

System Analysis & Signal Processing

Philip Denbigh

*with emphasis
on the use
of MATLAB®*



ADDISON-WESLEY

System analysis and signal processing

with emphasis on the use of MATLAB

PHILIP DENBIGH

University of Sussex



Harlow, England • Reading, Massachusetts • Menlow Park, California
New York • Don Mills, Ontario • Amsterdam • Bonn • Sydney • Singapore
Tokyo • Madrid • San Juan • Milan • Mexico City • Seoul • Taipei

© Addison Wesley Longman Ltd 1998

Addison Wesley Longman Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the World.

The right of Philip Denbigh to be identified as author of this work has been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the publisher or a licence permitting restricted copying in the United Kingdom issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London W1P 9HE.

The programs in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations nor does it accept any liabilities with respect to the programs.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Addison Wesley Longman Limited has made every attempt to supply trademark information about manufacturers and their products mentioned in this book. A list of the trademark designations and their owners appears on page xii.

Typeset in 10/12.5 Times by 25
Printed and bound in the United States of America

First printed 1998

ISBN 0-201-17860-5

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data

Denbigh, Philip.

System analysis and signal processing : with emphasis on the use
of MATLAB / Philip Denbigh.

p. cm.

Includes bibliographical references (p. 504) and index.

ISBN 0-201-17860-5 (alk. paper)

1. Signal processing—Computer simulation. 2. MATLAB. I. Title.
TK5102.9.D46 1998
621.382'2—dc21

97-40611
CIP

System analysis and signal processing

Preface

This is intended to be very much a *teaching* book, and the effort in writing it has been directed more at developing clear explanations of difficult concepts than at producing a definitive text. It covers a very broad syllabus on system analysis and continuous and discrete-time signal processing that is aimed at being suited to all years of an undergraduate programme, but also includes a small amount of more advanced material.

The book commences at a much more elementary level than would normally be associated with the later content of the book. The motive behind this is to ensure that the more difficult concepts are based upon a solid foundation. It is intended that the book should be substantially self-contained, showing how all the concepts build on each other – how the pieces of the jigsaw fit together. This means that a wide range of topics is covered and, in order to keep the book of reasonable length, the treatment is constrained to the essentials that are thought necessary for a good understanding. What has been left out is considered almost as important as what has been included. Details that some other texts include are deliberately omitted. This is for brevity, so as not to detract from those aspects which are thought to be more important, and so that the reader should not find the book heavy going or daunting. It is accepted that it may sometimes be useful to supplement the material with that from other texts. The bibliography contains a short list of books that have been selected for their readability and/or extended coverage of relevant material.

One special feature of the book is that an attempt is made to throw some physical insight into the Laplace transform, somewhat similar to that which is commonly and much more easily done with the Fourier transform. It is felt that the common practice of introducing the Laplace transform by a *definition*, without any *interpretation* of that definition, is unconvincing and too mathematical for most readers. It is hoped that the treatment given will be more acceptable.

A second major feature of the book is the extensive use in some chapters of the scientific software MATLAB. This is done, firstly because it produces impressive displays that help to explain the text, and secondly because it has become an invaluable and widely accepted software tool whose existence has affected signal processing design procedures. As such, an acquaintance with its power is considered to be an essential part of a signal processing syllabus. A particularly impressive feature of MATLAB is that student versions of it are available for the price of an average hardback textbook. These student versions do have limitations compared with the professional version of MATLAB, but are nevertheless very powerful and

totally adequate for many applications. For use with Microsoft Windows 3.1 the relevant student edition is Version 4, which is available as a diskette plus manual (MATLAB 1995). This can also be used on Microsoft Windows 95 and Windows NT, but the updated student edition for these is Version 5, and this is available as a CD-ROM plus manual (MATLAB 1997). Versions 4 and 5 are also available for the appropriate Macintosh computers. All the MATLAB commands in this book are restricted to commands that are valid for both of these student editions, as well as being fully compatible with the professional versions of MATLAB which may be available to students through a copy on their university network.

Although primarily concerned with system analysis and signal processing, a secondary function of this book is to provide an introductory tutorial on the use of MATLAB. For this reason the code used is listed in the text rather than on an attached diskette. Sometimes MATLAB is used solely for producing the figures, and it is felt inappropriate to include code in the main text. Since, however, this coding may be of interest to the reader learning MATLAB programming, a selected number of these figures are marked by an asterisk (for example Fig. 3.3*) and the code is listed in Appendix A.

The problems at the end of each chapter are few in number but have been carefully designed to make a significant contribution to the learning process. If the material is understood most of them can be solved quite quickly. A solutions manual is available to instructors.

