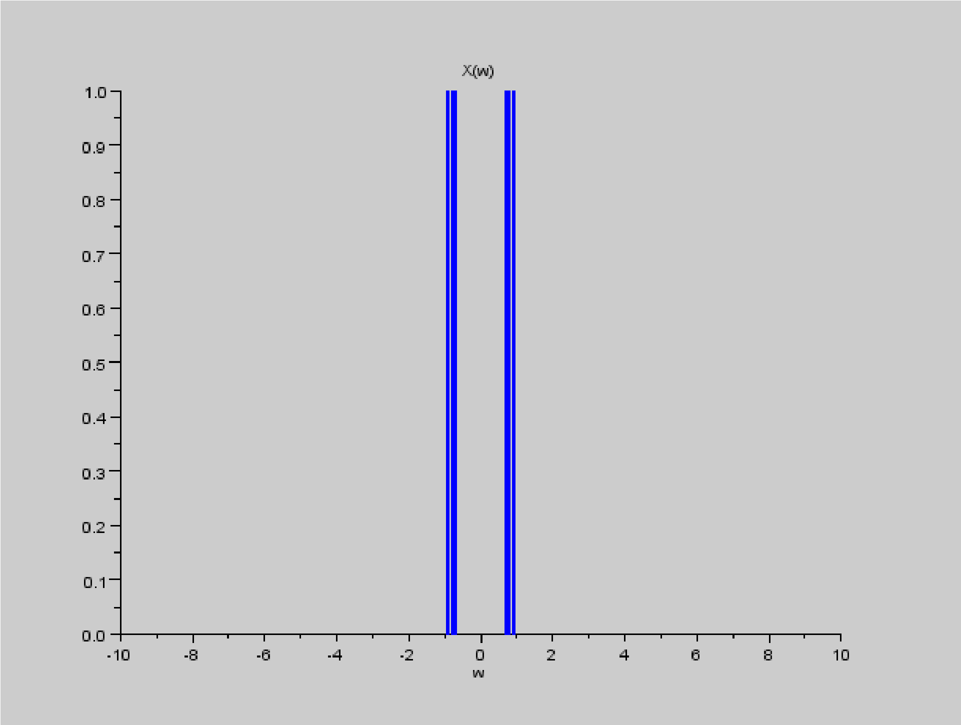


Figure 5.28: fourier transform

```
for i=1:length(t)
    if modulo(t(i),t0)==0 then
        x(i)=1;
    else
        x(i)=0;
    end
end
a=gca();
plot2d3(t,x);
plot(t,x,'r.')
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
wmax=10;
w=-wmax:0.1:wmax;
Xw=x'*exp(-%i*(w'*t))*dt;
figure
a=gca();
plot2d3(w,round(abs(Xw)));
poly1=a.children.children;
poly1.thickness=2;
poly1.foreground=2;
xtitle('X(w)','w')
//or the fourier series is doesnt work
//ck=1/t0;
//k=-10:0.1:10;
//x=ck*(exp(%i*2*%pi*t.*k/t0));
//wmax=10;
//w=-wmax:0.1:wmax;
//Xw=x*exp(-%i*(w'*t))*dt;
//clf();
//plot2d3(k,x);
```

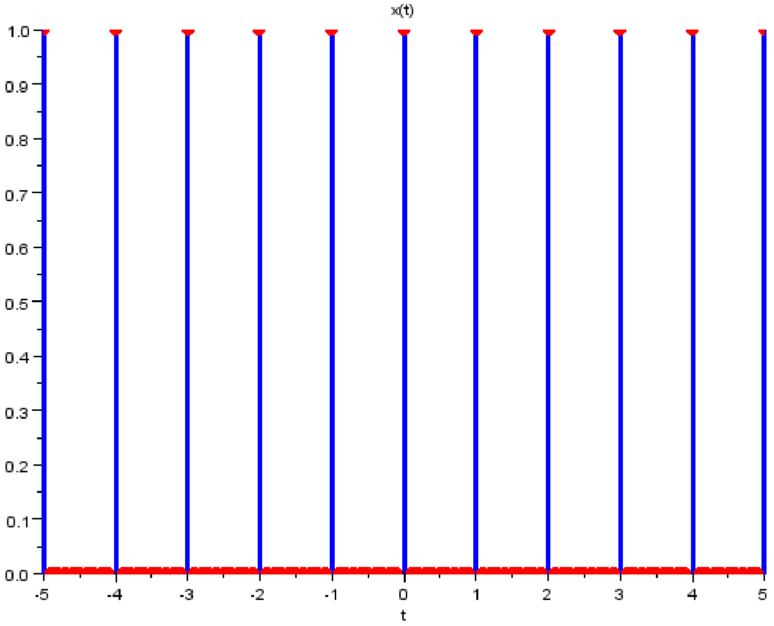


Figure 5.29: fourier transform of a periodic impulse train

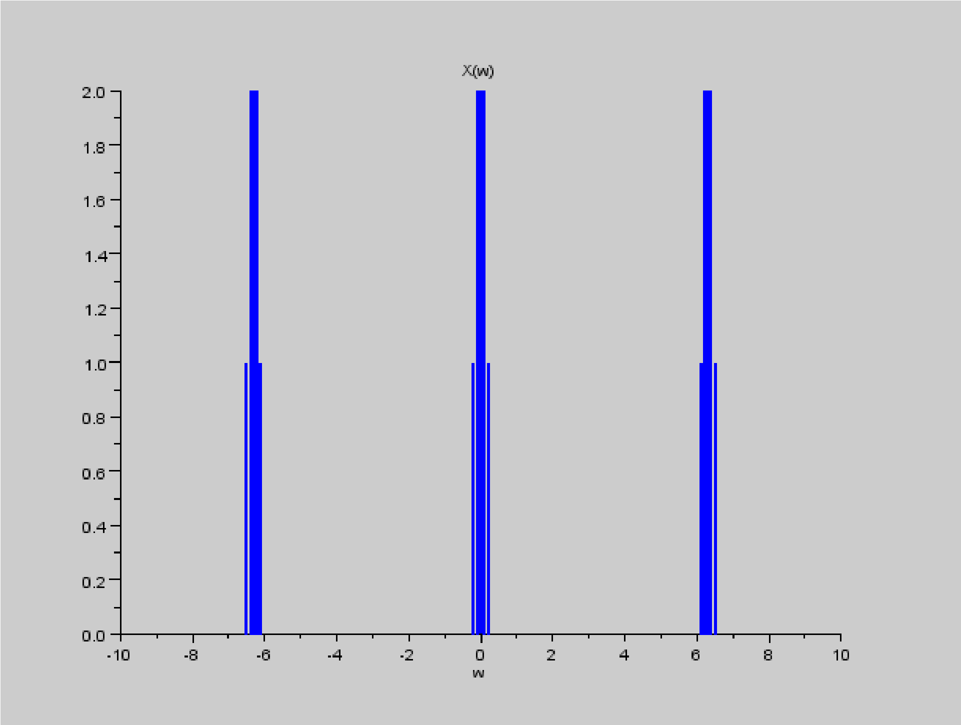


Figure 5.30: fourier transform of a periodic impulse train

Scilab code Exa 5.27 inverse fourier transform

```
clear;
clc;
close;
A=1/2;
dw=.1;
w0=4;
a=1;
w=-10:dw:10;
for i=1:length(w)
    if ((w(i)>(-w0-a) & w(i)<(-w0+a)) | (w(i)>(w0-a) & w(i)<(w0+a))) then
   Xw(i)=2;
else
    Xw(i)=0;
    end

end
a=gca();
plot(w,Xw);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
tmax=10;
t=-tmax:0.1:tmax;
x=Xw'*exp(-%i*(t'*w))*dw;
x_mag=(x);
figure
a=gca();
plot2d(t,x_mag);
poly1=a.children.children;
poly1.thickness=3;
```

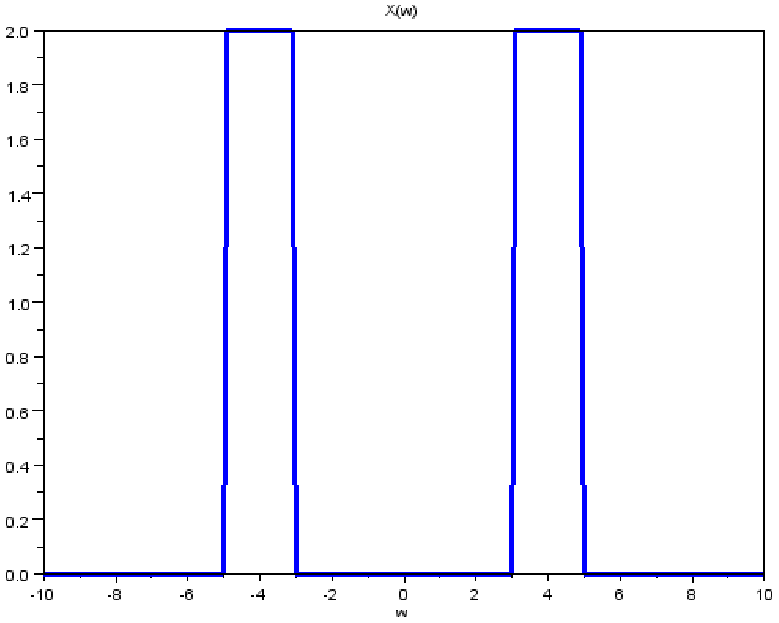


Figure 5.31: inverse fourier transform

```
poly1.foreground=2;
xtitle('x(t)','t')
```

Scilab code Exa 5.29 fourier transform of a signum function

```
clear;
clc;
close;
dt=.1;
```

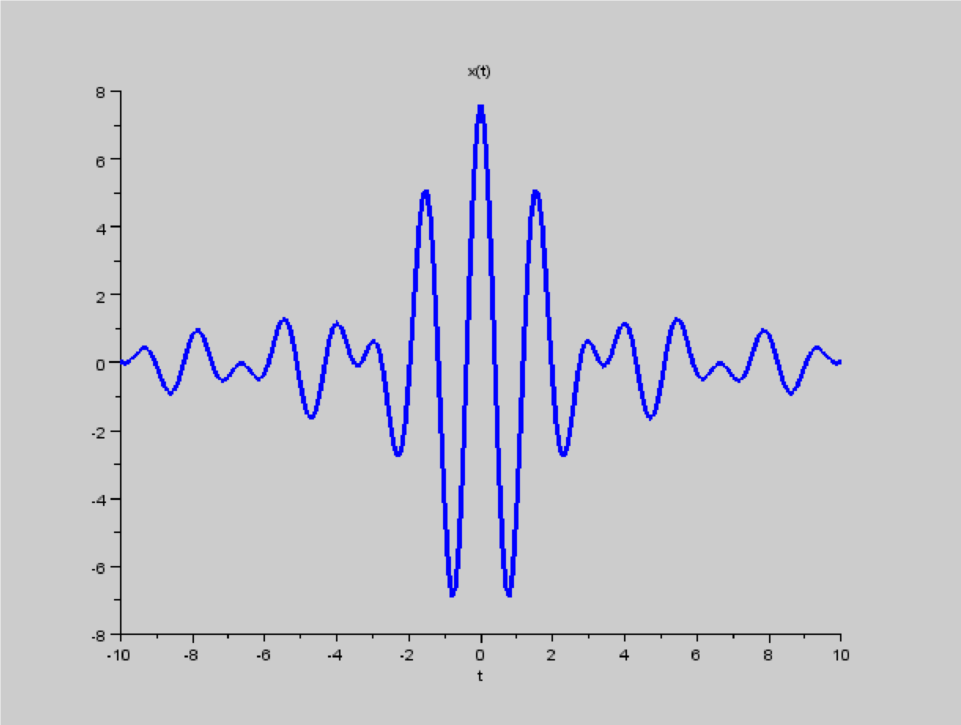


Figure 5.32: inverse fourier transform

```
t=-10:dt:10;
h=find(t==0);
x=[-ones(1,h) ones(1,length(t)-h)];
//sgn(t)=2u(t)-1 .: d(sgn(t))/dt=2delta(t);
a=gca();
plot(t,x);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('sgn(t)','t')
y=diff(x);
y=[y 0];
figure
a=gca();
plot2d3(t,y);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('diff(sgn(t))','t')
wmax=10;
w=-wmax:0.1:wmax;
Xw=y*exp(-%i*(w'*t))*dt*10;
for i=1:length(w)
    if w(i)<>0 then
 XW(i)=Xw(i)./(%i*w(i));
else
    XW(i)=20;
    end
end
figure
a=gca();
plot2d(w,abs(XW));
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
```

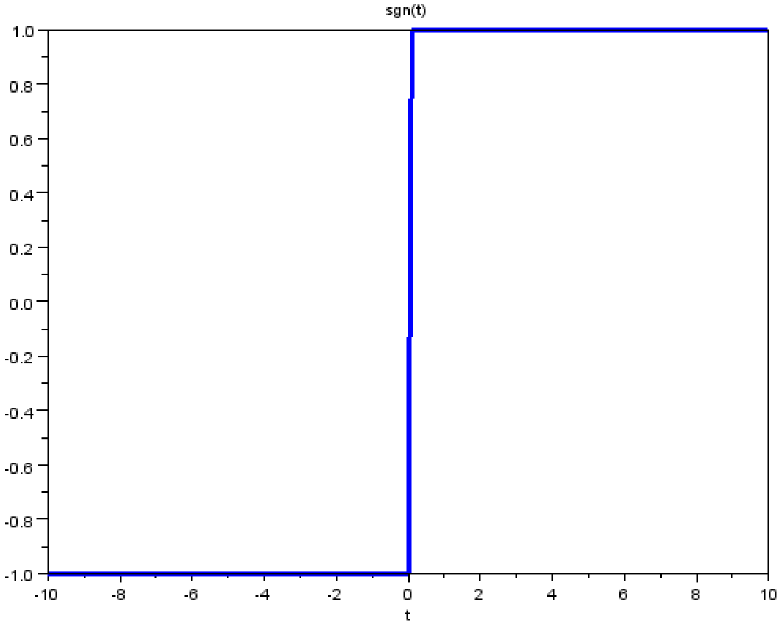


Figure 5.33: fourier transform of a signum function

Scilab code Exa 5.30 fourier transform of a step signal

```
clear;
clc;
close;
dt=.1;
t=-10:dt:10;
u=[zeros(1,find(t==0)) ones(1,length(t)-find(t==0))
```

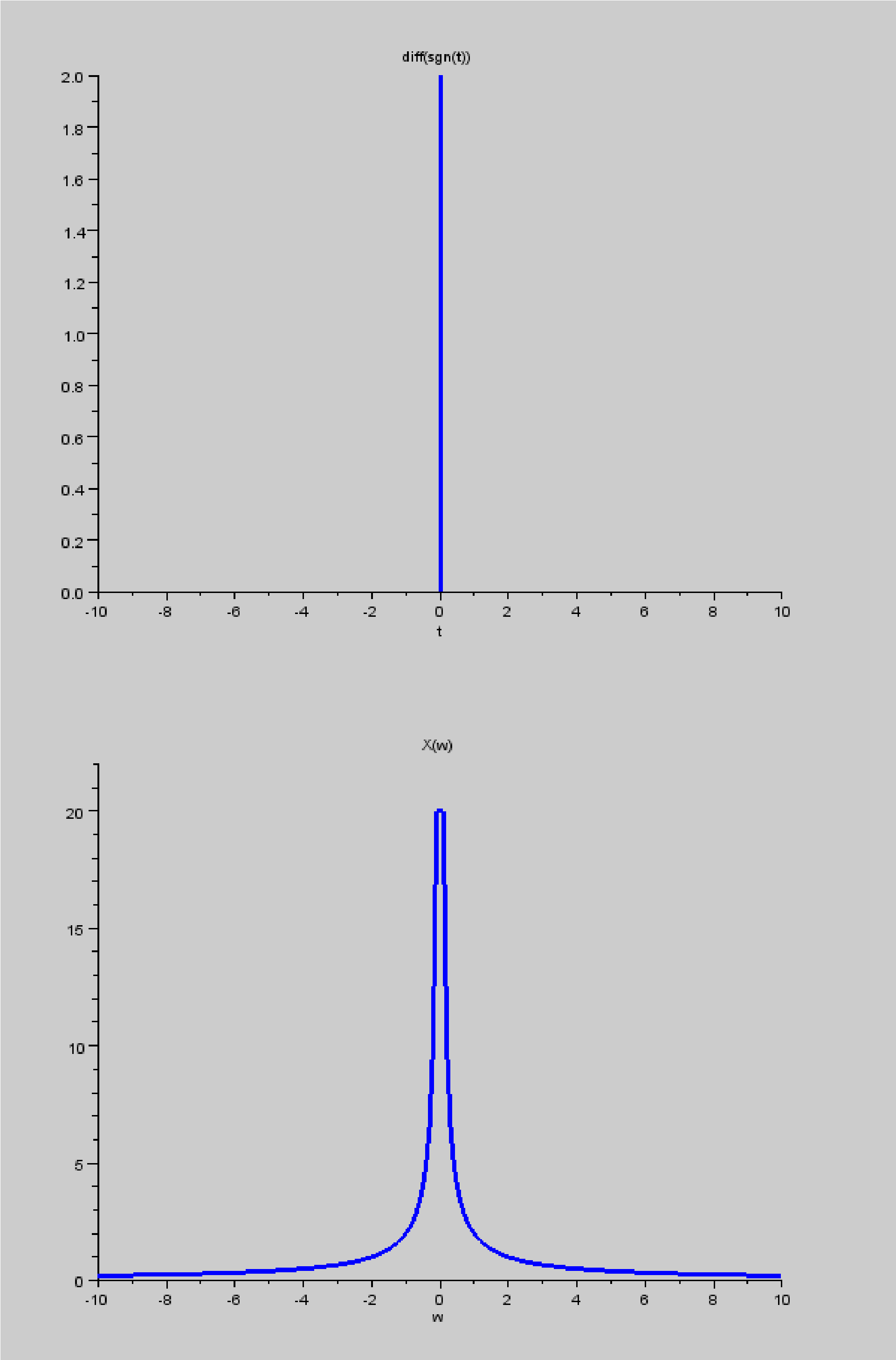


Figure 5.34: fourier transform of a signum function

```
   ];
a=gca();
plot(t,u);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('u(t)','t')
u1=1/2;//even part
figure
a=gca()
plot(t,u1);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('even part of u(t)','t')
h=find(t==0);
u2=[-ones(1,h) ones(1,length(t)-h)];//odd part
figure
a=gca()
plot(t,u2);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('odd part of u(t)','t')
//u(t)=u1(t)+u2(t)
//.: U[w]=U1[w]+U2[w] i.e U[w]=%pi*delta(w)+1/(%i*w)
w=-10:0.1:10;
for i=1:length(w)
    if w(i)==0 then
        delta(i)=1;
    else
        delta(i)=0;
    end
end
Uw=ones(1,length(w))./(%i*w)+%pi*delta';
figure
a=gca()
plot2d(w,abs(Uw));
```

```
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('U(w)','w')
```

Scilab code Exa 5.32 inverse fourier transform using convolution

```
clear;
clc;
close;
disp("X(w)=1/(a+jw)^2=1/(a+jw)*1/(a+jw) ");
disp("exp(-a*t)*u(t) <--> 1/(a+jw) ");
disp("therefore x(t)=convolution(exp(-a*t)*u(t),exp(-a*t)*u(t))");
a=2;
t=0:0.1:10;
y=exp(-a*t);
z=convol(y,y);
disp("t*exp(-a*t)*u(t) <--> 1/(a+jw)^2 ");
t1=[-t($:-1:2) t];
a=gca()
plot(t1,z);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
```

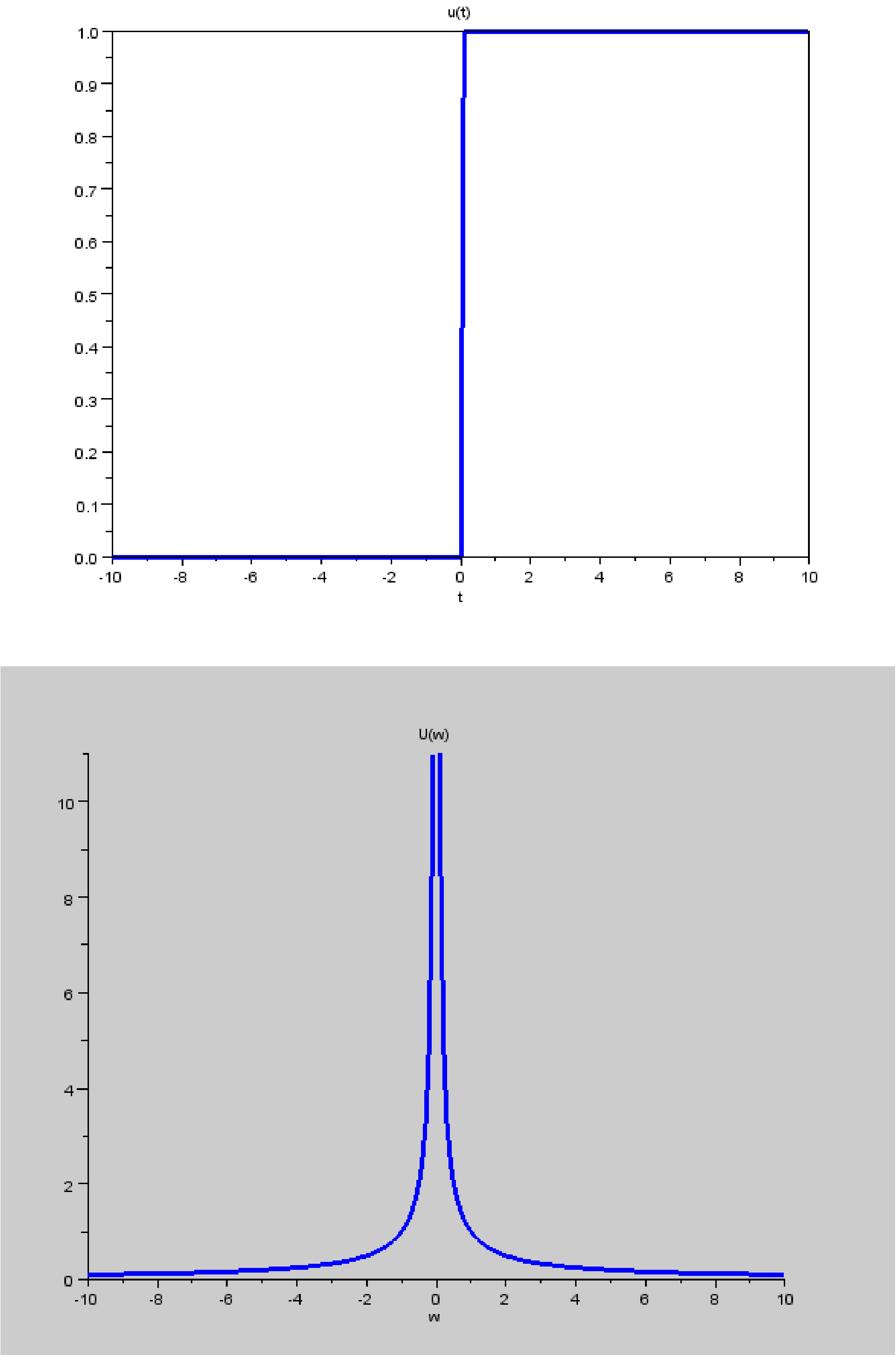


Figure 5.35: fourier transform of a step signal

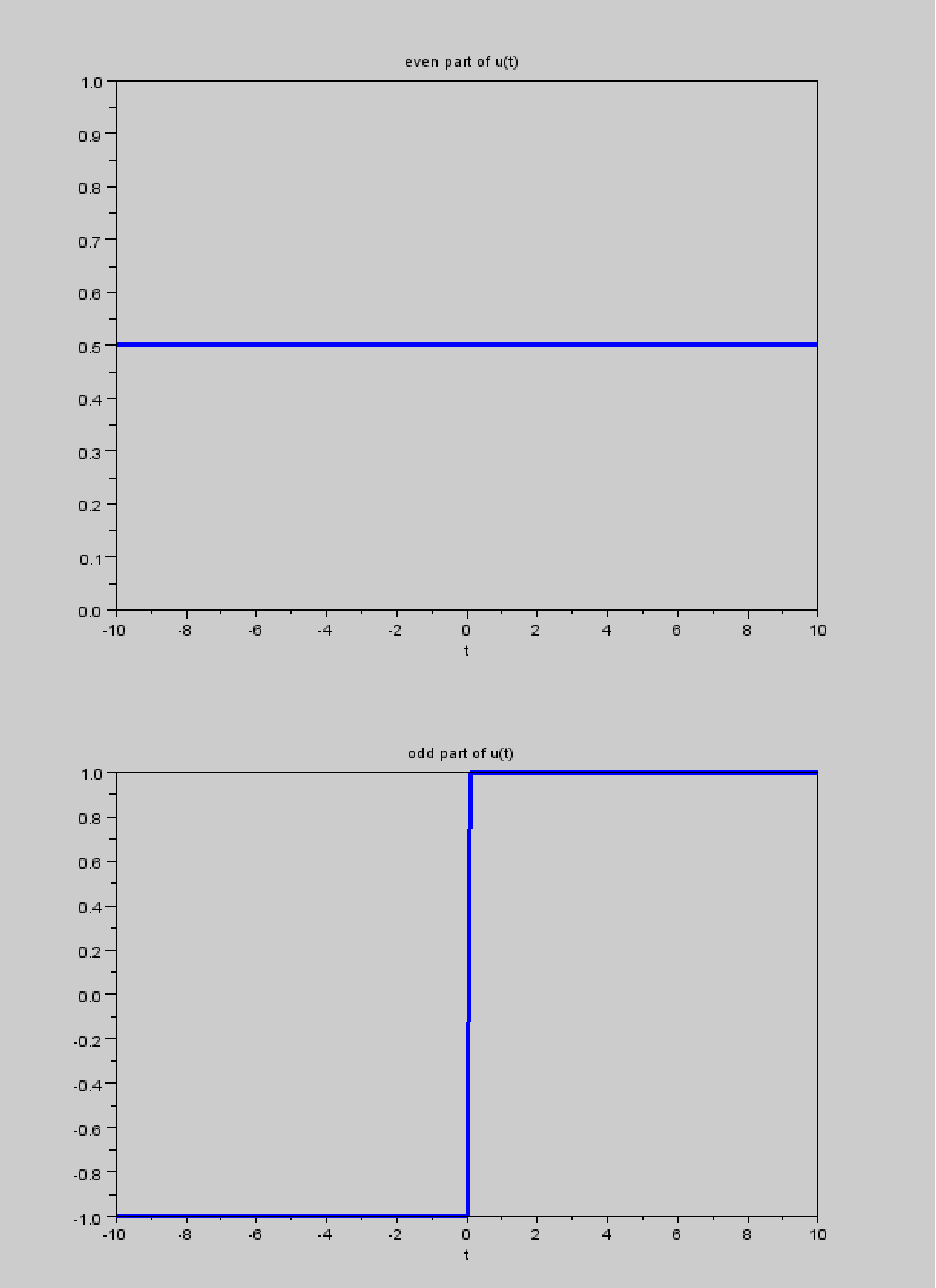


Figure 5.36: fourier transform of a step signal

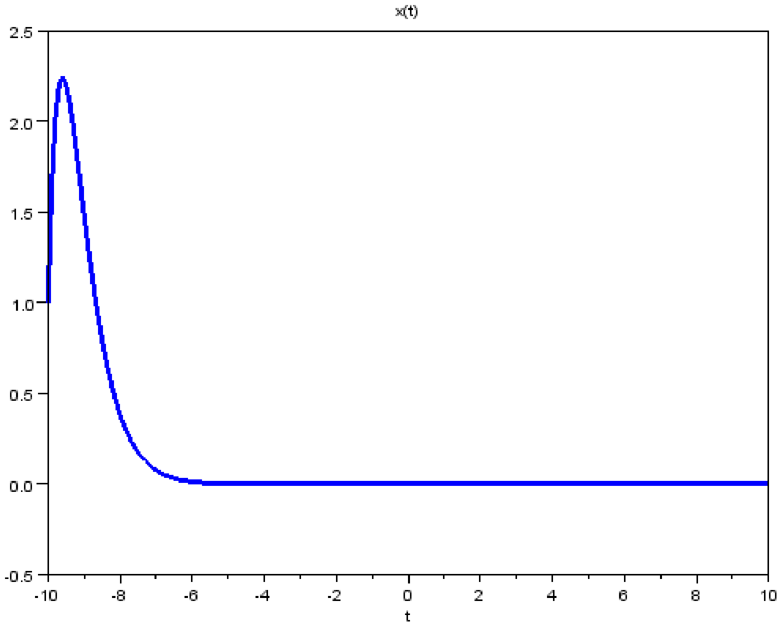


Figure 5.37: inverse fourier transform using convolution

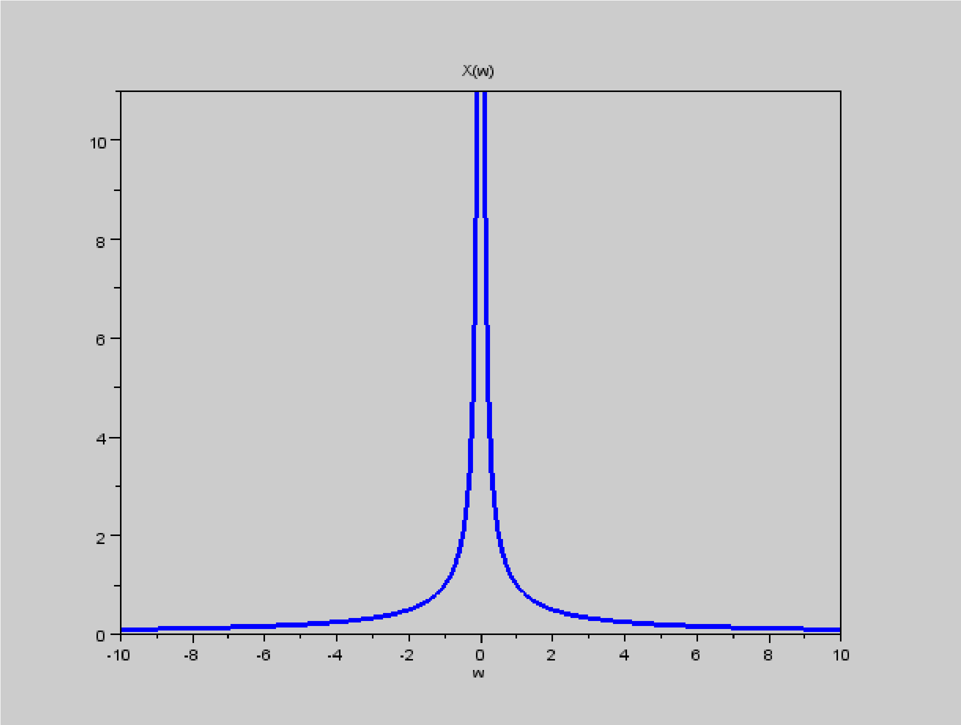


Figure 5.38: integration property

Scilab code Exa 5.34 integration property

```
clear;
clc;
close;
disp("u(t)=integral(delta(t))");
disp("integral(x(t)) <--> %pi*X[0]*delta(w)+X(w)/(%i*w)");
disp("delta(t) <--> 1");
w=-10:0.1:10;
for i=1:length(w)
    if w(i)==0 then
        delta(i)=1;
    else
        delta(i)=0;
    end
end
Xw=%pi*delta'+ones(1,length(w))./(%i*w);
disp( 'U[w]=%pi*delta(w)+1/(%i*w)');
figure
a=gca();
plot(w,abs(Xw));
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
```

