

Scilab Textbook Companion for  
Schaums Outlines Signals And Systems  
by H. P. Hsu<sup>1</sup>

Created by  
Puneetha Ramachandra  
B.Tech (pursuing)  
Electronics Engineering  
NIT, Surathkal  
College Teacher  
NA

Cross-Checked by  
K. Suryanarayan, IITB

August 11, 2013

<sup>1</sup>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	4
1 Signals and Systems	15
2 Linear Time Invariant Systems	63
3 Laplace transform and continuous time LTI systems	103
4 The z transform and discrete time LTI systems	118
5 Fourier analysis of continuous time system and signals	142
6 Fourier analysis of discrete time system and signals	247
7 state space analysis	318

# List of Scilab Codes

Exa 1.1	shifting and scaling of continuous time signal . . . . .	15
Exa 1.2	shifting and scaling of discrete time signal . . . . .	16
Exa 1.3	sampling of continuous time signal . . . . .	20
Exa 1.4	discrete time signal . . . . .	23
Exa 1.5.a	even and odd components . . . . .	26
Exa 1.5.b	even and odd components . . . . .	31
Exa 1.5.c	even and odd components . . . . .	34
Exa 1.5.d	even and odd components . . . . .	36
Exa 1.6	even and odd components . . . . .	36
Exa 1.9	periodicity of exponential signal . . . . .	37
Exa 1.10	periodicity of sinusoidal signal . . . . .	38
Exa 1.11	periodicity of exponential sequence . . . . .	40
Exa 1.16	fundamental period . . . . .	40
Exa 1.21	unit step signal . . . . .	42
Exa 1.22	continuous time signal . . . . .	43
Exa 1.23	discrete time signal . . . . .	47
Exa 1.31	first derivative of the signals . . . . .	52
Exa 1.35	linearity . . . . .	53
Exa 1.36	memoryless causal stable system . . . . .	53
Exa 1.38	memoryless causal time invariant system . . . . .	57
Exa 1.39	time invariancy . . . . .	58
Exa 1.41	linearity . . . . .	59
Exa 2.4	output response . . . . .	63
Exa 2.5	output response . . . . .	64
Exa 2.6	convolution . . . . .	67
Exa 2.7.a	convolution of two rectangular pulse . . . . .	69
Exa 2.7.b	convolution of two rectangular pulse . . . . .	72
Exa 2.7.c	convolution of two rectangular pulse . . . . .	74

Exa 2.8	periodic convolution . . . . .	77
Exa 2.9	output response . . . . .	78
Exa 2.10	output response . . . . .	85
Exa 2.14.a	cascaded system . . . . .	86
Exa 2.14.b	BIBO stability . . . . .	89
Exa 2.15	eigenfunction of the system . . . . .	89
Exa 2.16	eigen value of the system . . . . .	89
Exa 2.28	output response of a discrete time system . . . . .	91
Exa 2.29.a	convolution of discrete signals . . . . .	91
Exa 2.29.b	convolution of discrete signals . . . . .	95
Exa 2.30	convolution of discrete signals . . . . .	96
Exa 2.34	output response without using convolution . . . . .	98
Exa 2.36	causality . . . . .	100
Exa 2.38	BIBO stability and causality . . . . .	101
Exa 3.1.a	laplace transform . . . . .	103
Exa 3.1.b	laplace transform . . . . .	103
Exa 3.3	laplace transform . . . . .	103
Exa 3.5	pole zero plot . . . . .	104
Exa 3.6	ROC and pole zero plot . . . . .	104
Exa 3.13	derivative and shifting property . . . . .	107
Exa 3.16	inverse laplace transform . . . . .	109
Exa 3.17	inverse laplace transform . . . . .	109
Exa 3.18	inverse laplace transform . . . . .	110
Exa 3.19	inverse laplace transform . . . . .	110
Exa 3.20	inverse laplace transform by partial fractions . . . . .	111
Exa 3.21	inverse laplace transform of time shifted signal . . . . .	111
Exa 3.22	differentiation in s domain . . . . .	111
Exa 3.24	output response . . . . .	112
Exa 3.25	impulse response . . . . .	112
Exa 3.27	cascaded system transfer function . . . . .	112
Exa 3.28	first order differential equation . . . . .	113
Exa 3.29	impulse response . . . . .	113
Exa 3.30	causality and stability . . . . .	113
Exa 3.34	bilateral laplace transform . . . . .	114
Exa 3.36	unilateral laplace transform . . . . .	114
Exa 3.37	unilateral laplace transform method . . . . .	115
Exa 3.38	second order ODE . . . . .	115
Exa 3.39	RC circuit . . . . .	115

Exa 3.40	RC circuit response . . . . .	116
Exa 3.41	RLC circuit . . . . .	116
Exa 3.42	circuit analysis . . . . .	116
Exa 4.1.a	z transform . . . . .	118
Exa 4.1.b	z transform . . . . .	118
Exa 4.3	finite sequence z transform . . . . .	119
Exa 4.4	pole zero plot . . . . .	120
Exa 4.6.a	z transform and pole zero plot . . . . .	120
Exa 4.6.b	z transform and pole zero plot . . . . .	121
Exa 4.6.c	z transform and pole zero plot . . . . .	123
Exa 4.7	pole zero plot . . . . .	124
Exa 4.10	z transform . . . . .	128
Exa 4.12	differentiation property . . . . .	128
Exa 4.15	inverse z transform . . . . .	128
Exa 4.16	inverse z transform . . . . .	129
Exa 4.18.a	power series expansion technique . . . . .	130
Exa 4.18.b	power series expansion technique . . . . .	130
Exa 4.19	inverse z transform . . . . .	130
Exa 4.20	inverse z transform . . . . .	131
Exa 4.21	inverse z transform . . . . .	132
Exa 4.22	inverse z transform . . . . .	132
Exa 4.23	inverse z transform . . . . .	133
Exa 4.25	inverse z transform . . . . .	134
Exa 4.26	output response . . . . .	135
Exa 4.27	output response . . . . .	135
Exa 4.28	impulse response . . . . .	136
Exa 4.29	impulse response and output . . . . .	136
Exa 4.31	impulse response . . . . .	137
Exa 4.32	impulse and step response . . . . .	138
Exa 4.35	unilateral z transform . . . . .	139
Exa 4.37	unilateral z transform method . . . . .	139
Exa 4.38	difference equation . . . . .	140
Exa 5.4.a	fourier series representation . . . . .	142
Exa 5.4.b	fourier series representation . . . . .	142
Exa 5.4.c	fourier series representation . . . . .	143
Exa 5.4.d	fourier series representation . . . . .	143
Exa 5.4.e	fourier series representation . . . . .	144
Exa 5.5	fourier series of a periodic square wave . . . . .	145

Exa 5.6	fourier series of a periodic square wave . . . . .	146
Exa 5.7	fourier series of a periodic square wave . . . . .	149
Exa 5.8	fourier series of a periodic impulse train . . . . .	150
Exa 5.9	differentiation property of fourier series . . . . .	152
Exa 5.10	differentiation property of fourier series . . . . .	154
Exa 5.11	magnitude spectra of a periodic square wave . . . . .	157
Exa 5.19	fourier transform of a rectangular pulse . . . . .	161
Exa 5.20	fourier transform of a sinc function . . . . .	164
Exa 5.21	fourier transform . . . . .	167
Exa 5.22	fourier transform . . . . .	169
Exa 5.23.a	fourier transform . . . . .	171
Exa 5.23.b	fourier transform . . . . .	176
Exa 5.23.c	fourier transform . . . . .	176
Exa 5.23.d	fourier transform . . . . .	179
Exa 5.23.e	fourier transform . . . . .	182
Exa 5.25	fourier transform of a periodic impulse train . . . . .	185
Exa 5.27	inverse fourier transform . . . . .	190
Exa 5.29	fourier transform of a signum function . . . . .	191
Exa 5.30	fourier transform of a step signal . . . . .	194
Exa 5.32	inverse fourier transform using convolution . . . . .	197
Exa 5.34	integration property . . . . .	202
Exa 5.40.a	fourier transform . . . . .	202
Exa 5.40.b	fourier transform . . . . .	204
Exa 5.42	fourier transform of a exponential signal . . . . .	206
Exa 5.43	fourier transform of a gaussian pulse . . . . .	207
Exa 5.44	impulse response using fourier transform . . . . .	210
Exa 5.45.a	output response using fourier transform . . . . .	213
Exa 5.45.b	output response using fourier transform . . . . .	216
Exa 5.46	harmonics in the output response . . . . .	219
Exa 5.47.a	bode plot . . . . .	223
Exa 5.47.b	bode plot . . . . .	225
Exa 5.47.c	bode plot . . . . .	227
Exa 5.48	impulse response of a phase shifter . . . . .	227
Exa 5.52	output of a ideal LPF . . . . .	228
Exa 5.53	output of a ideal LPF . . . . .	232
Exa 5.54	ideal low pass filter . . . . .	235
Exa 5.55	equivalent bandwidth . . . . .	239
Exa 5.58	fourier spectrum . . . . .	240



Exa 6.3	fourier coefficients . . . . .	247
Exa 6.4	fourier coefficients of a periodic sequence . . . . .	248
Exa 6.5	fourier coefficients . . . . .	249
Exa 6.6	discrete fourier series representation . . . . .	252
Exa 6.11	fourier transform . . . . .	255
Exa 6.12	fourier transform of a rectangular pulse . . . . .	259
Exa 6.14	fourier transform . . . . .	261
Exa 6.15	inverse fourier transform of a rectangular pulse . . . . .	264
Exa 6.17	inverse fourier transform of a impulse signal . . . . .	265
Exa 6.18	fourier transform of constant signal . . . . .	268
Exa 6.19	fourier transform of a sinusoidal sequence . . . . .	268
Exa 6.22	fourier transform . . . . .	270
Exa 6.25	inverse fourier transform using convolution . . . . .	272
Exa 6.28	frequency response . . . . .	275
Exa 6.31	frequency response . . . . .	276
Exa 6.32	frequency response . . . . .	279
Exa 6.33	output response . . . . .	280
Exa 6.34	magnitude and phase response . . . . .	283
Exa 6.35	frequency response . . . . .	287
Exa 6.36	discrete time low pass filter . . . . .	289
Exa 6.38	conversion of LPF to HPF . . . . .	293
Exa 6.40	impulse response os a FIR filter . . . . .	294
Exa 6.41	three point moving average discrete time filter . . . . .	296
Exa 6.42	causal discrete time FIR filter . . . . .	299
Exa 6.43	Rc low pass filter . . . . .	302
Exa 6.44	frequency response . . . . .	305
Exa 6.45	bilinear transformation . . . . .	307
Exa 6.49	N point DFT . . . . .	307
Exa 6.50	DFT . . . . .	309
Exa 6.55	DFT using matrices . . . . .	313
Exa 6.56	DFT using matrices . . . . .	313
Exa 6.57	decimation in time FFT algorithm . . . . .	315
Exa 6.59	decimation in time FFT algorithm . . . . .	316
Exa 6.61	fourier spectrum using DFT . . . . .	316
Exa 7.5	state space representation . . . . .	318
Exa 7.7	state equation for discrete system . . . . .	318
Exa 7.8	state equation for discrete time system . . . . .	319
Exa 7.9	state equation for discrete time system . . . . .	319

Exa 7.11	state equation for discrete time system . . . . .	319
Exa 7.14	state equation for continuous time system . . . . .	320
Exa 7.15	state equation for continuous time system . . . . .	320
Exa 7.16	state equation for continuous time system . . . . .	320
Exa 7.18	state equation for continuous time system . . . . .	321
Exa 7.20	A power n . . . . .	321
Exa 7.24	A power n . . . . .	322
Exa 7.25	A power n . . . . .	322
Exa 7.26	decomposition of matrix A . . . . .	322
Exa 7.27	minimal polynomial . . . . .	323
Exa 7.29	step response . . . . .	323
Exa 7.30	impulse response . . . . .	324
Exa 7.31	difference equation . . . . .	325
Exa 7.32	stability . . . . .	326
Exa 7.35	observability and controllability . . . . .	326
Exa 7.36	finding vector x . . . . .	327
Exa 7.37	finding vector y . . . . .	327
Exa 7.38	observability and controllability . . . . .	328
Exa 7.39	e power At . . . . .	328
Exa 7.43	e power At . . . . .	329
Exa 7.44	e power At . . . . .	329
Exa 7.45	e power At . . . . .	330
Exa 7.47	nilpotent matrix . . . . .	330
Exa 7.48	second order ODE . . . . .	331
Exa 7.49	RC circuit response . . . . .	332
Exa 7.50	stability . . . . .	332
Exa 7.53	observability and controllability . . . . .	333
Exa 7.54	observability and controllability . . . . .	334

# List of Figures

1.1	shifting and scaling of continuous time signal . . . . .	17
1.2	shifting and scaling of continuous time signal . . . . .	18
1.3	shifting and scaling of discrete time signal . . . . .	21
1.4	shifting and scaling of discrete time signal . . . . .	22
1.5	sampling of continuous time signal . . . . .	24
1.6	sampling of continuous time signal . . . . .	25
1.7	discrete time signal . . . . .	27
1.8	discrete time signal . . . . .	28
1.9	even and odd components . . . . .	29
1.10	even and odd components . . . . .	32
1.11	even and odd components . . . . .	33
1.12	even and odd components . . . . .	35
1.13	even and odd components . . . . .	38
1.14	even and odd components . . . . .	39
1.15	unit step signal . . . . .	44
1.16	unit step signal . . . . .	45
1.17	continuous time signal . . . . .	48
1.18	continuous time signal . . . . .	49
1.19	discrete time signal . . . . .	50
1.20	first derivative of the signals . . . . .	54
1.21	first derivative of the signals . . . . .	55
1.22	linearity . . . . .	61
1.23	linearity . . . . .	62
2.1	output response . . . . .	65
2.2	output response . . . . .	66
2.3	convolution . . . . .	68
2.4	convolution of two rectangular pulse . . . . .	70
2.5	convolution of two rectangular pulse . . . . .	73

2.6	convolution of two rectangular pulse . . . . .	75
2.7	periodic convolution . . . . .	79
2.8	periodic convolution . . . . .	80
2.9	output response . . . . .	82
2.10	output response . . . . .	83
2.11	output response . . . . .	84
2.12	cascaded system . . . . .	87
2.13	output response of a discrete time system . . . . .	90
2.14	convolution of discrete signals . . . . .	93
2.15	convolution of discrete signals . . . . .	94
2.16	convolution of discrete signals . . . . .	97
2.17	output response without using convolution . . . . .	99
3.1	pole zero plot . . . . .	105
3.2	pole zero plot . . . . .	106
3.3	ROC and pole zero plot . . . . .	107
3.4	ROC and pole zero plot . . . . .	108
4.1	pole zero plot . . . . .	119
4.2	z transform and pole zero plot . . . . .	121
4.3	z transform and pole zero plot . . . . .	122
4.4	pole zero plot . . . . .	126
4.5	pole zero plot . . . . .	127
5.1	fourier series of a periodic square wave . . . . .	145
5.2	fourier series of a periodic square wave . . . . .	147
5.3	fourier series of a periodic square wave . . . . .	149
5.4	fourier series of a periodic impulse train . . . . .	151
5.5	differentiation property of fourier series . . . . .	154
5.6	differentiation property of fourier series . . . . .	155
5.7	differentiation property of fourier series . . . . .	157
5.8	differentiation property of fourier series . . . . .	158
5.9	magnitude spectra of a periodic square wave . . . . .	162
5.10	magnitude spectra of a periodic square wave . . . . .	163
5.11	fourier transform of a rectangular pulse . . . . .	164
5.12	fourier transform of a rectangular pulse . . . . .	165
5.13	fourier transform of a sinc function . . . . .	167
5.14	fourier transform of a sinc function . . . . .	168

5.15	fourier transform	169
5.16	fourier transform	170
5.17	fourier transform	172
5.18	fourier transform	173
5.19	fourier transform	174
5.20	fourier transform	175
5.21	fourier transform	177
5.22	fourier transform	178
5.23	fourier transform	180
5.24	fourier transform	181
5.25	fourier transform	183
5.26	fourier transform	184
5.27	fourier transform	185
5.28	fourier transform	186
5.29	fourier transform of a periodic impulse train	188
5.30	fourier transform of a periodic impulse train	189
5.31	inverse fourier transform	191
5.32	inverse fourier transform	192
5.33	fourier transform of a signum function	194
5.34	fourier transform of a signum function	195
5.35	fourier transform of a step signal	198
5.36	fourier transform of a step signal	199
5.37	inverse fourier transform using convolution	200
5.38	integration property	201
5.39	fourier transform	203
5.40	fourier transform	205
5.41	fourier transform	206
5.42	fourier transform of an exponential signal	208
5.43	fourier transform of an exponential signal	209
5.44	fourier transform of a gaussian pulse	211
5.45	fourier transform of a gaussian pulse	212
5.46	impulse response using fourier transform	214
5.47	impulse response using fourier transform	215
5.48	output response using fourier transform	217
5.49	output response using fourier transform	218
5.50	output response using fourier transform	220
5.51	output response using fourier transform	221
5.52	harmonics in the output response	222

5.53	bode plot	224
5.54	bode plot	225
5.55	bode plot	226
5.56	impulse response of a phase shifter	229
5.57	impulse response of a phase shifter	230
5.58	output of a ideal LPF	232
5.59	output of a ideal LPF	233
5.60	output of a ideal LPF	236
5.61	output of a ideal LPF	237
5.62	ideal low pass filter	238
5.63	equivalent bandwidth	239
5.64	fourier spectrum	245
5.65	fourier spectrum	246
6.1	fourier coefficients	248
6.2	fourier coefficients of a periodic sequence	250
6.3	fourier coefficients of a periodic sequence	251
6.4	fourier coefficients	252
6.5	fourier coefficients	253
6.6	discrete fourier series representation	256
6.7	discrete fourier series representation	257
6.8	fourier transform	258
6.9	fourier transform of a rectangular pulse	260
6.10	fourier transform	262
6.11	fourier transform	263
6.12	inverse fourier transform of a rectangular pulse	264
6.13	inverse fourier transform of a impulse signal	266
6.14	fourier transform of constant signal	267
6.15	fourier transform of a sinusoidal sequence	269
6.16	fourier transform	272
6.17	fourier transform	273
6.18	inverse fourier transform using convolution	274
6.19	frequency response	276
6.20	frequency response	277
6.21	frequency response	278
6.22	frequency response	281
6.23	frequency response	282
6.24	output response	284

6.25	output response . . . . .	285
6.26	magnitude and phase response . . . . .	287
6.27	magnitude and phase response . . . . .	288
6.28	frequency response . . . . .	290
6.29	frequency response . . . . .	291
6.30	discrete time low pass filter . . . . .	292
6.31	conversion of LPF to HPF . . . . .	293
6.32	impulse response os a FIR filter . . . . .	297
6.33	impulse response os a FIR filter . . . . .	298
6.34	three point moving average discrete time filter . . . . .	300
6.35	three point moving average discrete time filter . . . . .	301
6.36	causal discrete time FIR filter . . . . .	302
6.37	causal discrete time FIR filter . . . . .	303
6.38	Rc low pass filter . . . . .	304
6.39	bilinear transformation . . . . .	306
6.40	N point DFT . . . . .	309
6.41	N point DFT . . . . .	310
6.42	DFT . . . . .	311
6.43	DFT using matrices . . . . .	314

# Chapter 1

## Signals and Systems

Scilab code Exa 1.1 shifting and scaling of continuous time signal

```
1 //shifting and scaling
2 //example 1.1
3 clear all;
4 clc;
5 close;
6 t = 0:1/100:4;
7 for i = 1:length(t)
8     x(i) = (3/4)*t(i) ;
9 end
10 for i = length(t)+1:2*length(t)
11 x(i) = 0;
12 end
13 figure
14 a=gca();
15 t1=0:1/100:8;
16 plot(t1,x(1:$-1))
17 xtitle('x(t)')
18 figure
19 a=gca();
20 t2=t1+2;
21 plot(t2,x(1:$-1))
```



```

22 xtitle('x(t-2)')
23 a.y_location='origin'
24 figure
25 a=gca();
26 t3=0:1/200:4;
27 plot(t3,x(1:$-1))
28 xtitle('x(2t)')
29 figure
30 a=gca();
31 t4=0:1/50:16;
32 plot(t4,x(1:$-1))
33 xtitle('x(t/2)')
34 figure
35 a=gca();
36 t5=-8:1/100:0;
37 plot(t5,x($:-1:2))
38 xtitle('x(-t)')
39 a.y_location = "right";

```

---

### Scilab code Exa 1.2 shifting and scaling of discrete time signal

```

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

```

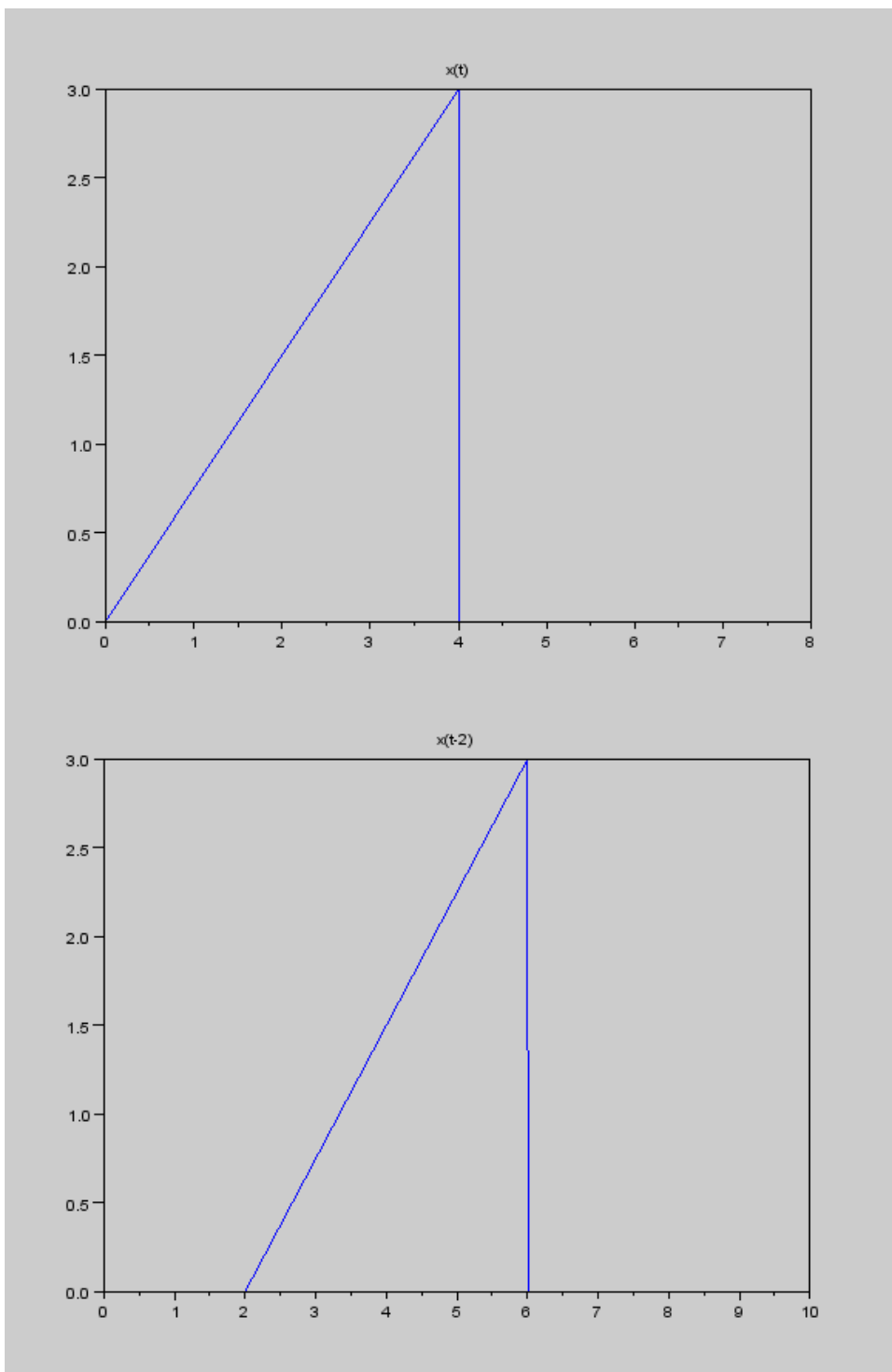
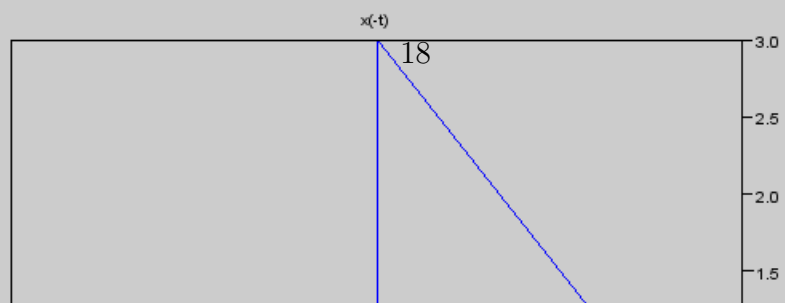
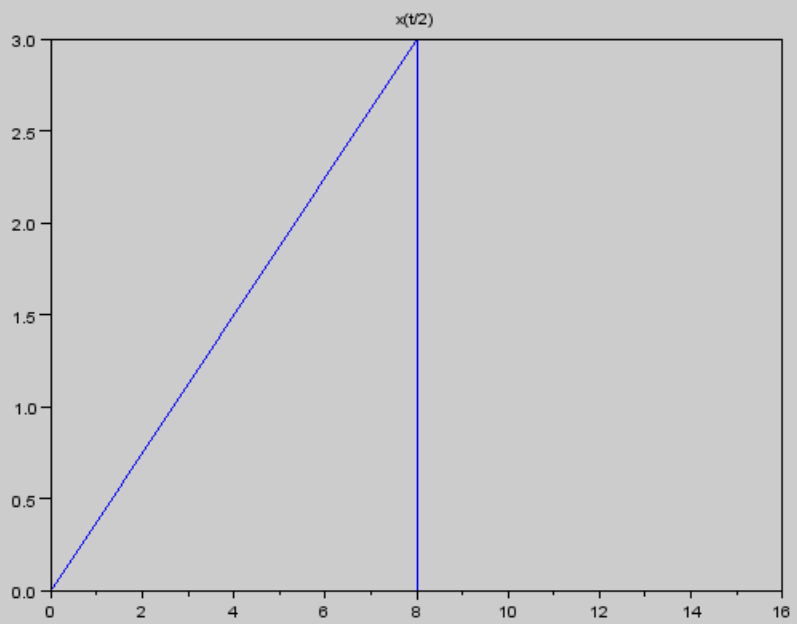
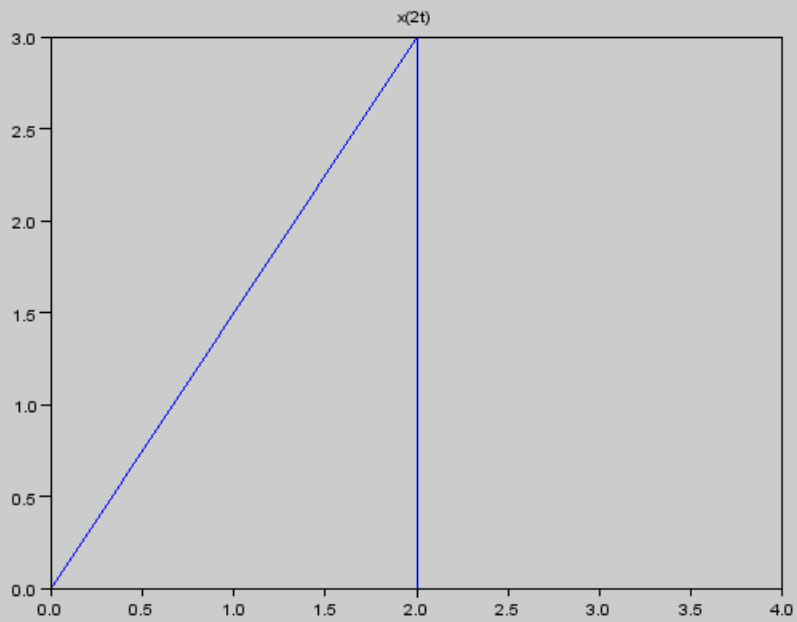


Figure 1.1: shifting and scaling of continuous time signal



```

11 x(i+1)=x(i);
12 x(i+2:9)=0;
13 figure
14 a=gca();
15 plot2d3(t,x)
16 plot(t,x,'r.')
17 xtitle('x[n]')
18 t1=t+2;
19 figure
20 a=gca();
21 plot2d3(t1,x)
22 plot(t1,x,'r.')
23 xtitle('x[n-2]')
24 a.thickness=2;
25 t2=-1:1/2:3;
26 figure
27 a=gca()
28 plot2d3(ceil(t2),x)
29 plot(ceil(t2),x,'r.')
30 xtitle('x[2n]')
31 a.thickness=2;
32 t3=-6:2;
33 figure
34 a=gca();
35 plot2d3(t3,x($:-1:1))
36 plot(t3,x($:-1:1),'r.')
37 xtitle('x[-n]')
38 a.y_location='right';
39 a.thickness=2;
40 t4=t3+2;
41 figure
42 a=gca();
43 plot2d3(t4,x($:-1:1))
44 plot(t4,x($:-1:1),'r.')
45 xtitle('x[-n+2]')
46 a.y_location='right';
47 a.thickness=2;

```

---

### Scilab code Exa 1.3 sampling of continuous time signal

```
1 //example_1.3
2 // sampling of continuos function
3 clear all;
4 clc;
5 close;
6 t=-1:1/100:1;
7 for i=1:length(t)
8     x(i)=1-abs(t(i))
9 end
10 figure
11 a=gca();
12 plot2d(t,x)
13 xtitle('x(t)')
14 a.y_location='middle'
15 figure
16 a=gca();
17 for i=1:length(t)
18     if t(i)<0 then
19         t1(i)=ceil(t(i)*4)
20     else
21         t1(i)=floor(t(i)*4)
22     end
23 end
24 plot2d3(ceil(t1),x)
25 xtitle('x[n]=x[n/4]')
26 figure
27 a=gca();
28 for i=1:length(t)
```

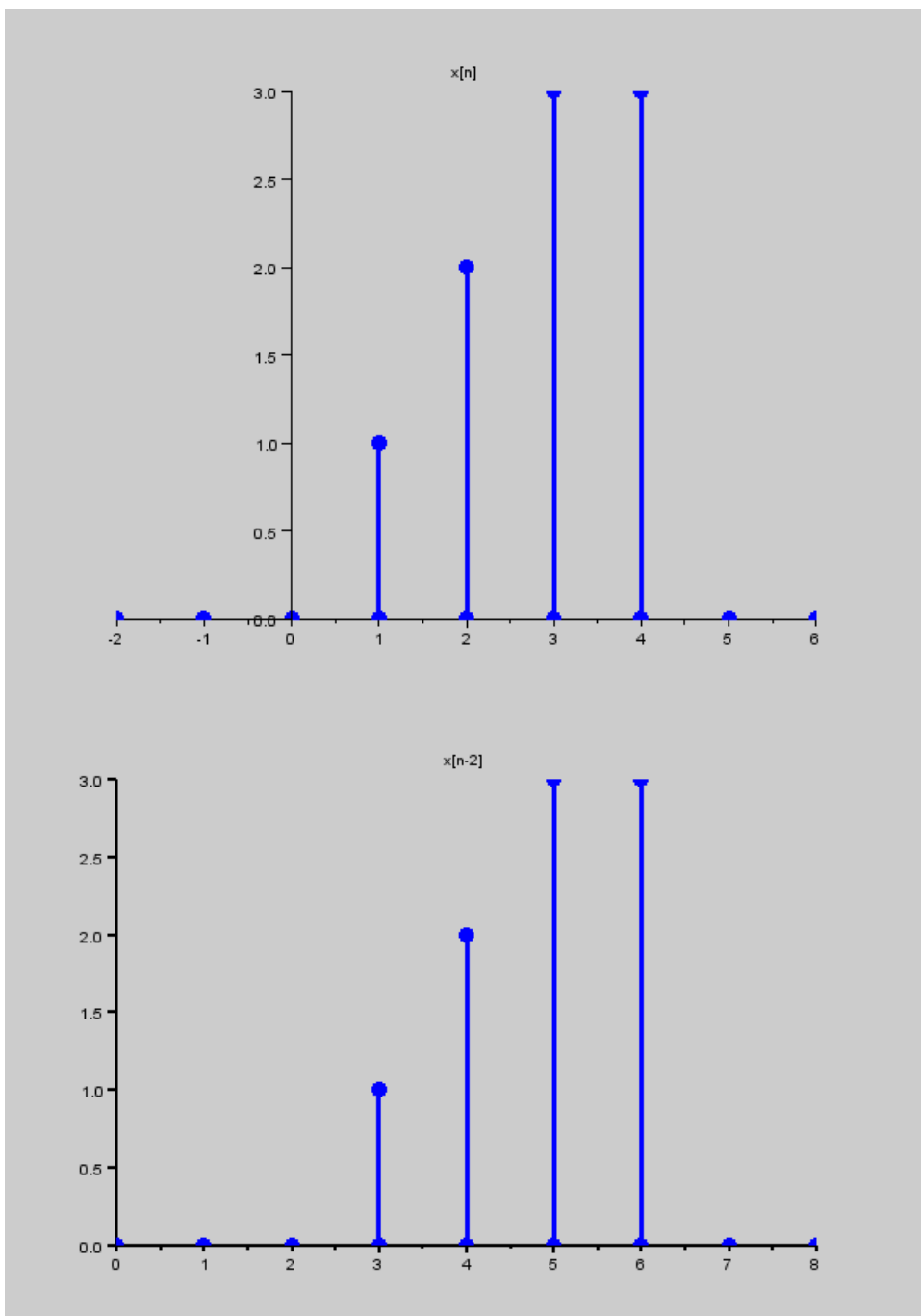
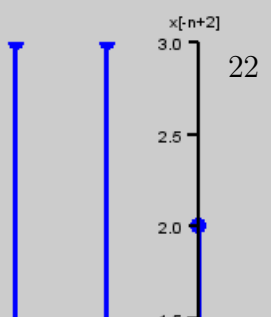
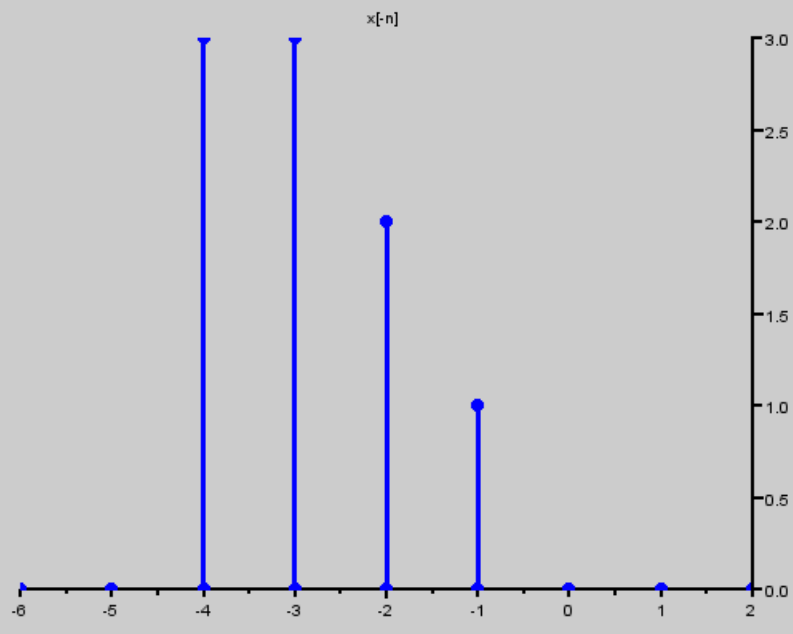
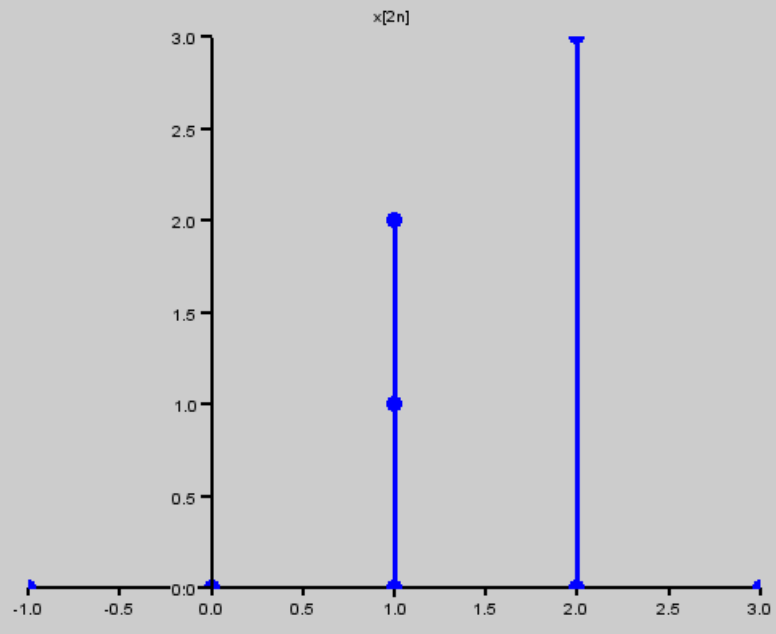


Figure 1.3: shifting and scaling of discrete time signal



```

29     if t(i)<0 then
30         t2(i)=ceil(t(i)*2)
31     else
32         t2(i)=floor(t(i)*2)
33     end
34 end
35 plot2d3(ceil(t2),x)
36 xtitle('x[n]=x[n/2]')
37 figure
38 a=gca();
39 for i=1:length(t)
40     if t(i)<0 then
41         t3(i)=ceil(t(i))
42     else
43         t3(i)=floor(t(i))
44     end
45 end
46 plot2d3(ceil(t3),x)
47 xtitle('x[n]')

```

---

#### Scilab code Exa 1.4 discrete time signal

```

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

```



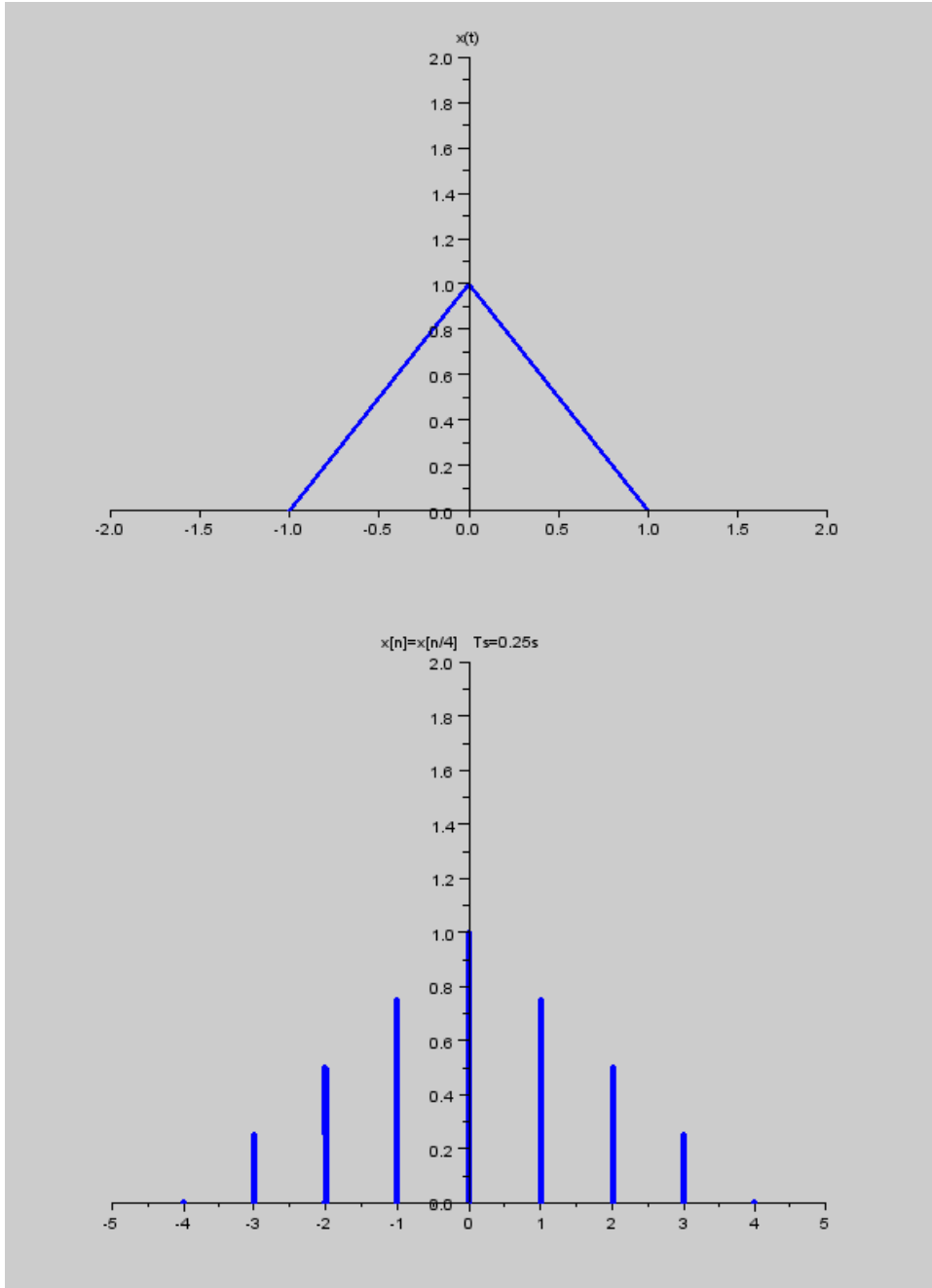


Figure 1.5: sampling of continuous time signal

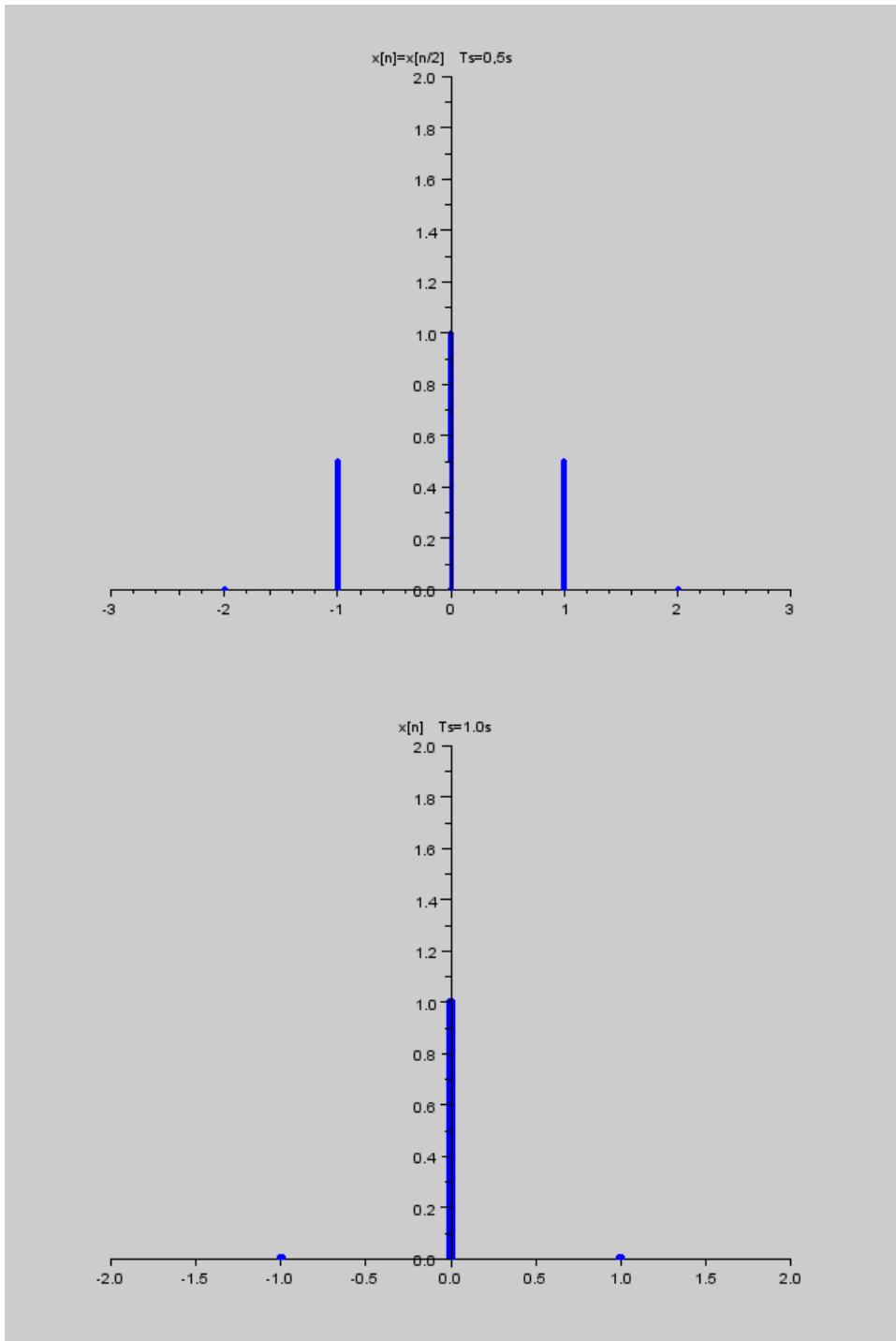


Figure 1.6: sampling of continuous time signal

```

    ));
10 figure
11 a=gca();
12 plot2d3(t3,x1)
13 plot(t3,x1,'r.')
14 xtitle('x1[n]')
15 figure
16 a=gca();
17 plot2d3(t3,x2)
18 plot(t3,x2,'r.')
19 xtitle('x2[n]')
20 a.x_location='middle'
21 figure
22 a=gca();
23 plot2d3(t3,x1+x2)
24 plot(t3,x1+x2,'r.')
25 xtitle('y1[n]=x1[n]+x2[n]')
26 a.x_location='origin'
27 figure
28 a=gca();
29 plot2d3(t3,2.*x1)
30 plot(t3,2.*x1,'r.')
31 xtitle('y2[n]=2 * x1[n]')
32 figure
33 a=gca();
34 plot2d3(t3,x1.*x2)
35 plot(t3,x1.*x2,'r.')
36 xtitle('y2[n]=x2[n] * x1[n]')

```

---

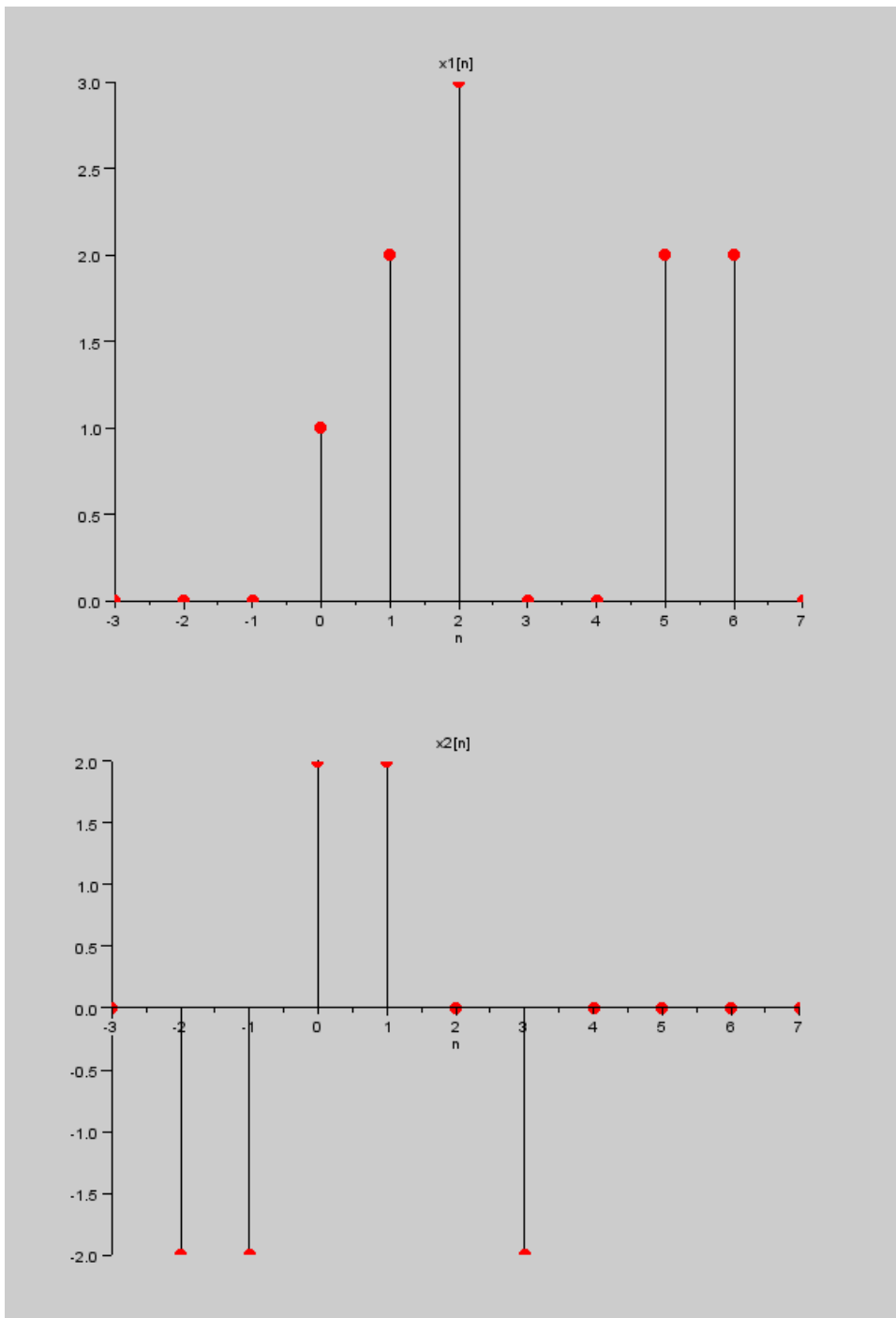
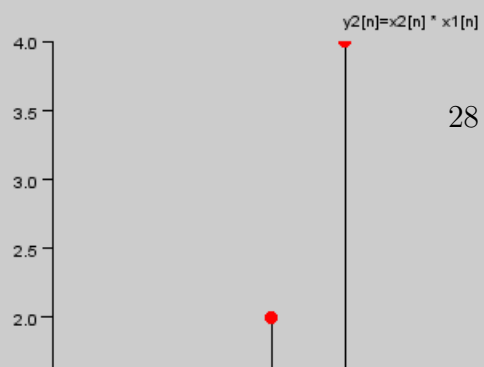
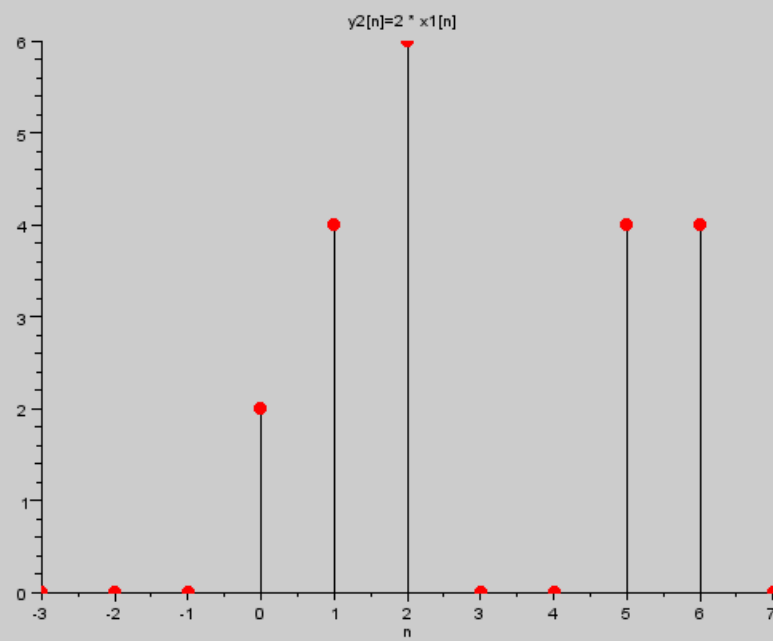
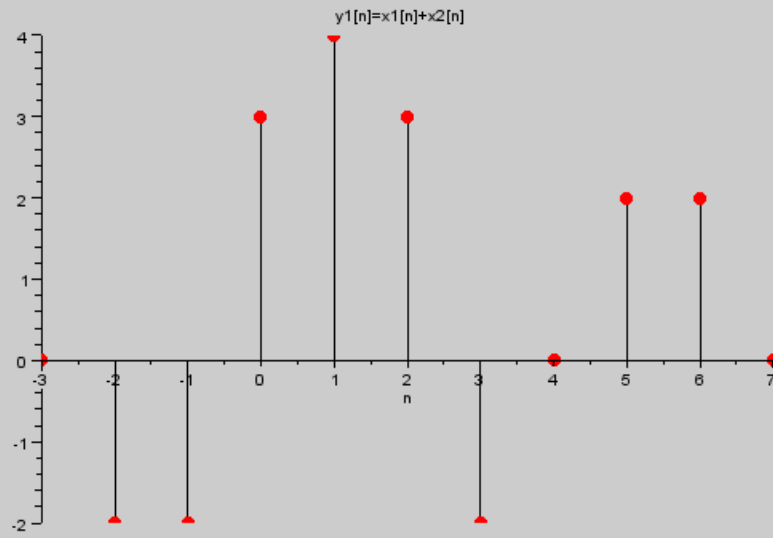
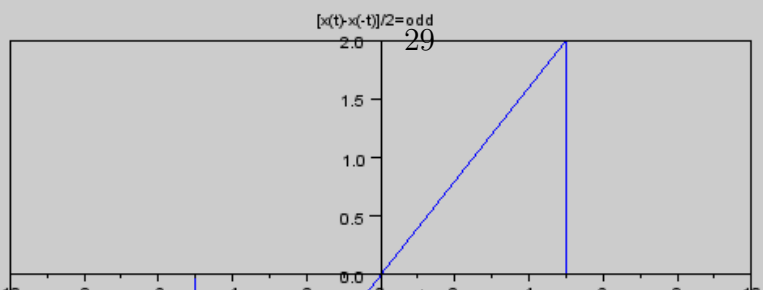
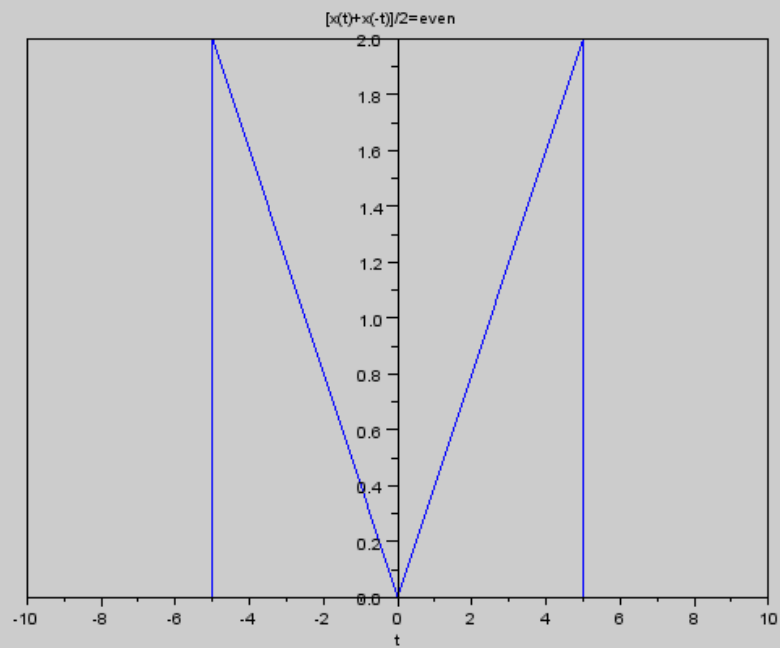
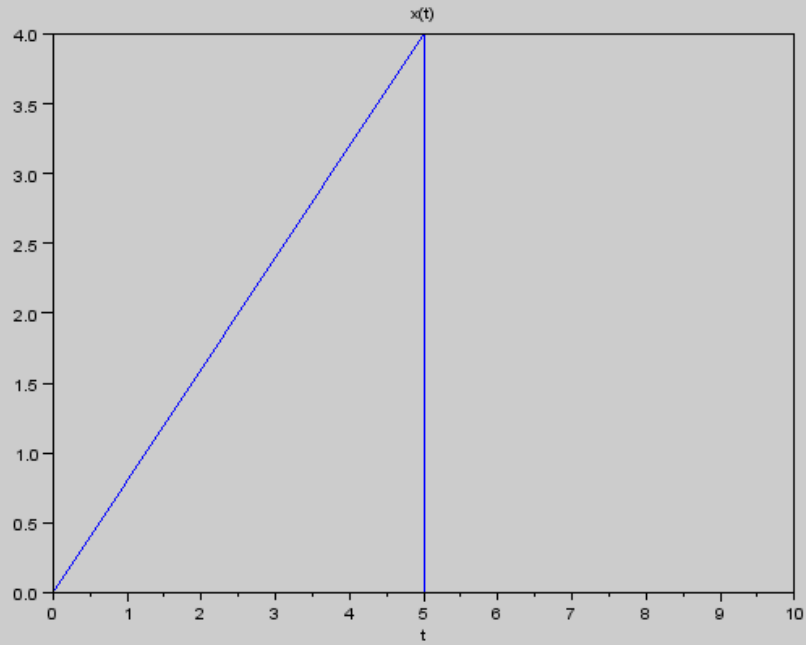


Figure 1.7: discrete time signal





### Scilab code Exa 1.5.a even and odd components

```
1 //ex_5 even and odd signals of x(t)
2 clear ;
3 clear x;
4 clear t;
5 clc;
6 close;
7 t = 0:1/100:5;
8 for i = 1:length(t)
9     x(i) = (4/5)*t(i) ;
10 end
11 for i = length(t)+1:2*length(t)
12 x(i) = 0;
13 end
14 figure
15 a=gca();
16 t1=0:1/100:10;
17 plot(t1,x(1:$-1))
18 xtitle('x(t)')
19 figure
20 a=gca();
21 t3=0:1/100:10;
22 plot(t3,x(1:$-1)/2)
23 xtitle('[x(t)+x(-t)]/2=even')
24 t2=-10:1/100:0;
25 plot(t2,x($:-1:2)/2)
26 a.y_location='origin'
27 figure
28 a=gca();
29 t4=0:1/100:10;
30 plot(t4,x(1:$-1)/2)
31 xtitle('[x(t)-x(-t)]/2=odd')
32 t5=-10:1/100:0;
33 plot(t5,-x($:-1:2)/2)
34 a.y_location='origin'
35 a.x_location='origin'
```

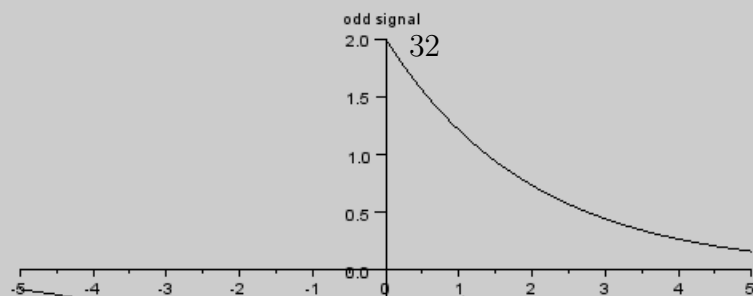
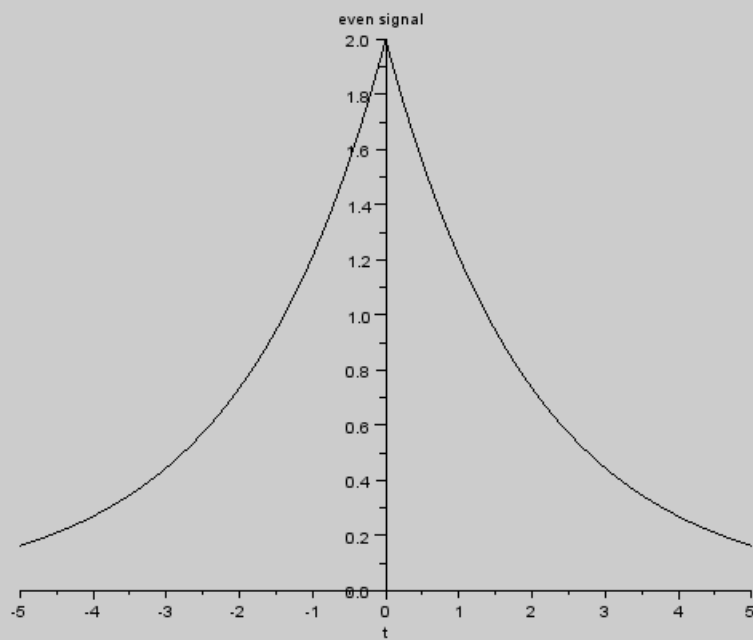
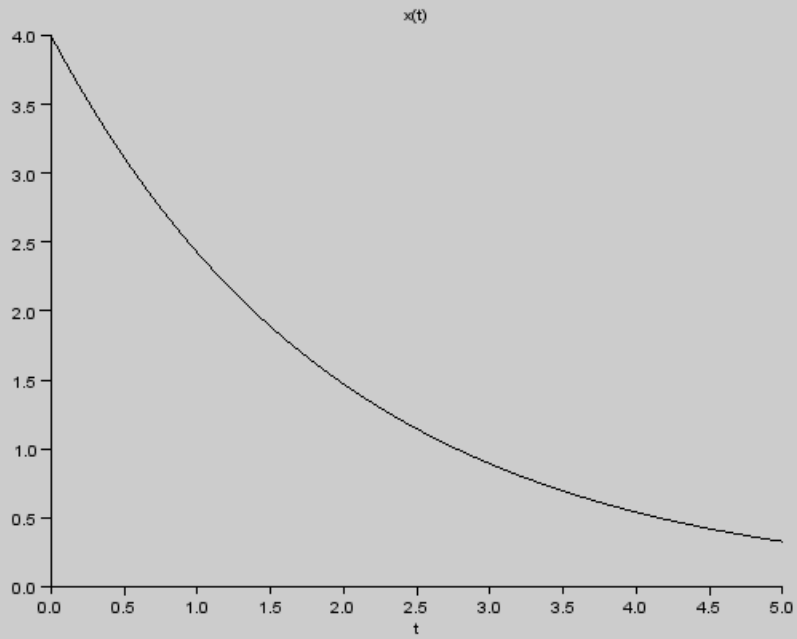
---

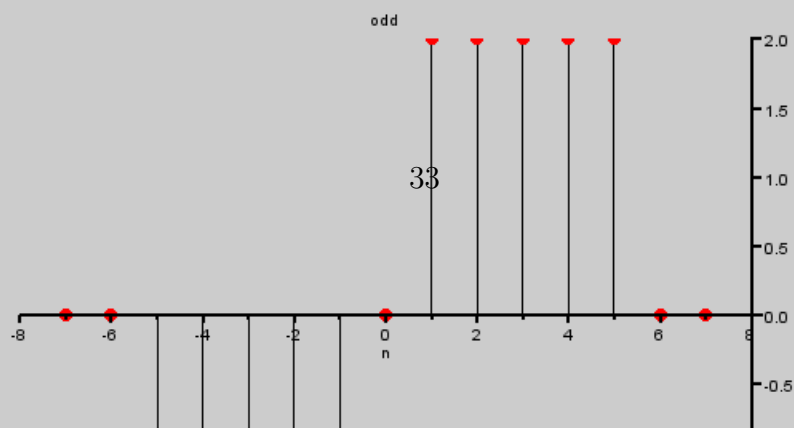
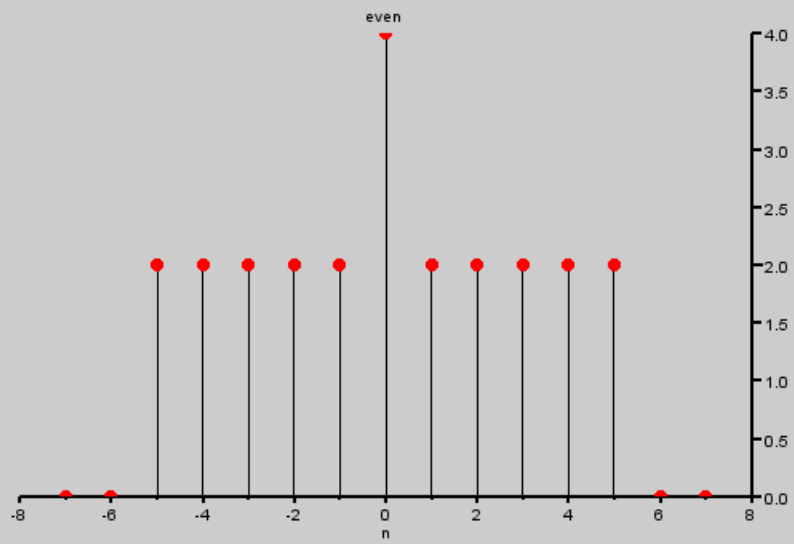
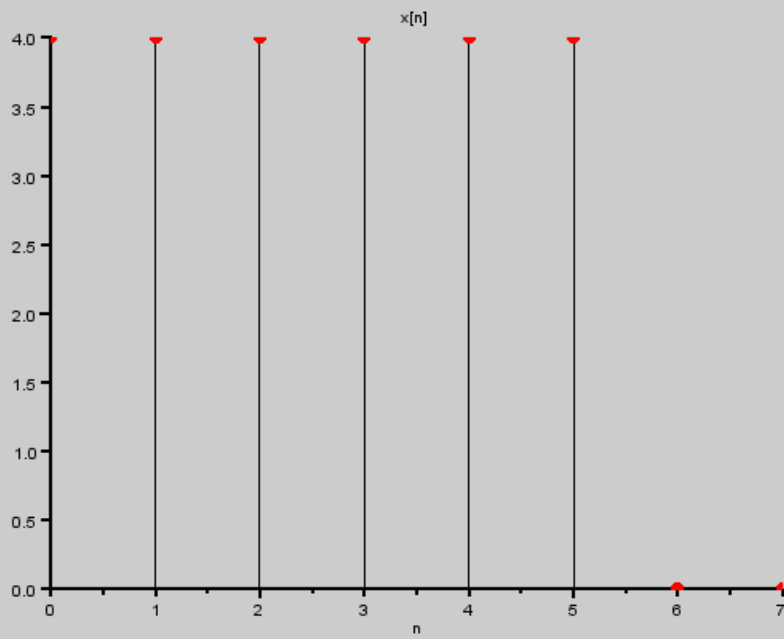
**Scilab code Exa 1.5.b** even and odd components

```
1 //ex_5 even and odd signals of x(t)
2 clear all;
3 clc;
4 close;
5 t = 0:1/100:5;
6 x=4*exp(-0.5.*t)
7 figure
8 a=gca();
9 xtitle('x(t)')
10 plot2d(t,x)
11 figure
12 a=gca();
13 xtitle('even signal')
14 plot2d(t,x/2)
15 t1=-5:1/100:0;
16 plot2d(t1,x($:-1:1)/2)
17 a.y_location='origin'
18 figure
19 a=gca();
20 xtitle('odd signal')
21 plot2d(t,x/2)
22 t1=-5:1/100:0;
23 plot2d(t1,-x($:-1:1)/2)
24 a.y_location='origin'
25 a.x_location='origin'
```

---



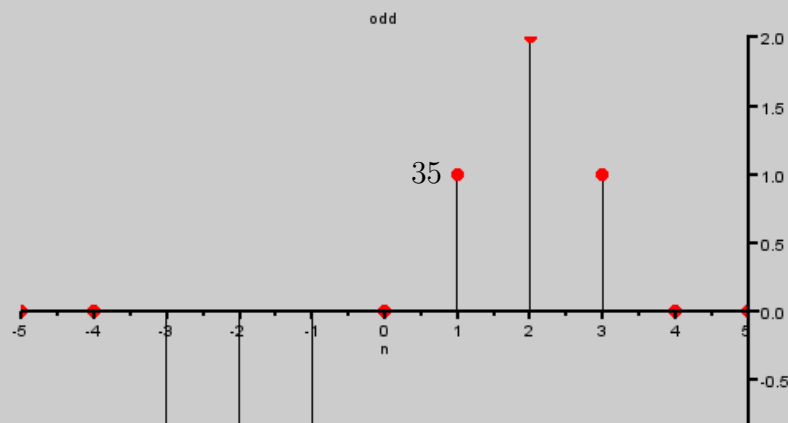
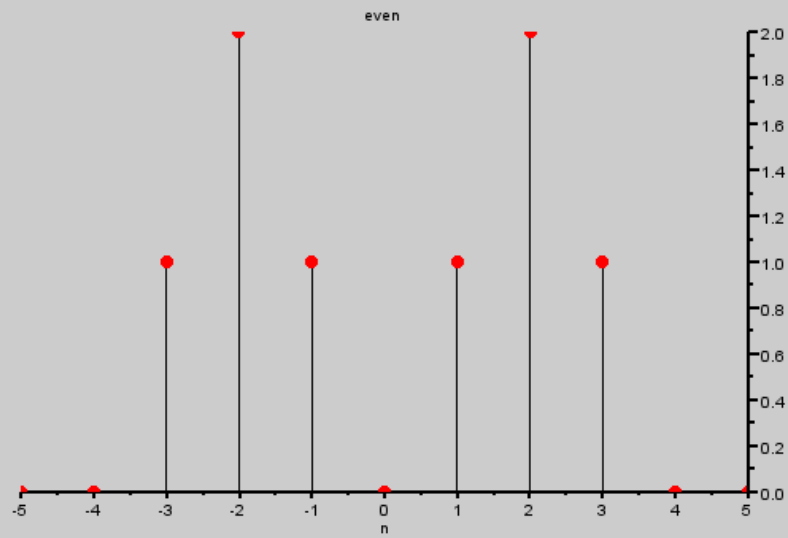
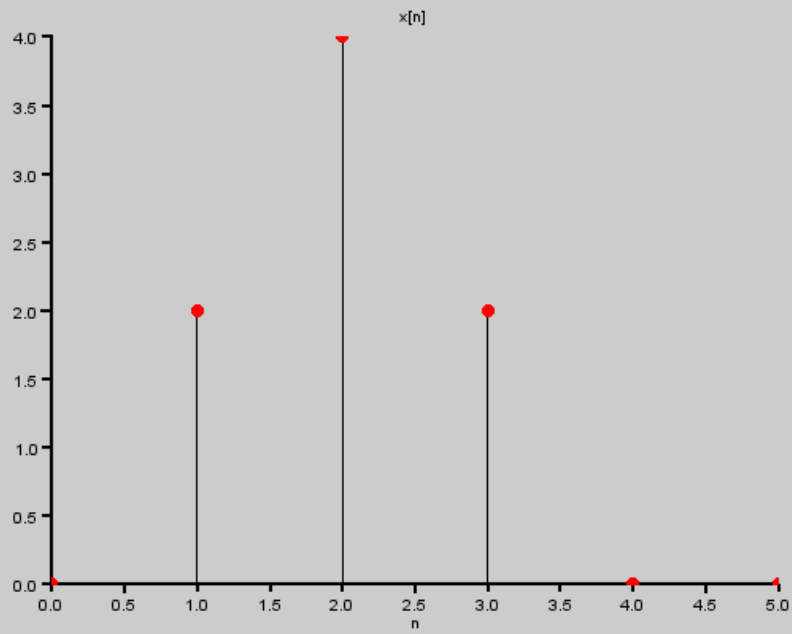