Scilab code Exa 5.40.a fourier transform

```
clear;
```

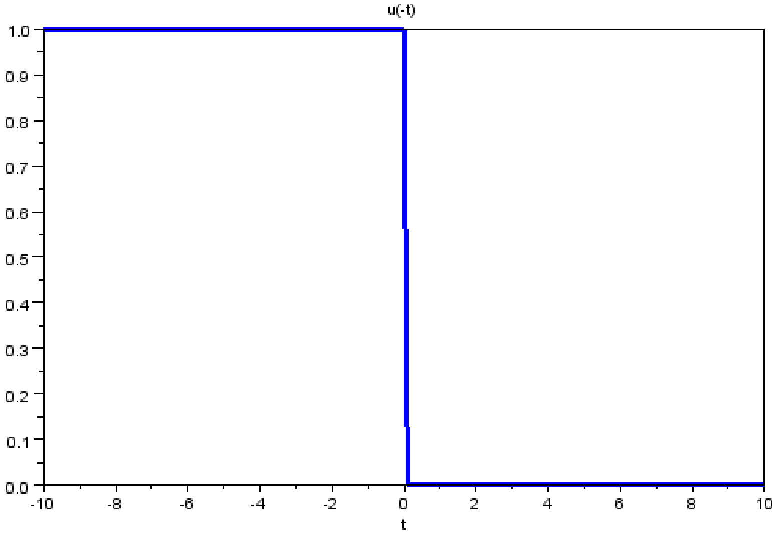


Figure 5.39: fourier transform

```
clc;
close;
disp('x(-t) <--> X(-w)=X*(w)');
t=-10:0.1:10;
u=[ones(1,find(t==0)) zeros(1,length(t)-find(t==0))];
a=gca();
plot(t,u);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('u(-t)','t')
w=-10:0.1:10;
for i=1:length(w)
    if w(i)==0 then
        delta(i)=1;
    else
        delta(i)=0;
    end
end
Xw=%pi*delta'-ones(1,length(w))./(%i*w);
disp( 'U[-w]=%pi*delta(w)-1/(%i*w)');
```

Scilab code Exa 5.40.b fourier transform

```
clear;
clc;
close;
disp('x(-t) <--> X(-w)=X*(w)');
a=0.1;
t=-10:0.1:10;
u=[ones(1,find(t==0)) zeros(1,length(t)-find(t==0))];
x=exp(-a*t).*u;
d=gca();
plot(t,x);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('e^(a*t)*u(-t)','t')
w=-10:0.1:10;
Xw=ones(1,length(w))./(a-(%i*w));
disp( 'exp(-a*t)*u(-t) <--> 1/(a-(%i*w))');
figure
d=gca();
plot(w,abs(Xw));
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('U(w)','w')
```

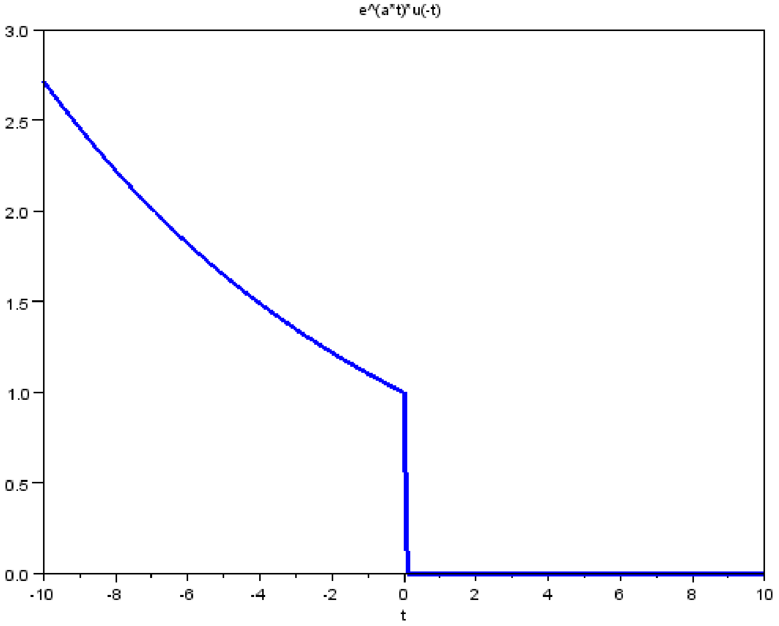


Figure 5.40: fourier transform

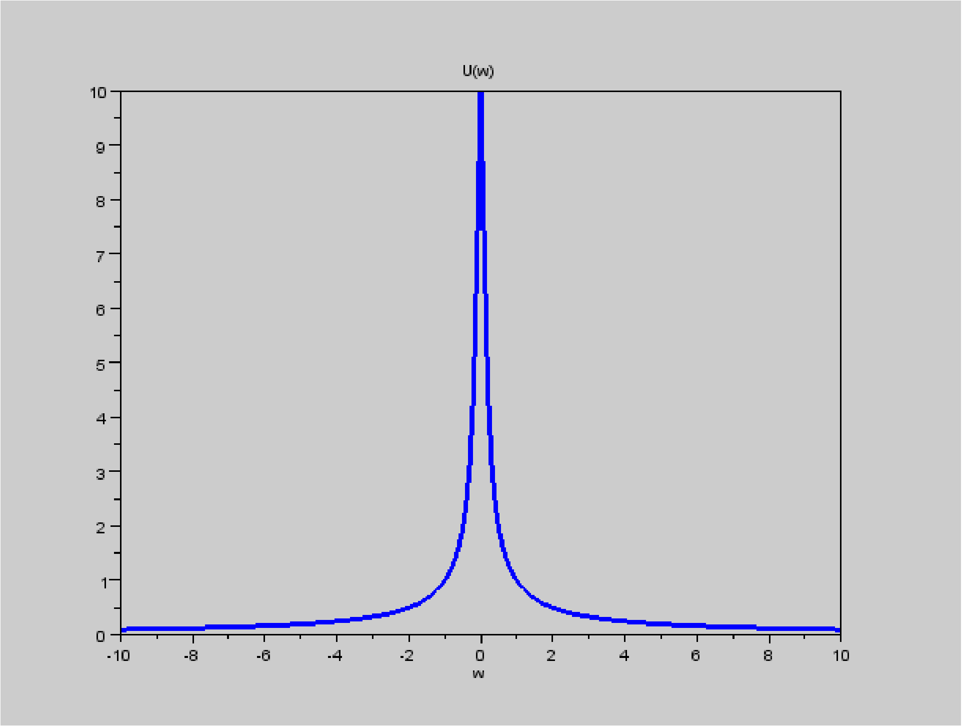


Figure 5.41: fourier transform

Scilab code Exa 5.42 fourier transform of a exponential signal

```
clear;
clc;
close;
a=2;
t=-10:0.1:10;
x=[exp(a*t(1:find(t==0))) exp(-a*t(find(t==0)+1:$))
    ];
d=gca()
```

```
plot(t,x);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
disp("1/2*exp(-a*|t|) is even part of 1/2*exp(-a*t)*u(t)");
disp("even(x(t)) <--> real(X(w))=a/(a^2+w^2)");
w=-10:0.01:10;
Xw=2*a*ones(1,length(w))./(a^2+w^2);
figure
d=gca()
plot(w,Xw);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
//not sure if it works properly
```

Scilab code Exa 5.43 fourier transform of a guassian pulse

```
clear;
clc;
close;
a=.5;
t=-10:0.1:10;
x=exp(-a*t.*t);
disp("guassian pulse signal x(t)=exp(-a*t^2)");
disp("X(w)=integral(exp(-a*t^2)*exp(-%i*w*t)) w.r.t dt");
disp("d(X(w))/dw=-%i*w/(2*a)*integral(exp(-a*t^2)*
```

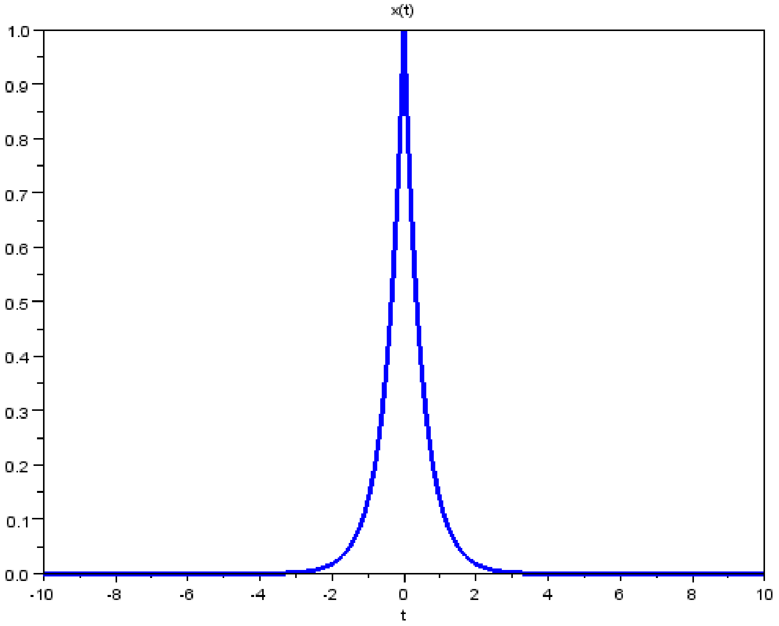


Figure 5.42: fourier transform of a exponential signal

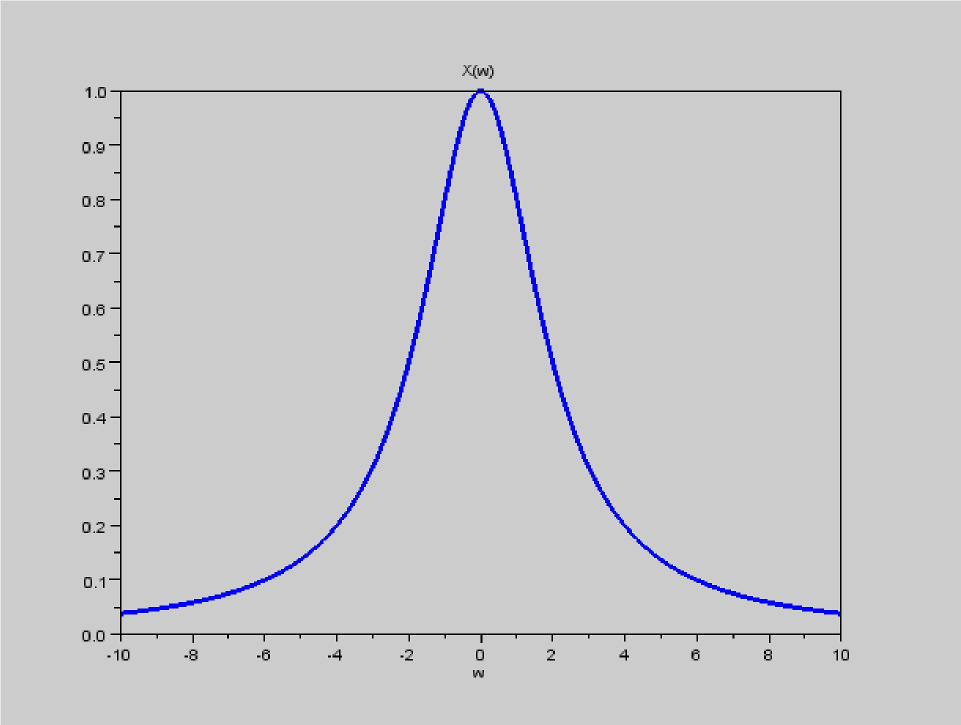


Figure 5.43: fourier transform of a exponential signal

```
exp(-%i*w*t))");
disp("d(X(w))/dw=-w*X(w)/2a");
disp("solving this we get X(w)=A*exp(-w^2/4a)")
disp("A=sqrt(%pi/a)");
d=gca()
plot(t,x);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
A=sqrt(%pi/a);
w=t;
Xw=A*exp(-w.*w/(4*a));
figure
d=gca()
plot(w,Xw);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
```

Scilab code Exa 5.44 impulse response using fourier transform

```
clear;
clc;
close;
disp("system given is dy(t)/dt+2y(t)=x(t)+dx(t)/dt");
disp("taking fourier transform on both sides we get");
disp("H(w)=Y(w)/X(w)=1-(1/(2+%j*w))");
```

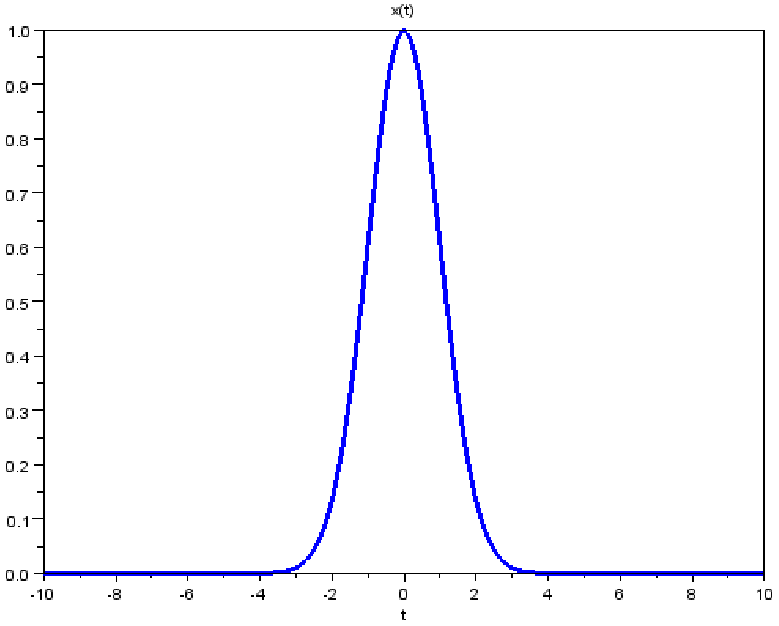


Figure 5.44: fourier transform of a guassian pulse

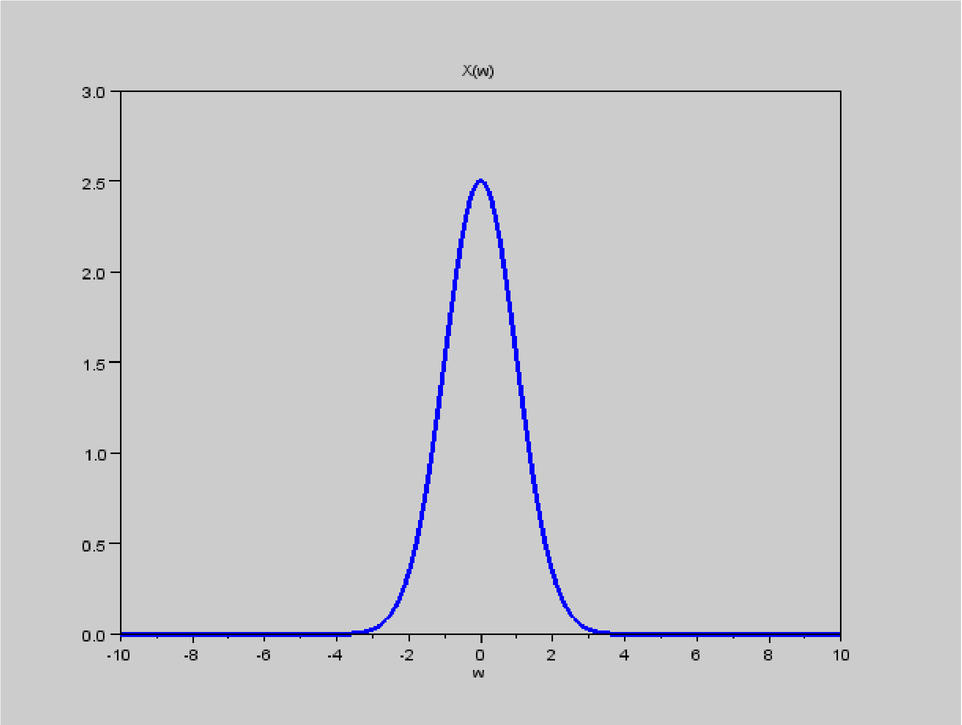


Figure 5.45: fourier transform of a guassian pulse

```
w=-10:0.1:10;
dw=.1;
Hw=1-ones(1,length(w))./(2+%i*w);
t=0:0.1:10;
d=gca()
plot(w,Hw);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
for i=1:length(t)
    if t(i)==0 then
        delta(i)=1;
    else
        delta(i)=0;
    end
end
h=delta'-exp(-2*t);
figure;
d=gca()
plot(t,(h));
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('h(t)','t')
```

Scilab code Exa 5.45.a output response using fourier transform

```
clear;
clc;
close;
```

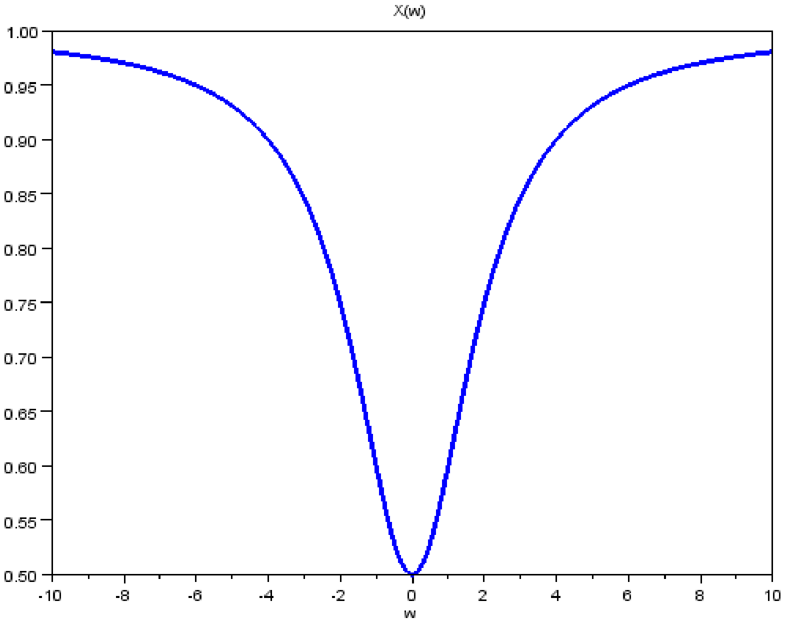


Figure 5.46: impulse response using fourier transform

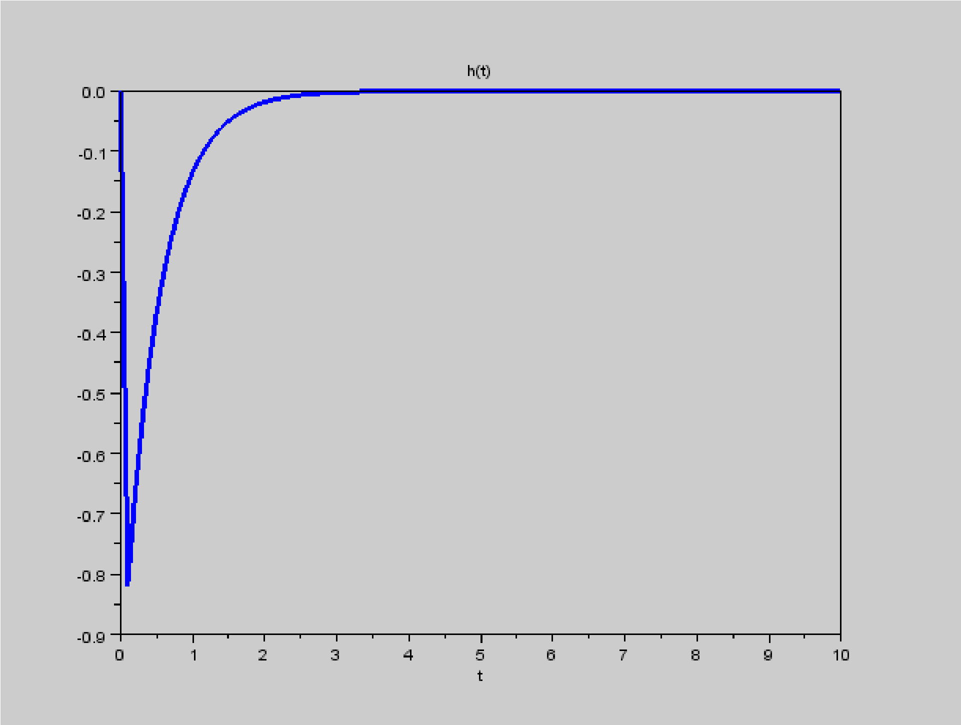


Figure 5.47: impulse response using fourier transform

```
disp(”dy(t)/dt+2y(t)=x(t)”);
w=0:0.1:10;
t=w;
dw=.1;
Xw=ones(1,length(w))./(1+%i*w);
Hw=ones(1,length(w))./(2+%i*w);
Yw=Xw.*Hw;
y=Yw*exp(%i*t'*w)*dw*.31;
d=gca()
plot(t,y);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('y(t)','t')
yy=exp(-t)-exp(-2*t);
disp(”y(t)=exp(−t)−exp(−2*t)”)
figure
d=gca()
plot(t,yy);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('y(t)','t')
```

Scilab code Exa 5.45.b output response using fourier transform

```
clear;
clc;
close;
disp(”dy(t)/dt+2y(t)=x(t)”);
w=0.1:0.1:10;
```

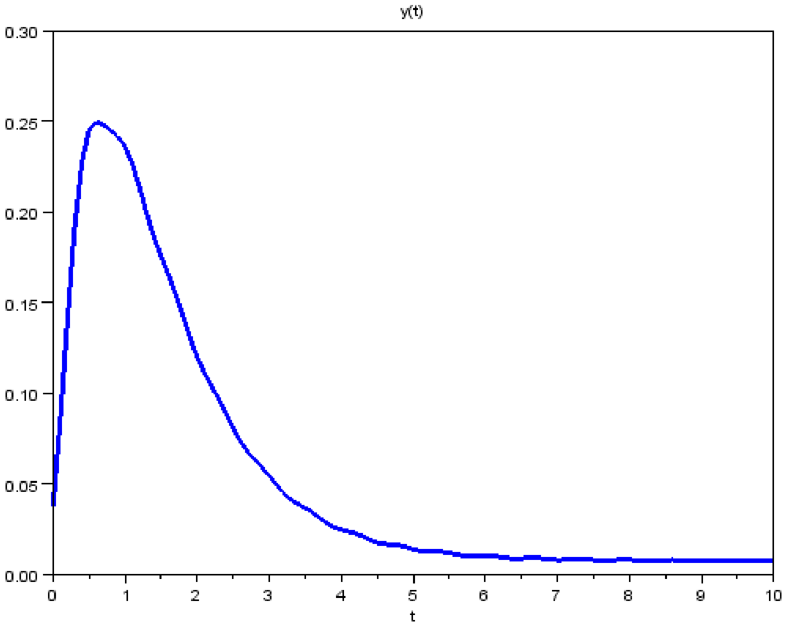


Figure 5.48: output response using fourier transform

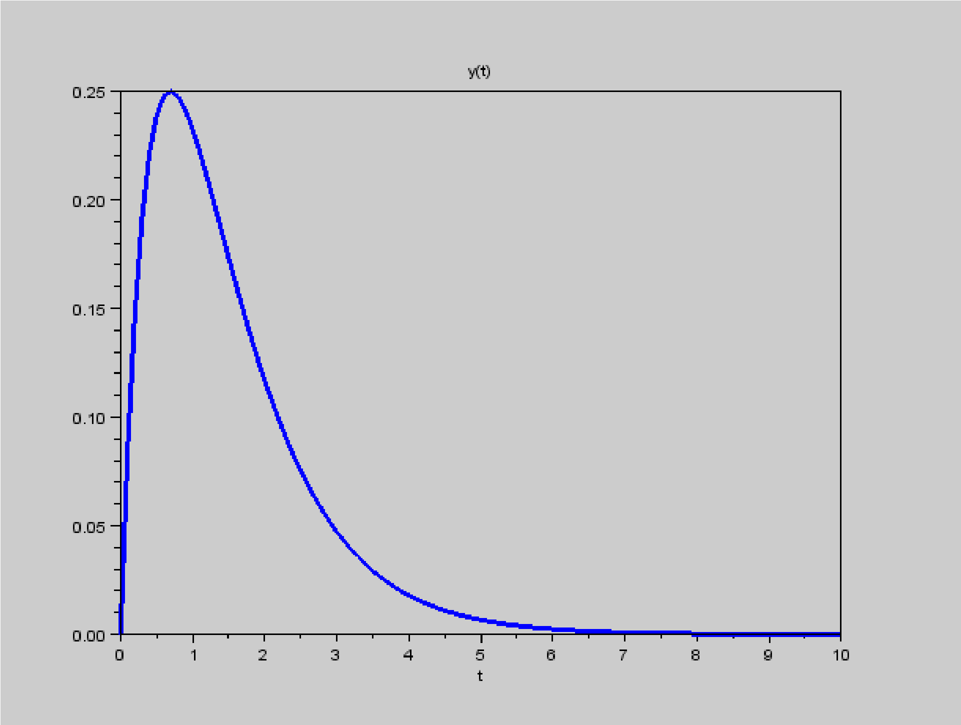


Figure 5.49: output response using fourier transform

```
t=w;
dw=.1;
Xw=ones(1,length(w))./(%i*w);
Hw=ones(1,length(w))./(2+%i*w);
Yw=Xw.*Hw;
y=Yw*exp(%i*t'*w)*dw;
d=gca()
plot(t,y);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('y(t)','t')
yy=0.5*(1-exp(-2*t));
disp("y(t)=0.5(1-exp(-2*t))")
figure
d=gca()
plot(t,yy);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('y(t)','t')
```

Scilab code Exa 5.46 harmonics in the output response

```
clear;
clc;
close;
T0=1;
t=-5.99:0.01:6;
```

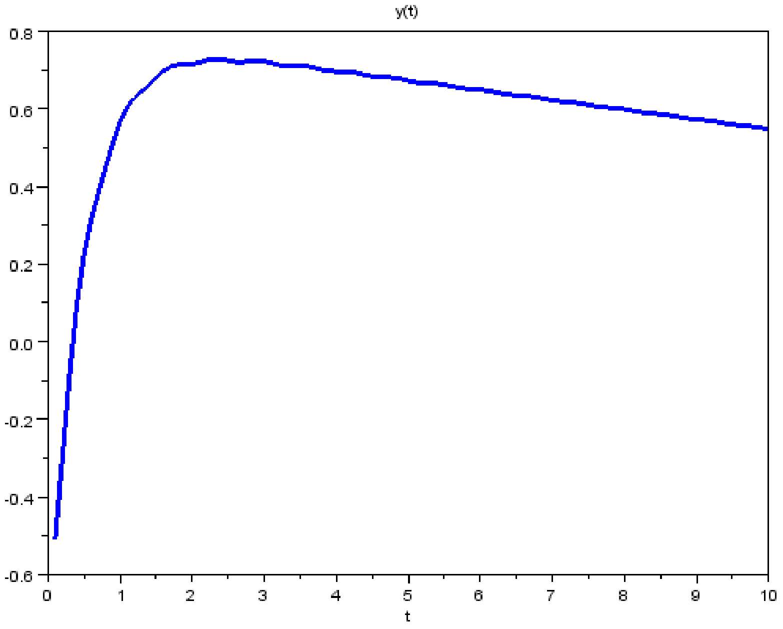


Figure 5.50: output response using fourier transform

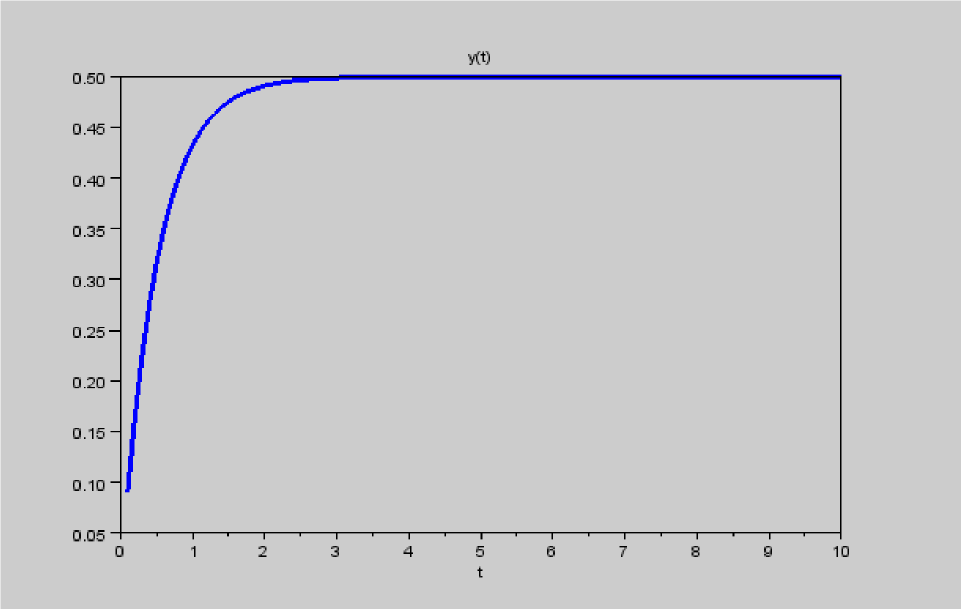


Figure 5.51: output response using fourier transform

```
t_temp=0.01:0.01:T0;
s=length(t)/length(t_temp);
x=[];
for i=1:s
    if modulo(i,2)==1  then
        x=[x zeros(1,length(t_temp))];
    else
        x=[x 10*ones(1,length(t_temp))];
    end
end
d=gca()
plot(t,x,'r')
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
disp("y(t)=sum(H(kw0)*exp(%j*k*w0*t))");
//fourier series of x(t)
```

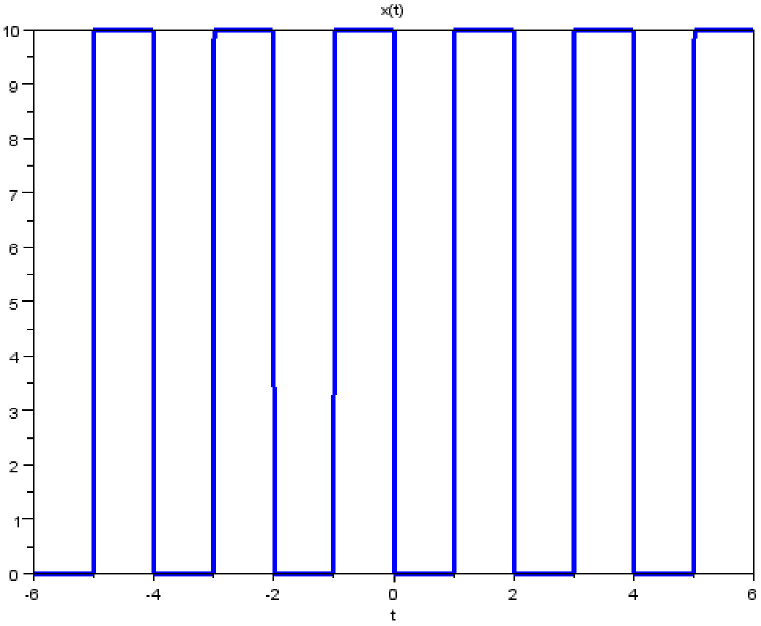


Figure 5.52: harmonics in the output response

```
w0=%pi;
for k=-10:10
    cc(k+11,:)=exp(-%i*k*w0*t);
    ck(k+11)=x*cc(k+11,:)'/length(t);
    if abs(ck(k+11))<0.01 then
    ck(k+11)=0;
    else if real(ck(k+11))<0.1  then
    ck(k+11)=%i*imag(ck(k+11));
    end
    end
    if k==1 then
        c1=ck(k+11);
    end
    if k==3 then
        c3=ck(k+11);
    end
end
yc1=2*abs(c1/(2+%i*w0));
yc3=2*abs(c3/(2+%i*w0*3));
disp(yc1,”first harmonic is”);
disp(yc3,”third harmonic is”);
```

Scilab code Exa 5.47.a bode plot

```
clear;
clc;
fmin=0.1;
fmax=100;
f=fmin:0.1:fmax;
Hdb=20*log(abs(1+%i*f/10));
figure
a=gca();
a.data_bounds=[fmin,-20;fmax,40];
```

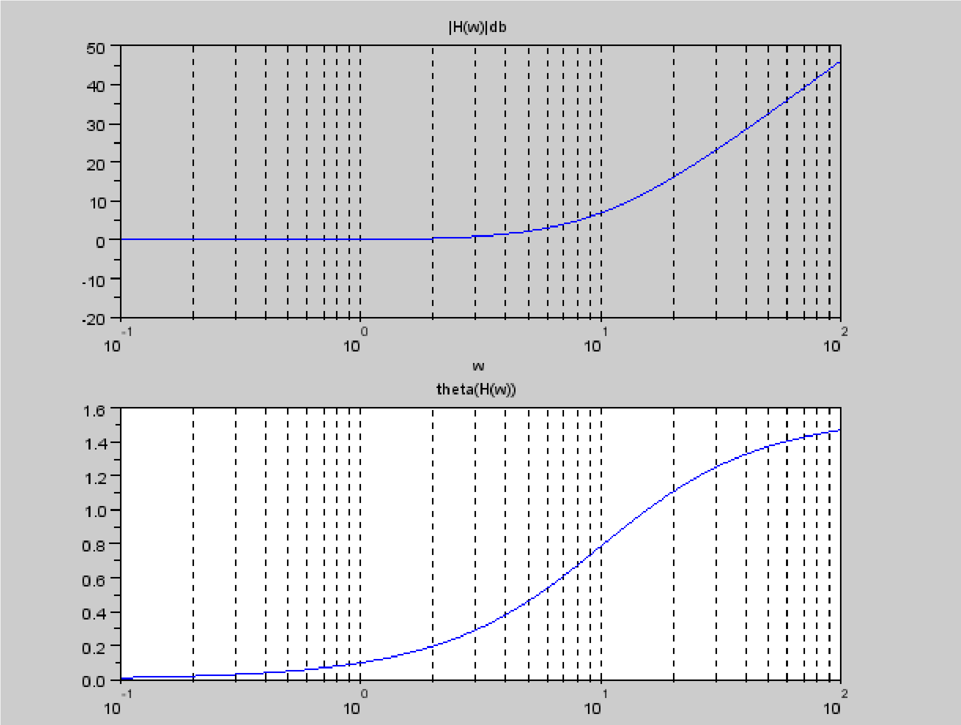


Figure 5.53: bode plot

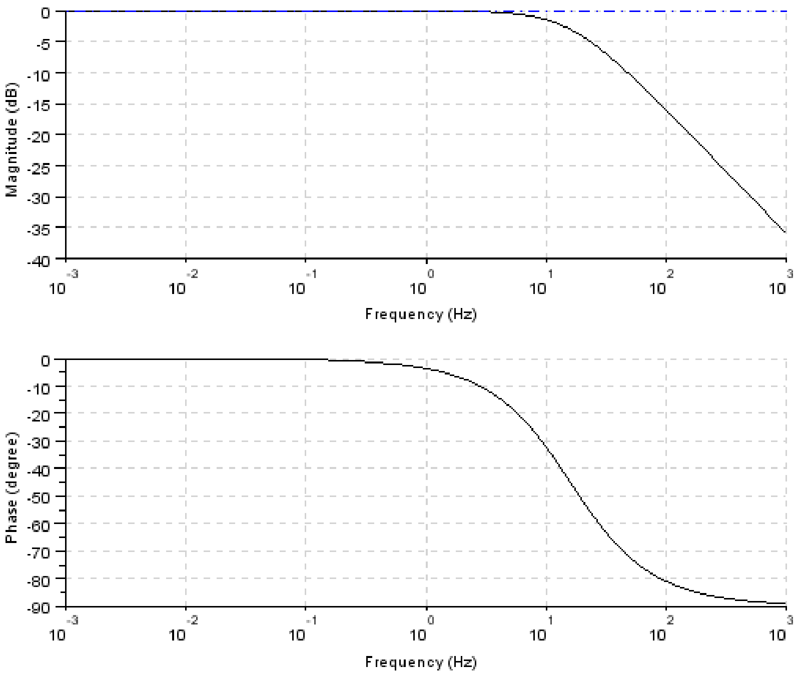


Figure 5.54: bode plot

```
a.log_flags=”lnn”;
subplot(2,1,1)
plot(f,Hdb);
xtitle(’|H(w)|db’,’w’)
subplot(2,1,2)
plot(f,atan(f/10));
xtitle(’theta(H(w))’,’w’)
a=gca();
a.data_bounds=[fmin,0;fmax,%pi/2];
a.log_flags=”lnn”;
```

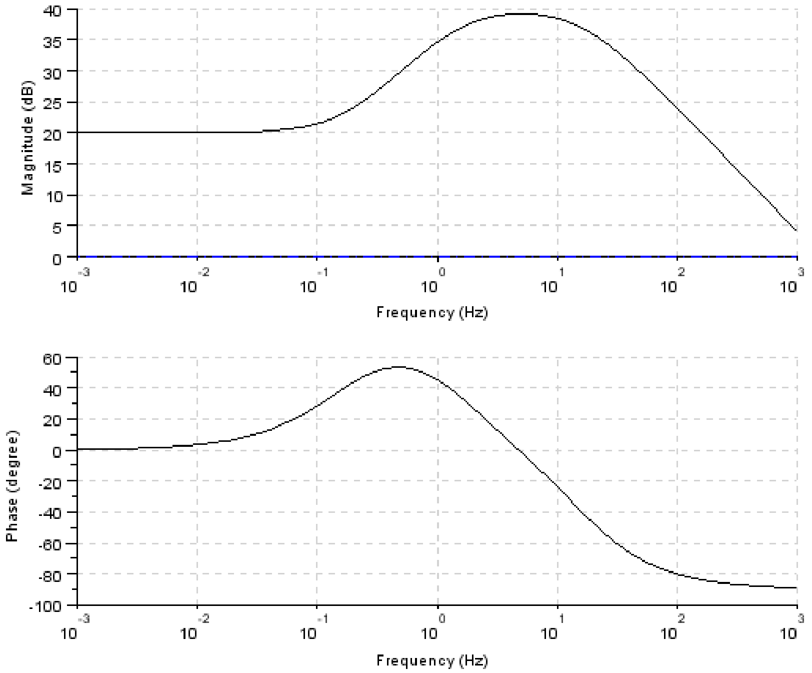


Figure 5.55: bode plot

Scilab code Exa 5.47.b bode plot

```
clear;
clc;
s=poly(0,'s');
H=syslin('c',[100/(100+s)]) ;
fmin=0.1;
fmax=100;
scf(1);clf;
bode(H,fmin,fmax);
show_margins(H) ;
```

Scilab code Exa 5.47.c bode plot

```
clear;
clc;
s=poly(0,'s');
H=syslin('c',[(10^4+s*10^4)/((100+s)*(10+s))]) ;
a=pfss(H);
fmin=0.1;
fmax=1000;
scf(1);clf;
bode(a(1),fmin,fmax);
show_margins(H) ;
```

Scilab code Exa 5.48 impulse response of a phase shifter

```
clear;
clc;
close;
w=-10:0.1:10;
Hw=[exp(%i*%pi/2)*ones(1,find(w==0)) exp(-%i*%pi/2)*
   ones(1,find(w==0)-1)];
d=gca();
plot(w,imag(Hw));
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('H(w)','w')
disp(”H(w)=-%i*sgn(w)”);
disp(”we know sgn(t) <--> 2/(j*w)”)
disp(”by duality property 2/(j*t) <-->2*%pi*sgn(-w)
   =-2*%pi*sgn(w) ”)
disp(”therefore 1/(%pi*t) <--> -j*sgn(w)”);
t=0.1:0.1:10;
h=ones(1,length(t))./(%pi*t);
figure
```

```
d=gca()
plot(t,h);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('h(t)','t')
w0=2;
x=cos(w0*t);
figure
d=gca();
plot(t,x);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
y=convol(x,h);
figure
d=gca()
plot(t,y(1:length(t)));
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('y(t)','t')
```

Scilab code Exa 5.52 output of a ideal LPF

```
clear;
clc;
close;
w=-10:0.1:10;
wc=2;
```

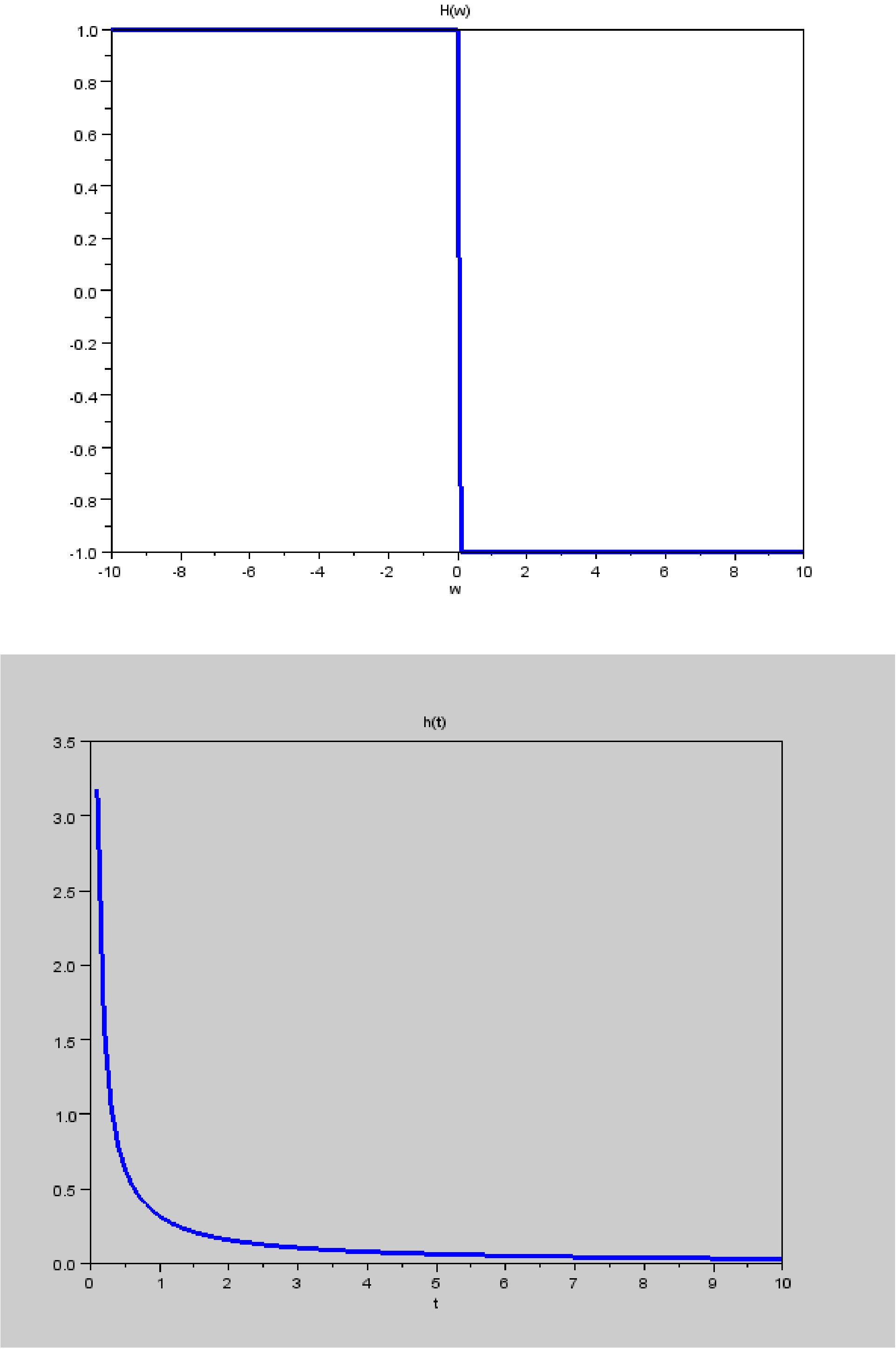


Figure 5.56: impulse response of a phase shifter

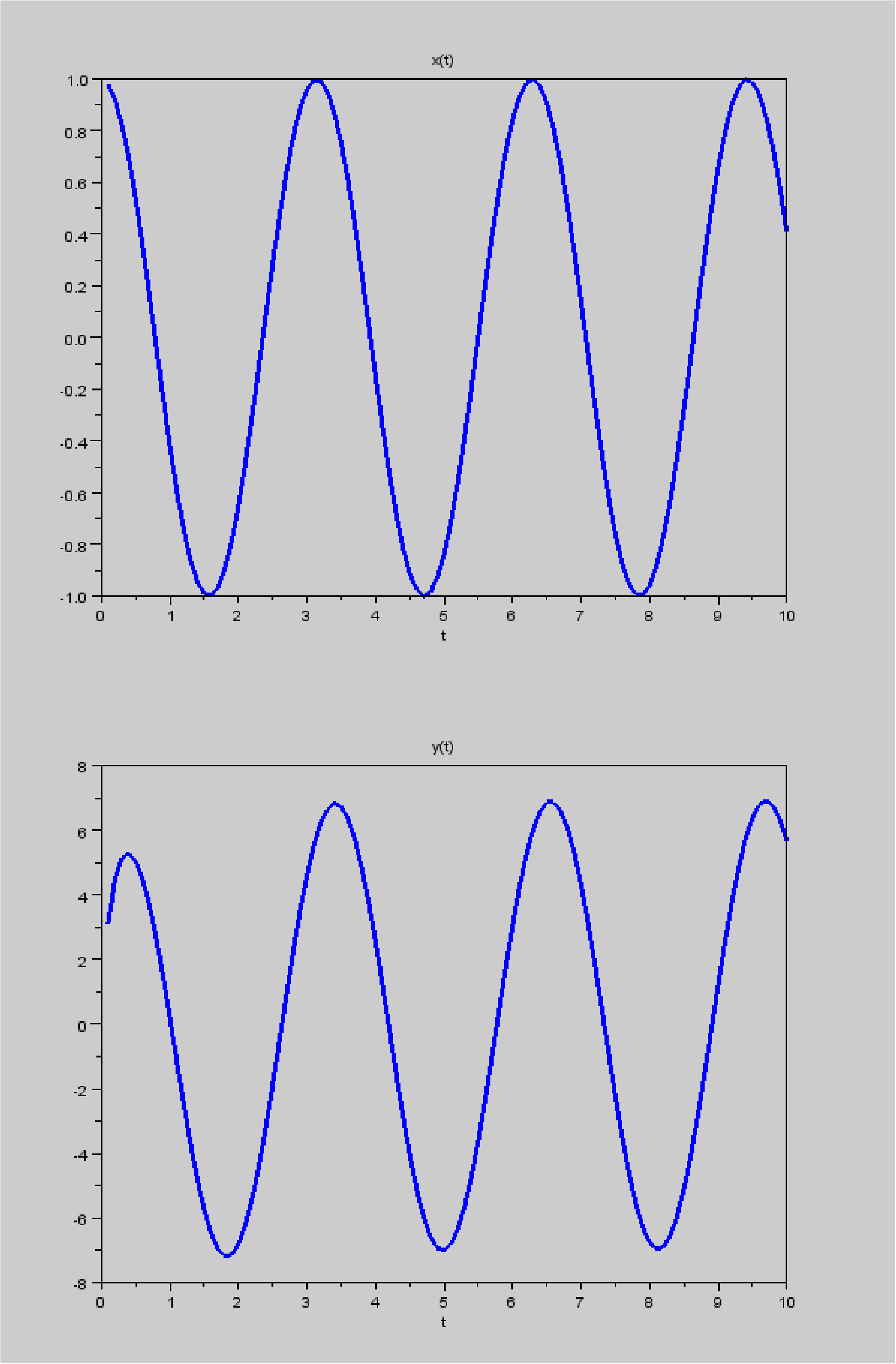


Figure 5.57: impulse response of a phase shifter

```
for i=1:length(w)
    if w(i)>-wc & w(i)<wc then
        Hw(i)=1;
    else
        Hw(i)=0;
    end
end
d=gca()
plot(w,Hw);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('H(w)','w')
disp("x(t)=sin(a*t)/(%pi*t)");
disp("X(w)= 1 for |w|<a ");
disp("      0 elsewhere");
a=1;
for i=1:length(w)
    if w(i)>-a & w(i)<a then
        Xw(i)=1;
    else
        Xw(i)=0;
    end
end
Yw=Xw.*Hw;
figure
d=gca()
plot(w,Yw);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('Y(w)','w')
disp("therefore y(t)=x(t) for a<wc");
disp("and y(t)=h(t) for a>wc");
disp("thus the output suffers distortion when a>wc")
   ;
```

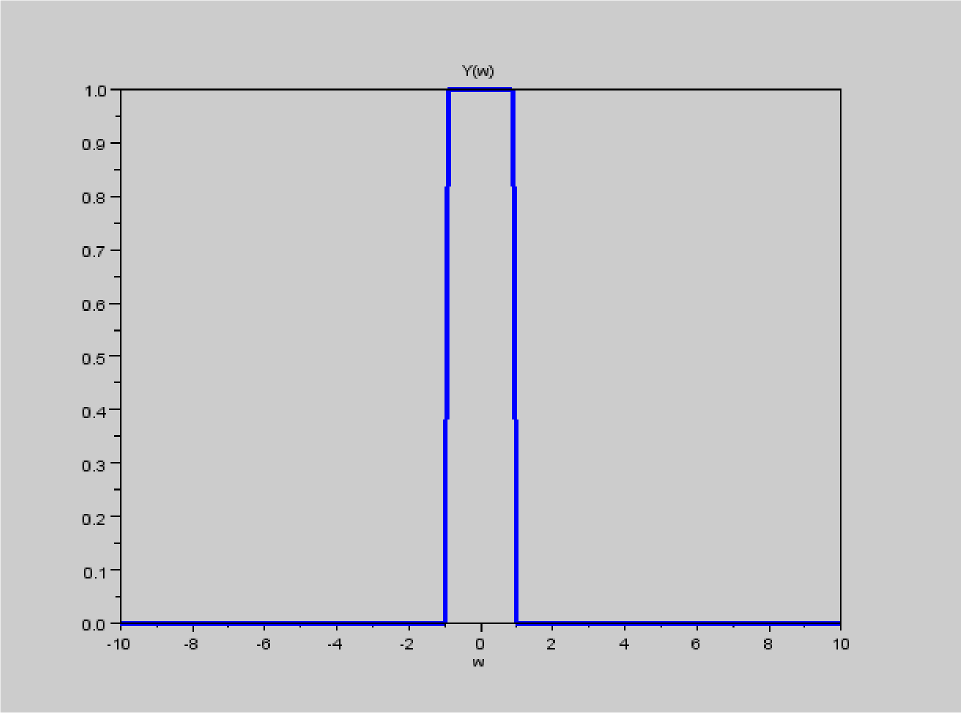


Figure 5.58: output of a ideal LPF

Scilab code Exa 5.53 output of a ideal LPF

```
clear;
clc;
close;
w=-20:0.1:20;
wc=4*%pi;
for i=1:length(w)
```

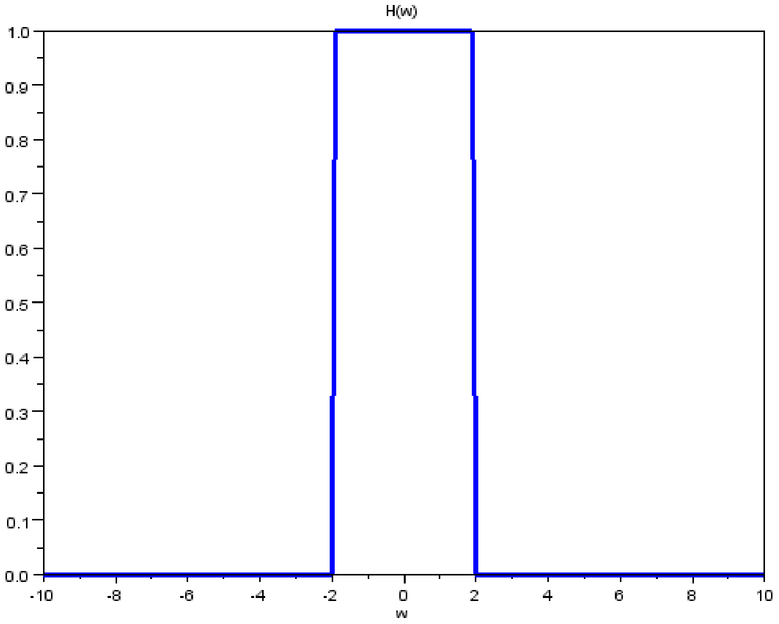


Figure 5.59: output of a ideal LPF

```
    if w(i)>-wc & w(i)<wc then
        Hw(i)=1;
    else
        Hw(i)=0;
    end
end
a=gca();
plot(w,Hw);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('H(w)','w')
T0=1;
t=-5.99:0.01:6;
t_temp=0.01:0.01:T0;
s=length(t)/length(t_temp);
x=[];
for i=1:s
    if modulo(i,2)==1  then
        x=[x zeros(1,length(t_temp))];
    else
        x=[x 10*ones(1,length(t_temp))];
    end
end
figure
a=gca();
plot(t,x,'r');
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
//fourier series of x(t)
w0=%pi;
for k=-10:10
    cc(k+11,:)=exp(-%i*k*w0*t);
    ck(k+11)=x*cc(k+11,:)'/length(t);
    if abs(ck(k+11))<0.01 then
    ck(k+11)=0;
    else if real(ck(k+11))<0.1  then
    ck(k+11)=%i*imag(ck(k+11));
    end
    end
    if k==1 then
        c1=ck(k+11);
    end
    if k==3 then
        c3=ck(k+11);
    end
end
yc1=2*abs(c1/(2+%i*w0));
yc3=2*abs(c3/(2+%i*w0*3));
disp("since frequencies above 4*%pi are cut off only first and third harmonics exists in the output");
y=5+yc1*sin(%pi*t)+yc3*sin(3*%pi*t);
figure
a=gca();
plot(t,y);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('y(t)','t')
disp("y=5+(20/%pi)*sin(%pi*t)+20/(%pi*3)*sin(3*%pi*t)");
```

Scilab code Exa 5.54 ideal low pass filter

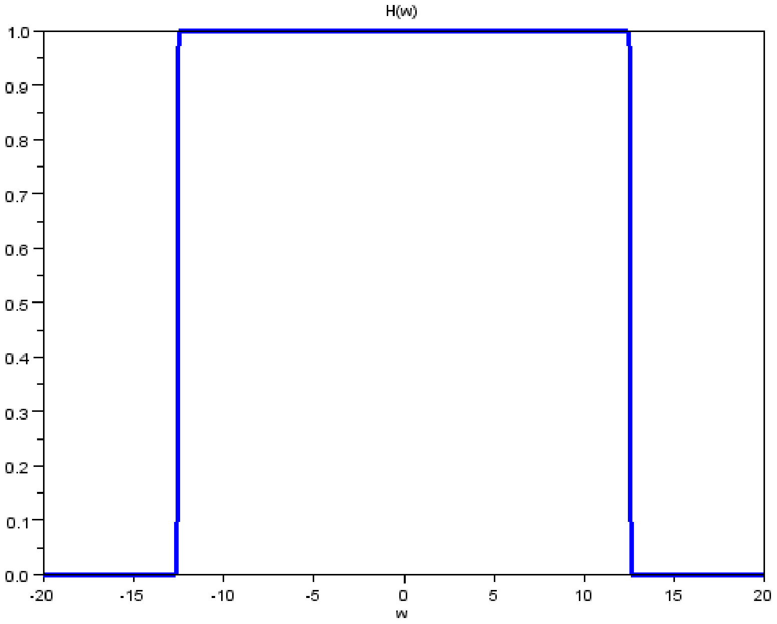


Figure 5.60: output of a ideal LPF

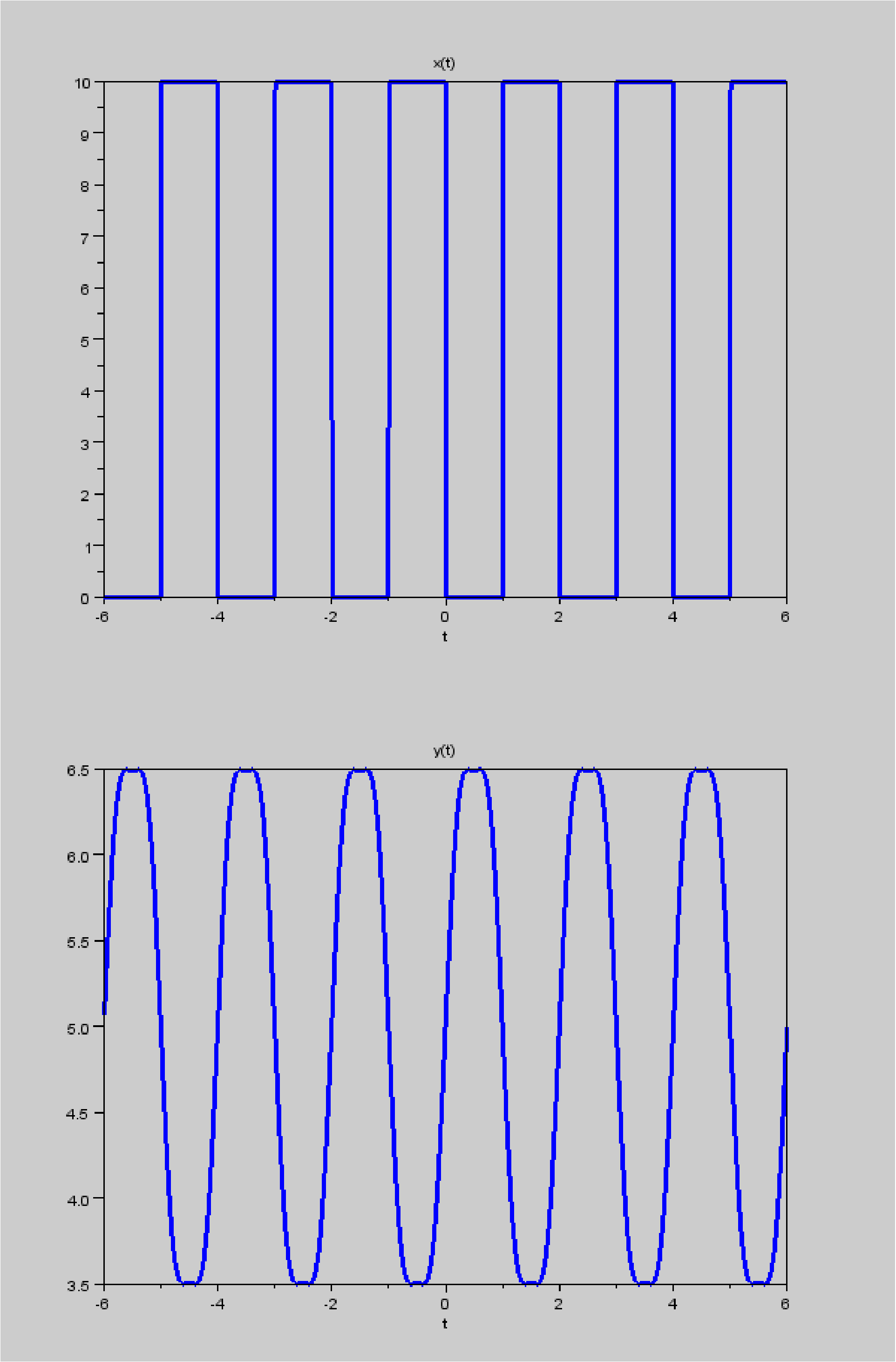


Figure 5.61: output of a ideal LPF

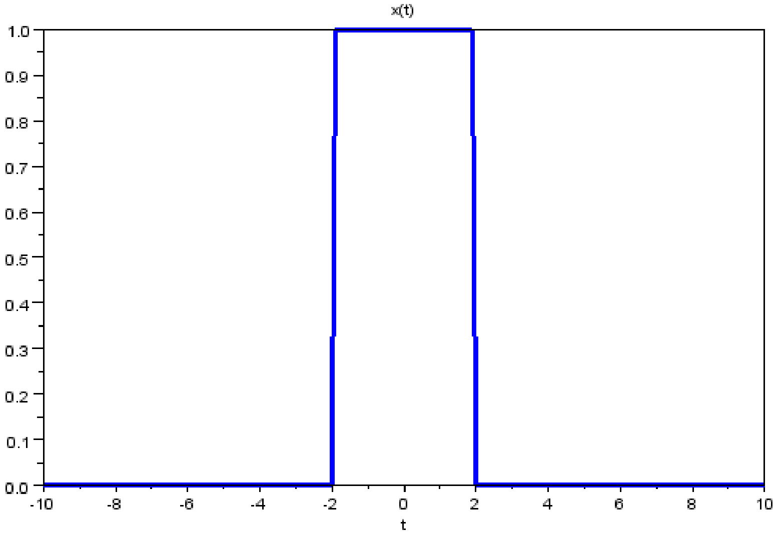


Figure 5.62: ideal low pass filter

```
syms t w0 W
w=-10:0.1:10;
wc=2;
for i=1:length(w)
    if w(i)>-wc & w(i)<wc then
        Hw(i)=1;
    else
        Hw(i)=0;
    end
end
a=gca();
plot(w,Hw);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle(’x(t)’,’t’)
disp(”we know y(t)=x(t) for w<wc”);
Xw=ones(1,length(w))./(2+%i*w);
```