### Scilab code Exa 1.5.c even and odd components

```
1 //ex_5 even and odd signals of x(t)
2 clear all;
3 clc;
4 close;
5 t=0:7;
6 x=[4 4 4 4 4 4 0 0];
7 figure
8 a=gca();
9 plot2d3(t,x)
10 plot(t,x,'r.')
11 xtitle('x[n]')
12 a.thickness=2;
13 t1=-7:0;
14 figure
15 a=gca();
16 t2=-7:7;
17 y=[x($:-1:2)./2 x(1) x(2:8)./2 ]
18 plot2d3(t2,y)
19 plot(t2,y,'r.')
20 xtitle('even')
21 a.y_location='right';
22 a.thickness=2;
23 figure
24 a=gca();
25 z=[-x($:-1:2)./2 0 x(2:8)./2 ]
26 plot2d3(t2,z)
27 plot(t2,z,'r.')
28 xtitle('odd')
29 a.y_location='right';
30 a.x_location='origin';
31 a.thickness=2;
```

---



### Scilab code Exa 1.5.d even and odd components

```
1 //ex_5 d even and odd signals of x(t)
2 clear all;
3 clc;
4 close;
5 t=0:5;
6 x=[ 0 2 4 2 0 0];
7 figure
8 a=gca();
9 plot2d3(t,x)
10 plot(t,x,'r.')
11 xtitle('x[n]','n')
12 a.thickness=2;
13 figure
14 a=gca();
15 t2=-5:5 ;
16 y=[ x($:-1:2)./2  x(1)  x(2:6)./2 ]
17 plot2d3(t2,y)
18 plot(t2,y,'r.')
19 xtitle('even','n')
20 a.y_location='right';
21 a.thickness=2;
22 figure
23 a=gca();
24 z=[ -x($:-1:2)./2  0  x(2:6)./2 ]
25 plot2d3(t2,z)
26 plot(t2,z,'r.')
27 xtitle('odd','n')
28 a.y_location='right';
29 a.x_location='origin';
30 a.thickness=2;
```

---

### Scilab code Exa 1.6 even and odd components

```

1 //ex_6 even and odd signal of e^jt
2 clear all;
3 clc;
4 close;
5 t = 0:1/100:5;
6 x=exp(%i.*t);
7 y=exp(-%i.*t);
8 even=x./2+y./2;
9 odd=x./2-y./2;
10 figure
11 a=gca();
12 plot2d(t,even)
13 a.x_location='origin'
14 xtitle('even','t')
15 figure
16 a=gca();
17 plot2d(t,odd./%i)
18 a.x_location='origin'
19 xtitle('odd','t')

```

---

**Scilab code Exa 1.9** periodicity of exponential signal

```

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

```

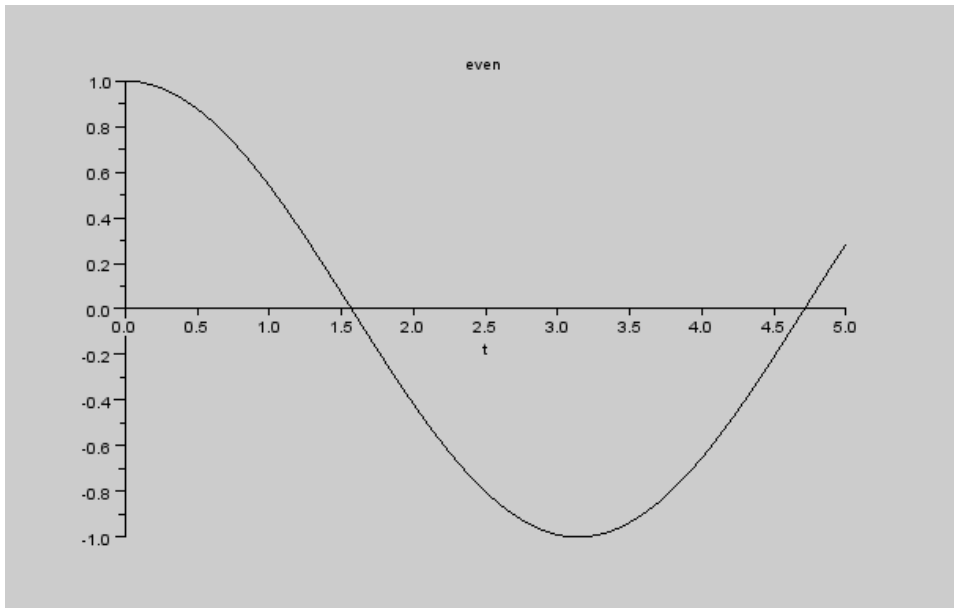


Figure 1.13: even and odd components

```

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

```

---

**Scilab code Exa 1.10** periodicity of sinusoidal signal

```

1 clear all;
2 clc;
3 close;
4 t=0:1/100:10;
5 w=1;
6 theta=%pi/3;
7 T=2*%pi/w;
8 x=cos(t*w+theta);

```

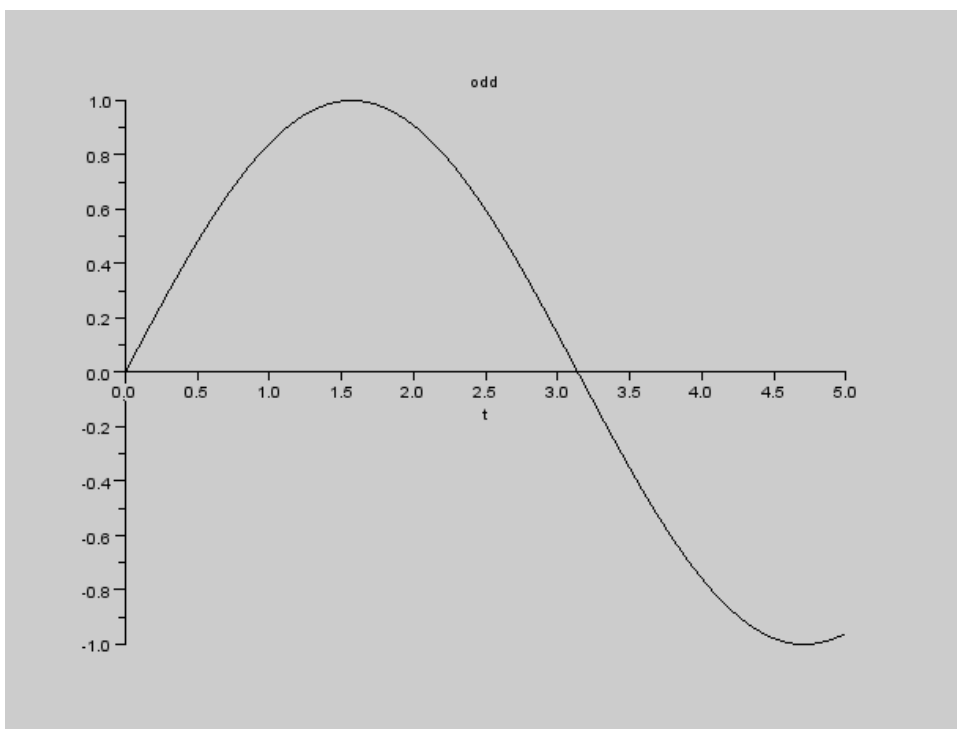


Figure 1.14: even and odd components



```

9 y=cos((t+T)*w+theta);
10 if ceil(x)==ceil(y) then
11     disp('cos(wo*t+theta) is periodic with T=2*pi/W0
12         ')
12 else
13     disp('nonperiodic ')
14 end

```

---

**Scilab code Exa 1.11** periodicity of exponential sequence

```

1 clear all;
2 clc;
3 close;
4 n=0:100;
5 w0=1;
6 N=2*pi/w0;
7 x=exp(i*w0*n);
8 y=exp(i*w0*(n+N));
9 if ceil(x)==ceil(y) then
10     disp('e^iwn is periodic with N=2*pi/W0')
11 else
12     disp('nonperiodic ')
13 end

```

---

**Scilab code Exa 1.16** fundamental period

```

1 //ex_16 to check if a function is periodic
2 clear all;
3 clc;
4 close;
5 //x(t)=cos(t+pi/4)
6 w=1
7 t=2*pi; //t=2*pi/w

```

```

8  disp(t, 'a')
9  //x(t)=sin(2*pi*t/3)
10 w=2*pi/3;
11 t=2*pi/w; //check if t is rational
12 if t==ceil(t) then
13     disp(t, 'b');
14 else
15     disp('non periodic', 'b')
16 end
17 //x(t)=cos(pi*t/3)+sin(pi*t/4)
18 w1=pi/3;
19 w2=pi/4;
20 t1=2*pi/w1;
21 t2=2*pi/w2;
22 t=lcm([t1 t2/2]);
23
24 if t==ceil(t) then
25     disp(t, 'c');
26 else
27     disp('non periodic')
28 end
29 //x(t)=cos(t)+sin(sqrt(2)*t)
30 w1=1;
31 w2=sqrt(2);
32 t1=2*pi/w1;
33 t2=2*pi/w2;
34 t=lcm([t1 t2]);
35 if t==ceil(t) then
36     disp(t, 'd');
37 else
38     disp('non periodic')
39 end
40 //x(t)=(sin(t))^2=(1-cos(2*t))/2
41 w=2;
42 t=2*pi/w;
43 disp(t, 'e')
44 //x(t)=e^(i*(pi/2)*t-1)
45 w=pi/2;

```

```

46 t=2*%pi/w;
47 disp(t, 'f')
48 //x[n]=e^(%i*(%pi/4))
49 w=%pi/4;
50 N=2*%pi/w;
51 disp(N, 'g')
52 //x[n]=cos(1*n/4)
53 w=1/4;
54 N=2*%pi/w;
55 if N==ceil(N) then
56     disp(N, 'h');
57 else
58     disp('non periodic', 'h')
59 end
60 //x[n]=cos(%pi*n/3)+sin(%pi*n/4)
61 w1=%pi/3;
62 w2=%pi/4;
63 N1=2*%pi/w1;
64 N2=2*%pi/w2;
65 N=lcm([N1 N2/2]);
66 if N==ceil(N) then
67     disp(N, 'i');
68 else
69     disp('non periodic', 'i')
70 end
71 //x[n]=(cos(%pi*n/8))^2=(1+cos(%pi*n/4))/2
72 w=%pi/4;
73 N=2*%pi/w;
74 disp(N, 'j')

```

---

### Scilab code Exa 1.21 unit step signal

```

1 //ex_21 to check if  $u(-t)=\{1 \text{ for } t<0 \text{ and } 0 \text{ for } t>0\}$ 
2 clear all;
3 clc;

```

```

4 close;
5 t=-10:1/100:10;
6 for i = 1:(length(t))
7     if t(i)<0 then
8         x(i)=0
9     else
10        x(i)=1
11    end
12 end
13 figure
14 a=gca();
15 plot(t,x)
16 xtitle('u(t)', 't')
17 a.data_bounds=[-10,-1;10,2];
18 figure
19 a=gca();
20 plot(t,x($:-1:1))
21 xtitle('u(-t)', 't')
22 a.data_bounds=[-10,-1;10,2];

```

---

### Scilab code Exa 1.22 continuous time signal

```

1 //ex_22 product of x(t) and unit step function
2 clear x;
3 clear t;
4 clear;
5 clear y;
6 clc;
7 close;
8 t=-1:1/100:2
9 for i=1:(length(t))

```

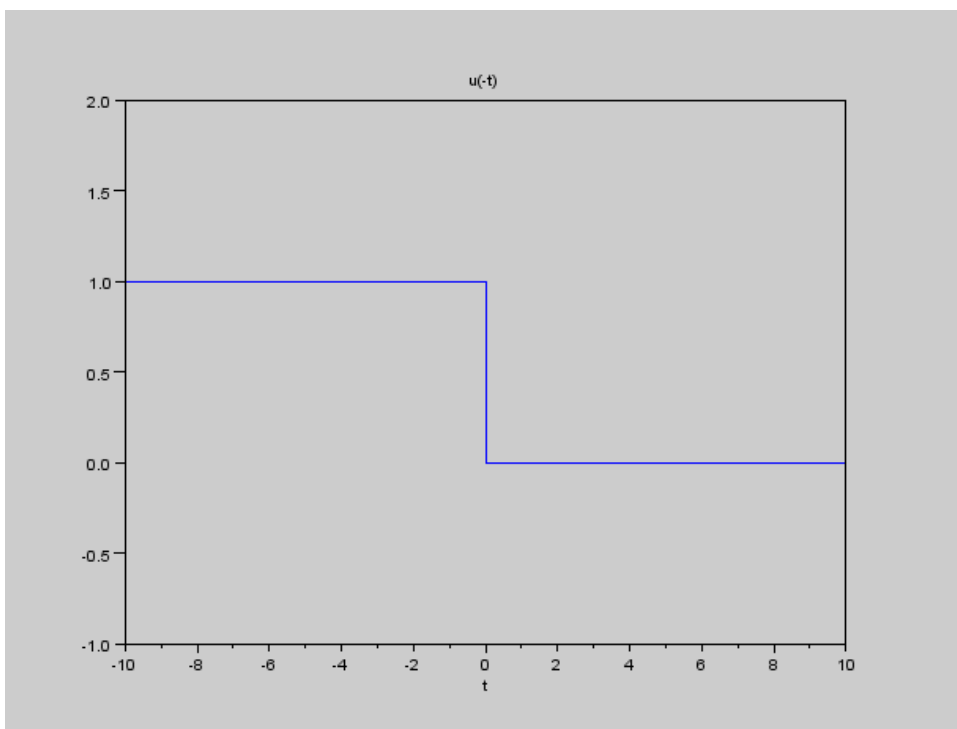


Figure 1.15: unit step signal

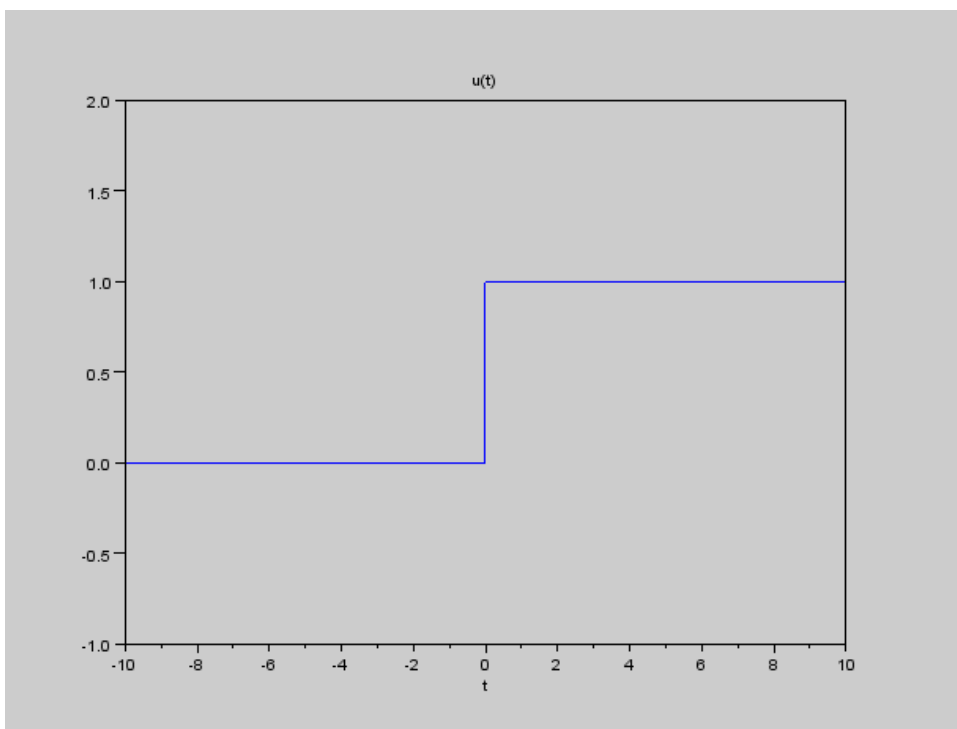


Figure 1.16: unit step signal

```

10     if t(i)<0 then
11         x(i)=t(i)+1;
12     elseif t(i)<1
13         x(i)=1
14     else
15         x(i)=2
16     end
17 end
18 figure
19 a=gca();
20 plot2d(t,x)
21 a.y_location='origin'
22 xtitle('x(t)', 't')
23 //a.x(t)*u(1-t)
24 for i = 1:(length(t))
25     if t(i)<1 then
26         u1(i)=1
27     else
28         u1(i)=0
29     end
30 end
31 y=x.*u1
32 figure
33 a=gca();
34 plot2d(t,y)
35 a.y_location='origin'
36 xtitle('x(t)*u(1-t)', 't')
37 for i = 1:(length(t))
38     if t(i)<1 & t(i)>0 then
39         u2(i)=1
40     else
41         u2(i)=0
42     end
43 end
44 y=x.*u2;
45 figure
46 a=gca();
47 plot2d(t,y)

```

```

48 a.y_location='origin'
49 xtitle('x(t)*u(t-1)', 't')
50 for i = 1:(length(t))
51     if t(i)==3/2 then
52         z(i)=x(i)
53     else
54         z(i)=0
55     end
56 end
57 figure
58 a=gca();
59 plot2d3(t,z)
60 poly_1=a.children.children;
61 poly_1.thickness=3;
62 a.y_location='origin'
63 xtitle('x(t)*delta(t-3/2)', 't')

```

---

### Scilab code Exa 1.23 discrete time signal

```

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

```



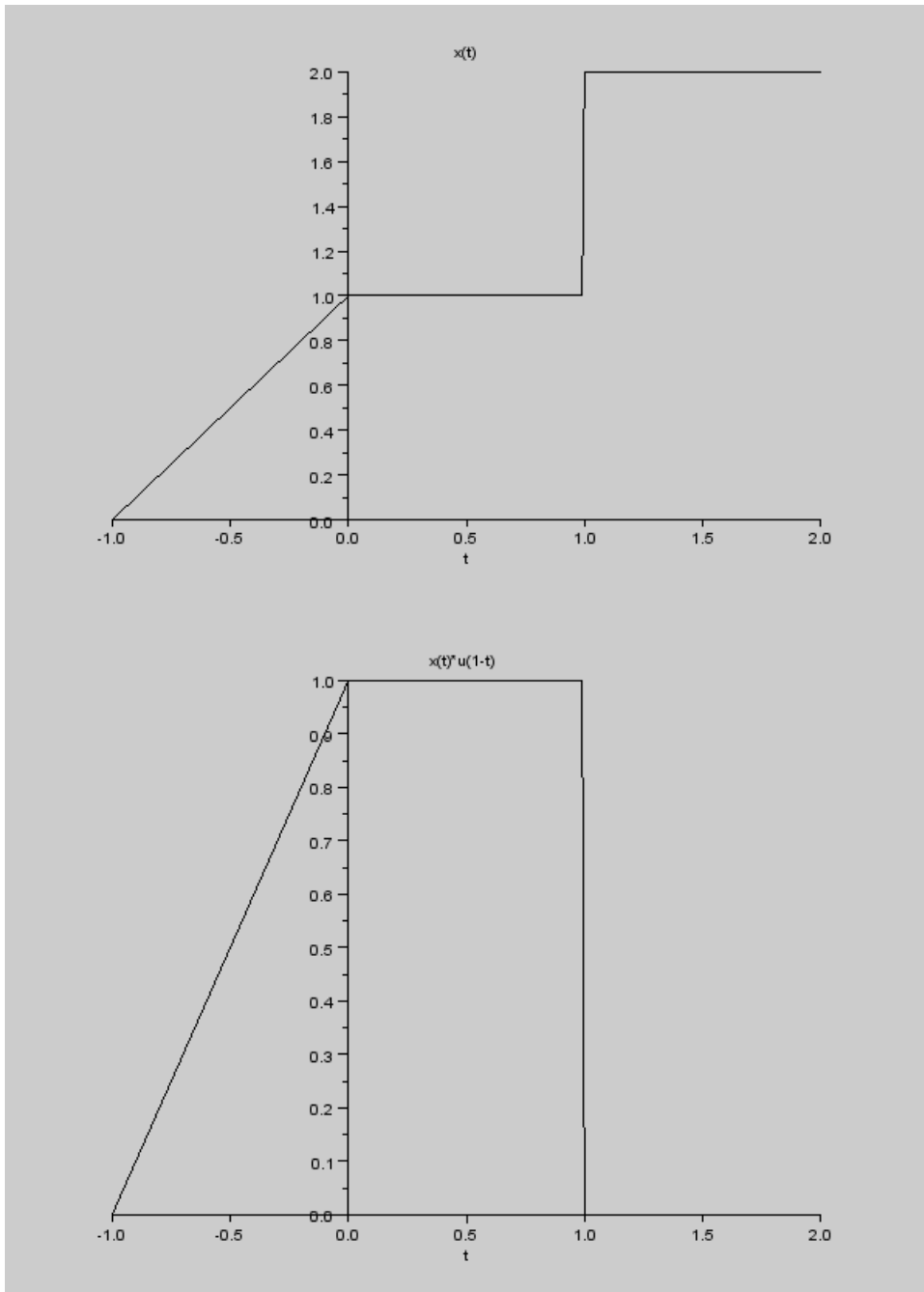


Figure 1.17: continuous time signal

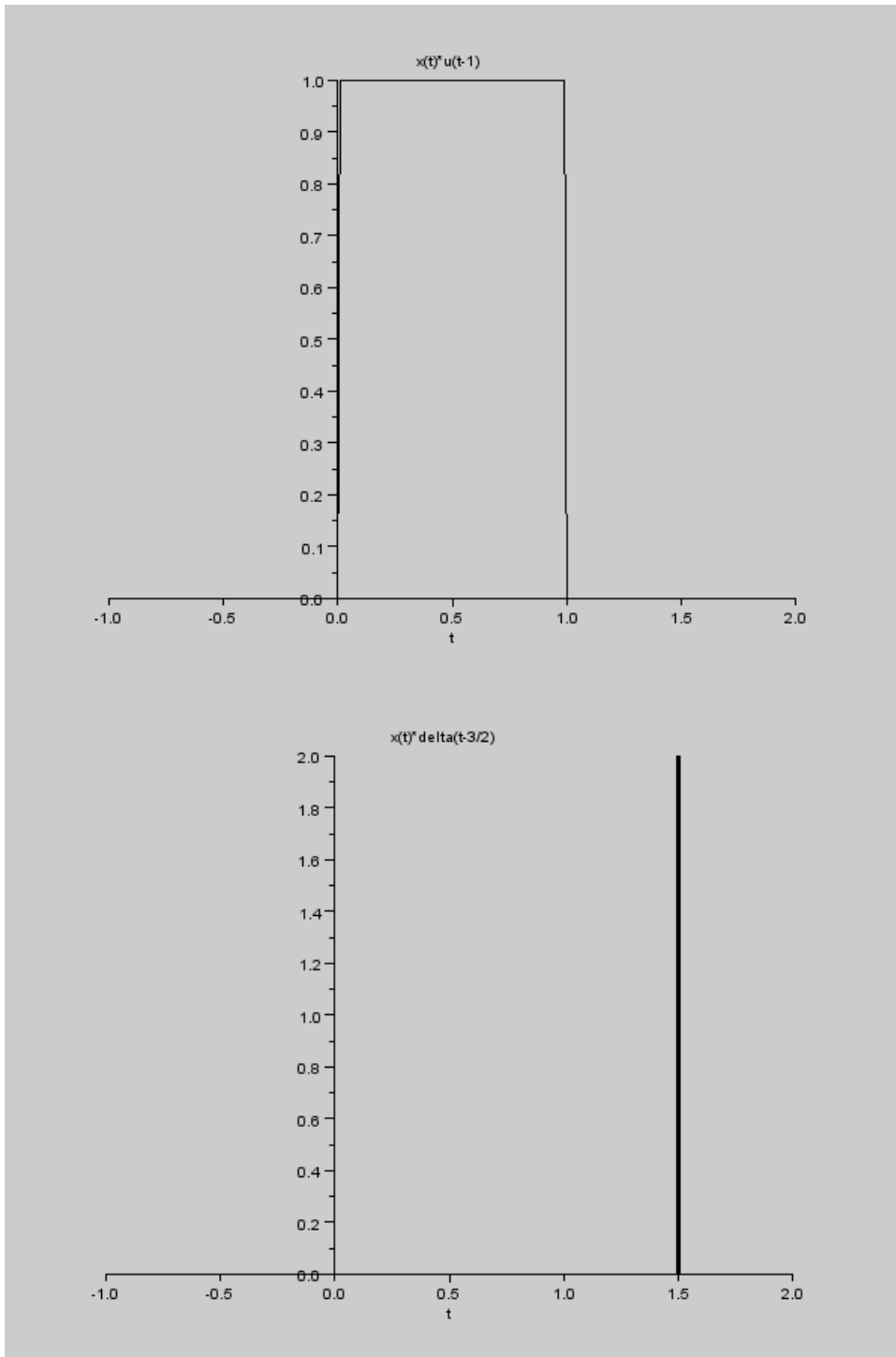
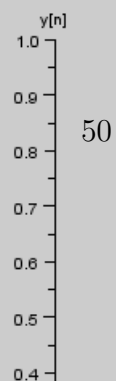
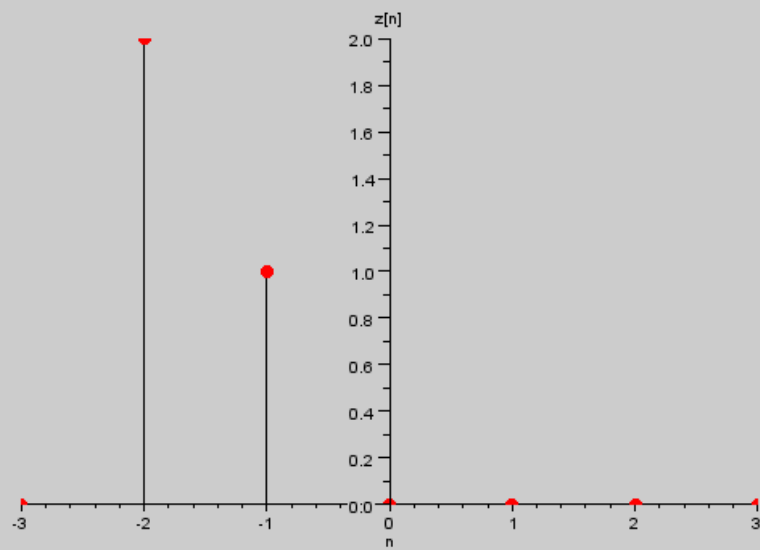
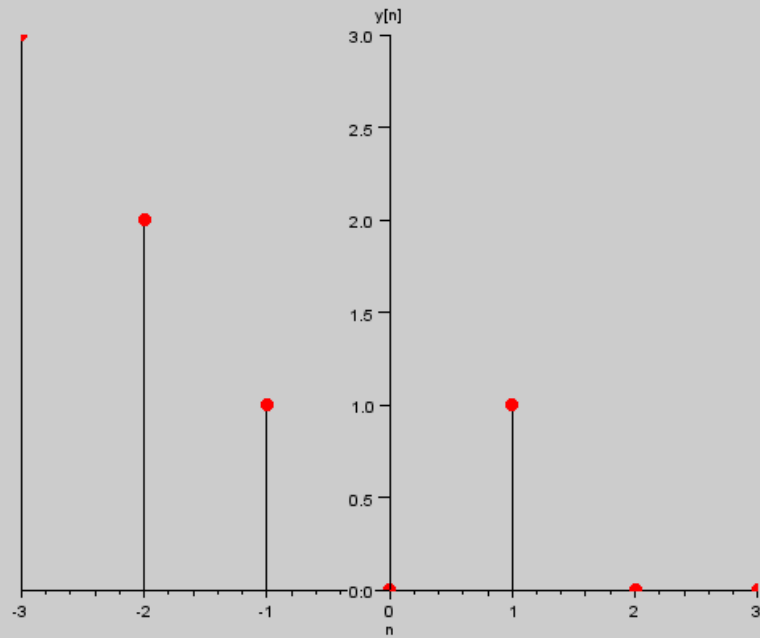


Figure 1.18: continuous time signal



```

10         u1(i)=1;
11     else
12         u1(i)=0;
13     end
14 end
15 y=x.*u1';
16 figure
17 a=gca();
18 plot2d3(t,y)
19 plot(t,y,'r.')
20 xtitle('y[n]','n')
21 a.y_location='origin'
22 //u[n+2]-u[n]
23 for i=1:length(t);
24     if t(i)<1 & t(i)>=-2 then
25         u2(i)=1;
26     else
27         u2(i)=0;
28     end
29 end
30 z=x.*u2';
31 figure
32 a=gca();
33 plot2d3(t,z)
34 plot(t,z,'r.')
35 xtitle('z[n]','n')
36 a.y_location='origin'
37 //$(n-1)
38 for i=1:length(t);
39     if t(i)==1 then
40         del(i)=1;
41     else
42         del(i)=0;
43     end
44 end
45 p=x.*del';
46 figure
47 a=gca();

```

```

48 plot2d3(t,p)
49 plot(t,p,'r.')
50 xtitle('y[n]','n')
51 a.y_location='origin'

```

---

### Scilab code Exa 1.31 first derivative of the signals

```

1 clear all;
2 close;
3 clc;
4 t=-10:0.1:10;
5 a=2;
6 //x(t)=u(t)-u(t-a)
7 x=[zeros(1,find(t==0)-1) ones(1,find(t==a)-find(t
    ==0)+1) zeros(1,length(t)-find(t==a))];
8 subplot(2,1,1)
9 plot(t,x)
10 xtitle('x(t)','t')
11 subplot(2,1,2)
12 plot2d3(t(1:$-1),diff(x))
13 xtitle('diff(x(t))','t')
14 //x(t)=t*(u(t)-u(t-a))
15 xb=t.*x;
16 figure
17 subplot(2,1,1)
18 plot(t,xb)
19 xtitle('x(t)','t')
20 subplot(2,1,2)
21 plot2d(t(1:$-1),diff(xb))
22 xtitle('diff(x(t))','t')
23 //x(t)=sgn(t)
24 x=[-ones(1,find(t==0)-1) ones(1,length(t)-find(t==0)
    +1)];
25 figure
26 subplot(2,1,1)

```

```
27 plot(t,x)
28 xtitle('x(t)', 't')
29 subplot(2,1,2)
30 plot2d(t(1:$-1),diff(x))
31 xtitle('diff(x(t))', 't')
```

---

#### Scilab code Exa 1.35 linearity

```
1 //ex_35 to check if a system is linear or non-linear
2 clear all;
3 clc;
4 close;
5 x1=2;
6 x2=3;
7 y1=x1*x1;
8 y2=x2*x2;
9 y=y1+y2;
10 z=(x1+x2)*(x1+x2);
11 if z==y then
12     disp('the system is linear')
13 else
14     disp("the system is nonlinear")
15 end
```

---

#### Scilab code Exa 1.36 memoryless causal stable system

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

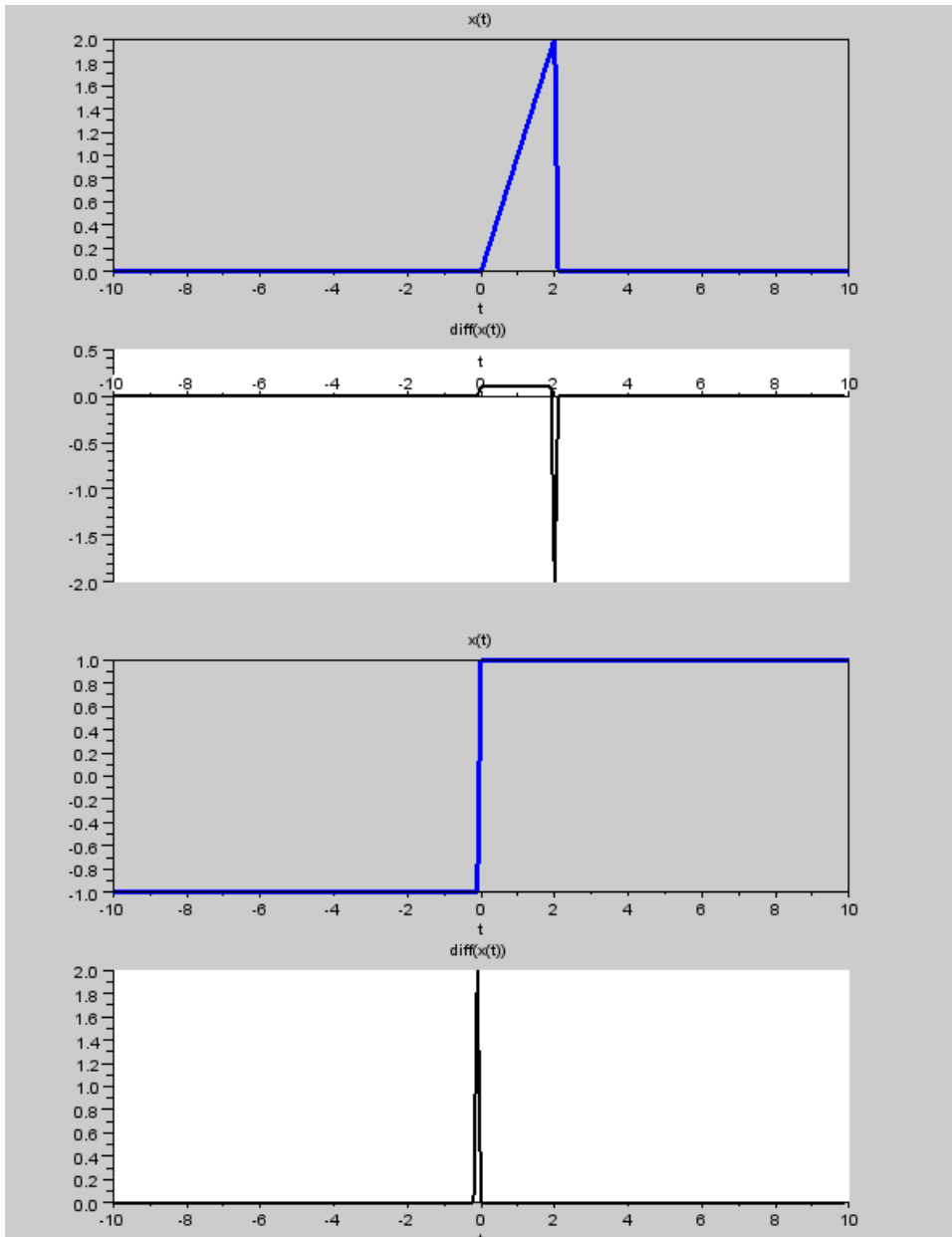


Figure 1.20: first derivative of the signals

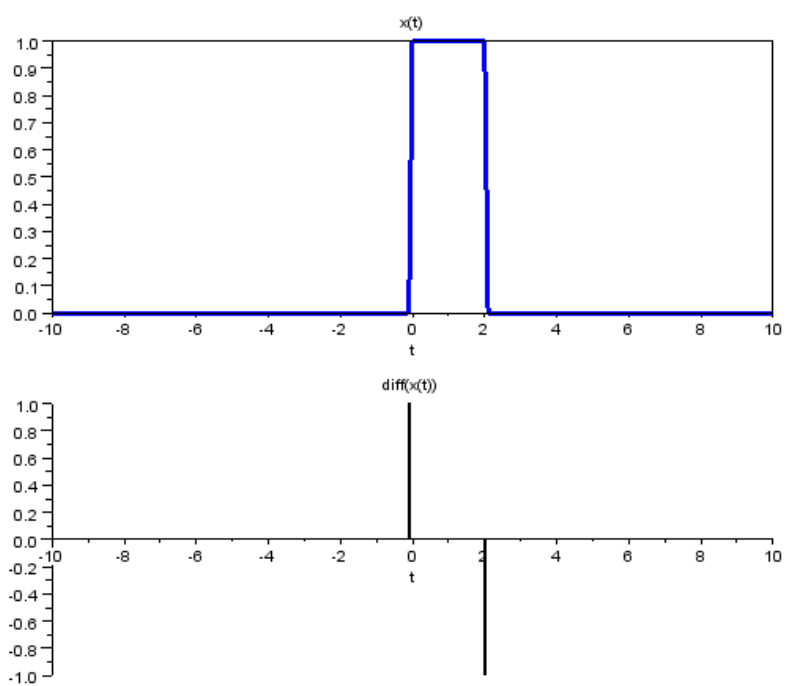


Figure 1.21: first derivative of the signals



```

3  clc;
4  s = 2; //shift
5  T = 20; //lenght of signal
6  x(1)=1;
7  for n = 2:T
8      x(n) = n;
9      y(n) = x(n-1);
10 end
11 if y(2)==x(2) then
12     disp("memoryless")
13 else
14     disp("not memoryless")
15 end
16 //causal if it does'nt depend on future
17 if y(2)==x(2) | y(2)==x(1) then
18     disp('causal')
19 else
20     disp('non casual')
21 end
22 x1=x;
23 y1=y;
24 x2(1)=2;
25 for n = 2:T
26     x2(n) = 2;
27     y2(n) = x2(n-1);
28 end
29 z=y1+y2;
30 for n = 2:T
31     y3(n) = (x2(n-1)+x1(n-1));
32 end
33 if z==y3 then
34     disp('linear')
35 else
36     disp("nonlinear")
37 end
38 Ip = x(T-s);
39 Op = y(T-s);
40 if(Ip == Op)

```

```

41     disp(' Time In-variant system ');
42 else
43     disp('Time Variant system ');
44 end
45 Max =20;
46 dd=1;
47 for n=2:T
48     if y(n)>Max then
49         dd=0
50     end
51 end
52 if dd==0
53     disp('unstable ')
54 else
55     disp('stable ');
56 end

```

---

**Scilab code Exa 1.38** memoryless causal time invariant system

```

1 // ex_38 check if  $y[n] = n \cdot x[n]$  is memoryless ,
   causal , linear , time-varaint
2 clear;
3 clc;
4 s = 2; //shift
5 T = 20; //lenght of signal
6 for n = 1:T
7     x(n) = n;
8     y(n) = n*x(n);
9 end
10 if y(1)==x(1) then
11     disp("memoryless and causal")
12 else
13     disp("noncausal")
14 end
15 x1=x;

```

```

16 y1=y;
17 for n = 1:T
18     x2(n) = 2;
19     y2(n) = n*x2(n);
20 end
21 z=y1+y2;
22 for n = 1:T
23     y3(n) = n*(x2(n)+x1(n));
24 end
25 if z==y3 then
26     disp('linear')
27 else
28     disp("nonlinear")
29 end
30 Ip = x(T-s);
31 Op = y(T-s);
32 if(Ip == Op)
33     disp(' Time In-variant system ');
34 else
35     disp('Time Variant system ');
36 end
37 Max = 50;
38 S = 0;
39 for n=1:T
40     S = S+y(n);
41 end
42 if (S >Max)
43     disp('unstable')
44 else
45     disp('stable');
46 end

```

---

Scilab code Exa 1.39 time invariancy

```

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

```

```

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

```

---

#### Scilab code Exa 1.41 linearity

```

1  clear all;
2  close;
3  clc;
4  n=-2:4;
5  x1=[0 0 0 0 2 0 0];
6  y1=[0 0 0 0 1 2 0];
7  x2=[0 1 0 0 0 0 0];
8  y2=[0 2 1 0 0 0 0];
9  x3=[0 0 0 1 2 0 0];
10 y3=[0 0 0 2 3 1 0];
11 subplot(3,2,1); plot2d3(n,x1); plot(n,x1,'r. '); xtitle('x1')
12 subplot(3,2,2); plot2d3(n,y1); plot(n,y1,'r. '); xtitle('y1')
13 subplot(3,2,3); plot2d3(n,x2); plot(n,x2,'r. '); xtitle('x2')

```

```

    'x2')
14 subplot(3,2,4);plot2d3(n,y2);plot(n,y2,'r. ');xtitle(
    'y2')
15 subplot(3,2,5);plot2d3(n,x3);plot(n,x3,'r. ');xtitle(
    'x3')
16 subplot(3,2,6);plot2d3(n,y3);plot(n,y3,'r. ');xtitle(
    'y3')
17 disp("it can be seen that  $x3[n]=x1[n]+x2[n-2]$ 
    therefore for linear system  $y3[n]=y1[n]+y2$ 
     $[n-2]$ ")
18 figure
19 subplot(4,1,1);plot2d3(n,y1);plot(n,y1,'r. ');xtitle(
    'y1')
20 subplot(4,1,2);plot2d3(n+2,y2);plot(n+2,y2,'r. ');
    xtitle('y2[n-2]')
21 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]')
22 subplot(4,1,4);plot2d3(n,y3);plot(n,y3,'r. ');xtitle(
    'y3[n]')
23 disp("from the figure  $y3[n] \neq y1[n]+y2[n-2]$  therefore
    the system is not linear")

```

---

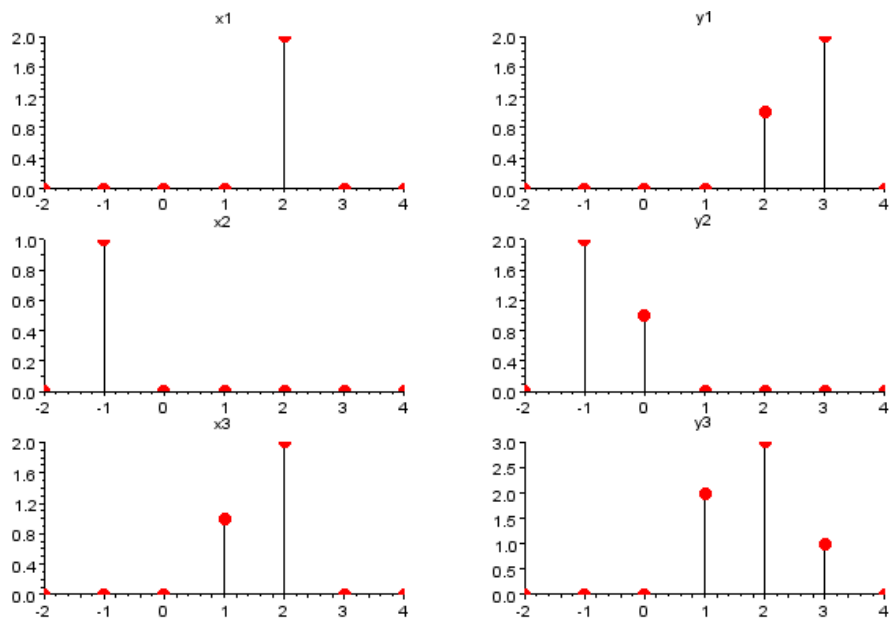


Figure 1.22: linearity

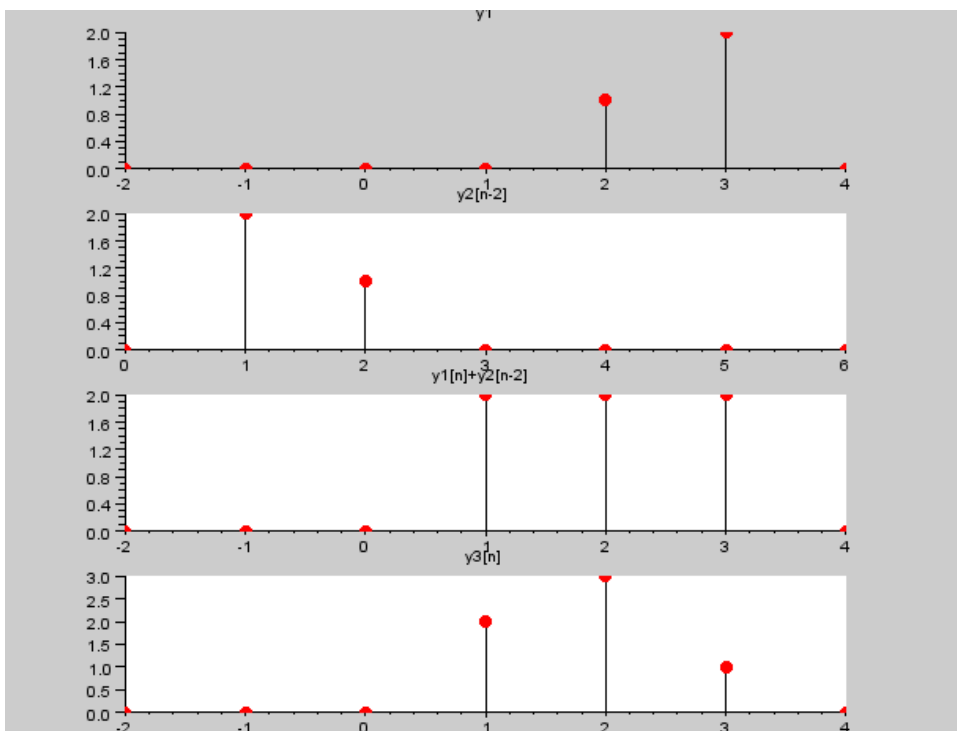


Figure 1.23: linearity

## Chapter 2

# Linear Time Invariant Systems

Scilab code Exa 2.4 output response

```
1 //Example 2.4: Convolution Integral
2 clear;
3 close;
4 clc;
5 t = -5:1/100:5;
6 for i=1:length(t)
7     if t(i)<0 then
8         h(i)=0;
9         x(i)=0;
10    else
11        h(i)=exp(-t(i));
12        x(i)=1;
13    end
14 end
15 y = convol(x,h)./100;
16 figure
17 a=gca();
18 a.x_location="origin";
19 plot2d(t,h)
20 xtitle('Impulse Response ', 't', 'h(t)');
21 a.thickness = 2;
```



```

22 figure
23 a=gca();
24 plot2d(t,x)
25 xtitle('Input Response','t','x(t)');
26 a.thickness = 2;
27 figure
28 a=gca();
29 t1=-10:1/100:10;
30 a.y_location="origin";
31 a.x_location="origin";
32 d=find(t1==5);
33 plot2d(t1(1:d),y(1:d))
34 xtitle('Output Response','t','y(t)');
35 a.thickness = 2;

```

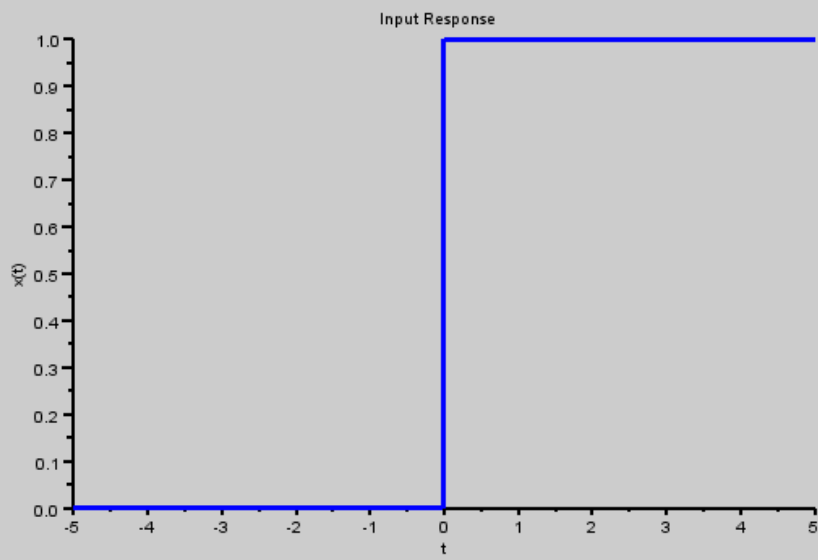
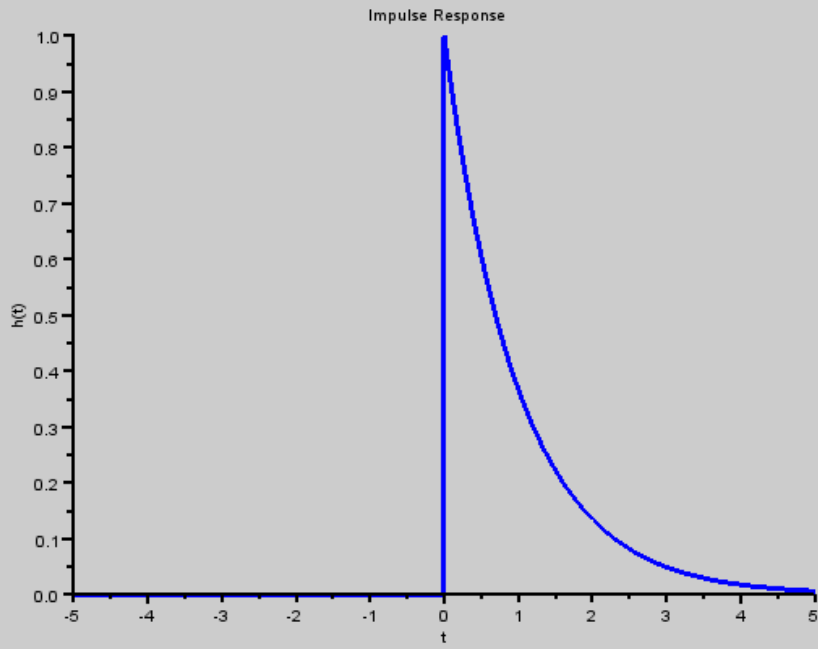
---

### Scilab code Exa 2.5 output response

```

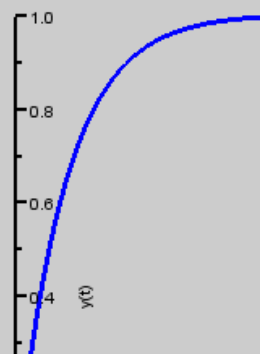
1 //Example 2.4: Convolution Integral
2 clear;
3 close;
4 clc;
5 t =-5:1/100:5;
6 for i=1:length(t)
7     if t(i)<0 then
8         h(i)=0;
9         x(i)=exp(t(i));
10    else
11        h(i)=exp(-t(i));
12        x(i)=0;
13    end
14 end

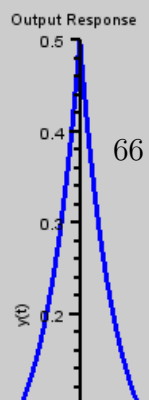
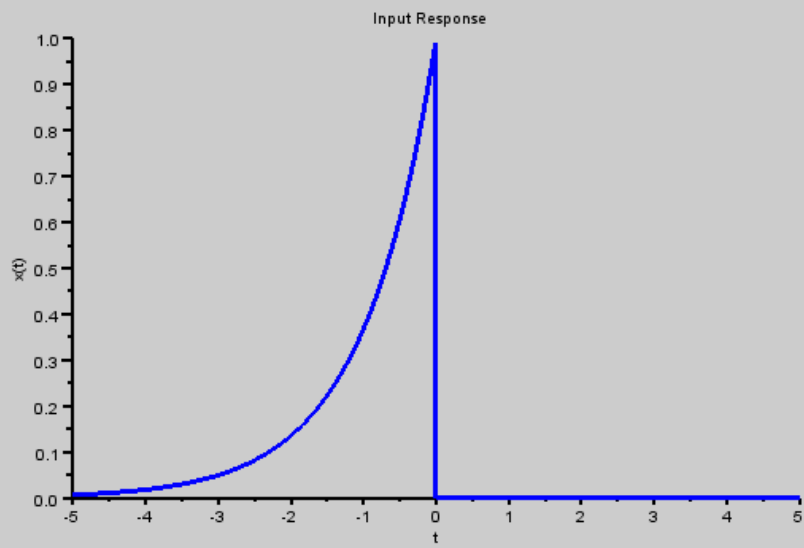
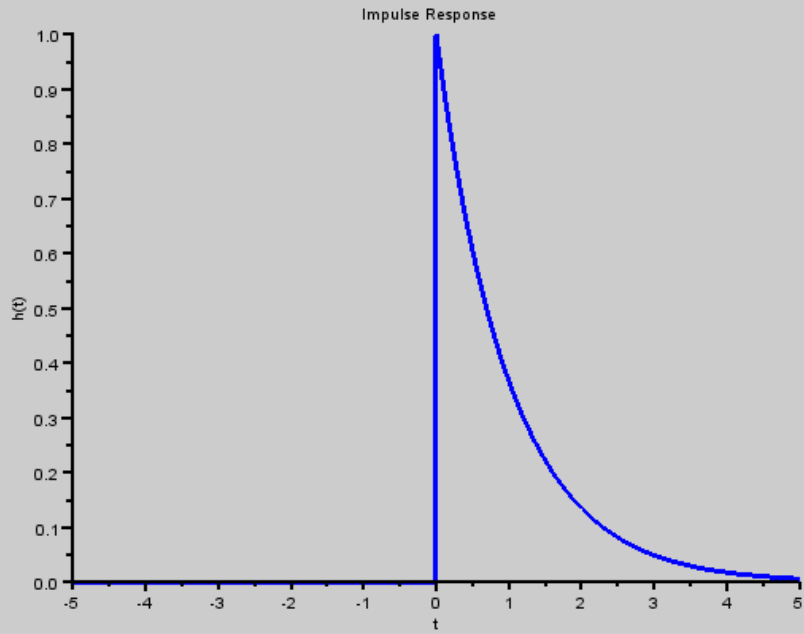
```



### Output Response

65





```

15 y = convol(x,h)./100;
16 figure
17 a=gca();
18 a.x_location="origin";
19 plot2d(t,h)
20 xtitle('Impulse Response','t','h(t)');
21 a.thickness = 2;
22 figure
23 a=gca();
24 plot2d(t,x)
25 xtitle('Input Response','t','x(t)');
26 a.thickness = 2;
27 figure
28 a=gca();
29 t1=-10:1/100:10;
30 a.y_location="origin";
31 plot2d(t1,y)
32 xtitle('Output Response','t','y(t)');
33 a.thickness = 2;

```

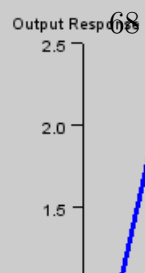
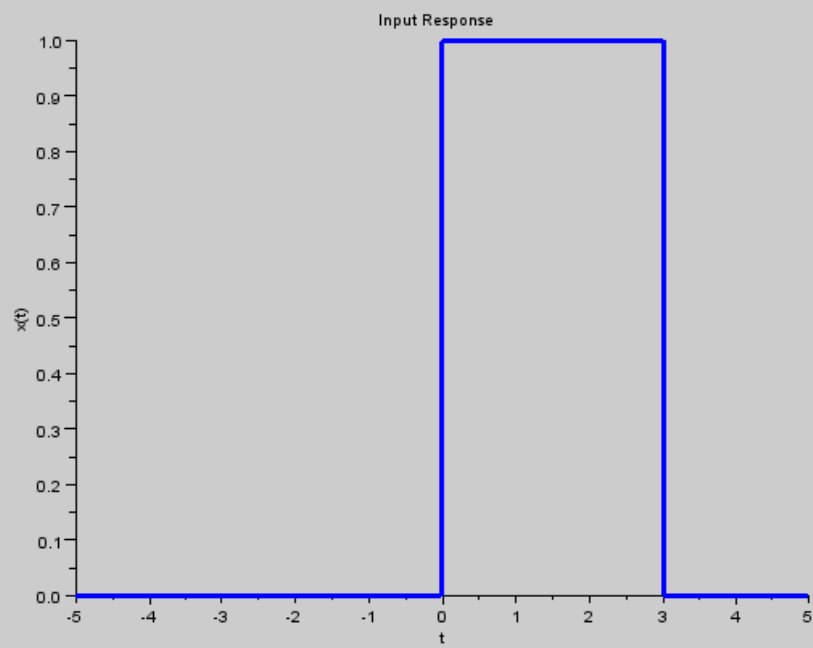
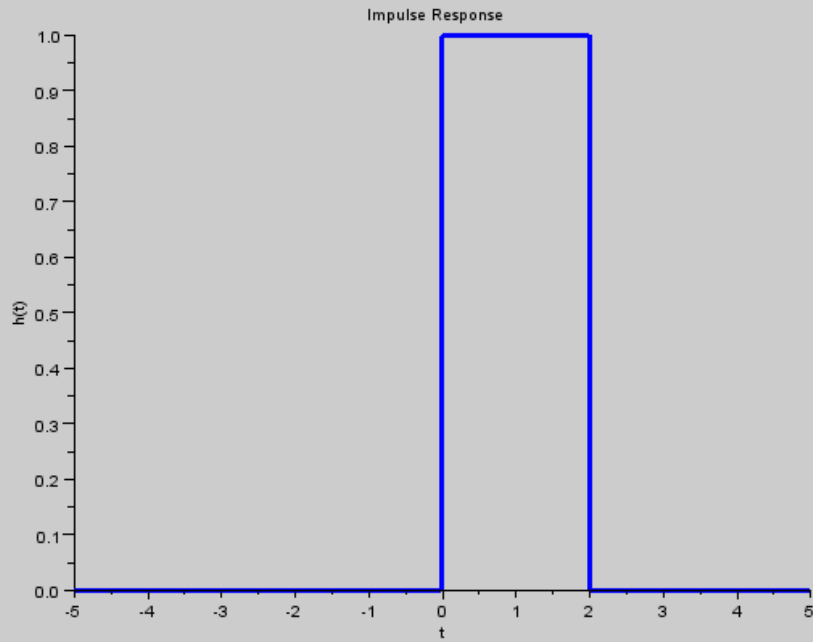
---

### Scilab code Exa 2.6 convolution

```

1 //Example 2.6: Convolution Integral
2 clear;
3 close;
4 clc;
5 t =-5:1/100:5;
6 for i=1:length(t)
7     if t(i)<0 then
8         h(i)=0;
9         x(i)=0;
10    elseif t(i)<=2
11        h(i)=1;

```

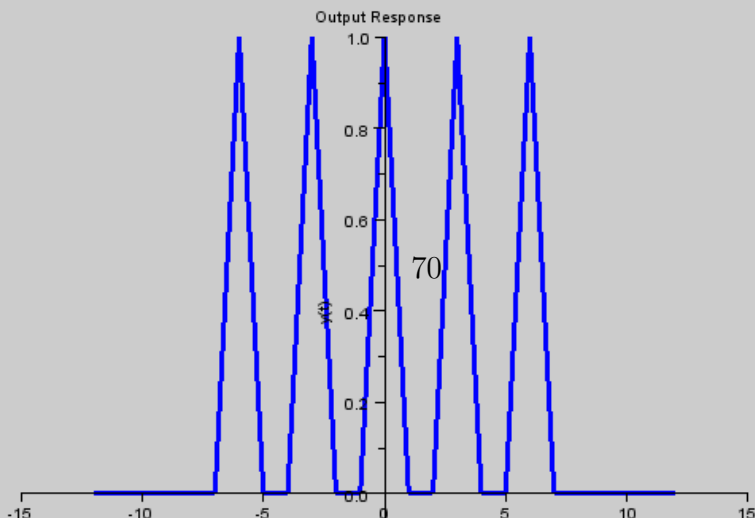
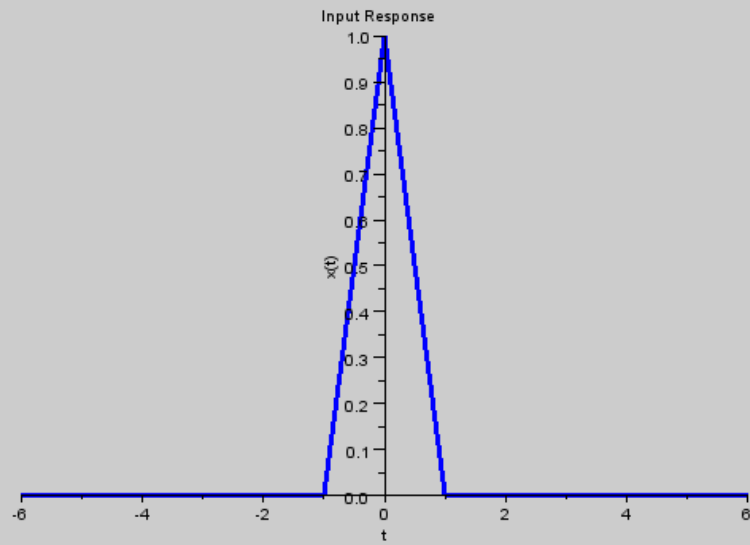
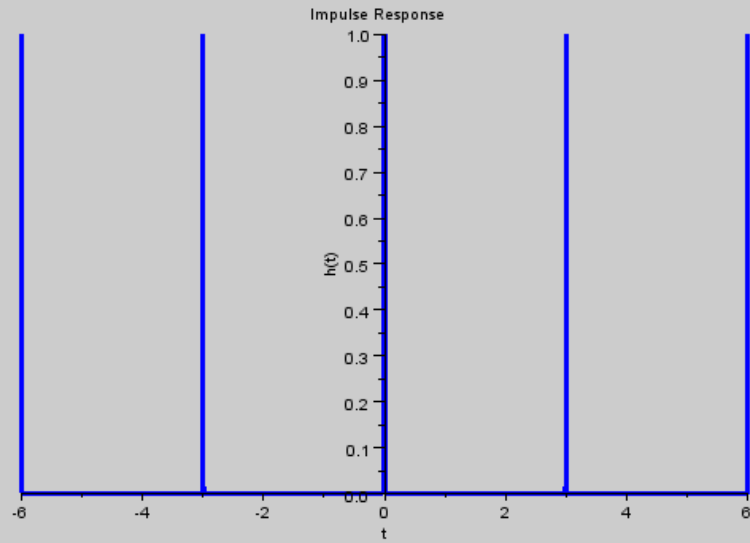


```

12         x(i)=1;
13     elseif t(i)<=3
14         h(i)=0;
15         x(i)=1;
16     else
17         h(i)=0;
18         x(i)=0;
19     end
20 end
21 y = convol(x,h)./100;
22 figure
23 a=gca();
24 a.x_location="origin";
25 plot2d(t,h)
26 xtitle('Impulse Response','t','h(t)');
27 a.children.children.thickness = 3;
28 a.children.children.foreground= 2;
29 figure
30 a=gca();
31 plot2d(t,x)
32 xtitle('Input Response','t','x(t)');
33 a.children.children.thickness = 3;
34 a.children.children.foreground= 2;
35 figure
36 a=gca();
37 t1=-10:1/100:10;
38 a.y_location="origin";
39 a.x_location="origin";
40 plot2d(t1,y)
41 xtitle('Output Response','t','y(t)');
42 a.children.children.thickness = 3;
43 a.children.children.foreground= 2;

```

---



70

Scilab code Exa 2.7.a convolution of two rectangular pulse

```
1 //Example 2.7: Convolution Integral
2 clear;
3 close;
4 clc;
5 t = -6:1/100:6;
6 for i=1:length(t)
7     if modulo(t(i),3)==0 then
8         h(i)=1;
9     else
10        h(i)=0;
11    end
12    if t(i)<-1 then
13        x(i)=0;
14    elseif t(i)<0
15        x(i)=1+t(i);
16    elseif t(i)<1
17        x(i)=1-t(i);
18    else
19        x(i)=0;
20    end
21 end
22 y = convol(x,h);
23 figure
24 a=gca();
25 a.x_location="origin";
26 a.y_location="origin";
27 plot2d(t,h)
28 xtitle('Impulse Response','t','h(t)');
29 a.children.children.thickness = 3;
30 a.children.children.foreground= 2;
31 figure
32 a=gca();
33 a.x_location="origin";
34 a.y_location="origin";
35 plot2d(t,x)
36 xtitle('Input Response','t','x(t)');
```



```

37 a.children.children.thickness = 3;
38 a.children.children.foreground= 2;
39 figure
40 a=gca();
41 t1=-12:1/100:12;
42 a.y_location="origin";
43 a.x_location="origin";
44 plot2d(t1,y)
45 xtitle('Output Response','t','y(t)');
46 a.children.children.thickness = 3;
47 a.children.children.foreground= 2;

```

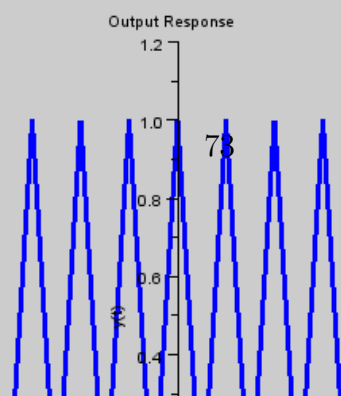
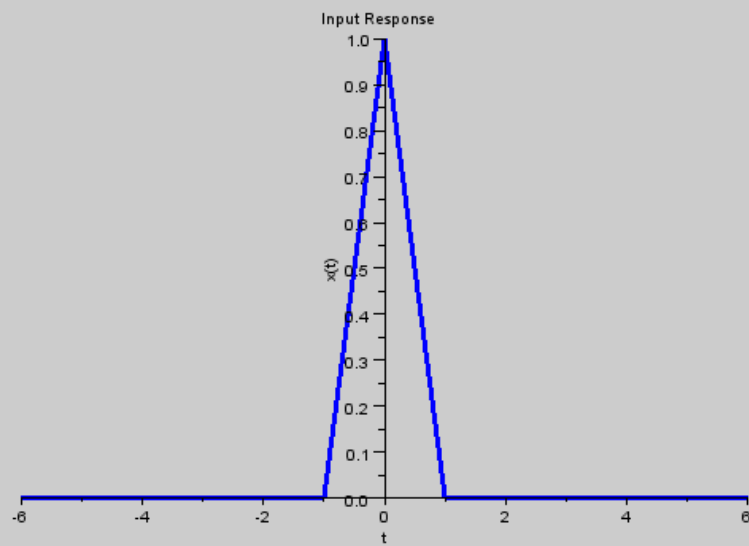
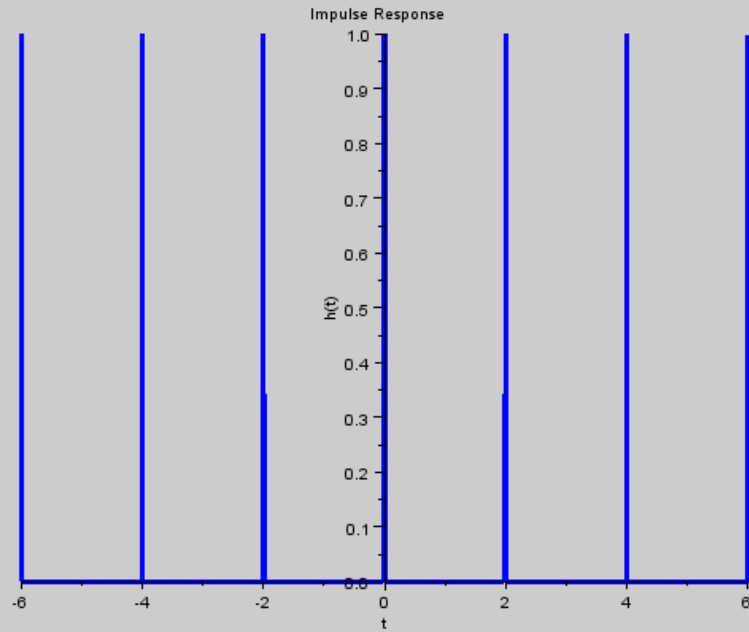
---

Scilab code Exa 2.7.b convolution of two rectangular pulse

```

1 //Example 2.7: Convolution Integral
2 clear;
3 close;
4 clc;
5 t =-6:1/100:6;
6 for i=1:length(t)
7     if modulo(t(i),2)==0 then
8         h(i)=1;
9     else
10        h(i)=0;
11    end
12    if t(i)<-1 then
13        x(i)=0;
14    elseif t(i)<0
15        x(i)=1+t(i);
16    elseif t(i)<1
17        x(i)=1-t(i);
18    else
19        x(i)=0;

```



73

```

20     end
21 end
22 y = convol(x,h);
23 figure
24 a=gca();
25 a.x_location="origin";
26 a.y_location="origin";
27 plot2d(t,h)
28 xtitle('Impulse Response','t','h(t)');
29 a.children.children.thickness = 3;
30 a.children.children.foreground= 2;
31 figure
32 a=gca();
33 a.x_location="origin";
34 a.y_location="origin";
35 plot2d(t,x)
36 xtitle('Input Response','t','x(t)');
37 a.children.children.thickness = 3;
38 a.children.children.foreground= 2;
39 figure
40 a=gca();
41 t1=-12:1/100:12;
42 a.y_location="origin";
43 a.x_location="origin";
44 plot2d(t1,y)
45 xtitle('Output Response','t','y(t)');
46 a.children.children.thickness = 3;
47 a.children.children.foreground= 2;

```

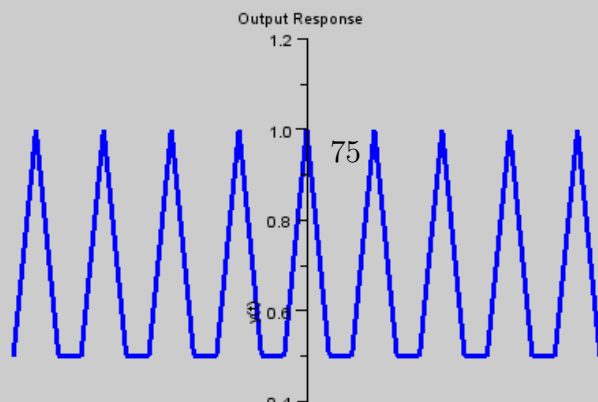
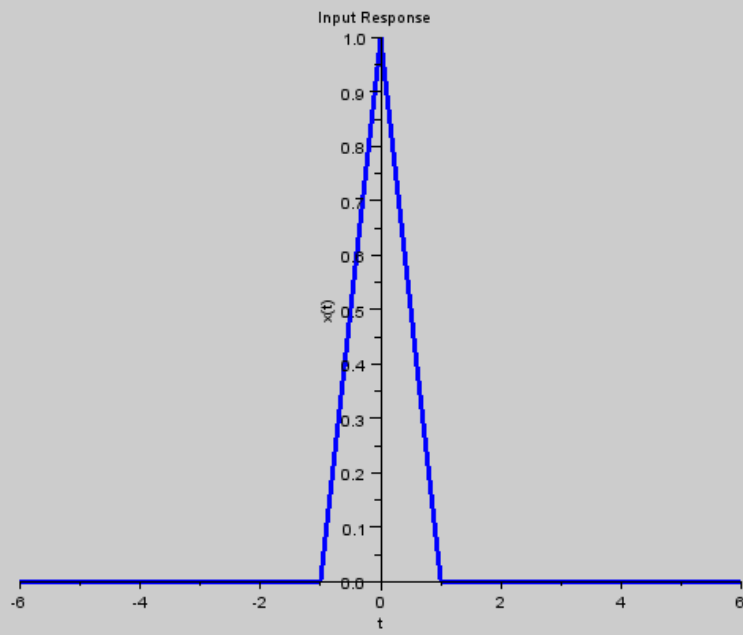
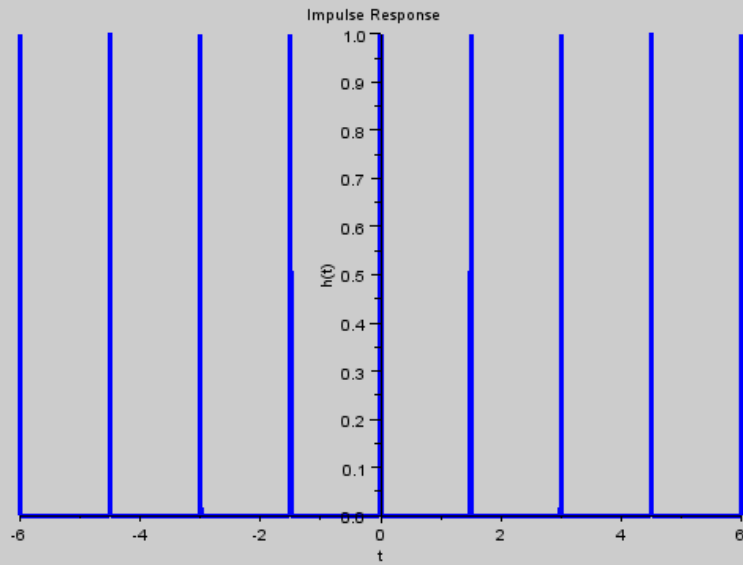
---

Scilab code Exa 2.7.c convolution of two rectangular pulse

```

1 //Example 2.7: Convolution Integral
2 clear;

```



```

3 close;
4 clc;
5 t = -6:1/100:6;
6 for i=1:length(t)
7     if modulo(t(i),1.5)==0 then
8         h(i)=1;
9     else
10        h(i)=0;
11    end
12    if t(i)<-1 then
13        x(i)=0;
14    elseif t(i)<0
15        x(i)=1+t(i);
16    elseif t(i)<1
17        x(i)=1-t(i);
18    else
19        x(i)=0;
20    end
21 end
22 y = convol(x,h);
23 figure
24 a=gca();
25 a.x_location="origin";
26 a.y_location="origin";
27 plot2d(t,h)
28 xtitle('Impulse Response','t','h(t)');
29 a.children.children.thickness = 3;
30 a.children.children.foreground= 2;
31 figure
32 a=gca();
33 a.x_location="origin";
34 a.y_location="origin";
35 plot2d(t,x)
36 xtitle('Input Response','t','x(t)');
37 a.children.children.thickness = 3;
38 a.children.children.foreground= 2;
39 figure
40 a=gca();

```

```

41 t1=-12:1/100:12;
42 a.y_location=" origin";
43 a.x_location=" origin";
44 b=find(t1==6.5);
45 c=find(t1==-6.5);
46 plot2d(t1(c:b),y(c:b))
47 xtitle('Output Response','t','y(t)');
48 a.children.children.thickness = 3;
49 a.children.children.foreground= 2;

```

---

### Scilab code Exa 2.8 periodic convolution

```

1 //Example 2.8: Convolution Integral
2 clear;
3 close;
4 clc;
5 t =-4*10:1/100:4*10;
6 t2=-4:1/100:0;
7 for i=1:length(t2)
8     if t2(i)<-2 then
9         x(i)=1;
10        else
11            x(i)=0;
12        end
13 end
14 fac=ceil(length(t)/length(t2));
15 s=[];
16 for i=1:fac;
17     s=[s ;x];
18 end
19 y = convol(s,s)./2000;
20 figure
21 a=gca();
22 a.x_location=" origin";
23 a.y_location=" origin";

```

```

24 b=find(t==8);
25 c=find(t==-8);
26 plot2d(t(c:b),s(c:b))
27 xtitle('Input Response','t','x(t)');
28 a.children.children.thickness = 3;
29 a.children.children.foreground= 2;
30 figure
31 a=gca();
32 t1=-8*10:1/100:8*10;
33 a.y_location="origin";
34 a.x_location="origin";
35 b=find(t1==8);
36 c=find(t1==-8);
37 plot2d(t1(c:b),y(c:b))
38 xtitle('Output Response','t','y(t)');
39 a.children.children.thickness = 3;
40 a.children.children.foreground= 2;

```

---

Scilab code Exa 2.9 output response

```

1 //Example 2.9: Convolution Integral
2 clear;
3 close;
4 clc;
5 t =-2:1/100:2;
6 for i=1:length(t)
7     if t(i)<-1|t(i)>1 then
8         x(i)=0;
9     else
10        x(i)=1;
11    end

```

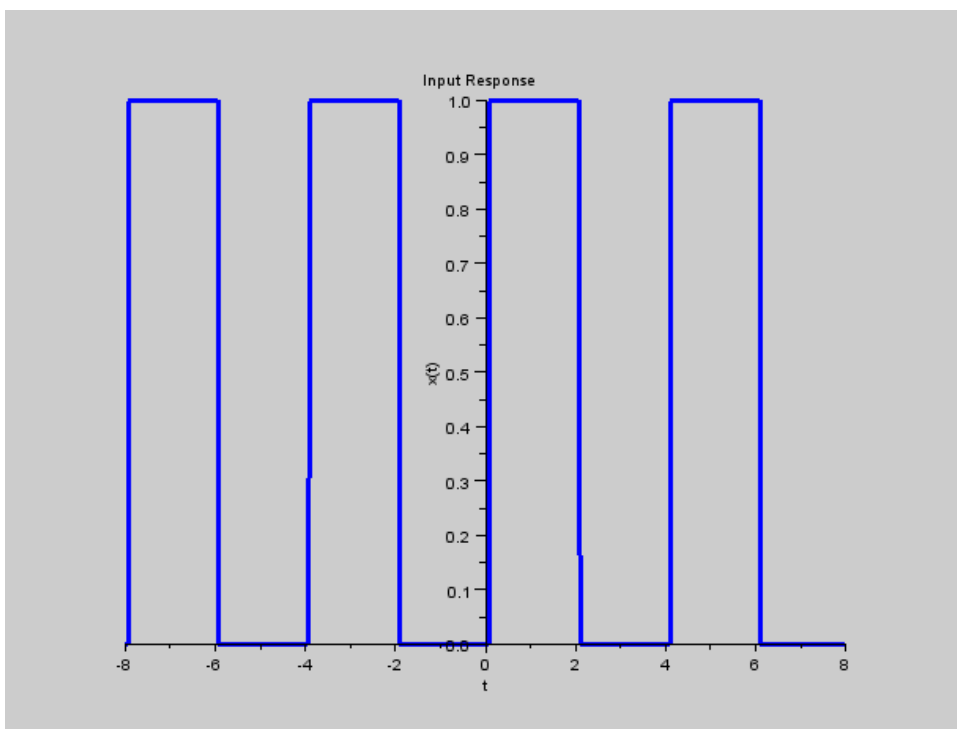


Figure 2.7: periodic convolution



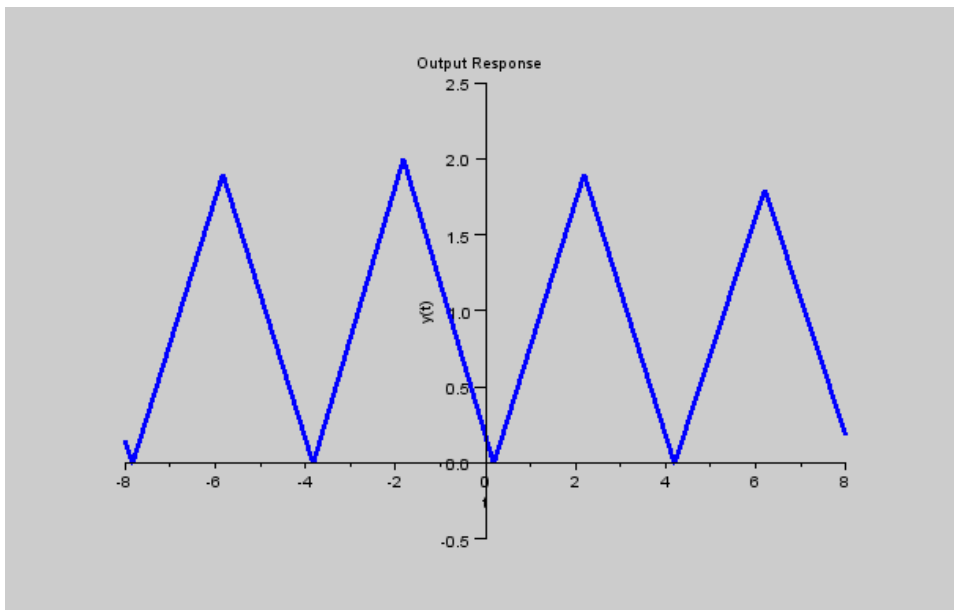


Figure 2.8: periodic convolution

```

12 end
13 ty=-2:1/100:4;
14 for i=1:length(ty)
15     if ty(i)<-1|ty(i)>3 then
16         y(i)=0;
17     elseif ty(i)<1
18         y(i)=1+ty(i);
19     else
20         y(i)=3-ty(i);
21     end
22 end
23 figure
24 a=gca();
25 a.x_location="origin";
26 a.y_location="origin";
27 plot2d(t,x)
28 xtitle('Input Response','t','x(t)');
29 a.children.children.thickness = 3;

```

```

30 a.children.children.foreground= 2;
31 figure
32 a=gca();
33 a.y_location=" origin";
34 a.x_location=" origin";
35 plot2d(ty,y)
36 xtitle('Output Response ', 't', 'y(t)');
37 a.children.children.thickness = 3;
38 a.children.children.foreground= 2;
39 //since it is a time invariant system for x(t-2) o/p
    is y(t-2)
40 ty1=ty+2;
41 figure
42 a=gca();
43 a.y_location=" origin";
44 a.x_location=" origin";
45 plot2d(ty1,y)
46 xtitle('Output Response ', 't', 'y(t-2)');
47 a.children.children.thickness = 3;
48 a.children.children.foreground= 2;
49 //since the system is linear ,for x(t)/2 o/p is y(t)
    /2
50 figure
51 a=gca();
52 a.y_location=" origin";
53 a.x_location=" origin";
54 plot2d(ty,y./2)
55 xtitle('Output Response ', 't', '.5*y(t)');
56 a.children.children.thickness = 3;
57 a.children.children.foreground= 2;

```

---

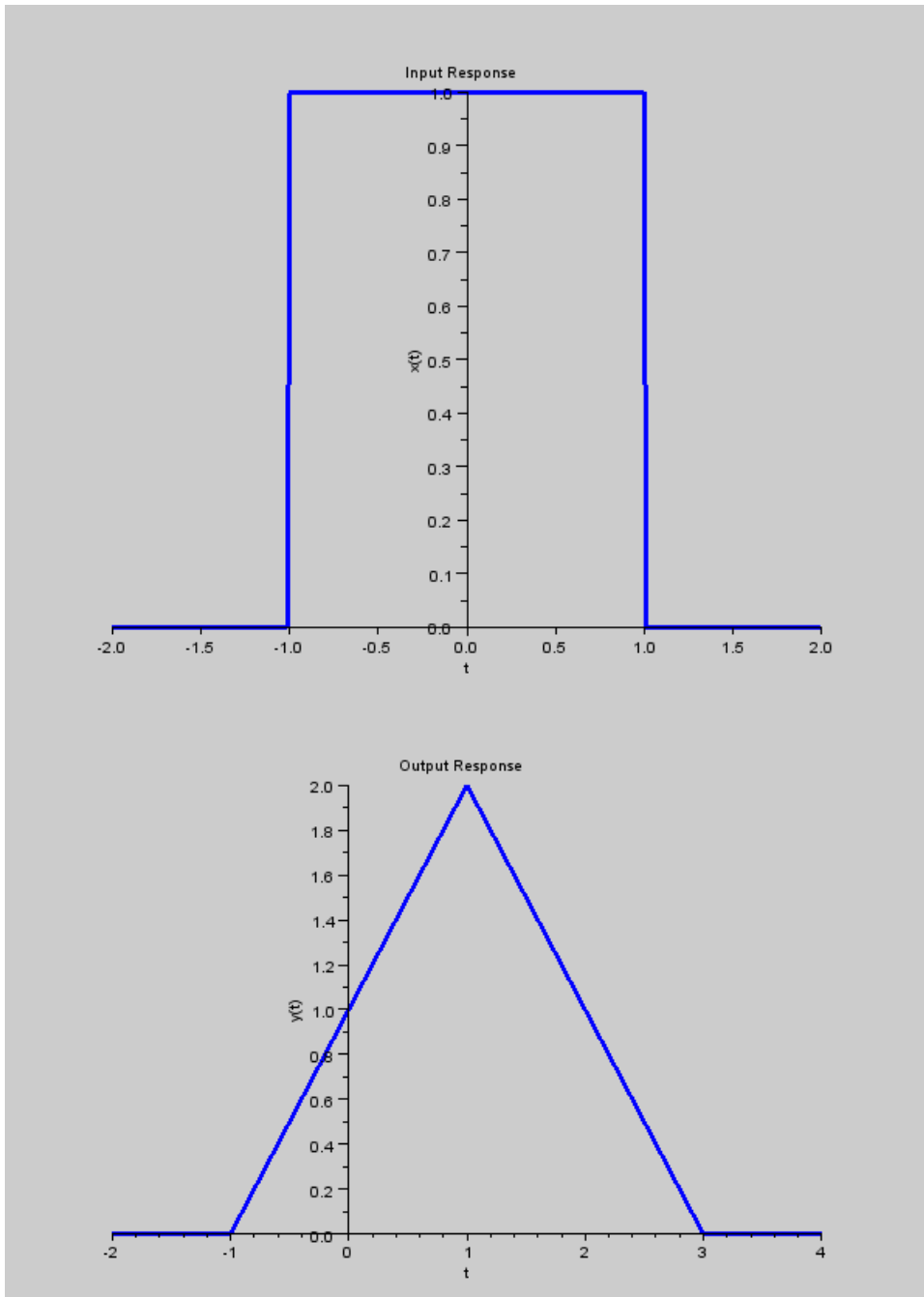


Figure 2.9: output response

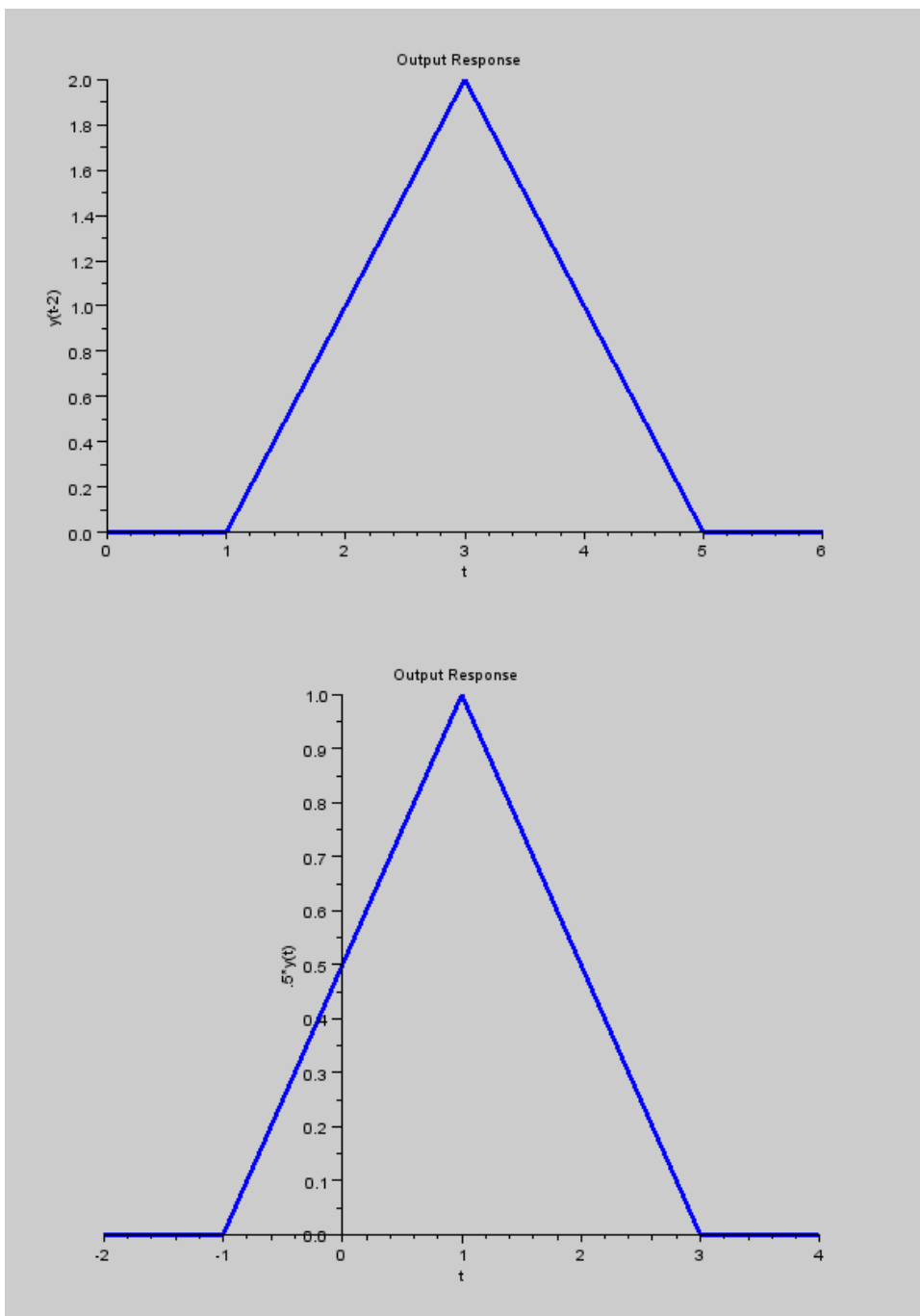
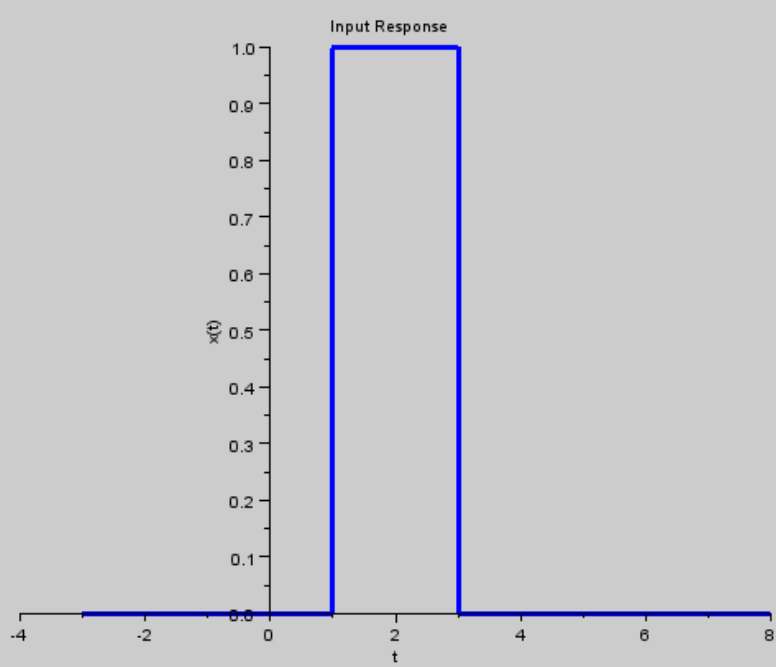
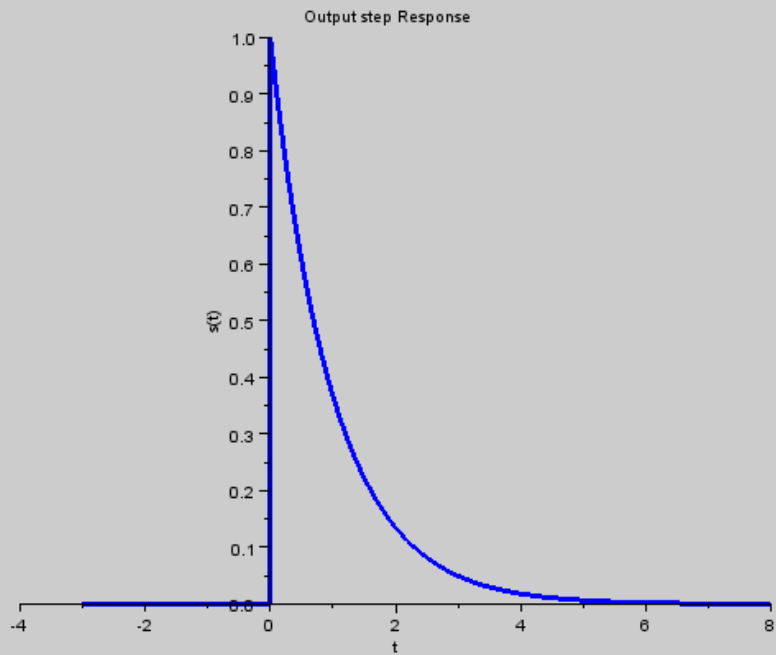
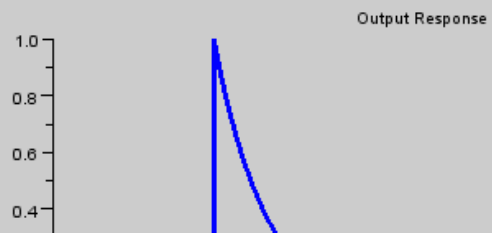


Figure 2.10: output response



84



## Scilab code Exa 2.10 output response

```
1 //Example 2.10: Convolution Integral
2 clear;
3 close;
4 clc;
5 t =-3:1/100:8;
6 s=[];
7 ss=[];
8 for i=1:length(t)
9     if t(i)<1|t(i)>3 then
10         x(i)=0;
11     else
12         x(i)=1;
13     end
14     if t(i)<0 then
15         s(i)=0;
16     else
17         s(i)=exp(-t(i));
18     end
19 end
20 figure
21 a=gca();
22 a.y_location="origin";
23 a.x_location="origin";
24 plot2d(t,s)
25 xtitle('Output step Response','t','s(t)');
26 a.children.children.thickness = 3;
27 a.children.children.foreground= 2;
28 figure
29 a=gca();
30 a.x_location="origin";
31 a.y_location="origin";
32 plot2d(t,x)
```

```

33 xtitle('Input Response', 't', 'x(t)');
34 a.children.children.thickness = 3;
35 a.children.children.foreground= 2;
36 t1=t+1;
37 t2=t+3;
38 s=s';
39 tt=min(min(t1,t2)):1/100:max(max(t1,t2));
40 ee=zeros(tt);
41 x=find(tt==-2);
42 y=find(tt==0);
43 z=find(tt==9);
44 for i=1:1:length(tt)
45     if i<y then
46         ee(i)=s(i);
47     elseif i<z
48         ee(i)=s(i)-s(i-y+1);
49     else
50         ee(i)=-s(i-y+1);
51     end
52 end
53 figure
54 a=gca();
55 a.y_location="left";
56 a.x_location="origin";
57 plot2d(tt,ee)
58 xtitle('Output Response', 't', 's(t-1)-s(t-3)');
59 a.children.children.thickness = 3;
60 a.children.children.foreground= 2;

```

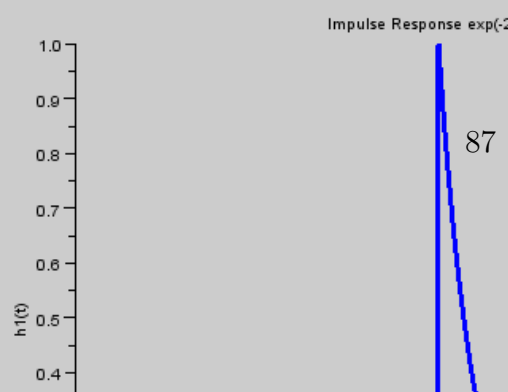
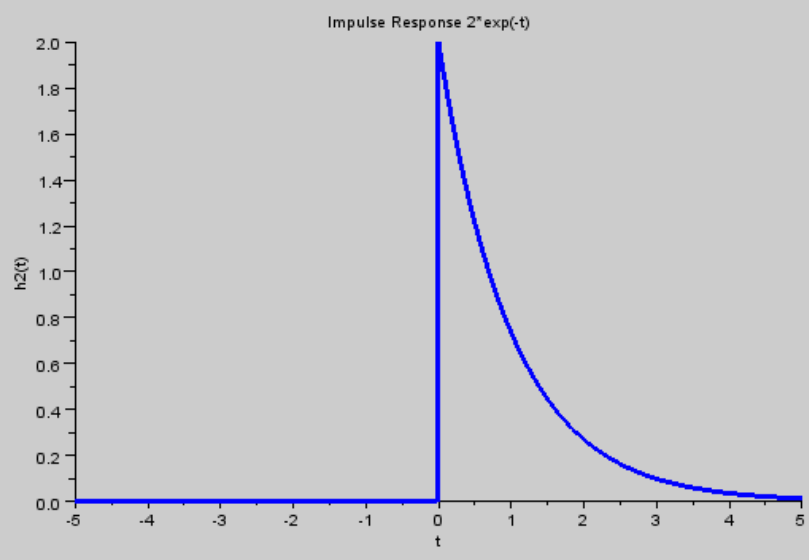
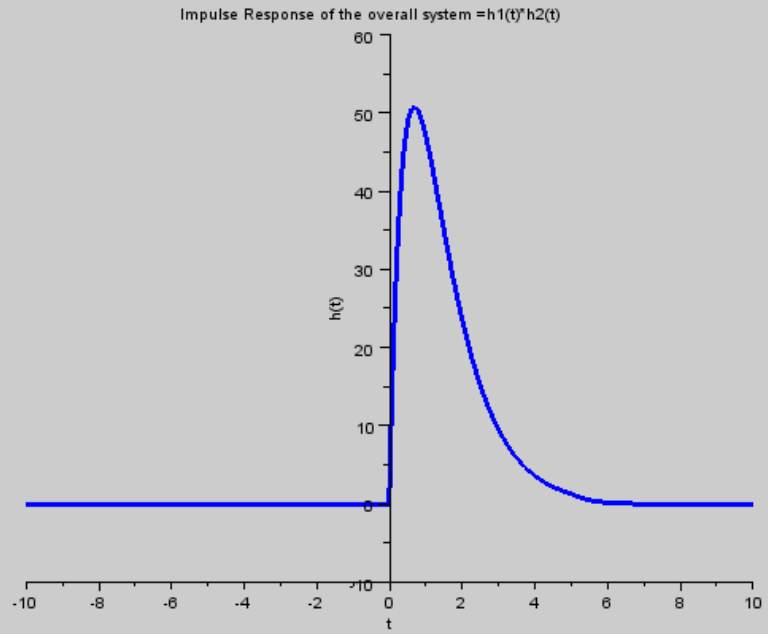
---

Scilab code Exa 2.14.a cascaded system

```

1 //Example 2.14: Convolution Integral
2 clear;

```





```

3 close;
4 clc;
5 t =-5:1/100:5;
6 for i=1:length(t)
7     if t(i)<0 then
8         h1(i)=0;
9         h2(i)=0;
10    else
11        h1(i)=exp(-2.*t(i));
12        h2(i)=2*exp(-t(i));
13    end
14 end
15 h=convol(h1,h2);
16 figure
17 a=gca();
18 a.x_location="origin";
19 plot2d(t,h1)
20 xtitle('Impulse Response exp(-2*t)', 't', 'h1(t)');
21 a.children.children.thickness = 3;
22 a.children.children.foreground= 2;
23 figure
24 a=gca();
25 plot2d(t,h2)
26 xtitle('Impulse Response 2*exp(-t)', 't', 'h2(t)');
27 a.children.children.thickness = 3;
28 a.children.children.foreground= 2;
29 figure
30 a=gca();
31 t1=-10:1/100:10;
32 a.y_location="origin";
33 plot2d(t1,h)
34 xtitle('Impulse Response of the overall system =h1(t
    )*h2(t)', 't', 'h(t)');
35 a.children.children.thickness = 3;
36 a.children.children.foreground= 2;

```

---

### Scilab code Exa 2.14.b BIBO stability

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

---

### Scilab code Exa 2.15 eigenfunction of the system

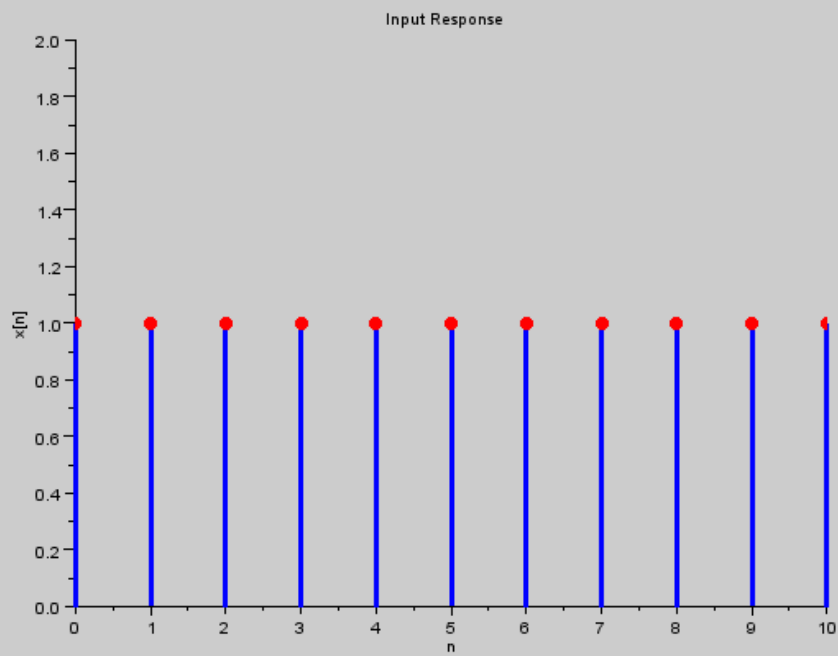
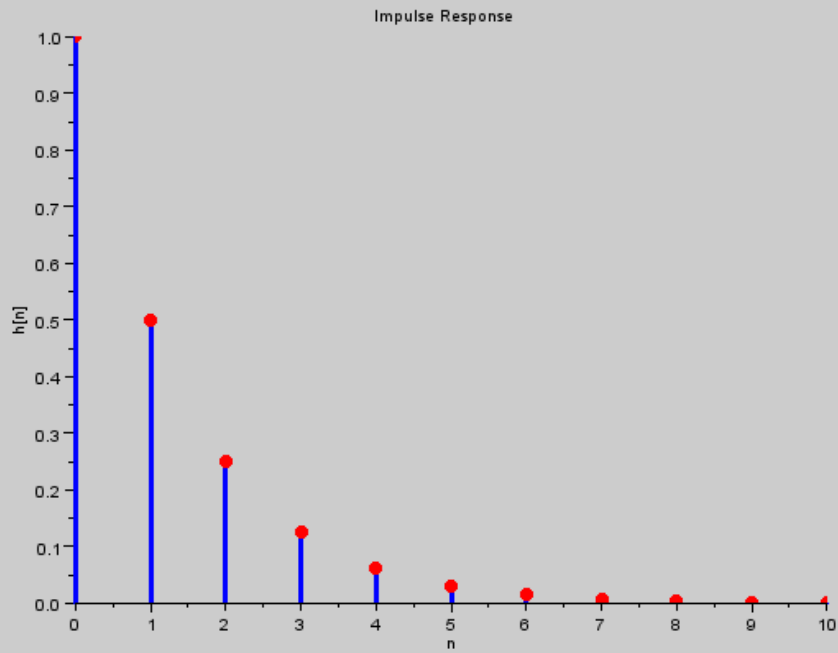
```
1 syms s T t
2 y=integ(exp(-(t-T))*exp(s*T),T,-%inf,t)
3 x=exp(s*t)
4 lamda=y/x//eigen value
5 disp(lamda,"b) lamda=")
6 lamda_=laplace(exp(-t))
7 disp(lamda_,"c) lamda=")
```

---

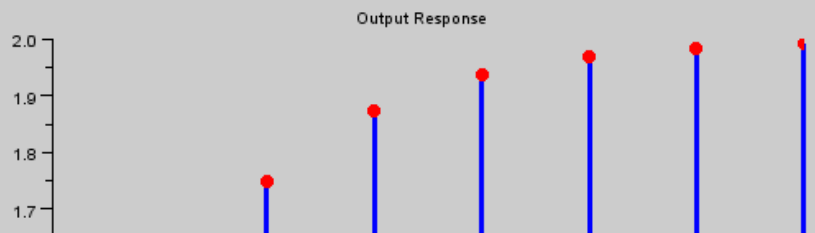
### Scilab code Exa 2.16 eigen value of the system

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

---



90



Scilab code Exa 2.28 output response of a discrete time system

```
1 clear all;
2 clc;
3 n=0:10;
4 alpha=.5;
5 x=ones(n);
6 h=alpha^n;
7 y=convol(x,h);
8 figure
9 a=gca();
10 a.x_location="origin";
11 plot2d3(n,h)
12 plot(n,h,'r. ');
13 xtitle('Impulse Response','n','h[n]');
14 a.children.children.thickness = 3;
15 a.children.children.foreground= 2;
16 figure
17 a=gca();
18 plot2d3(n,x)
19 plot(n,x,'r. ');
20 xtitle('Input Response','n','x[n]');
21 a.children.children.thickness = 3;
22 a.children.children.foreground= 2;
23 figure
24 b=gca();
25 N=0:20;
26 a=find(N==7)
27 plot2d3(N(1:a),y(1:a))
28 plot(N(1:a),y(1:a),'r. ');
29 xtitle('Output Response','n','y[n]');
30 b.children.children.thickness = 3;
31 b.children.children.foreground= 2;
```

---

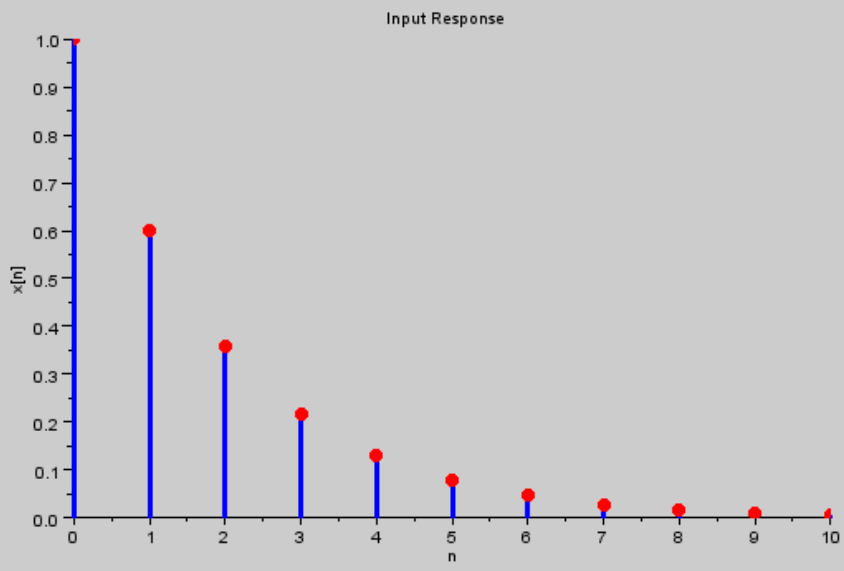
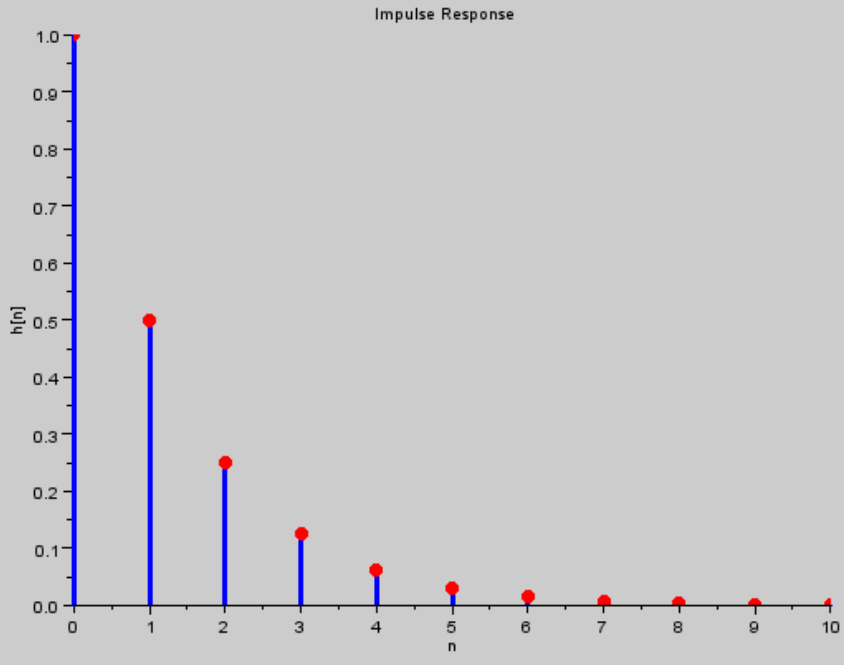
Scilab code Exa 2.29.a convolution of discrete signals

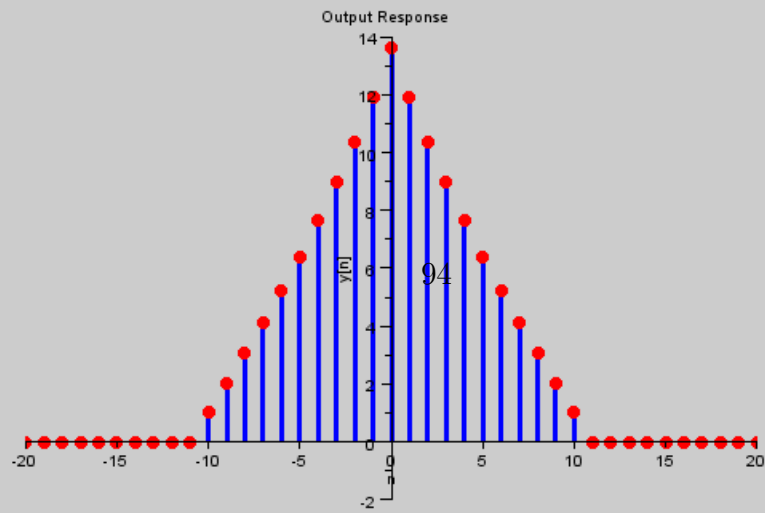
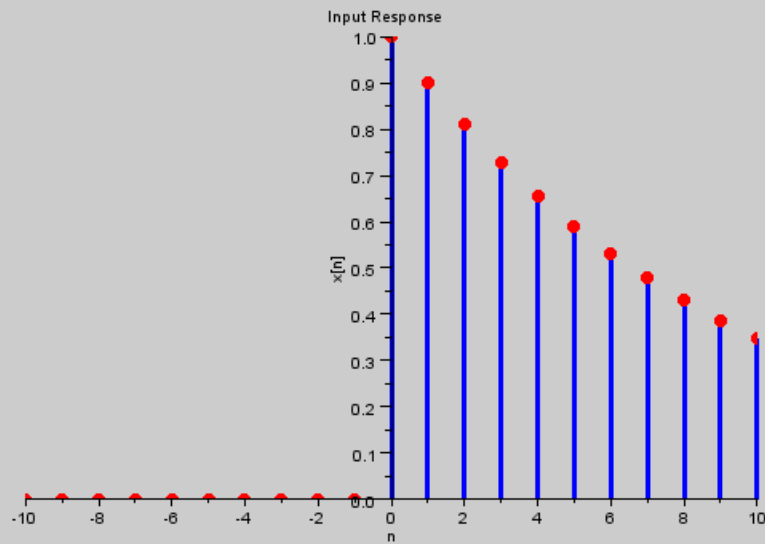
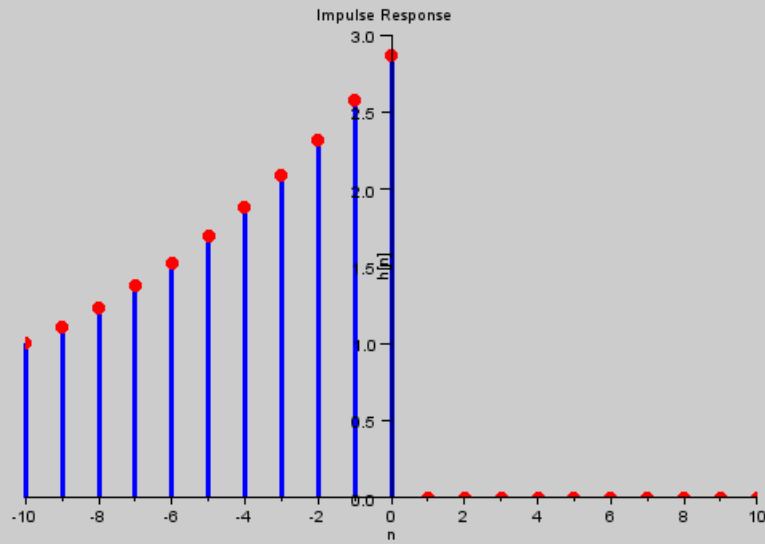
```

1 clear all;
2 clc;
3 n=0:10;
4 alpha=.5;
5 betaa=.6;
6 x=betaa^n;
7 h=alpha^n;
8 y=convol(x,h);
9 figure
10 a=gca();
11 a.x_location="origin";
12 plot2d3(n,h)
13 plot(n,h,'r. ');
14 xtitle('Impulse Response','n','h[n]');
15 a.children.children.thickness = 3;
16 a.children.children.foreground= 2;
17 figure
18 a=gca();
19 plot2d3(n,x)
20 plot(n,x,'r. ');
21 xtitle('Input Response','n','x[n]');
22 a.children.children.thickness = 3;
23 a.children.children.foreground= 2;
24 figure
25 a=gca();
26 N=0:20;
27 plot2d3(N,y)
28 plot(N,y,'r. ');
29 xtitle('Output Response','n','y[n]');
30 a.children.children.thickness = 3;
31 a.children.children.foreground= 2;

```

---





## Scilab code Exa 2.29.b convolution of discrete signals

```
1 clear all;
2 clc;
3 n=0:10;
4 alpha=.9;
5 x=[ zeros(1,length(n)-1) alpha^n];
6 h=[alpha^-n];
7 h=[h zeros(1,length(n)-1)];
8 y=convol(x,h);
9 figure
10 a=gca();
11 a.x_location="origin";
12 n=-10:10;
13 plot2d3(n,h)
14 plot(n,h,'r. ');
15 xtitle('Impulse Response','n','h[n]');
16 a.children.children.thickness = 3;
17 a.children.children.foreground= 2;
18 a.y_location="origin";
19 figure
20 a=gca();
21 plot2d3(n,x)
22 plot(n,x,'r. ');
23 a.y_location="origin";
24 a.x_location="origin";
25 xtitle('Input Response','n','x[n]');
26 a.children.children.thickness = 3;
27 a.children.children.foreground= 2;
28 figure
29 a=gca();
30 N=-20:20;
31 plot2d3(N,y)
32 plot(N,y,'r. ');
33 a.x_location="origin";
34 a.y_location="origin";
35 xtitle('Output Response','n','y[n]');
36 a.children.children.thickness = 3;
```

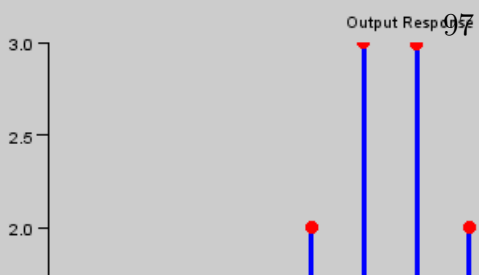
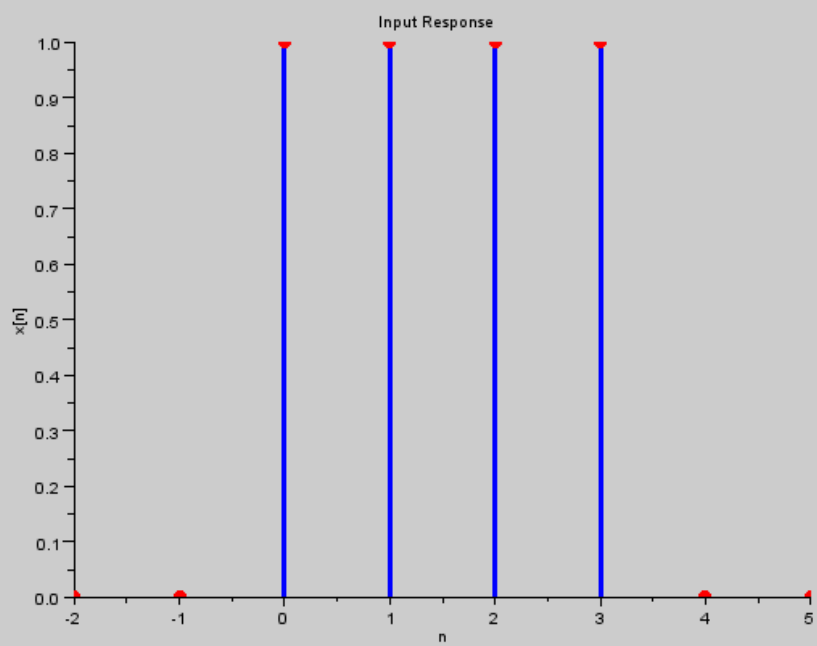
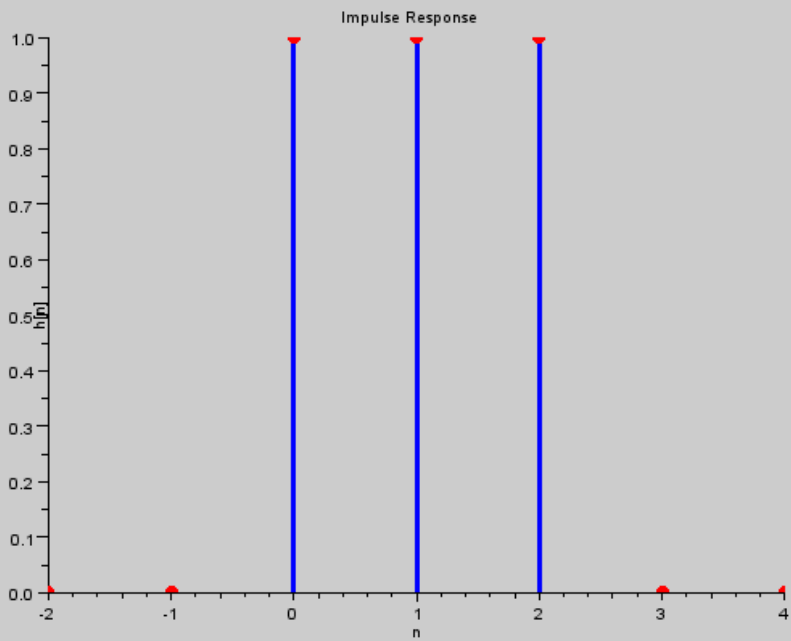


```
37 a.children.children.foreground= 2;
```

---

**Scilab code Exa 2.30** convolution of discrete signals

```
1 clear all;
2 clc;
3 x=[0 0 1 1 1 1 0 0];
4 h=[0 0 1 1 1 0 0];
5 nx=-2:length(x)-3;
6 nh=-2:length(h)-3;
7 y=convol(x,h);
8 ny=min(nx)+min(nh):max(nx)+max(nh);
9 figure
10 a=gca();
11 a.x_location="origin";
12 plot2d3(nh,h)
13 plot(nh,h,'r. ');
14 xtitle('Impulse Response','n','h[n]');
15 a.children.children.thickness = 3;
16 a.children.children.foreground= 2;
17 a.y_location="left";
18 figure
19 a=gca();
20 plot2d3(nx,x)
21 plot(nx,x,'r. ');
22 a.y_location="left";
23 a.x_location="origin";
24 xtitle('Input Response','n','x[n]');
25 a.children.children.thickness = 3;
26 a.children.children.foreground= 2;
27 figure
28 a=gca();
29 plot2d3(ny,y)
```



```

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

```

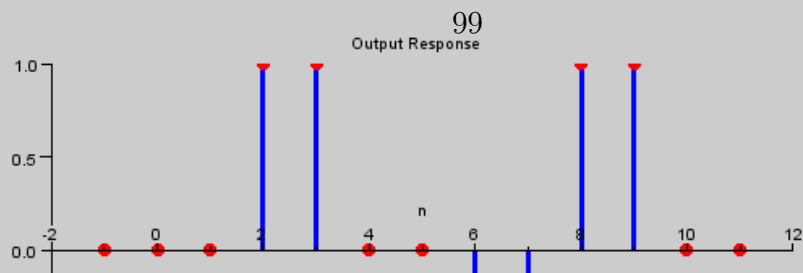
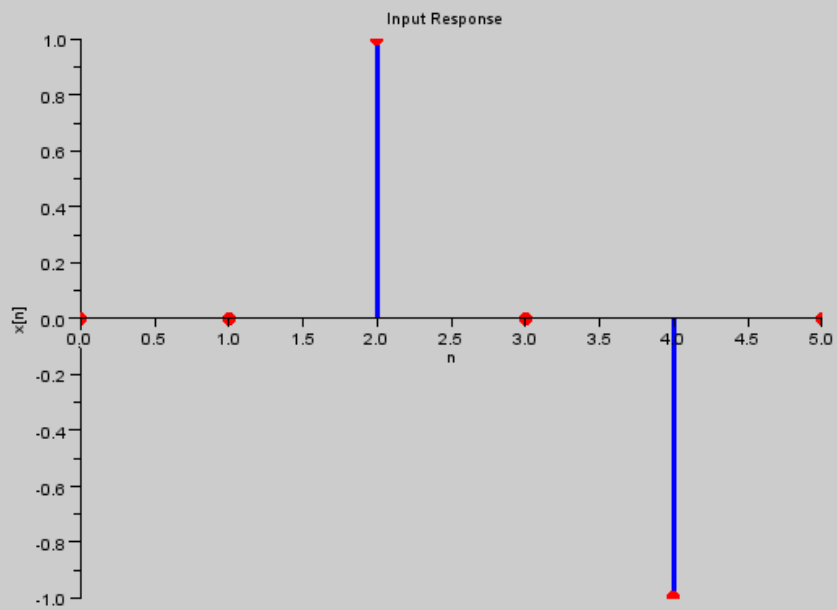
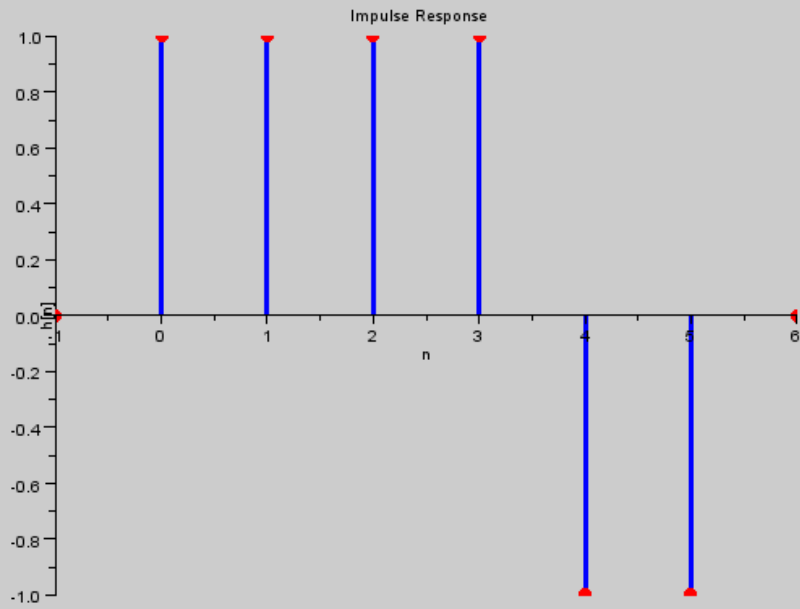
---

Scilab code Exa 2.34 output response without using convolution

```

1 clear all;
2 clc;
3 h=[0 1 1 1 1 -1 -1 0];
4 x=[0 0 1 0 -1 0];
5 nx=0:length(x)-1;
6 nh=-1:length(h)-2;
7 //y=convol(x,h);
8 ny=min(nx)+min(nh):max(nx)+max(nh);
9 //or x[n]=delta[n-2]-delta[n-4] therefore y[n]=h[n
    -2]-h[n-4]
10 n1=nh+2;
11 n2=nh+4;
12 ny=min(nx)+min(nh):max(nx)+max(nh);
13 j=1;
14 k=1;
15 h2=zeros(ny);
16 h4=h2;
17 a=find(ny==n1(1))
18 for j=1:length(nh)
19     h2(a+j-1)=h(j)
20     end
21 a=find(ny==n2(1))
22 for j=1:length(nh)
23     h4(a+j-1)=h(j)

```



```

24         end
25 y=h2-h4;
26 figure
27 a=gca();
28 a.x_location="origin";
29 plot2d3(nh,h)
30 plot(nh,h,'r. ');
31 xtitle('Impulse Response','n','h[n]');
32 a.children.children.thickness = 3;
33 a.children.children.foreground= 2;
34 a.y_location="left";
35 figure
36 a=gca();
37 plot2d3(nx,x)
38 plot(nx,x,'r. ');
39 a.y_location="left";
40 a.x_location="origin";
41 xtitle('Input Response','n','x[n]');
42 a.children.children.thickness = 3;
43 a.children.children.foreground= 2;
44 figure
45 a=gca();
46 plot2d3(ny,y)
47 plot(ny,y,'r. ');
48 a.x_location="origin";
49 a.y_location="left";
50 xtitle('Output Response','n','y[n]');
51 a.children.children.thickness = 3;
52 a.children.children.foreground= 2;

```

---

### Scilab code Exa 2.36 causality

```

1 clear all;
2 clc;
3 n=-5:5;

```

```

4 for i=1:length(n)
5     if(n(i)>=-1)
6         h(i)=2^-(n(i)+1);
7     else
8         h(i)=0;
9     end
10 end
11 causal=%t;
12 for i=1:length(n)
13     if n(i)<0 & h(i)~=0 then
14         causal=%f;
15     end
16 end
17 disp(causal,"the statement that the system is causal
        is");

```

---

**Scilab code Exa 2.38** BIBO stability and causality

```

1 clear all;
2 clc;
3 n=-5:5;
4 alpha=.6;
5 for i=1:length(n)
6     if(n(i)>=0)
7         h(i)=alpha^n(i);
8     else
9         h(i)=0;
10    end
11 end
12 causal=%t;
13 for i=1:length(n)
14     if n(i)<0 & h(i)~=0 then
15         causal=%f;
16     end
17 end

```

```
18 disp(causal,"the statement that the system is causal
    is");
19 n=0:100000;
20 for i=1:length(n)
21     if(n(i)>=0)
22         h(i)=alpha^n(i);
23     else
24         h(i)=0;
25     end
26 end
27 bibo=sum(h);
28 if (bibo<%inf) then
29     disp("system is bibo stable");
30 else
31     disp("system is not stable");
32 end
```

---

# Chapter 3

## Laplace transform and continuous time LTI systems

Scilab code Exa 3.1.a laplace transform

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

---

Scilab code Exa 3.1.b laplace transform

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

---

Scilab code Exa 3.3 laplace transform

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

---



### Scilab code Exa 3.5 pole zero plot

```
1 clc
2 syms t s
3 s1=%s;
4 x=laplace(exp(-2*t)+exp(-3*t),t,s)
5 y=laplace(exp(-3*t)-%e^(2*t))
6 z=laplace(%e^(2*t)-%e^(-3*t))
7 disp(z,y,x,"laplace transform of a b c is")
8 x=1/(s1+2)+1/(s1+3);
9 plzr(x)
10 y=1/(s1+3)-1/(s1-2);
11 figure
12 plzr(y)
13 z=1/(s1-2)-1/(s1+3);
14 figure
15 plzr(z)
16 disp("there is no region of convergence for c hence
      no transform exists")
```

---

### Scilab code Exa 3.6 ROC and pole zero plot

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

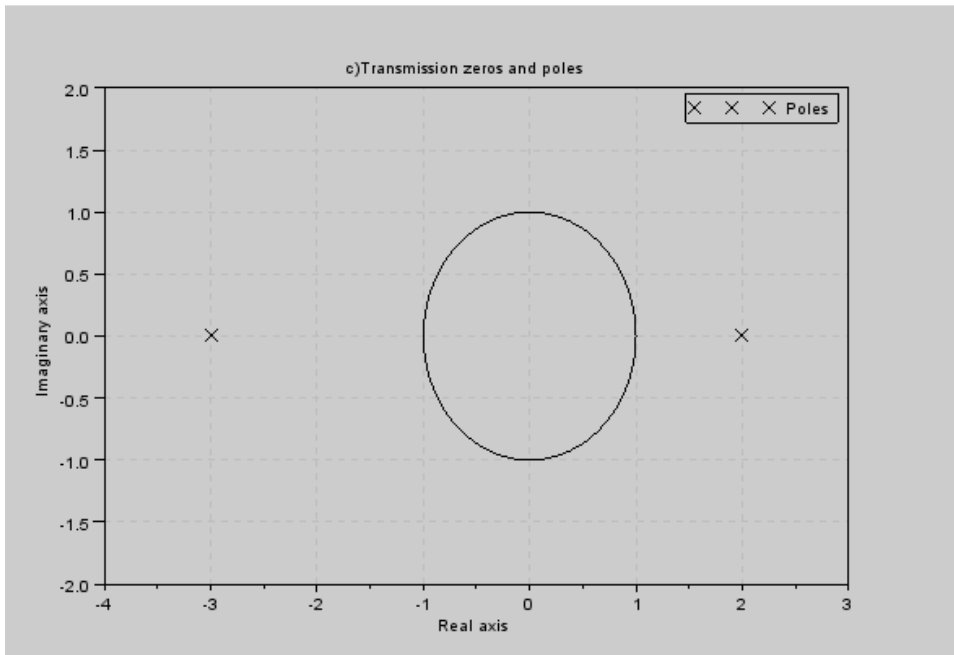
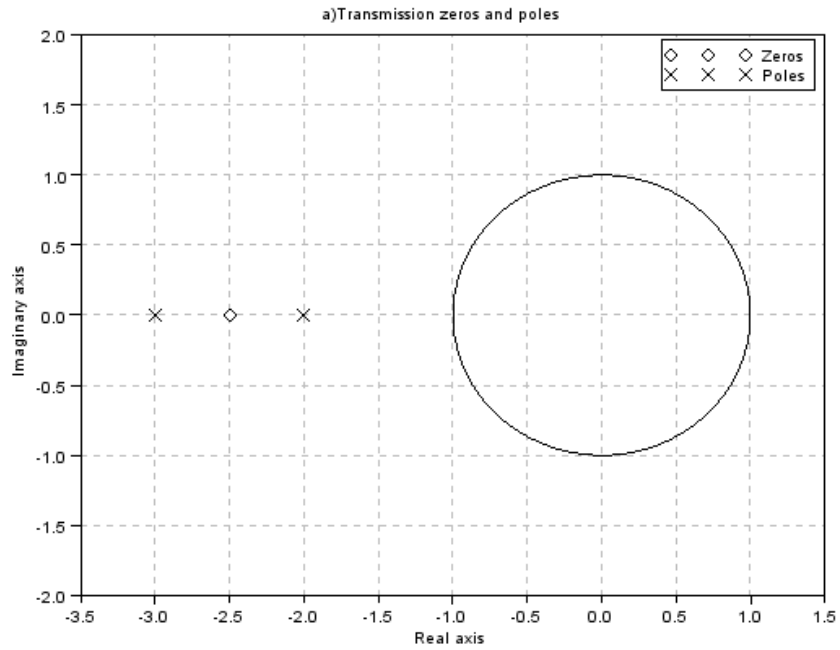


Figure 3.1: pole zero plot

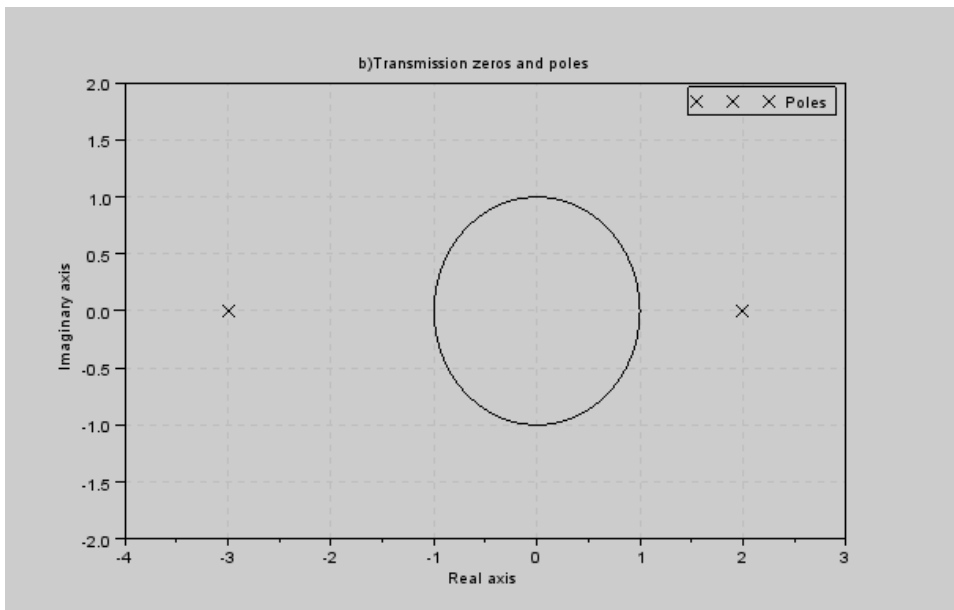


Figure 3.2: pole zero plot

```

7 t=-5:0.1:5;
8 x=%e^(-a*abs(t));
9 subplot(2,1,1)
10 plot(t,x)
11 subplot(2,1,2)
12 x=1/(s1+a)-1/(s1-a);
13 plzr(x)
14 //a<0
15 a=-0.5;
16 t=-5:0.1:5;
17 x=%e^(-a*abs(t));
18 figure
19 subplot(2,1,1)
20 plot(t,x)
21 subplot(2,1,2)
22 x=1/(s1+a)-1/(s1-a);
23 plzr(x)
24 disp("there is no region of convergence when a<0")

```

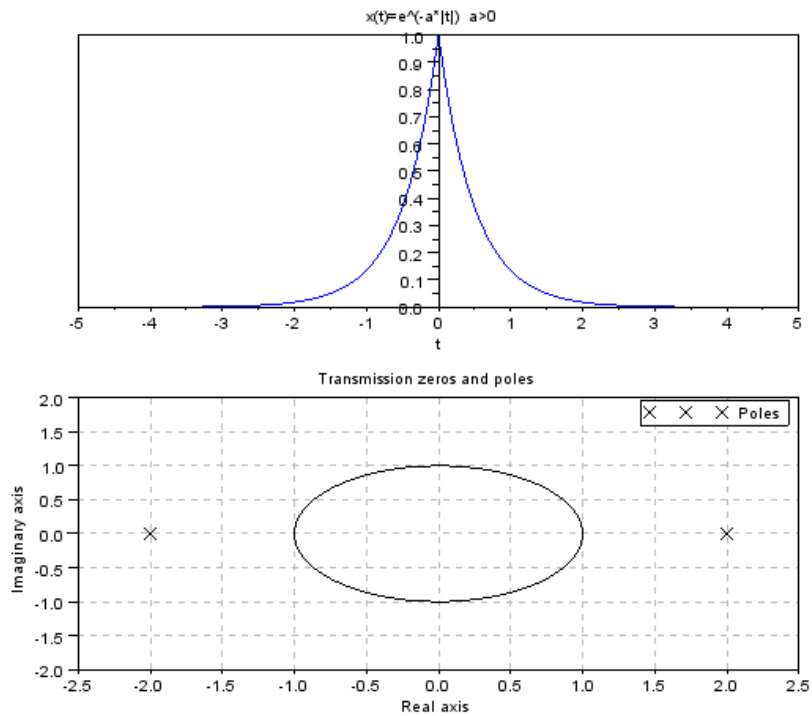


Figure 3.3: ROC and pole zero plot

hence no transform exists for  $a < 0$ )

---

### Scilab code Exa 3.13 derivative and shifting property

```

1 syms t s a w
2 // given  $u(t) \leftrightarrow 1/s$ 
3 //  $\delta(t) = \text{diff}(u(t))$ 
4 u=laplace(1);
5 d=s*u

```

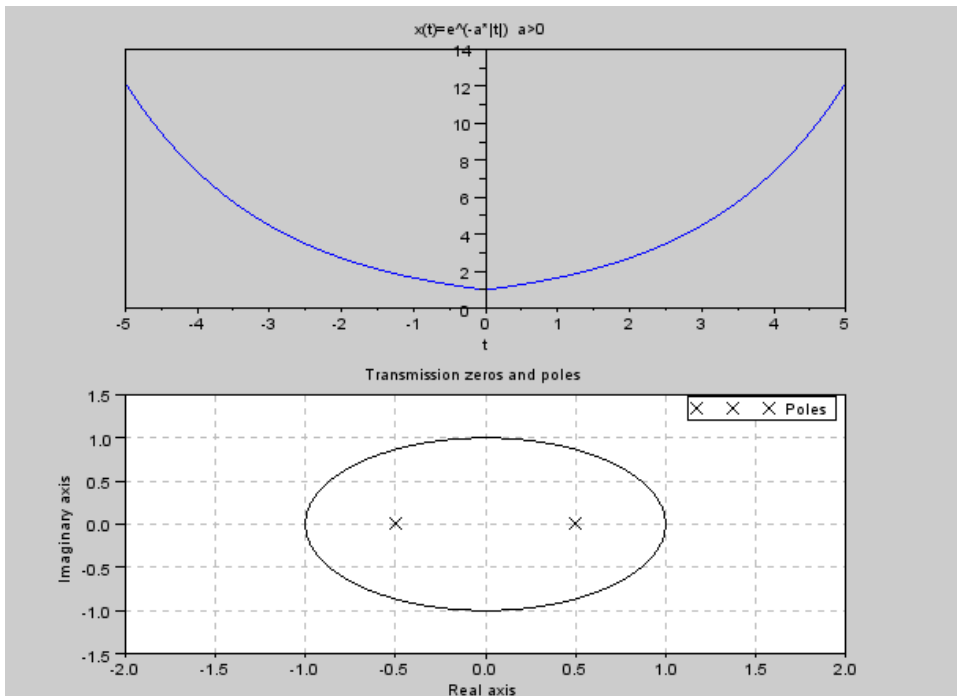


Figure 3.4: ROC and pole zero plot

```

6 disp(d," delta (t) <-->")
7 d1=s*d
8 disp(d1," diff(delta(t)) <-->")
9 tu=-diff(u,s)
10 disp(tu," t*u(t) <-->")
11 eu=laplace(%e^-(a*t))
12 disp(eu," e^-a*t*u(t) <-->")
13 teu=-diff(eu,s)
14 disp(teu," t*e^-a*t*u(t) <-->")
15 cu=laplace(cos(w*t))
16 disp(cu," cos(w0*t)*u(t) <-->")
17 ecu=laplace(%e^(-a*t)*cos(w*t))
18 disp(tu," e^(-a*t)*cos(w0*t)*u(t) <-->")

```

---

**Scilab code Exa 3.16** inverse laplace transform

```

1 clear all;
2 syms t s
3 x=1/(s+1)
4 f1=ilaplace(x)
5 disp(f1*'u(t)',,"a) x(t)=")
6 y=-1/(s+1)
7 f2=ilaplace(y)
8 disp(f2*'u(-t)',,"b) x(t)=")
9 z=s/(s^2+4)
10 f3=ilaplace(z)
11 disp(f3*'u(t)',,"c) x(t)=")
12 zz=(s+1)/((s+1)^2+4)
13 f4=ilaplace(zz)
14 disp(f4*'u(t)',,"d) x(t)=")

```

---

**Scilab code Exa 3.17** inverse laplace transform

```

1 clear all;
2 syms t s
3 x=(2*s+4)/(s^2+4*s+3)
4 f1=ilaplace(x)
5 disp(f1*'u(t)', 'a) x(t)=')
6 y=-(x)
7 f2=ilaplace(y)
8 disp(f2*'u(-t)', 'b) x(t)=')
9 q= %s
10 z=pfss((2*q+4)/(q^2+4*q+3))
11 f3=ilaplace(-z(1))
12 f4=ilaplace(z(2))
13 ff=f3+f4
14 disp(f3*'u(-t)'+f4*'u(t)', 'c) x(t)=')

```

---

### Scilab code Exa 3.18 inverse laplace transform

```

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

```

---

### Scilab code Exa 3.19 inverse laplace transform

```

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

```

---

**Scilab code Exa 3.20** inverse laplace transform by partial fractions

```
1 syms t s
2 s= %s
3 a1=pfss((2*s+1)/(s+2))
4 f1=ilaplace(a1(1))
5 fx=f1
6 disp(fx*'u(t)'+ '2*delta(t)', "a) x(t)=")
7 a2=pfss((s^2+6*s+7)/(s^2+3*s+2))
8 f1=ilaplace(a2(1))
9 f2=ilaplace(a2(2))
10 fy=f1+f2
11 disp(fy*'u(t)'+ 'delta(t)', "b) x(t)=")
12 a3=pfss((s^3+2*s^2+6)/(s^2+3*s))
13 f1=ilaplace(a3(1))
14 f2=ilaplace(a3(2))
15 fz=f1+f2
16 disp(fz*'u(t)'+ '-delta(t)+delta1(t)', "c) x(t)=")
```

---

**Scilab code Exa 3.21** inverse laplace transform of time shifted signal

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

---

**Scilab code Exa 3.22** differentiation in s domain

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



---

**Scilab code Exa 3.24** output response

```
1 syms t s a
2 H=laplace(%e^(-a*t))
3 X=laplace(-%e^(a*t))
4 Y=X*H
5 y=ilaplace(Y)
6 disp(y,"y(t)=")
```

---

**Scilab code Exa 3.25** impulse response

```
1 syms t s
2 X=laplace(1)
3 Y=laplace(2*%e^(-3*t))
4 H=Y/X;
5 disp(H,"H(s)=")
6 s=%s;
7 h=2*s/(s+3);
8 hp=pfss(h);
9 h1=ilaplace(hp(1));
10 disp(h1*'u(t)'+ '2*delta(t)', "h(t)=")
11 //part b)
12 X=laplace(%e^-t)
13 Y=X*H;
14 y=ilaplace(Y);
15 disp(y*'u(t)', "y(t)=")
```

---

**Scilab code Exa 3.27** cascaded system transfer function

```

1 clear all;
2 clc;
3 syms t s
4 H1=laplace(%e^(-2*t))
5 H2=laplace(2*%e^(-t))
6 H=H1*H2;
7 h=ilaplace(H)
8 disp(h* 'u(t) ', 'h(t)=')

```

---

**Scilab code Exa 3.28** first order differential equation

```

1 clear all;
2 clc;
3 syms t s a
4 H=1/(s+a);
5 h=ilaplace(H)
6 disp(h* 'u(t) ', 'h(t)=')