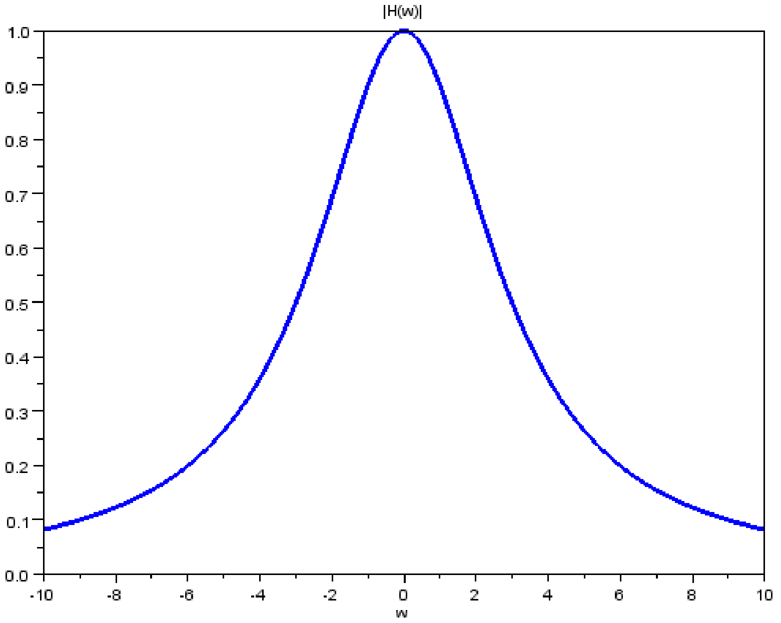


Figure 5.63: equivalent bandwidth

```
ex=integ(exp(-4*t),t,0,%inf);
disp(”energy in x(t) is 1/4”);
ey=(1/(2*%pi))*integ(1/(4+W^2),W,-wc,wc);
ratio=ey/ex;
disp(”the ratio is two therefore wc=2 rad/s ”);
```

Scilab code Exa 5.55 equivalent bandwidth

```
clear;
clc;
close;
```

```
w=-10:0.1:10;
w0=3;//w0=1/(rc)
for i=1:length(w)
    Hw(i)=1/(1+%i*(w(i)/w0));
    if abs(Hw(i))==1 then
        wmax=w(i);
        Hmax=Hw(i)
    end
    if abs(Hw(i))==abs(1/(1+%i)) then
        w3db=w(i);
    end
end
disp("rad/s",w3db,"the 3-db bandwidth is w=");
weq=(1/Hmax^2)*(%pi)/(2*w0);//integral(|H(w)|^2)=%pi*rc/2;
disp("rad/s",weq,"the equiva;ent bandwidth is weq=");
a=gca();
plot(w,abs(Hw)^2);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('|H(w)|','w')
```

Scilab code Exa 5.58 fourier spectrum

```
w1 = -1:0.1:1;
w2 = -10:0.1:10;
X1 = ones(1,length(w1));
X2 = zeros(1,length(w2));
X = [X2(1:find(w2==-1)-2),0.5,X1,0.5,X2(find(w2==1)+2:$)];
a=gca();
subplot(2,1,1)
plot(w2,X);
```

```
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(w)','w')
tmax=10;
dw=0.1;
t=-tmax:0.1:tmax;
x=X*exp(%i*(t'*w2))*dw;
subplot(2,1,2)
a=gca();
plot(t,x);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
wM=1.5;
//Ts<%pi/wM
Ts=1;
for i=1:length(t)
    if modulo(t(i),Ts)==0 then
        deltaT(i)=1;
    else
        deltaT(i)=0;
    end
end
figure
subplot(2,1,1)
a=gca();
plot2d3(t,deltaT);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('detlaT(t) Ts<%pi/wM','t')
ws=floor(2*%pi/Ts);
for i=1:length(w2)
    if modulo(w2(i),ws)==0 then
        deltaW(i)=1;
    else
```

```
        deltaW(i)=0;
    end
end
subplot(2,1,2)
a=gca();
plot2d3(w2,deltaW);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('deltaW(w)','w')
xs=x.*deltaT';
figure
subplot(2,1,1)
a=gca();
plot2d3(t,xs)
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('xs(t)','t')
Xsw=convol(X,deltaW');
subplot(2,1,2)
a=gca();
plot2d(w2,Xsw(101:301));
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
//Ts>%Pi/wM
Ts=3.5;
for i=1:length(t)
    if modulo(t(i),Ts)==0 then
        deltaT(i)=1;
    else
        deltaT(i)=0;
    end
end
figure
subplot(2,1,1)
a=gca();
```

```
plot2d3(t,deltaT);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('deltaT(t) Ts<%pi/wM','t')
ws=floor(2*%pi/Ts);
for i=1:length(w2)
    if modulo(w2(i),ws)==0 then
        deltaW(i)=1;
    else
        deltaW(i)=0;
    end
end
subplot(2,1,2)
a=gca();
plot2d3(w2,deltaW);
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('deltaW(w)','w')
xs=x.*deltaT';
figure
subplot(2,1,1)
a=gca();
plot2d3(t,xs)
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('xs(t)','t')
subplot(2,1,2)
d=gca();
for i=1:length(w2)
    if modulo(w2(i),ws)==0 then
        plot(w2+w2(i),X)
    end
end
poly1=d.children.children;
poly1.thickness=1;
```

```
d.data_bounds=[-5 0;5 2];

poly1.foreground=2;
xtitle('Xs(w)','w')
```

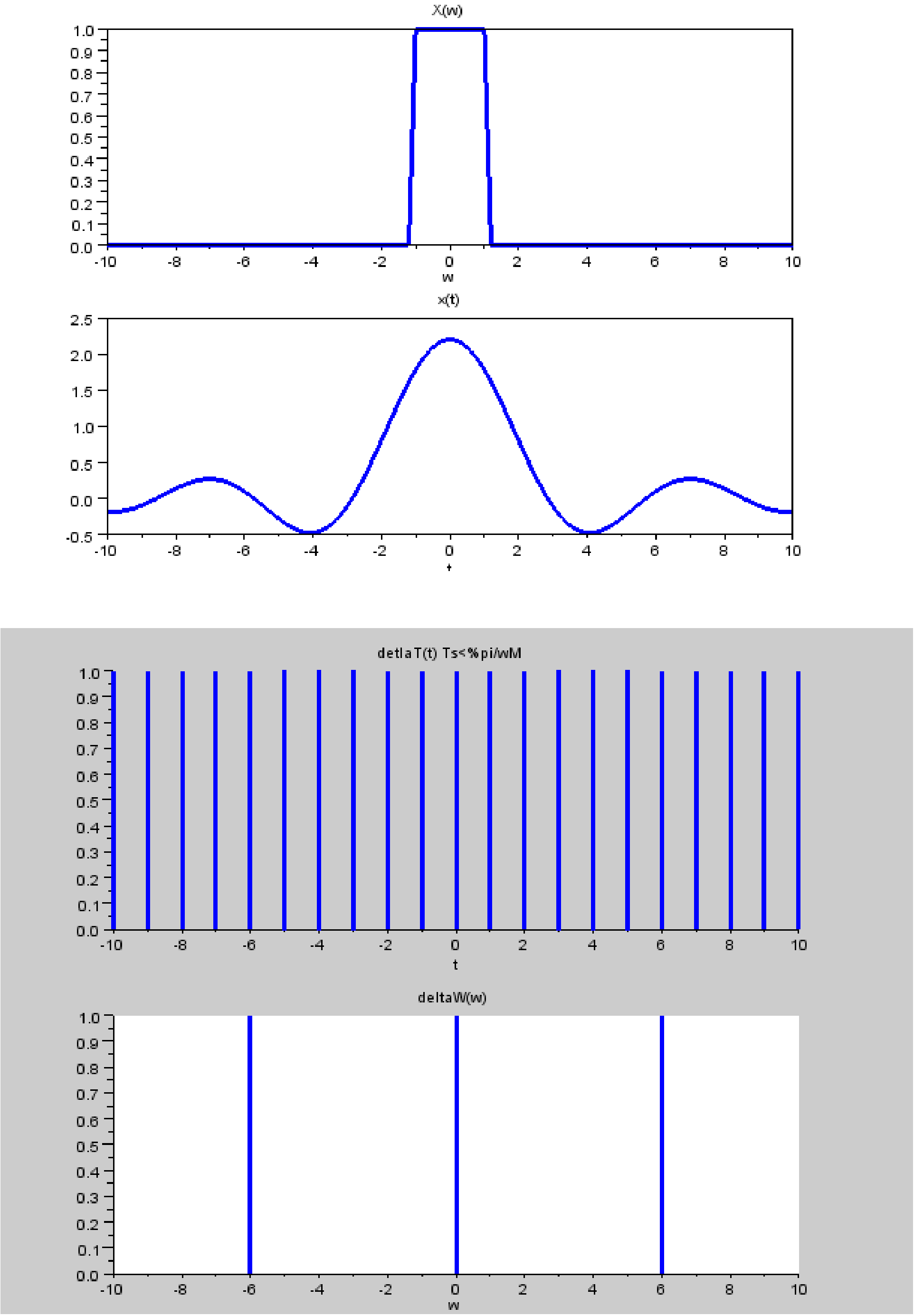
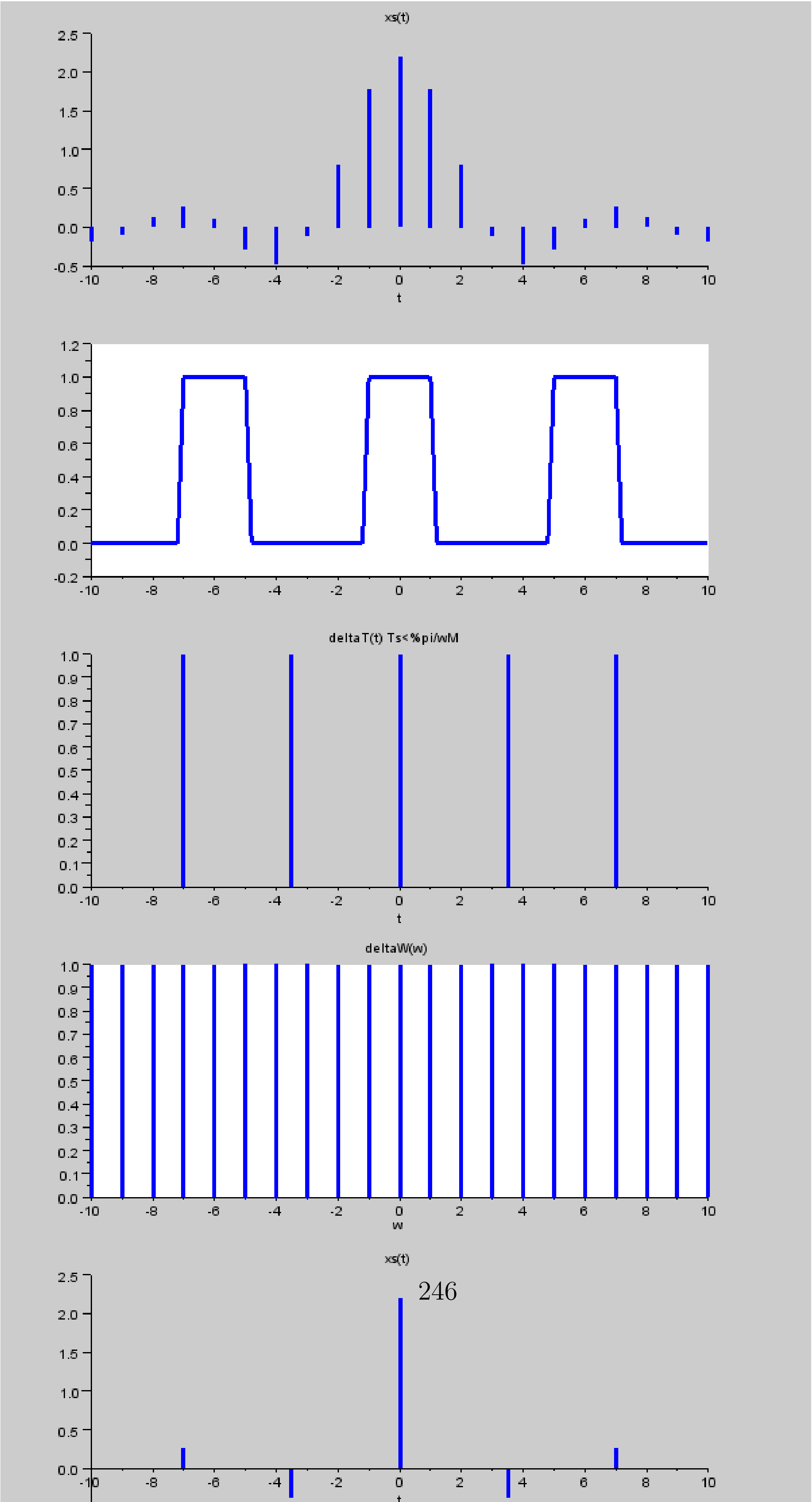


Figure 5.64: fourier spectrum


xs(t)
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-10
-8
-6
-4
-2
0
2
4
6
8
10
t
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-10
-8
-6
-4
-2
0
2
4
6
8
10
deltaT(t) Ts<%pi/wM
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10
t
deltaW(w)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10
w
xs(t)
2.5
2.0
1.5
1.0
0.5
0.0
-10
-8
-6
-4
-2
0
2
4
6
8
10
t

Chapter 6

Fourier analysis of discrete time system and signals

Scilab code Exa 6.3 fourier coefficients

```
clear;
close;
clc;
w0=2*%pi/4;//%pi/2
N0=4;
n=-8:7;
y=0:3;
x=[];
for i=1:length(n)/4
    x=[x y];
end
plot2d3(n,x');
plot(n,x','r.')
xtitle('x[n]','n')
for k=0:3
    c(k+1)=0;
    for n=0:3
```

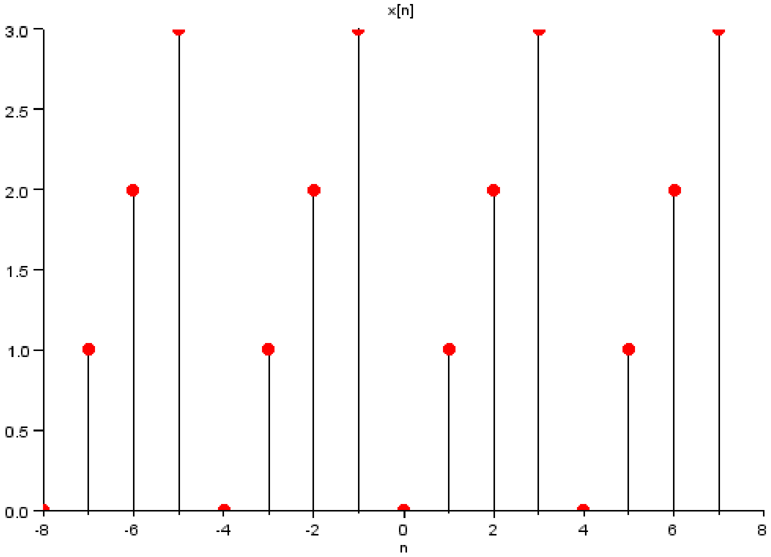


Figure 6.1: fourier coefficients

```
        c(k+1)=c(k+1)+  (1/4)*(y(n+1))*(-%i)^(k*n);
    end
end
```

Scilab code Exa 6.4 fourier coefficients of a periodic sequence

```
clear;
close;
clc;
T0=10;
w0=%pi/5;
y=[ones(1,5) zeros(1,5)];
n=-20:19;
x=[];
for i=1:length(n)/T0
```

```
    x=[x y];
end
plot2d3(n,x);
plot(n,x,'r.')
xtitle('x[n]','n')
for k=0:9
     c(k+1)=0;
    for n=0:4
       c(k+1)=c(k+1)+ (1/4)*(y(n+1))*(%e)^(%i*w0*k*n);
    end
end
c=[c($:-1:2) ;c];
k=-9:9;
figure
plot2d3(k,abs(c));
plot(k,abs(c),'r.--')
xtitle('|ck|','k')
```

Scilab code Exa 6.5 fourier coefficients

```
clear;
close;
clc;
T0=4;
w0=%pi/2;
y=[1 0 0 0];
n=-8:7;
x=[];
for i=1:length(n)/T0
    x=[x y];
```

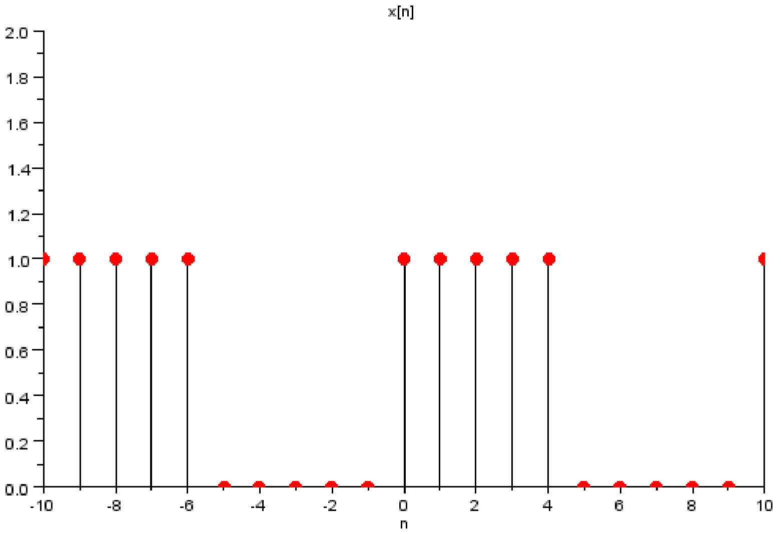


Figure 6.2: fourier coefficients of a periodic sequence

```
end
plot2d3(n,x);
plot(n,x,'r.');
xtitle('x[n]','n')
for k=-9:9
     c(k+10)=0;
    for n=0:3
      c(k+10)=c(k+10)+ (1/4)*(y(n+1))*(%e)^(%i*w0*k*n);
    end
end
k=-9:9;
figure
plot2d3(k,abs(c));
plot(k,c,'r.')
xtitle('x[n]','n')
```

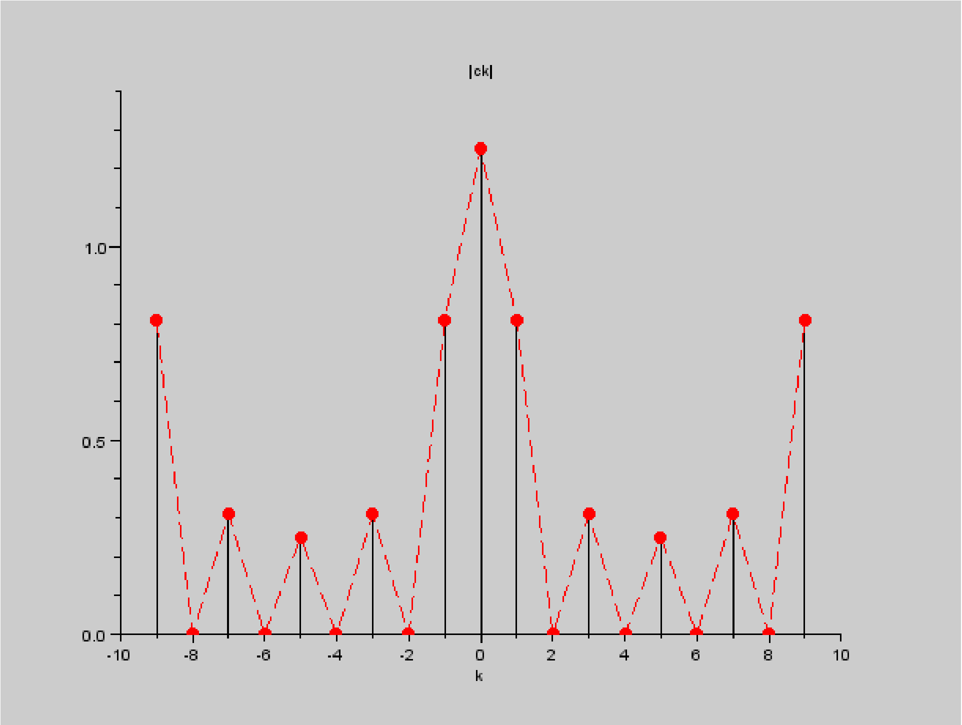


Figure 6.3: fourier coefficients of a periodic sequence

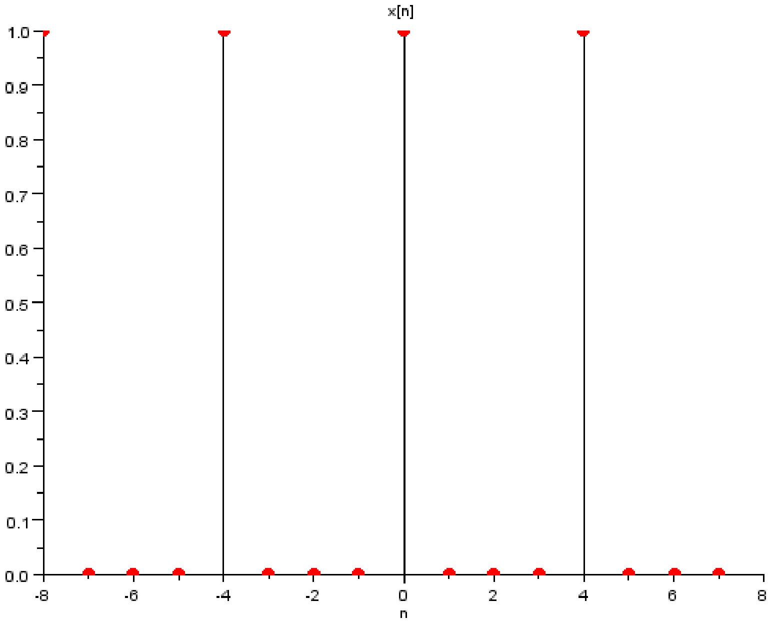


Figure 6.4: fourier coefficients

Scilab code Exa 6.6 discrete fourier series representation

```
clear;
close;
clc;
//cos(%pi*n/4)
N0=8;
w0=2*%pi/N0;
```

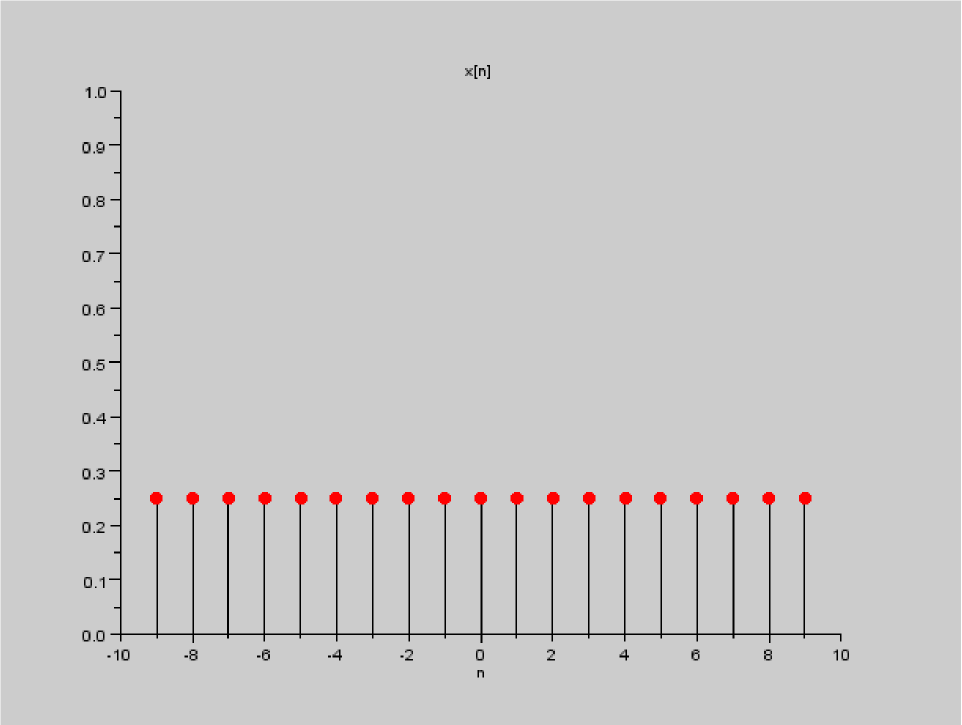


Figure 6.5: fourier coefficients

```
n=-8:8;
x=cos(%pi*n/4);
subplot(2,1,1)
xtitle('x[n]','n')
plot2d3(n,x);
plot(n,x,'r.');
for k=-6:6
     c(k+7)=0;
    for n=0:7
       c(k+7)=c(k+7)+ (1/8)*(x(n+1))*(%e)^(%i*w0*k*n);
    end
end
k=-6:6;
subplot(2,1,2)
xtitle('|ck|','k')
plot2d3(k,abs(c));
plot(k,c,'r.')
//cos(%pi*n/3)+sin(%pi*n/4)
N0=24;
w0=2*%pi/N0;
n=-24:24;
x=cos(%pi*n/3)+sin(%pi*n/4);
figure
subplot(2,1,1)
xtitle('x[n]','n')
plot2d3(n,x);
plot(n,x,'r.--');
for k=-24:24
     c(k+25)=0;
    for n=0:23
       c(k+25)=c(k+25)+ (1/N0)*(x(n+1))*(%e)^(%i*w0*k*n);
    end
end
k=-24:24;
subplot(2,1,2)
xtitle('|ck|','k')
```

```
plot2d3(k,abs(c));
plot(k,c,'r.')
//[cos(%pi*n/8)]^2
N0=8;
w0=2*%pi/N0;
n=-8:8;
x=[cos(%pi*n/8)]^2;
clear c;
figure
subplot(2,1,1)
xtitle('x[n]','n')
plot2d3(n,x);
plot(n,x,'r.--');
for k=-6:6
     c(k+7)=0;
    for n=0:7
        c(k+7)=c(k+7)+ (1/N0)*(x(n+1))*(%e)^(%i*w0*k*n);
    end
end
k=-6:6;
subplot(2,1,2)
xtitle('|ck|','k')
plot2d3(k,abs(c));
plot(k,c,'r.')
disp("fourier series is x[n]=1/N0*sum(c(k)*e^%i*w0*n*k)");
```