```

---

**Scilab code Exa 3.29** impulse response

```

1 clear all;
2 clc;
3 syms t s1 a
4 s=%s
5 H=(s+1)/(s+2);
6 hp=pfss(H)
7 h1=ilaplace(hp(1))
8 disp(h1* 'u(t) '+ 'delta(t) ', 'h(t)=')

```

---

**Scilab code Exa 3.30** causality and stability

```

1 clear all;
2 clc;
3 syms t s1 a
4 s=%s;
5 H=1/((s+2)*(s-1));
6 hp=pfss(H)
7 h1=ilaplace(hp(1))
8 h2=ilaplace(hp(2))
9 disp((h1+h2)*'u(t)', "when the system is causal h(t)=
    ")
10 disp(h1*'u(-t)'+h2*'u(-t)', "when the system is
    stable h(t)=")
11 disp((h1+h2)*'u(-t)', "when the system is neither
    stable nor causal h(t)=")

```

---

#### Scilab code Exa 3.34 bilateral laplace transform

```

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

```

---

#### Scilab code Exa 3.36 unilateral laplace transform

```

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

```

---

### Scilab code Exa 3.37 unilateral laplace transform method

```
1 clear all;
2 clc;
3 syms n s a b y0 K t;
4 X=laplace(K*%e^(-b*t));
5 Y=y0/(s+a)+X/(s+a);
6 y=ilaplace(Y)
7 disp(y,"y(t)=")
```

---

### Scilab code Exa 3.38 second order ODE

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

---

### Scilab code Exa 3.39 RC circuit

```
1 clear all;
2 clc;
3 syms n s t R C V v0;
4 I=(V-v0)/(s*(R+1/(C*s)));
5 i=ilaplace(I);
6 disp(i*'u(t)', "i(t)=")
7 Vc=(V/(R*C*s)+v0)/(s+1/(R*C));
8 vc=ilaplace(Vc)
9 disp(vc,"vc(t)=")
```

---

### Scilab code Exa 3.40 RC circuit response

```
1 clear all;
2 clc;
3 syms n s t R C V v0;
4 I=(V-v0)/(s*(R+1/(C*s)));
5 i=ilaplace(I);
6 disp(i*'u(t)', 'i(t)=')
7 Vc=(V/(R*C*s)+v0)/(s+1/(R*C));
8 vc=ilaplace(Vc)
9 disp(vc*'u(t)', 'vc(t)=')
```

---

### Scilab code Exa 3.41 RLC circuit

```
1 clear all;
2 clc;
3 syms s t n;
4 I=1/(s/2+2+20/s)
5 i=ilaplace(I)
6 disp(i, 'i(t)=')
```

---

### Scilab code Exa 3.42 circuit analysis

```
1 clear all;
2 clc;
3 s=%s;
4 I1=(s+1)/(s+1/4);
5 I1p=pfss(I1)
6 i1=ilaplace(I1p(1))
```

```
7 disp(i1*'u(t)'+ 'delta(t)', 'i1(t)=')
8 I2=(s-1/2)/(s+1/4);
9 I2p=pfss(I2)
10 i2=ilaplace(I2p(1))
11 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

```
1 clear all;
2 clc;
3 syms n z a;
4 x=a^n;
5 X=symsum(-x*z^-n,n,-%inf,-1)
6 disp(X,"ans=")
```

---

Scilab code Exa 4.1.b z transform

```
1 clear all;
2 clc;
3 syms n z a;
4 x=a^-n;
5 X=symsum(x*z^-n,n,-%inf,-1)
6 disp(X,"ans=")
```

---

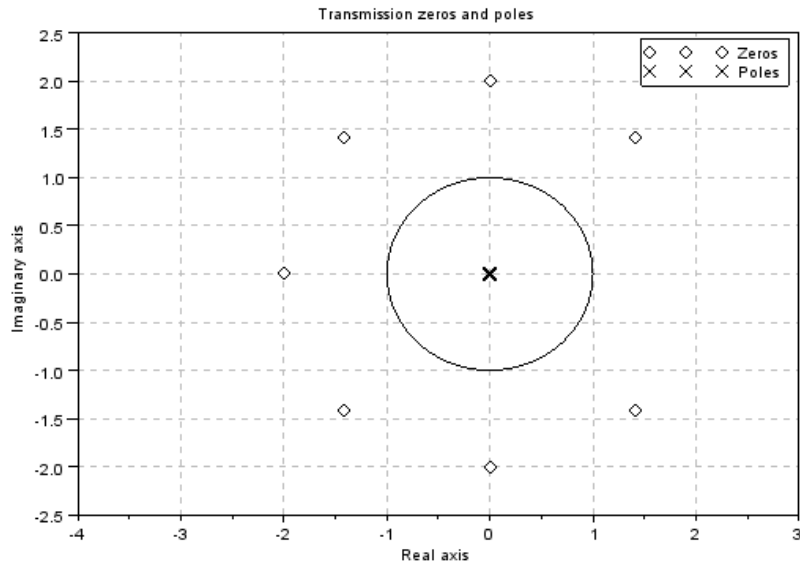


Figure 4.1: pole zero plot

Scilab code Exa 4.3 finite sequence z transform

```

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

```

---



#### Scilab code Exa 4.4 pole zero plot

```
1 clear all;
2 clc;
3 syms N n z a;
4 x=a^n;
5 X=symsum(x*z^-n,n,0,N-1)
6 //pole zero map for N=8,a=2,
7 z=%s;
8 X1=%s;
9 X1=0;
10 for i=0:7
11     X1=X1+(2*z^-1)^i
12 end
13 plzr(X1)
```

---

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

```
1 clear all;
2 clc;
3 syms n z;
4 x1=(1/2)^n;
5 X1=symsum(x1*(z^-n),n,0,%inf)
6 x2=3^-n;
7 X2=symsum(x2*(z^-n),n,0,%inf)
8 X=X1+X2
9 z=%s;
10 XX=%s;
11 XX=z/(z-.5)+z/(z-(1/3));
12 plzr(XX)
13 a=denom(XX)
14 b=roots(a)
15 i=1;
```

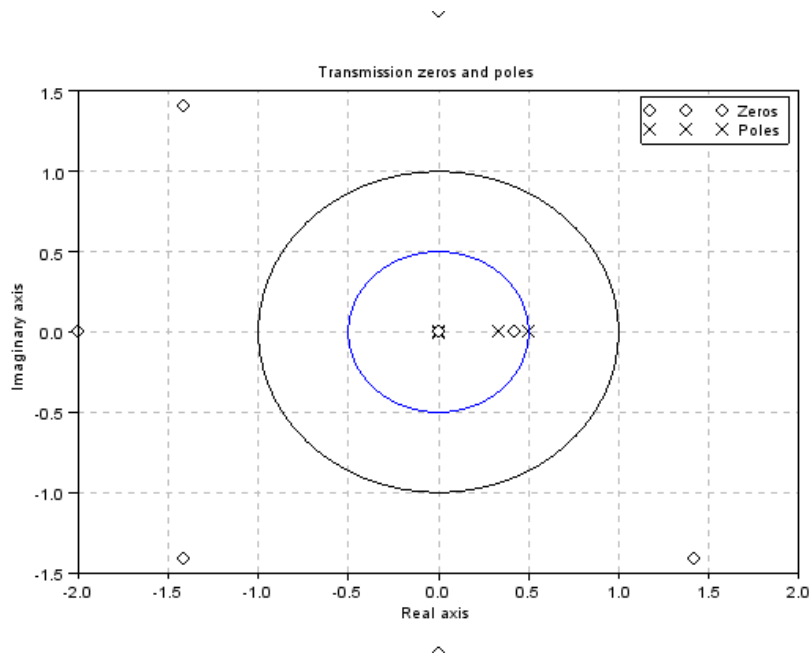


Figure 4.2: z transform and pole zero plot

```

16 for theta=0:1/50:360
17     rx(i)=.5*cos(theta);
18     ry(i)=.5*sin(theta);
19     i=i+1;
20 end
21 plot(rx,ry)
22 // the region outside blue circle indicates roc

```

---

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

```

1 clear all;
2 clc;
3 syms n z;

```

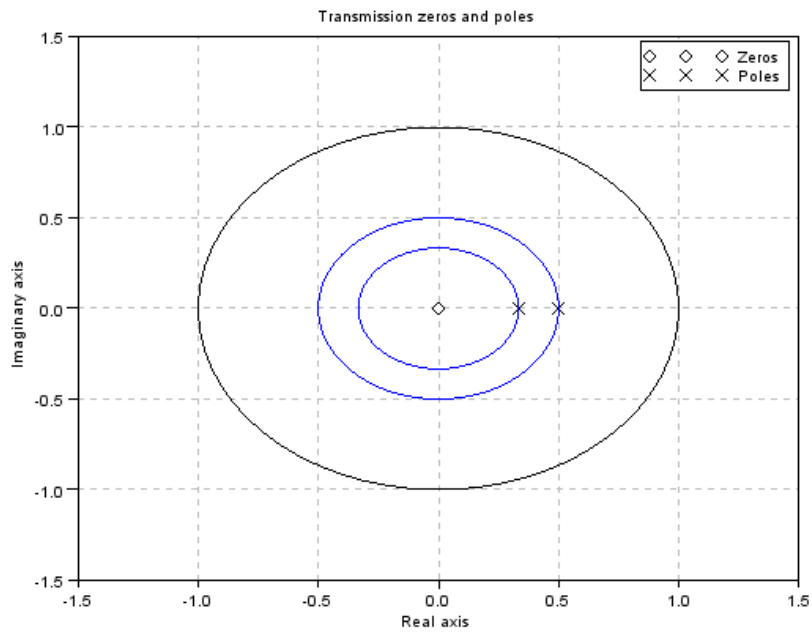


Figure 4.3: z transform and pole zero plot

```

4 x1=(1/2)^n;
5 X1=symsum(x1*(z^-n),n,-%inf,-1)
6 x2=3^-n;
7 X2=symsum(x2*(z^-n),n,0,%inf)
8 X=X1+X2
9 z=%s;
10 XX=%s;
11 XX=-z/(z-.5)+z/(z-(1/3));
12 plzr(XX)
13 a=denom(XX)
14 b=roots(a)
15 i=1;
16 for theta=0:1/50:360
17     rx(i)=b(1)*cos(theta);
18     ry(i)=b(1)*sin(theta);
19     i=i+1;
20 end
21 plot(rx,ry)
22 i=1;
23 for theta=0:1/50:360
24     rx(i)=b(2)*cos(theta);
25     ry(i)=b(2)*sin(theta);
26     i=i+1;
27 end
28 plot(rx,ry)
29 // the region between the blue circles indicates roc

```

---

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

```

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

```

```

7 X2=symsum(x2*(z^-n),n,0,%inf)
8 //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

```

1 clear ;
2 clear x;
3 clear n;
4 clc;
5 //0<a<1
6 a=.7 ;
7 n=-10:10;
8 for i=1:length(n)
9     if n(i)>0 then
10         x(i)=a^n(i);
11     else
12         x(i)=a^-n(i);
13     end
14 end
15 figure
16 a=gca();
17 a.x_location="origin";
18 xtitle('x[n] for a<1', 'n', 'x[n] ');
19 a.thickness = 2;
20 plot2d3(n,x)
21 plot(n,x, 'r. ')
22 //a>1
23 a=1.3;
24 for i=1:length(n)
25     if n(i)>0 then
26         x(i)=a^n(i);
27     else
28         x(i)=a^-n(i);
29     end

```

```

30 end
31 figure
32 a=gca();
33 a.x_location="origin";
34 xtitle('x[n] for a>1', 'n', 'x[n]');
35 a.thickness = 2;
36 plot2d3(n,x)
37 plot(n,x, 'r. ')
38 //|z|>a then X(z)=z/(z-a) if |z|<1/a then X(z)=-z/(
    z-1/a)
39 z=%s;
40 a=.5;
41 xx=z/(z-a)-z/(z-(1/a));
42 figure
43 plzr(xx);
44 d=denom(xx);
45 r=roots(d);
46 i=1;
47 for theta=0:1/50:360
48     rx(i)=r(1)*cos(theta);
49     ry(i)=r(1)*sin(theta);
50     i=i+1;
51 end
52 plot(rx,ry)
53 i=1;
54 for theta=0:1/50:360
55     rx(i)=r(2)*cos(theta);
56     ry(i)=r(2)*sin(theta);
57     i=i+1;
58 end
59 plot(rx,ry)
60 //the region between the blue lines is the ROC

```

---

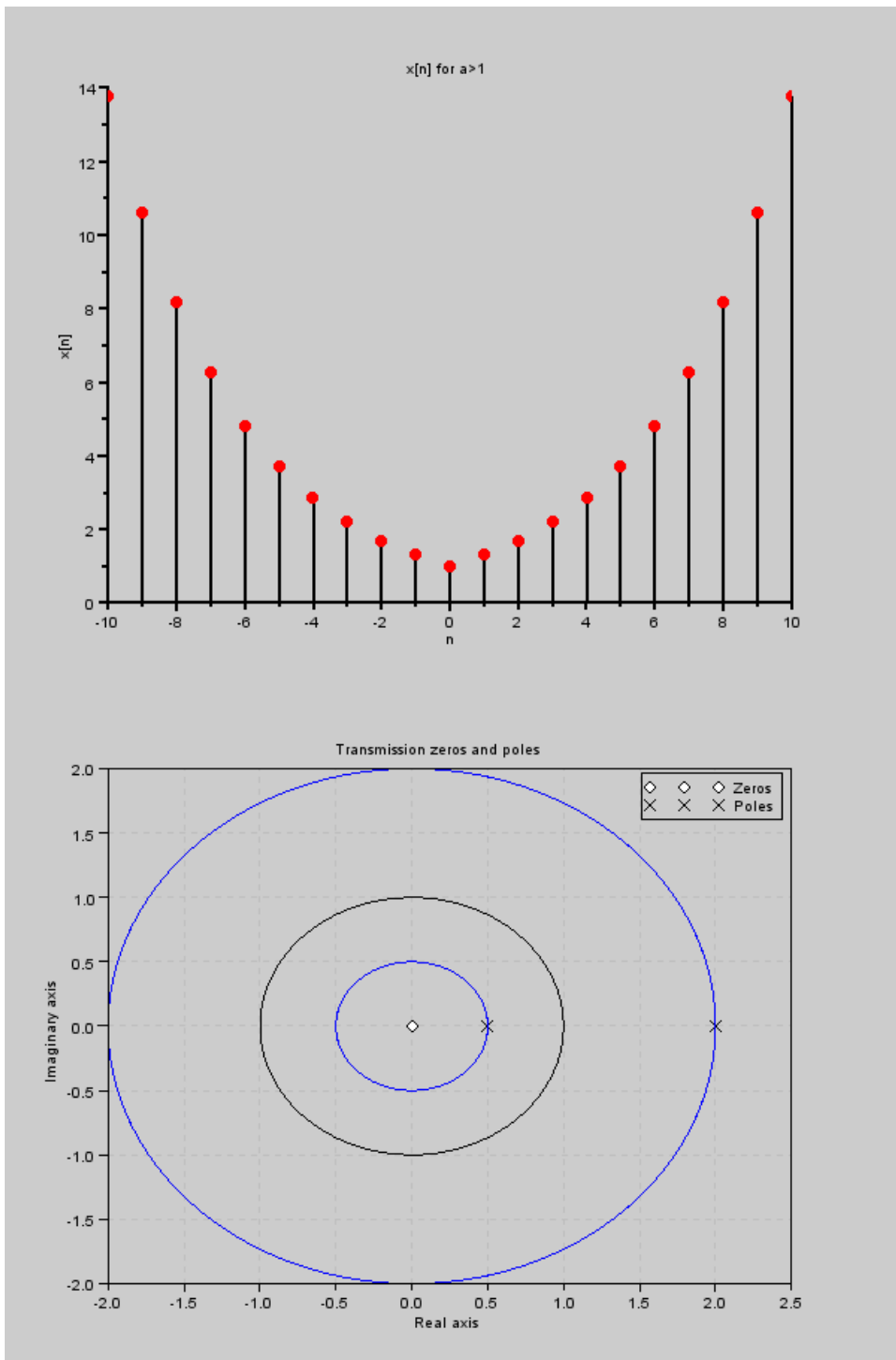


Figure 4.4: pole zero plot

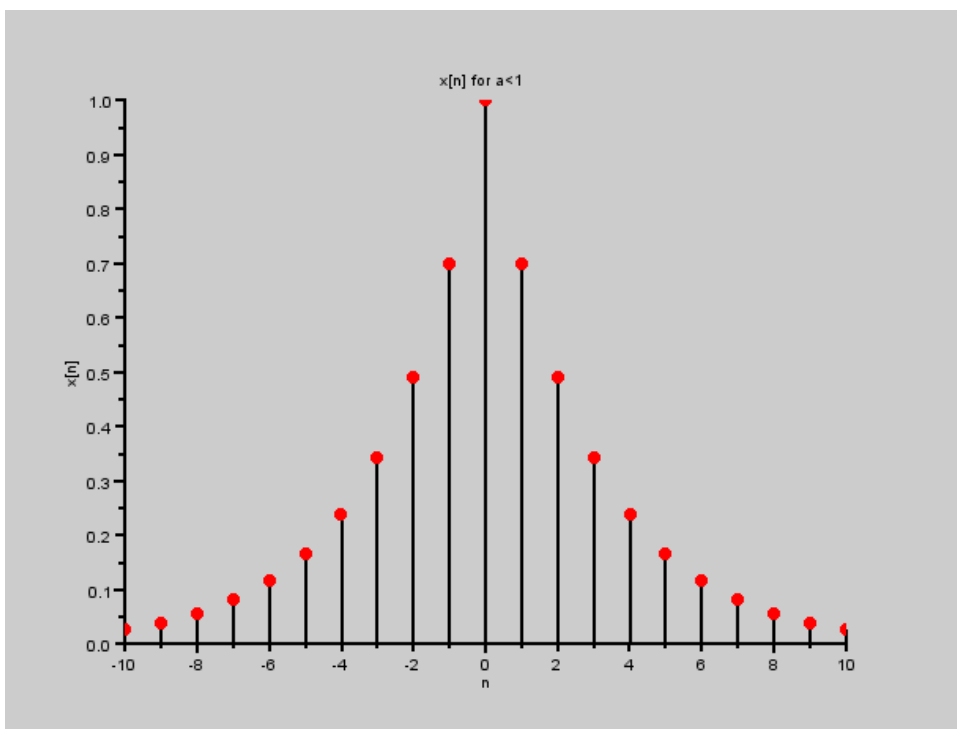


Figure 4.5: pole zero plot



#### Scilab code Exa 4.10 z transform

```
1  clc;
2  syms n z n0 a;
3  X=symsum(-1*z^-n,n,n0,%inf)
4  disp(X,"u[n-n0] <-->")
5  X2=symsum(z^-n,n,n0,n0);
6  disp(X2,"delta[n-n0] <-->")
7  X=symsum((a)^(n+1)*z^-n,n,-1,%inf)
8  disp(X,"a^(n+1)*u[n+1] <-->")
9  X=symsum(1*z^-n,n,-%inf,0)
10 disp(X,"u[-n] <-->")
11 X=symsum(a^-n*z^-n,n,-%inf,0)
12 disp(X,"a^-n*u[-n] <-->")
```

---

#### Scilab code Exa 4.12 differentiation property

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

---

#### Scilab code Exa 4.15 inverse z transform

```

1 clear all;
2 clc;
3 syms z;
4 X=[z^2 .5*z -5/2 z^-1];
5 n=-2:1;
6 a=size(X);
7 for i = 1:a(2)
8     x(i)=X(i)*(z^n(i));
9 end
10 disp(x, 'x[n]= ')

```

---

Scilab code Exa 4.16 inverse z transform

```

1 clear all;
2 clc;
3 z = %z;
4 a=5;
5 syms n z1;
6 X =1/(1-a*z^-1);
7 X1 = denom(X);
8 zp = roots(X1);
9 X1 = 1/(1-a*z1^-1);
10 F = X1*(z1^(n-1))*(z1-zp(1));
11 ha = limit(F,z1,zp(1));
12 disp(ha* 'u(n)', 'han]= ')
13 a=.5
14 X =1/(1-a*z^-1);
15 X1 = denom(X);
16 zp = roots(X1);
17 X1 = 1/(1-a*z1^-1);
18 F = X1*(z1^(n-1))*(z1-zp(1));
19 hb = limit(F,z1,zp(1));
20 disp(hb* '-u(-n-1)', 'hb[n]= ')

```

---

**Scilab code Exa 4.18.a** power series expansion technique

```
1 z=%z;
2 x=ldiv(z,2*z^2-3*z+1,5)
3 for i=5:-1:1
4     y(6-i)=x(i)*2^i;
5 end
6 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

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

---

**Scilab code Exa 4.19** inverse z transform

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

```

10 disp(x1, 'x1[n]=')
11 x2 = limit(F2, z1, zp(2));
12 disp(x2, 'x2[n]=')
13 x = x1+x2;
14 disp(x, 'x[n]=')
15 //a) when |z|<.5
16 n=-10:0;
17 disp(-x*'u(-n-1)', "when |z|<1/2 x[n]=")
18 xn=-(1-2^-n);
19 mprintf('x[n]={... ,%.2f,%.2f,%.2f,0}', xn($-3), xn($
    -2), xn($-1));
20 //b) when |z|>1
21 n=0:10;
22 disp(x*'u(n)', "when |z|>1 x[n]=")
23 xn=(1-2^-n);
24 mprintf('x[n]={%.2f,%.2f,%.2f...}', xn(1), xn(2), xn(3)
    );

```

---

#### Scilab code Exa 4.20 inverse z transform

```

1 z = %z;
2 syms n z1;
3 X = z/((z-1)*(z-2)^2);
4 X1 = denom(X);
5 zp = roots(X1);
6 X1 = z1/((z1-1)*(z1-2)^2);
7 F1 = X1/z1*(z1-zp(3))^2;
8 F2 = X1/z1*(z1-zp(1));
9 Y2 = limit(F1, z1, zp(3));
10 C1 = limit(F2, z1, zp(1));
11 F3=(X1/z1-(Y2*F1+C1*F2))*(z1-zp(3));
12 Y1 = limit(F3, z1, 0);
13
14 Xa=z1/(z1-zp(1));
15 F2 = Xa*z1^(n-1)*(z1-zp(1));

```

```

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

```

---

#### Scilab code Exa 4.21 inverse z transform

```

1 z = %z;
2 syms n z1 ;
3 X =1/((z-1)*(z-2));
4 //Xz=2z+1+X;
5 X1 = denom(X);
6 zp = roots(X1);
7 X1 = 1/((z1-1)*(z1-2));
8 F1 = X1*(z1^(n-1))*(z1-zp(1));
9 F2 = X1*(z1^(n-1))*(z1-zp(2));
10 x1 = limit(F1,z1,zp(1));
11 disp(x1, 'x1[n]= ');
12 x2 = limit(F2,z1,zp(2));
13 disp(x2, 'x2[n]= ');
14 x = x1+x2;
15 disp(x, 'xt[n]= ');
16 disp(x* '-u(-n-1) '+ '2*delta(n+1)+3/4*delta(n) ', "x[n]=
    ")

```

---

#### Scilab code Exa 4.22 inverse z transform

```

1 clc;

```

```

2 z = %z;
3 syms n z1 ;
4 X = 3/(z-2);
5 X1 = denom(X);
6 zp = roots(X1);
7 X1 = 3/(z1-2);
8 F1 = X1*(z1^(n-1))*(z1-zp(1));
9 x1 = limit(F1,z1,zp(1));
10 disp(x1, 'x1[n]= ')
11 disp(x1*'u(n-1)', 'x[n]=")

```

---

#### Scilab code Exa 4.23 inverse z transform

```

1 clc;
2 z = %z;
3 syms n z1 ;
4 X = z/((z+1)*(z+3));
5 X1 = denom(X);
6 zp = roots(X1);
7 X1 = z1/((z1+1)*(z1+3));
8 F1 = X1*(z1^(n-1))*(z1-zp(1));
9 F2 = X1*(z1^(n-1))*(z1-zp(2));
10 x1 = limit(F1,z1,zp(1));
11 disp(x1, 'x1[n]= ')
12 x2 = limit(F2,z1,zp(2));
13 disp(x2, 'x2[n]= ')
14 xt = x1+x2;
15 disp(xt*'u(n)', 'xt[n]= ')
16 //x[n]=2*xt[n-1]+xt[n-3]+3*xt[n-5];F1 = X1*(z1^(n-1)
    )*(z1-zp(1));
17 F1 = X1*(z1^(n-2))*(z1-zp(1));
18 F2 = X1*(z1^(n-2))*(z1-zp(2));
19 x1 = limit(F1,z1,zp(1));
20 disp(x1, 'x1[n]= ')
21 x2 = limit(F2,z1,zp(2));

```

```

22 disp(x2, 'x2[n]= ')
23 xt1 = x1+x2;
24 F1 = X1*(z1^(n-4))*(z1-zp(1));
25 F2 = X1*(z1^(n-4))*(z1-zp(2));
26 x1 = limit(F1,z1,zp(1));
27 disp(x1, 'x1[n]= ')
28 x2 = limit(F2,z1,zp(2));
29 disp(x2, 'x2[n]= ')
30 xt3 = x1+x2;
31 F1 = X1*(z1^(n-6))*(z1-zp(1));
32 F2 = X1*(z1^(n-6))*(z1-zp(2));
33 x1 = limit(F1,z1,zp(1));
34 disp(x1, 'x1[n]= ')
35 x2 = limit(F2,z1,zp(2));
36 disp(x2, 'x2[n]= ')
37 xt5 = x1+x2;
38 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

```

1  clc;
2  syms n z a;
3  X=symsum(-1*z^-n,n,0,%inf)
4  disp(X,"u[n] <-->")
5  h=a^n;
6  H=symsum(h*z^-n,n,0,%inf)
7  disp(H,"H(z)=")
8  Y=X*H;
9  disp(Y,"Y(z)=")
10 Y=z^2/((z-1)*(z-a));
11 F1=Y*z^(n-1)*(z-a);
12 F2=Y*z^(n-1)*(z-1);
13 x1=limit(F1,z,a);
14 x2=limit(F2,z,1);

```

```
15 x=x1+x2;
16 disp(x* 'u(n) ', 'x[n]=')
```

---

Scilab code Exa 4.26 output response

```
1 clc;
2 syms n z a b;
3 x=a^n;
4 X=symsum(x*z^-n,n,0,%inf)
5 disp(X, "a^n*u[n] <-->")
6 h=b^n;
7 H=symsum(h*z^-n,n,0,%inf)
8 disp(H, "H(z)=")
9 Y=X*H;
10 disp(Y, "Y(z)=")
11 Y=z^2/((z-a)*(z-b));
12 F1=Y*(z-a)*z^(n-1);
13 F2=Y*(z-b)*z^(n-1);
14 x1=limit(F1,z,a);
15 x2=limit(F2,z,b);
16 x=x1+x2;
17 disp(x* 'u(n) ', 'x[n]=')
```

---

Scilab code Exa 4.27 output response

```
1 clc;
2 x=[1 1 1 1];
3 h=[1 1 1];
4 Xz=0;
5 z=poly(0, "z");
6 for i=1:length(x)
7     Xz=Xz+x(i)*z^(1-i);
8 end
```



```

9 Hz=0;
10 for i=1:length(h)
11     Hz=Hz+h(i)*z^(1-i);
12 end
13 Yz=Xz*Hz;
14 y=coeff( numer(Yz) )
15 disp(y,"y[n]=")

```

---

**Scilab code Exa 4.28** impulse response

```

1 clc;
2 syms n z a;
3 X=symsum(1*z^-n,n,0,%inf)
4 disp(X,"u[n] <-->")
5 y=a^n;
6 Y=symsum(y*z^-n,n,0,%inf)
7 disp(Y,"a^n*u[n] <-->")
8 H=Y/X;
9 disp(H,"H(z)=");
10 H=(z-1)/(z-a);
11 F1=H*z^(n-1)*(z-a);
12 h=limit(F1,z,a);
13 disp(h*'u(n)'+ '1/a*delta(n)', "h[n]=")

```

---

**Scilab code Exa 4.29** impulse response and output

```

1 clc;
2 syms n z1 z;
3 z=%z;
4 X=symsum(1*z1^-n,n,0,%inf)
5 disp(X,"u[n] <-->")
6 y=2*3^-n;
7 Y=symsum(y*z1^-n,n,0,%inf)

```

```

8 disp(Y,"Y(z)=")
9 H=Y/X;
10 disp(H,"H(z)=")
11 Hz=2*(z-1)/(z-1/3);
12 Hd=denom(Hz);
13 zp=roots(Hd);
14 Hz=2*(z1-1)/(z1-1/3);
15 F1 = Hz*(z1^(n-1))*(z1-zp(1));
16 x1 = limit(F1,z1,zp(1));
17 disp(x1,'x1[n]=')
18 x =x1;
19 disp(x*'u(n)'+ '6*delta(n)', 'y[n]=');
20 disp("part b")
21 x=(1/2)^n;
22 X=symsum(x*z1^-n,n,0,%inf)
23 disp(X,"X(z)")
24 Yz=X*Hz;
25 disp(Yz,"Y(z)=")
26 Yz=2*z1*(z1-1)/((z1-1/2)*(z1-1/3));
27 F1 = Yz*(z1^(n-1))*(z1-1/2);
28 F2 = Yz*(z1^(n-1))*(z1-1/3);
29 y1 = limit(F1,z1,1/2);
30 disp(y1,'y1[n]=')
31 y2 = limit(F2,z1,1/3);
32 disp(y2,'y2[n]=')
33 y = y1+y2;
34 disp(y*'u(n)', 'y[n]=')

```

---

#### Scilab code Exa 4.31 impulse response

```

1 clear all;
2 clc;
3 syms n z1 a b;
4 h=a^n;
5 H=symsum(h*z1^-n,n,0,%inf)

```

```

6 disp(H,"H(z)");
7 disp("since  $|z| > |a|$   $z = \infty$  is included hence the
      system is causal")
8 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

```

1 z = %z;
2 syms n z1; //To find out Inverse z transform z must
      be linear z = z1
3 X = z^2/((z-(1/4))*(z-(1/2)))
4 X1 = denom(X);
5 zp = roots(X1);
6 X1 = z1^2/((z1-(1/4))*(z1-(1/2)))
7 F1 = X1*(z1^(n-1))*(z1-zp(1));
8 F2 = X1*(z1^(n-1))*(z1-zp(2));
9 h1 = limit(F1,z1,zp(1));
10 disp(h1,'h1[n]=')
11 h2 = limit(F2,z1,zp(2));
12 disp(h2,'h2[n]=')
13 h = h1+h2;
14 disp(h,'h[n]=')
15 //step response
16 Xz = z/(z-1); //u[n]
17 Y = Xz*X;
18 Y1 = denom(Y);
19 zp = roots(Y1);
20 Y1 = z1^3/((z1-1)*(z1-(1/4))*(z1-(1/2)))
21 F1 = Y1*(z1^(n-1))*(z1-zp(1));
22 F2 = Y1*(z1^(n-1))*(z1-zp(2));
23 F3 = Y1*(z1^(n-1))*(z1-zp(3));
24 y1 = limit(F1,z1,zp(1));
25 disp(y1,'y1[n]=')

```

```

26 y2 = limit(F2,z1,zp(2));
27 disp(y2,'y2[n]=')
28 y3 = limit(F3,z1,zp(3));
29 disp(y3,'y3[n]=')
30 y = y1+y2+y3;
31 disp(y*'u(n)', 'y[n]=')

```

---

#### Scilab code Exa 4.35 unilateral z transform

```

1 //unilateral z-transform
2 clear all;
3 clc;
4 syms n z a;
5 x=a^n;
6 X=symsum(x*z^-n,n,0,%inf)
7 disp(X,"ans=")
8 //x[n]=a^(n+1)*u[n+1]
9 x=a^(n+1);
10 X=symsum(x*z^-n,n,0,%inf)
11 disp(X,"ans=")

```

---

#### Scilab code Exa 4.37 unilateral z transform method

```

1 clc;
2 clear all;
3 syms n z a b y1 K;
4 Yz=a*y1*(z/(z-a))+K*z^2/((z-b)*(z-a));
5 //y1=y[-1]
6 Y1=z/(z-a);
7 Y2=z^2/((z-b)*(z-a));
8 FY1=Y1*z^(n-1)*(z-a);
9 y1n= limit(FY1,z,a);
10 FY21 = Y2*(z^(n-1))*(z-a);

```

```

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

```

---

### Scilab code Exa 4.38 difference equation

```

1 clear all;
2 clc;
3 syms z n ;
4 x=(1/3)^n;
5 X=symsum(x*z^-n,n,0,%inf)
6 disp(X,"X(z)=")
7 Xz=z/(z-1/3);
8 Yz=z/(2*(z-1/2))+z^2/((z-1/3)*(z-1/2));
9 //y[-1]=1
10 Y1=z/(z-1/2);
11 Y2=z^2/((z-1/3)*(z-1/2));
12 FY1=Y1*z^(n-1)*(z-1/2);
13 y1n= limit(FY1,z,1/2);
14 FY21 = Y2*(z^(n-1))*(z-1/2);
15 FY22 = Y2*(z^(n-1))*(z-1/3);
16 y21n = limit(FY21,z,1/2);
17 y22n = limit(FY22,z,1/3);
18 y=(1/2)*y1n+(y22n+y21n);
19 disp(y*'u(n+1)', 'part a) y[n]=')
20 //b)
21 x=(1/2)^n;
22 X=symsum(x*z^-n,n,0,%inf)
23 disp(X,"X(z)=")
24 Xz=z/(z-1/2);
25 Yz=z*(3*z^2-2*z+.5)/(3*(z-1)*(z-1/2)*(z-1/3));
26 //y[-1]=1 y[-2]=2
27 F1 = Yz*(z^(n-1))*(z-1);

```

```
28 F2 = Yz*(z^(n-1))*(z-1/2);
29 F3=  Yz*(z^(n-1))*(z-1/3);
30 y1 = limit(F1,z,1);
31 disp(y1,'y1[n]=')
32 y2 = limit(F2,z,1/2);
33 disp(y2,'y2[n]=')
34 y3 = limit(F3,z,1/3);
35 disp(y3,'y3[n]=')
36 y = y1+y2+y3;
37 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

```
1 clear all;
2 close;
3 clc;
4 T=1;
5 t=0:0.01:1;
6 w0=2*pi*T;
7 x=cos(w0*t);
8 for k=-5:5
9     cc(k+6,:)=exp(-%i*k*w0*t);
10    ck(k+6)=x*cc(k+6,:)/length(t);
11    if abs(ck(k+6))<0.01 then
12        ck(k+6)=0;
13    end
14 end
```

---

Scilab code Exa 5.4.b fourier series representation

```

1 clear all;
2 close;
3 clc;
4 T=1;
5 t=0:0.01:1;
6 w0=2*pi*T;
7 x=sin(w0*t);
8 for k=-5:5
9     cc(k+6,:)=exp(-%i*k*w0*t);
10    ck(k+6)=x*cc(k+6,:)/length(t);
11    if abs(ck(k+6))<0.01 then
12        ck(k+6)=0;
13    end
14 end

```

---

Scilab code Exa 5.4.c fourier series representation

```

1 clear ;
2 clear cc;
3 close;
4 clc;
5 t=0:0.01:20;
6 x=cos(2*t+%pi/4);
7 w0=2*pi*1;
8 for k=-5:5
9     cc(k+6,:)=exp(-%i*k*2*t);
10    ck(k+6)=x*cc(k+6,:)/length(t);
11    if abs(ck(k+6))<0.1 then
12        ck(k+6)=0;
13    end
14 end

```

---

Scilab code Exa 5.4.d fourier series representation



```

1 clear ;
2 close;
3 clc;
4 t=0:0.01:50;
5 x1=cos(4*t);
6 x2=sin(6*t);
7 w1=4;
8 w2=6;
9 w=int32([w1,w2]);
10 w0=gcd(w);
11 x=x1+x2;
12 for k=-5:5
13     cc(k+6,:)=exp(-%i*k.*t*2);
14     ck(k+6)=x*cc(k+6,:)/length(t);
15     if abs(ck(k+6))<0.1 then
16         ck(k+6)=0;
17     end
18 end

```

---

#### Scilab code Exa 5.4.e fourier series representation

```

1 clear ;
2 close;
3 clc;
4 t=0:0.01:50;
5 x=(sin(t))^2;
6 for k=-5:5
7     cc(k+6,:)=exp(-%i*k*2*t);
8     ck(k+6)=x*cc(k+6,:)/length(t);
9     if abs(ck(k+6))<0.1 then
10         ck(k+6)=0;
11     end
12 end

```

---

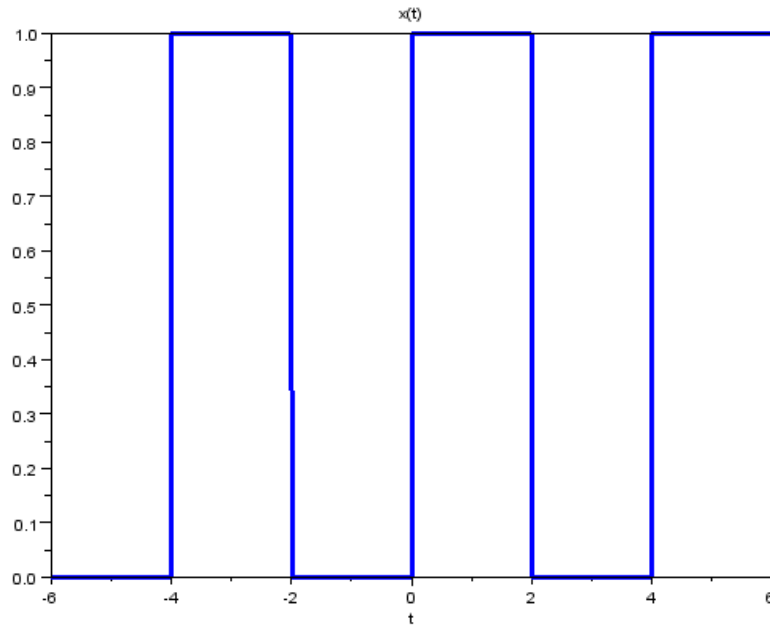


Figure 5.1: fourier series of a periodic square wave

**Scilab code Exa 5.5** fourier series of a periodic square wave

```

1 clear all;
2 close;
3 clc;
4 T0=2
5 t=-5.99:0.01:6;
6 t_temp=0.01:0.01:T0;
7 s=length(t)/length(t_temp);
8 x=[];

```

```

9 for i=1:s
10     if modulo(i,2)==1 then
11         x=[x zeros(1,length(t_temp))];
12     else
13         x=[x ones(1,length(t_temp))];
14     end
15 end
16 a=gca();
17 plot(t,x,'r')
18 poly1=a.children.children;
19 poly1.thickness=3;
20 poly1.foreground=2;
21 xtitle('x(t)', 't')
22 w0=%pi/2;
23 for k=-10:10
24     cc(k+11,:)=exp(-%i*k*w0*t);
25     ck(k+11)=x*cc(k+11,:)/length(t);
26     if abs(ck(k+11))<0.01 then
27         ck(k+11)=0;
28     else if real(ck(k+11))<0.1 then
29         ck(k+11)=%i*imag(ck(k+11));
30     end
31 end
32 if k==0 then
33     c0=ck(k+11);
34 end
35 end
36 //trigometric form
37 a0=2*c0;
38 a=2*real(ck);
39 b=2*imag(ck);

```

---

Scilab code Exa 5.6 fourier series of a periodic square wave

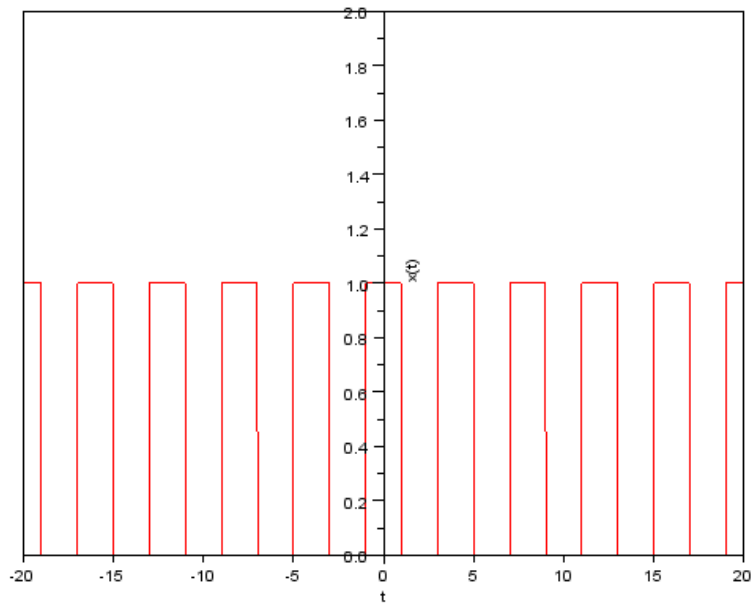


Figure 5.2: fourier series of a periodic square wave

```

1 clear ;
2 close;
3 clc;
4 T0=4;
5 t=-20.99:0.01:21;
6 t_temp=0.01:0.01:T0/2;
7 s=length(t)/length(t_temp);
8 x=[];
9 for i=1:s
10     if modulo(i,2)==0 then
11         x=[x zeros(1,length(t_temp))];
12     else
13         x=[x ones(1,length(t_temp))];
14     end
15 end
16 a=gca();
17 plot(t,x,'r')
18 poly1=a.children.children;
19 poly1.thickness=3;
20 poly1.foreground=2;
21 xtitle('x(t)', 't')
22 w0=%pi/2;
23 for k=-10:10
24     cc(k+11,:)=exp(-%i*k*w0*t);
25     ck(k+11)=x*cc(k+11,:)/length(t);
26     if abs(ck(k+11))<0.01 then
27         ck(k+11)=0;
28     else if imag(ck(k+11))<0.01 then
29         ck(k+11)=real(ck(k+11));
30     end
31     end
32     if k==0 then
33         c0=ck(k+11);
34     end
35 end
36 //trigometric form
37 a0=2*c0;
38 a=2*real(ck);

```

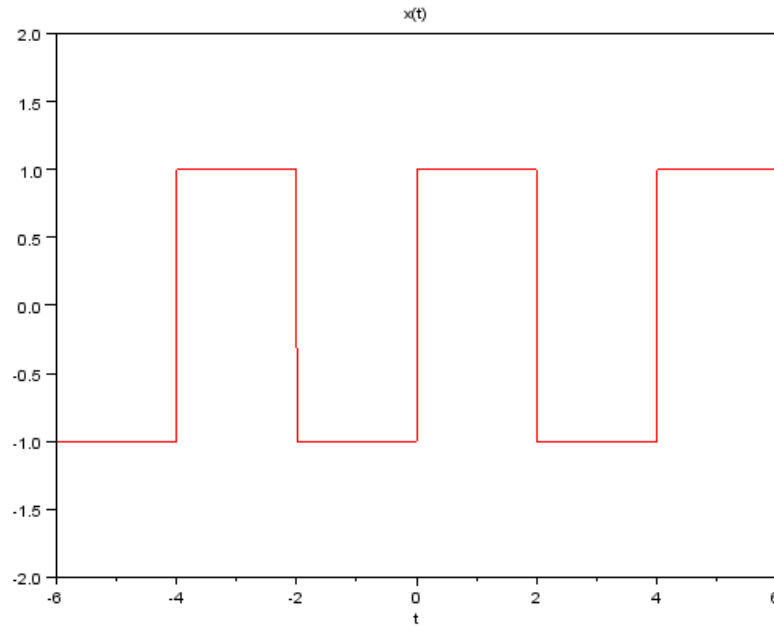


Figure 5.3: fourier series of a periodic square wave

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

---

**Scilab code Exa 5.7** fourier series of a periodic square wave

```

1 clear all;
2 close;
3 clc;
4 T0=4;
5 t=-5.99:0.01:6;
6 t_temp=0.01:0.01:T0/2;
7 s=length(t)/length(t_temp);

```

```

8 x=[];
9 for i=1:s
10     if modulo(i,2)==1 then
11         x=[x -ones(1,length(t_temp))];
12     else
13         x=[x ones(1,length(t_temp))];
14     end
15 end
16 plot(t,x,'r')
17 w0=%pi/2;
18 for k=-10:10
19     cc(k+11,:)=exp(-%i*k*w0*t);
20     ck(k+11)=x*cc(k+11,:)/length(t);
21     if abs(ck(k+11))<0.01 then
22         ck(k+11)=0;
23     else if real(ck(k+11))<0.1 then
24         ck(k+11)=%i*imag(ck(k+11));
25     end
26 end
27 if k==0 then
28     c0=ck(k+11);
29 end
30 end
31 //trigometric form
32 a0=2*c0;
33 a=2*real(ck);
34 b=2*imag(ck);

```

---

**Scilab code Exa 5.8** fourier series of a periodic impulse train

```

1 clear all;
2 close;
3 clc;

```

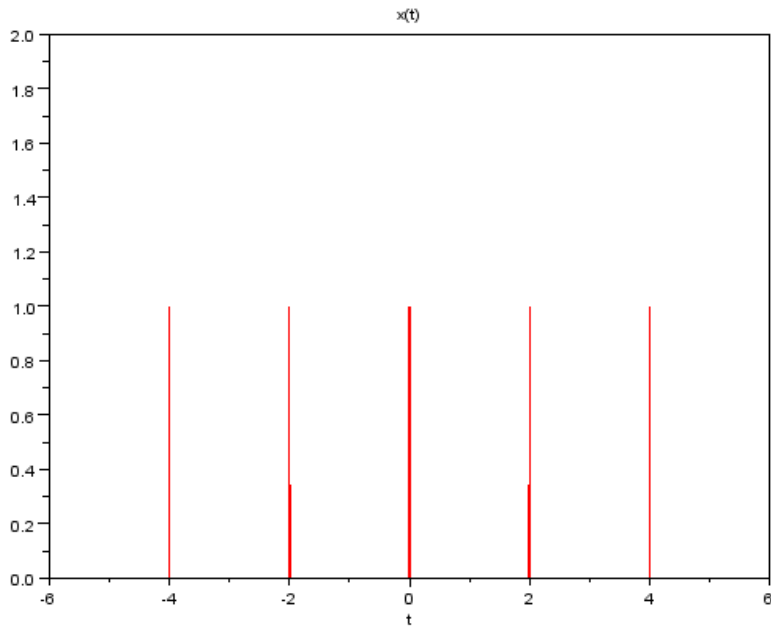


Figure 5.4: fourier series of a periodic impulse train



```

4 T0=2;
5 t=-6:0.01:6;
6 x=[];
7 for i=1:length(t)
8     if modulo(t(i),T0)==0 then
9         x(i)=1;
10    else
11        x(i)=0;
12    end
13 end
14 plot(t,x,'r')
15 //by sifting property
16 ck=1/T0;
17 //trigometric form
18 a0=2*ck;
19 a=2*real(ck);
20 b=2*imag(ck);

```

---

**Scilab code Exa 5.9** differentiation property of fourier series

```

1 clear all;
2 close;
3 clc;
4 T0=4;
5 t=-5.99:0.01:6;
6 t_temp=0.01:0.01:T0/2;
7 s=length(t)/length(t_temp);
8 dx=[];
9 x=[];
10 for i=1:s
11     if modulo(i,2)==1 then
12         dx=[dx -ones(1,length(t_temp))];
13         x=[x .5*t_temp($:-1:1)];
14     else
15         dx=[dx ones(1,length(t_temp))];

```

```

16         x=[x .5*t_temp];
17     end
18 end
19 figure
20 a=gca();
21 plot2d(t,x,8)
22 poly1=a.children.children;
23 poly1.thickness=3;
24 poly1.foreground=2;
25 xtitle('x(t)', 't')
26 figure
27 a=gca();
28 plot(t,dx, 'r')
29 poly1=a.children.children;
30 poly1.thickness=3;
31 poly1.foreground=2;
32 xtitle('x(t)', 't')
33 w0=%pi/2;
34 for k=-10:10
35     cc(k+11,:)=exp(-%i*k*w0*t);
36     cdk(k+11)=dx*cc(k+11,:)'/length(t);
37     if abs(cdk(k+11))<0.01 then
38         cdk(k+11)=0;
39     else if real(cdk(k+11))<0.1 then
40         cdk(k+11)=-%i*imag(cdk(k+11));
41     end
42     end
43     if k==0 then
44         ck(k+11)=.5;
45         c0=ck(k+11);
46     else
47         ck(k+11)=cdk(k+11)/(%i*k*w0);
48     end
49 end
50 //trigometric form
51 a0=2*c0;
52 a=2*real(ck);
53 b=2*imag(ck);

```

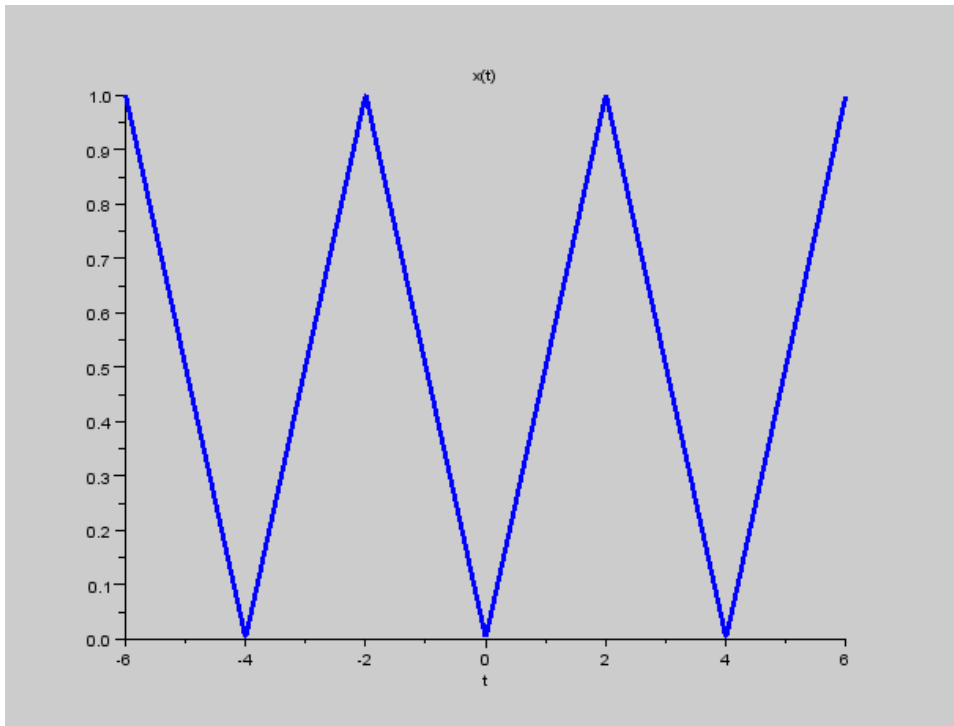


Figure 5.5: differentiation property of fourier series

---

Scilab code Exa 5.10 differentiation property of fourier series

```
1 clear all;  
2 close;  
3 clc;  
4 T0=4;  
5 t=-7.99:0.01:8;  
6 t_temp=0.01:0.01:T0;
```

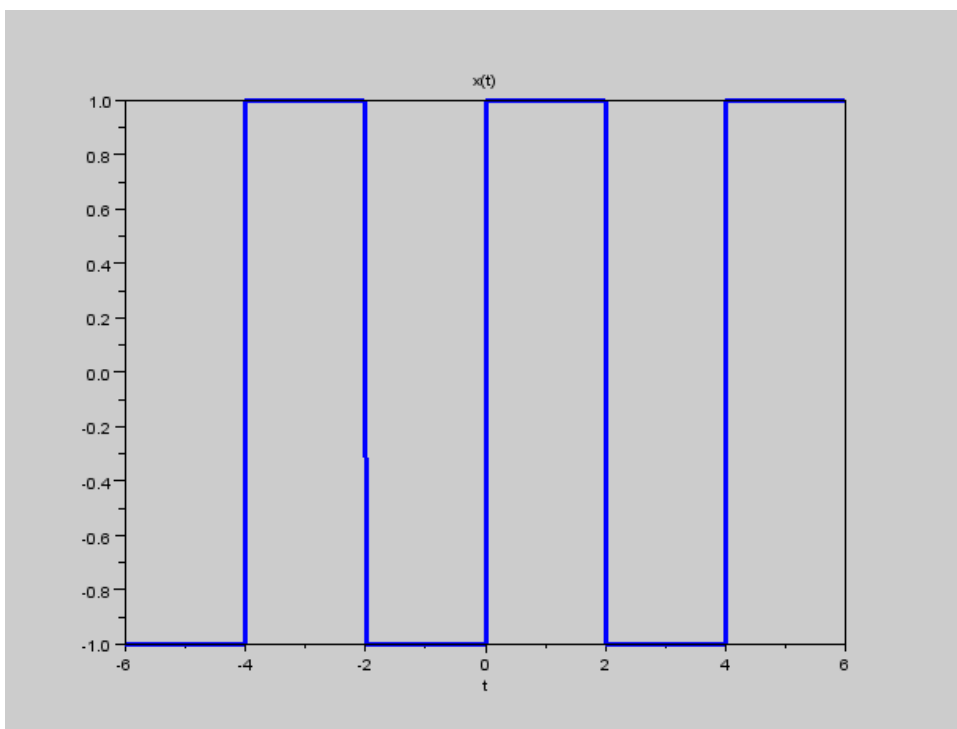


Figure 5.6: differentiation property of fourier series

```

7 s=length(t)/length(t_temp);
8 dx=[];
9 x=[];
10 for i=1:s
11     x=[x 0.5*t_temp($:-1:1)];
12 end
13 for i=1:length(t)
14     if modulo(t(i),T0)==0 then
15         dx(i)=1;
16     else
17         dx(i)=0;
18     end
19 end
20 dx1=-1/T0;
21 figure
22 a=gca();
23 plot2d(t,x,6)
24 poly1=a.children.children;
25 poly1.thickness=3;
26 poly1.foreground=2;
27 xtitle('x(t)', 't')
28 figure
29 plot(t,dx, 'r');
30 plot(t,dx1);
31 poly1=a.children.children;
32 poly1.thickness=3;
33 poly1.foreground=2;
34 xtitle('diff(x(t))', 't')
35 w0=%pi/2;
36 //by sifting property
37 cdk=1/T0;
38 for k=-10:10
39     if k~=0 then
40         ck(k+11)=cdk/(%i*k*w0);
41     else
42         ck(k+11)=.5;
43         c0=ck(k+11);
44     end

```

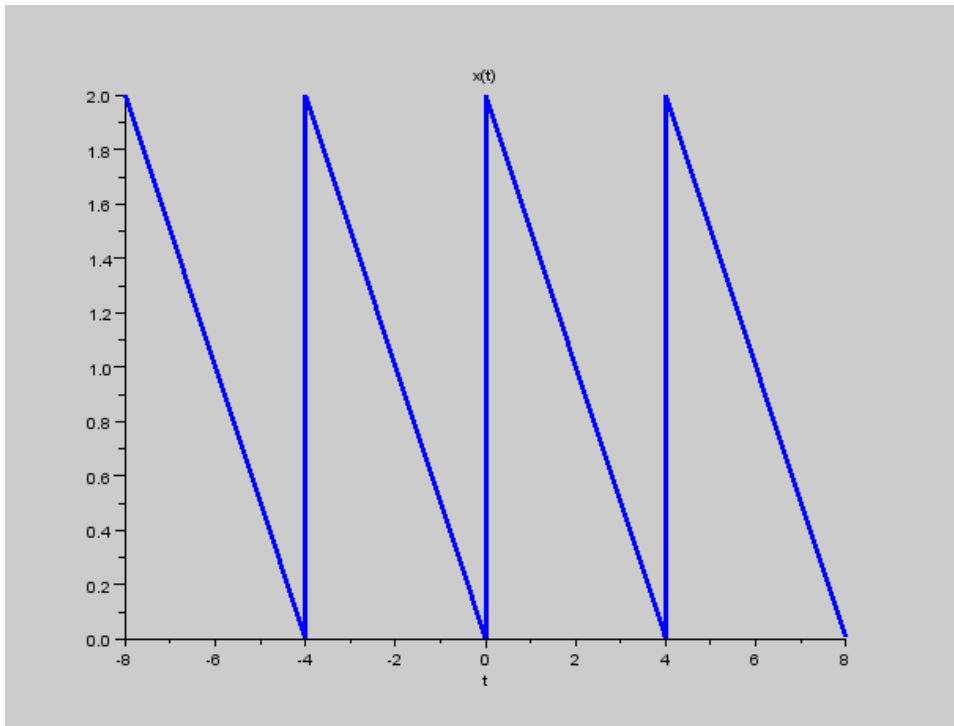


Figure 5.7: differentiation property of fourier series

```

45 end
46 a0=2*c0;
47 ak=2*real(ck);
48 bk=-2*imag(ck);

```

---

Scilab code Exa 5.11 magnitude spectra of a periodic square wave

```

1 clear ;
2 close;

```

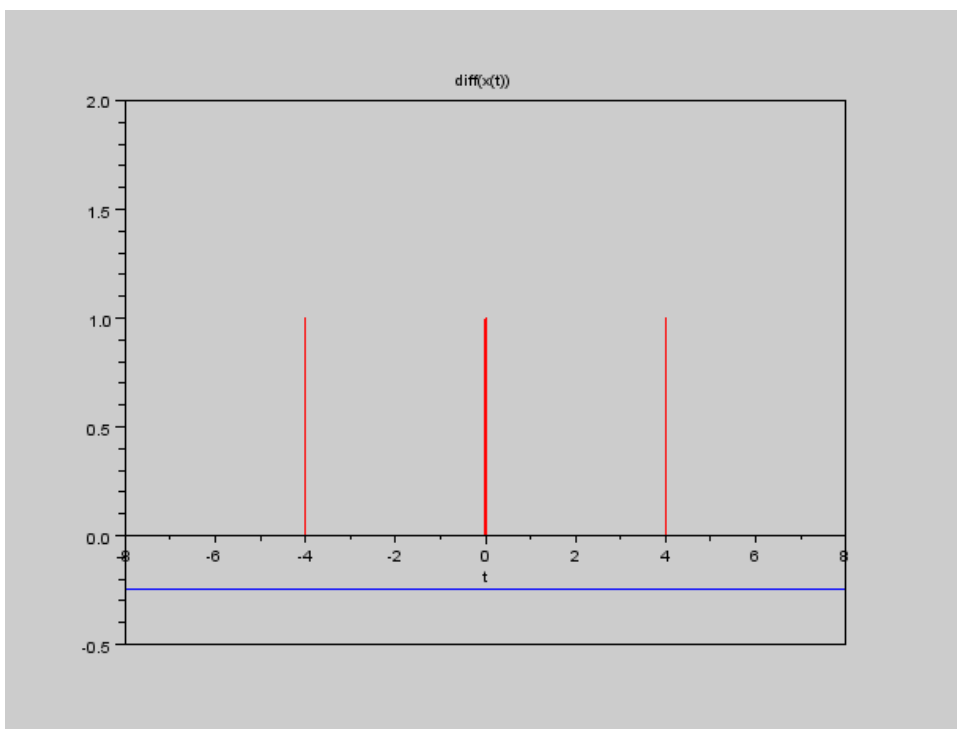


Figure 5.8: differentiation property of fourier series

```

3  clc;
4  T0=16;
5  t=-79.99:0.01:80;
6  t_temp=0.01:0.01:T0/4;
7  s=length(t)/length(t_temp);
8  x=[];
9  for i=1:s
10     if modulo(i,4)==1 then
11         x=[x ones(1,length(t_temp))];
12     else
13         x=[x zeros(1,length(t_temp))];
14     end
15 end
16 a=gca();
17 plot(t,x,'r')
18 poly1=a.children.children;
19 poly1.thickness=3;
20 poly1.foreground=2;
21 xtitle('x(t)', 't')
22 w0=%pi/8;
23 for k=-30:30;
24     cc(k+31,:)=exp(-%i*k*w0*t);
25     ck(k+31)=x*cc(k+31,:)/length(t);
26     if abs(ck(k+31))<0.01 then
27         ck(k+31)=0;
28         //else if real(ck(k+11))<0.1 then
29         //ck(k+11)=%i*imag(ck(k+11));
30         //end
31     end
32     if k==0 then
33         c0=ck(k+31);
34     end
35 end
36 k=-30:30;
37 figure
38 a=gca();
39 plot2d3(k,abs(ck));
40 poly1=a.children.children;

```



```

41 poly1.thickness=3;
42 poly1.foreground=2;
43 xtitle(' |ck| ', 'k ')
44 //part b
45 t_temp=0.01:0.01:T0/8;
46 s=length(t)/length(t_temp);
47 x=[];
48 for i=1:s
49     if modulo(i,8)==1 then
50         x=[x ones(1,length(t_temp))];
51     else
52         x=[x zeros(1,length(t_temp))];
53     end
54 end
55 figure
56 a=gca();
57 plot(t,x,'r')
58 poly1=a.children.children;
59 poly1.thickness=3;
60 poly1.foreground=2;
61 xtitle('x(t)', 't')
62 w0=%pi/8;
63 for k=-30:30;
64     cc(k+31,:)=exp(-%i*k*w0*t);
65     ck(k+31)=x*cc(k+31,:)/length(t);
66     if abs(ck(k+31))<0.01 then
67         ck(k+31)=0;
68         //else if real(ck(k+11))<0.1 then
69         //ck(k+11)=%i*imag(ck(k+11));
70         //end
71     end
72     if k==0 then
73         c0=ck(k+31);
74     end
75 end
76 k=-30:30;
77 figure
78 a=gca();

```

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

---

**Scilab code Exa 5.19** fourier transform of a rectangular pulse

```
1 clear;
2 clc;
3 close;
4 A=1;
5 t0=1;
6 dt=.1;
7 t=-10:dt:10;
8 for i=1:length(t)
9     if t(i)>-t0 & t(i)<t0 then
10         x(i)=A;
11     else
12         x(i)=0;
13     end
14 end
15 a=gca();
16 plot(t,x);
17 poly1=a.children.children;
18 poly1.thickness=3;
19 poly1.foreground=2;
20 xtitle('x(t)', 't')
21 // clf();
22 wmax=10;
23 w=-wmax:0.1:wmax;
```

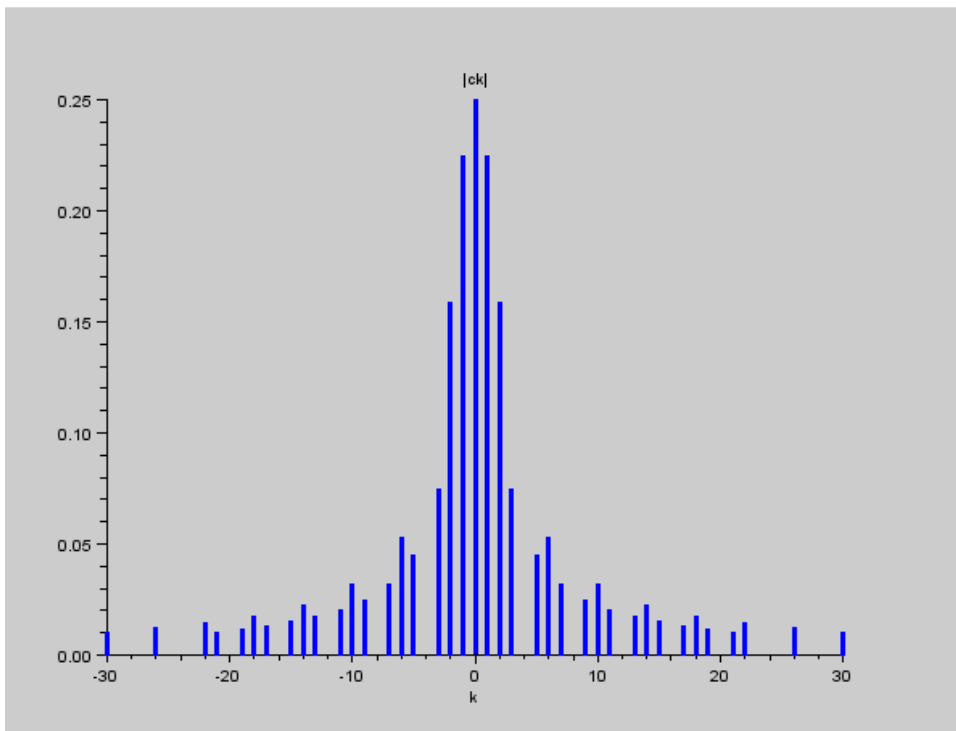
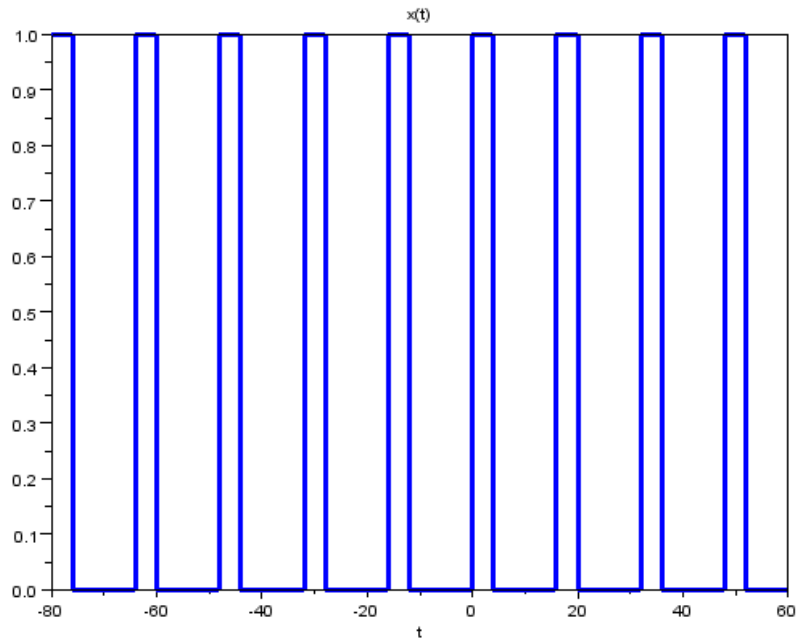


Figure 5.9: magnitude spectra of a periodic square wave

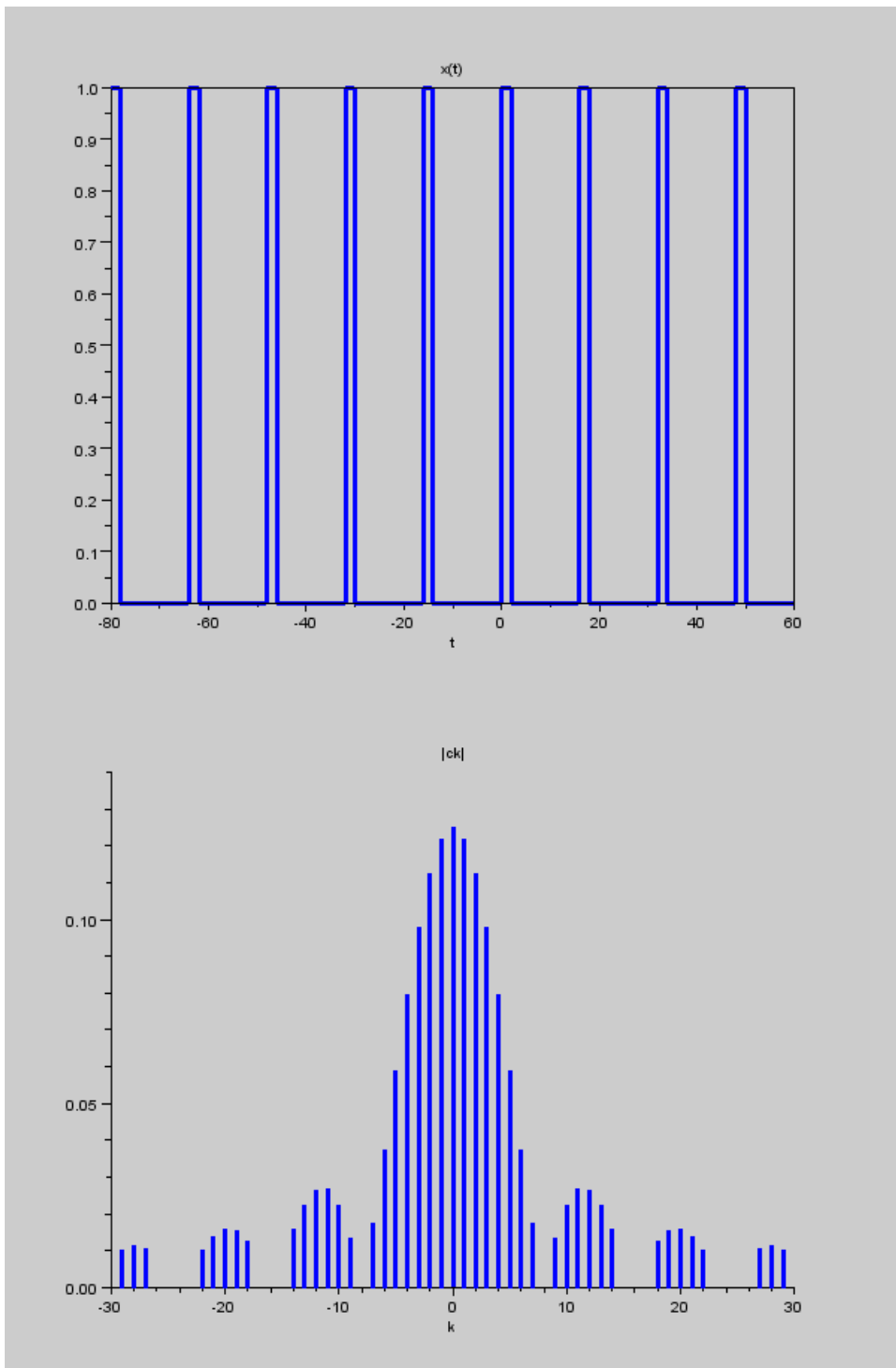


Figure 5.10: magnitude spectra of a periodic square wave

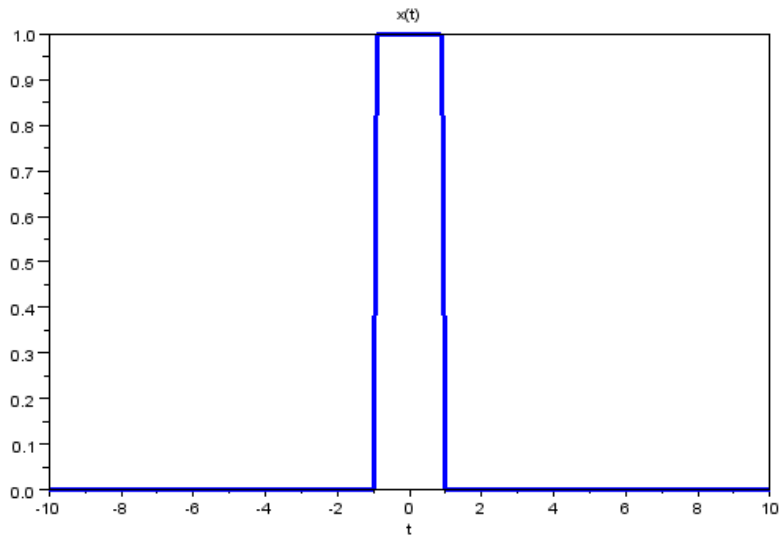


Figure 5.11: fourier transform of a rectangular pulse

```

24 Xw=x'*exp(-%i*(w'*t))*dt;
25 Xw_mag=(Xw);
26 Xw_mag=[Xw_mag($:-1:1) Xw_mag];
27 w=[w($:-1:1) w];
28 figure
29 a=gca();
30 plot2d(w,Xw_mag);
31 poly1=a.children.children;
32 poly1.thickness=3;
33 poly1.foreground=2;
34 xtitle(' |X(w)| ', 'w')