Scilab code Exa 6.11 fourier transform

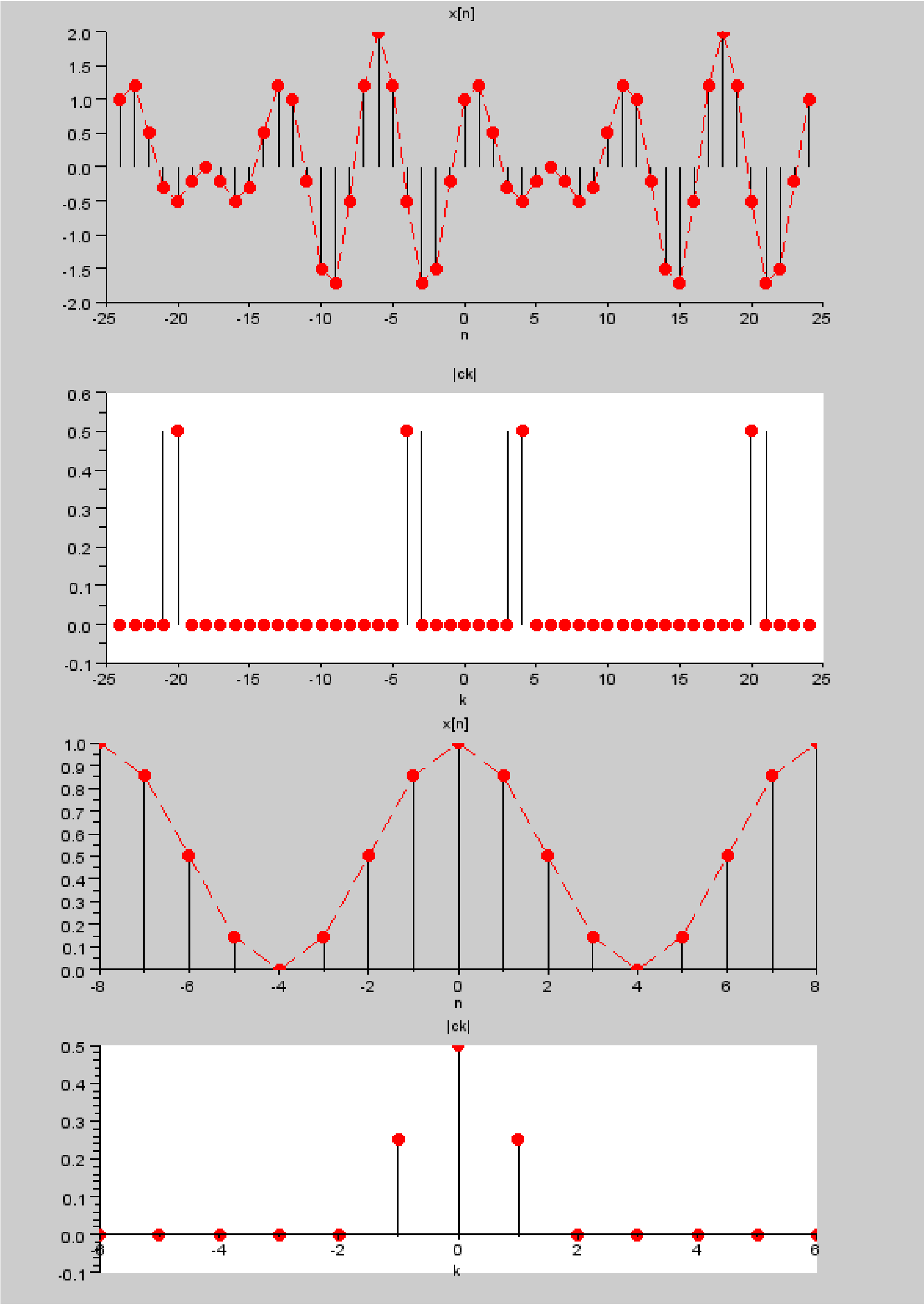


Figure 6.6: discrete fourier series representation

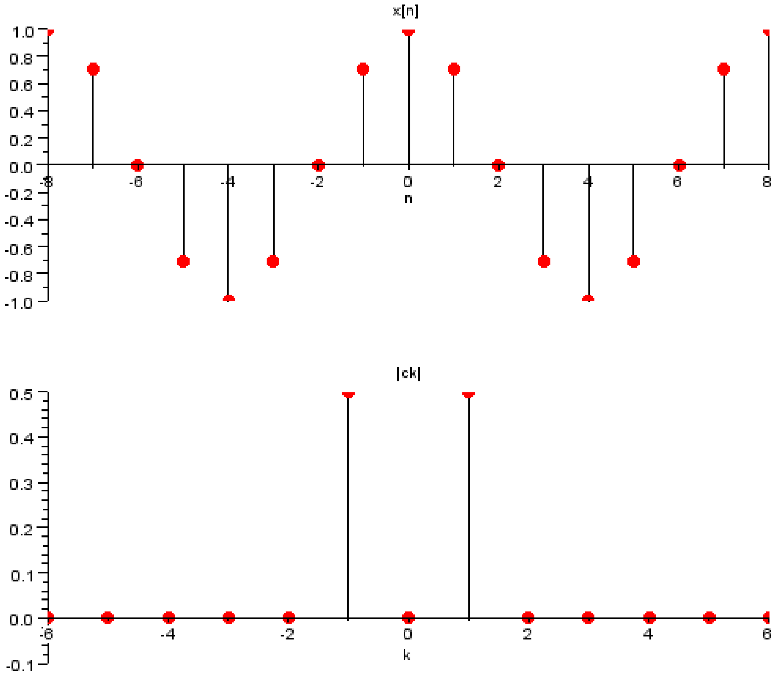


Figure 6.7: discrete fourier series representation

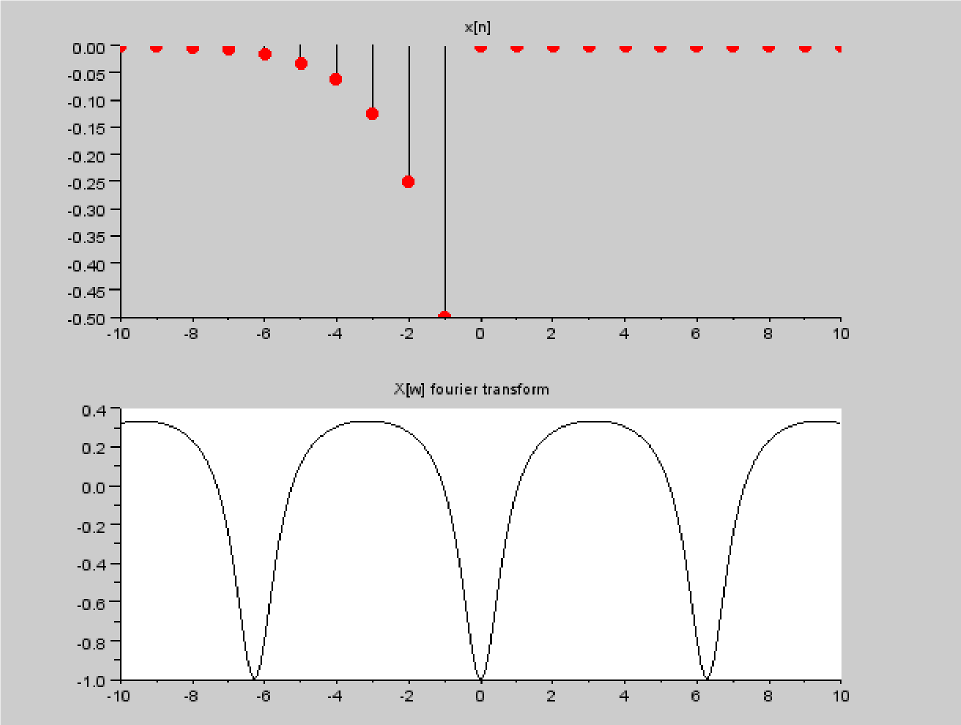


Figure 6.8: fourier transform

```
clear;
close;
clc;
n=-10:10;
a=2;
for i=1:length(n)
    if n(i)<=-1 then
        x(i)=-2^n(i);
    else
        x(i)=0;
    end
end
x=x’;
figure
subplot(2,1,1)
plot2d3(n,x);
title(”x[n]”)
plot(n,x,’r.’)
w=-10:0.1:10;
Xw=x*exp(-%i*n’*w);
subplot(2,1,2)
plot2d(w,Xw);
title(”X[w] fourier transform”)
```

Scilab code Exa 6.12 fourier transform of a rectangular pulse

```
clear;
close;
clc;
n=-10:10;
N=5;
for i=1:length(n)
    if n(i)>=0 & n(i)<=N then
```

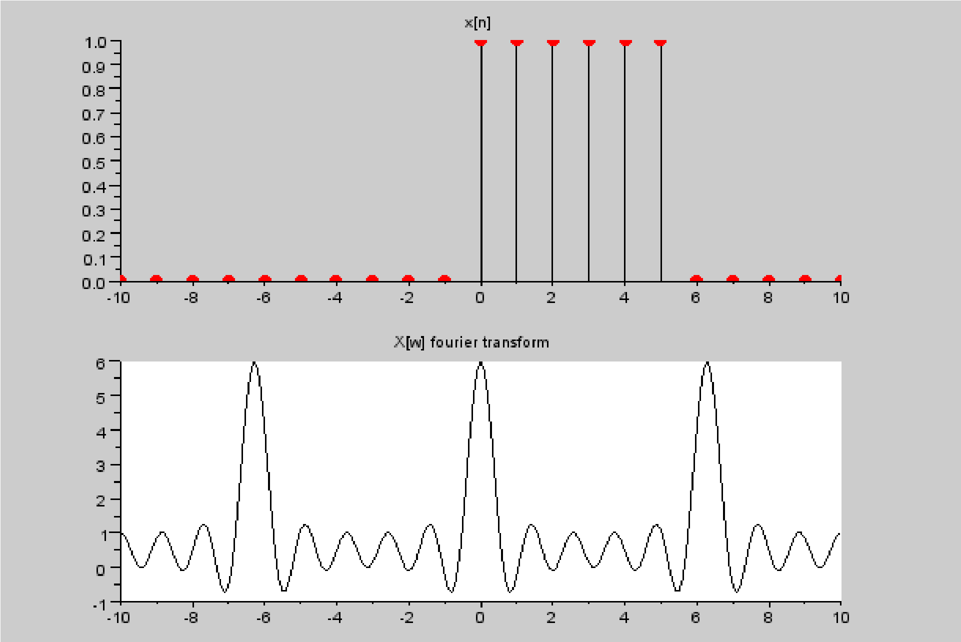


Figure 6.9: fourier transform of a rectangular pulse

```
        x(i)=1;
    else
        x(i)=0;
    end
end
x=x’;
figure
subplot(2,1,1)
plot2d3(n,x);
title(”x[n]”)
plot(n,x,’r.’)
w=-10:0.1:10;
Xw=x*exp(-%i*n’*w);
subplot(2,1,2)
plot2d(w,Xw);
title(”X[w] fourier transform”)
```

Scilab code Exa 6.14 fourier transform

```
clear;
close;
clc;
n=-10:10;
N=4;
for i=1:length(n)
    if n(i)>=-N & n(i)<=N then
        x(i)=1;
    else
        x(i)=0;
    end
end
x=x’;
figure
subplot(2,1,1)
plot2d3(n,x);
title(”x[n]”)
plot(n,x,’r.’)
w=-10:0.1:10;
Xw=x*exp(-%i*n’*w);
subplot(2,1,2)
plot2d(w,Xw);
title(”X[w] fourier transform”)
N=8;
for i=1:length(n)
    if n(i)>=-N & n(i)<=N then
        x(i)=1;
    else
        x(i)=0;
    end
end
x=x’;
```

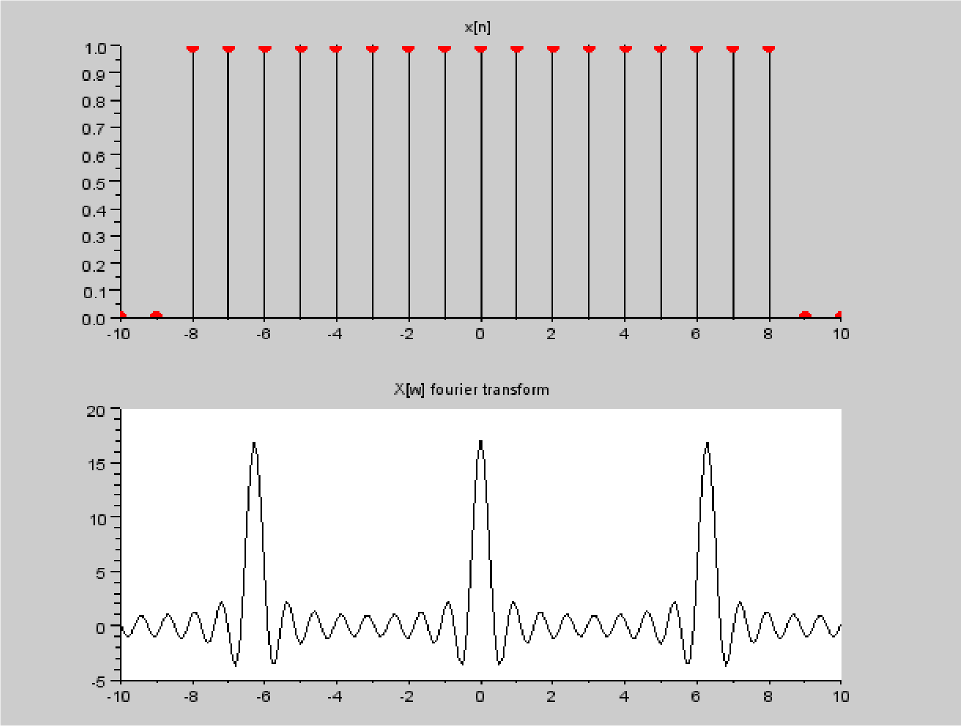


Figure 6.10: fourier transform

```
figure
subplot(2,1,1)
plot2d3(n,x);
title(”x[n]”)
plot(n,x,’r.’)
w=-10:0.1:10;
Xw=x’*exp(-%i*n’*w);
subplot(2,1,2)
plot2d(w,Xw);
title(”X[w] fourier transform”)
```

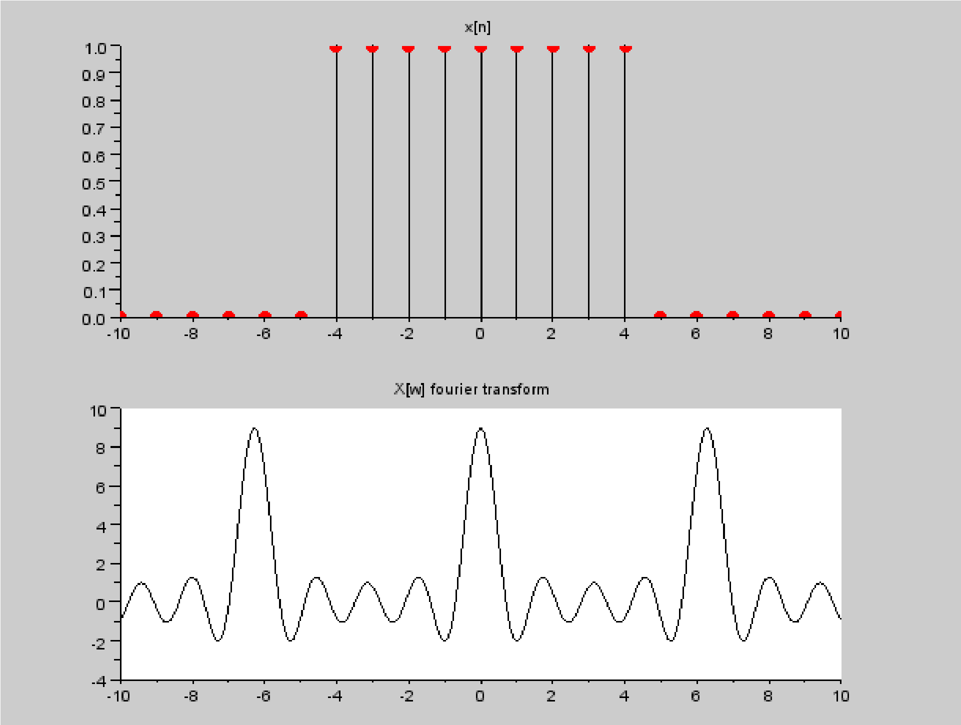


Figure 6.11: fourier transform

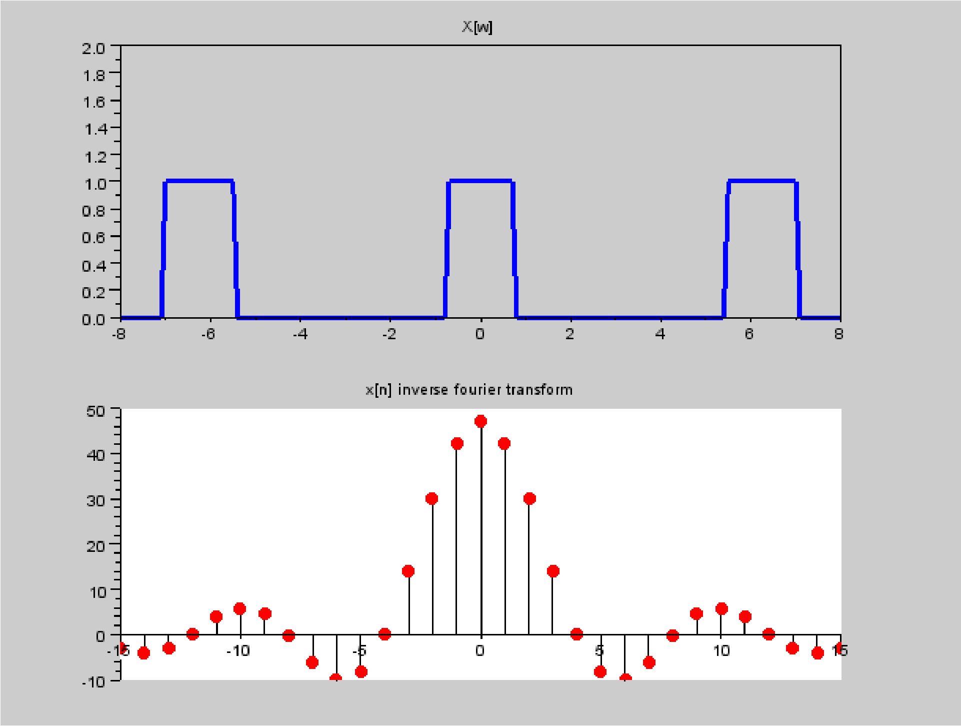


Figure 6.12: inverse fourier transform of a rectangular pulse

Scilab code Exa 6.15 inverse fourier transform of a rectangular pulse

```
clear;
close;
clc;
W=%pi/4;
w=-10:0.1:10;
for i=1:length(w)
    if (w(i)>=-2*%pi-W & w(i)<=-2*%pi+W)  then
   X(i)=1;
```

```
elseif (w(i)>=-W & w(i)<=W)
    X(i)=1;
elseif (w(i)>=2*%pi-W & w(i)<=2*%pi+W)
    X(i)=1;
else
    X(i)=0;
    end
end
figure
subplot(2,1,1)
plot(w,X);
title("X[w]");
n=-15:15;
x=X'*exp(-%i*w'*n);
subplot(2,1,2)
plot2d3(n,x);
plot(n,x,'r.')
title("x[n] inverse fourier transform")
```

Scilab code Exa 6.17 inverse fourier transform of a impulse signal

```
clear;
close;
clc;
w=-5:0.01:5;
W0=2;
for i=1:length(w)
    if w(i)==W0 then
        delta(i)=2*%pi;
    else
        delta(i)=0;
    end
end
```

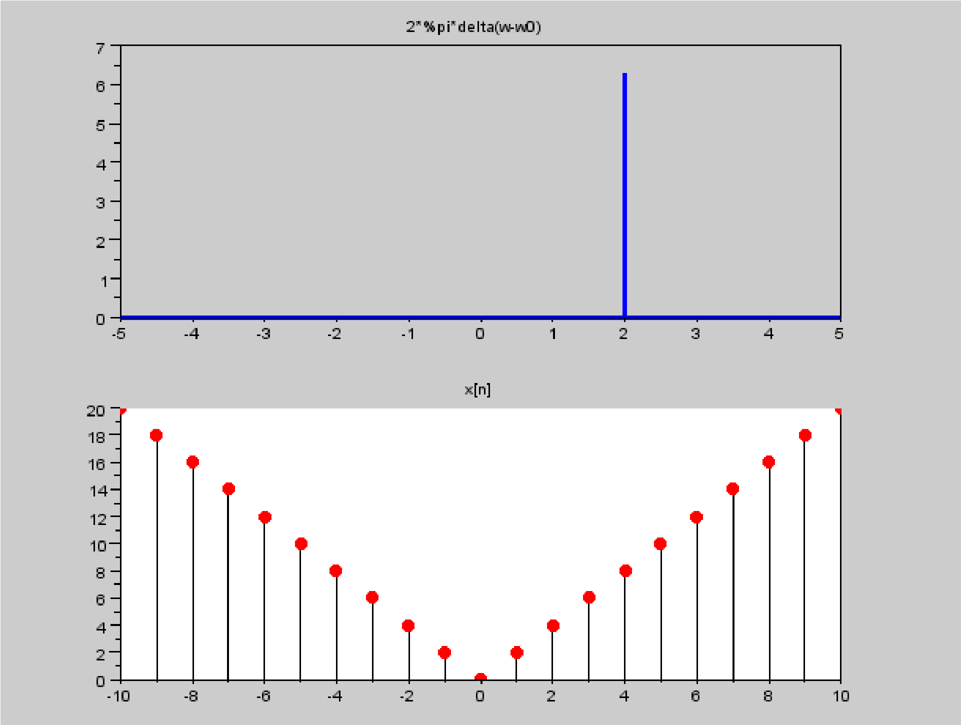


Figure 6.13: inverse fourier transform of a impulse signal

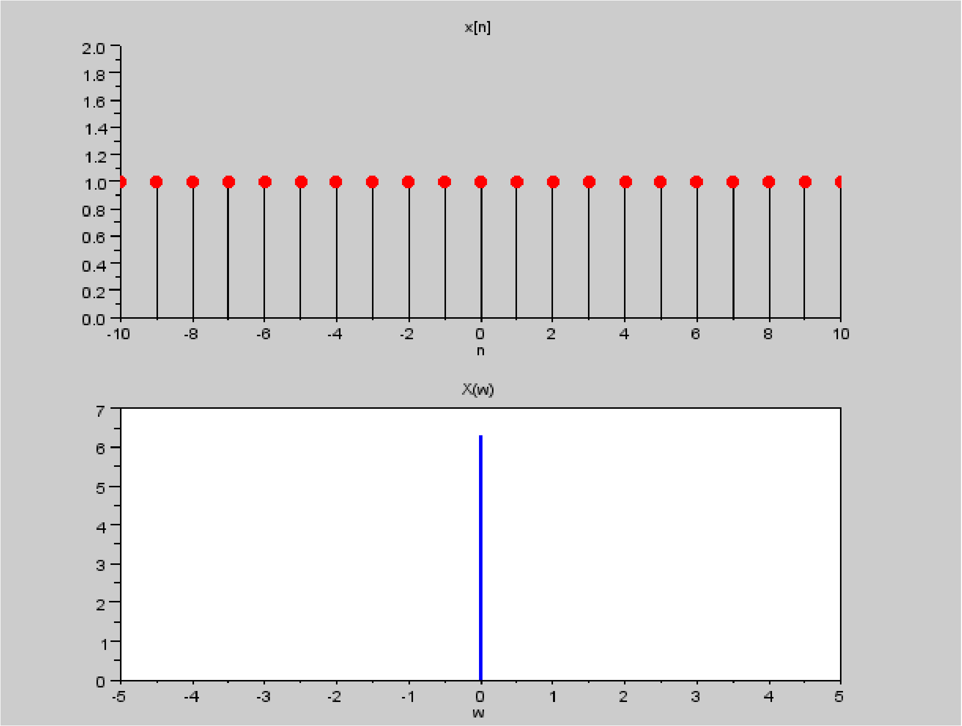


Figure 6.14: fourier transform of constant signal

```
figure
subplot(2,1,1)
plot(w,delta);
title('2*%pi*delta(w-w0)')
//by shifting property
n=-10:10;
x=%e^-%i*W0*n;
subplot(2,1,2)
plot2d3(n,abs(x))
plot(n,abs(x),'r.')
title('x[n]')
```

Scilab code Exa 6.18 fourier transform of constant signal

```
clear;
close;
clc;
n=-10:10;
x=ones(1,length(n));
figure
subplot(2,1,1)
plot2d3(n,x)
plot(n,x,'r.')
xtitle('x[n]','n')
w=-5:0.01:5;
for i=1:length(w)
    if w(i)==0 then
        delta(i)=1;
    else
        delta(i)=0;
    end
end
Xw=2*%pi*delta';
subplot(2,1,2)
plot(w,Xw);
xtitle('X(w)','w')
```

Scilab code Exa 6.19 fourier transform of a sinusoidal sequence

```
clear;
close;
clc;
n=-10:10;
W0=1;
x=cos(W0*n);
```

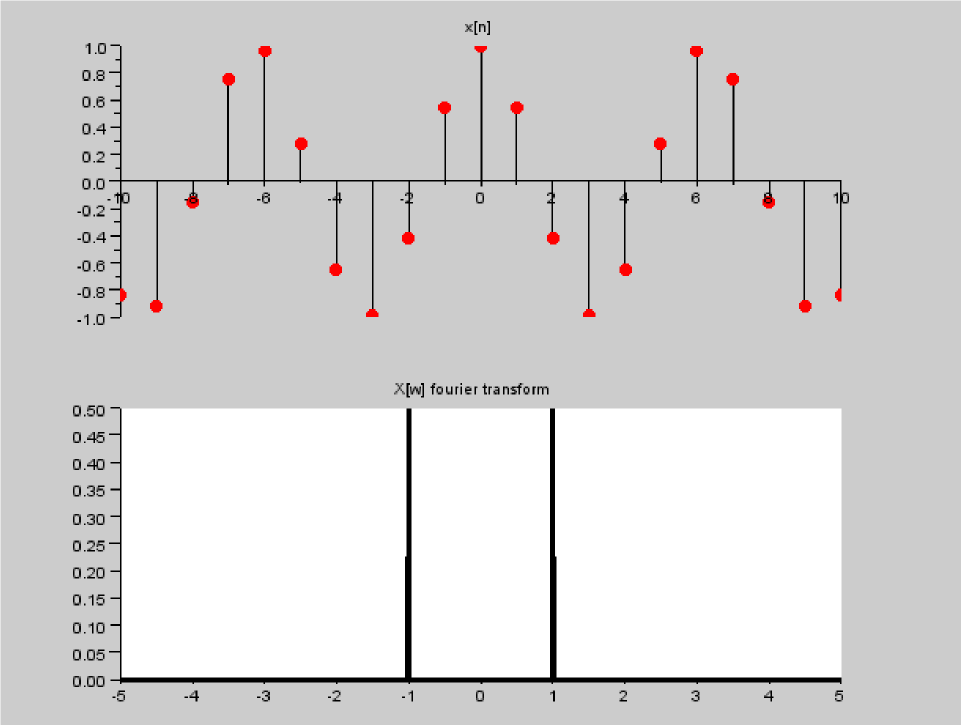


Figure 6.15: fourier transform of a sinusoidal sequence

```
figure
subplot(2,1,1)
plot2d3(n,x);
title("x[n]")
plot(n,x,'r.')
//cos(W0*n)=1/2*(e^-%j*W0*n+e^%j*W0*n)
w=-5:0.01:5;
for i=1:length(w)
    if w(i)==W0 then
        deltaW0(i)=1;
    else
        deltaW0(i)=0;
    end
end
delta_W0=deltaW0($:-1:1);
//by frequency shifting property
Xw=1/2*[deltaW0+delta_W0];
subplot(2,1,2)
plot2d(w,Xw);
title("X[w] fourier transform")
```

Scilab code Exa 6.22 fourier transform

```
clear;
close;
clc;
n=-10:10;
N=2;
for i=1:length(n)
    if n(i)>=-N & n(i)<=N then
        x(i)=1;
    else
        x(i)=0;
    end
end
```

```
x=x’;
figure
subplot(2,1,1)
plot2d3(n,x);
title(”x[n]”)
plot(n,x,’r.’)
w=-10:0.1:10;
Xw=x*exp(-%i*n’*w);
subplot(2,1,2)
plot2d(w,Xw);
title(”X[w] fourier transform”)
//time scaled sequence x2[n]
n2=-20:2:20;
figure
subplot(2,1,1)
plot2d3(n2,x);
title(”x2[n]”)
plot(n2,x,’r.’)
w2=-5:0.05:5;
subplot(2,1,2)
plot2d(w2,Xw);
title(”X2[w] fourier transform”)
//time scaled sequence x3[n]
n3=-30:3:30;
figure
subplot(2,1,1)
plot2d3(n3,x);
title(”x3[n]”)
plot(n3,x,’r.’)
w3=w/3;
subplot(2,1,2)
plot2d(w3,Xw);
title(”X3[w] fourier transform”)
```

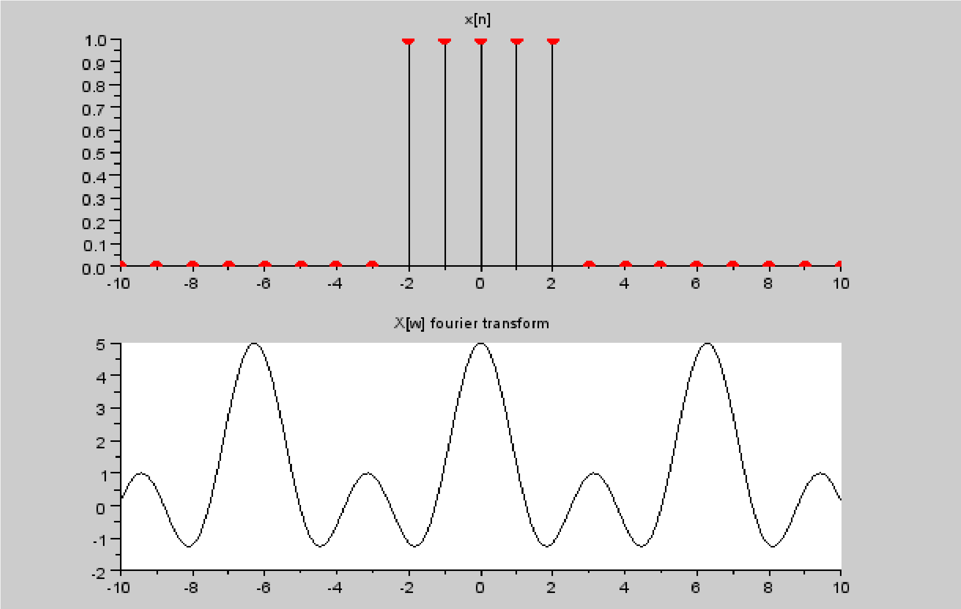


Figure 6.16: fourier transform

Scilab code Exa 6.25 inverse fourier transform using convolution

```
clear;
close;
clc;
a=0.7;
n=-10:10;
disp("X(w)=1/(1-a*e^(%i*w))^2=[1/(1-a*e^(%i*w))]*[1/(1-a*e^(%i*w))]");
disp("we know a^n*u[n] <--> 1/(1-a*e^(%i*w))   ")
//therefore by convolution property
xpartial=[zeros(1,find(n==0)-1) a^n(find(n==0):$)];
x=convol(xpartial,xpartial);
```