```

---

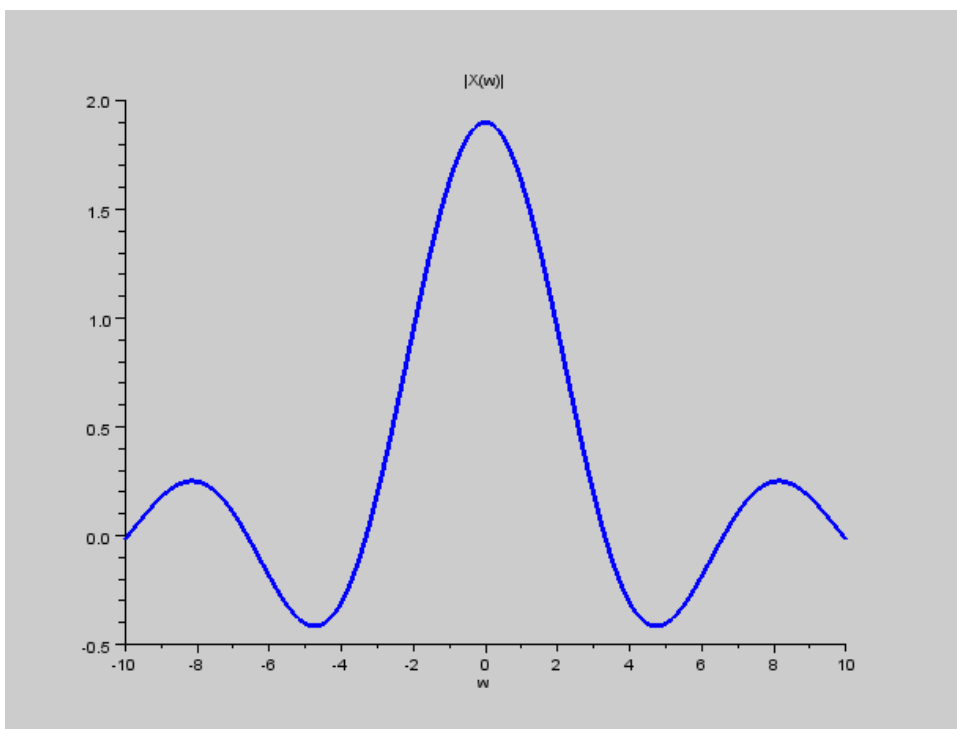


Figure 5.12: fourier transform of a rectangular pulse

### Scilab code Exa 5.20 fourier transform of a sinc function

```
1 clear;
2 clc;
3 close;
4 A=1;
5 t0=1;
6 dt=.1;
7 t=-10:dt:10;
8 for i=1:length(t)
9     if t(i)==0 then
10        x(i)=A/%pi;
11    else
12        x(i)=sin(A*t(i))./(%pi*t(i));
13    end
14 end
15 a=gca();
16 plot(t,x);
17 poly1=a.children.children;
18 poly1.thickness=3;
19 poly1.foreground=2;
20 xtitle('x(t)', 't')
21 // clf();
22 wmax=10;
23 w=-wmax:0.1:wmax;
24 Xw=x'*exp(-%i*(w'*t))*dt;
25 Xw_mag=(Xw);
26 //Xw_mag=[Xw_mag($:-1:1) Xw_mag];
27 figure
28 a=gca();
29 plot2d(w,Xw_mag);
30 poly1=a.children.children;
31 poly1.thickness=3;
32 poly1.foreground=2;
33 xtitle('|X(w)|', 'w')
```

---

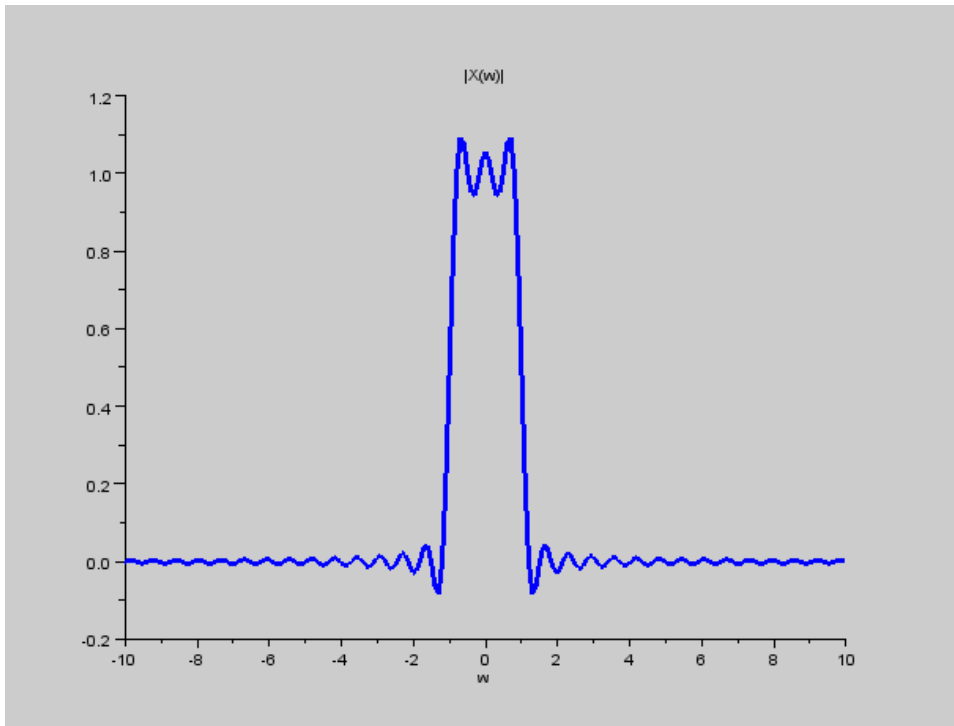


Figure 5.13: fourier transform of a sinc function

**Scilab code Exa 5.21** fourier transform

```
1 clear;  
2 clc;  
3 close;  
4 a=1;  
5 dt=.01;  
6 t=0:dt:10;
```



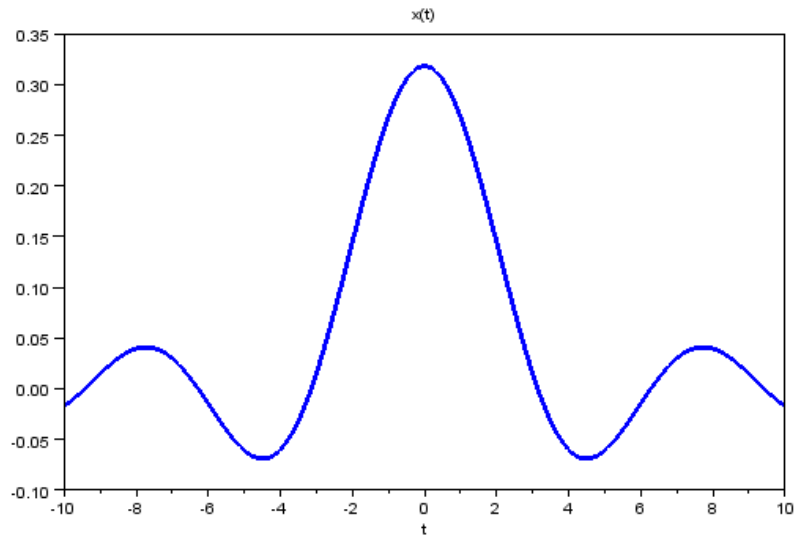


Figure 5.14: fourier transform of a sinc function

```

7 x=exp(-a*t);
8 y=[x($:-1:1) x];
9 t1=-10:dt:10.01;
10 a=gca();
11 plot(t1,y);
12 poly1=a.children.children;
13 poly1.thickness=3;
14 poly1.foreground=2;
15 xtitle('x(t)', 't')
16 //clf();
17 //stacksize(10000,100000)
18 wmax=10;
19 w=0:0.01:wmax;
20 Xw=x*exp(-%i*(w'*t))*dt;
21 Xw_mag=(Xw);
22 Xw_mag=[Xw_mag($:-1:1) Xw_mag];
23 w=[-w($:-1:1) w];
24 figure

```

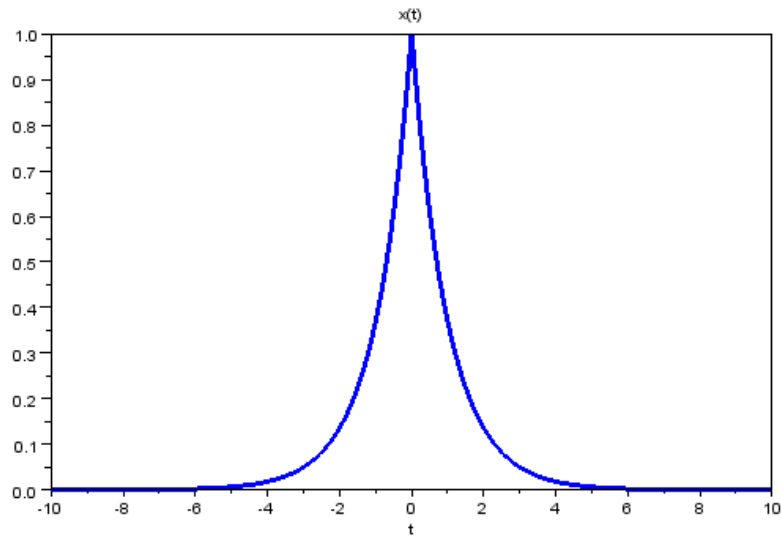


Figure 5.15: fourier transform

```
25 a=gca();
26 plot2d(w,Xw_mag);
27 poly1=a.children.children;
28 poly1.thickness=3;
29 poly1.foreground=2;
30 xtitle('X(w)', 'w')
```

---

Scilab code Exa 5.22 fourier transform

```
1 clear;
2 clc;
3 close;
```

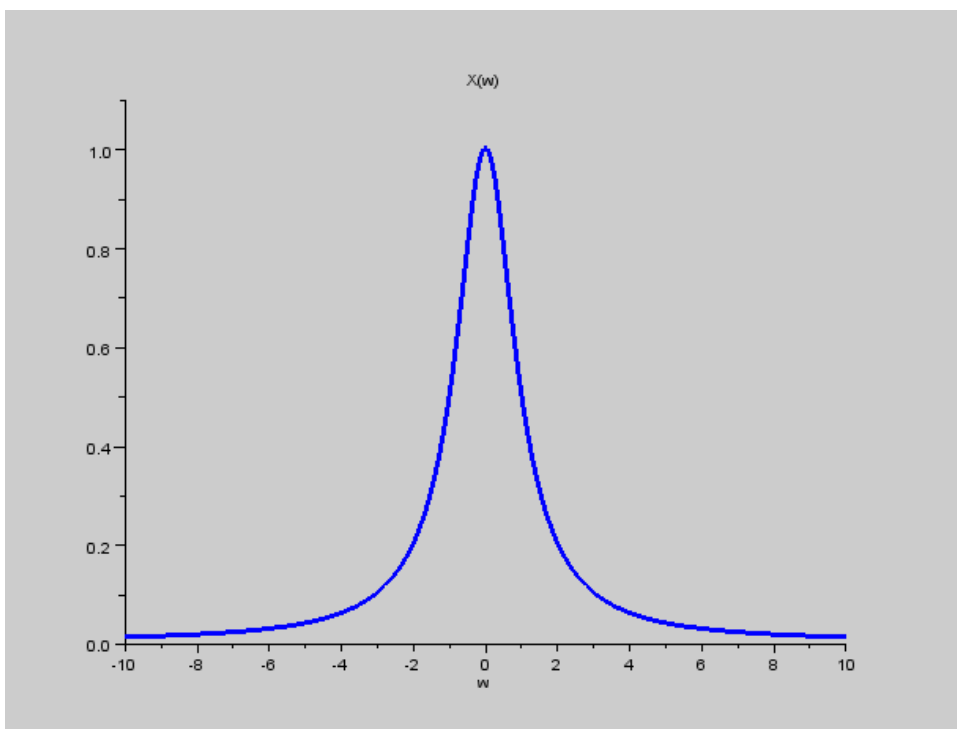


Figure 5.16: fourier transform

```

4 a=1.5;
5 dt=.1;
6 t=0:dt:10;
7 s=size(t);
8 x=ones(1,s(2))./(a^2+t^2);
9 y=[x($:-1:1) x];
10 t1=-10:dt:10.1;
11 a=gca();
12 plot(t1,y);
13 poly1=a.children.children;
14 poly1.thickness=3;
15 poly1.foreground=2;
16 xtitle('x(t)', 't')
17
18 // clf();
19 wmax=10;
20 w=0:0.1:wmax;
21 Xw=x*exp(-%i*(w'*t))*dt;
22 Xw_mag=(Xw);
23 Xw_mag=[Xw_mag($:-1:1) Xw_mag];
24 w=[-w($:-1:1) w];
25 figure
26 a=gca()
27 plot2d(w,Xw_mag);
28 poly1=a.children.children;
29 poly1.thickness=3;
30 poly1.foreground=2;
31 xtitle('X(w)', 'w')

```

---

Scilab code Exa 5.23.a fourier transform

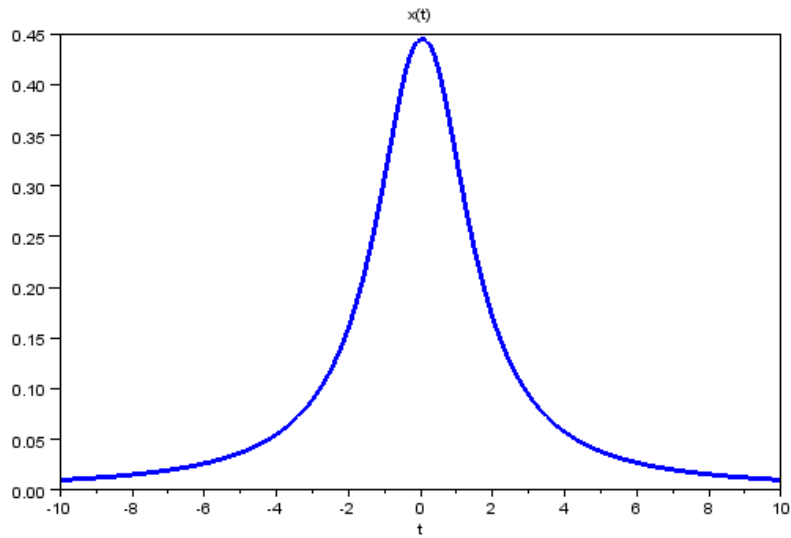


Figure 5.17: fourier transform

```

1 clear;
2 clc;
3 close;
4 dt=.1
5 t=-10:dt:10;
6 for i=1:length(t)
7     if t(i)==0 then
8         x(i)=1;
9     else
10        x(i)=0;
11    end
12 end
13 a=gca();
14 plot2d3(t,x);
15 poly1=a.children.children;
16 poly1.thickness=3;
17 poly1.foreground=2;
18 xtitle('x(t)', 't')

```

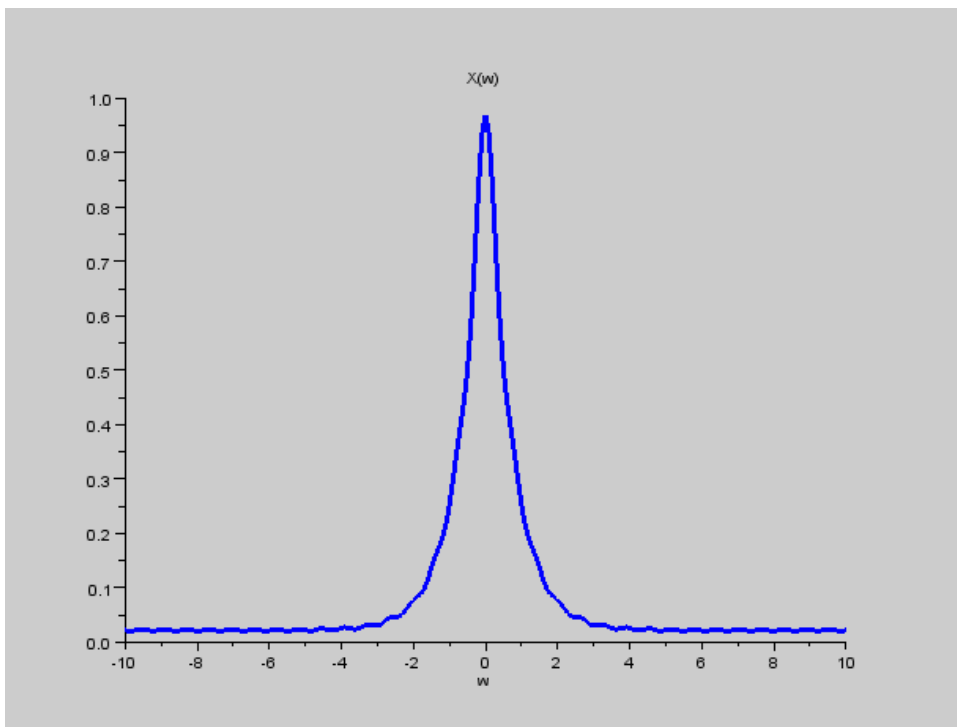


Figure 5.18: fourier transform

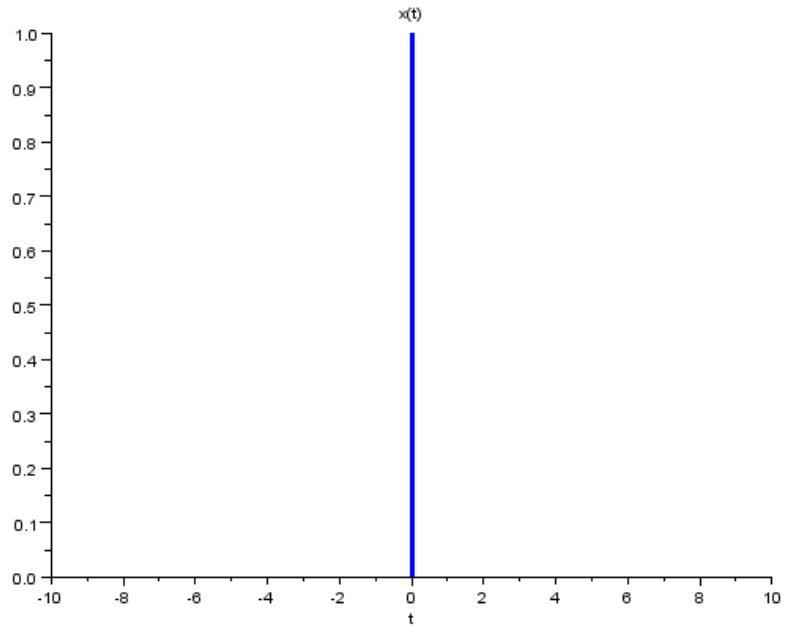


Figure 5.19: fourier transform

```

19 wmax=10;
20 w=-wmax:0.1:wmax;
21 Xw=x'*exp(-%i*(w'*t))*dt;
22 Xw_mag=(Xw);
23 figure
24 a=gca();
25 plot2d(w,Xw_mag);
26 poly1=a.children.children;
27 poly1.thickness=3;
28 poly1.foreground=2;
29 xtitle('X(w)', 'w')

```

---

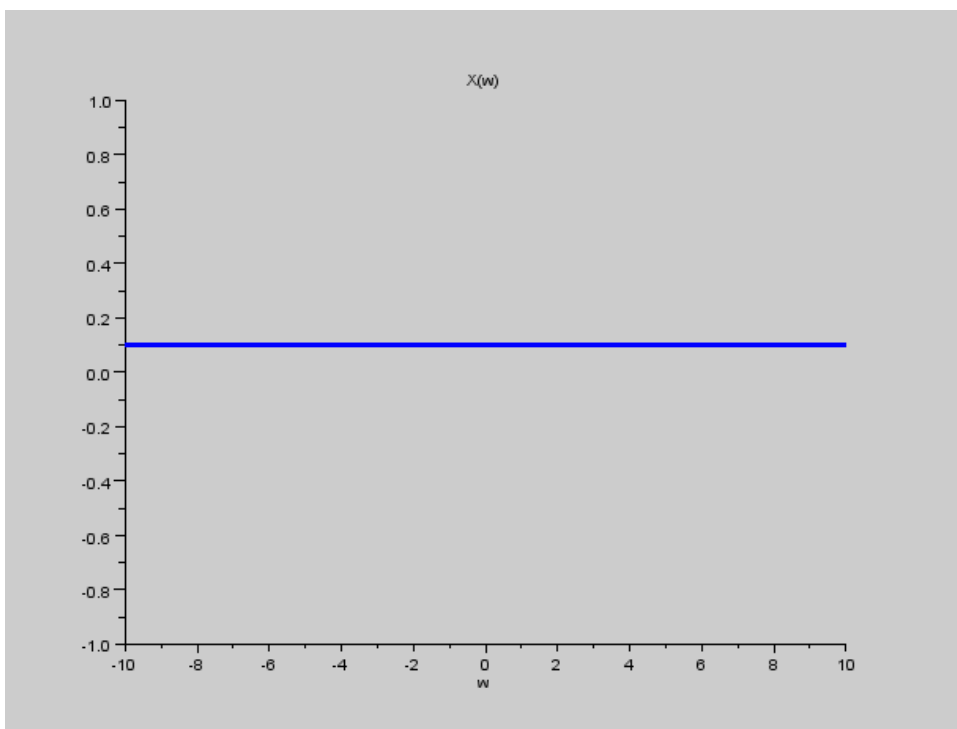


Figure 5.20: fourier transform



Scilab code Exa 5.23.b fourier transform

```
1 //frequency shifting
2 clear;
3 clc;
4 close;
5 dt=.1;
6 w0=1;//positive number
7 t=-10:dt:10;
8 x=exp(%i*t*w0);
9 a=gca();
10 plot(t,x);
11 poly1=a.children.children;
12 poly1.thickness=3;
13 poly1.foreground=2;
14 xtitle('x(t)', 't')
15 wmax=10;
16 w=-wmax:0.1:wmax;
17 Xw=x*exp(-%i*(w'*t))*dt;
18 figure
19 a=gca();
20 plot2d(w,round(Xw/10));
21 poly1=a.children.children;
22 poly1.thickness=3;
23 poly1.foreground=2;
24 xtitle('X(w)', 'w')
```

---

Scilab code Exa 5.23.c fourier transform

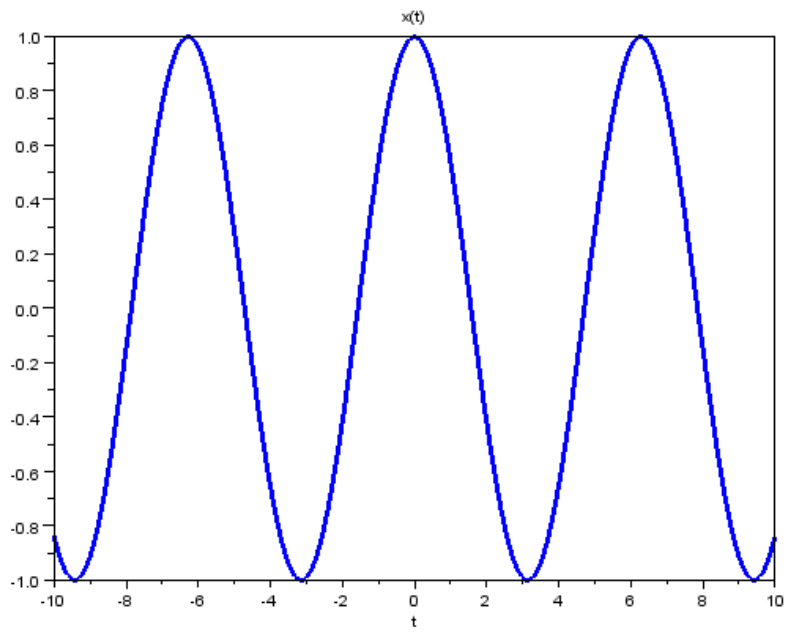


Figure 5.21: fourier transform

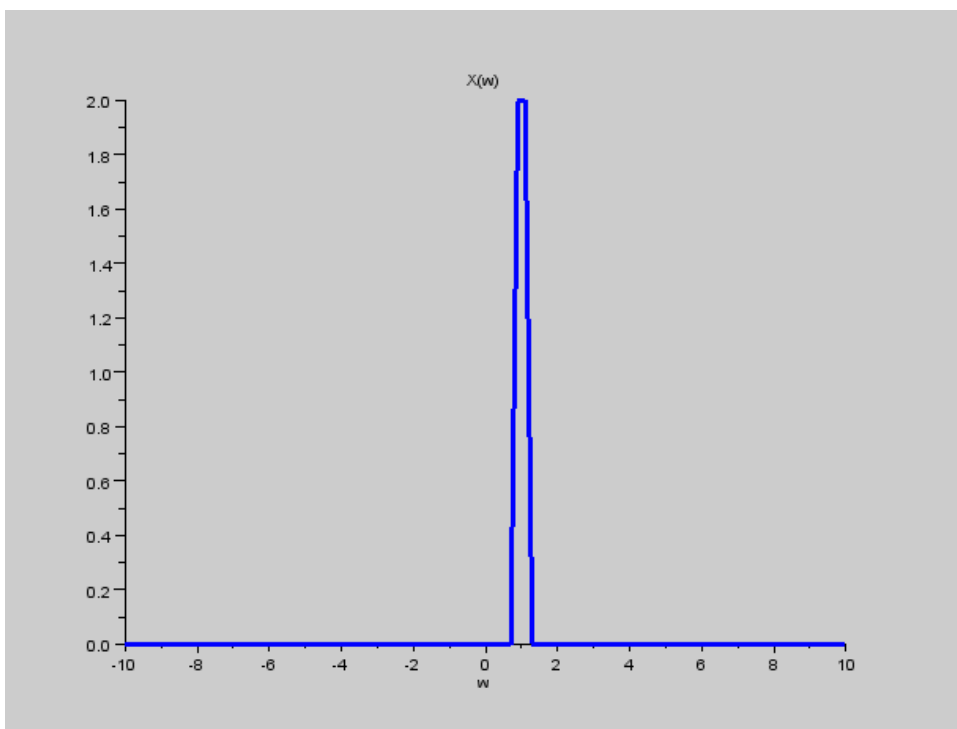


Figure 5.22: fourier transform

```

1 //frequency shifting
2 clear;
3 clc;
4 close;
5 dt=.1;
6 w0=1; //positive number
7 t=-10:dt:10;
8 x=exp(-%i*t*w0);
9 a=gca();
10 plot(t,x);
11 poly1=a.children.children;
12 poly1.thickness=3;
13 poly1.foreground=2;
14 xtitle('x(t)', 't')
15 wmax=10;
16 w=-wmax:0.1:wmax;
17 Xw=x*exp(-%i*(w'*t))*dt;
18 figure
19 a=gca();
20 plot2d3(w, round(Xw/10));
21 poly1=a.children.children;
22 poly1.thickness=1;
23 poly1.foreground=2;
24 xtitle('X(w)', 'w')
25 plot(w, round(Xw/10), 'b. ')

```

---

Scilab code Exa 5.23.d fourier transform

```

1 //frequency shifting
2 clear;
3 clc;

```

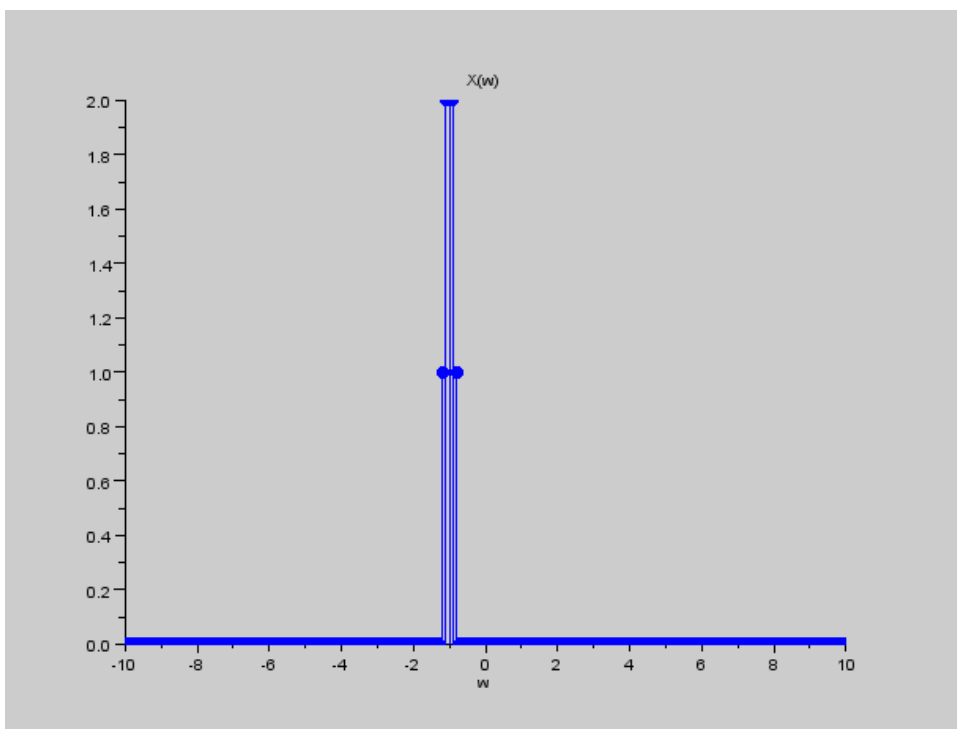


Figure 5.23: fourier transform

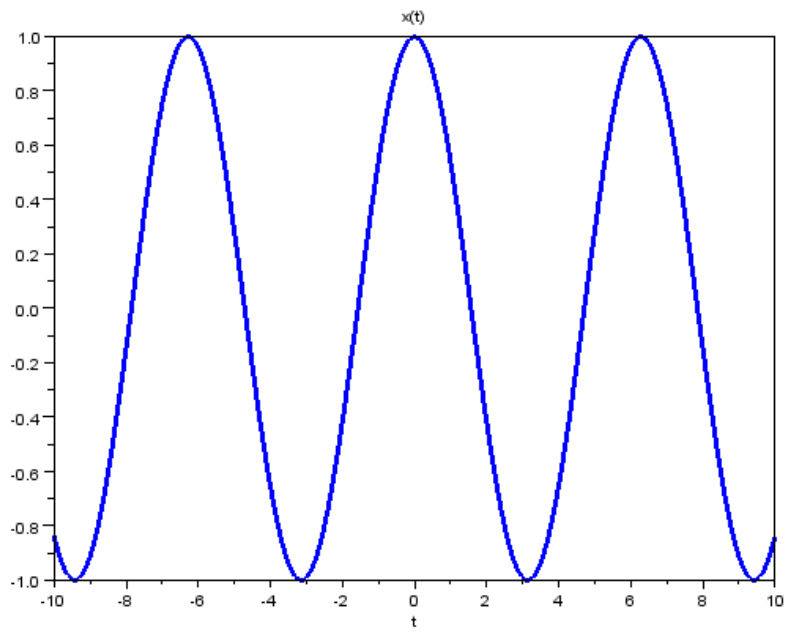


Figure 5.24: fourier transform

```

4 close;
5 dt=.1;
6 w0=2; //positive number
7 t=-10:dt:10;
8 x=cos(w0*t); //or 1/2*(exp(-j*w*t)+exp(j*w*t))
9 a=gca();
10 plot(t,x);
11 poly1=a.children.children;
12 poly1.thickness=3;
13 poly1.foreground=2;
14 xtitle('x(t)', 't')
15 wmax=10;
16 w=-wmax:0.1:wmax;
17 Xw=x*exp(-%i*(w'*t))*dt;
18 figure
19 a=gca();
20 plot2d3(w, round(abs(Xw/10)));
21 poly1=a.children.children;
22 poly1.thickness=2;
23 poly1.foreground=2;
24 xtitle('X(w)', 'w')

```

---

Scilab code Exa 5.23.e fourier transform

```

1 //frequency shifting
2 clear;
3 clc;
4 close;
5 dt=.1;
6 w0=.8; //positive number
7 t=-10:dt:10;

```

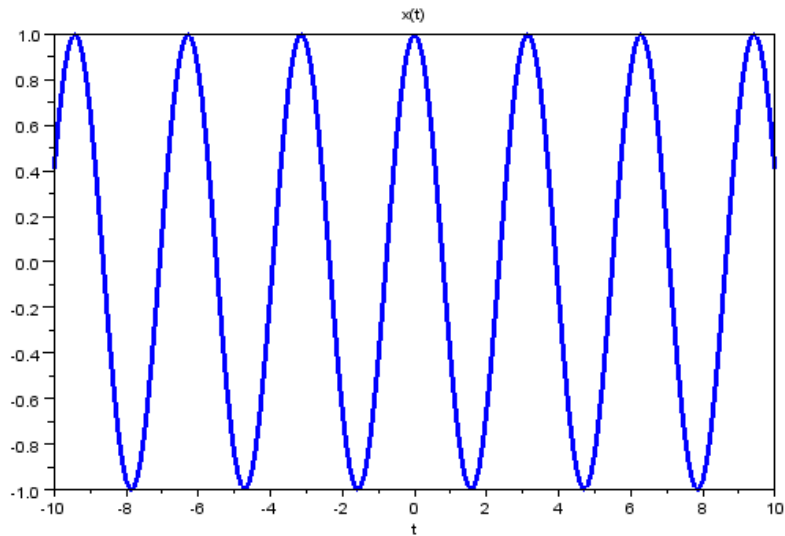


Figure 5.25: fourier transform

```

8 x=sin(w0*t); // or 1/2*(exp(-j*w*t)+exp(j*w*t))
9 plot(t,x);
10 a=gca()
11 poly1=a.children.children;
12 poly1.thickness=3;
13 poly1.foreground=2;
14 xtitle('x(t)', 't')
15 wmax=10;
16 w=-wmax:0.1:wmax;
17 Xw=x*exp(-%i*(w'*t))*dt;
18 figure
19 a=gca()
20 plot2d3(w, round(abs(Xw/10)));
21 poly1=a.children.children;
22 poly1.thickness=2;
23 poly1.foreground=2;
24 xtitle('X(w)', 'w')

```

---



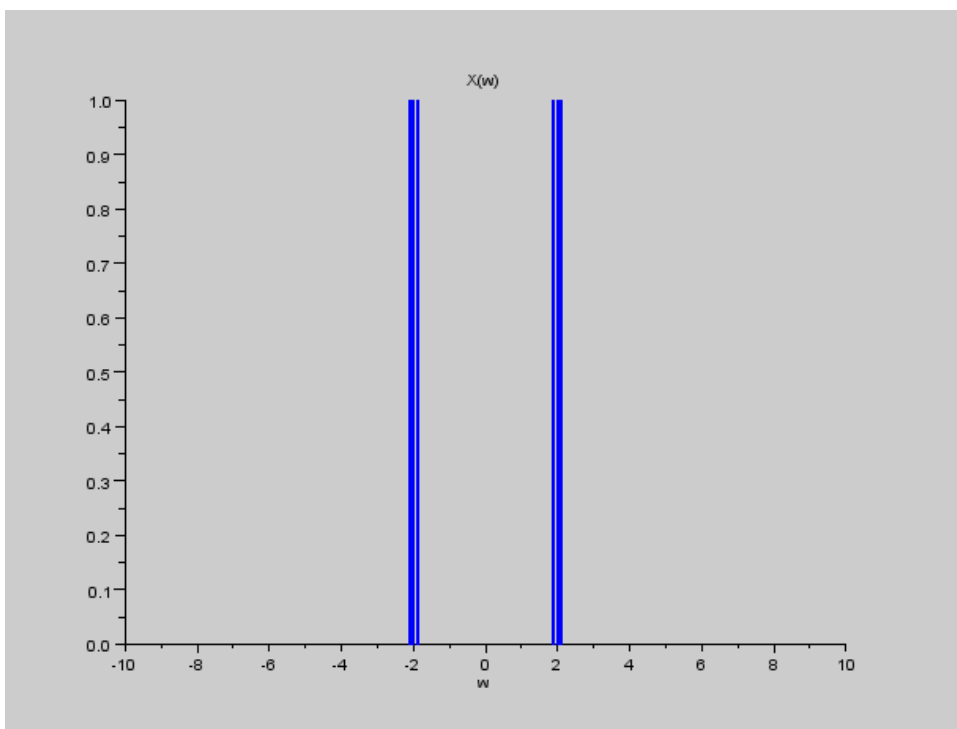


Figure 5.26: fourier transform

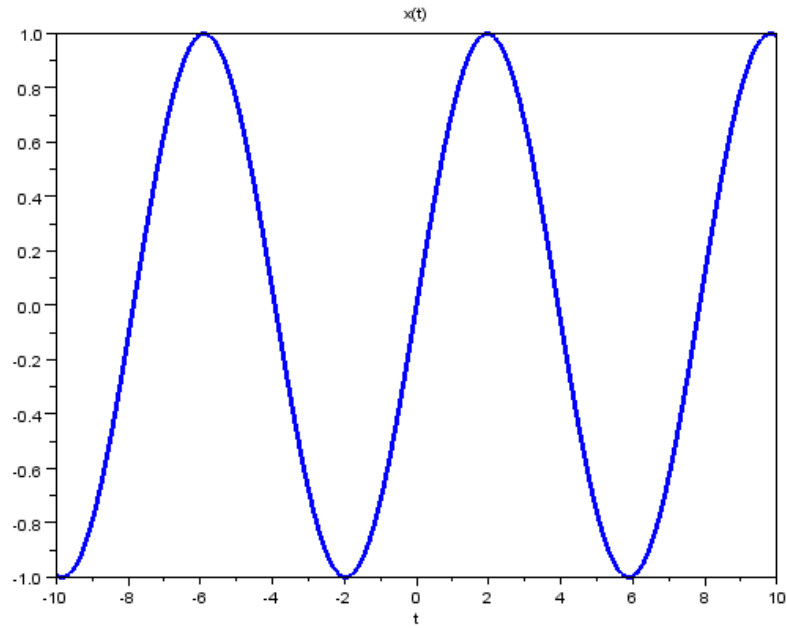


Figure 5.27: fourier transform

**Scilab code Exa 5.25** fourier transform of a periodic impulse train

```
1 clear;  
2 clc;  
3 close;  
4 dt=.1;  
5 t0=1; //positive number  
6 t=-10:dt:10;
```

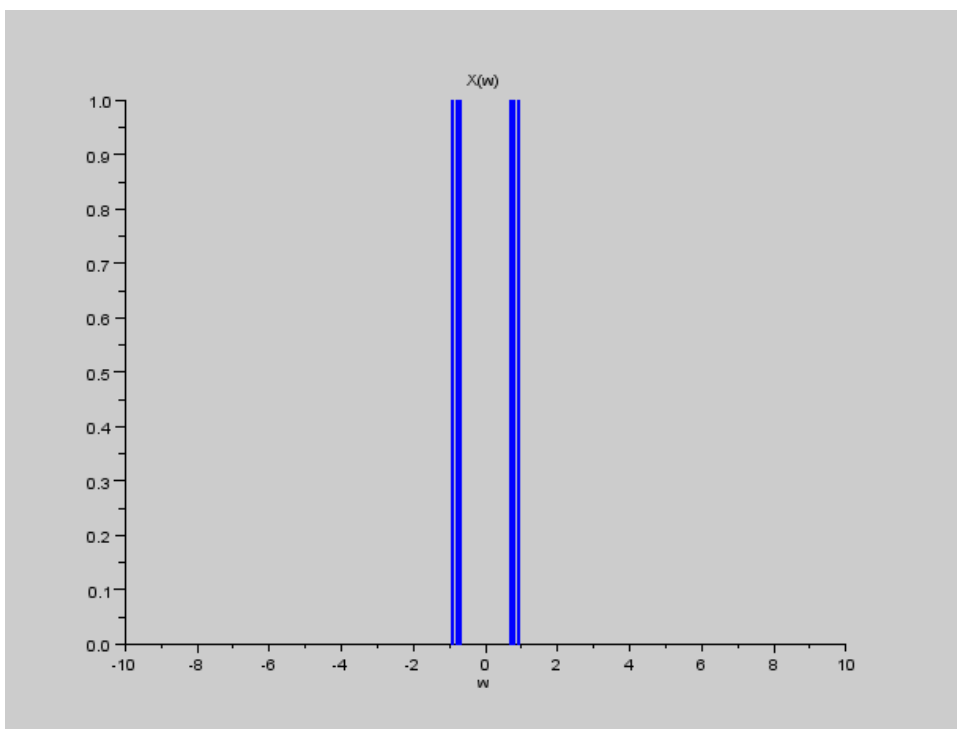


Figure 5.28: fourier transform

```

7 for i=1:length(t)
8     if modulo(t(i),t0)==0 then
9         x(i)=1;
10    else
11        x(i)=0;
12    end
13 end
14 a=gca();
15 plot2d3(t,x);
16 plot(t,x,'r. ');
17 poly1=a.children.children;
18 poly1.thickness=3;
19 poly1.foreground=2;
20 xtitle('x(t)', 't')
21 wmax=10;
22 w=-wmax:0.1:wmax;
23 Xw=x'*exp(-%i*(w'*t))*dt;
24 figure
25 a=gca();
26 plot2d3(w,round(abs(Xw)));
27 poly1=a.children.children;
28 poly1.thickness=2;
29 poly1.foreground=2;
30 xtitle('X(w)', 'w')
31 //or the fourier series is doesnt work
32 //ck=1/t0;
33 //k=-10:0.1:10;
34 //x=ck*(exp(%i*2*%pi*t.*k/t0));
35 //wmax=10;
36 //w=-wmax:0.1:wmax;
37 //Xw=x*exp(-%i*(w'*t))*dt;
38 //clf();
39 //plot2d3(k,x);

```

---

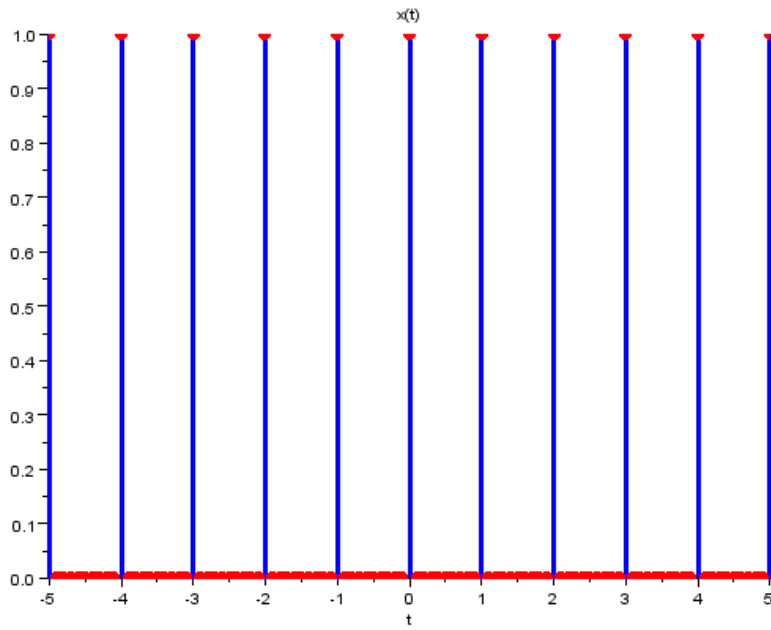


Figure 5.29: fourier transform of a periodic impulse train

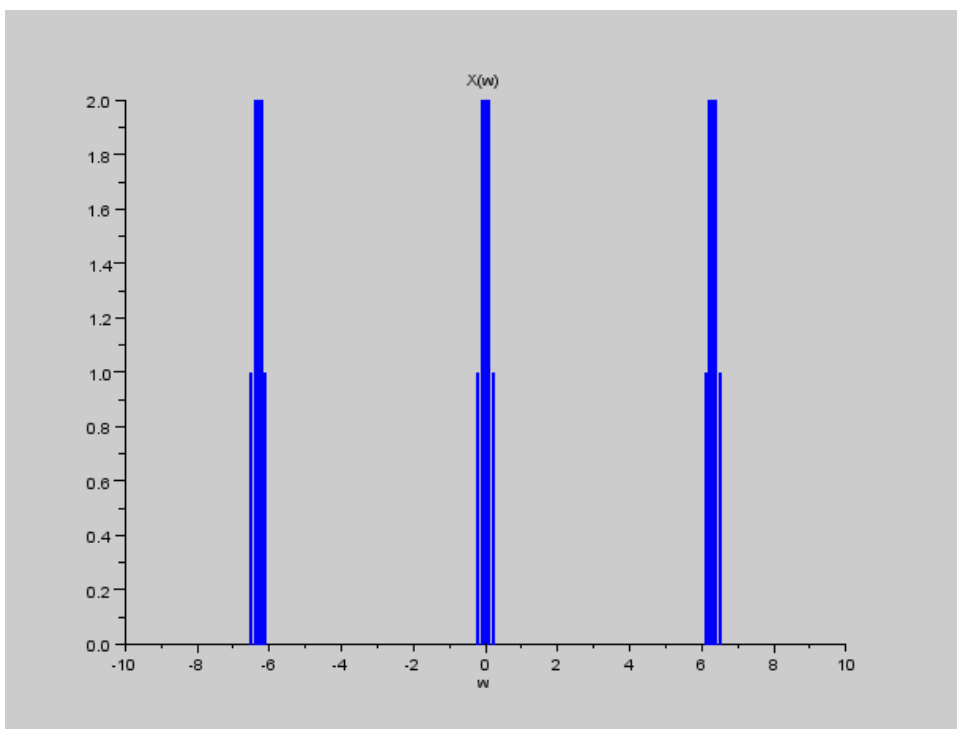


Figure 5.30: fourier transform of a periodic impulse train

### Scilab code Exa 5.27 inverse fourier transform

```
1 clear;
2 clc;
3 close;
4 A=1/2;
5 dw=.1;
6 w0=4;
7 a=1;
8 w=-10:dw:10;
9 for i=1:length(w)
10     if ((w(i)>(-w0-a) & w(i)<(-w0+a)) | (w(i)>(w0-a)
        & w(i)<(w0+a))) then
11         Xw(i)=2;
12     else
13         Xw(i)=0;
14     end
15
16 end
17 a=gca();
18 plot(w,Xw);
19 poly1=a.children.children;
20 poly1.thickness=3;
21 poly1.foreground=2;
22 xtitle('X(w)', 'w')
23 tmax=10;
24 t=-tmax:0.1:tmax;
25 x=Xw'*exp(-%i*(t'*w))*dw;
26 x_mag=(x);
27 figure
28 a=gca();
29 plot2d(t,x_mag);
30 poly1=a.children.children;
31 poly1.thickness=3;
```

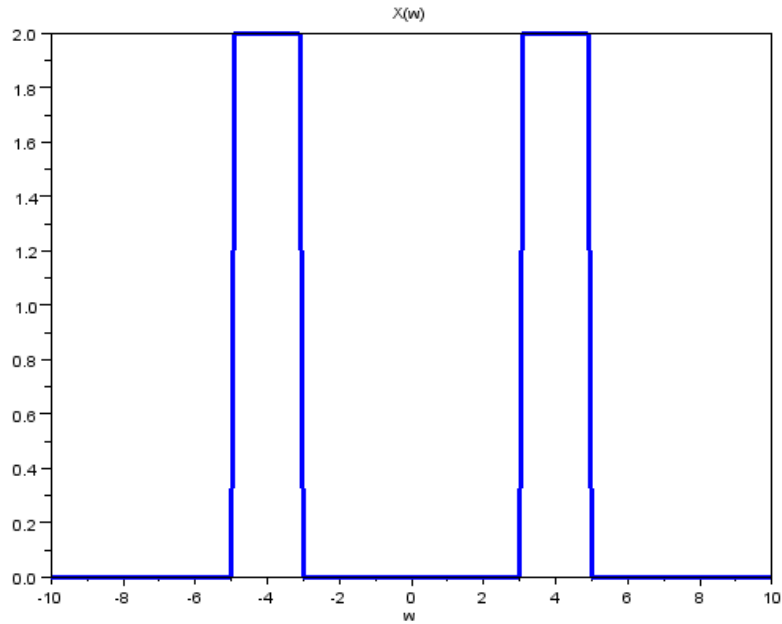


Figure 5.31: inverse fourier transform

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

---

**Scilab code Exa 5.29** fourier transform of a signum function

```
1 clear;  
2 clc;  
3 close;  
4 dt=.1;
```



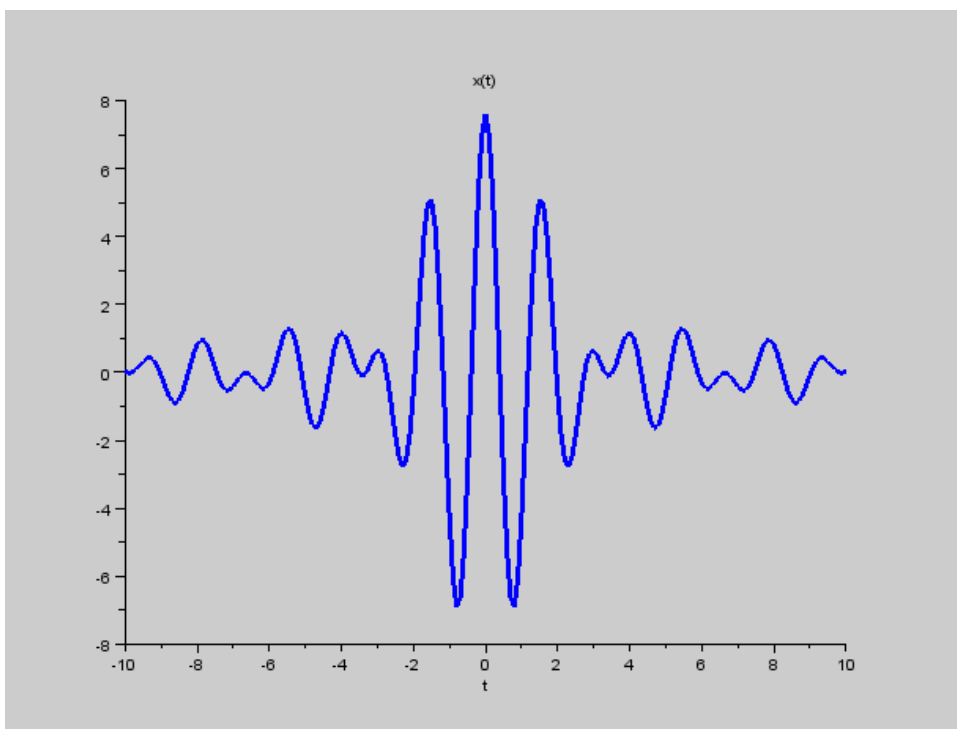


Figure 5.32: inverse fourier transform

```

5 t=-10:dt:10;
6 h=find(t==0);
7 x=[-ones(1,h) ones(1,length(t)-h)];
8 //sgn(t)=2u(t)-1  ∴ d(sgn(t))/dt=2delta(t);
9 a=gca();
10 plot(t,x);
11 poly1=a.children.children;
12 poly1.thickness=3;
13 poly1.foreground=2;
14 xtitle('sgn(t)', 't')
15 y=diff(x);
16 y=[y 0];
17 figure
18 a=gca();
19 plot2d3(t,y);
20 poly1=a.children.children;
21 poly1.thickness=3;
22 poly1.foreground=2;
23 xtitle('diff(sgn(t))', 't')
24 wmax=10;
25 w=-wmax:0.1:wmax;
26 Xw=y*exp(-%i*(w'*t))*dt*10;
27 for i=1:length(w)
28     if w(i)<>0 then
29         XW(i)=Xw(i)./(%i*w(i));
30     else
31         XW(i)=20;
32     end
33 end
34 figure
35 a=gca();
36 plot2d(w,abs(XW));
37 poly1=a.children.children;
38 poly1.thickness=3;
39 poly1.foreground=2;
40 xtitle('X(w)', 'w')

```

---

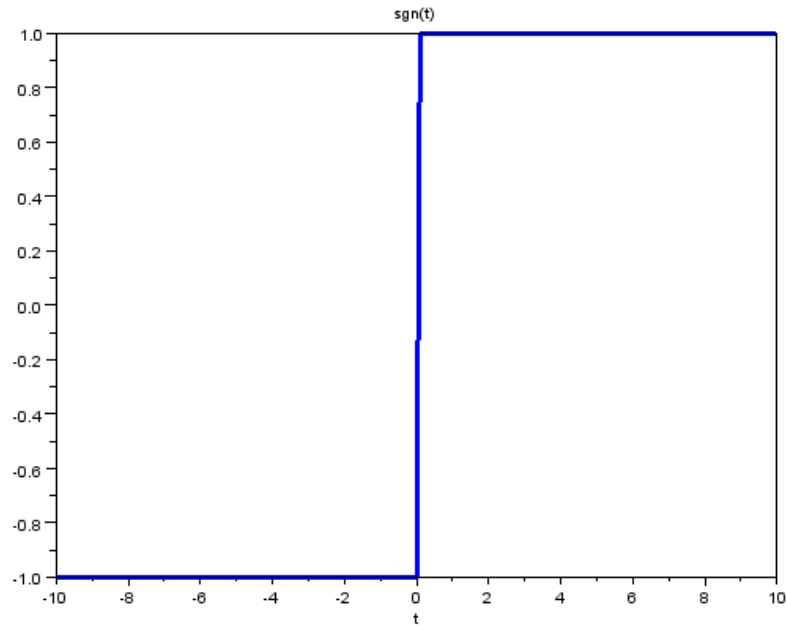


Figure 5.33: fourier transform of a signum function

**Scilab code Exa 5.30** fourier transform of a step signal

```

1  clear;
2  clc;
3  close;
4  dt=.1;
5  t=-10:dt:10;
6  u=[zeros(1,find(t==0)) ones(1,length(t)-find(t==0))]

```

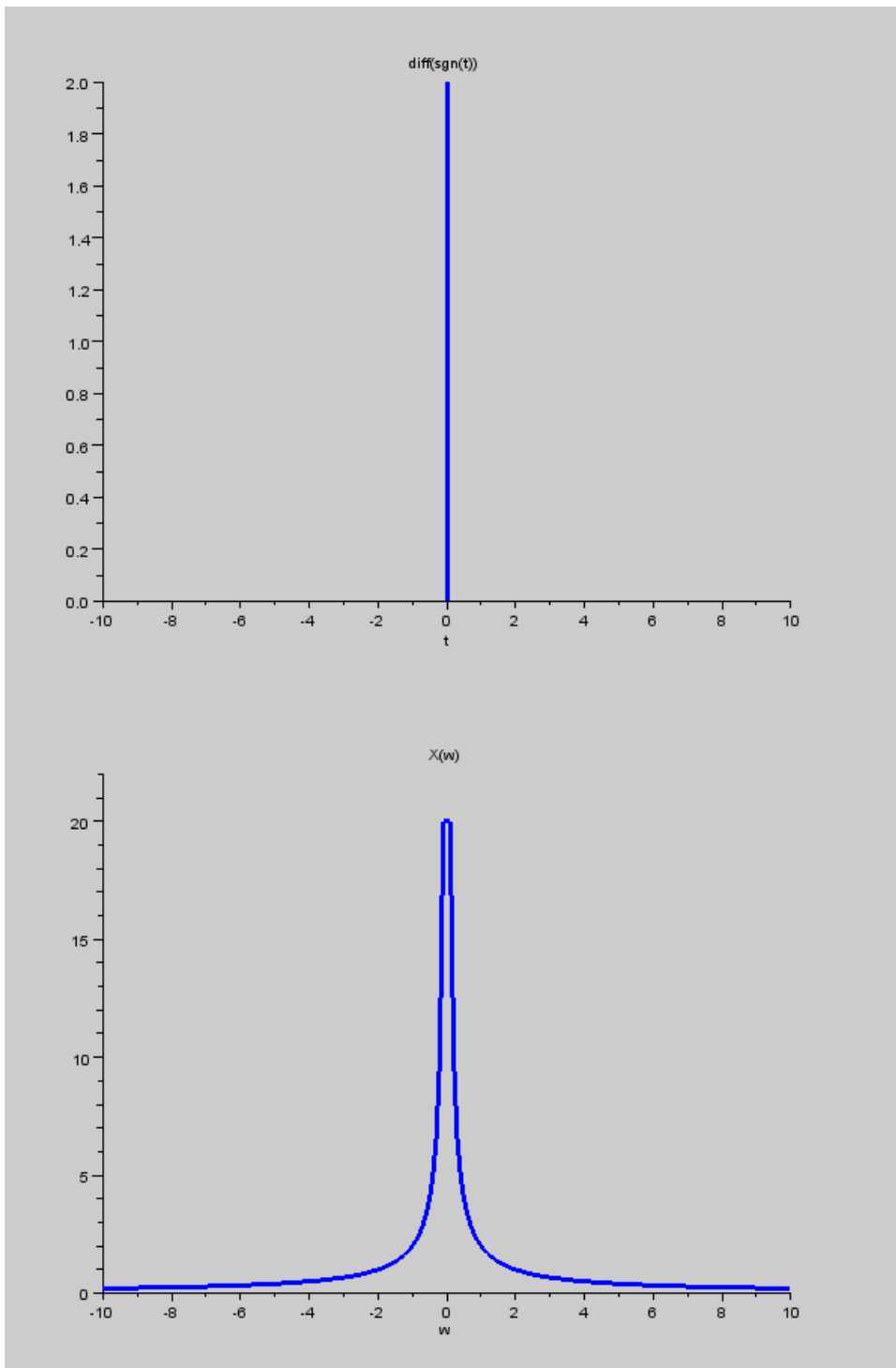


Figure 5.34: fourier transform of a signum function

```

];
7 a=gca();
8 plot(t,u);
9 poly1=a.children.children;
10 poly1.thickness=3;
11 poly1.foreground=2;
12 xtitle('u(t)', 't')
13 u1=1/2; //even part
14 figure
15 a=gca()
16 plot(t,u1);
17 poly1=a.children.children;
18 poly1.thickness=3;
19 poly1.foreground=2;
20 xtitle('even part of u(t)', 't')
21 h=find(t==0);
22 u2=[-ones(1,h) ones(1,length(t)-h)]; //odd part
23 figure
24 a=gca()
25 plot(t,u2);
26 poly1=a.children.children;
27 poly1.thickness=3;
28 poly1.foreground=2;
29 xtitle('odd part of u(t)', 't')
30 //u(t)=u1(t)+u2(t)
31 //.: U[w]=U1[w]+U2[w] i.e U[w]=%pi*delta(w)+1/(%i*w)
32 w=-10:0.1:10;
33 for i=1:length(w)
34     if w(i)==0 then
35         delta(i)=1;
36     else
37         delta(i)=0;
38     end
39 end
40 Uw=ones(1,length(w))./(%i*w)+%pi*delta';
41 figure
42 a=gca()
43 plot2d(w,abs(Uw));

```

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

---

**Scilab code Exa 5.32** inverse fourier transform using convolution

```
1 clear;
2 clc;
3 close;
4 disp("X(w)=1/(a+jw)^2=1/(a+jw)*1/(a+jw)");
5 disp("exp(-a*t)*u(t) <--> 1/(a+jw)");
6 disp("therefore x(t)=convolution(exp(-a*t)*u(t),exp(-a*t)*u(t))");
7 a=2;
8 t=0:0.1:10;
9 y=exp(-a*t);
10 z=convol(y,y);
11 disp("t*exp(-a*t)*u(t) <--> 1/(a+jw)^2");
12 t1=[-t($:-1:2) t];
13 a=gca();
14 plot(t1,z);
15 poly1=a.children.children;
16 poly1.thickness=3;
17 poly1.foreground=2;
18 xtitle('x(t)', 't')
```

---

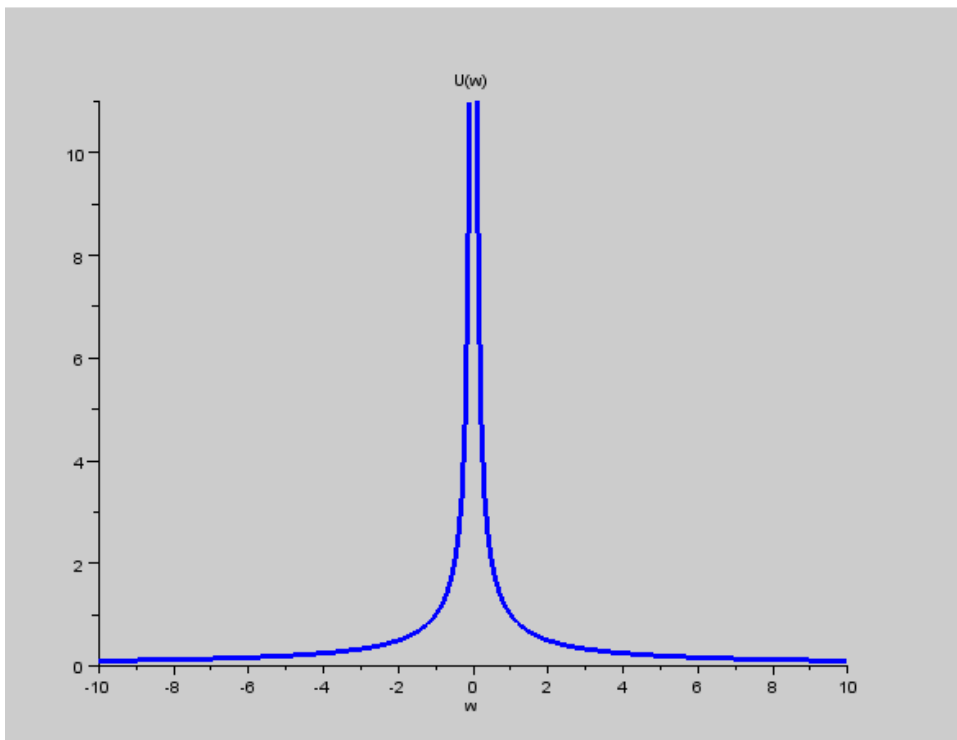
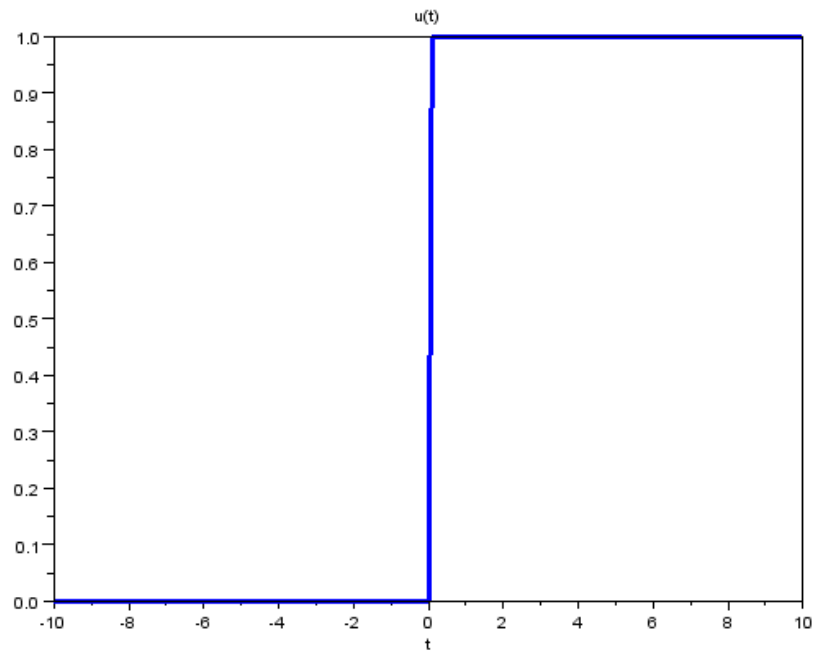


Figure 5.35: fourier transform of a step signal

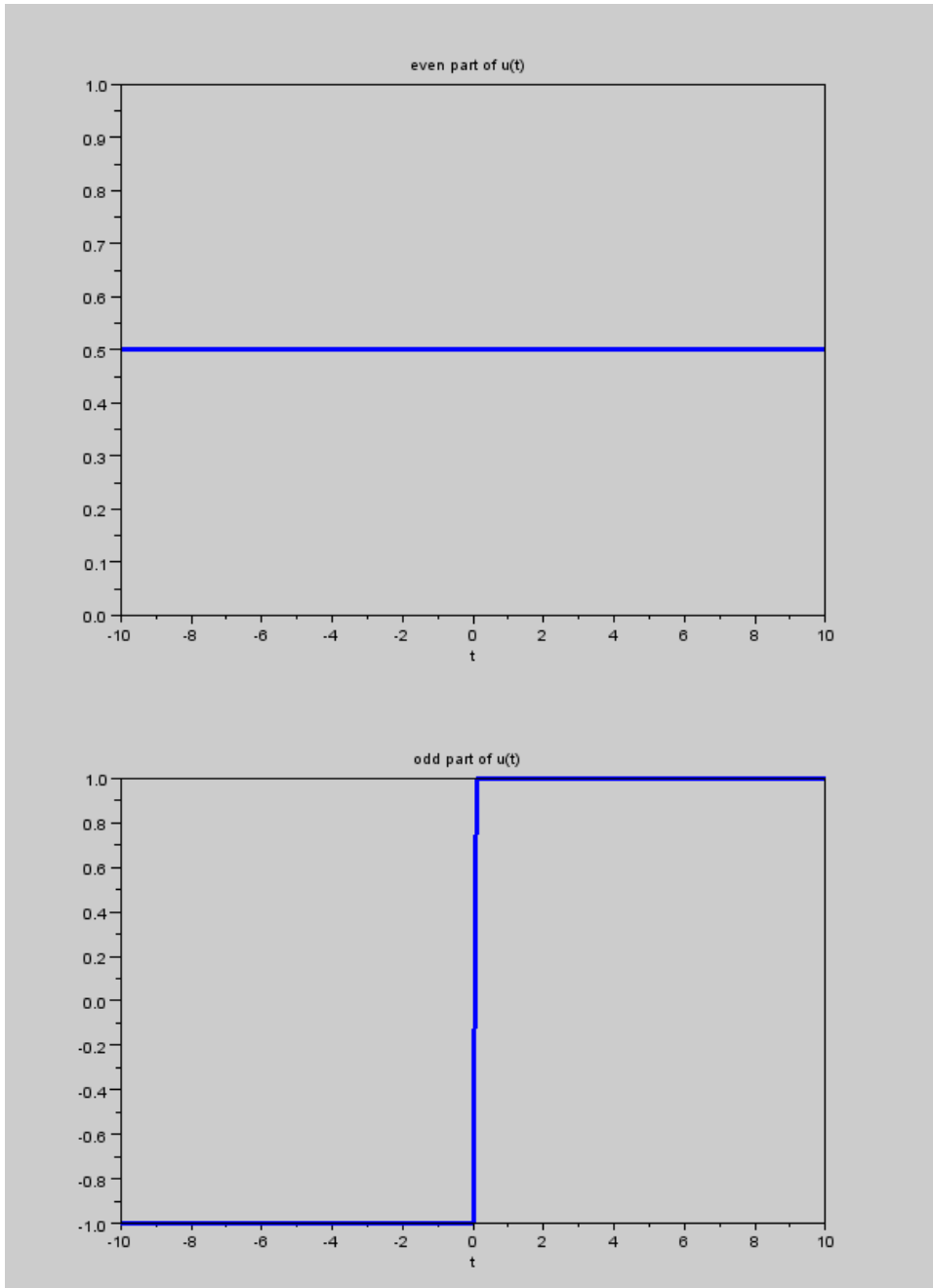


Figure 5.36: fourier transform of a step signal



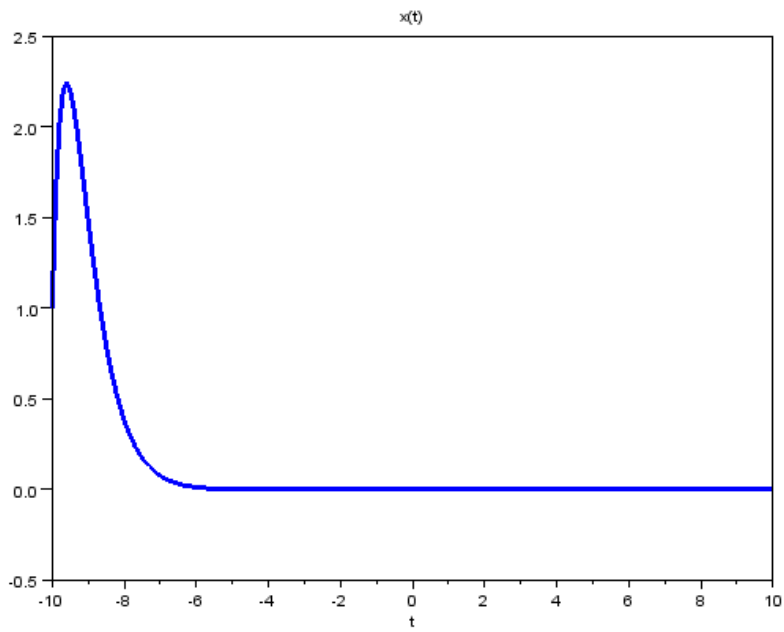


Figure 5.37: inverse fourier transform using convolution

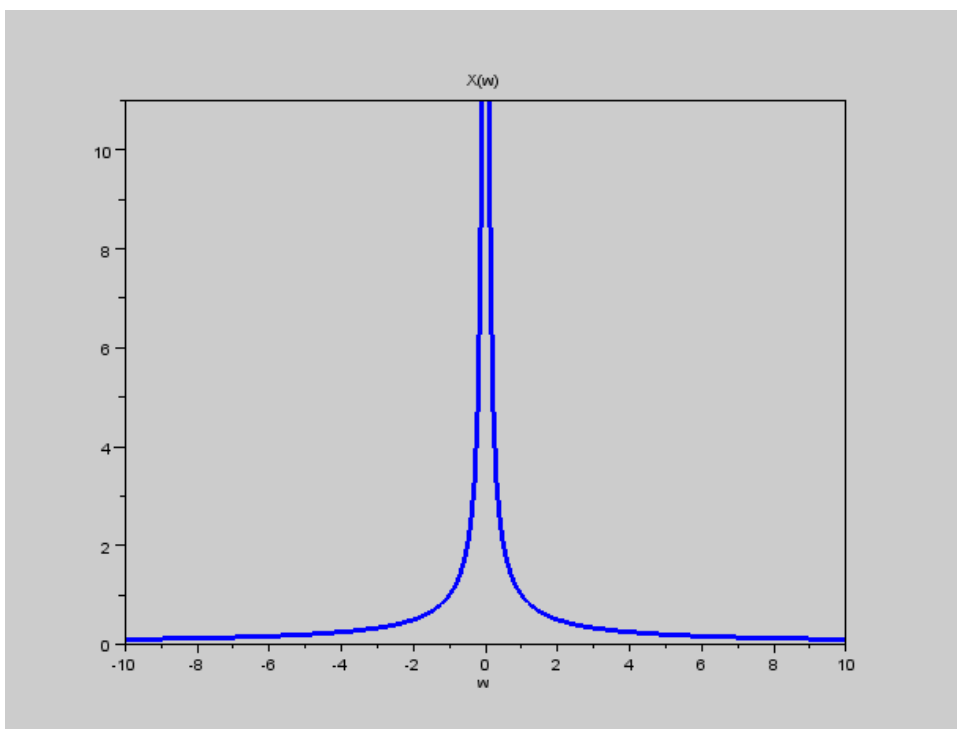


Figure 5.38: integration property

### Scilab code Exa 5.34 integration property

```
1 clear;
2 clc;
3 close;
4 disp("u(t)=integral(delta(t))");
5 disp("integral(x(t)) <--> %pi*X[0]*delta(w)+X(w)/(%i
      *w)");
6 disp("delta(t) <--> 1");
7 w=-10:0.1:10;
8 for i=1:length(w)
9     if w(i)==0 then
10         delta(i)=1;
11     else
12         delta(i)=0;
13     end
14 end
15 Xw=%pi*delta'+ones(1,length(w))./(%i*w);
16 disp('U[w]=%pi*delta(w)+1/(%i*w)');
17 figure
18 a=gca();
19 plot(w,abs(Xw));
20 poly1=a.children.children;
21 poly1.thickness=3;
22 poly1.foreground=2;
23 xtitle('X(w)', 'w')
```