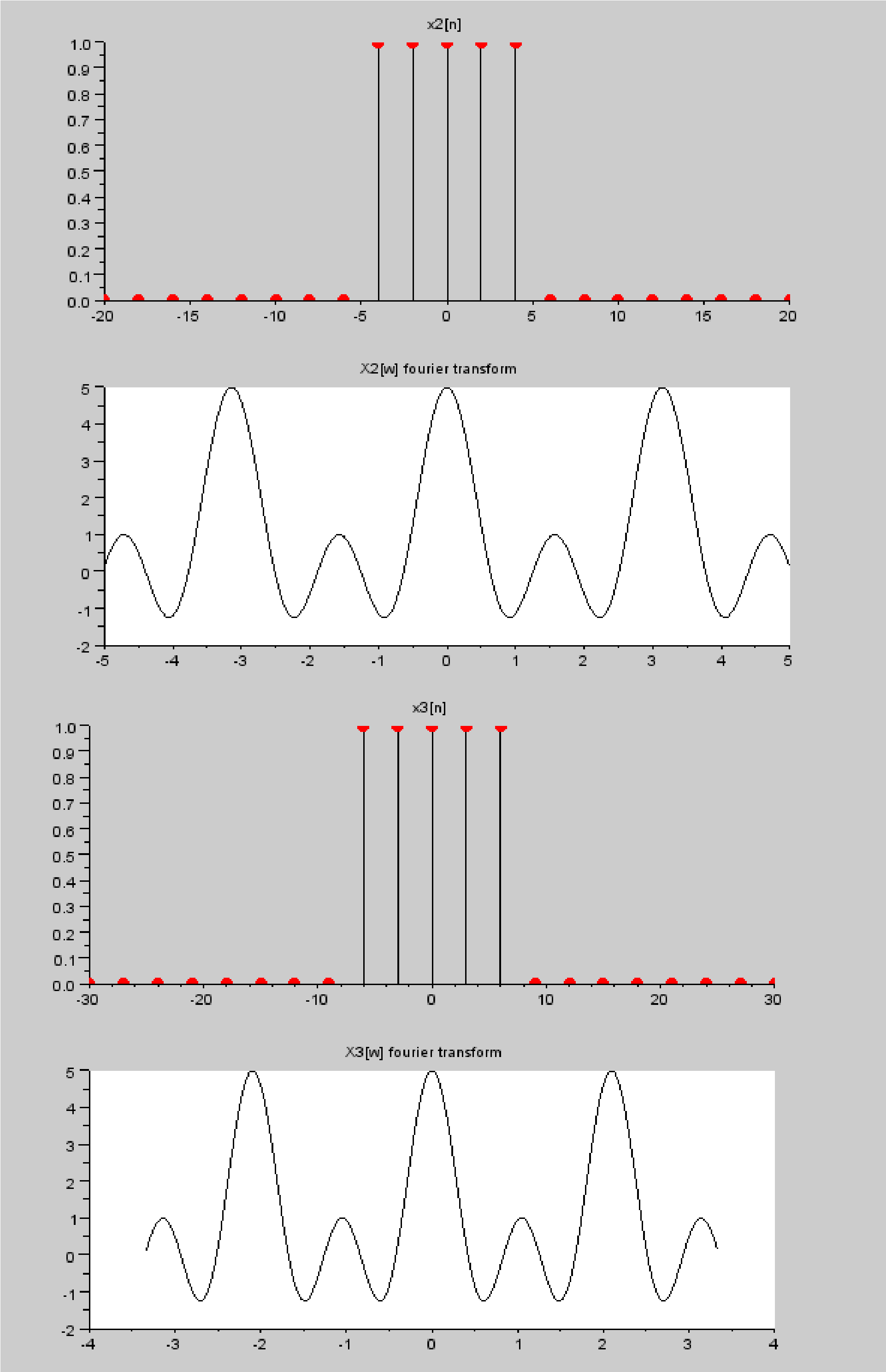


Figure 6.17: fourier transform

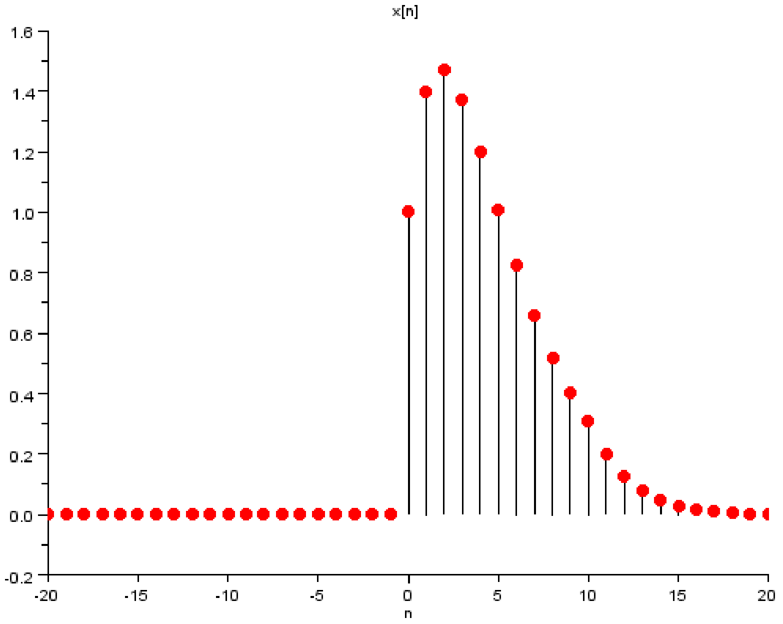


Figure 6.18: inverse fourier transform using convolution

```
n1=-20:20;
plot2d3(n1,x);
plot(n1,x,'r.');
xtitle('x[n]','n')
```

Scilab code Exa 6.28 frequency response

```
clear;
close;
clc;
n=-10:10;
x=[zeros(1,find(n==0)-1) ones(1,length(n)-find(n==0)
   +1)];
plot2d3(n,x)
plot(n,x,'r.')
xtitle('x[n]','n')
figure
w=-2:2;
Xw=x*exp(-%i*n'*w);
subplot(2,1,1)
plot2d(w,real(Xw));
xtitle('U[w] fourier transform','w')
for i=1:length(w)
    if w(i)==0 then
        delta(i)=1;
    else
        delta(i)=0;
    end
end
Xwproof=%pi*delta'+ones(1,length(w))./(1-%e^(-%i*w))
   ;
subplot(2,1,2)
plot(w,Xwproof)
xtitle('%pi*delta+1/(1-e^i*w','w')
disp("it can be seen that both the figures are
```

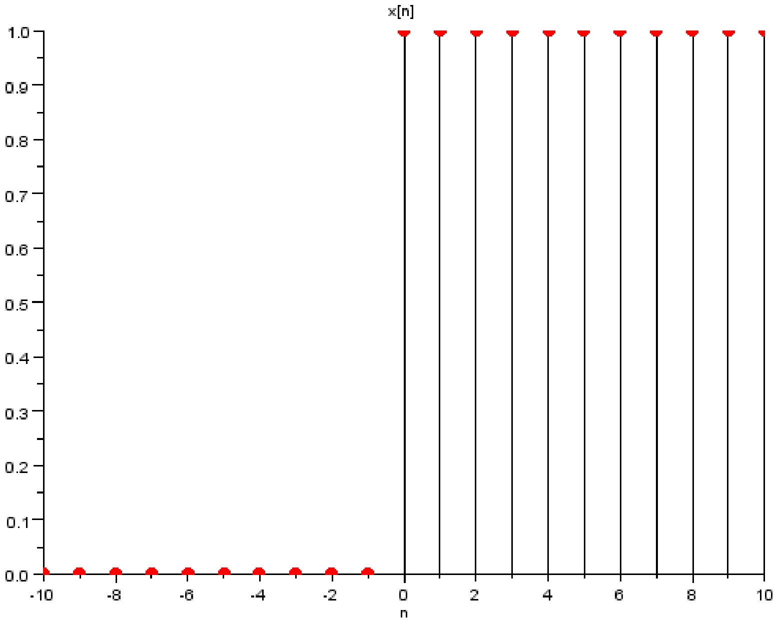


Figure 6.19: frequency response

```
    approximately same hence           X(w)=%pi*delta
    +1/(1-e^i*w");
```

Scilab code Exa 6.31 frequency response

```
clear;
clc;
```

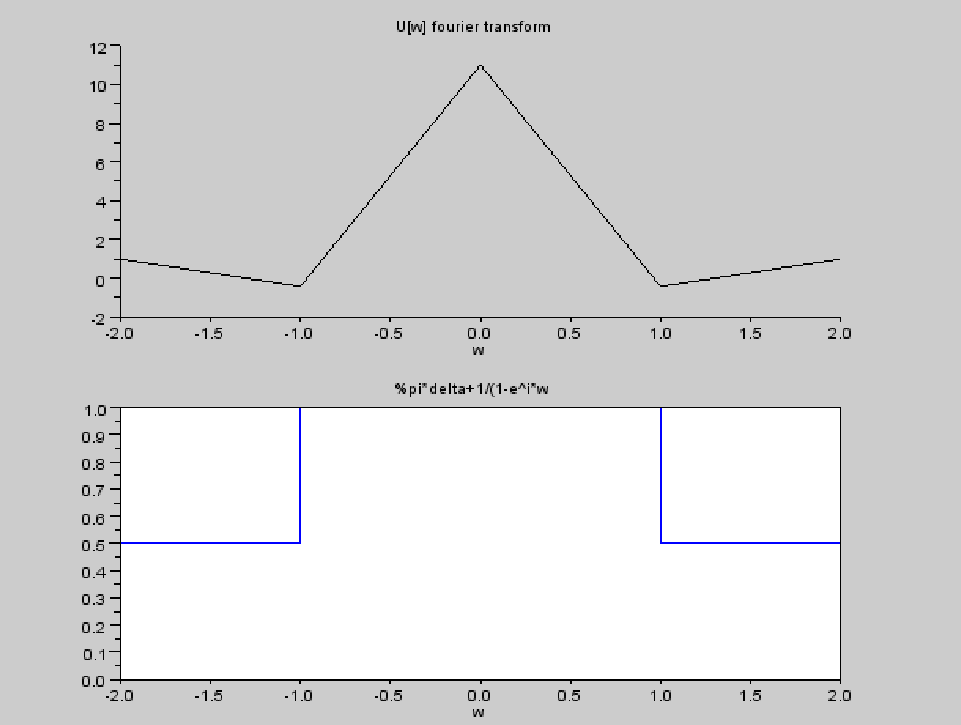


Figure 6.20: frequency response

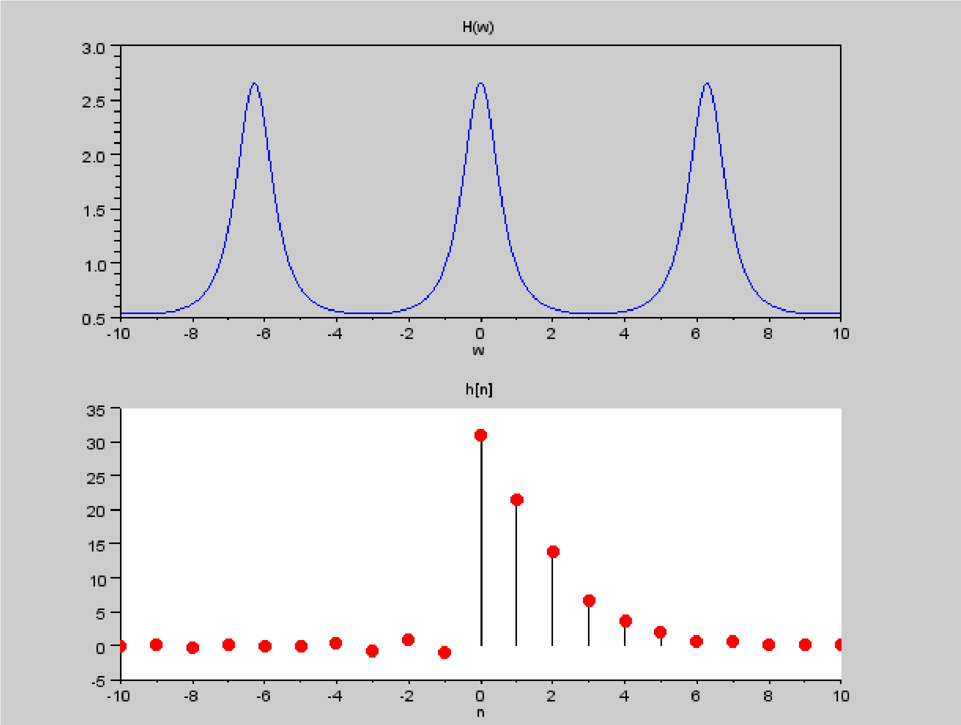


Figure 6.21: frequency response

```
close;
disp("y[n]-3/4y[n-1]+1/8y[n-2]=x[n]");
disp("taking fourier transform on both sides");
disp("H(w)=Y(w)/X(w)=1/(1-(3/4)*e^(-j*w)+(1/8)*e^(-j
   *2*w))");
w=-10:0.1:10;
Hw=ones(1,length(w))./(1-(3/4)*%e^(-%i*w)+(1/8)*%e
   ^(-%i*2*w));
figure
subplot(2,1,1)
plot(w,Hw);
xtitle('H(w)','w')
n=-10:10;
h=(1/(2*%pi))*Hw*exp(%i*w'*n);
subplot(2,1,2)
plot2d3(n,h)
plot(n,h,'r.')
xtitle('h[n]','n')
```

Scilab code Exa 6.32 frequency response

```
clear;
clc;
close;
disp("y[n]-1/2*y[n-1]=x[n]+1/2*x[n-1]");
disp("taking fourier transform on both sides");
disp("H(w)=Y(w)/X(w)=(1+1/2*e^-j*w)/(1-1/2*e^-j*w)")
   ;
w=-10:0.1:10;
Hw=(1+(1/2)*%e^(-%i*w))./(1-(1/2)*%e^(-%i*w));
figure
subplot(2,1,1)
plot(w,Hw);
xtitle('H(w)','w')
n=-10:10;
```

```
h=(1/(2*%pi))*Hw*exp(%i*w'*n);
subplot(2,1,2)
plot2d3(n,h)
xtitle('h[n]','n')
plot(n,h,'r.')
x=cos(%pi*n/2);
figure
subplot(2,1,1)
plot2d3(n,x);
xtitle('x[n]','n')
plot(n,x,'r.')
y=convol(x,h);
subplot(2,1,2)
plot2d3(n,y(11:31))
xtitle('y[n]','n')
plot(n,y(11:31),'r.')
```

Scilab code Exa 6.33 output response

```
clear;
clc;
close;
n=-20:19;
h=sinc(%pi*n/4)/4;
disp("it can be seen that h[n] is a filter that allows frequencies between      -%pi/4 and +%pi/4")
disp("therefore only the dc part of x[n] is passed i.e c0")
subplot(2,1,1)
plot2d3(n,h);
```

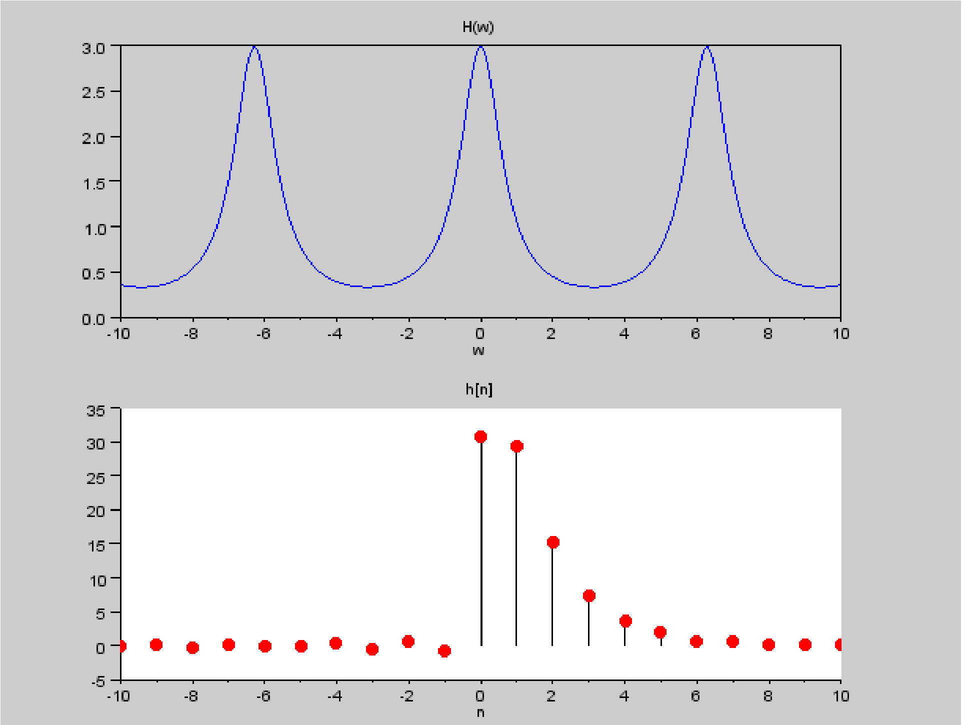


Figure 6.22: frequency response

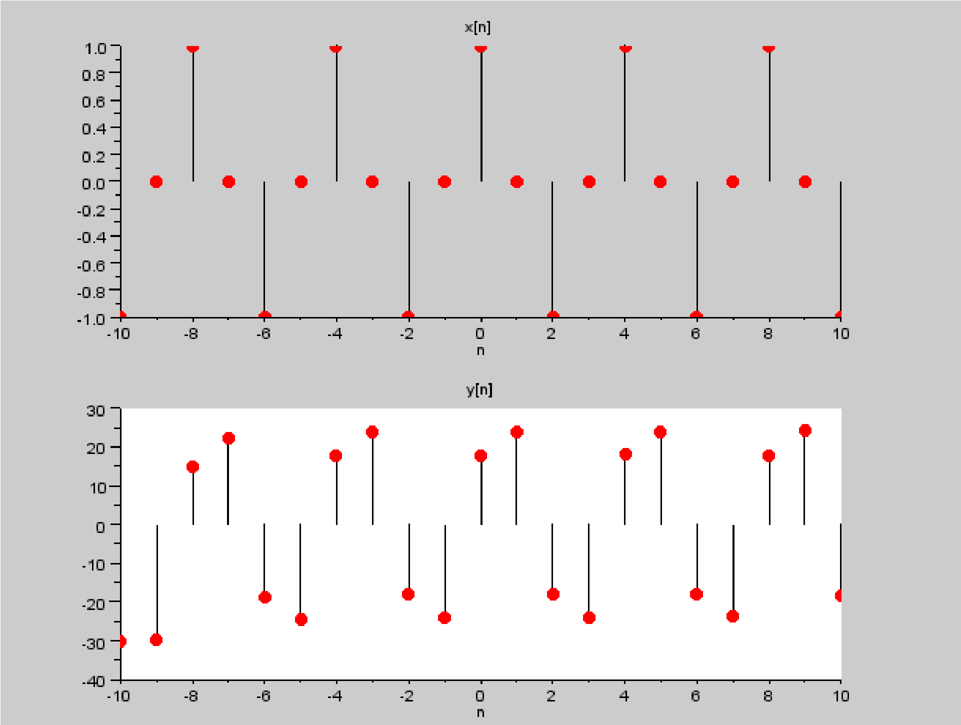


Figure 6.23: frequency response

```
plot(n,h,'r.');
xtitle('h[n]','n')
w=-4:0.01:4;
Hw=h*exp(-%i*n'*w);
subplot(2,1,2)
plot2d(w,Hw);
title("H[w] fourier transform")
T0=5;
w0=%pi*2/5;
z=[ones(1,3) zeros(1,2)];
x=[];
for i=1:length(n)/T0
    x=[x z];
end
figure
subplot(2,1,1)
plot2d3(n,x);
plot(n,x,'r.')
xtitle('x[n]','n')
c0= (1/5)*sum(z);
y=ones(1,length(n))*c0;
subplot(2,1,2)
plot2d3(n,y);
plot(n,y,'r.--')
xtitle('y[n]','n')
```

Scilab code Exa 6.34 magnitude and phase response

```
clear;
clc;
close;
```

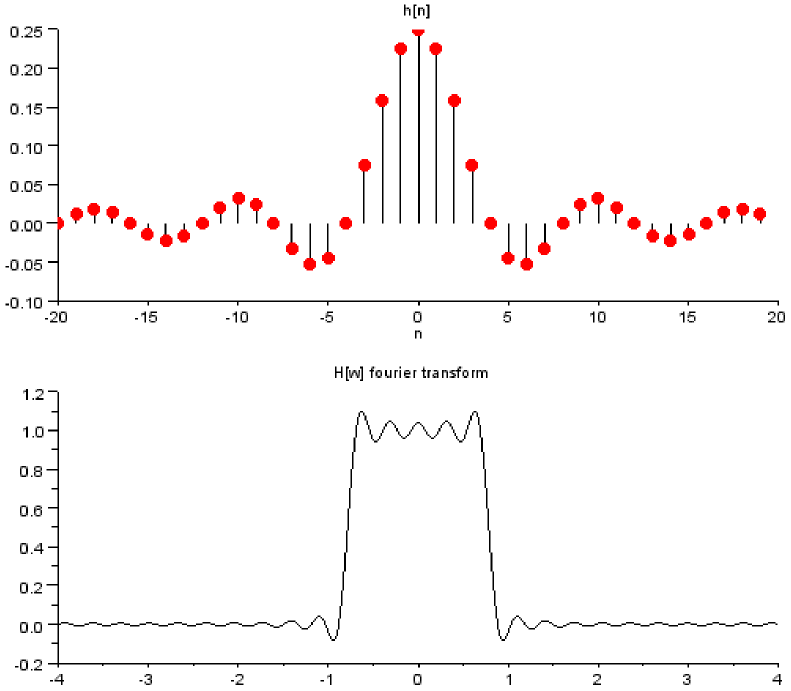


Figure 6.24: output response

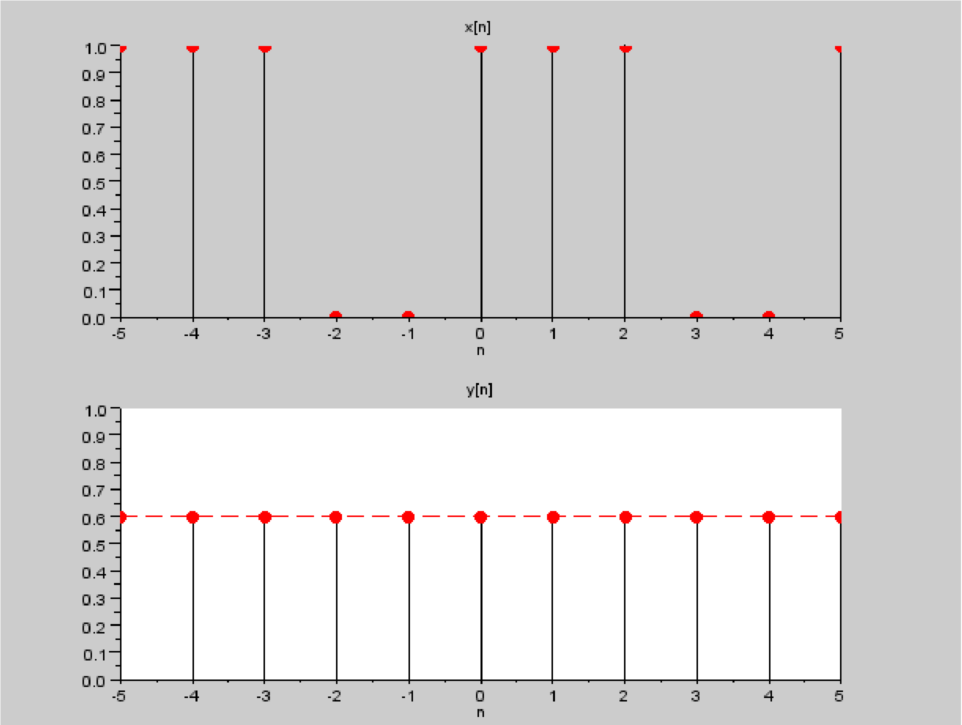


Figure 6.25: output response

```
disp(”given system is y[n]=x[n]+x[n−1]”);
disp(”taking fourier transform H(w)=Y(w)/X(w)=1+e^−j*w”);
//for impulse response x[n]=delta[n]
n=-10:10;
for i=1:length(n)
    if n(i)==0 then
        delta1(i)=1;
        delta2(i)=0;
    elseif n(i)==1
        delta2(i)=1;
        delta1(i)=0
    else
        delta1(i)=0;
        delta2(i)=0
    end
end
h=delta1+delta2;
plot2d3(n,h);
plot(n,h,'r.')
xtitle('h[n]','n')
figure
subplot(2,1,1)
w=-3:0.01:3;
Hw=1+%e^(-%i*w);
plot(w,abs(Hw))
xtitle('|H(w)|','w')
subplot(2,1,2)
a=gca();
plot(w,phasemag(Hw)*%pi/180)
xtitle('theta(H(w))','w')
a.y_location=”origin”;
//3−db bandwidth
cutoff=find(round(100*abs(Hw))==round(100*max(abs(Hw))/sqrt(2)));
threedb=w(cutoff(3));
disp(threedb,”3db bandwidth=”)
```

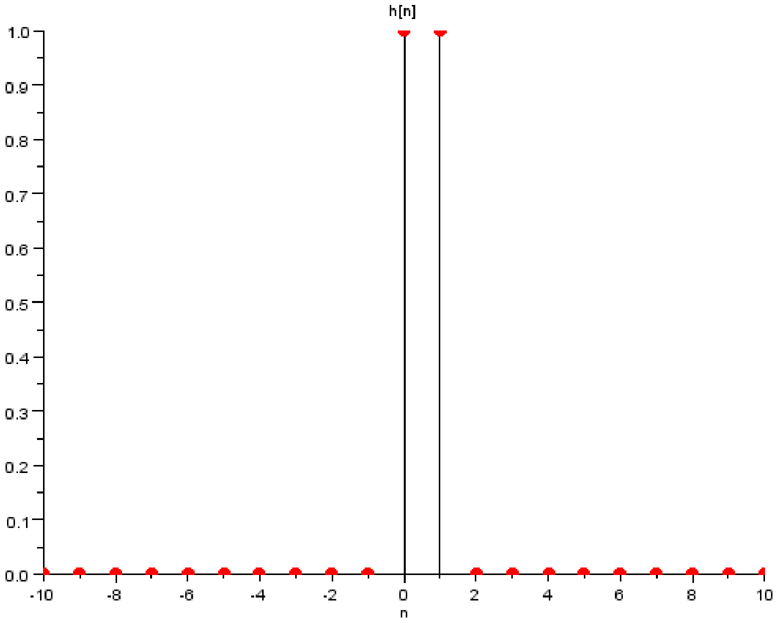


Figure 6.26: magnitude and phase response

Scilab code Exa 6.35 frequency response

```
clear;
clc;
close;
disp("given system is y[n]-a*y[n-1]=x[n]");
disp("taking fourier transform H(w)=Y(w)/X(w)=1/(1-a*e^-j*w)");
```

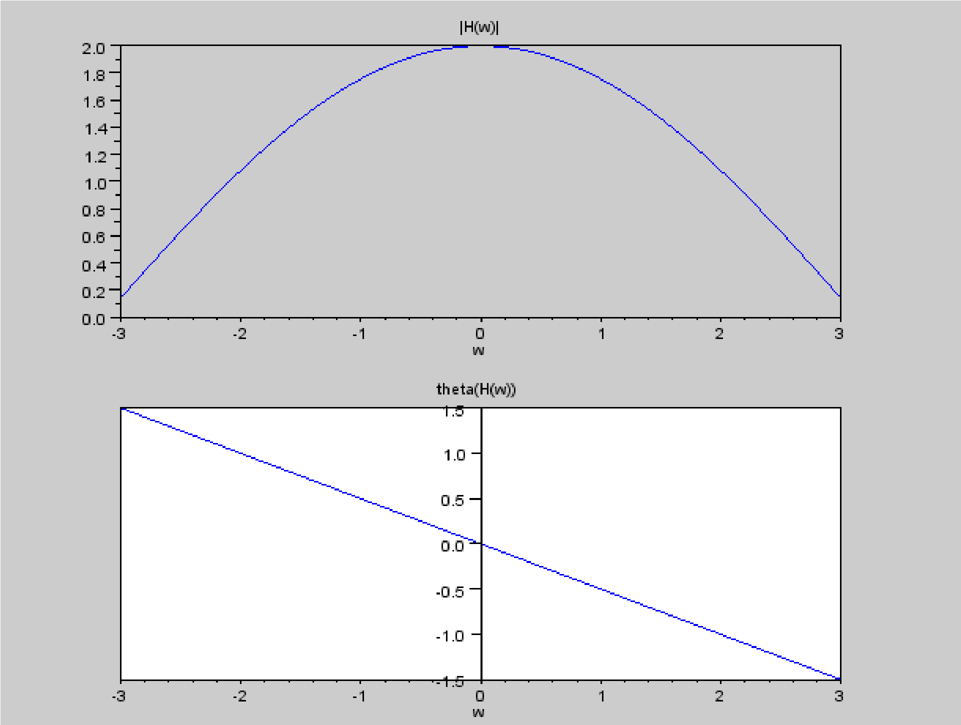


Figure 6.27: magnitude and phase response

```
//impulse response
n=-10:10;
w=-3:0.01:3;
a=.5;
Hw=ones(1,length(w))./(1-a*%e^(-%i*w));
h=(1/2*%pi)*Hw*exp(%i*w'*n);
disp("impulse response is a^n*u[n]")
plot2d3(n,h);
plot(n,h,'r.')
xtitle('h[n]','n')
figure
plot(w,abs(Hw),'r')
a=0.9;
Hw=ones(1,length(w))./(1-a*%e^(-%i*w));
plot(w,abs(Hw),'b')
xtitle('|H(w)|','w')
legend(['a=0.5';'a=0.9']);
```

Scilab code Exa 6.36 discrete time low pass filter

```
clear;
clc;
close;
wLPF=-3:0.1:3;
wc=1;
for i=1:length(wLPF)
    if wLPF(i)>-wc & wLPF(i)<wc then
        HwLPF(i)=1;
    else
```

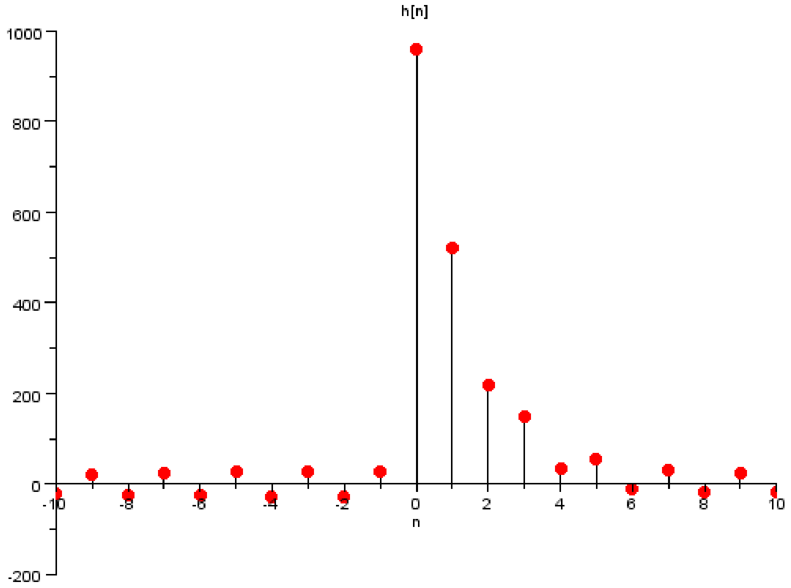


Figure 6.28: frequency response

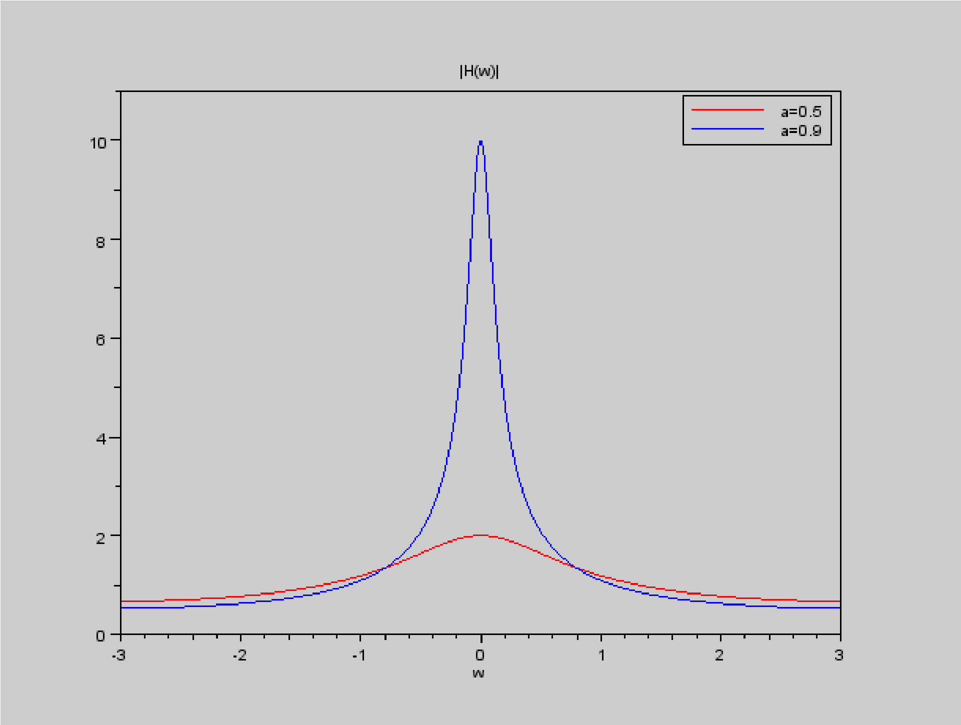


Figure 6.29: frequency response

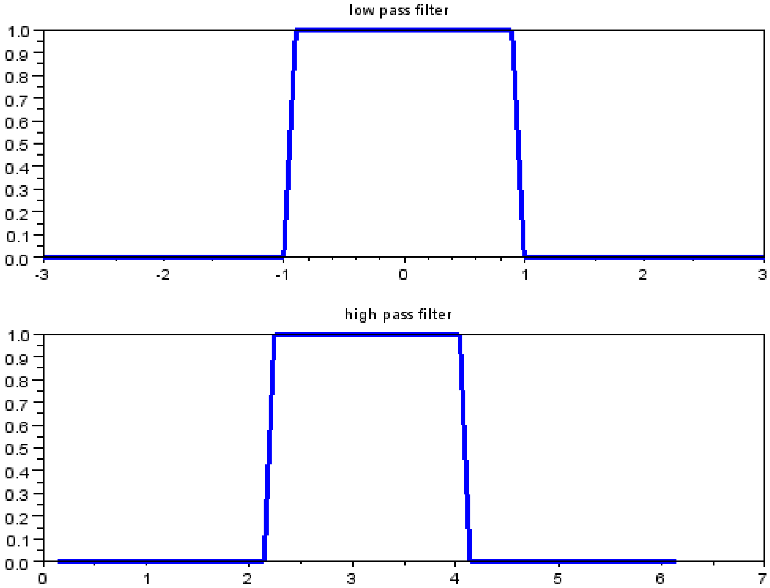


Figure 6.30: discrete time low pass filter

```
        HwLPF(i)=0;
    end
end
subplot(2,1,1)
plot(wLPF,HwLPF)
a=gca()
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
title("low pass filter")
disp("given h[n]=(-1)^n * hLPF[n]=e^(j*%pi*n) * hLPF[n]");
disp("by shifting property H(w)=HLPF(w-%pi)")
w=wLPF+%pi;
subplot(2,1,2)
plot(w,HwLPF)
a=gca()
poly1=a.children.children;
```

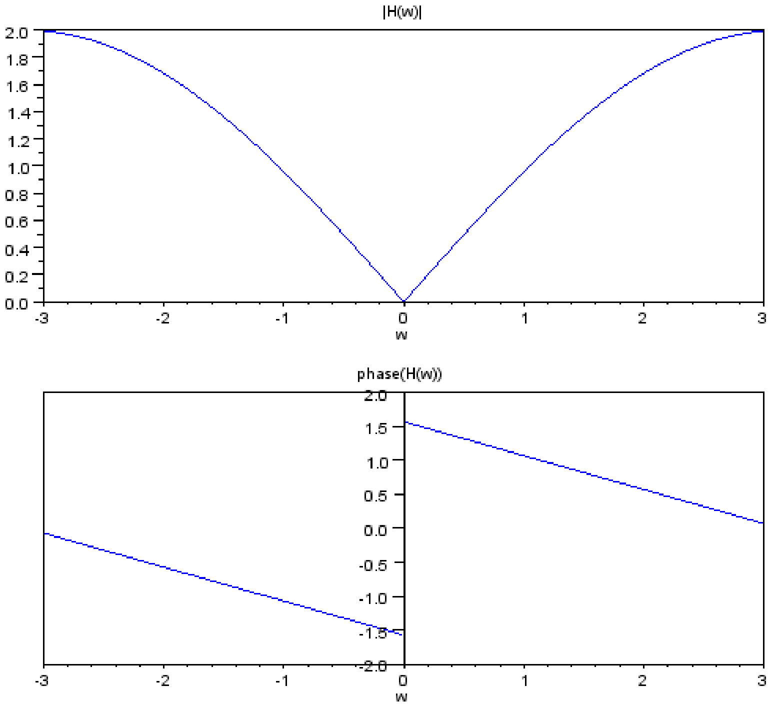


Figure 6.31: convertion of LPF to HPF

```
poly1.thickness=3;
poly1.foreground=2;
title("high pass filter")
```

Scilab code Exa 6.38 convertion of LPF to HPF

```
clear;
clc;
close;
disp("given y[n]=x[n]+x[n-1]  this can be converted
   to high pass filter by      multiplying with (-1)
^n")
disp("then y[n]=x[n]-x[n-1] taking fourier transform");
disp("H(w)=1-e^-j*w");
w=-3:0.01:3;
Hw=1-%e^(-%i*w);
subplot(2,1,1)
plot(w,abs(Hw));
xtitle('|H(w)|','w')
subplot(2,1,2)
plot(w(1:find(w==0)-1),phasemag(Hw(1:find(w==0)-1))*%pi/180)
a=gca();
a.y_location="origin";
plot(w(find(w==0)+1:$),phasemag(Hw(find(w==0)+1:$))*%pi/180)
xtitle('phase(H(w))','w')
```

Scilab code Exa 6.40 impulse response os a FIR filter

```
close;
clc;
clear;
N=7;
n=-4:12;
for i=1:length(n)
    if n(i)>=0 & n(i)<=(N-1)/2 then
        h(i)=n(i);
    elseif n(i)>=(N-1)/2 & n(i)<N
        h(i)=-n(i)+(13+(-1)^N)/2;
    else
        h(i)=0;

    end
end
```

```
subplot(2,1,1)
plot2d3(n,h)
plot(n,h,'r.')
title("N is odd")
N=8;
n=-4:12;
for i=1:length(n)
    if n(i)>=0 & n(i)<=(N-1)/2 then
        h(i)=n(i);
    elseif n(i)>=(N-1)/2 & n(i)<N
        h(i)=-n(i)+(13+(-1)^N)/2;
    else
        h(i)=0;

    end
end
subplot(2,1,2)
plot2d3(n,h)
plot(n,h,'r.')
title("N is even")
disp("given h[n]=h[N-1-n] taking DFTF we get")
disp("H(w)=H*(w) e^(-j*(N-1)*w)");
disp("phase(w)=-1/2(N-1)*w");
//h[n]=-h[N-1-n]
N=7;
n=-4:12;
for i=1:length(n)
    if n(i)>=0 & n(i)<N then
        h(i)=n(i)-(N-1)/2;
    else
        h(i)=0;

    end
end
figure
subplot(2,1,1)
plot2d3(n,h)
plot(n,h,'r.')
```

```
title(”N is odd”)
N=8;
n=-4:12;
for i=1:length(n)
    if n(i)>=0 & n(i)<N then
        h(i)=n(i)-(N-1)/2;
      else
        h(i)=0;
    end
end
subplot(2,1,2)
plot2d3(n,h)
plot(n,h,’r.’)
title(”N is even”)
disp(”given h[n]=−h[N−1−n] taking DFTF we get”)
disp(”H(w)=−H∗(w) eˆ(−j∗(N−1)∗w)”);
disp(”phase(w)=%pi/2 − 1/2(N−1)∗w”)
```

Scilab code Exa 6.41 three point moving average discrete time filter

```
close;
clc;
clear;
disp(”given system y[n]=1/3∗{x[n]+x[n−1]+x[n−2]}”);
disp(”taking h[n]=1/3∗{delta[n]+delta[n−1]+delta[n
   −2]}”)
n=-10:10;
for i=1:length(n)
    if n(i)==0 then
        delta1(i)=1;
        delta2(i)=0;
```

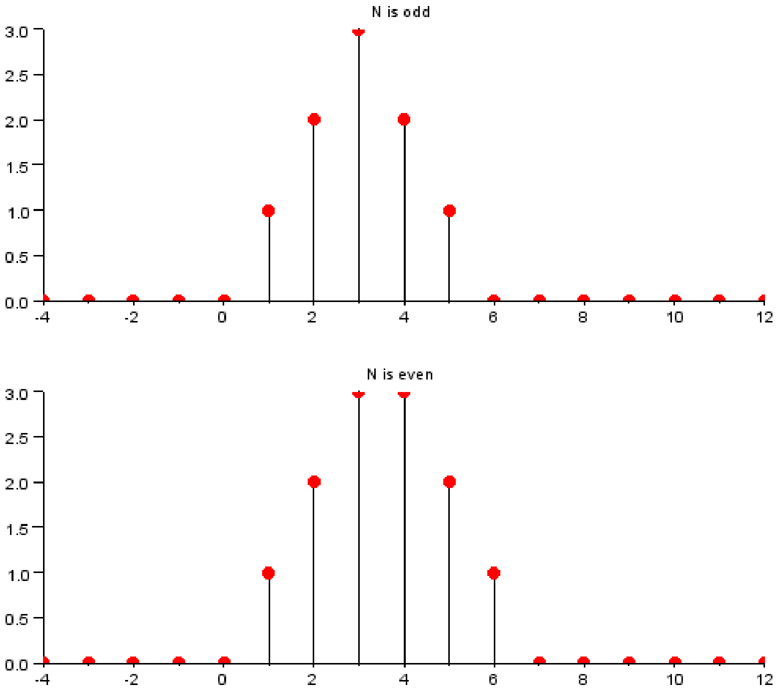


Figure 6.32: impulse response os a FIR filter

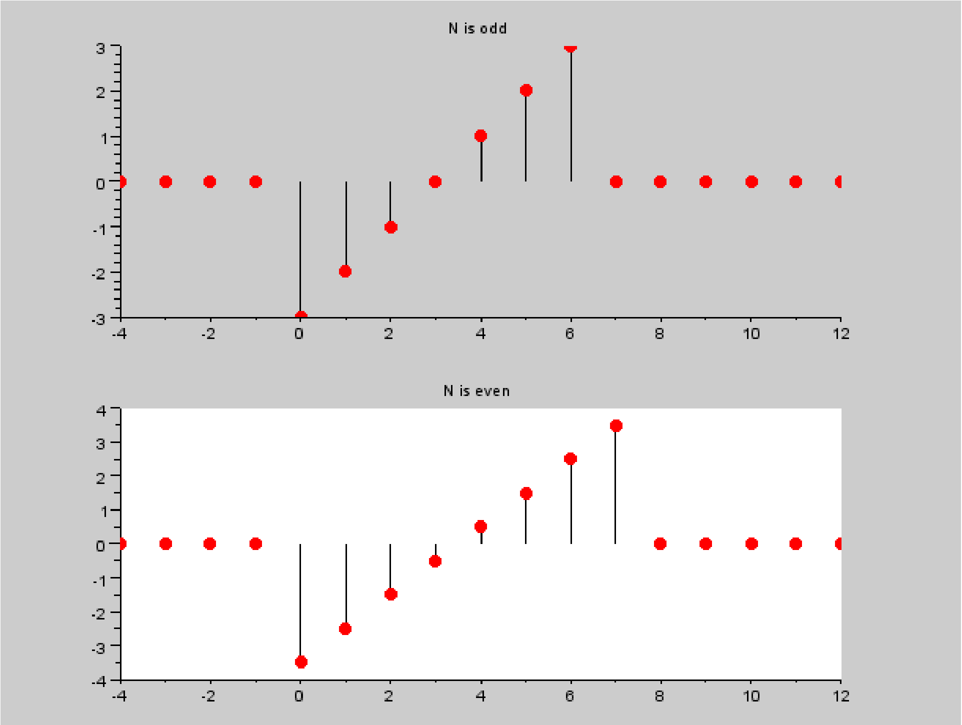


Figure 6.33: impulse response os a FIR filter

```
        delta3(i)=0;
    elseif n(i)==1
        delta2(i)=1;
        delta1(i)=0;
        delta3(i)=0;
    elseif n(i)==2
        delta1(i)=0;
        delta2(i)=0;
        delta3(i)=1;
    else
        delta1(i)=0;
        delta2(i)=0;
        delta3(i)=0;
    end
end
h={delta1+delta2+delta3}/3;
plot2d3(n,h)
plot(n,h,'r.')
xtitle('h[n]','n')
disp("taking DTFT H(w)={1+e^-j*w+e^-2*j*w}/3");
w=-3:0.1:3;
Hw={1+%e^(-%i*w)+%e^(-2*%i*w)}/3;
figure
subplot(2,1,1)
plot(w,abs(Hw));
xtitle('|H(w)|','w')
subplot(2,1,2)
plot(w,phasemag(Hw)*%pi/180);
xtitle('phase(H(w))','w')
```

Scilab code Exa 6.42 causal discrete time FIR filter

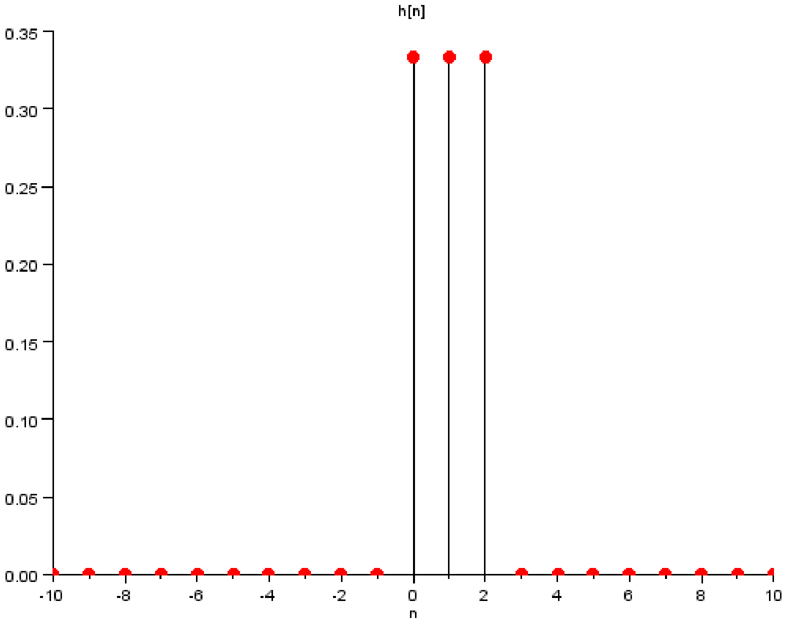


Figure 6.34: three point moving average discrete time filter

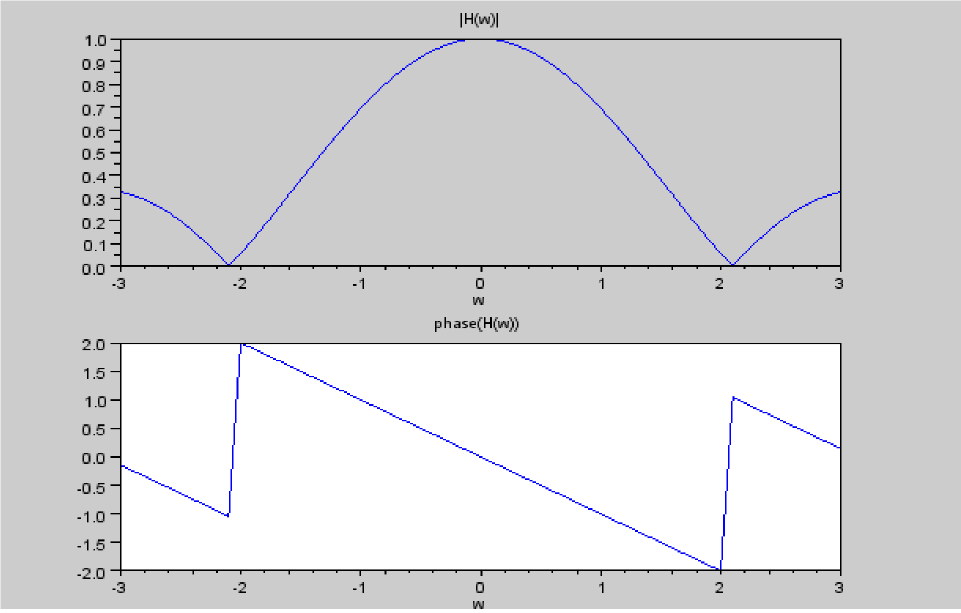


Figure 6.35: three point moving average discrete time filter

```
close;
clc;
clear;
n=-10:10;
h=[zeros(1,find(n==0)-1) 2 2 -2 -2 zeros(1,length(n)-find(n==0)-3)];
plot2d3(n,h)
plot(n,h,'r.')
title("impulse response h[n]")
disp("taking DTFT H(w)=sin(w/2)+sin(3*w/2)")
w=-3:0.01:3;
Hw=h*exp(-%i*n'*w);
figure
subplot(2,1,1)
plot(w,abs(Hw)/4)
xtitle('|H(w)|','w')
subplot(2,1,2)
plot(w(1:find(w==0)-1),phasemag(Hw(1:find(w==0)-1))*
```

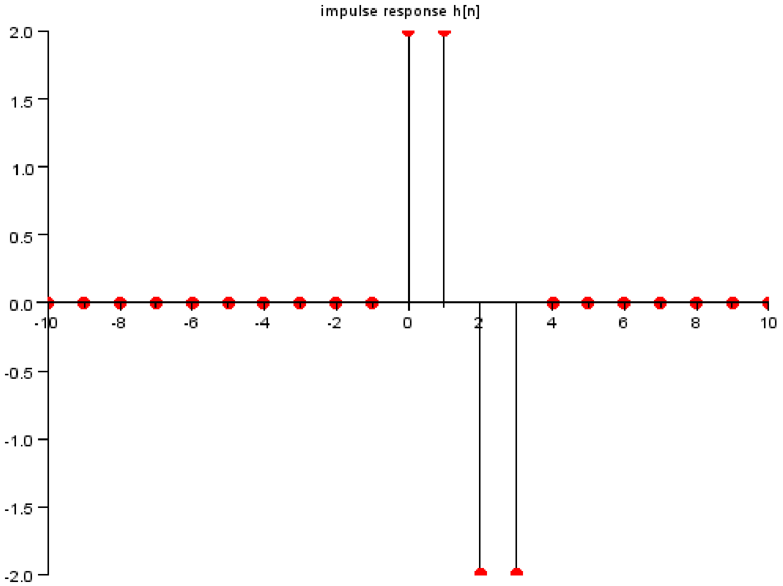


Figure 6.36: causal discrete time FIR filter

```
    %pi/180)
a=gca();
a.y_location=”origin”;
xtitle(’phase(H(w))’,’w’)
plot(w(find(w==0)+1:$),phasemag(Hw(find(w==0)+1:$))*
    %pi/180)
```

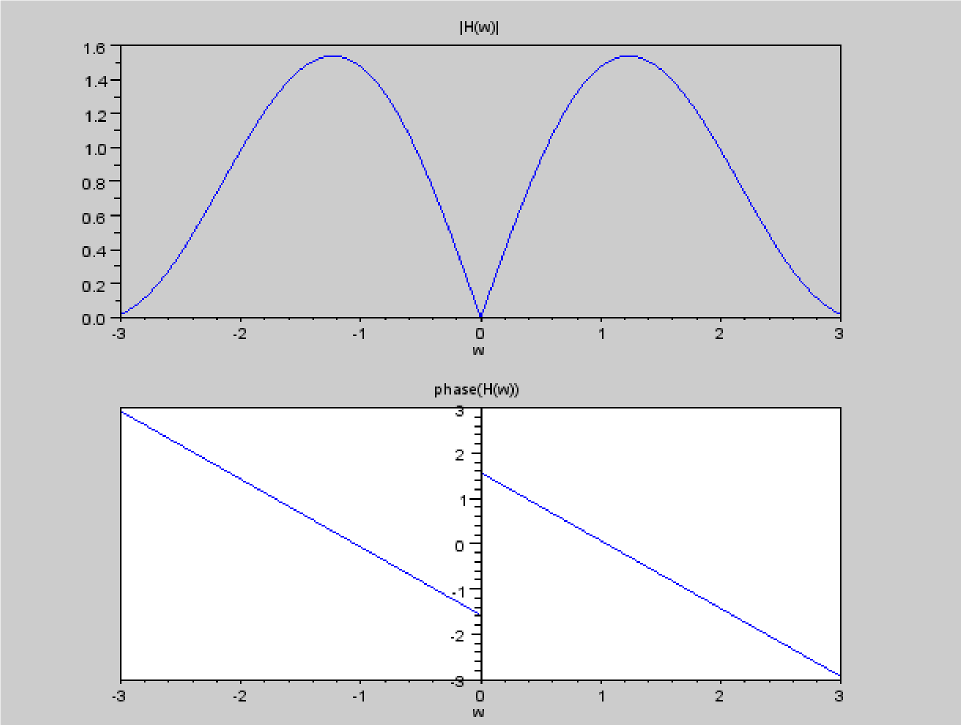


Figure 6.37: causal discrete time FIR filter

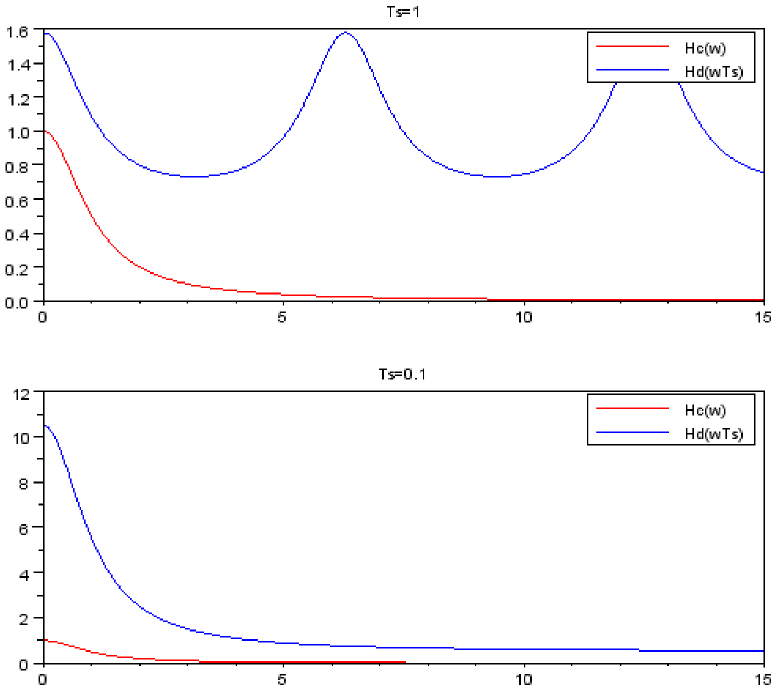


Figure 6.38: Rc low pass filter

Scilab code Exa 6.43 Rc low pass filter

```
close;
clear;
clc;
disp("the system can be represented as Yc(s)=Xc(s)/(1+RCs)");
disp("therefore Hc(s)=1/(1+RCs) hence hc(t)=e^-t*u(t) by frequency shifting      property")
disp("therefore hd[n]=hc(t)=e^(-n*Ts)*u[n]");
disp("taking z-transform Hd(z)=1/(1-e^(-n*Ts)*z^-1)");
Ts=1;
w=0:0.1:15;
Hcw=ones(1,length(w))./(1+%i*w);
subplot(2,1,1)
plot(w,Hcw,'r')
//z=%e^%i*w*Ts
title("Ts=1")
Hdw=ones(1,length(w))./(1-exp(-Ts-%i*w*Ts));
plot(w,Hdw,'b')
legend(["Hc(w)";"Hd(wTs)"])
Ts=0.1;
w=0:0.1:15;
Hcw=ones(1,length(w))./(1+%i*w);
subplot(2,1,2)
plot(w,Hcw,'r')
//z=%e^%i*w*Ts
title("Ts=0.1")
Hdw=ones(1,length(w))./(1-exp(-Ts-%i*w*Ts));
//Hdw=ones(1,length(w))./(1+%e^(-2*Ts)-2*%e^(-Ts)*cos(w*Ts))^.5;
plot(w,Hdw,'b')
legend(["Hc(w)";"Hd(wTs)"])
```

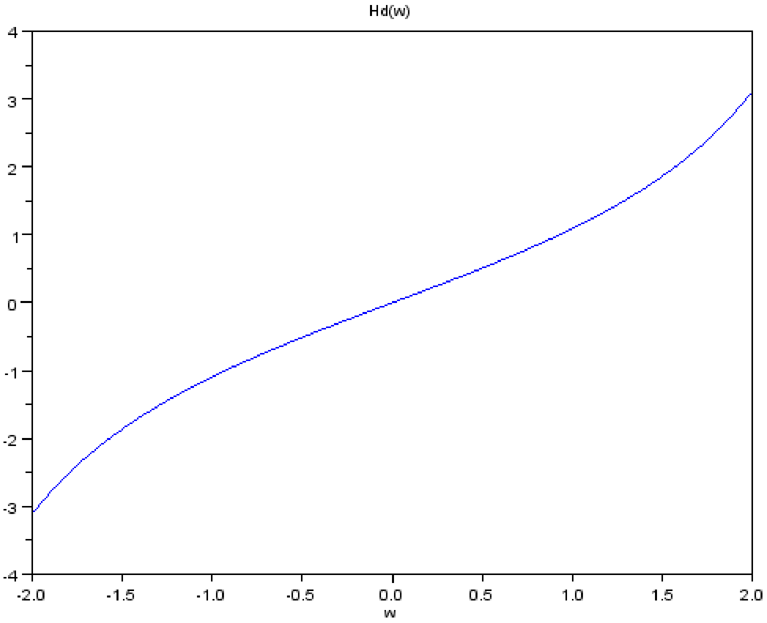


Figure 6.39: bilinear transformation

Scilab code Exa 6.44 frequency response

```
close;
clear;
clc;
disp("the system is Hc(s)=1/(s+1)*(s+2)");
disp("therefore hc(t)=(e^-t-e^-2*t)*u(t) by
   frequency shifting     property")
disp("therefore hd[n]=hc(t)=(e^(-n*Ts)-e^(-2*n*Ts))*
   u[n]");
disp("taking z-transform Hd(z)=1/(1-e^(-n*Ts)*z^-1)
    -  1/(1-e^(-2*n*Ts)*z^-1");
```

Scilab code Exa 6.45 bilinear transformation

```
close;
clear;
clc;
disp("Hc(s)=s")
disp(" bilinear transformation s=2*(1-z^-1)/Ts*(1+z^-1)");
disp("Hd(z)=2*(1-z^-1)/Ts*(1+z^-1)")
//z=e^-j*w*Ts
w=-1:0.01:1;
Ts=2;
Hdw=2*(1-%e^(-%i*w*Ts))./(1+%e^(-%i*w*Ts));
//HdwTs=2*tan(w*Ts/2)/Ts;
plot(w*Ts,imag(Hdw));
xtitle('Hd(w)','w')
```