---

### Scilab code Exa 5.40.a fourier transform

```
1 clear;
```

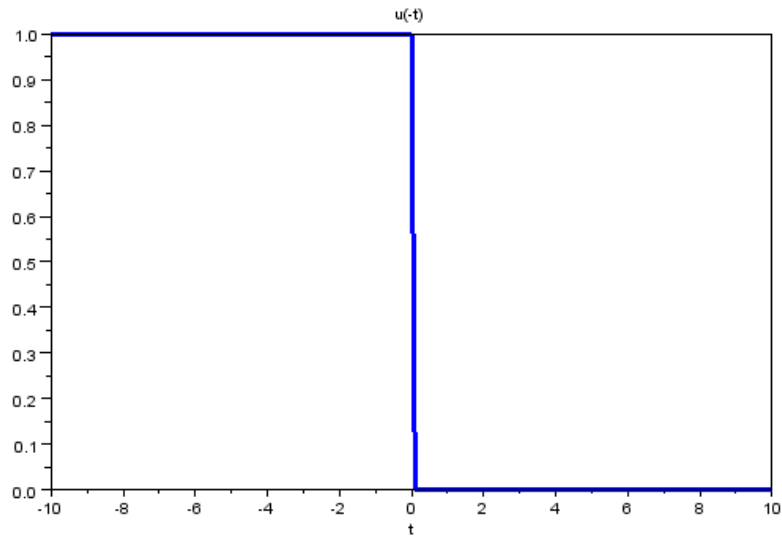


Figure 5.39: fourier transform

```

2  clc;
3  close;
4  disp('x(-t) <--> X(-w)=X*(w)');
5  t=-10:0.1:10;
6  u=[ones(1,find(t==0)) zeros(1,length(t)-find(t==0))
    ];
7  a=gca();
8  plot(t,u);
9  poly1=a.children.children;
10 poly1.thickness=3;
11 poly1.foreground=2;
12 xtitle('u(-t)', 't')
13 w=-10:0.1:10;
14 for i=1:length(w)
15     if w(i)==0 then
16         delta(i)=1;
17     else
18         delta(i)=0;

```

```

19     end
20 end
21 Xw=%pi*delta'-ones(1,length(w))./(%i*w);
22 disp('U[-w]=%pi*delta(w)-1/(%i*w)');

```

---

### Scilab code Exa 5.40.b fourier transform

```

1 clear;
2 clc;
3 close;
4 disp('x(-t) <--> X(-w)=X*(w)');
5 a=0.1;
6 t=-10:0.1:10;
7 u=[ones(1,find(t==0)) zeros(1,length(t)-find(t==0))
    ];
8 x=exp(-a*t).*u;
9 d=gca();
10 plot(t,x);
11 poly1=d.children.children;
12 poly1.thickness=3;
13 poly1.foreground=2;
14 xtitle('e^(a*t)*u(-t)', 't');
15 w=-10:0.1:10;
16 Xw=ones(1,length(w))./(a-(%i*w));
17 disp('exp(-a*t)*u(-t) <--> 1/(a-(%i*w))');
18 figure
19 d=gca();
20 plot(w,abs(Xw));
21 poly1=d.children.children;
22 poly1.thickness=3;
23 poly1.foreground=2;
24 xtitle('U(w)', 'w')

```

---

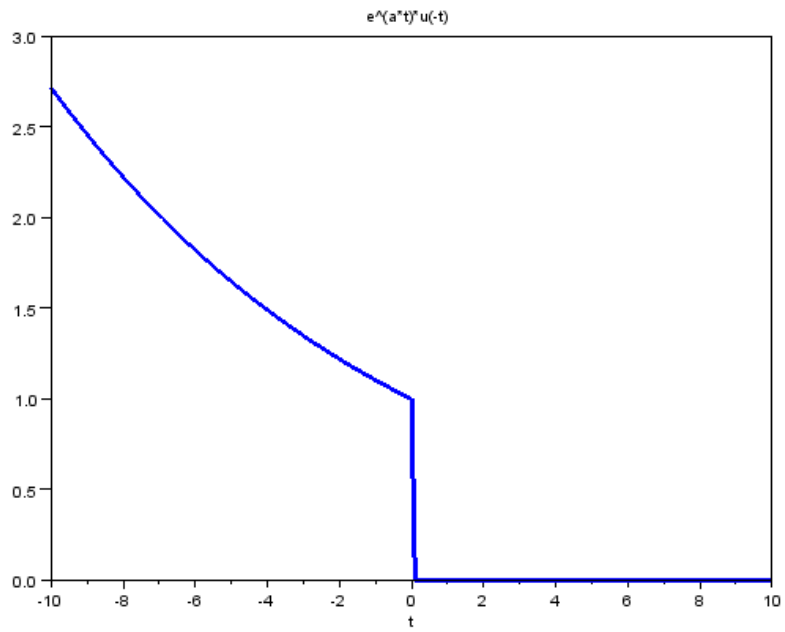


Figure 5.40: fourier transform

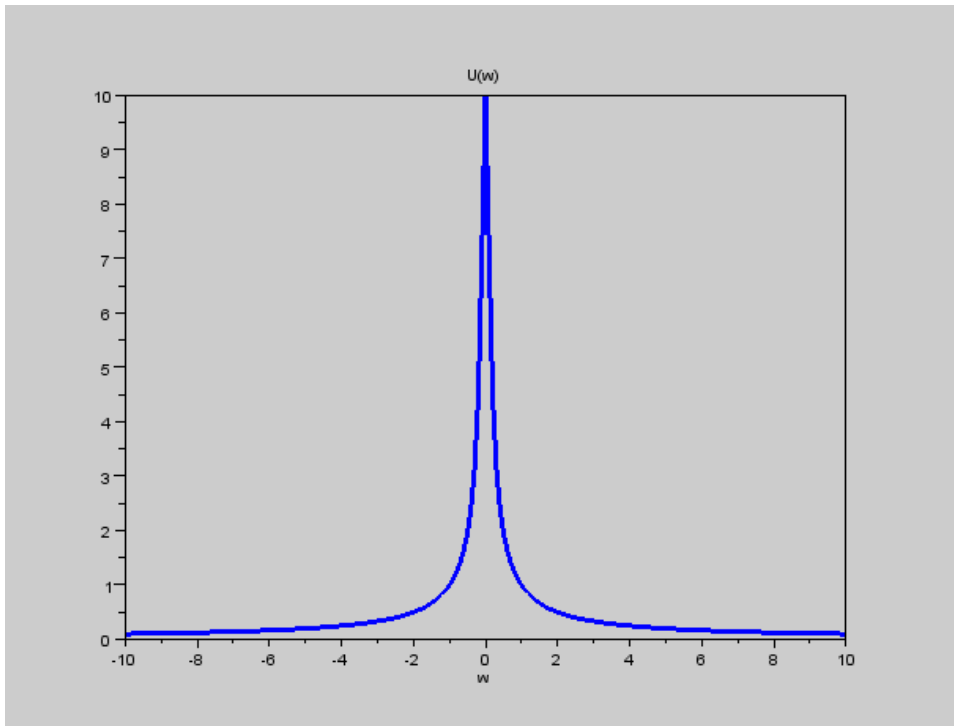


Figure 5.41: fourier transform

**Scilab code Exa 5.42** fourier transform of a exponential signal

```

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

```

```

8 plot(t,x);
9 poly1=d.children.children;
10 poly1.thickness=3;
11 poly1.foreground=2;
12 xtitle('x(t)', 't')
13 disp("1/2*exp(-a*|t|) is even part of 1/2*exp(-a*t)*
      u(t)");
14 disp("even(x(t)) <--> real(X(w))=a/(a^2+w^2)");
15 w=-10:0.01:10;
16 Xw=2*a*ones(1,length(w))./(a^2+w^2);
17 figure
18 d=gca()
19 plot(w,Xw);
20 poly1=d.children.children;
21 poly1.thickness=3;
22 poly1.foreground=2;
23 xtitle('X(w)', 'w')
24 //not sure if it works properly

```

---

**Scilab code Exa 5.43** fourier transform of a gaussian pulse

```

1 clear;
2 clc;
3 close;
4 a=.5;
5 t=-10:0.1:10;
6 x=exp(-a*t.*t);
7 disp("gaussian pulse signal x(t)=exp(-a*t^2)");
8 disp("X(w)=integral(exp(-a*t^2)*exp(-%i*w*t)) w.r.t
      dt");
9 disp("d(X(w))/dw=-%i*w/(2*a)*integral(exp(-a*t^2)*

```



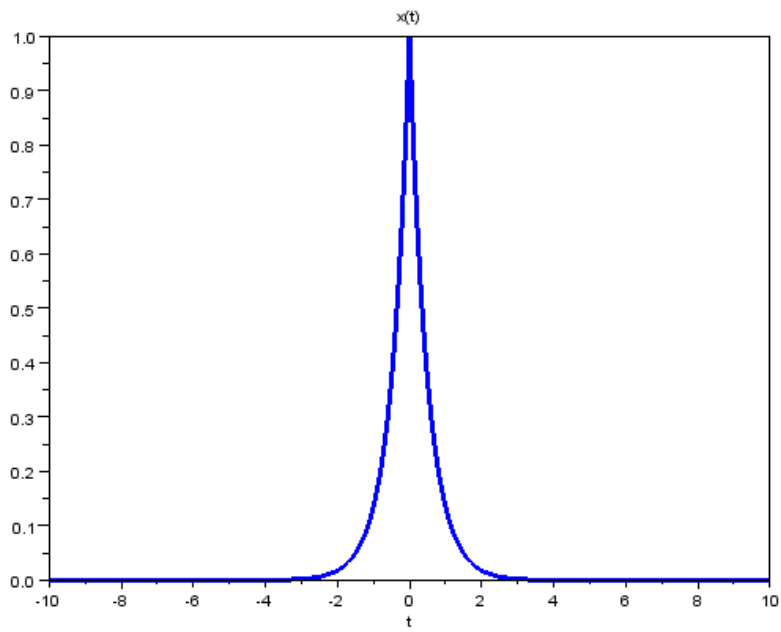


Figure 5.42: fourier transform of a exponential signal

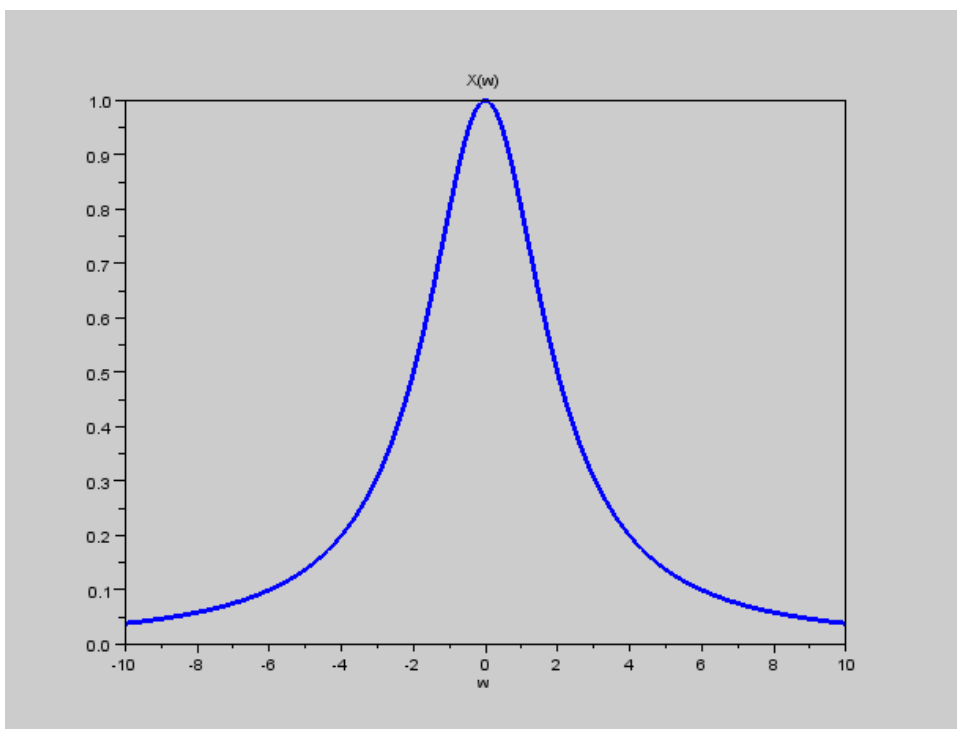


Figure 5.43: fourier transform of a exponential signal

```

    exp(-%i*w*t))");
10 disp("d(X(w))/dw=-w*X(w)/2a");
11 disp("solving this we get X(w)=A*exp(-w^2/4a)")
12 disp("A=sqrt(%pi/a)");
13 d=gca();
14 plot(t,x);
15 poly1=d.children.children;
16 poly1.thickness=3;
17 poly1.foreground=2;
18 xtitle('x(t)', 't')
19 A=sqrt(%pi/a);
20 w=t;
21 Xw=A*exp(-w.*w/(4*a));
22 figure
23 d=gca();
24 plot(w,Xw);
25 poly1=d.children.children;
26 poly1.thickness=3;
27 poly1.foreground=2;
28 xtitle('X(w)', 'w')

```

---

**Scilab code Exa 5.44** impulse response using fourier transform

```

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

```

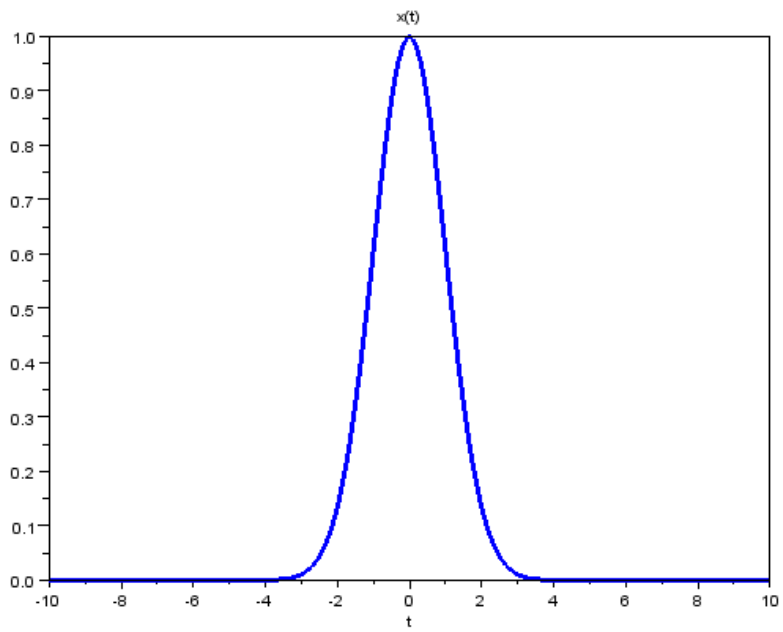


Figure 5.44: fourier transform of a gaussian pulse

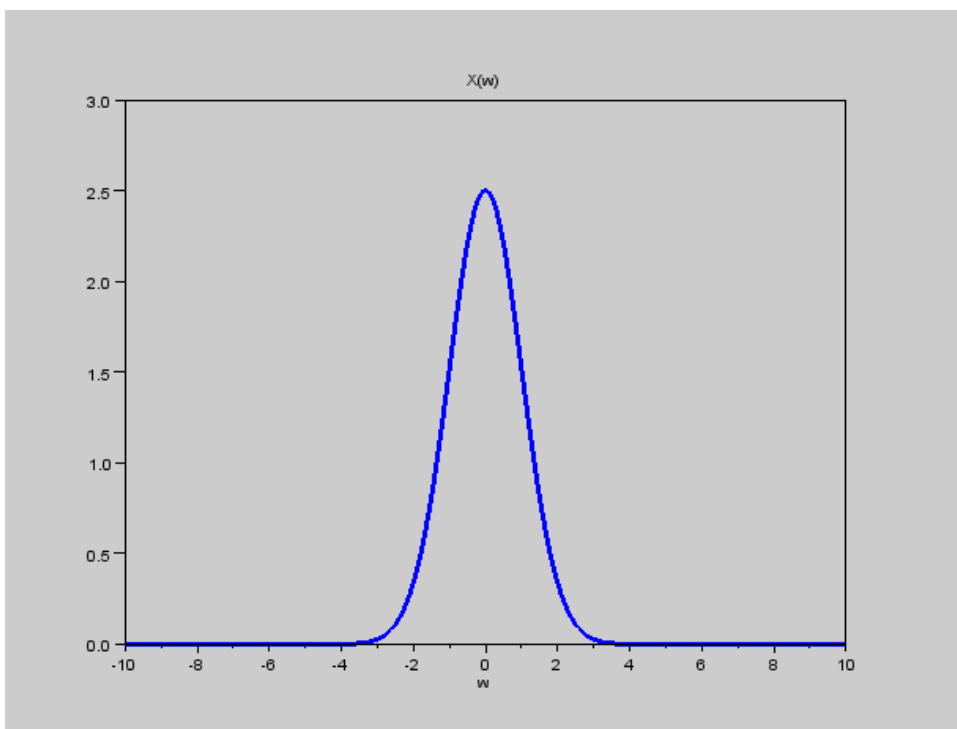


Figure 5.45: fourier transform of a gaussian pulse

```

7 w=-10:0.1:10;
8 dw=.1;
9 Hw=1-ones(1,length(w))./(2+%i*w);
10 t=0:0.1:10;
11 d=gca()
12 plot(w,Hw);
13 poly1=d.children.children;
14 poly1.thickness=3;
15 poly1.foreground=2;
16 xtitle('X(w)', 'w')
17 for i=1:length(t)
18     if t(i)==0 then
19         delta(i)=1;
20     else
21         delta(i)=0;
22     end
23 end
24 h=delta'-exp(-2*t);
25 figure;
26 d=gca()
27 plot(t,(h));
28 poly1=d.children.children;
29 poly1.thickness=3;
30 poly1.foreground=2;
31 xtitle('h(t)', 't')

```

---

Scilab code Exa 5.45.a output response using fourier transform

```

1 clear;
2 clc;
3 close;

```

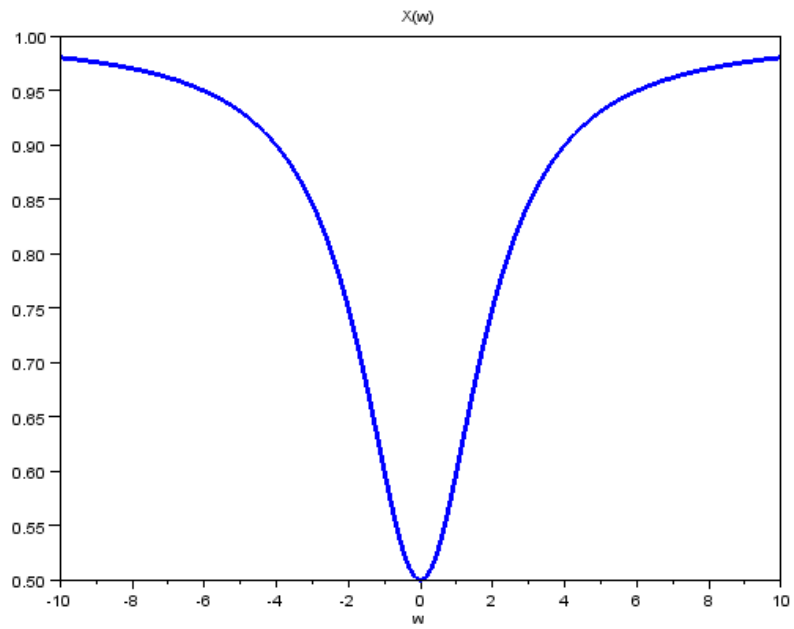


Figure 5.46: impulse response using fourier transform

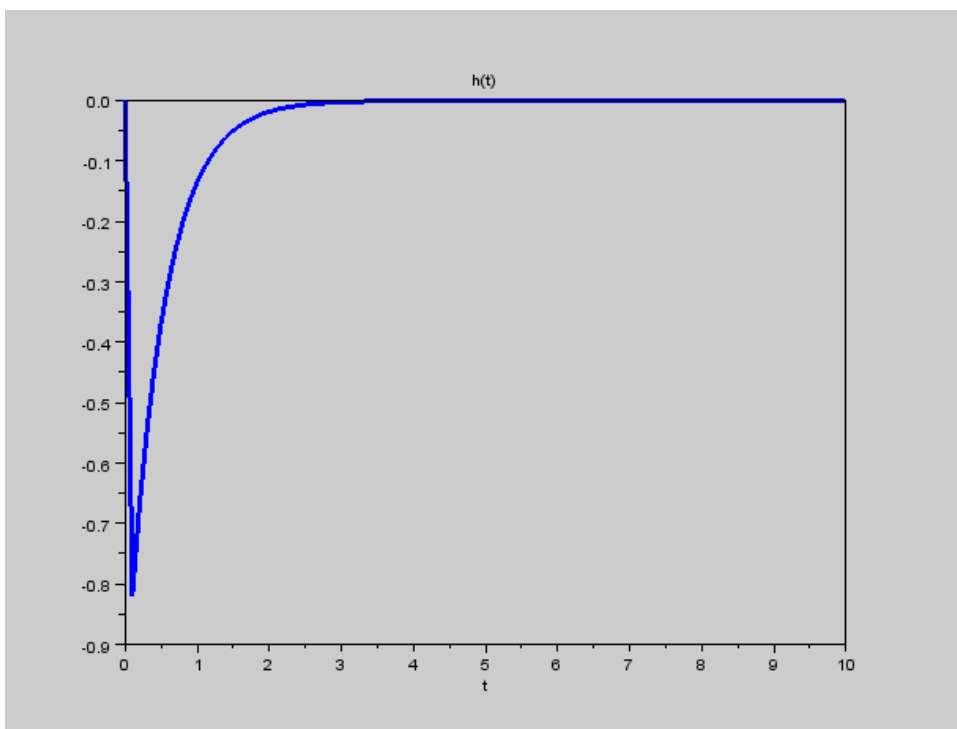


Figure 5.47: impulse response using fourier transform



```

4 disp("dy(t)/dt+2y(t)=x(t)");
5 w=0:0.1:10;
6 t=w;
7 dw=.1;
8 Xw=ones(1,length(w))./(1+%i*w);
9 Hw=ones(1,length(w))./(2+%i*w);
10 Yw=Xw.*Hw;
11 y=Yw*exp(%i*t'*w)*dw*.31;
12 d=gca()
13 plot(t,y);
14 poly1=d.children.children;
15 poly1.thickness=3;
16 poly1.foreground=2;
17 xtitle('y(t)', 't')
18 yy=exp(-t)-exp(-2*t);
19 disp("y(t)=exp(-t)-exp(-2*t)")
20 figure
21 d=gca()
22 plot(t,yy);
23 poly1=d.children.children;
24 poly1.thickness=3;
25 poly1.foreground=2;
26 xtitle('y(t)', 't')

```

---

Scilab code Exa 5.45.b output response using fourier transform

```

1 clear;
2 clc;
3 close;
4 disp("dy(t)/dt+2y(t)=x(t)");
5 w=0.1:0.1:10;

```

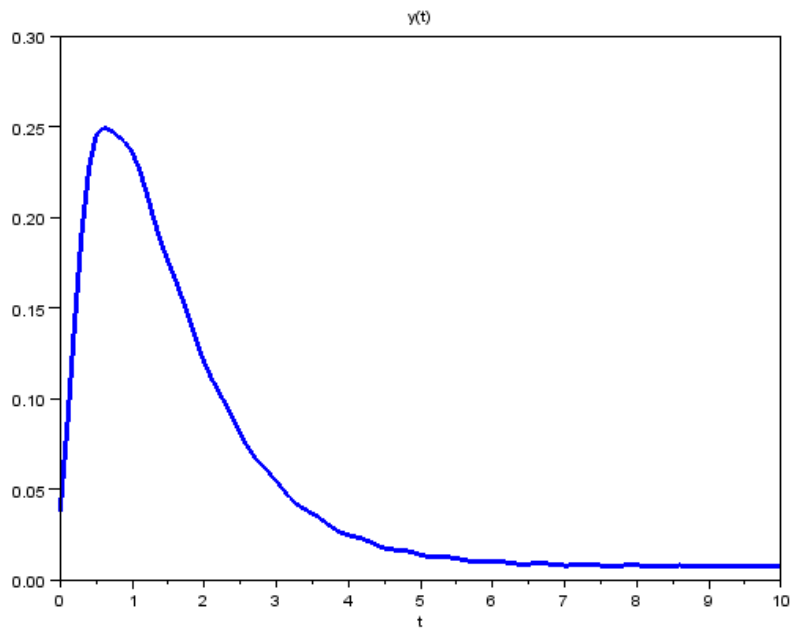


Figure 5.48: output response using fourier transform

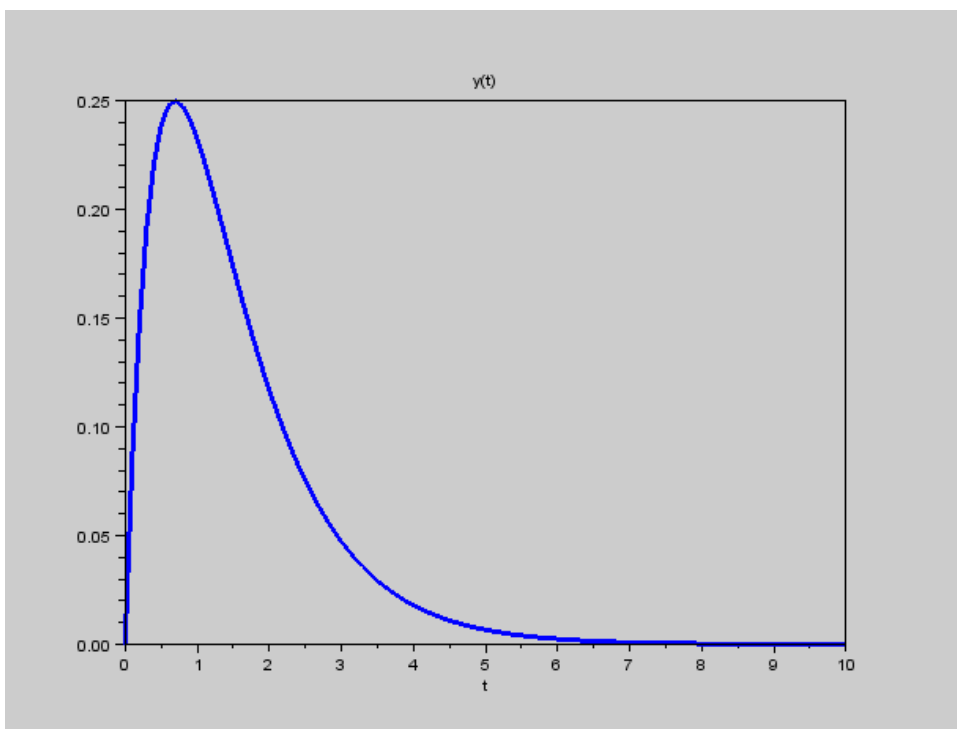


Figure 5.49: output response using fourier transform

```

6 t=w;
7 dw=.1;
8 Xw=ones(1,length(w))./(%i*w);
9 Hw=ones(1,length(w))./(2+%i*w);
10 Yw=Xw.*Hw;
11 y=Yw*exp(%i*t'*w)*dw;
12 d=gca()
13 plot(t,y);
14 poly1=d.children.children;
15 poly1.thickness=3;
16 poly1.foreground=2;
17 xtitle('y(t)', 't')
18 yy=0.5*(1-exp(-2*t));
19 disp("y(t)=0.5(1-exp(-2*t))")
20 figure
21 d=gca()
22 plot(t,yy);
23 poly1=d.children.children;
24 poly1.thickness=3;
25 poly1.foreground=2;
26 xtitle('y(t)', 't')

```

---

**Scilab code Exa 5.46** harmonics in the output response

```

1 clear;
2 clc;
3 close;
4 T0=1;
5 t=-5.99:0.01:6;

```

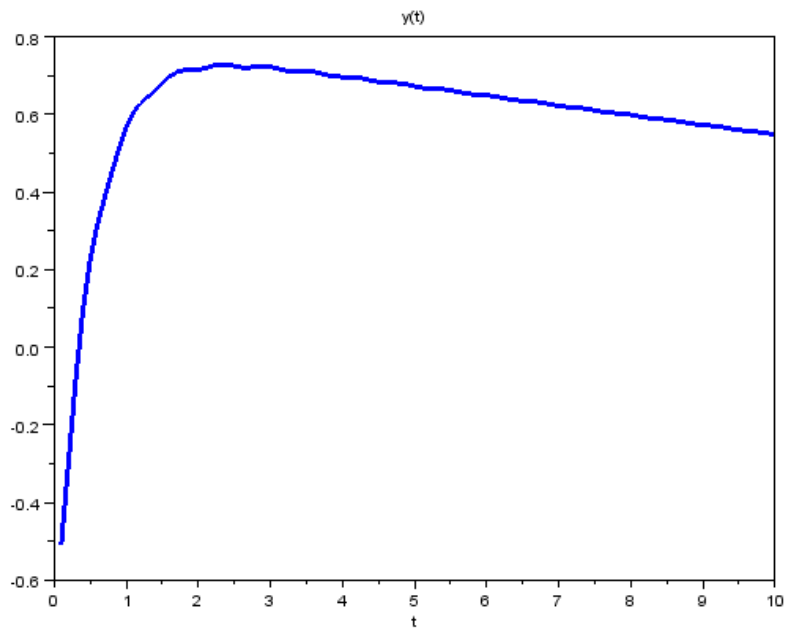


Figure 5.50: output response using fourier transform

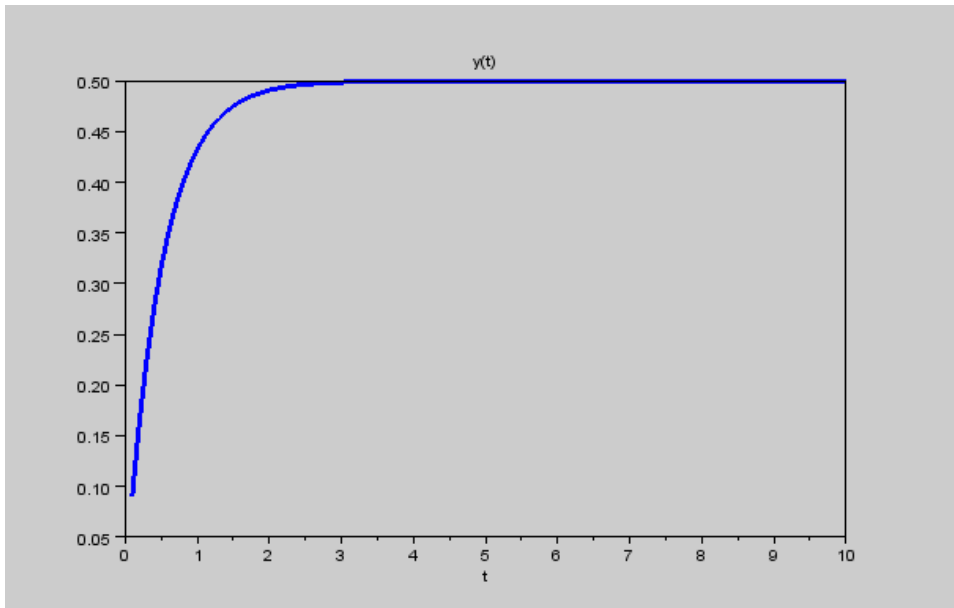


Figure 5.51: output response using fourier transform

```

6 t_temp=0.01:0.01:T0;
7 s=length(t)/length(t_temp);
8 x=[];
9 for i=1:s
10     if modulo(i,2)==1 then
11         x=[x zeros(1,length(t_temp))];
12     else
13         x=[x 10*ones(1,length(t_temp))];
14     end
15 end
16 d=gca()
17 plot(t,x,'r')
18 poly1=d.children.children;
19 poly1.thickness=3;
20 poly1.foreground=2;
21 xtitle('x(t)', 't')
22 disp("y(t)=sum(H(kw0)*exp(%j*k*w0*t))");
23 //fourier series of x(t)

```

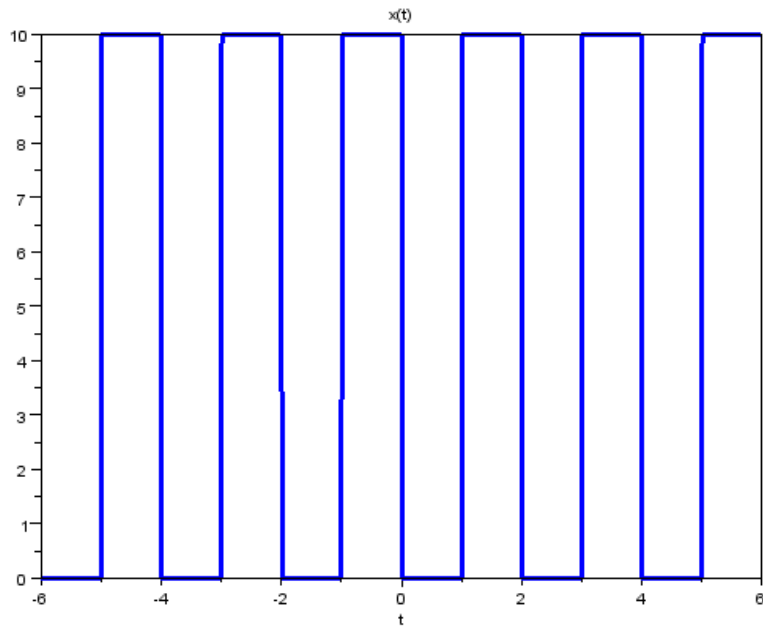


Figure 5.52: harmonics in the output response

```

24 w0=%pi;
25 for k=-10:10
26     cc(k+11,:)=exp(-%i*k*w0*t);
27     ck(k+11)=x*cc(k+11,:)/length(t);
28     if abs(ck(k+11))<0.01 then
29         ck(k+11)=0;
30     else if real(ck(k+11))<0.1 then
31         ck(k+11)=%i*imag(ck(k+11));
32     end
33     end
34     if k==1 then
35         c1=ck(k+11);
36     end
37     if k==3 then
38         c3=ck(k+11);
39     end
40 end
41 yc1=2*abs(c1/(2+%i*w0));
42 yc3=2*abs(c3/(2+%i*w0*3));
43 disp(yc1," first harmonic is");
44 disp(yc3," third harmonic is");

```

---

Scilab code Exa 5.47.a bode plot

```

1 clear;
2 clc;
3 fmin=0.1;
4 fmax=100;
5 f=fmin:0.1:fmax;
6 Hdb=20*log(abs(1+%i*f/10));
7 figure
8 a=gca();
9 a.data_bounds=[fmin,-20;fmax,40];

```



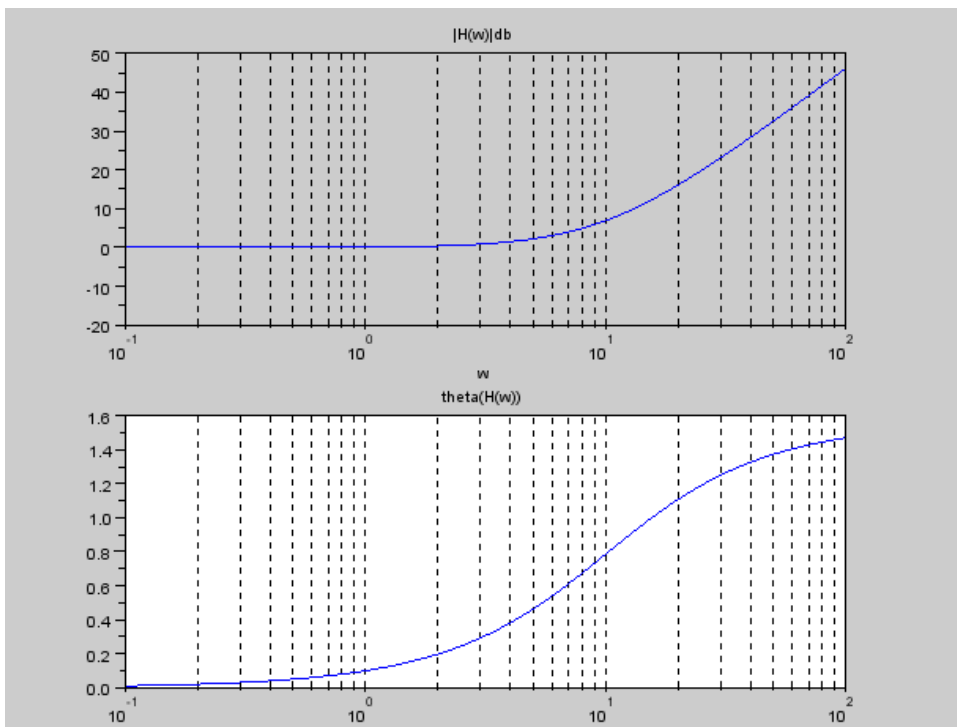


Figure 5.53: bode plot

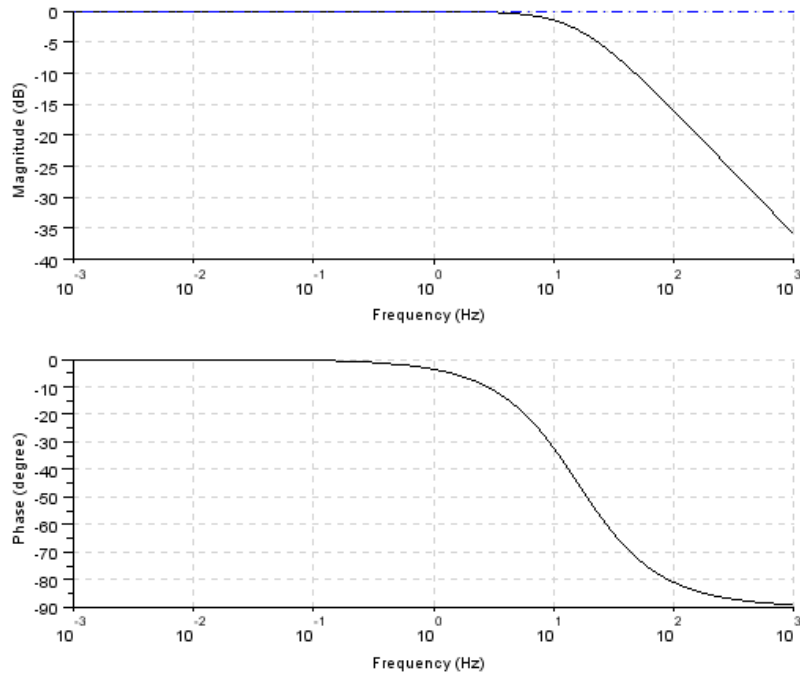


Figure 5.54: bode plot

```

10 a.log_flags="lnn";
11 subplot(2,1,1)
12 plot(f,Hdb);
13 xtitle(' |H(w) | db ', 'w')
14 subplot(2,1,2)
15 plot(f,atan(f/10));
16 xtitle(' theta(H(w)) ', 'w')
17 a=gca();
18 a.data_bounds=[fmin,0;fmax,%pi/2];
19 a.log_flags="lnn";

```

---

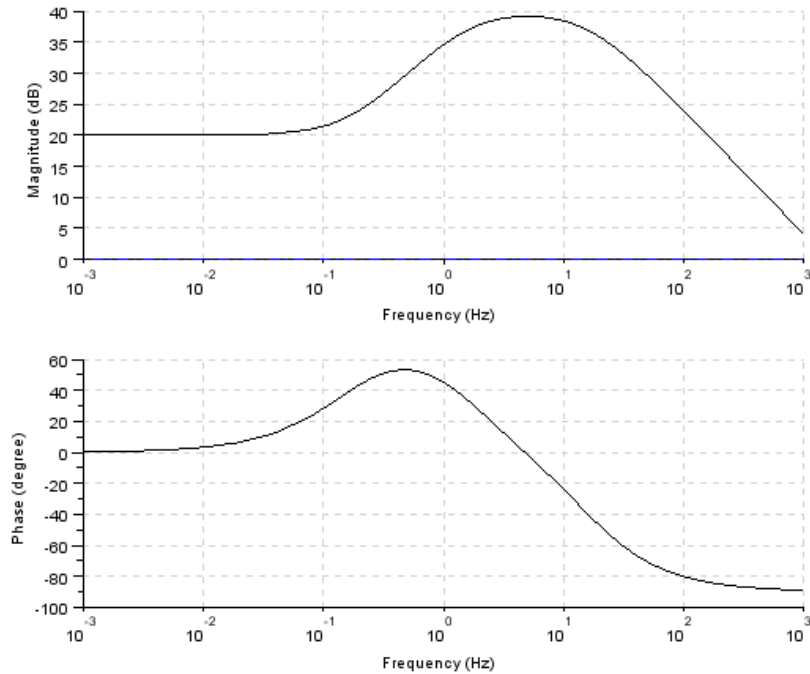


Figure 5.55: bode plot

Scilab code Exa 5.47.b bode plot

```

1 clear;
2 clc;
3 s=poly(0, 's');
4 H=syslin('c', [100/(100+s)]) ;
5 fmin=0.1;
6 fmax=100;
7 scf(1);clf;
8 bode(H, fmin, fmax);
9 show_margins(H) ;

```

---

### Scilab code Exa 5.47.c bode plot

```
1 clear;
2 clc;
3 s=poly(0, 's');
4 H=symlin('c', [(10^4+s*10^4)/((100+s)*(10+s))] );
5 a=pfss(H);
6 fmin=0.1;
7 fmax=1000;
8 scf(1);clf;
9 bode(a(1),fmin,fmax);
10 show_margins(H) ;
```

---

### Scilab code Exa 5.48 impulse response of a phase shifter

```
1 clear;
2 clc;
3 close;
4 w=-10:0.1:10;
5 Hw=[exp(%i*%pi/2)*ones(1,find(w==0)) exp(-%i*%pi/2)*
      ones(1,find(w==0)-1)];
6 d=gca();
7 plot(w, imag(Hw));
8 poly1=d.children.children;
9 poly1.thickness=3;
10 poly1.foreground=2;
11 xtitle('H(w)', 'w')
12 disp("H(w)=-%i*sgn(w)");
13 disp("we know sgn(t) <--> 2/(j*w)");
14 disp("by duality property 2/(j*t) <--> 2*%pi*sgn(-w)
      =-2*%pi*sgn(w) ");
15 disp("therefore 1/(%pi*t) <--> -j*sgn(w)");
16 t=0.1:0.1:10;
17 h=ones(1,length(t))./(%pi*t);
18 figure
```

```

19 d=gca()
20 plot(t,h);
21 poly1=d.children.children;
22 poly1.thickness=3;
23 poly1.foreground=2;
24 xtitle('h(t)', 't')
25 w0=2;
26 x=cos(w0*t);
27 figure
28 d=gca();
29 plot(t,x);
30 poly1=d.children.children;
31 poly1.thickness=3;
32 poly1.foreground=2;
33 xtitle('x(t)', 't')
34 y=convol(x,h);
35 figure
36 d=gca()
37 plot(t,y(1:length(t)));
38 poly1=d.children.children;
39 poly1.thickness=3;
40 poly1.foreground=2;
41 xtitle('y(t)', 't')

```

---

**Scilab code Exa 5.52** output of a ideal LPF

```

1 clear;
2 clc;
3 close;
4 w=-10:0.1:10;
5 wc=2;

```

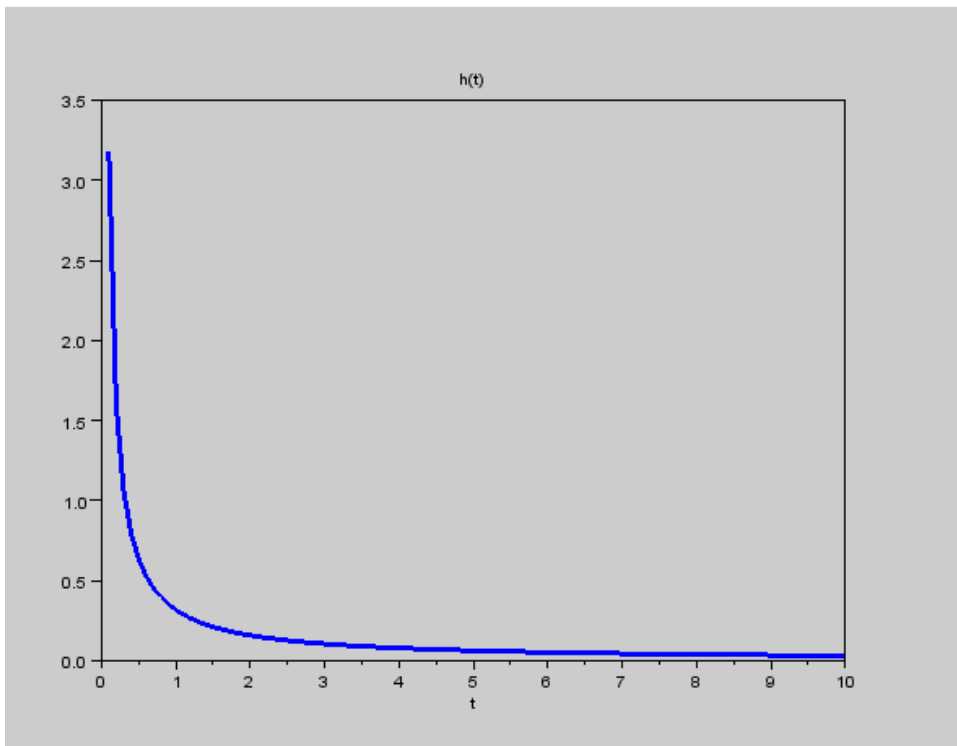
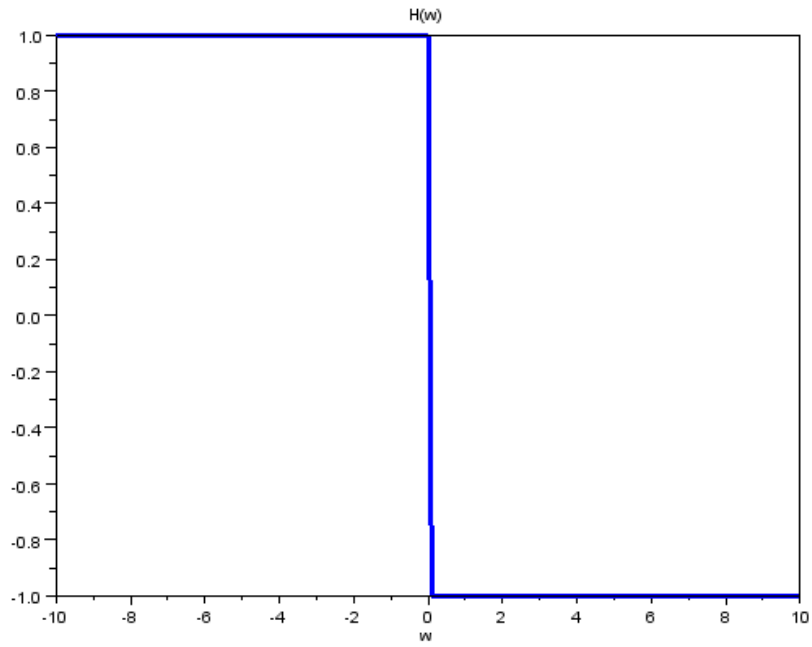


Figure 5.56: impulse response of a phase shifter

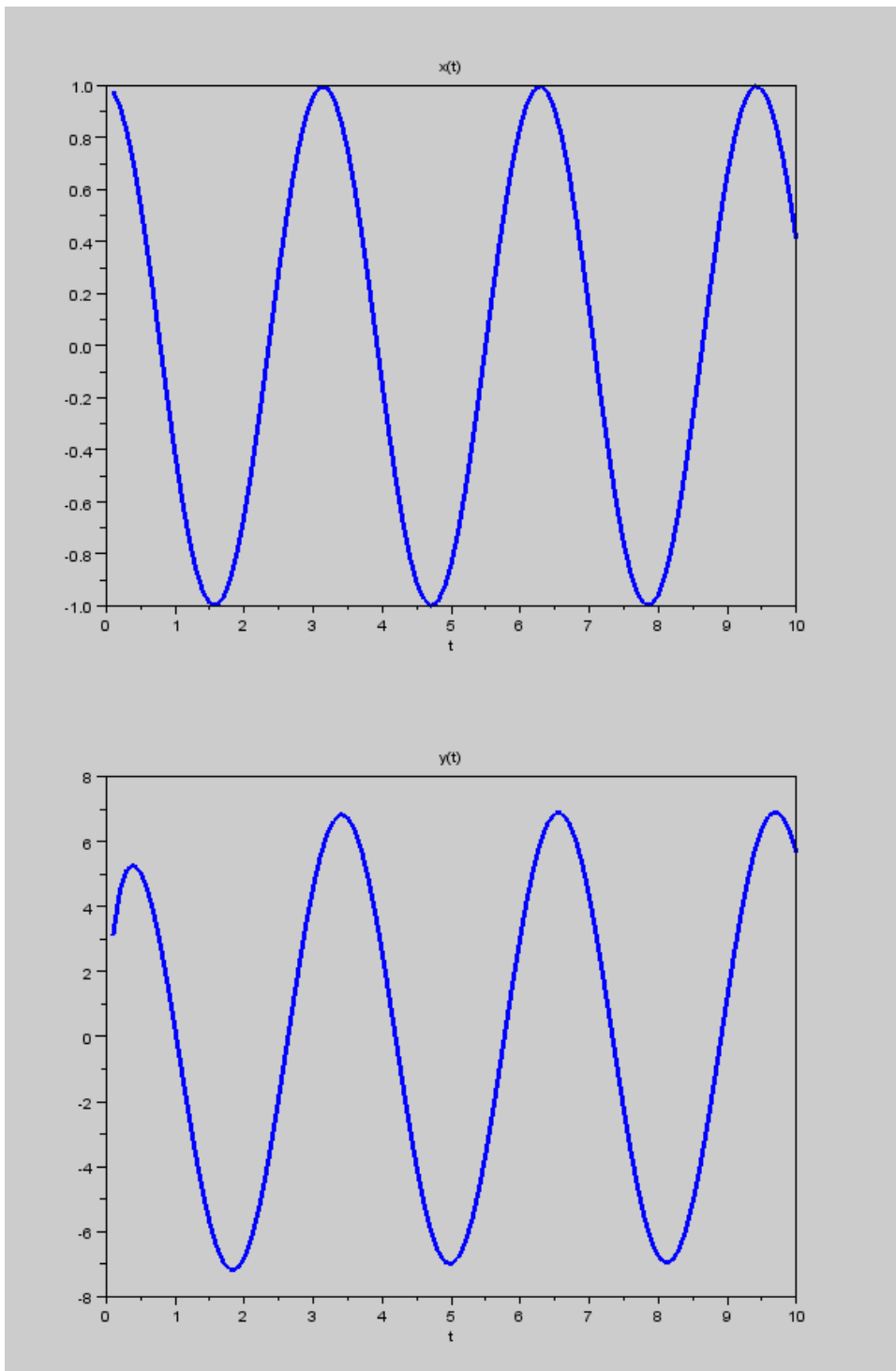


Figure 5.57: impulse response of a phase shifter

```

6 for i=1:length(w)
7     if w(i)>-wc & w(i)<wc then
8         Hw(i)=1;
9     else
10        Hw(i)=0;
11    end
12 end
13 d=gca()
14 plot(w,Hw);
15 poly1=d.children.children;
16 poly1.thickness=3;
17 poly1.foreground=2;
18 xtitle('H(w)', 'w')
19 disp("x(t)=sin(a*t)/(%pi*t)");
20 disp("X(w)= 1 for |w|<a ");
21 disp("      0 elsewhere");
22 a=1;
23 for i=1:length(w)
24     if w(i)>-a & w(i)<a then
25         Xw(i)=1;
26     else
27         Xw(i)=0;
28     end
29 end
30 Yw=Xw.*Hw;
31 figure
32 d=gca()
33 plot(w,Yw);
34 poly1=d.children.children;
35 poly1.thickness=3;
36 poly1.foreground=2;
37 xtitle('Y(w)', 'w')
38 disp("therefore y(t)=x(t) for a<wc");
39 disp("and y(t)=h(t) for a>wc");
40 disp("thus the output suffers distortion when a>wc")
    ;

```

---



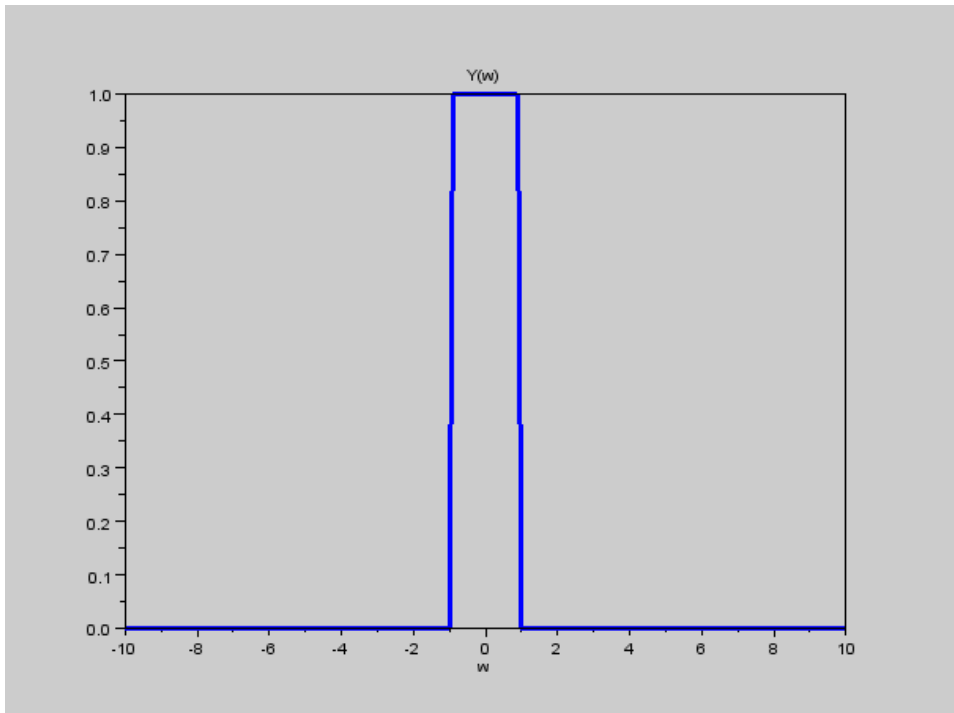


Figure 5.58: output of a ideal LPF

**Scilab code Exa 5.53** output of a ideal LPF

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

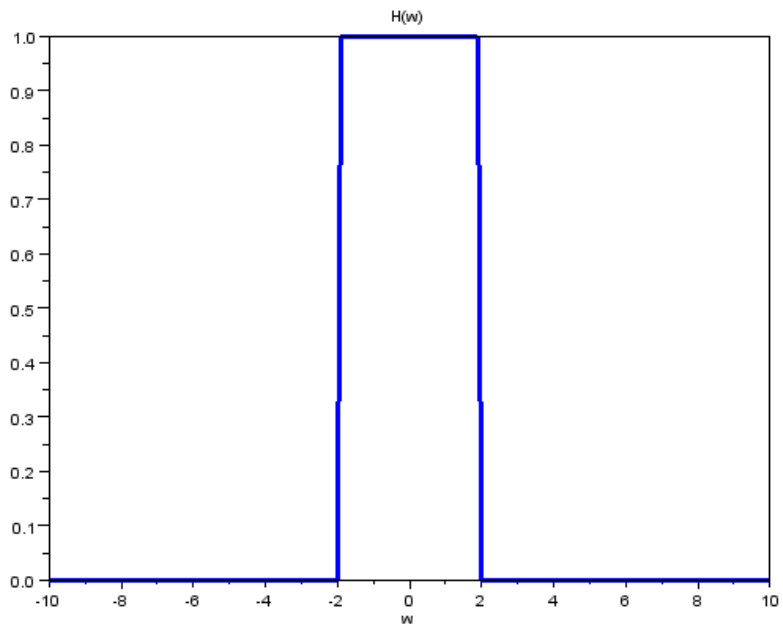


Figure 5.59: output of a ideal LPF

```

7     if w(i)>-wc & w(i)<wc then
8         Hw(i)=1;
9     else
10        Hw(i)=0;
11    end
12 end
13 a=gca();
14 plot(w,Hw);
15 poly1=a.children.children;
16 poly1.thickness=3;
17 poly1.foreground=2;
18 xtitle('H(w)', 'w')
19 T0=1;
20 t=-5.99:0.01:6;
21 t_temp=0.01:0.01:T0;
22 s=length(t)/length(t_temp);
23 x=[];
24 for i=1:s
25     if modulo(i,2)==1 then
26         x=[x zeros(1,length(t_temp))];
27     else
28         x=[x 10*ones(1,length(t_temp))];
29     end
30 end
31 figure
32 a=gca();
33 plot(t,x, 'r');
34 poly1=a.children.children;
35 poly1.thickness=3;
36 poly1.foreground=2;
37 xtitle('x(t)', 't')
38 //fourier series of x(t)
39 w0=%pi;
40 for k=-10:10
41     cc(k+11,:)=exp(-%i*k*w0*t);
42     ck(k+11)=x*cc(k+11,:)/length(t);
43     if abs(ck(k+11))<0.01 then
44         ck(k+11)=0;

```

```

45     else if real(ck(k+11)) < 0.1 then
46         ck(k+11) = %i * imag(ck(k+11));
47     end
48     end
49     if k == 1 then
50         c1 = ck(k+11);
51     end
52     if k == 3 then
53         c3 = ck(k+11);
54     end
55 end
56 yc1 = 2 * abs(c1 / (2 + %i * w0));
57 yc3 = 2 * abs(c3 / (2 + %i * w0 * 3));
58 disp("since frequencies above 4 * %pi are cut off only
      first and third harmonics exists in the output")
      ;
59 y = 5 + yc1 * sin(%pi * t) + yc3 * sin(3 * %pi * t);
60 figure
61 a = gca();
62 plot(t, y);
63 poly1 = a.children.children;
64 poly1.thickness = 3;
65 poly1.foreground = 2;
66 xtitle('y(t)', 't')
67 disp("y = 5 + (20 / %pi) * sin(%pi * t) + 20 / (%pi * 3) * sin(3 * %pi * t
      )");

```

---

Scilab code Exa 5.54 ideal low pass filter

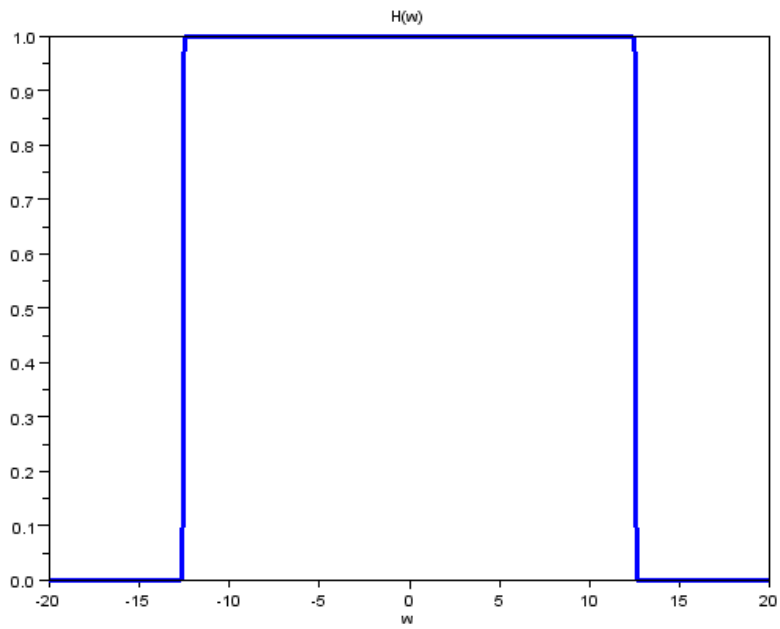


Figure 5.60: output of a ideal LPF

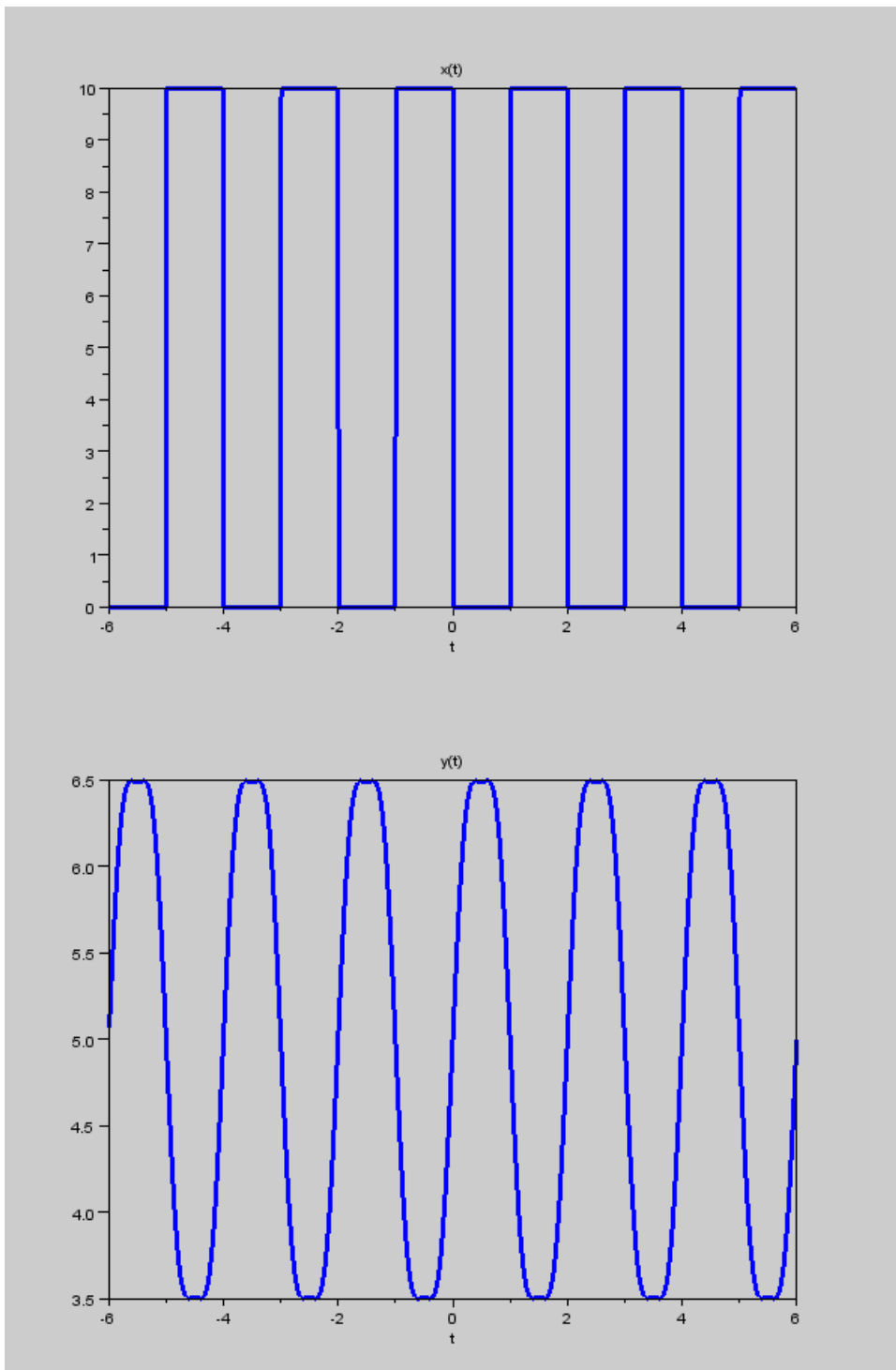


Figure 5.61: output of an ideal LPF

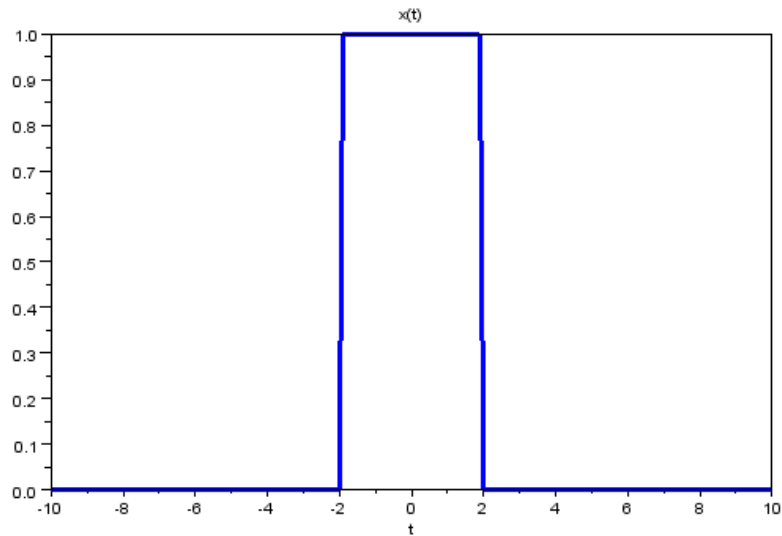


Figure 5.62: ideal low pass filter

```

1 syms t w0 W
2 w=-10:0.1:10;
3 wc=2;
4 for i=1:length(w)
5     if w(i)>-wc & w(i)<wc then
6         Hw(i)=1;
7     else
8         Hw(i)=0;
9     end
10 end
11 a=gca();
12 plot(w,Hw);
13 poly1=a.children.children;
14 poly1.thickness=3;
15 poly1.foreground=2;
16 xtitle('x(t)', 't')
17 disp("we know y(t)=x(t) for w<wc");
18 Xw=ones(1,length(w))./(2+%i*w);

```

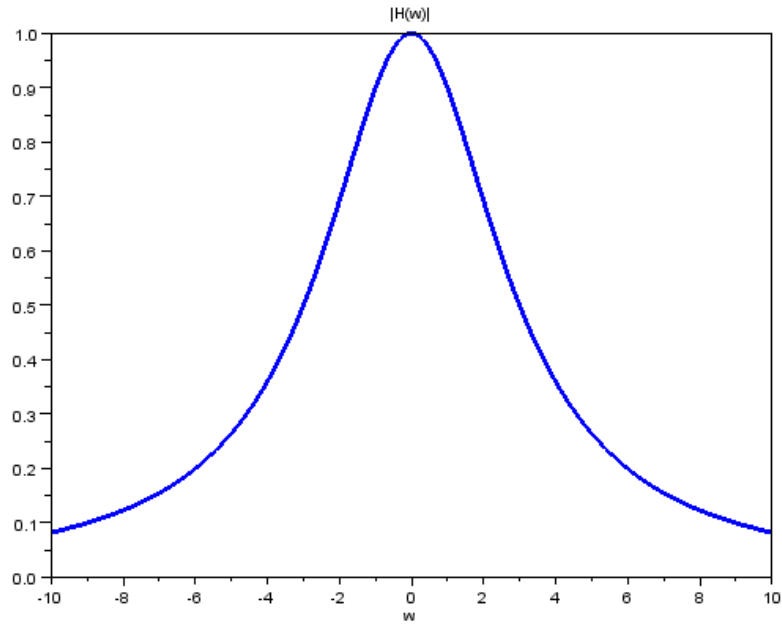


Figure 5.63: equivalent bandwidth

```

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

```

---

#### Scilab code Exa 5.55 equivalent bandwidth

```

1 clear;
2 clc;
3 close;

```



```

4 w=-10:0.1:10;
5 w0=3; //w0=1/(rc)
6 for i=1:length(w)
7     Hw(i)=1/(1+%i*(w(i)/w0));
8     if abs(Hw(i))==1 then
9         wmax=w(i);
10        Hmax=Hw(i)
11    end
12    if abs(Hw(i))==abs(1/(1+%i)) then
13        w3db=w(i);
14    end
15 end
16 disp("rad/s",w3db,"the 3-db bandwidth is w=");
17 weq=(1/Hmax^2)*(%pi)/(2*w0); //integral(|H(w)|^2)=%pi
    *rc/2;
18 disp("rad/s",weq,"the equivalent bandwidth is weq=")
    ;
19 a=gca();
20 plot(w,abs(Hw)^2);
21 poly1=a.children.children;
22 poly1.thickness=3;
23 poly1.foreground=2;
24 xtitle(' |H(w)| ', 'w')

```

---

#### Scilab code Exa 5.58 fourier spectrum

```

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

```

```

 9 poly1=a.children.children;
10 poly1.thickness=3;
11 poly1.foreground=2;
12 xtitle('X(w)', 'w')
13 tmax=10;
14 dw=0.1;
15 t=-tmax:0.1:tmax;
16 x=X*exp(%i*(t'*w2))*dw;
17 subplot(2,1,2)
18 a=gca();
19 plot(t,x);
20 poly1=a.children.children;
21 poly1.thickness=3;
22 poly1.foreground=2;
23 xtitle('x(t)', 't')
24 wM=1.5;
25 //Ts<%pi/wM
26 Ts=1;
27 for i=1:length(t)
28     if modulo(t(i),Ts)==0 then
29         deltaT(i)=1;
30     else
31         deltaT(i)=0;
32     end
33 end
34 figure
35 subplot(2,1,1)
36 a=gca();
37 plot2d3(t,deltaT);
38 poly1=a.children.children;
39 poly1.thickness=3;
40 poly1.foreground=2;
41 xtitle('deltaT(t) Ts<%pi/wM', 't')
42 ws=floor(2*%pi/Ts);
43 for i=1:length(w2)
44     if modulo(w2(i),ws)==0 then
45         deltaW(i)=1;
46     else

```

```

47         deltaW(i)=0;
48     end
49 end
50 subplot(2,1,2)
51 a=gca();
52 plot2d3(w2,deltaW);
53 poly1=a.children.children;
54 poly1.thickness=3;
55 poly1.foreground=2;
56 xtitle('deltaW(w)', 'w')
57 xs=x.*deltaT';
58 figure
59 subplot(2,1,1)
60 a=gca();
61 plot2d3(t,xs)
62 poly1=a.children.children;
63 poly1.thickness=3;
64 poly1.foreground=2;
65 xtitle('xs(t)', 't')
66 Xsw=convol(X,deltaW');
67 subplot(2,1,2)
68 a=gca();
69 plot2d(w2,Xsw(101:301));
70 poly1=a.children.children;
71 poly1.thickness=3;
72 poly1.foreground=2;
73 //Ts>%Pi/wM
74 Ts=3.5;
75 for i=1:length(t)
76     if modulo(t(i),Ts)==0 then
77         deltaT(i)=1;
78     else
79         deltaT(i)=0;
80     end
81 end
82 figure
83 subplot(2,1,1)
84 a=gca();

```

```

85 plot2d3(t,deltaT);
86 poly1=a.children.children;
87 poly1.thickness=3;
88 poly1.foreground=2;
89 xtitle('deltaT(t) Ts<%pi/wM','t')
90 ws=floor(2*%pi/Ts);
91 for i=1:length(w2)
92     if modulo(w2(i),ws)==0 then
93         deltaW(i)=1;
94     else
95         deltaW(i)=0;
96     end
97 end
98 subplot(2,1,2)
99 a=gca();
100 plot2d3(w2,deltaW);
101 poly1=a.children.children;
102 poly1.thickness=3;
103 poly1.foreground=2;
104 xtitle('deltaW(w)','w')
105 xs=x.*deltaT';
106 figure
107 subplot(2,1,1)
108 a=gca();
109 plot2d3(t,xs)
110 poly1=a.children.children;
111 poly1.thickness=3;
112 poly1.foreground=2;
113 xtitle('xs(t)','t')
114 subplot(2,1,2)
115 d=gca();
116 for i=1:length(w2)
117     if modulo(w2(i),ws)==0 then
118         plot(w2+w2(i),X)
119     end
120 end
121 poly1=d.children.children;
122 poly1.thickness=1;

```

```
123 d.data_bounds=[-5 0;5 2];
124
125 poly1.foreground=2;
126 xtitle('Xs(w)', 'w')
```

---

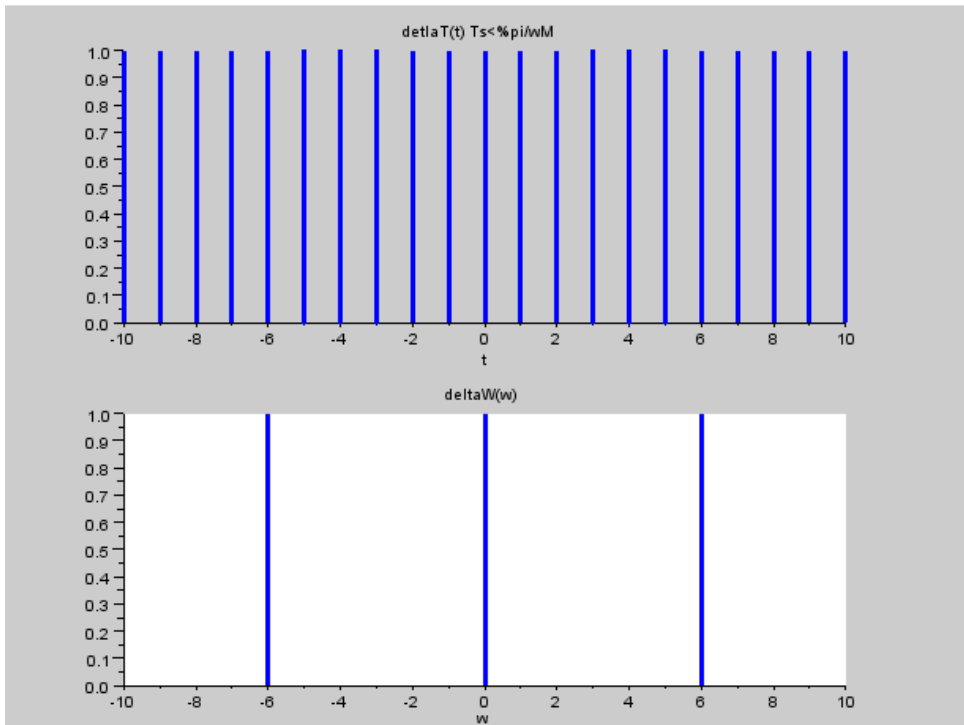
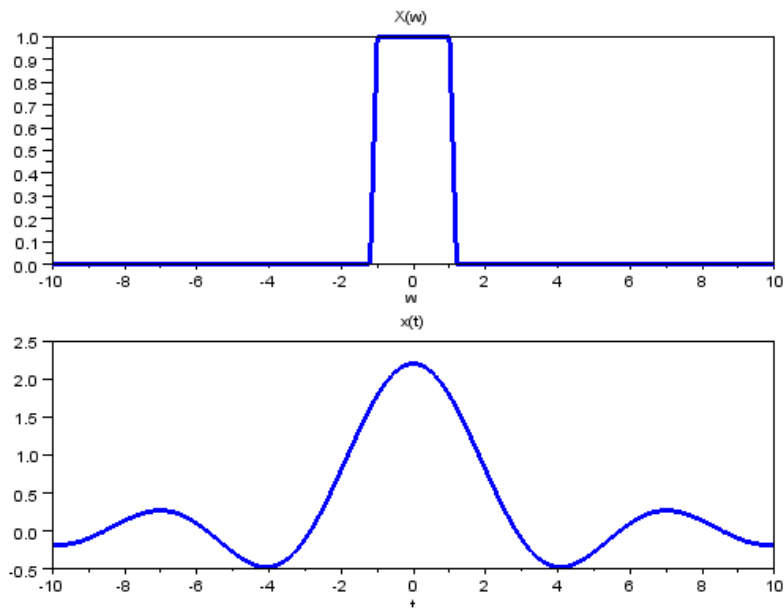
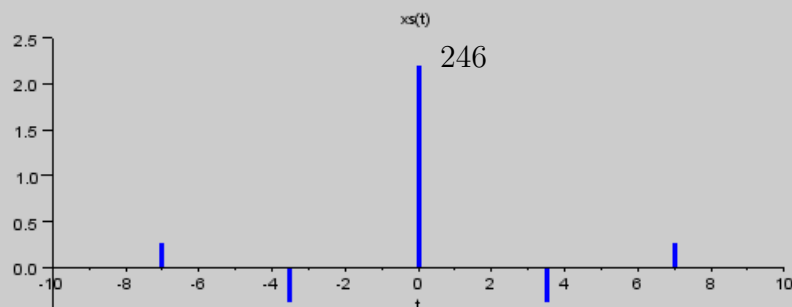
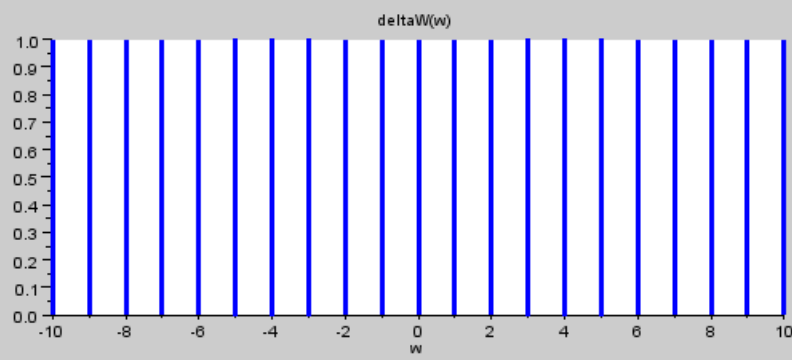
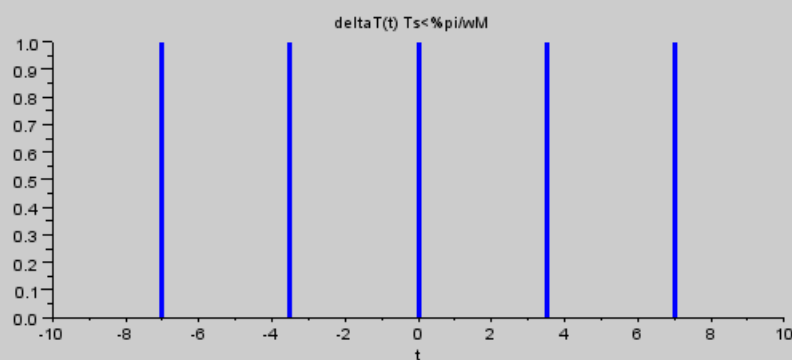
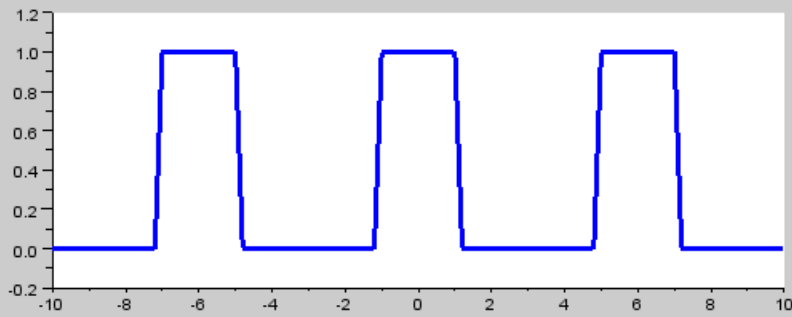
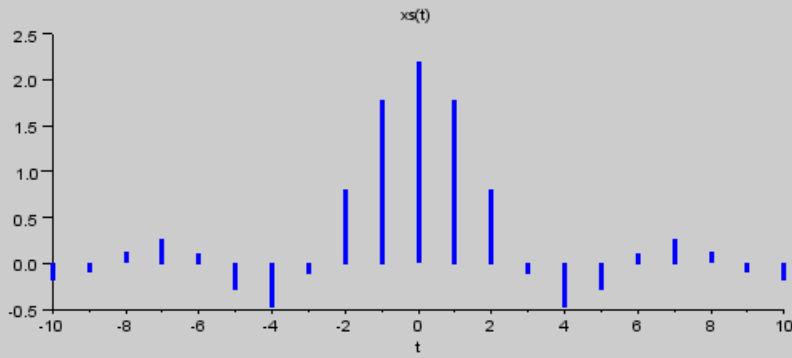


Figure 5.64: fourier spectrum



# Chapter 6

## Fourier analysis of discrete time system and signals

Scilab code Exa 6.3 fourier coefficients

```
1 clear;
2 close;
3 clc;
4 w0=2*pi/4; //pi/2
5 N0=4;
6 n=-8:7;
7 y=0:3;
8 x=[];
9 for i=1:length(n)/4
10     x=[x y];
11 end
12 plot2d3(n,x');
13 plot(n,x','r. ');
14 xtitle('x[n]','n')
15 for k=0:3
16     c(k+1)=0;
17     for n=0:3
```



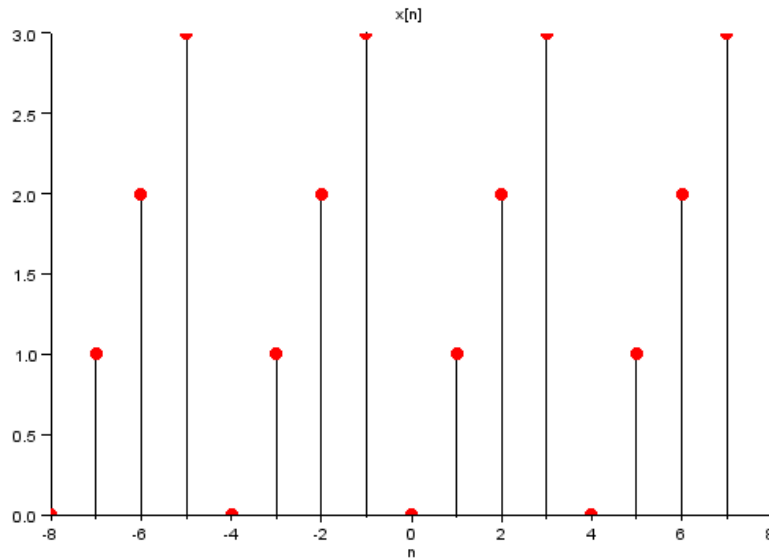


Figure 6.1: fourier coefficients

```

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

```

---

**Scilab code Exa 6.4** fourier coefficients of a periodic sequence

```

1 clear;
2 close;
3 clc;
4 T0=10;
5 w0=%pi/5;
6 y=[ones(1,5) zeros(1,5)];
7 n=-20:19;
8 x=[];
9 for i=1:length(n)/T0

```

```

10     x=[x y];
11 end
12 plot2d3(n,x);
13 plot(n,x,'r.')
14 xtitle('x[n]','n')
15 for k=0:9
16     c(k+1)=0;
17     for n=0:4
18         c(k+1)=c(k+1)+(1/4)*(y(n+1))*(%e)^(%i*w0*k*n
19         );
19     end
20 end
21 c=[c($:-1:2) ;c];
22 k=-9:9;
23 figure
24 plot2d3(k,abs(c));
25 plot(k,abs(c),'r.--')
26 xtitle('|ck|','k')

```

---

### Scilab code Exa 6.5 fourier coefficients

```

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

```

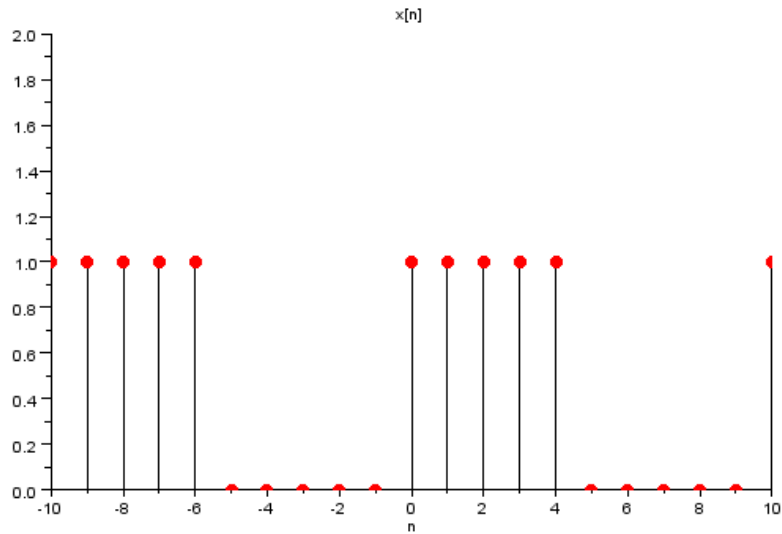


Figure 6.2: fourier coefficients of a periodic sequence

```

11 end
12 plot2d3(n,x);
13 plot(n,x,'r. ');
14 xtitle('x[n]','n')
15 for k=-9:9
16     c(k+10)=0;
17     for n=0:3
18         c(k+10)=c(k+10)+(1/4)*(y(n+1))*(%e)^(%i*w0*k
19             *n);
20     end
21 end
22 k=-9:9;
23 figure
24 plot2d3(k,abs(c));
25 plot(k,c,'r. ')
26 xtitle('x[n]','n')

```

---

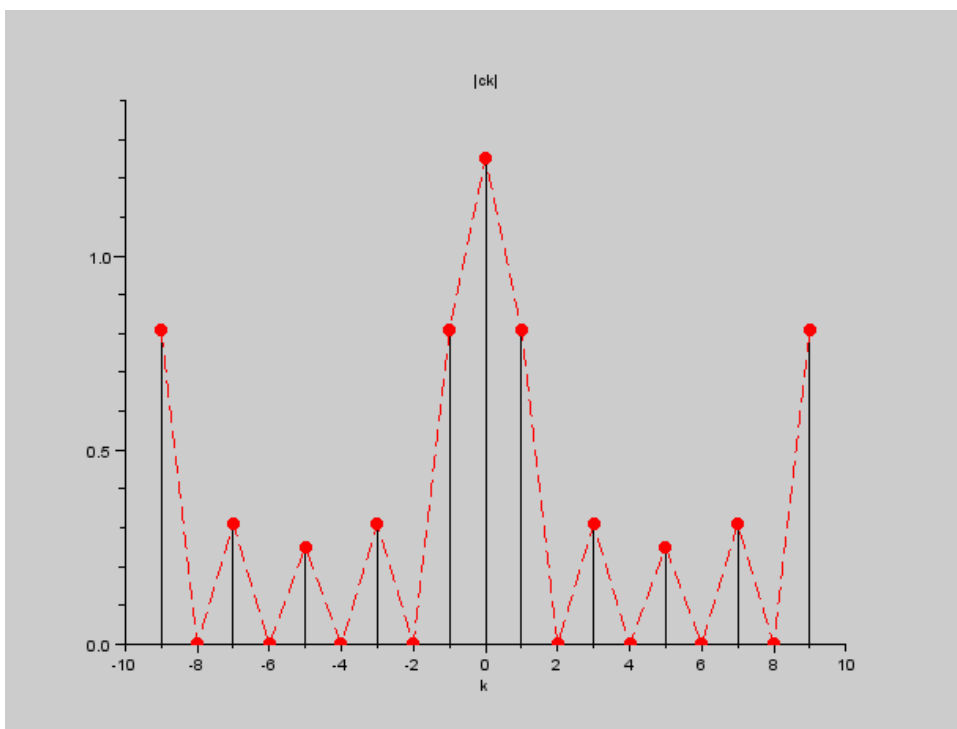


Figure 6.3: fourier coefficients of a periodic sequence

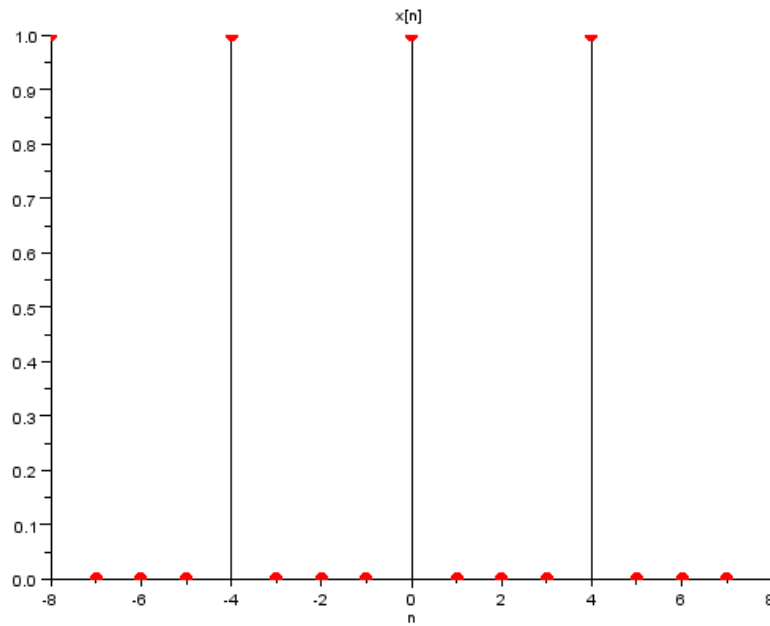


Figure 6.4: fourier coefficients

**Scilab code Exa 6.6** discrete fourier series representation

```

1 clear;
2 close;
3 clc;
4 //cos(%pi*n/4)
5 N0=8;
6 w0=2*%pi/N0;

```

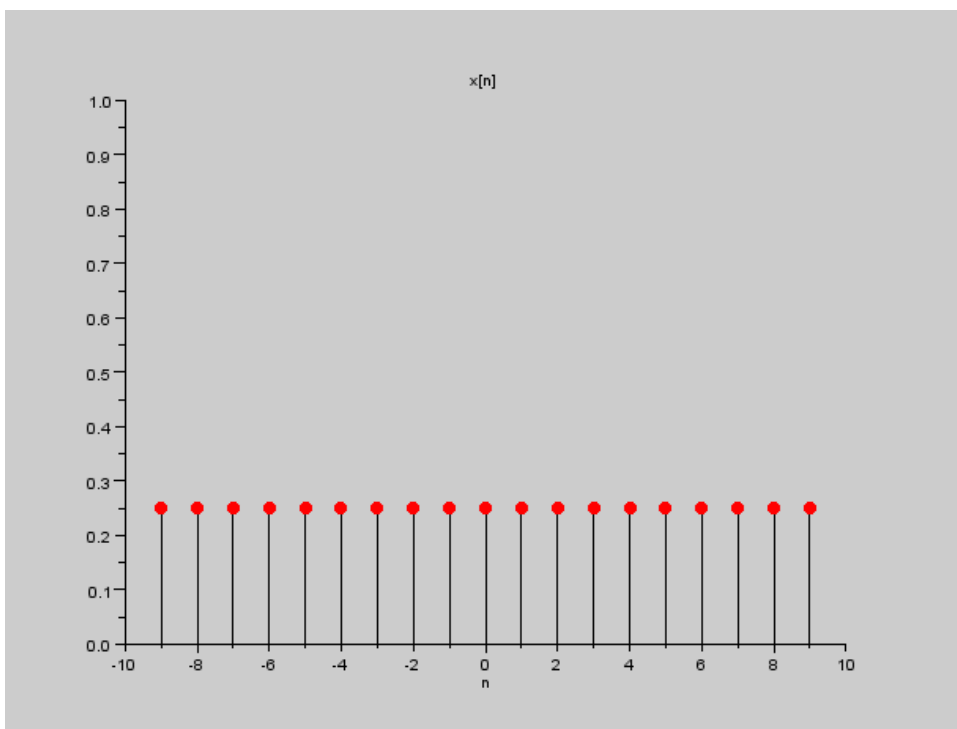


Figure 6.5: fourier coefficients

```

7 n=-8:8;
8 x=cos(%pi*n/4);
9 subplot(2,1,1)
10 xtitle('x[n]', 'n')
11 plot2d3(n,x);
12 plot(n,x, 'r. ');
13 for k=-6:6
14     c(k+7)=0;
15     for n=0:7
16         c(k+7)=c(k+7)+ (1/8)*(x(n+1))*(%e)^(%i*w0*k*n
17             );
18     end
19 end
19 k=-6:6;
20 subplot(2,1,2)
21 xtitle('|ck|', 'k')
22 plot2d3(k,abs(c));
23 plot(k,c, 'r. ')
24 //cos(%pi*n/3)+sin(%pi*n/4)
25 N0=24;
26 w0=2*%pi/N0;
27 n=-24:24;
28 x=cos(%pi*n/3)+sin(%pi*n/4);
29 figure
30 subplot(2,1,1)
31 xtitle('x[n]', 'n')
32 plot2d3(n,x);
33 plot(n,x, 'r.-- ');
34 for k=-24:24
35     c(k+25)=0;
36     for n=0:23
37         c(k+25)=c(k+25)+ (1/N0)*(x(n+1))*(%e)^(%i*w0*
38             k*n);
39     end
40 end
40 k=-24:24;
41 subplot(2,1,2)
42 xtitle('|ck|', 'k')

```

```

43 plot2d3(k,abs(c));
44 plot(k,c,'r.')
45 // [cos(%pi*n/8)]^2
46 N0=8;
47 w0=2*pi/N0;
48 n=-8:8;
49 x=[cos(%pi*n/8)]^2;
50 clear c;
51 figure
52 subplot(2,1,1)
53 xtitle('x[n]','n')
54 plot2d3(n,x);
55 plot(n,x,'r.--');
56 for k=-6:6
57     c(k+7)=0;
58     for n=0:7
59         c(k+7)=c(k+7)+(1/N0)*(x(n+1))*e^(%i*w0*k*
60             n);
61     end
62 end
63 k=-6:6;
64 subplot(2,1,2)
65 xtitle('|ck|','k')
66 plot2d3(k,abs(c));
67 plot(k,c,'r.')
68 disp("fourier series is x[n]=1/N0*sum(c(k)*e^%i*w0*n
69     *k)");

```

---

Scilab code Exa 6.11 fourier transform



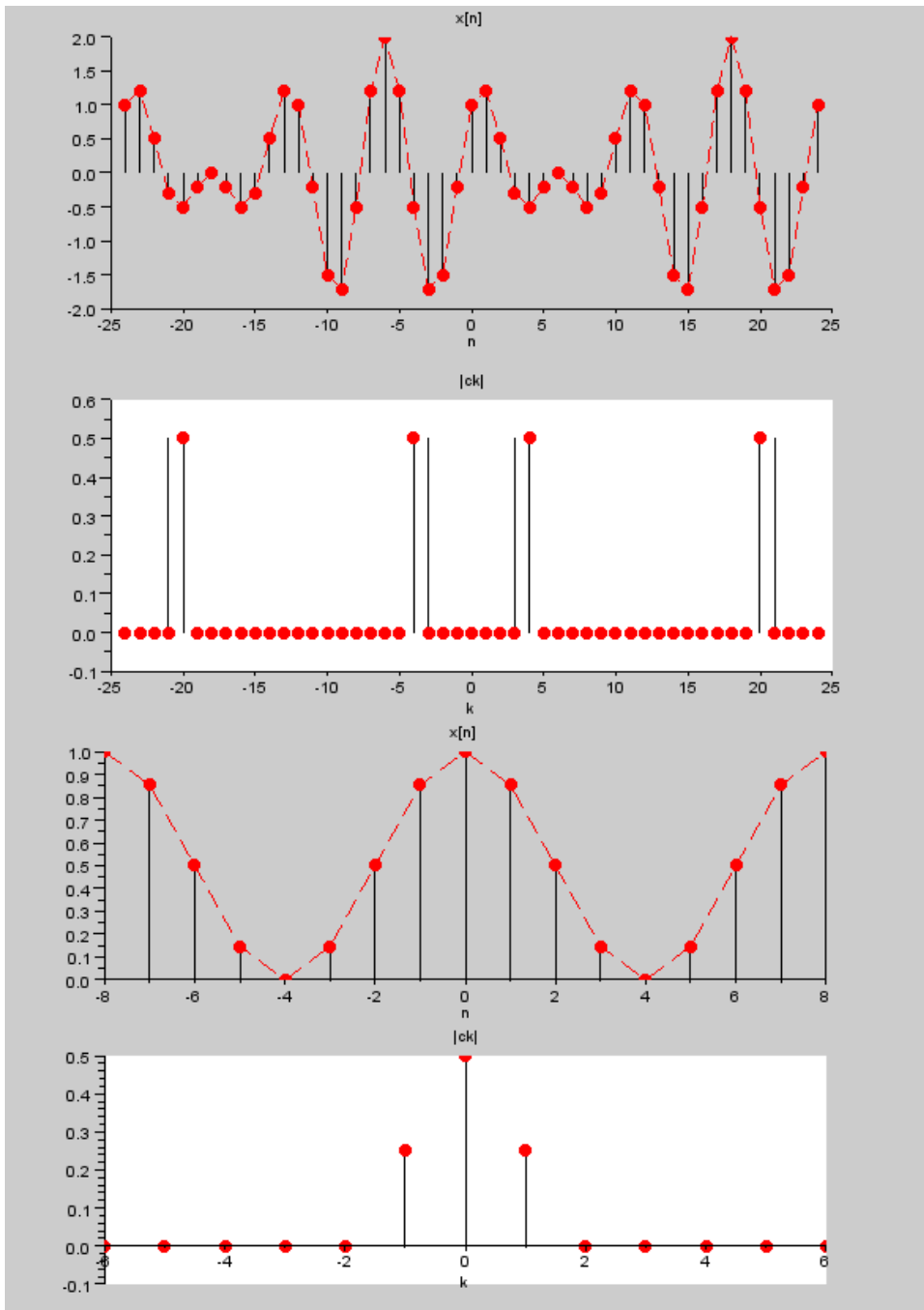


Figure 6.6: discrete fourier series representation

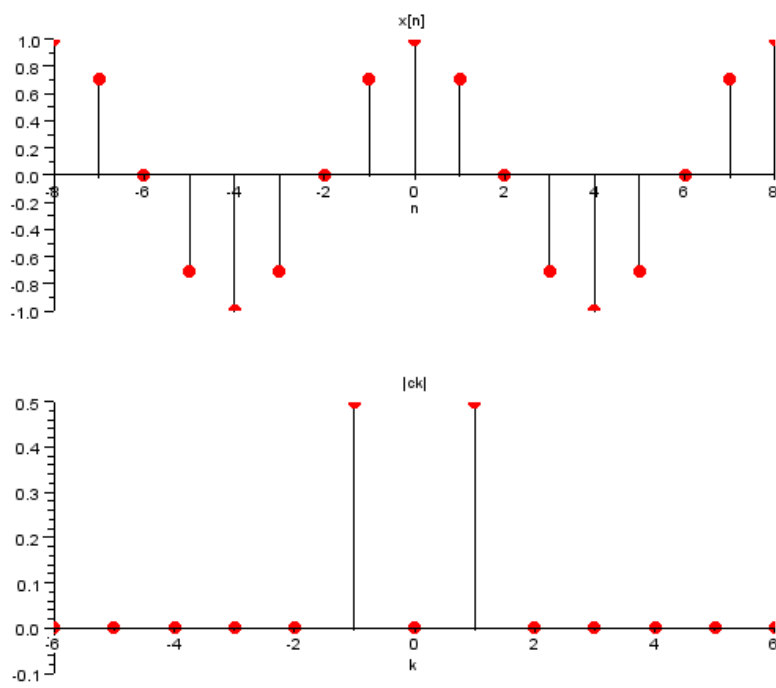


Figure 6.7: discrete fourier series representation

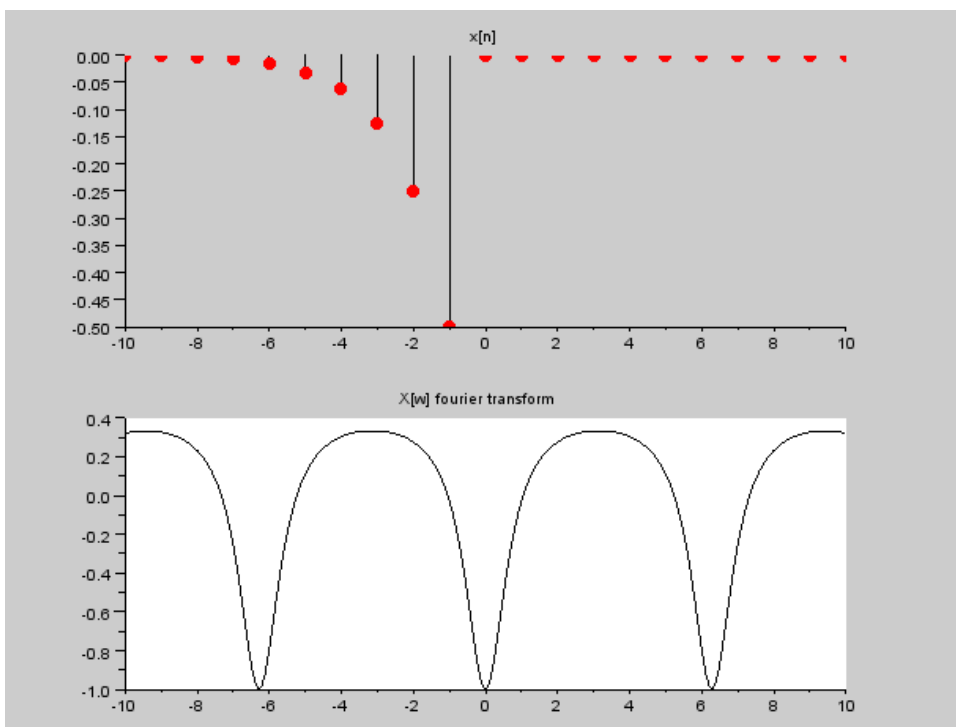


Figure 6.8: fourier transform

```

1 clear;
2 close;
3 clc;
4 n=-10:10;
5 a=2;
6 for i=1:length(n)
7     if n(i)<=-1 then
8         x(i)=-2^n(i);
9     else
10        x(i)=0;
11    end
12 end
13 x=x';
14 figure
15 subplot(2,1,1)
16 plot2d3(n,x);
17 title("x[n]")
18 plot(n,x,'r.')
19 w=-10:0.1:10;
20 Xw=x*exp(-%i*n'*w);
21 subplot(2,1,2)
22 plot2d(w,Xw);
23 title("X[w]  fourier transform")

```

---

Scilab code Exa 6.12 fourier transform of a rectangular pulse

```

1 clear;
2 close;
3 clc;
4 n=-10:10;
5 N=5;
6 for i=1:length(n)
7     if n(i)>=0 & n(i)<=N then

```

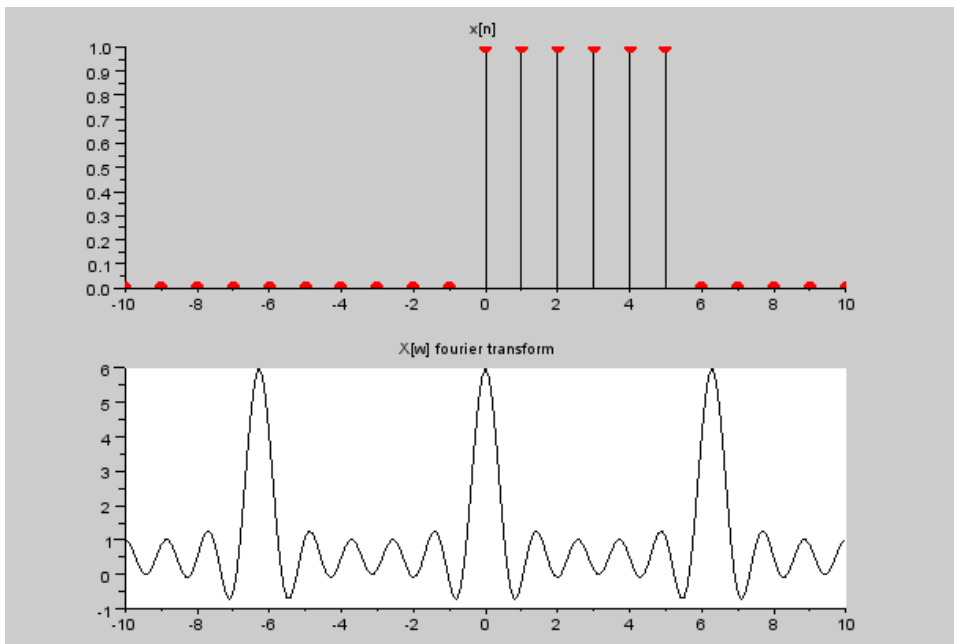


Figure 6.9: fourier transform of a rectangular pulse

```

8         x(i)=1;
9     else
10        x(i)=0;
11    end
12 end
13 x=x';
14 figure
15 subplot(2,1,1)
16 plot2d3(n,x);
17 title("x[n]")
18 plot(n,x,'r.')
19 w=-10:0.1:10;
20 Xw=x*exp(-%i*n'*w);
21 subplot(2,1,2)
22 plot2d(w,Xw);
23 title("X[w] fourier transform")

```

---

### Scilab code Exa 6.14 fourier transform

```
1 clear;
2 close;
3 clc;
4 n=-10:10;
5 N=4;
6 for i=1:length(n)
7     if n(i)>=-N & n(i)<=N then
8         x(i)=1;
9     else
10        x(i)=0;
11    end
12 end
13 x=x';
14 figure
15 subplot(2,1,1)
16 plot2d3(n,x);
17 title("x[n]")
18 plot(n,x,'r.')
19 w=-10:0.1:10;
20 Xw=x*exp(-%i*n'*w);
21 subplot(2,1,2)
22 plot2d(w,Xw);
23 title("X[w] fourier transform")
24 N=8;
25 for i=1:length(n)
26     if n(i)>=-N & n(i)<=N then
27         x(i)=1;
28     else
29         x(i)=0;
30     end
31 end
32 x=x';
```

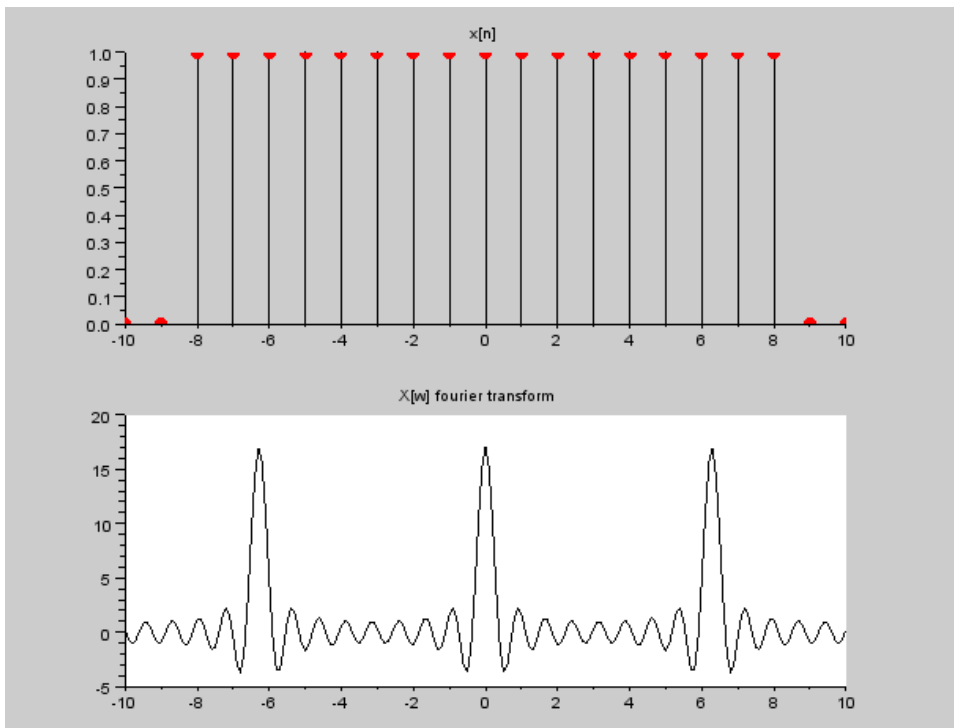


Figure 6.10: fourier transform

```

33 figure
34 subplot(2,1,1)
35 plot2d3(n,x);
36 title("x[n]")
37 plot(n,x,'r.')
38 w=-10:0.1:10;
39 Xw=x'*exp(-%i*n'*w);
40 subplot(2,1,2)
41 plot2d(w,Xw);
42 title("X[w] fourier transform")

```

---

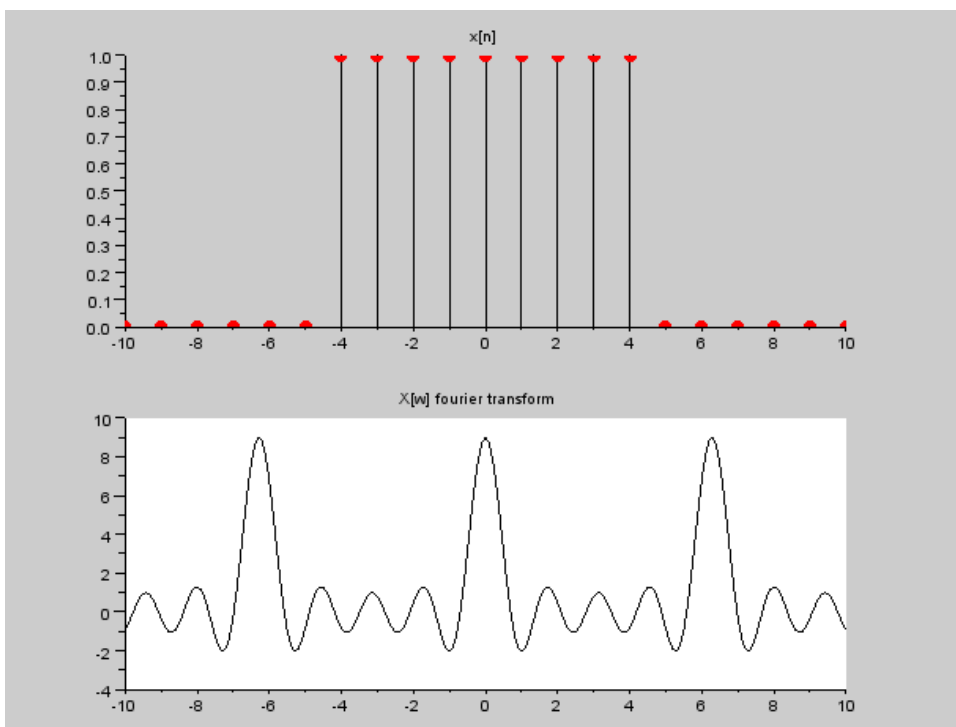


Figure 6.11: fourier transform



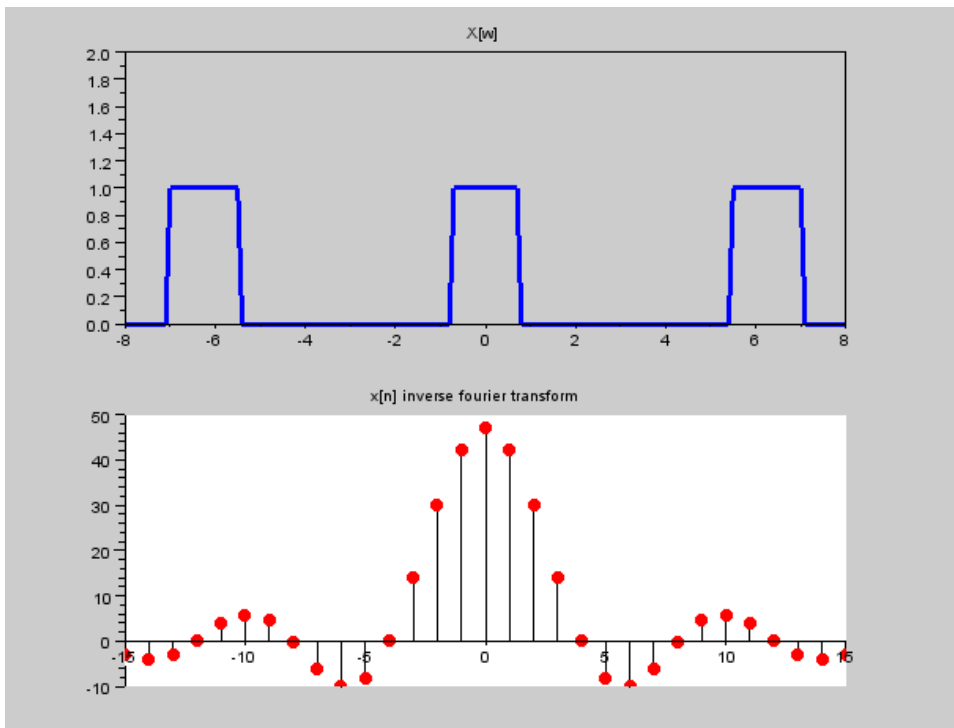


Figure 6.12: inverse fourier transform of a rectangular pulse

**Scilab code Exa 6.15** inverse fourier transform of a rectangular pulse

```

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

```

```

9 elseif (w(i)>=-W & w(i)<=W)
10     X(i)=1;
11 elseif (w(i)>=2*%pi-W & w(i)<=2*%pi+W)
12     X(i)=1;
13 else
14     X(i)=0;
15     end
16 end
17 figure
18 subplot(2,1,1)
19 plot(w,X);
20 title("X[w]");
21 n=-15:15;
22 x=X'*exp(-%i*w'*n);
23 subplot(2,1,2)
24 plot2d3(n,x);
25 plot(n,x,'r. ')
26 title("x[n] inverse fourier transform")

```

---

**Scilab code Exa 6.17** inverse fourier transform of a impulse signal

```

1 clear;
2 close;
3 clc;
4 w=-5:0.01:5;
5 W0=2;
6 for i=1:length(w)
7     if w(i)==W0 then
8         delta(i)=2*%pi;
9     else
10        delta(i)=0;
11    end
12 end

```

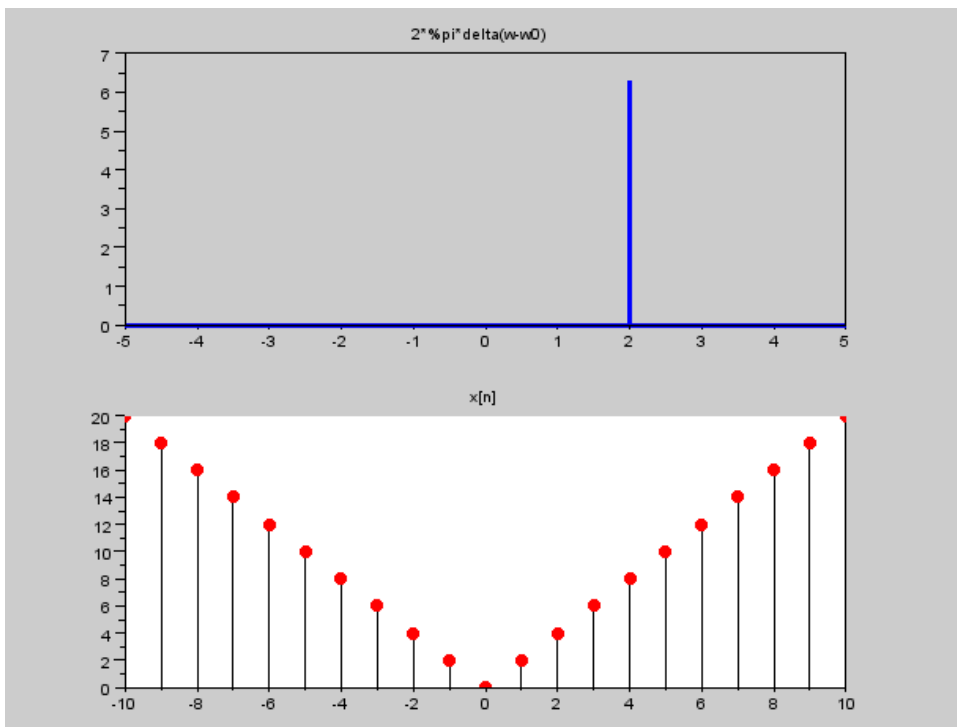


Figure 6.13: inverse fourier transform of a impulse signal

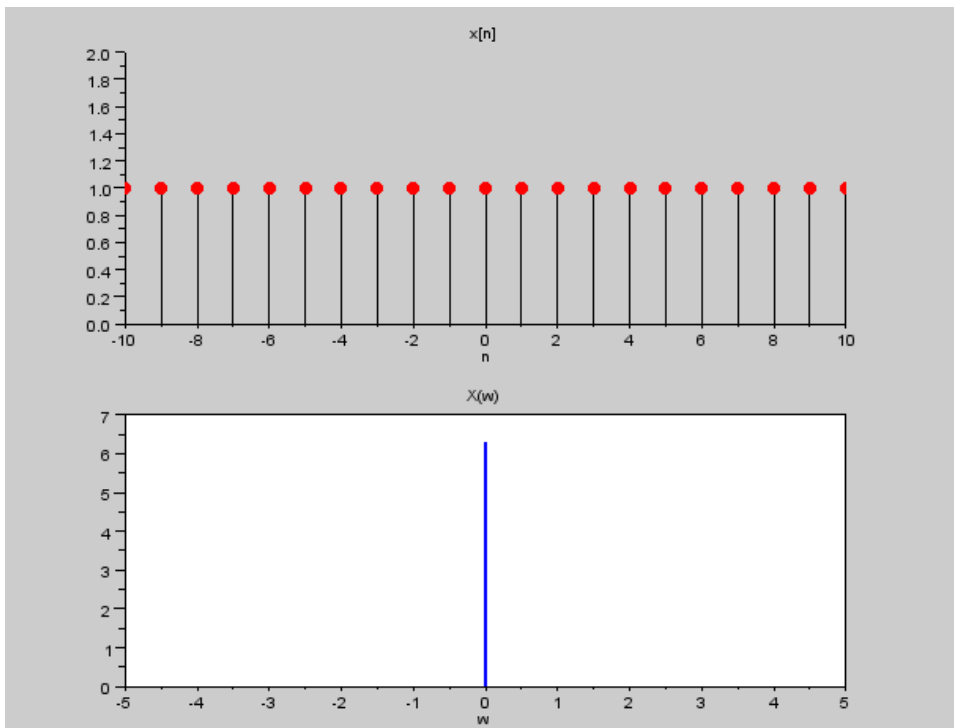


Figure 6.14: fourier transform of constant signal

```

13 figure
14 subplot(2,1,1)
15 plot(w,delta);
16 title('2*%pi*delta(w-w0)')
17 //by shifting property
18 n=-10:10;
19 x=%e^-%i*W0*n;
20 subplot(2,1,2)
21 plot2d3(n,abs(x))
22 plot(n,abs(x),'r.')
23 title('x[n]')

```

---

**Scilab code Exa 6.18** fourier transform of constant signal

```
1 clear;
2 close;
3 clc;
4 n=-10:10;
5 x=ones(1,length(n));
6 figure
7 subplot(2,1,1)
8 plot2d3(n,x)
9 plot(n,x,'r.')
10 xtitle('x[n]','n')
11 w=-5:0.01:5;
12 for i=1:length(w)
13     if w(i)==0 then
14         delta(i)=1;
15     else
16         delta(i)=0;
17     end
18 end
19 Xw=2*%pi*delta';
20 subplot(2,1,2)
21 plot(w,Xw);
22 xtitle('X(w)','w')
```

---

**Scilab code Exa 6.19** fourier transform of a sinusoidal sequence

```
1 clear;
2 close;
3 clc;
4 n=-10:10;
5 W0=1;
6 x=cos(W0*n);
```

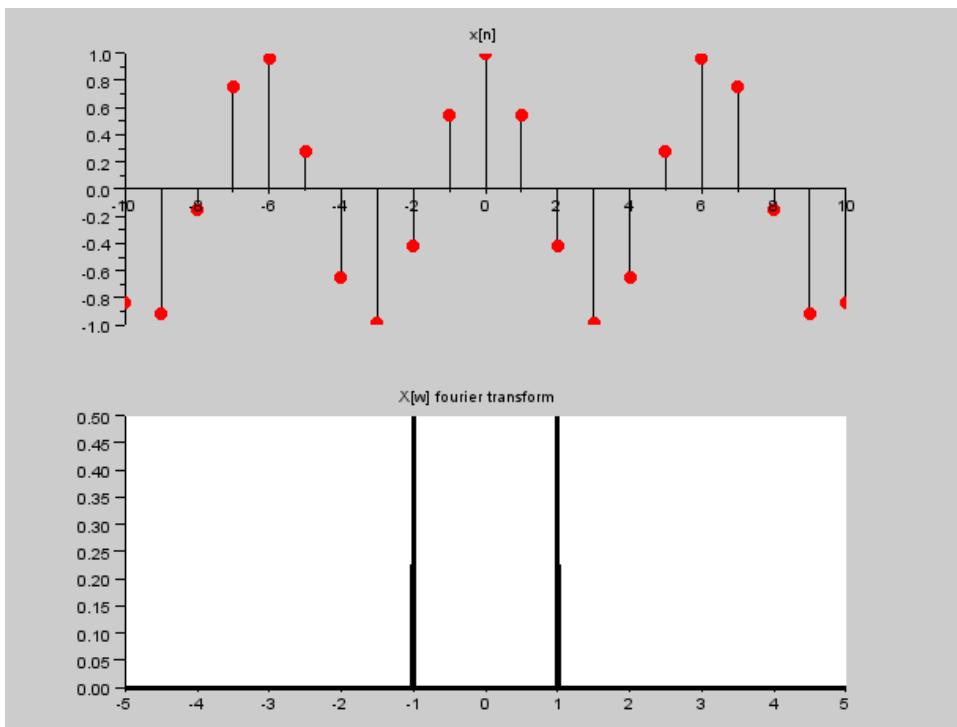


Figure 6.15: fourier transform of a sinusoidal sequence

```

7 figure
8 subplot(2,1,1)
9 plot2d3(n,x);
10 title("x[n]")
11 plot(n,x,'r.')
12 // cos(W0*n)=1/2*(e^{-%j*W0*n}+e^{%j*W0*n})
13 w=-5:0.01:5;
14 for i=1:length(w)
15     if w(i)==W0 then
16         deltaW0(i)=1;
17     else
18         deltaW0(i)=0;
19     end
20 end
21 delta_W0=deltaW0($:-1:1);
22 //by frequency shifting property
23 Xw=1/2*[deltaW0+delta_W0];
24 subplot(2,1,2)
25 plot2d(w,Xw);
26 title("X[w] fourier transform")

```

---

#### Scilab code Exa 6.22 fourier transform

```

1 clear;
2 close;
3 clc;
4 n=-10:10;
5 N=2;
6 for i=1:length(n)
7     if n(i)>=-N & n(i)<=N then
8         x(i)=1;
9     else
10        x(i)=0;
11    end
12 end

```

```

13 x=x';
14 figure
15 subplot(2,1,1)
16 plot2d3(n,x);
17 title("x[n]")
18 plot(n,x,'r.')
19 w=-10:0.1:10;
20 Xw=x*exp(-%i*n'*w);
21 subplot(2,1,2)
22 plot2d(w,Xw);
23 title("X[w] fourier transform")
24 //time scaled sequence x2[n]
25 n2=-20:2:20;
26 figure
27 subplot(2,1,1)
28 plot2d3(n2,x);
29 title("x2[n]")
30 plot(n2,x,'r.')
31 w2=-5:0.05:5;
32 subplot(2,1,2)
33 plot2d(w2,Xw);
34 title("X2[w] fourier transform")
35 //time scaled sequence x3[n]
36 n3=-30:3:30;
37 figure
38 subplot(2,1,1)
39 plot2d3(n3,x);
40 title("x3[n]")
41 plot(n3,x,'r.')
42 w3=w/3;
43 subplot(2,1,2)
44 plot2d(w3,Xw);
45 title("X3[w] fourier transform")

```

---



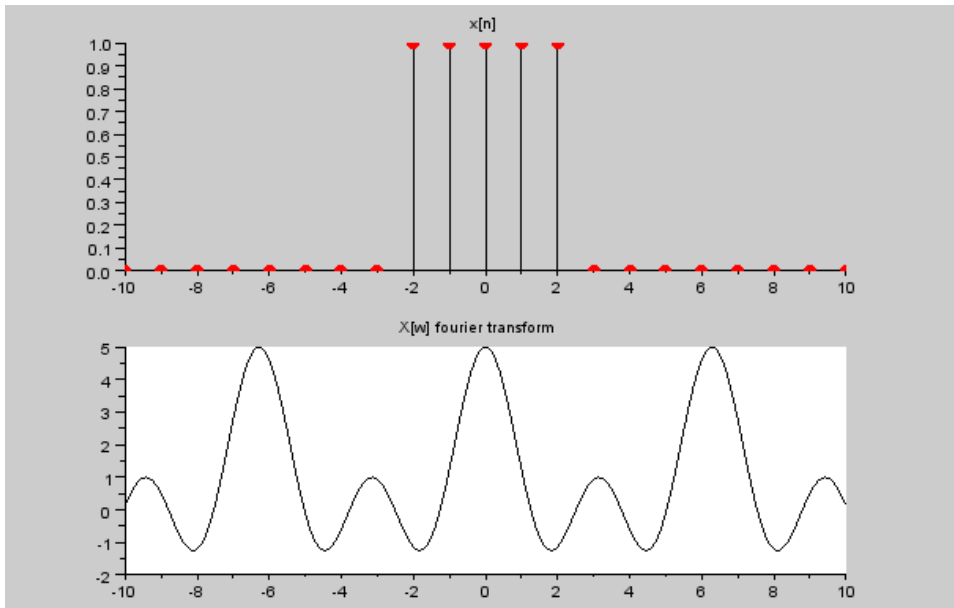


Figure 6.16: fourier transform

**Scilab code Exa 6.25** inverse fourier transform using convolution

```

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

```

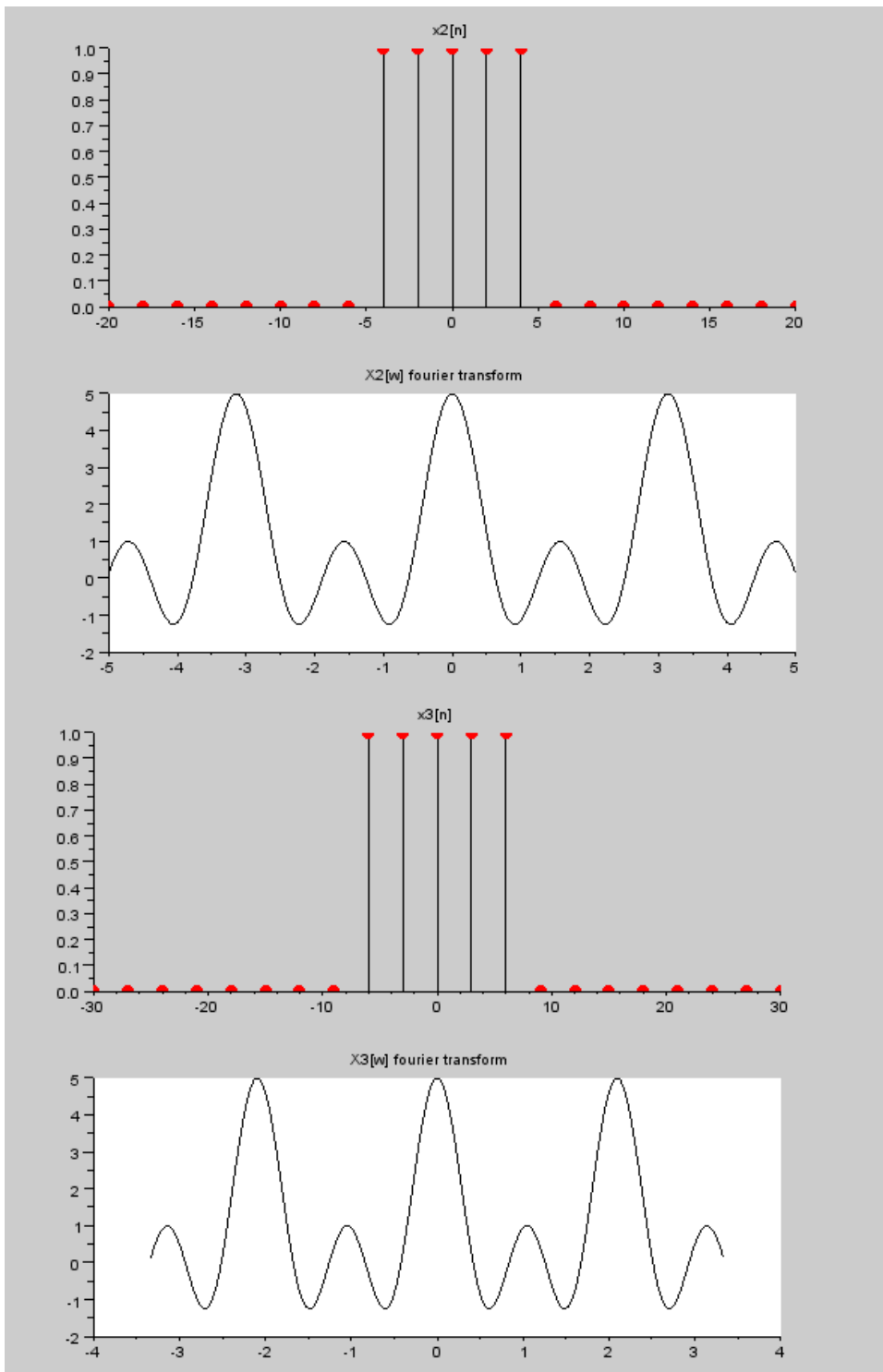


Figure 6.17: <sup>273</sup>fourier transform

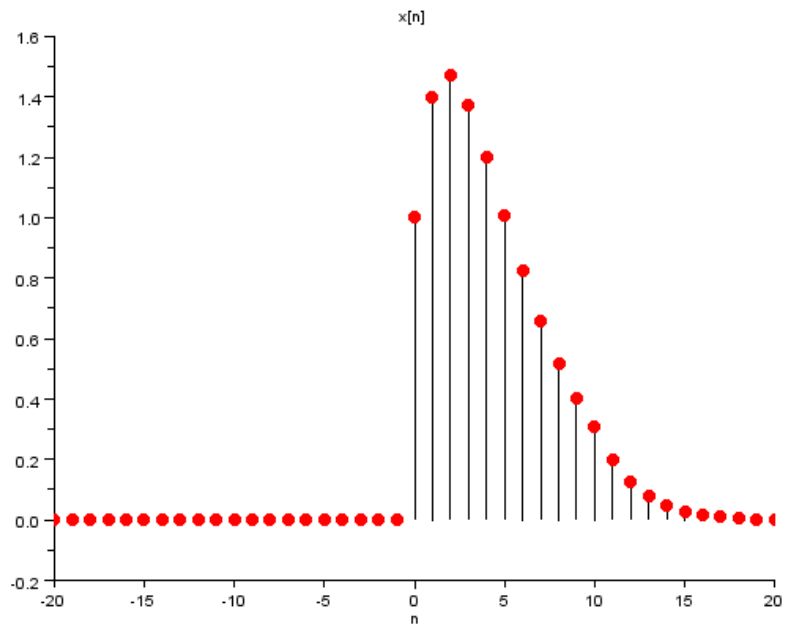


Figure 6.18: inverse fourier transform using convolution

```

11 n1=-20:20;
12 plot2d3(n1,x);
13 plot(n1,x,'r. ');
14 xtitle('x[n]','n')

```

---

### Scilab code Exa 6.28 frequency response

```

1 clear;
2 close;
3 clc;
4 n=-10:10;
5 x=[zeros(1,find(n==0)-1) ones(1,length(n)-find(n==0)
    +1)];
6 plot2d3(n,x)
7 plot(n,x,'r. ')
8 xtitle('x[n]','n')
9 figure
10 w=-2:2;
11 Xw=x*exp(-%i*n'*w);
12 subplot(2,1,1)
13 plot2d(w,real(Xw));
14 xtitle('U[w] fourier transform','w')
15 for i=1:length(w)
16     if w(i)==0 then
17         delta(i)=1;
18     else
19         delta(i)=0;
20     end
21 end
22 Xwproof=%pi*delta'+ones(1,length(w))./(1-%e^(-%i*w))
    ;
23 subplot(2,1,2)
24 plot(w,Xwproof)
25 xtitle('%pi*delta+1/(1-e^i*w)','w')
26 disp("it can be seen that both the figures are

```

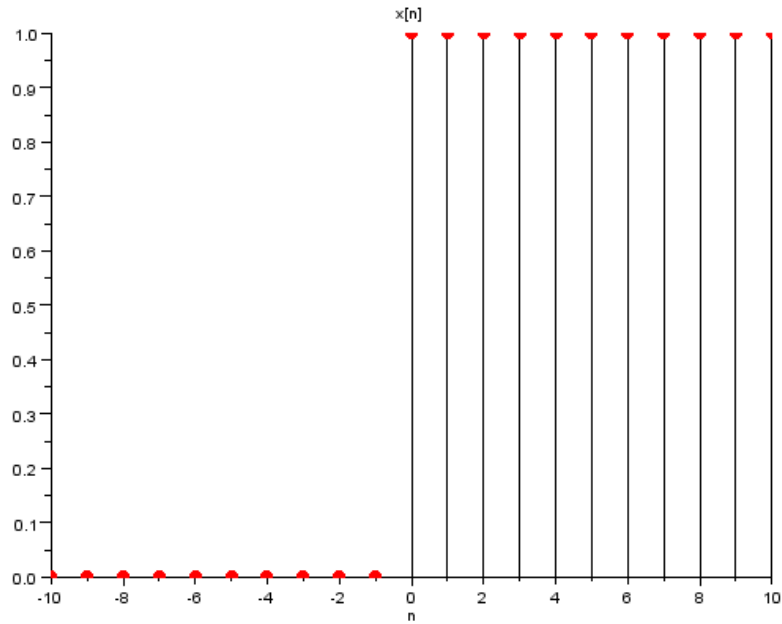


Figure 6.19: frequency response

approximately same hence  
 $+1/(1-e^{i\omega})$ ;

$$X(\omega) = \sum_{n=0}^{\infty} e^{-i\omega n}$$

**Scilab code Exa 6.31** frequency response

```
1 clear;
2 clc;
```

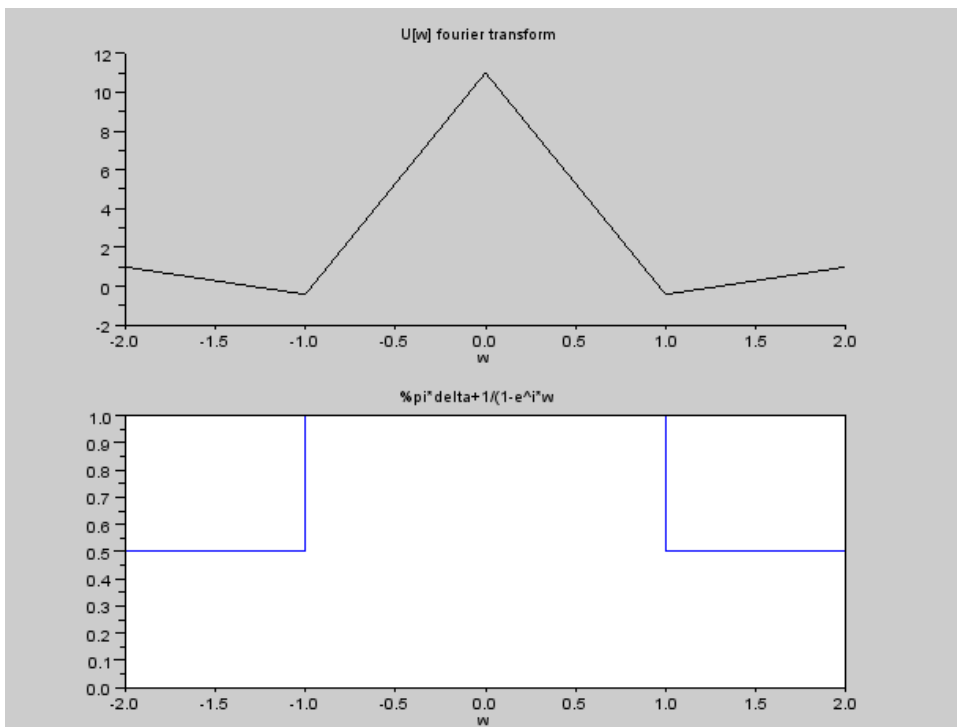


Figure 6.20: frequency response

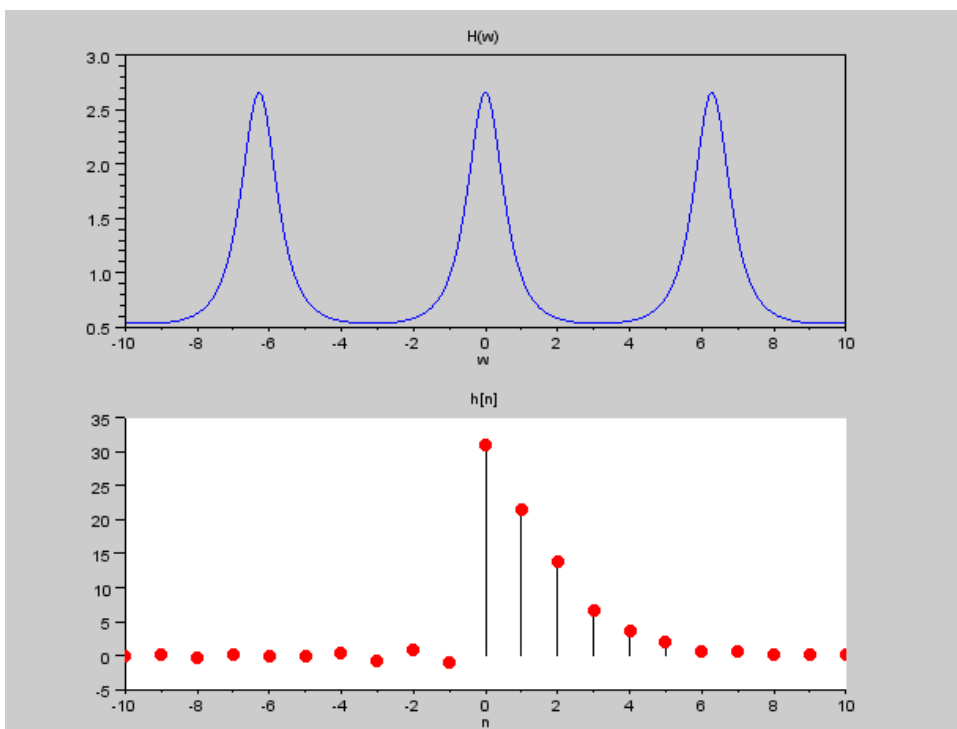


Figure 6.21: frequency response

```

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

```

---

### Scilab code Exa 6.32 frequency response

```

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

```



```

14 h=(1/(2*%pi))*Hw*exp(%i*w'*n);
15 subplot(2,1,2)
16 plot2d3(n,h)
17 xtitle('h[n]', 'n')
18 plot(n,h, 'r. ')
19 x=cos(%pi*n/2);
20 figure
21 subplot(2,1,1)
22 plot2d3(n,x);
23 xtitle('x[n]', 'n')
24 plot(n,x, 'r. ')
25 y=convol(x,h);
26 subplot(2,1,2)
27 plot2d3(n,y(11:31))
28 xtitle('y[n]', 'n')
29 plot(n,y(11:31), 'r. ')

```

---

Scilab code Exa 6.33 output response

```

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