Scilab code Exa 6.49 N point DFT

```
close;
clc;
clear;
n=0:10;
N=8;
for i=1:length(n)
    if n(i)==0 then
        delta(i)=1;
    else
        delta(i)=0;
    end
end
x=delta';
```

```
subplot(2,1,1)
plot2d3(n,x);
a=gca();
plot(n,x,'r.')
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x[n]','n')
X=dft(x,-1);
Y=X(1:find(n==N)-1);
subplot(2,1,2)
plot2d3(n,[Y' 0 0 0]);
a=gca();
plot(n,[Y' 0 0 0],'r.')
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X[k]','k')
for  i=1:length(n)
    if n(i)>=0 & n(i)<N then
        x(i)=1;
    else
        x(i)=0;
    end

end
figure
subplot(2,1,1)
plot2d3(n,x);
plot(n,x,'r.')
a=gca();
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x[n]','n')
subplot(2,1,2)
plot2d3(n,round(dft(x,-1)));
plot(n,round(dft(x,-1)),'r.')
```

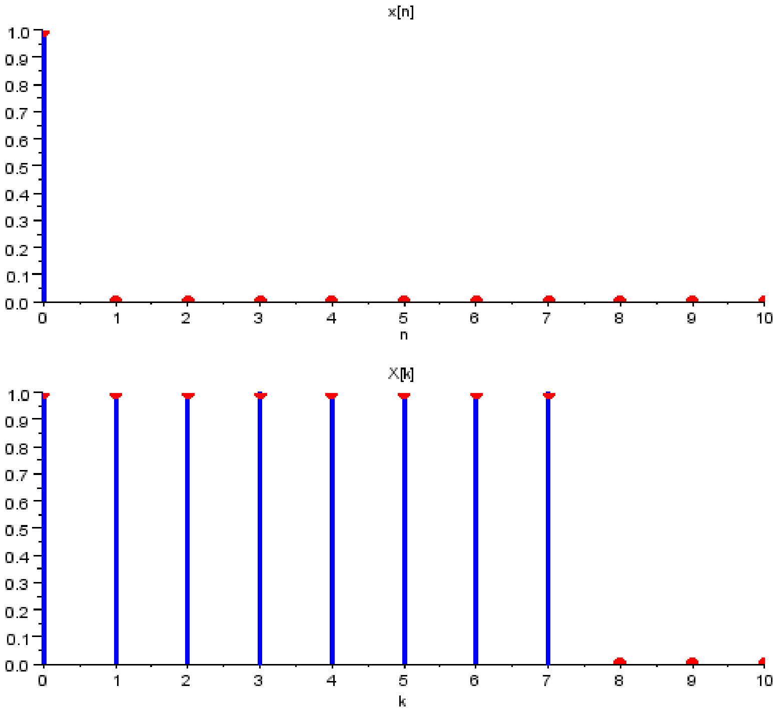


Figure 6.40: N point DFT

```
a=gca();
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X[k]','k')
```

Scilab code Exa 6.50 DFT

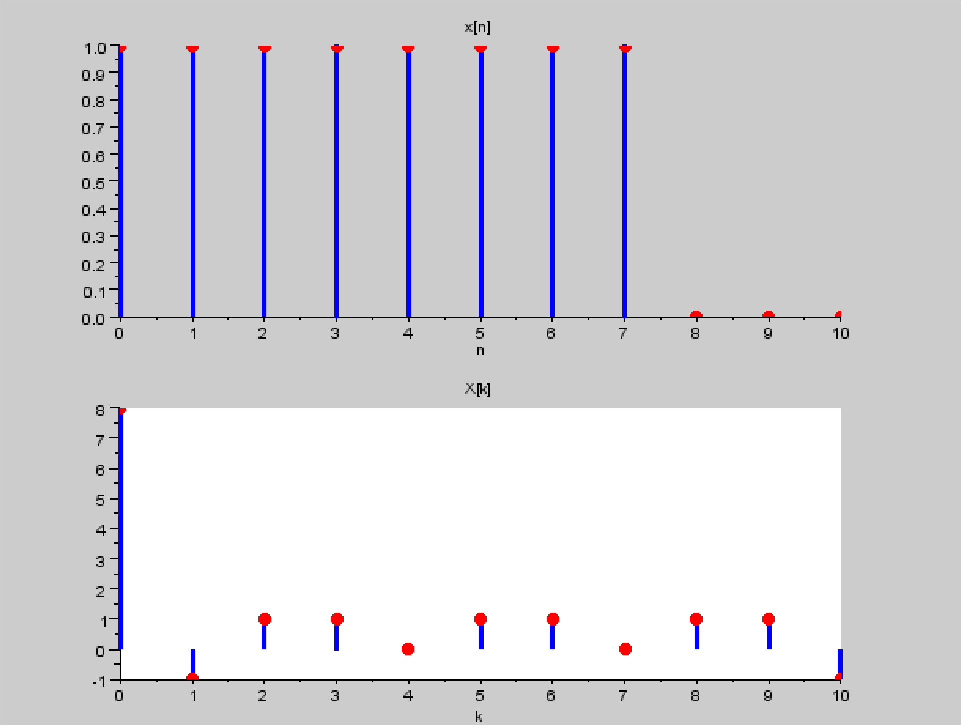


Figure 6.41: N point DFT

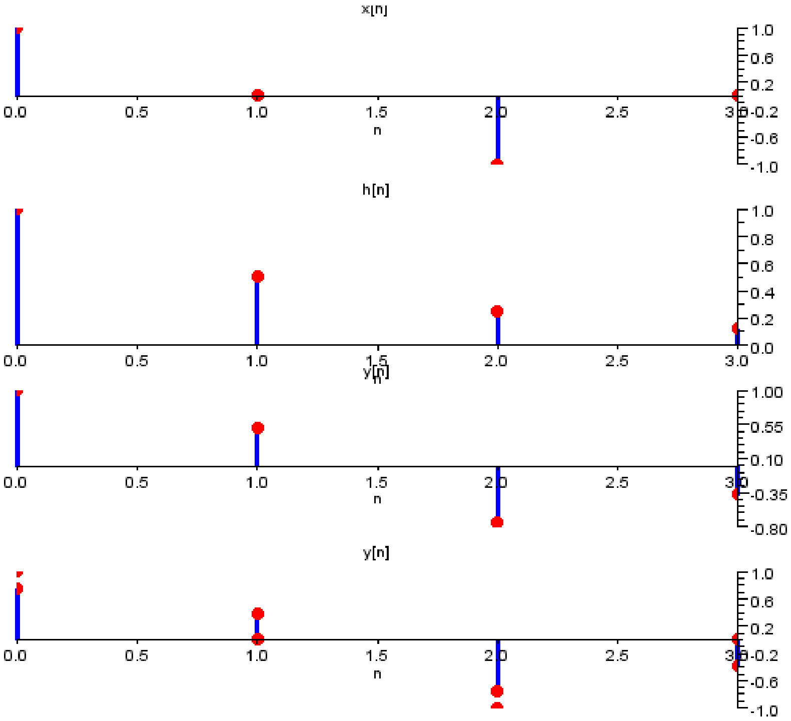


Figure 6.42: DFT

```
close;
clc;
clear;
n=0:3;
x=cos(%pi*n/2);
subplot(4,1,1)
plot2d3(n,x)
xtitle('x[n]','n')
a=gca();
a.x_location="origin";
a.y_location="right";
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
plot(n,x,'r.')
subplot(4,1,2)
h=.5^n;
plot2d3(n,h);
xtitle('h[n]','n')
a=gca();
a.x_location="origin";
a.y_location="right";
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
plot(n,h,'r.')
y=convol(x,h)
subplot(4,1,3)
plot2d3(n,y(1:4));
xtitle('y[n]','n')
plot(n,y(1:4),'r.')
a=gca();
a.x_location="origin";
a.y_location="right";
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
clear  y;
```

```
X=dft(x,-1);
H=dft(h,-1);
Y=H.*X;
y=dft(Y,1);
subplot(4,1,4)
plot2d3(n,y);
xtitle('y[n]','n')
plot(n,y,'r.')
a=gca();
a.x_location="origin";
a.y_location="right";
poly1=a.children.children;
poly1.thickness=3;
poly1.foreground=2;
plot(n,x,'r.')
```

Scilab code Exa 6.55 DFT using matrices

```
x=[0 1 2 3];
X=dft(x,-1);
disp(X,"DFT is X(k)=")
x=dft(X,1);
disp(round(x),"IDFT is x[n]=")
```

Scilab code Exa 6.56 DFT using matrices

```
clc;
N=4;
n=-10:10;
x=[zeros(1,find(n==0)-1) rand(1,-find(n==0)+find(n==
   N)+1) zeros(1,length(n)-find(n==N))];
```

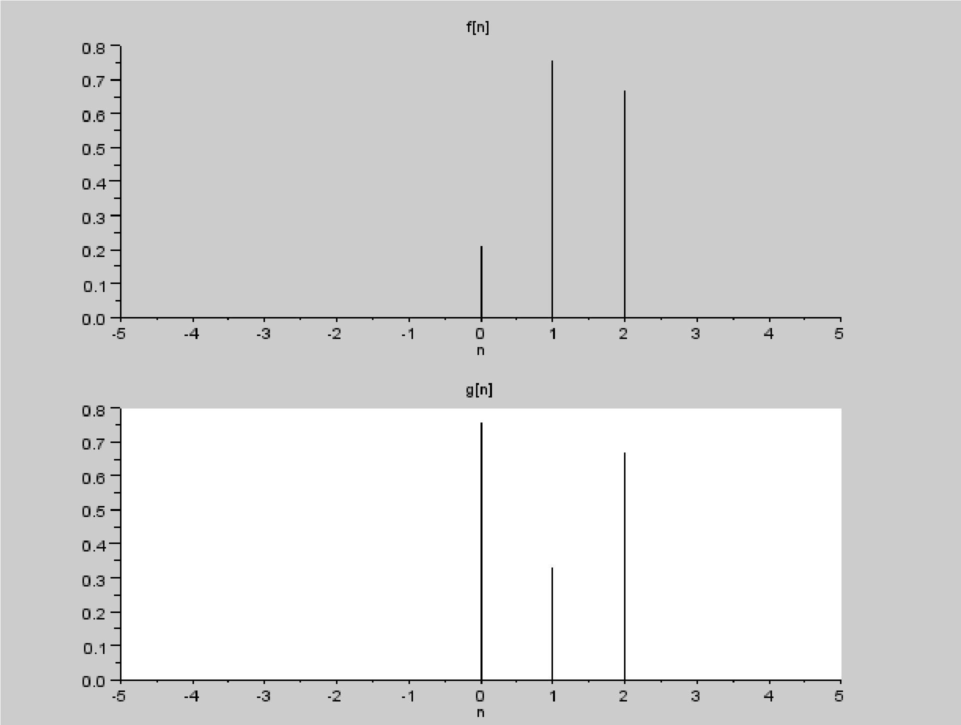


Figure 6.43: DFT using matrices

```
plot2d3(n,x)
nf=-5:0.5:5;
figure
subplot(2,1,1)
plot2d3(ceil(nf),x);
xtitle('f[n]','n')
subplot(2,1,2)
ng=nf-1/2;
plot2d3(ceil(ng),x);
xtitle('g[n]','n')
disp("from the graphs it is visible that f[n]=g[n]=0 outside 0<=n<=N/2-1");
```

Scilab code Exa 6.57 decimation in time FFT algorithm

```
clear;
close;
clc;
x=[1 1 -1 -1 -1 1 1 -1];
N=8;
n=0:7;
k=0:N/2-1;
WkN=exp(-%i*2*%pi*k/N);
f=[];
g=[];
for i=1:length(n)
    if modulo(n(i),2)==0 then
        f=[f x(i)];
    else
        g=[g x(i)];
    end
end
Fk=dft(f,-1);
Gk=dft(g,-1);
//Xk=Fk-WkN*Gk
```

```
Xk=Fk-WkN'.*Gk;
Xk=[Xk; Xk];
disp(round(Xk))
```

Scilab code Exa 6.59 decimation in time FFT algorithm

```
clear;
close;
clc;
x=[1 1 -1 -1 -1 1 1 -1];
N=8;
n=0:7;
p=x(1:4)+x(5:8);
k=0:N/2-1;
WkN=exp(-%i*2*%pi*k/N);
q=[x(1:4)-x(5:8)].*WkN;
Pk=dft(p,-1);
Qk=dft(q,-1);
//X[2k]=P[k]  X[2k+1]=Q[k]
for i=1:length(k)
    if modulo(i,2)==0 then
        Xk(i)=Qk(i/2);
    else
       Xk(i)=Pk(ceil((i)/2));
    end
end
Xk=[Xk;Xk];
disp(round(Xk));
```

Scilab code Exa 6.61 fourier spectrum using DFT

```
clear;
close;
```

```
clc;
T1=10;
wM=100;
N=ceil(wM*T1/%pi);
dw=2*wM/N;
dt=%pi/wM;
WN=exp(-%i*2*%pi/N);
n=0:N-1;
x=%e^-(n*dt);
k=0:10;
Xk=dt*x*exp(-%i*2*%pi*n'*k/N);
disp(Xk(find(k==0)),Xk(find(k==1)),Xk(find(k==10)),"X[0] X[1] X[10] from X[k]")
w=0:dw:10*dw;
Xw=ones(1,length(w))./(%i*w+1);
disp(Xw(find(w==0)),Xw(find(w==dw)),Xw(find(w==10*dw)),"X(0) X(dw) X(10*dw) from X(w)")
disp("it can be seen that X[k] gives a good approximation to X(w)")
```

Chapter 7

state space analysis

Scilab code Exa 7.5 state space representation

```
clc;
clear;
close;
s=%s;
r=2;
L=4;
c=6;
tf=syslin('c',(1/(L*s+r+(1/(c*s)))));
ss=tf2ss(tf)
```

Scilab code Exa 7.7 state equation for discrete system

```
clc;
close
clear;
z=%s;
Hz=syslin('d',(1/(1-(3/4)*z^-1+(1/8)*z^-2)));
ss=tf2ss(Hz);
```

Scilab code Exa 7.8 state equation for discrete time system

```
clc;
close
clear;
z=%s;
tf=syslin('d',((1+(1/2)*z^-1)/(1-(3/4)*z^-1+(1/8)*z
   ^-2)));
ss=tf2ss(tf);
```

Scilab code Exa 7.9 state equation for discrete time system

```
clc;
close
clear;
z=%s;
b0=1;
b1=2;
b2=3;
a1=4;
a2=5;
tf=syslin('d',((b0+b1*z^-1+b2*z^-2)/(1+a1*z^-1+a2*z
   ^-2)));
ss=tf2ss(tf);
```

Scilab code Exa 7.11 state equation for discrete time system

```
clc;
clear;
```

```
close;
z=%s;
Hz=syslin('d',(z/(2*z^2-3*z+1)));
ss=tf2ss(Hz)
disp(ss)
```

Scilab code Exa 7.14 state equation for continuous time system

```
clc;
close
clear;
s=%s;
tf=syslin('c',(1/(s^2+3*s+2)));
ss=tf2ss(tf);
```

Scilab code Exa 7.15 state equation for continuous time system

```
clc;
close
clear;
s=%s;
tf=syslin('c',((4*s+1)/(s^2+3*s+2)));
ss=tf2ss(tf);
disp(ss)
```

Scilab code Exa 7.16 state equation for continuous time system

```
clc;
close
clear;
```

```
s=%s;
b0=1;
b1=2;
b2=3;
b3=4;
a1=5;
a2=6;
a3=7;
tf=syslin('d',((b0*s^3+b1*s^2+b2*s^1+b3)/(s^3+a1*s^2+a2*s+a3)));
ss=tf2ss(tf);
disp(ss)
```

Scilab code Exa 7.18 state equation for continuous time system

```
clc;
close
clear;
s=%s;
tf=syslin('c',((3*s+7)/((s+1)*(s+2)*(s+5))));
ss=tf2ss(tf);
disp(ss)
```

Scilab code Exa 7.20 A power n

```
clc;
syms n;
A=[0 1;-1/8 3/4];
e=spec(A);
b1=(e(2)^n-e(1)^n)/(e(2)-e(1));
b0=e(2)^n-b1*e(2);
An=b0*eye(A)+b1*A;
disp(An,"A^n")
```

Scilab code Exa 7.24 A power n

```
clc;
syms n;
A=[0 1;-1/3 4/3];
e=spec(A);
b1=(e(2)^n-e(1)^n)/(e(2)-e(1));
b0=e(2)^n-b1*e(2);
An=b0*eye(A)+b1*A;
disp(An,"A^n")
```

Scilab code Exa 7.25 A power n

```
clc;
syms n;
A=[2 1;0 2];
e=spec(A);//since we get equal eigen values
b1=n*e(1)^(n-1);
b0=e(1)^n-b1*e(1);
An=b0*eye(A)+b1*A;
disp(An,"A^n")
```

Scilab code Exa 7.26 decomposition of matrix A

```
clear;
clc;
close;
A=[2 1;0 2];
D=[2 0; 0 2];
```

```
N=[0 1;0 0];
disp(N^2,”N*N=”);
p=D*N;
q=N*D;
if (p==q) then
    disp(”D and N commute”)
end
```

Scilab code Exa 7.27 minimal polynomial

```
clc;
syms n;
A=[2 0 0;0 -2 1;0 4 1];
eig=spec(A);
m=poly(eig(1:2),”A”,[”roots”]);
disp(m,”minimal polynomial is”)
e=spec(A);
b1=(e(2)^n-e(1)^n)/(e(2)-e(1));
b0=e(2)^n-b1*e(2);
An=b0*eye(A)+b1*A;
disp(An,”A^n”)
```

Scilab code Exa 7.29 step response

```
clc;
A=[0 1;-1/8 3/4];
B=[0;1];
C=[-1/8 3/4];
D=[1];
Hz=ss2tf(syslin(’d’,A,B,C,D));
disp(Hz,”H(z)=”);
z = %z;
```

```
syms n z1; //To find out Inverse z transform z must be linear z = z1
X =z ^2 /((z -(1/2) )*(z -(1/4) ))
X1 = denom (X);
zp = roots (X1);
X1 = z1 ^2 /(( z1 -(1/4) )*(z1 -(1/2) ))
F1 = X1 *( z1 ^(n -1) )*(z1 -zp (1) );
F2 = X1 *( z1 ^(n -1) )*(z1 -zp (2) );
h1 = limit (F1 ,z1 ,zp (1) );
disp (h1 , 'h1[n]= ')
h2 = limit (F2 ,z1 ,zp (2) );
disp (h2 , 'h2[n]= ')
h = h1+h2;
disp (h, 'h[n]= ')
n=0:10;
x=[zeros(1,find(n==0)-1) ones(1,length(n)-find(n==0)+1)];
hn=2^(1-n)-4^-n;
y=convol(x,hn);
plot2d3(n,y(1:length(n)))
figure
plot2d3(n,8/3-2^(1-n)+4^-n)
disp("hence from the figure we can say y[n]=8/3-2^(1-n)+4^-n")
```

Scilab code Exa 7.30 impulse response

```
clc;
A=[0 1;-1/8 3/4];
B=[0;1];
C=[-1/8 3/4];
D=[1];
Hz=ss2tf(syslin('d',A,B,C,D));
disp(Hz,"H(z)=");
z = %z;
```

```
syms n z1; //To find out Inverse z transform must
//be linear z = z1
X =z ^2 /((z -(1/2) )*(z -(1/4) ))
X1 = denom (X);
zp = roots (X1);
X1 = z1 ^2 /(( z1 -(1/4) )*(z1 -(1/2) ))
F1 = X1 *( z1 ^(n -1) )*(z1 -zp (1) );
F2 = X1 *( z1 ^(n -1) )*(z1 -zp (2) );
h1 = limit (F1 ,z1 ,zp (1) );
disp (h1 , ' h1 [ n]= ' )
h2 = limit (F2 ,z1 ,zp (2) );
disp (h2 , ' h2 [ n]= ' )
h = h1+h2;
disp ('for n>=0',h, ' h [ n]= ' )
```

Scilab code Exa 7.31 difference equation

```
clc;
A=[0 1;-1/3 4/3];
B=[0;1/3];
C=[-1/3 4/3];
D=[1/3];
Hz=ss2tf(syslin('d',A,B,C,D));
disp(Hz,"H(z)=");
z = %z;
syms n z1;
X =z ^2 /(1-4*z+3*z^2)
X1 = denom (X);
zp = roots (X1);
X1 = z1 ^2 /((z1-1)*(z1-1/3))
F1 = X1 *( z1 ^(n -1) )*(z1 -1/3 );
F2 = X1 *( z1 ^(n -1) )*(z1 -1 );
h1 = limit (F1 ,z1 ,zp (1) );
disp (h1 , ' h1 [ n]= ' )
```

```
h2 = limit (F2 ,z1 ,zp (2) );
disp (h2 , ' h2 [ n]= ' )
h = h1+h2;
disp ('for n>=0',h, ' h [ n]= ' )
n=0:10;
x=2^-n.*[zeros(1,find(n==0)-1) ones(1,length(n)-find(n==0)+1)];
hn= 3/2-(3^-n)/2;
y=convol(x,hn);
plot2d3(n,y(1:length(n)))
figure
plot2d3(n,(3/2)-2^-n+(3^-n)/2)
disp("hence from the figure we can say y[n]=(3/2)-2^-n+(3^-n)/2")
```

Scilab code Exa 7.32 stability

```
clc;
A=[0 3/2;-1/2 2];
B=[0.71;-0.71];
C=[0.71 -0.71];
D=[0];
Hz=ss2tf(syslin('d',A,B,C,D));
disp(Hz,"H(z)=");
disp(" it has only one pole at z=1/2 and it lies inside the unit circle          hence the system is BIBO stable")
```

Scilab code Exa 7.35 observability and controllability

```
clc;
A=[0 1 ;-1/8 3/4];
B=[0;1];
```

```
C=[-1/8 3/4];
D=[1];
Mc=[B A*B];
if (det(Mc)<>0) then
    disp("and is controllable",rank(Mc),"Mc has a rank of")
else
    disp("and is uncontrollable",rank(Mc),"Mc has a rank of");
end
Mo=[C;C*A];
if (det(Mo)<>0) then
    disp("and is observable",rank(Mo),"Mo has a rank of")
else
    disp("and is unobservable",rank(Mo),"Mo has a rank of");
end
Hz=ss2tf(syslin('d',A,B,C,D));
disp(Hz,"H(z)=");
```

Scilab code Exa 7.36 finding vector x

```
clc;
A=[0 1;-1/8 3/4];
B=[0;1];
q2=[0 ;0];
q0=[0;1];
//we know q2=A^2*q0+[B A*B]*X
X=inv([B A*B])*[q2-A^2*q0];
disp(X,"therefore [x(1);x(0)]=")
```

Scilab code Exa 7.37 finding vector y

```
clc;
c=[-1/8 3/4];
A=[0 1;-1/8 3/4];
y=[1;0];
//we know y=[c;c*A]*q0
Q0=inv([c;c*A])*y;
disp(Q0,"therefore [q1(0);q2(0)]=")
```

Scilab code Exa 7.38 observability and controllability

```
clc;
A=[0 3/2;-1/2 2];
B=[1;0];
C=[1 -1];
D=[0];
Mc=[B A*B];
if (det(Mc)<>0) then
    disp("and is controllable",rank(Mc),"Mc has a rank of")
else
    disp("and is uncontrollable",rank(Mc),"Mc has a rank of");
end
Mo=[C;C*A];
if (det(Mo)<>0) then
    disp("and is observable",rank(Mo),"Mo has a rank of")
else
    disp("and is unobservable",rank(Mo),"Mo has a rank of");
end
```

Scilab code Exa 7.39 e power At

```
clc;
A=[0 1;-6 -5];
syms t
//we know %e^(at)=b0*I+b1*A;
e=spec(A);//eigen values
b0=e(1)*%e^(e(2)*t)-e(2)*%e^(e(1)*t);
b1=%e^(e(1)*t)-%e^(e(2)*t);
eAt=b0*eye(A)+b1*A;
disp(eAt,”e^(At)”)
```

Scilab code Exa 7.43 e power At

```
clc;
A=[-2 1;1 -2];
syms t
//we know %e^(at)=b0*I+b1*A;
e=spec(A);//eigen values
b0=e(1)*%e^(e(2)*t)-e(2)*%e^(e(1)*t);
b1=%e^(e(1)*t)-%e^(e(2)*t);
eAt=b0*eye(A)+b1*A;
disp(eAt,”e^(At)”)
```

Scilab code Exa 7.44 e power At

```
clc;
syms t
A=[0 -2 1;0 0 3;0 0 0];
z=A*A*A;
if z==0 then
    disp(”A is a nilpotent of index 3”);
else
    disp(”A is not nilpotent”)
end
```

```
//we know %e^(at)=b0*I+b1*A+b2*A^2;
e=spec(A);//eigen values
b2=t^2*%e^(e(3)*t);
b1=t*%e^(e(2)*t)-b2*2*e(2);
b0=%e^(e(1)*t)-b1*e(1)-b2*e(1)^2;
eAt=b0*eye(A)+b1*A+b2*A^2;
disp(eAt,"e^(At)")
```

Scilab code Exa 7.45 e power At

```
clc;
syms t
A=[0 -2 1;0 0 3;0 0 0];
//we know %e^(at)=b0*I+b1*A+b2*A^2;
e=spec(A);//eigen values
b2=t^2*%e^(e(3)*t);
b1=t*%e^(e(2)*t)-b2*2*e(2);
b0=%e^(e(1)*t)-b1*e(1)-b2*e(1)^2;
eAt=b0*eye(A)+b1*A+b2*A^2;
disp(eAt,"e^(At)")
```

Scilab code Exa 7.47 nilpotent matrix

```
clc;
A=[2 1 0;0 2 1;0 0 2];
V=[2 0 0;0 2 0; 0 0 2];
N=[0 1 0;0 0 1;0 0 0];
z=N*N*N;
if z==0 then
    disp("N is a nilpotent of index 3");
else
    disp("N is not nilpotent")
end
```

```
if V*N==N*V then
    disp(”V and N commute”)
else
    disp(”V and N dont commute”)
end
syms t
//e^(A*t)=e^(v*t)*e^(N*t)
e=spec(V);//eigen values
b2=t^2*%e^(e(3)*t);
b1=t*%e^(e(2)*t)-b2*2*e(2);
b0=%e^(e(1)*t)-b1*e(1)-b2*e(1)^2;
eVt=b0*eye(V)+b1*V+b2*V^2;
disp(eVt,”e^(Vt)”)
e=spec(N);//eigen values
b2=t^2*%e^(e(3)*t);
b1=t*%e^(e(2)*t)-b2*2*e(2);
b0=%e^(e(1)*t)-b1*e(1)-b2*e(1)^2;
eNt=b0*eye(N)+b1*N+b2*N^2;
disp(eNt,”e^(Nt)”)
eAt=eVt*eNt;
disp(eAt,”e^(A*t)=”)
```

Scilab code Exa 7.48 second order ODE

```
clc;
A=[0 1;-6 -5];
b=[0;1];
c=[1 0];
q0=[2;1];
syms t T
//y=c*e^(A*t)*q0+integ(c*e^A*(t-T)*b*x,0,t)
e=spec(A);//eigen values
b0=e(1)*%e^(e(2)*t)-e(2)*%e^(e(1)*t);
b1=%e^(e(1)*t)-%e^(e(2)*t);
eAt=b0*eye(A)+b1*A;
```

```
b0=e(1)*%e^(e(2)*(t-T))-e(2)*%e^(e(1)*(t-T));
b1=%e^(e(1)*(t-T))-%e^(e(2)*(t-T));
eAtT=b0*eye(A)+b1*A;
disp(eAt,"e^(At)");
ceAtq0=c*eAt*q0;
ceAtTq0=c*eAtT*q0;
y=ceAtq0+integ(ceAtTq0*%e^-T,T);
y0=limit(y,T,0);
yt=limit(y,T,t);
Y=yt-y0;
disp(Y,"y(t)=")
```

Scilab code Exa 7.49 RC circuit response

```
clc;
A=[-2 1;1 -2];
q0=[.5;1];
//Q=e^(A*t)*q0+integ(e^A*(t-T)*b*x,T,t0,t)
X=0;
//therefore integ term is zero
e=spec(A);//eigen values
b0=e(1)*%e^(e(2)*t)-e(2)*%e^(e(1)*t);
b1=%e^(e(1)*t)-%e^(e(2)*t);
eAt=b0*eye(A)+b1*A;
Q=eAt*q0;
disp(Q,"thus [vc1(t)  vc2(t)]=");
```

Scilab code Exa 7.50 stability

```
clc;
A=[0 1;2 1];
b=[1;-1];
c=[1 -1];
```

```
d=[0];
e=spec(A);
if real(e(1))>0 |real(e(2))>0 then
    disp("the sytem is not asymptotically stable")
else
    disp("the system is asymptotically stable")
end
Hs=ss2tf(syslin('c',A,b,c,d));
disp(Hs,"H(s)=")
disp("there is only one pole and it is located at -1 which is in the LHP        hence the system is BIBO stable")
```

Scilab code Exa 7.53 observability and controllability

```
clc;
A=[0 1;2 1];
B=[1;-1];
C=[1 -1];
D=[0];
Mc=[B A*B];
if (det(Mc)<>0) then
    disp("and is controllable",rank(Mc),"Mc has a rank of")
else
    disp("and is not controllable",rank(Mc),"Mc has a rank of");
end
Mo=[C;C*A];
if (det(Mo)<>0) then
    disp("and is observable",rank(Mo),"Mo has a rank of")
else
    disp("and is unobservable",rank(Mo),"Mo has a rank of");
```

```
end
```

Scilab code Exa 7.54 observability and controllability

```
clc;
A=[1  2;0 3];
B=[1;1];
C=[1 -1];
D=[0];
Mc=[B A*B];
if (det(Mc)<>0) then
    disp("and is controllable",rank(Mc),"Mc has a rank of")
else
    disp("and is uncontrollable",rank(Mc),"Mc has a rank of");
end
Mo=[C;C*A];
if (det(Mo)<>0) then
    disp("and is observable",rank(Mo),"Mo has a rank of")
else
    disp("and is unobservable",rank(Mo),"Mo has a rank of");
end
Hs=ss2tf(syslin('c',A,B,C,D));
disp(Hs,"H(s)=")
```