```

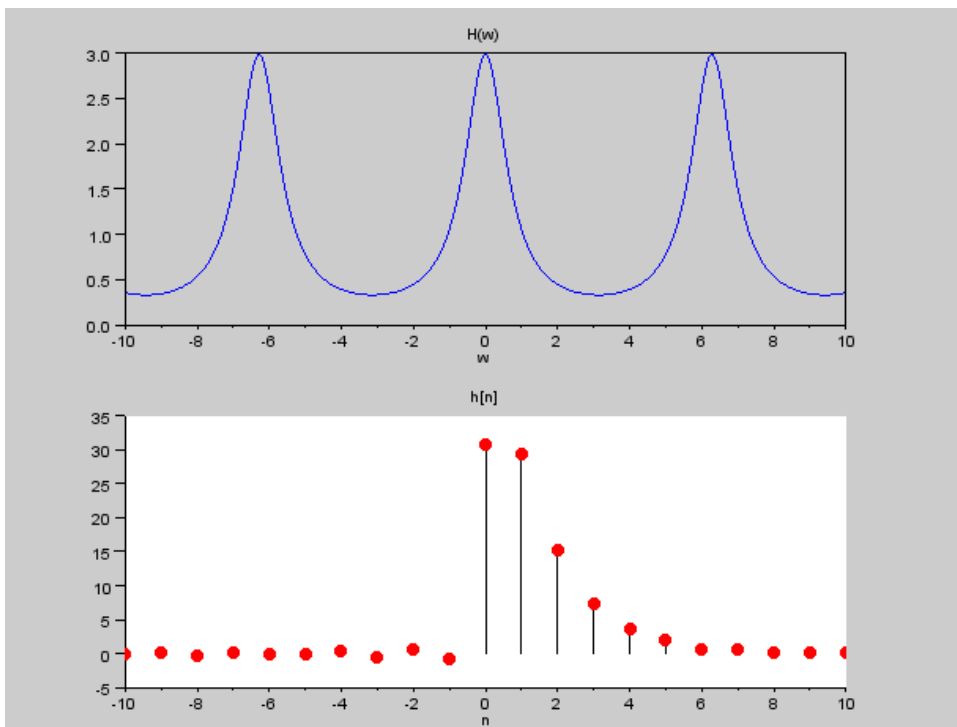


Figure 6.22: frequency response

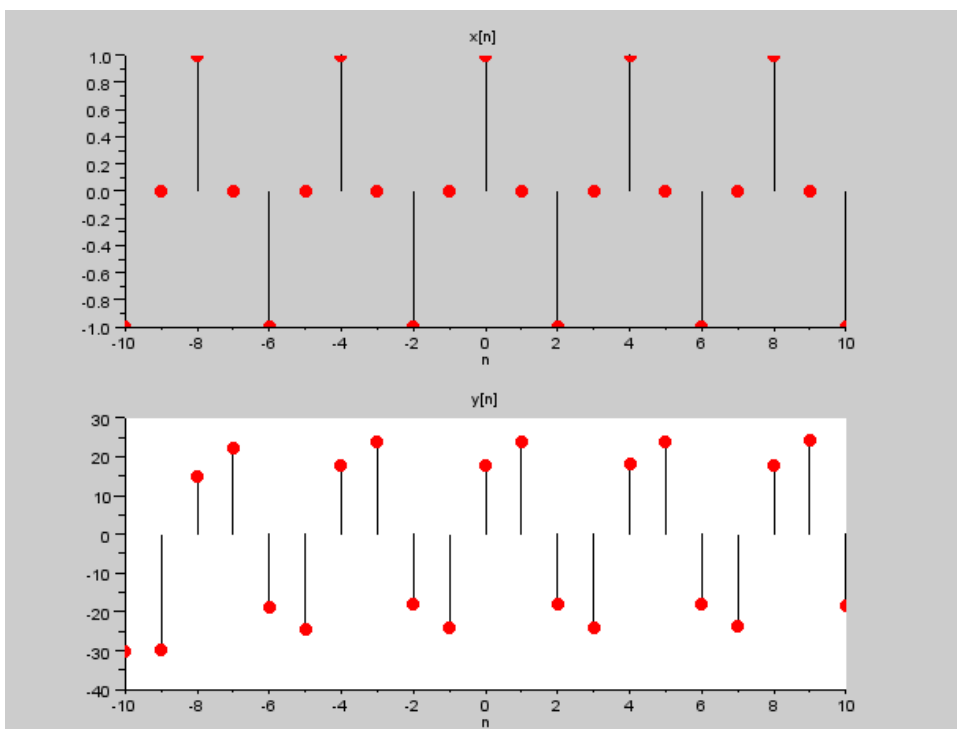


Figure 6.23: frequency response

```

10 plot(n,h, 'r. ');
11 xtitle('h[n]', 'n')
12 w=-4:0.01:4;
13 Hw=h*exp(-%i*n'*w);
14 subplot(2,1,2)
15 plot2d(w,Hw);
16 title("H[w] fourier transform")
17 T0=5;
18 w0=%pi*2/5;
19 z=[ones(1,3) zeros(1,2)];
20 x=[];
21 for i=1:length(n)/T0
22     x=[x z];
23 end
24 figure
25 subplot(2,1,1)
26 plot2d3(n,x);
27 plot(n,x, 'r. ')
28 xtitle('x[n]', 'n')
29 c0= (1/5)*sum(z);
30 y=ones(1, length(n))*c0;
31 subplot(2,1,2)
32 plot2d3(n,y);
33 plot(n,y, 'r.-- ')
34 xtitle('y[n]', 'n')

```

---

Scilab code Exa 6.34 magnitude and phase response

```

1 clear;
2 clc;
3 close;

```

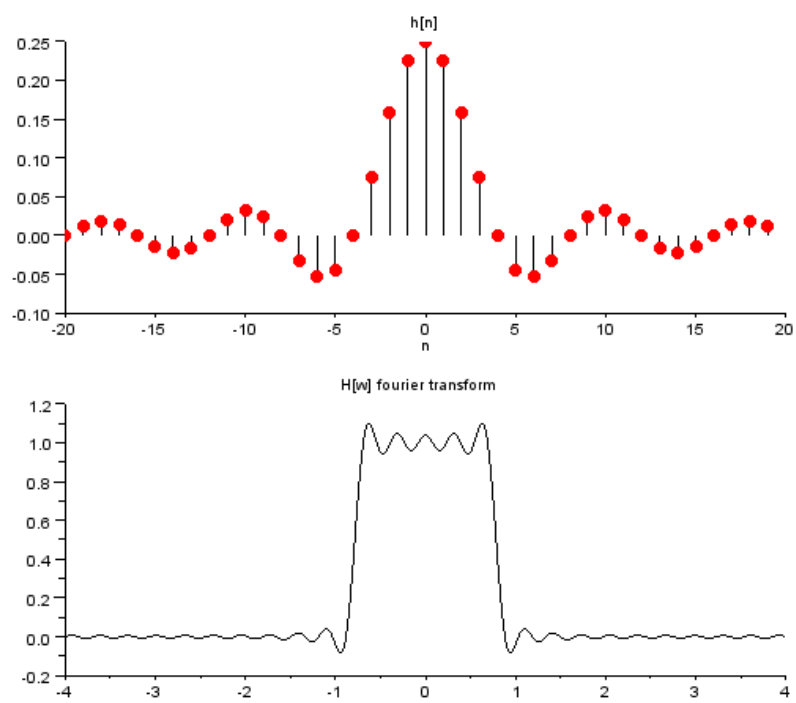


Figure 6.24: output response

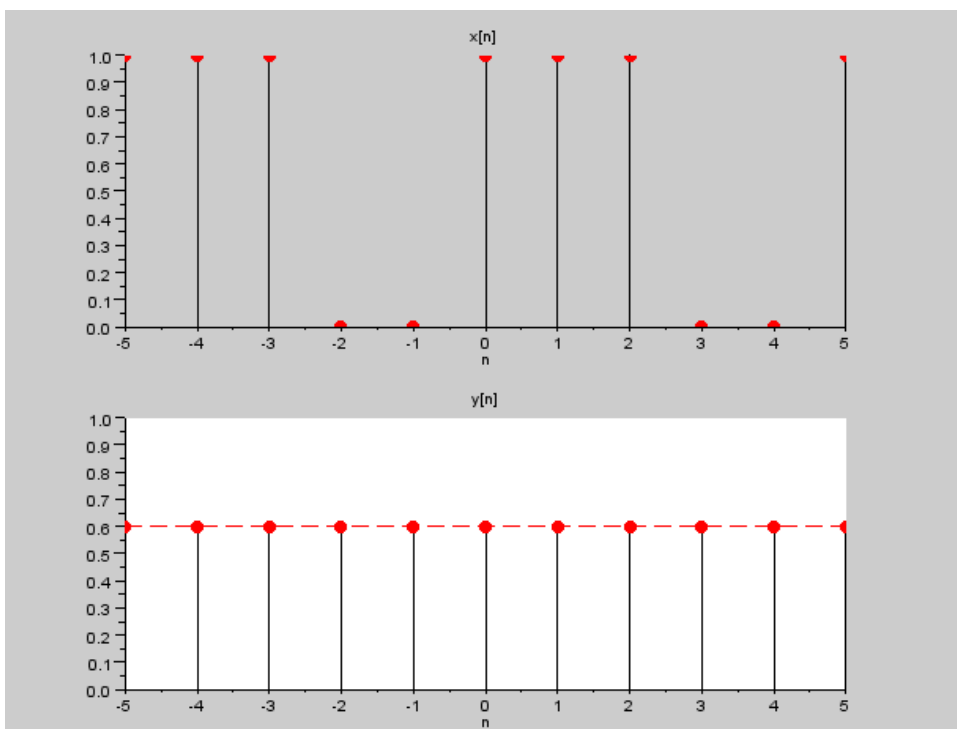


Figure 6.25: output response

```

4 disp("given system is  $y[n]=x[n]+x[n-1]$ ");
5 disp("taking fourier transform  $H(w)=Y(w)/X(w)=1+e^{-j}$ 
      *w");
6 //for impulse response  $x[n]=\delta[n]$ 
7 n=-10:10;
8 for i=1:length(n)
9     if n(i)==0 then
10         delta1(i)=1;
11         delta2(i)=0;
12     elseif n(i)==1
13         delta2(i)=1;
14         delta1(i)=0
15     else
16         delta1(i)=0;
17         delta2(i)=0
18     end
19 end
20 h=delta1+delta2;
21 plot2d3(n,h);
22 plot(n,h,'r. ');
23 xtitle('h[n]','n')
24 figure
25 subplot(2,1,1)
26 w=-3:0.01:3;
27 Hw=1+%e^(-%i*w);
28 plot(w,abs(Hw))
29 xtitle('|H(w)|','w')
30 subplot(2,1,2)
31 a=gca();
32 plot(w,phasemag(Hw)*%pi/180)
33 xtitle('theta(H(w))','w')
34 a.y_location="origin";
35 //3-dB bandwidth
36 cutoff=find(round(100*abs(Hw))==round(100*max(abs(Hw)
      )/sqrt(2))));
37 threedb=w(cutoff(3));
38 disp(threedb,"3db bandwidth=")

```

---

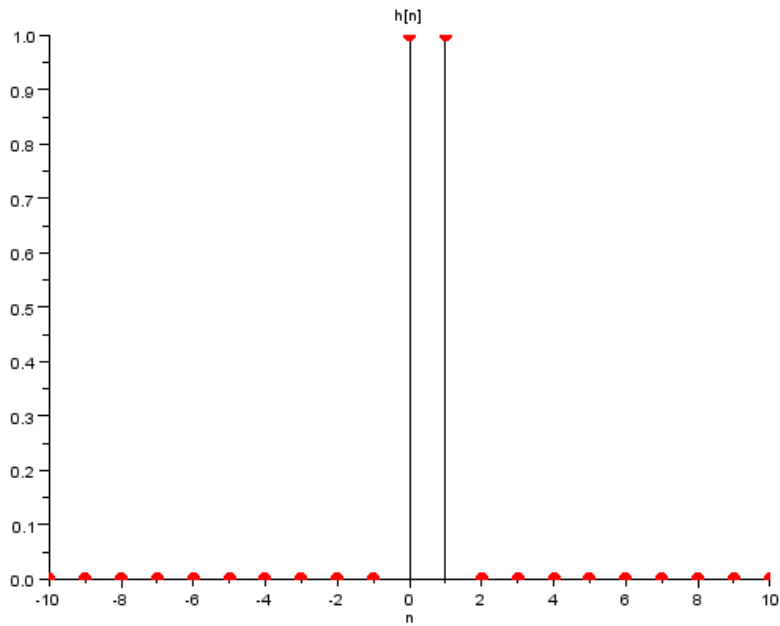


Figure 6.26: magnitude and phase response

**Scilab code Exa 6.35** frequency response

```

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

```



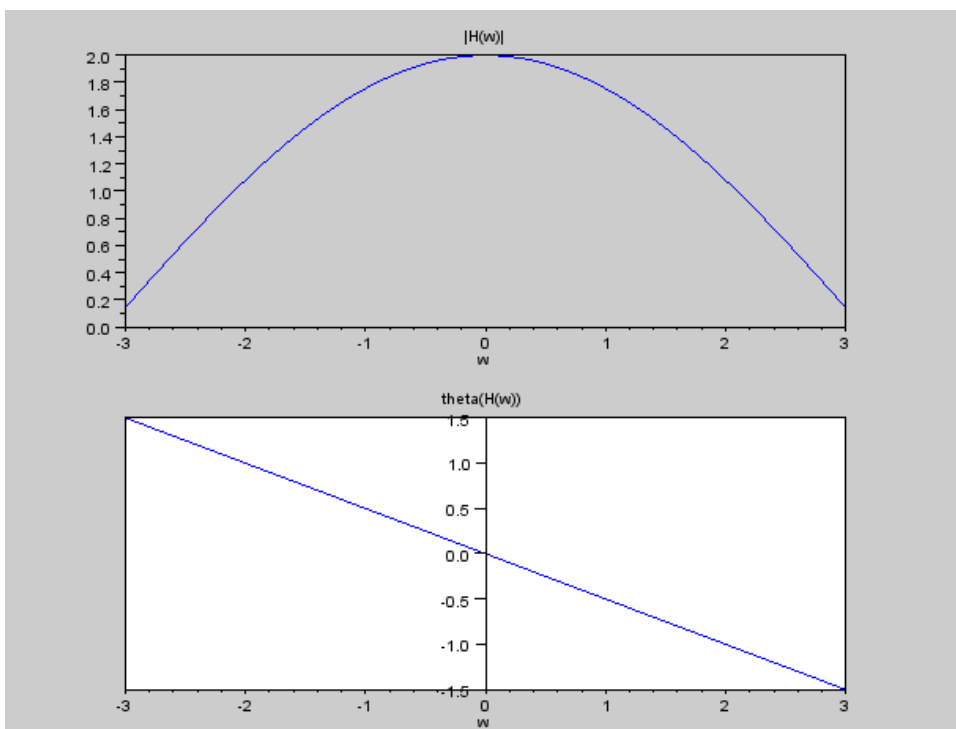


Figure 6.27: magnitude and phase response

```

6 //impulse response
7 n=-10:10;
8 w=-3:0.01:3;
9 a=.5;
10 Hw=ones(1,length(w))./(1-a*%e^(-%i*w));
11 h=(1/2*%pi)*Hw*exp(%i*w'*n);
12 disp("impulse response is a^n*u[n]")
13 plot2d3(n,h);
14 plot(n,h,'r.')
15 xtitle('h[n]','n')
16 figure
17 plot(w,abs(Hw),'r')
18 a=0.9;
19 Hw=ones(1,length(w))./(1-a*%e^(-%i*w));
20 plot(w,abs(Hw),'b')
21 xtitle('|H(w)|','w')
22 legend(['a=0.5'; 'a=0.9']);

```

---

Scilab code Exa 6.36 discrete time low pass filter

```

1 clear;
2 clc;
3 close;
4 wLPF=-3:0.1:3;
5 wc=1;
6 for i=1:length(wLPF)
7     if wLPF(i)>-wc & wLPF(i)<wc then
8         HwLPF(i)=1;
9     else

```

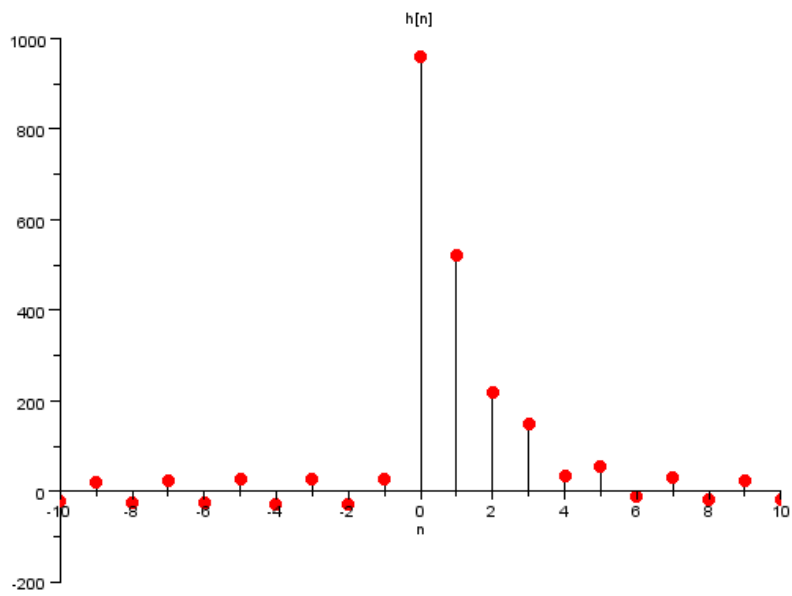


Figure 6.28: frequency response

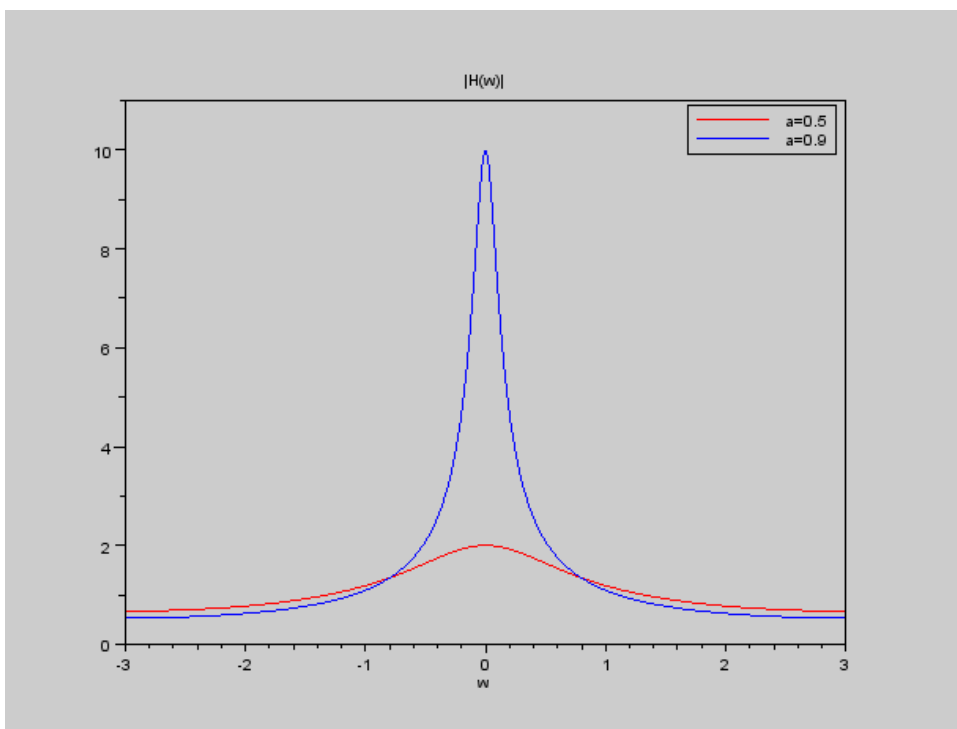


Figure 6.29: frequency response

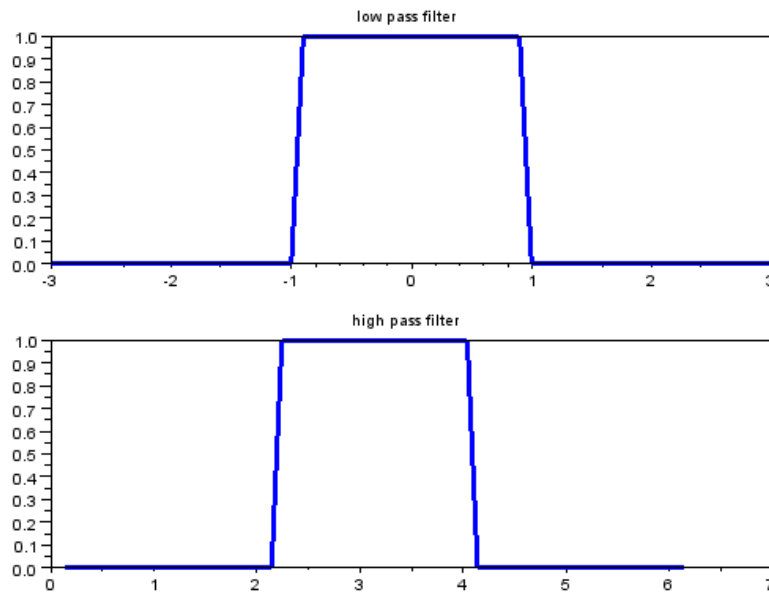


Figure 6.30: discrete time low pass filter

```

10         HwLPF(i)=0;
11     end
12 end
13 subplot(2,1,1)
14 plot(wLPF,HwLPF)
15 a=gca()
16 poly1=a.children.children;
17 poly1.thickness=3;
18 poly1.foreground=2;
19 title("low pass filter")
20 disp("given  $h[n]=(-1)^n * hLPF[n]=e^{j*\%pi*n} * hLPF$ 
       $[n]$ ");
21 disp("by shifting property  $H(w)=HLPF(w-\%pi)$ ")
22 w=wLPF+%pi;
23 subplot(2,1,2)
24 plot(w,HwLPF)
25 a=gca()
26 poly1=a.children.children;

```

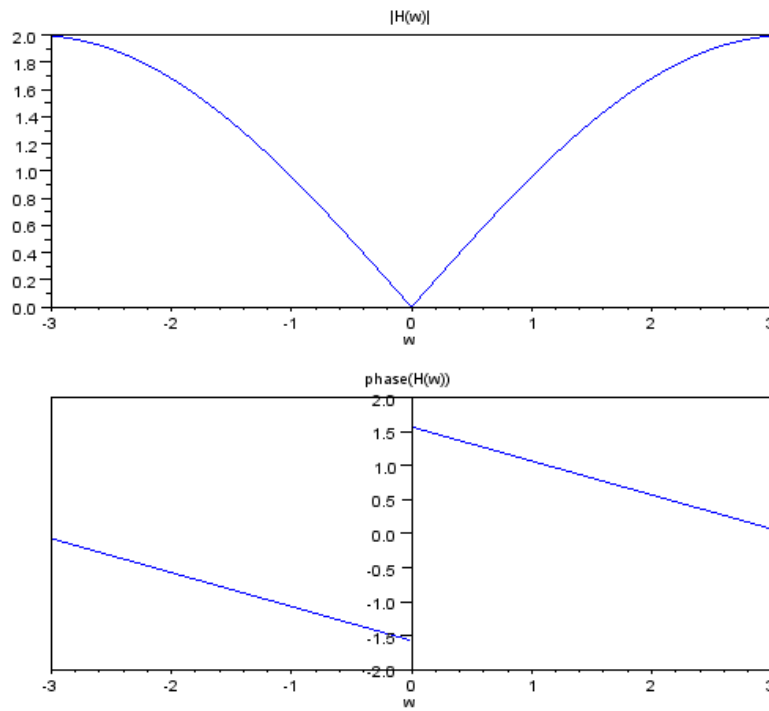


Figure 6.31: conversion of LPF to HPF

```

27 poly1.thickness=3;
28 poly1.foreground=2;
29 title("high pass filter")

```

---

**Scilab code Exa 6.38** conversion of LPF to HPF

```

1 clear;
2 clc;
3 close;
4 disp("given  $y[n]=x[n]+x[n-1]$  this can be converted
      to high pass filter by multiplying with  $(-1)$ ")

```

```

    ^n")
5  disp("then  $y[n]=x[n]-x[n-1]$  taking fourier transform
    ");
6  disp("H(w)=1-e-j*w");
7  w=-3:0.01:3;
8  Hw=1-%e(-i*w);
9  subplot(2,1,1)
10 plot(w,abs(Hw));
11 xtitle(' |H(w)| ', 'w')
12 subplot(2,1,2)
13 plot(w(1:find(w==0)-1),phasemag(Hw(1:find(w==0)-1))*
    %pi/180)
14 a=gca();
15 a.y_location="origin";
16 plot(w(find(w==0)+1:$),phasemag(Hw(find(w==0)+1:$))*
    %pi/180)
17 xtitle(' phase(H(w)) ', 'w')

```

---

**Scilab code Exa 6.40** impulse response os a FIR filter

```

1  close;
2  clc;
3  clear;
4  N=7;
5  n=-4:12;
6  for i=1:length(n)
7      if n(i)>=0 & n(i)<=(N-1)/2 then
8          h(i)=n(i);
9      elseif n(i)>=(N-1)/2 & n(i)<N
10         h(i)=-n(i)+(13+(-1)^N)/2;
11     else
12         h(i)=0;
13
14     end
15 end

```

```

16 subplot(2,1,1)
17 plot2d3(n,h)
18 plot(n,h,'r.')
19 title("N is odd")
20 N=8;
21 n=-4:12;
22 for i=1:length(n)
23     if n(i)>=0 & n(i)<=(N-1)/2 then
24         h(i)=n(i);
25     elseif n(i)>=(N-1)/2 & n(i)<N
26         h(i)=-n(i)+(13+(-1)^N)/2;
27     else
28         h(i)=0;
29
30     end
31 end
32 subplot(2,1,2)
33 plot2d3(n,h)
34 plot(n,h,'r.')
35 title("N is even")
36 disp("given h[n]=h[N-1-n] taking DFTF we get")
37 disp("H(w)=H*(w) e^(-j*(N-1)*w)");
38 disp("phase(w)=-1/2(N-1)*w");
39 //h[n]=-h[N-1-n]
40 N=7;
41 n=-4:12;
42 for i=1:length(n)
43     if n(i)>=0 & n(i)<N then
44         h(i)=n(i)-(N-1)/2;
45     else
46         h(i)=0;
47
48     end
49 end
50 figure
51 subplot(2,1,1)
52 plot2d3(n,h)
53 plot(n,h,'r.')

```



```

54 title("N is odd")
55 N=8;
56 n=-4:12;
57 for i=1:length(n)
58     if n(i)>=0 & n(i)<N then
59         h(i)=n(i)-(N-1)/2;
60     else
61         h(i)=0;
62     end
63 end
64 subplot(2,1,2)
65 plot2d3(n,h)
66 plot(n,h,'r.')
67 title("N is even")
68 disp("given h[n]=-h[N-1-n] taking DFTF we get")
69 disp("H(w)=-H*(w) e^(-j*(N-1)*w)");
70 disp("phase(w)=%pi/2 - 1/2(N-1)*w")

```

---

**Scilab code Exa 6.41** three point moving average discrete time filter

```

1 close;
2 clc;
3 clear;
4 disp("given system y[n]=1/3*{x[n]+x[n-1]+x[n-2]}");
5 disp("taking h[n]=1/3*{delta[n]+delta[n-1]+delta[n-2]}");
6 n=-10:10;
7 for i=1:length(n)
8     if n(i)==0 then
9         delta1(i)=1;
10        delta2(i)=0;

```

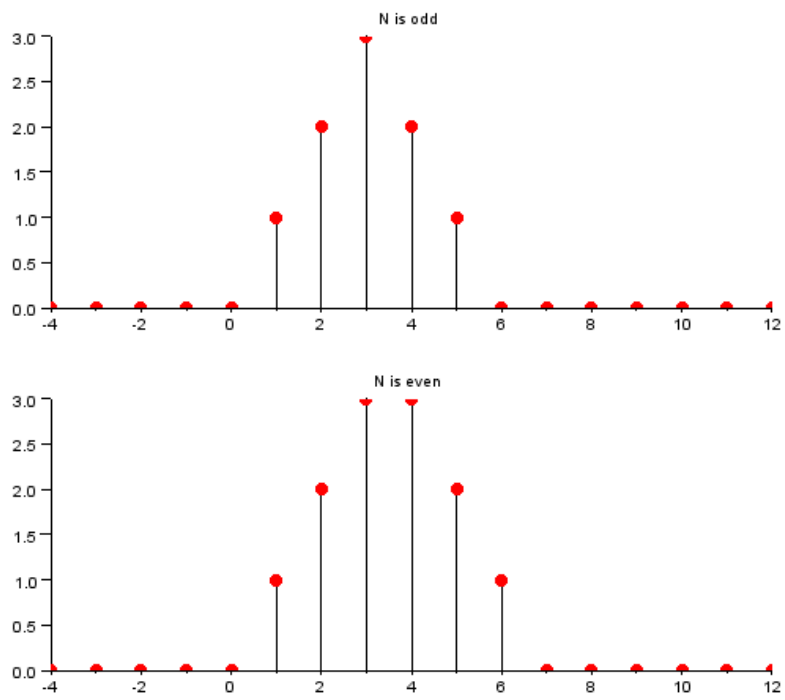


Figure 6.32: impulse response of a FIR filter

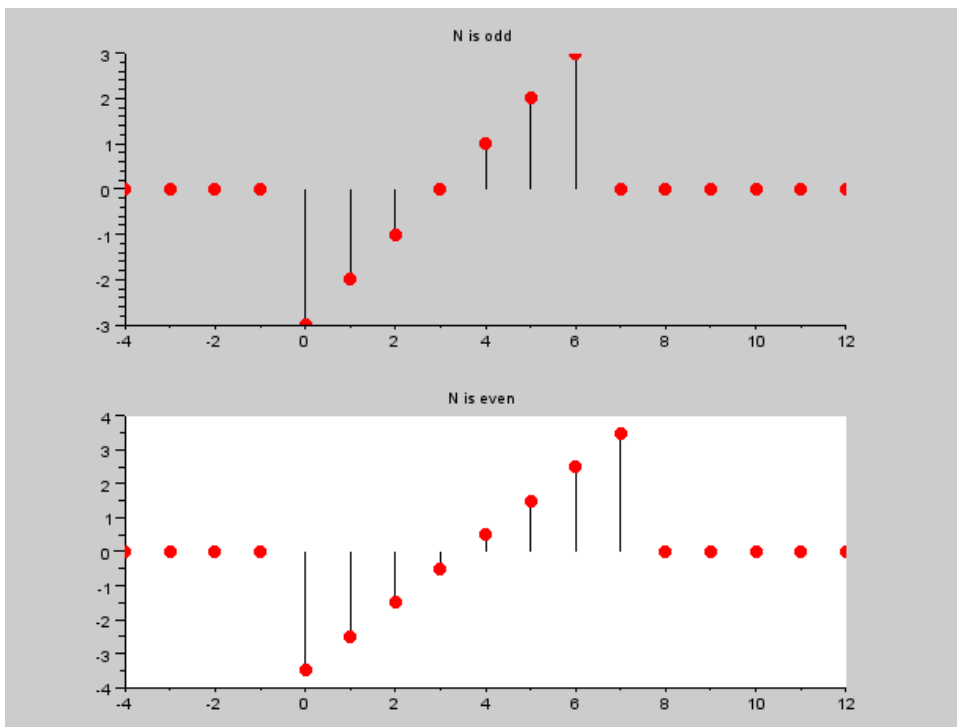


Figure 6.33: impulse response of a FIR filter

```

11         delta3(i)=0;
12     elseif n(i)==1
13         delta2(i)=1;
14         delta1(i)=0;
15         delta3(i)=0;
16     elseif n(i)==2
17         delta1(i)=0;
18         delta2(i)=0;
19         delta3(i)=1;
20     else
21         delta1(i)=0;
22         delta2(i)=0;
23         delta3(i)=0;
24     end
25 end
26 h={delta1+delta2+delta3}/3;
27 plot2d3(n,h)
28 plot(n,h,'r.')
29 xtitle('h[n]','n')
30 disp("taking DTFT  $H(w)=\{1+e^{-j*w}+e^{-2*j*w}\}/3$ ");
31 w=-3:0.1:3;
32 Hw={1+%e^(-%i*w)+%e^(-2*%i*w)}/3;
33 figure
34 subplot(2,1,1)
35 plot(w,abs(Hw));
36 xtitle('|H(w)|','w')
37 subplot(2,1,2)
38 plot(w,phasemag(Hw)*%pi/180);
39 xtitle('phase(H(w))','w')

```

---

Scilab code Exa 6.42 causal discrete time FIR filter

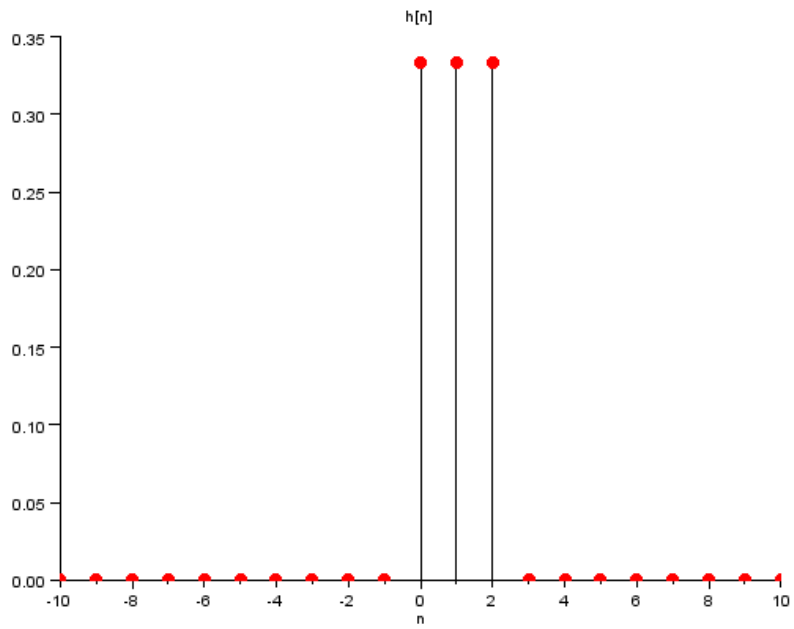


Figure 6.34: three point moving average discrete time filter

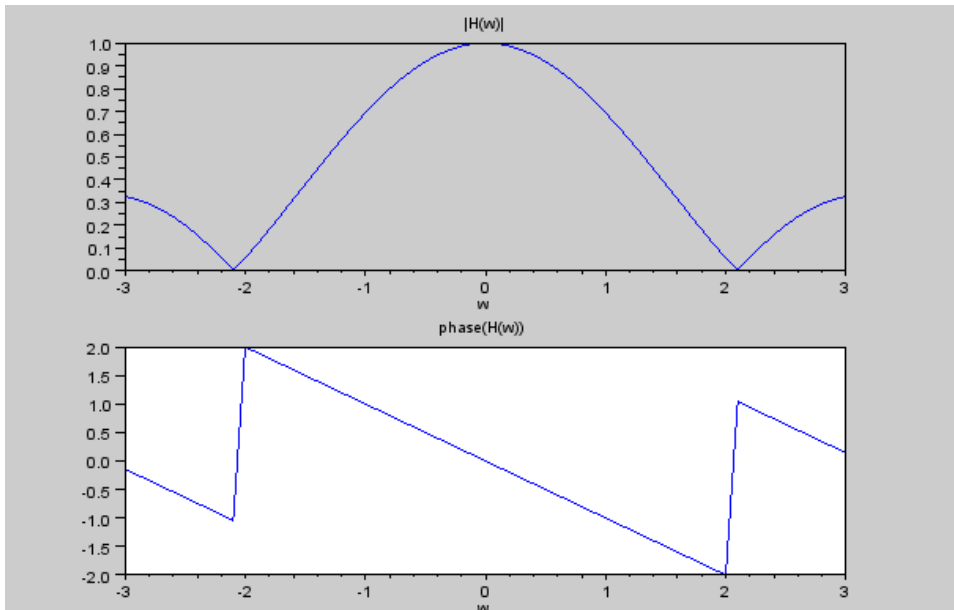


Figure 6.35: three point moving average discrete time filter

```

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

```

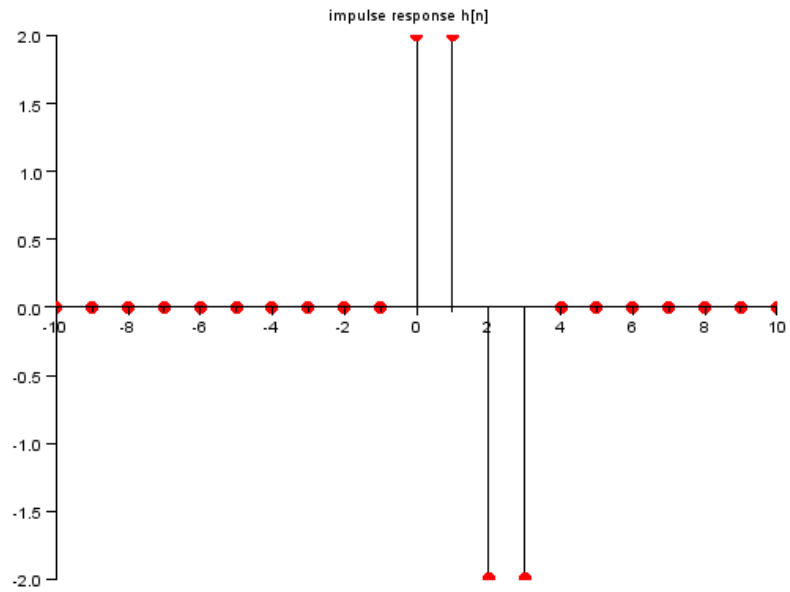


Figure 6.36: causal discrete time FIR filter

```

    %pi/180)
18 a=gca();
19 a.y_location="origin";
20 xtitle('phase(H(w))','w')
21 plot(w(find(w==0)+1:$),phasemag(Hw(find(w==0)+1:$))*
    %pi/180)

```

---

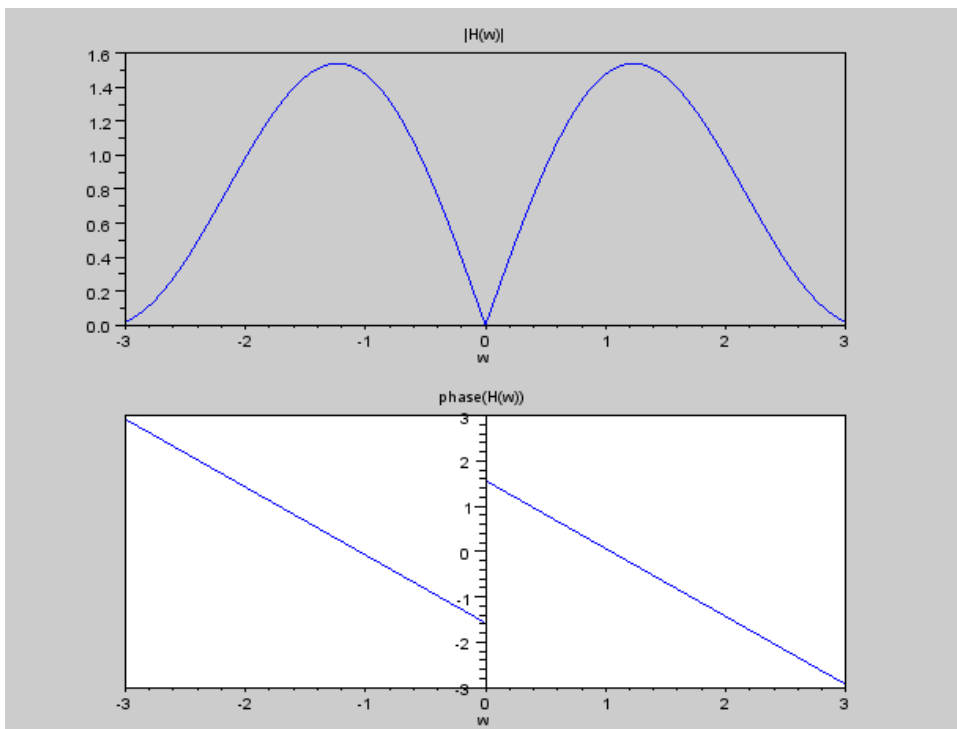


Figure 6.37: causal discrete time FIR filter



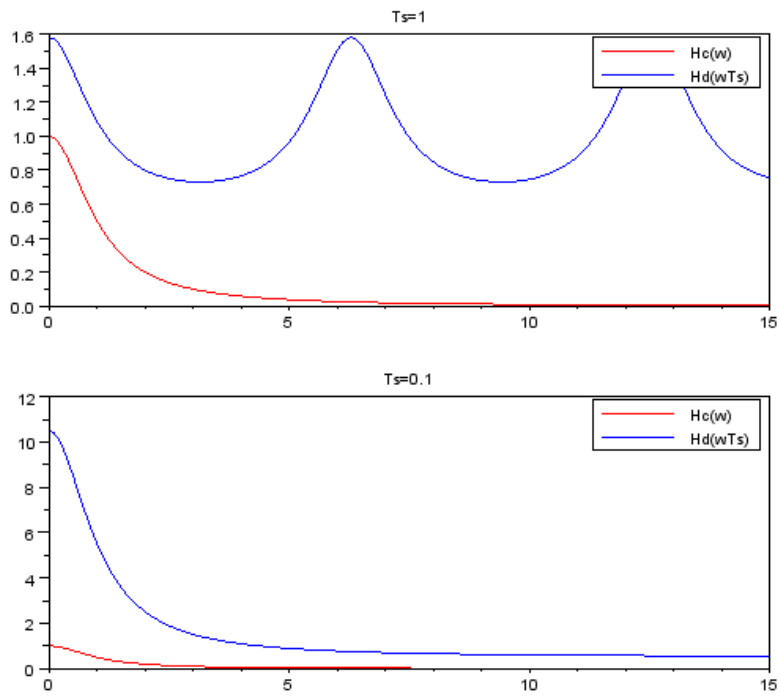


Figure 6.38: Rc low pass filter

### Scilab code Exa 6.43 Rc low pass filter

```
1 close;
2 clear;
3 clc;
4 disp("the system can be represented as  $Y_c(s)=X_c(s)$ 
       $/(1+RCs)$ ");
5 disp("therefore  $H_c(s)=1/(1+RCs)$  hence  $h_c(t)=e^{-t}u(t)$ 
      ) by frequency shifting property");
6 disp("therefore  $h_d[n]=h_c(t)=e^{(-n*Ts)*u[n]}$ ");
7 disp("taking z-transform  $H_d(z)=1/(1-e^{(-n*Ts)*z^{-1}})$ 
      ");
8 Ts=1;
9 w=0:0.1:15;
10 Hcw=ones(1,length(w))./(1+%i*w);
11 subplot(2,1,1)
12 plot(w,Hcw,'r')
13 //z=%e^%i*w*Ts
14 title("Ts=1")
15 Hdw=ones(1,length(w))./(1-exp(-Ts-%i*w*Ts));
16 plot(w,Hdw,'b')
17 legend(["Hc(w)";"Hd(wTs)"])
18 Ts=0.1;
19 w=0:0.1:15;
20 Hcw=ones(1,length(w))./(1+%i*w);
21 subplot(2,1,2)
22 plot(w,Hcw,'r')
23 //z=%e^%i*w*Ts
24 title("Ts=0.1")
25 Hdw=ones(1,length(w))./(1-exp(-Ts-%i*w*Ts));
26 //Hdw=ones(1,length(w))./(1+%e^(-2*Ts)-2*%e^(-Ts)*
      cos(w*Ts))^0.5;
27 plot(w,Hdw,'b')
28 legend(["Hc(w)";"Hd(wTs)"])
```

---

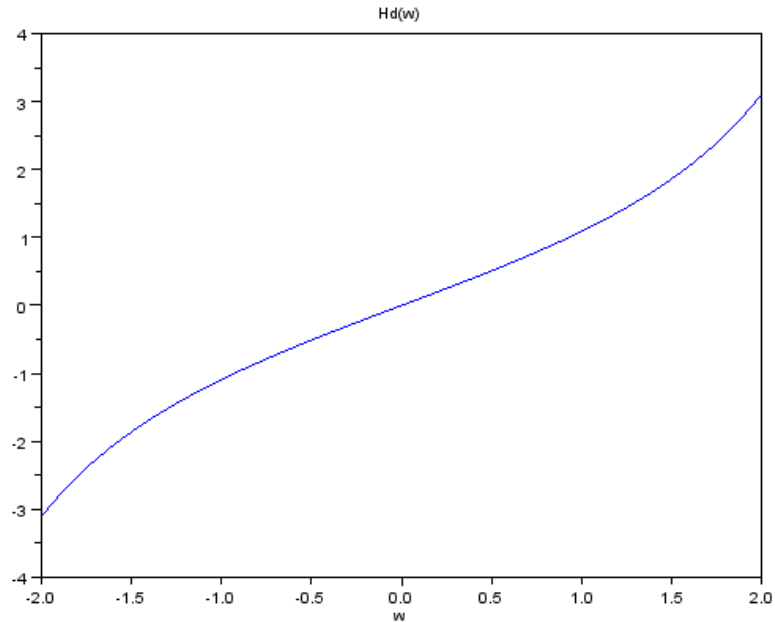


Figure 6.39: bilinear transformation

Scilab code Exa 6.44 frequency response

```

1 close;
2 clear;
3 clc;
4 disp("the system is Hc(s)=1/(s+1)*(s+2)");
5 disp("therefore hc(t)=(e^-t-e^-2*t)*u(t) by
      frequency shifting property");
6 disp("therefore hd[n]=hc(t)=(e^(-n*Ts)-e^(-2*n*Ts))*
      u[n]");
7 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

```
1 close;
2 clear;
3 clc;
4 disp("Hc(s)=s")
5 disp(" bilinear transformation s=2*(1-z^-1)/Ts*(1+z
      ^-1)");
6 disp("Hd(z)=2*(1-z^-1)/Ts*(1+z^-1)")
7 //z=e^-j*w*Ts
8 w=-1:0.01:1;
9 Ts=2;
10 Hdw=2*(1-%e^(-%i*w*Ts))./(1+%e^(-%i*w*Ts));
11 //HdwTs=2*tan(w*Ts/2)/Ts;
12 plot(w*Ts, imag(Hdw));
13 xtitle('Hd(w)', 'w')
```

---

### Scilab code Exa 6.49 N point DFT

```
1 close;
2 clc;
3 clear;
4 n=0:10;
5 N=8;
6 for i=1:length(n)
7     if n(i)==0 then
8         delta(i)=1;
9     else
10        delta(i)=0;
11    end
12 end
13 x=delta';
```

```

14 subplot(2,1,1)
15 plot2d3(n,x);
16 a=gca();
17 plot(n,x,'r.')
18 poly1=a.children.children;
19 poly1.thickness=3;
20 poly1.foreground=2;
21 xtitle('x[n]','n')
22 X=dft(x,-1);
23 Y=X(1:find(n==N)-1);
24 subplot(2,1,2)
25 plot2d3(n,[Y' 0 0 0]);
26 a=gca();
27 plot(n,[Y' 0 0 0],'r.')
28 poly1=a.children.children;
29 poly1.thickness=3;
30 poly1.foreground=2;
31 xtitle('X[k]','k')
32 for i=1:length(n)
33     if n(i)>=0 & n(i)<N then
34         x(i)=1;
35     else
36         x(i)=0;
37     end
38
39 end
40 figure
41 subplot(2,1,1)
42 plot2d3(n,x);
43 plot(n,x,'r.')
44 a=gca();
45 poly1=a.children.children;
46 poly1.thickness=3;
47 poly1.foreground=2;
48 xtitle('x[n]','n')
49 subplot(2,1,2)
50 plot2d3(n,round(dft(x,-1)));
51 plot(n,round(dft(x,-1)),'r.')

```

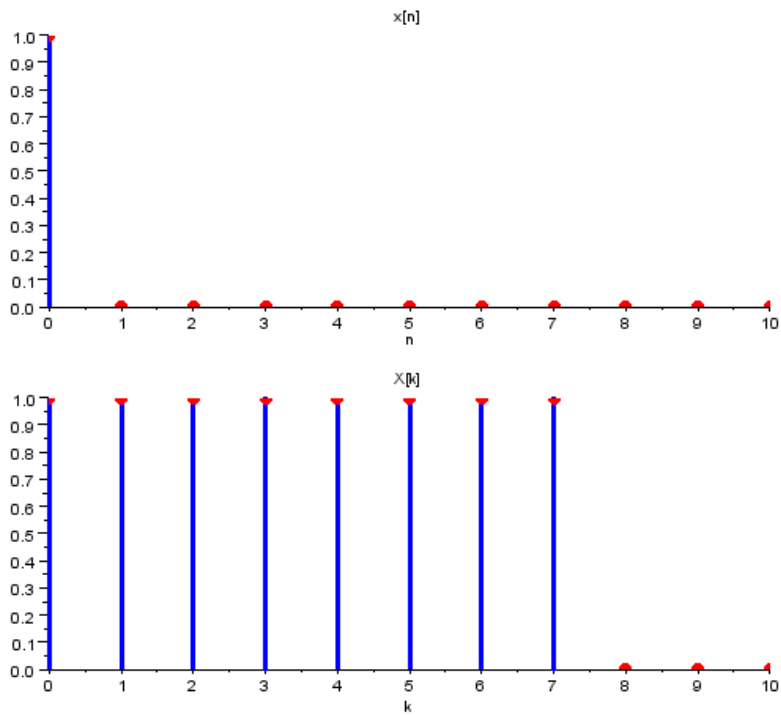


Figure 6.40: N point DFT

```

52 a=gca();
53 poly1=a.children.children;
54 poly1.thickness=3;
55 poly1.foreground=2;
56 xtitle('X[k]','k')

```

---

Scilab code Exa 6.50 DFT

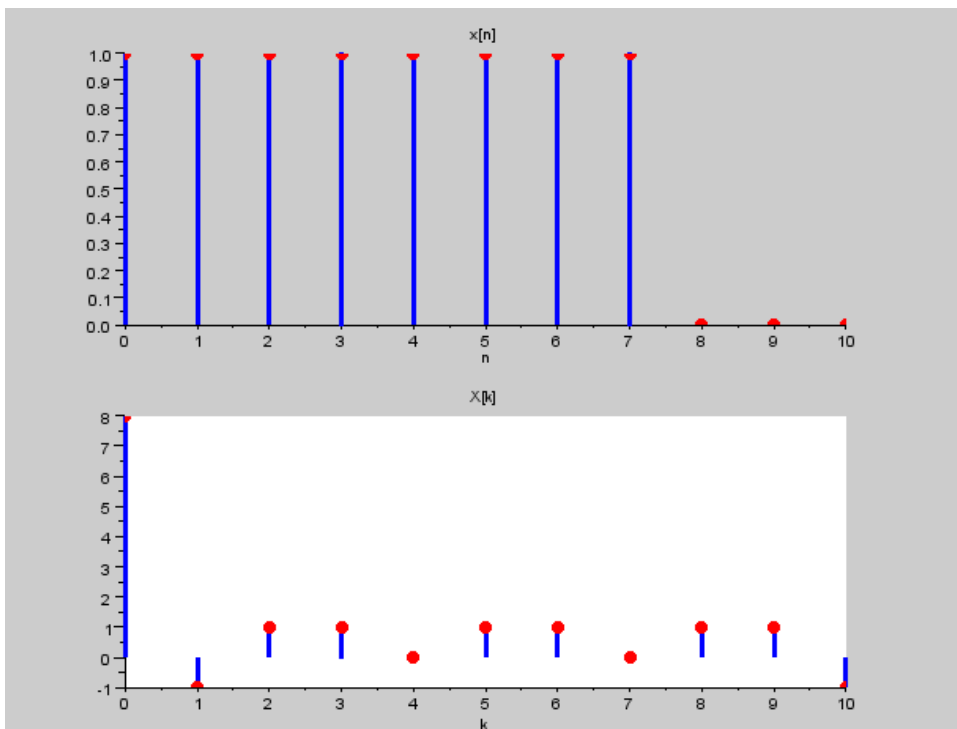


Figure 6.41: N point DFT

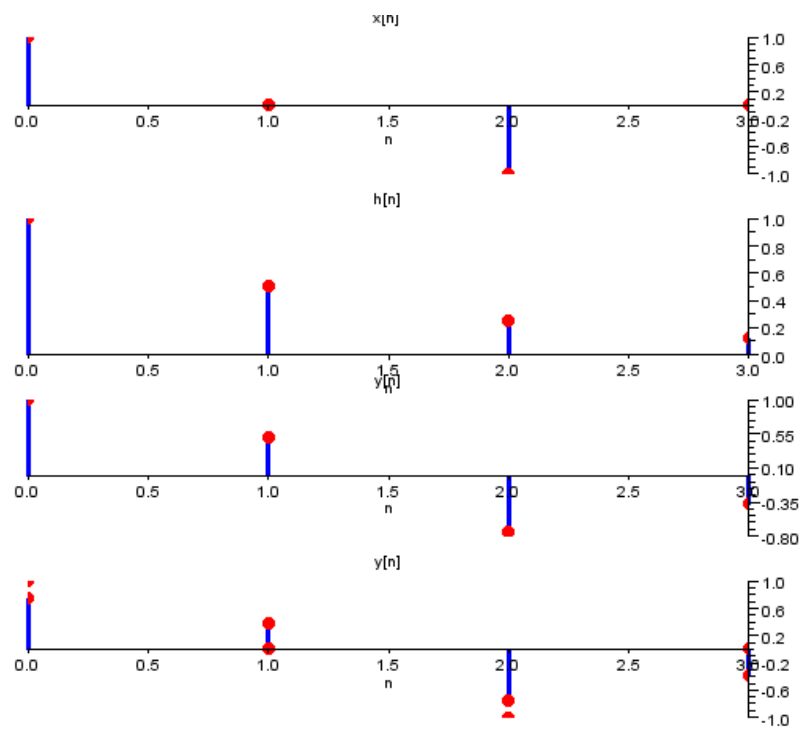


Figure 6.42: DFT



```

1  close;
2  clc;
3  clear;
4  n=0:3;
5  x=cos(%pi*n/2);
6  subplot(4,1,1)
7  plot2d3(n,x)
8  xtitle('x[n]', 'n')
9  a=gca();
10 a.x_location="origin";
11 a.y_location="right";
12 poly1=a.children.children;
13 poly1.thickness=3;
14 poly1.foreground=2;
15 plot(n,x, 'r. ')
16 subplot(4,1,2)
17 h=.5^n;
18 plot2d3(n,h);
19 xtitle('h[n]', 'n')
20 a=gca();
21 a.x_location="origin";
22 a.y_location="right";
23 poly1=a.children.children;
24 poly1.thickness=3;
25 poly1.foreground=2;
26 plot(n,h, 'r. ')
27 y=convol(x,h)
28 subplot(4,1,3)
29 plot2d3(n,y(1:4));
30 xtitle('y[n]', 'n')
31 plot(n,y(1:4), 'r. ')
32 a=gca();
33 a.x_location="origin";
34 a.y_location="right";
35 poly1=a.children.children;
36 poly1.thickness=3;
37 poly1.foreground=2;
38 clear y;

```

```

39 X=dft(x,-1);
40 H=dft(h,-1);
41 Y=H.*X;
42 y=dft(Y,1);
43 subplot(4,1,4)
44 plot2d3(n,y);
45 xtitle('y[n]','n')
46 plot(n,y,'r.')
47 a=gca();
48 a.x_location="origin";
49 a.y_location="right";
50 poly1=a.children.children;
51 poly1.thickness=3;
52 poly1.foreground=2;
53 plot(n,x,'r.')

```

---

#### Scilab code Exa 6.55 DFT using matrices

```

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

```

---

#### Scilab code Exa 6.56 DFT using matrices

```

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

```

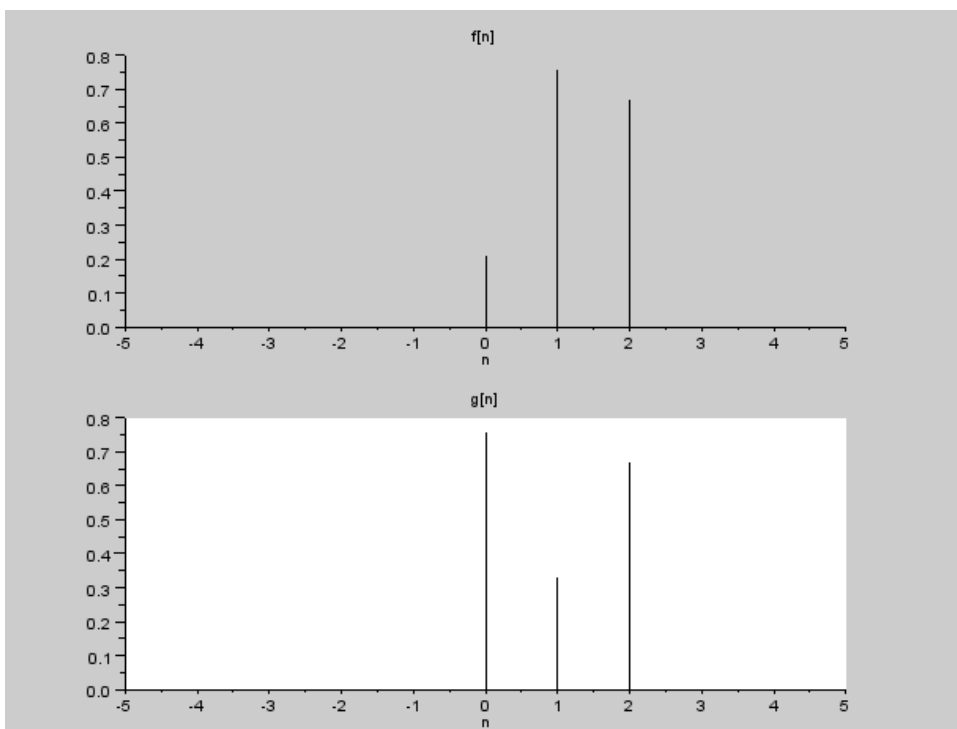


Figure 6.43: DFT using matrices

```

5 plot2d3(n,x)
6 nf=-5:0.5:5;
7 figure
8 subplot(2,1,1)
9 plot2d3(ceil(nf),x);
10 xtitle('f[n]','n')
11 subplot(2,1,2)
12 ng=nf-1/2;
13 plot2d3(ceil(ng),x);
14 xtitle('g[n]','n')
15 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

```

1 clear;
2 close;
3 clc;
4 x=[1 1 -1 -1 -1 1 1 -1];
5 N=8;
6 n=0:7;
7 k=0:N/2-1;
8 WkN=exp(-%i*2*%pi*k/N);
9 f=[];
10 g=[];
11 for i=1:length(n)
12     if modulo(n(i),2)==0 then
13         f=[f x(i)];
14     else
15         g=[g x(i)];
16     end
17 end
18 Fk=dft(f,-1);
19 Gk=dft(g,-1);
20 //Xk=Fk-WkN*Gk

```

```

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

```

---

**Scilab code Exa 6.59** decimation in time FFT algorithm

```

1 clear;
2 close;
3 clc;
4 x=[1 1 -1 -1 -1 1 1 -1];
5 N=8;
6 n=0:7;
7 p=x(1:4)+x(5:8);
8 k=0:N/2-1;
9 WkN=exp(-%i*2*%pi*k/N);
10 q=[x(1:4)-x(5:8)].*WkN;
11 Pk=dft(p,-1);
12 Qk=dft(q,-1);
13 //X[2k]=P[k] X[2k+1]=Q[k]
14 for i=1:length(k)
15     if modulo(i,2)==0 then
16         Xk(i)=Qk(i/2);
17     else
18         Xk(i)=Pk(ceil((i)/2));
19     end
20 end
21 Xk=[Xk;Xk];
22 disp(round(Xk));

```

---

**Scilab code Exa 6.61** fourier spectrum using DFT

```

1 clear;
2 close;

```

```

3  clc;
4  T1=10;
5  wM=100;
6  N=ceil(wM*T1/%pi);
7  dw=2*wM/N;
8  dt=%pi/wM;
9  WN=exp(-i*2*%pi/N);
10 n=0:N-1;
11 x=%e-(n*dt);
12 k=0:10;
13 Xk=dt*x*exp(-i*2*%pi*n'*k/N);
14 disp(Xk(find(k==0)),Xk(find(k==1)),Xk(find(k==10)),”
      X[0] X[1] X[10] from X[k]”)
15 w=0:dw:10*dw;
16 Xw=ones(1,length(w))./(%i*w+1);
17 disp(Xw(find(w==0)),Xw(find(w==dw)),Xw(find(w==10*dw
      )),”X(0) X(dw) X(10*dw) from X(w)”)
18 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

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

---

Scilab code Exa 7.7 state equation for discrete system

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

---

**Scilab code Exa 7.8** state equation for discrete time system

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

---

**Scilab code Exa 7.9** state equation for discrete time system

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

---

**Scilab code Exa 7.11** state equation for discrete time system

```
1 clc;
2 clear;
```



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

---

**Scilab code Exa 7.14** state equation for continuous time system

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

---

**Scilab code Exa 7.15** state equation for continuous time system

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

---

**Scilab code Exa 7.16** state equation for continuous time system

```
1 clc;
2 close
3 clear;
```

```

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

```

---

**Scilab code Exa 7.18** state equation for continuous time system

```

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

```

---

**Scilab code Exa 7.20** A power n

```

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

```

---

**Scilab code Exa 7.24** A power n

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

---

**Scilab code Exa 7.25** A power n

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

---

**Scilab code Exa 7.26** decomposition of matrix A

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

```

6 N=[0 1;0 0];
7 disp(N^2,"N*N=");
8 p=D*N;
9 q=N*D;
10 if (p==q) then
11     disp("D and N commute")
12 end

```

---

### Scilab code Exa 7.27 minimal polynomial

```

1 clc;
2 syms n;
3 A=[2 0 0;0 -2 1;0 4 1];
4 eig=spec(A);
5 m=poly(eig(1:2),"A",["roots"]);
6 disp(m,"minimal polynomial is")
7 e=spec(A);
8 b1=(e(2)^n-e(1)^n)/(e(2)-e(1));
9 b0=e(2)^n-b1*e(2);
10 An=b0*eye(A)+b1*A;
11 disp(An,"A^n")

```

---

### Scilab code Exa 7.29 step response

```

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

```

```

9 syms n z1; //To find out Inverse z t r a n
  s f o r m z must be l i n e a r z = z1
10 X =z ^2 /((z -(1/2) )*(z -(1/4) ))
11 X1 = denom (X);
12 zp = roots (X1);
13 X1 = z1 ^2 /(( z1 -(1/4) )*(z1 -(1/2) ))
14 F1 = X1 *( z1 ^(n -1) )*(z1 -zp (1) );
15 F2 = X1 *( z1 ^(n -1) )*(z1 -zp (2) );
16 h1 = limit (F1 ,z1 ,zp (1) );
17 disp (h1 , ' h1 [ n]= ' )
18 h2 = limit (F2 ,z1 ,zp (2) );
19 disp (h2 , ' h2 [ n]= ' )
20 h = h1+h2;
21 disp (h, ' h [ n]= ' )
22 n=0:10;
23 x=[zeros(1,find(n==0)-1) ones(1,length(n)-find(n==0)
    +1)];
24 hn=2^(1-n)-4^-n;
25 y=convol(x,hn);
26 plot2d3(n,y(1:length(n)))
27 figure
28 plot2d3(n,8/3-2^(1-n)+4^-n)
29 disp("hence from the figure we can say y[n
    ]=8/3-2^(1-n)+4^-n")

```

---

### Scilab code Exa 7.30 impulse response

```

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

```

```

9 syms n z1; //To find out Inverse z t r a n
  s f o r m z must
10 //be l i n e a r z = z1
11 X =z ^2 /((z -(1/2) )*(z -(1/4) ))
12 X1 = denom (X);
13 zp = roots (X1);
14 X1 = z1 ^2 /((z1 -(1/4) )*(z1 -(1/2) ))
15 F1 = X1 *( z1 ^(n -1) )*(z1 -zp (1) );
16 F2 = X1 *( z1 ^(n -1) )*(z1 -zp (2) );
17 h1 = limit (F1 ,z1 ,zp (1) );
18 disp (h1 , ' h1 [ n]= ' )
19 h2 = limit (F2 ,z1 ,zp (2) );
20 disp (h2 , ' h2 [ n]= ' )
21 h = h1+h2;
22 disp ( 'for n>=0',h, ' h [ n]= ' )

```

---

### Scilab code Exa 7.31 difference equation

```

1 clc;
2 A=[0 1;-1/3 4/3];
3 B=[0;1/3];
4 C=[-1/3 4/3];
5 D=[1/3];
6 Hz=ss2tf(syslin('d',A,B,C,D));
7 disp(Hz,"H(z)=");
8 z = %z;
9 syms n z1;
10 X =z ^2 /(1-4*z+3*z^2)
11 X1 = denom (X);
12 zp = roots (X1);
13 X1 = z1 ^2 /((z1-1)*(z1-1/3))
14 F1 = X1 *( z1 ^(n -1) )*(z1 -1/3 );
15 F2 = X1 *( z1 ^(n -1) )*(z1 -1 );
16 h1 = limit (F1 ,z1 ,zp (1) );
17 disp (h1 , ' h1 [ n]= ' )

```

```

18 h2 = limit (F2 ,z1 ,zp (2) );
19 disp (h2 , ' h2 [ n]= ' )
20 h = h1+h2;
21 disp ( 'for n>=0',h, ' h [ n]= ' )
22 n=0:10;
23 x=2^-n.*[zeros(1,find(n==0)-1) ones(1,length(n)-find
      (n==0)+1)];
24 hn= 3/2-(3^-n)/2;
25 y=convol(x,hn);
26 plot2d3(n,y(1:length(n)))
27 figure
28 plot2d3(n,(3/2)-2^-n+(3^-n)/2)
29 disp("hence from the figure we can say y[n]=(3/2)
      -2^-n+(3^-n)/2")

```

---

#### Scilab code Exa 7.32 stability

```

1 clc;
2 A=[0 3/2;-1/2 2];
3 B=[0.71;-0.71];
4 C=[0.71 -0.71];
5 D=[0];
6 Hz=ss2tf(syslin('d',A,B,C,D));
7 disp(Hz,"H(z)=");
8 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

```

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

```

```

4 C=[-1/8 3/4];
5 D=[1];
6 Mc=[B A*B];
7 if (det(Mc)<>0) then
8     disp("and is controllable",rank(Mc),"Mc has a
          rank of")
9 else
10    disp("and is uncontrollable",rank(Mc),"Mc has a
         rank of");
11 end
12 Mo=[C;C*A];
13 if (det(Mo)<>0) then
14    disp("and is observable",rank(Mo),"Mo has a rank
         of")
15 else
16    disp("and is unobservable",rank(Mo),"Mo has a
         rank of");
17 end
18 Hz=ss2tf(syslin('d',A,B,C,D));
19 disp(Hz,"H(z)=");

```

---

**Scilab code Exa 7.36** finding vector x

```

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

```

---

**Scilab code Exa 7.37** finding vector y



```

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

```

---

**Scilab code Exa 7.38** observability and controllability

```

1  clc;
2  A=[0 3/2;-1/2 2];
3  B=[1;0];
4  C=[1 -1];
5  D=[0];
6  Mc=[B A*B];
7  if (det(Mc)<>0) then
8      disp("and is controllable",rank(Mc),"Mc has a
          rank of")
9  else
10     disp("and is uncontrollable",rank(Mc),"Mc has a
          rank of");
11 end
12 Mo=[C;C*A];
13 if (det(Mo)<>0) then
14     disp("and is observable",rank(Mo),"Mo has a rank
          of")
15 else
16     disp("and is unobservable",rank(Mo),"Mo has a
          rank of");
17 end

```

---

**Scilab code Exa 7.39** e power At

```

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

```

---

Scilab code Exa 7.43 e power At

```

1  clc;
2  A=[-2 1;1 -2];
3  syms t
4  //we know %e^(at)=b0*I+b1*A;
5  e=spec(A); //eigen values
6  b0=e(1)*%e^(e(2)*t)-e(2)*%e^(e(1)*t);
7  b1=%e^(e(1)*t)-%e^(e(2)*t);
8  eAt=b0*eye(A)+b1*A;
9  disp(eAt,"e^(At)")

```

---

Scilab code Exa 7.44 e power At

```

1  clc;
2  syms t
3  A=[0 -2 1;0 0 3;0 0 0];
4  z=A*A*A;
5  if z==0 then
6      disp("A is a nilpotent of index 3");
7  else
8      disp("A is not nilpotent")
9  end

```

```

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

```

---

#### Scilab code Exa 7.45 e power At

```

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

```

---

#### Scilab code Exa 7.47 nilpotent matrix

```

1 clc;
2 A=[2 1 0;0 2 1;0 0 2];
3 V=[2 0 0;0 2 0; 0 0 2];
4 N=[0 1 0;0 0 1;0 0 0];
5 z=N*N*N;
6 if z==0 then
7     disp("N is a nilpotent of index 3");
8 else
9     disp("N is not nilpotent")
10 end

```

```

11 if V*N==N*V then
12     disp("V and N commute")
13 else
14     disp("V and N dont commute")
15 end
16 syms t
17 //e^(A*t)=e^(v*t)*e^(N*t)
18 e=spec(V); //eigen values
19 b2=t^2*%e^(e(3)*t);
20 b1=t*%e^(e(2)*t)-b2*2*e(2);
21 b0=%e^(e(1)*t)-b1*e(1)-b2*e(1)^2;
22 eVt=b0*eye(V)+b1*V+b2*V^2;
23 disp(eVt,"e^(Vt)")
24 e=spec(N); //eigen values
25 b2=t^2*%e^(e(3)*t);
26 b1=t*%e^(e(2)*t)-b2*2*e(2);
27 b0=%e^(e(1)*t)-b1*e(1)-b2*e(1)^2;
28 eNt=b0*eye(N)+b1*N+b2*N^2;
29 disp(eNt,"e^(Nt)")
30 eAt=eVt*eNt;
31 disp(eAt,"e^(A*t)=")

```

---

#### Scilab code Exa 7.48 second order ODE

```

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

```

```

12 b0=e(1)*%e^(e(2)*(t-T))-e(2)*%e^(e(1)*(t-T));
13 b1=%e^(e(1)*(t-T))-e^(e(2)*(t-T));
14 eAtT=b0*eye(A)+b1*A;
15 disp(eAt,"e^(At)");
16 ceAtq0=c*eAt*q0;
17 ceAtTq0=c*eAtT*q0;
18 y=ceAtq0+integ(ceAtTq0*%e^-T,T);
19 y0=limit(y,T,0);
20 yt=limit(y,T,t);
21 Y=yt-y0;
22 disp(Y,"y(t)=")

```

---

#### Scilab code Exa 7.49 RC circuit response

```

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

```

---

#### Scilab code Exa 7.50 stability

```

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

```

```

5 d=[0];
6 e=spec(A);
7 if real(e(1))>0 | real(e(2))>0 then
8     disp("the sytem is not asymptotically stable")
9 else
10    disp("the system is asymptotically stable")
11 end
12 Hs=ss2tf(syslin('c',A,b,c,d));
13 disp(Hs,"H(s)=")
14 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

```

1  clc;
2  A=[0 1;2 1];
3  B=[1;-1];
4  C=[1 -1];
5  D=[0];
6  Mc=[B A*B];
7  if (det(Mc)<>0) then
8     disp("and is controllable",rank(Mc),"Mc has a
          rank of")
9  else
10     disp("and is not controllable",rank(Mc),"Mc has
          a rank of");
11 end
12 Mo=[C;C*A];
13 if (det(Mo)<>0) then
14     disp("and is observable",rank(Mo),"Mo has a rank
          of")
15 else
16     disp("and is unobservable",rank(Mo),"Mo has a
          rank of");

```

17 end

---

**Scilab code Exa 7.54** observability and controllability

```
1  clc;
2  A=[1  2;0  3];
3  B=[1;1];
4  C=[1  -1];
5  D=[0];
6  Mc=[B  A*B];
7  if (det(Mc)<>0) then
8      disp("and is controllable",rank(Mc),"Mc has a
          rank of")
9  else
10     disp("and is uncontrollable",rank(Mc),"Mc has a
          rank of");
11 end
12 Mo=[C;C*A];
13 if (det(Mo)<>0) then
14     disp("and is observable",rank(Mo),"Mo has a rank
          of")
15 else
16     disp("and is unobservable",rank(Mo),"Mo has a
          rank of");
17 end
18 Hs=ss2tf(syslin('c',A,B,C,D));
19 disp(Hs,"H(s)=")
```

---