It is important to end with some acknowledgements. I should like to express my gratitude to the anonymous reviewers of the original manuscript for making a variety of constructive suggestions, many of which were subsequently incorporated. Also, I should like to thank Ahmad Shamsoddini for passing on some of his knowledge on the less well-documented aspects of MATLAB and for his meticulous reading of my corrected proofs. As well as finding a number of remaining typos he pointed out and helped remedy a few unclear, and even erroneous, statements. However it is almost inevitable that some mistakes will remain and if readers find any, or can suggest improvements or areas that need clarification, I would be grateful to hear of them. My e-mail address is p.n.denbigh@sussex.ac.uk.

Philip Denbigh
December 1997

Contents

1. Getting started in MATLAB and an introduction to systems and signal processing	1
1.1 Preview	1
1.2 Getting started in MATLAB	1
1.2.1 Variables	2
1.2.2 Sequences, arrays and vectors	3
1.2.3 Generating vectors	4
1.2.4 Addressing vectors	7
1.2.5 Array mathematics	8
1.2.6 Multiple commands on one line and comments	9
1.2.7 Element indexing and time indexing	9
1.2.8 Augmenting a vector with zeros to correspond with an extended timing index vector	11
1.2.9 Linear algebra and matrix operations	12
1.2.10 Plotting data values	13
1.2.11 Hardcopy	15
1.2.12 Transfer of a figure to a word processor document	16
1.2.13 M-files	16
1.3 Some signals and systems terminology	17
1.4 Some examples of analogue systems and analogue signal processing	20
1.5 Some examples of digital systems and digital signal processing	21
1.6 Justification for the digital processing of signals	27
1.7 Some signals of special importance	27
1.8 Summary	30
1.9 Problems	30
2. Impulse functions, impulse responses and convolution	32
2.1 Preview	32
2.2 The unit sample function (or discrete-time unit impulse function)	33
2.3 Discrete-time impulse responses and the convolution sum	33
2.4 Alternative forms and interpretations of the convolution sum	38
2.5 Convolution using MATLAB	40
2.6 Continuous-time impulse functions	44
2.7 Continuous-time impulse responses and the convolution integral	45
2.8 A graphical interpretation of the convolution integral	47
2.9 Convolution of analogue signals using MATLAB	49
2.10 Some examples of convolution not related to networks	50

2.11	Some properties of convolution	53
2.12	Deconvolution	53
2.13	Summary	54
2.14	Problems	54
3.	The steady state response of analogue networks to cosinusoids and to the complex exponential $e^{j\omega t}$	56
3.1	Preview	56
3.2	The properties of network elements	56
3.3	The difficulty of solving network equations	57
3.4	The use of the complex exponential waveform $\exp(j\omega t)$	59
3.5	Impedance functions	61
3.5.1	Inductance	61
3.5.2	Resistance	63
3.5.3	Capacitance	63
3.6	Admittance functions	64
3.7	Elements in series and in parallel	64
3.8	Frequency transfer functions and Bode plots	67
3.9	Summary	69
3.10	Problems	70
4.	Phasors	71
4.1	Preview	71
4.2	Vector representation of cosinusoids	71
4.3	Phasors and phasor diagrams	75
4.4	Phasors in network analysis	77
4.5	The use of phasors in power calculations	79
4.6	The application of MATLAB to network analysis	82
4.7	Phasors not related to network analysis	83
4.8	Contra-rotating phasors	88
4.9	Summary	89
4.10	Problems	89
5.	Line spectra and the Fourier series	91
5.1	Preview	91
5.2	One-sided frequency domain descriptions of sinusoids and cosinusoids	91
5.3	Two-sided frequency domain descriptions of cosinusoids	92
5.4	Negative frequencies	94
5.5	Periodic signals and the trigonometric Fourier series	94
5.6	The exponential Fourier series	97
5.7	Plotting line spectra using MATLAB	101
5.8	Further properties and examples of Fourier series	103
5.9	Summary	107
5.10	Problems	107
6.	Spectral density functions and the Fourier transform	109
6.1	Preview	109
6.2	Energy signals and power signals	110

6.3	The Fourier transform of an aperiodic energy signal	111
6.4	The Fourier transform of a rectangular pulse	112
6.5	The Fourier transform of an impulse	114
6.6	Symmetry in Fourier transforms	114
6.7	Some physical insight into the evaluation of Fourier transforms	115
6.8	The inverse Fourier transform	117
6.9	Reconstruction of a waveform from its Fourier transform	118
6.10	Fourier transforms of power signals	119
6.11	Fourier transforms of periodic signals	119
6.12	Cyclic frequency versus radian frequency	122
6.13	Signal transmission through linear networks	123
6.14	The importance of the Fourier transform in network analysis	125
6.15	The energy density spectrum	126
6.16	Power spectral density (PSD)	128
6.17	The equivalence of convolution in one domain to multiplication in the other domain	131
6.18	Some other properties of Fourier transforms	135
6.18.1	Superposition	135
6.18.2	Duality	135
6.18.3	Compression and expansion	136
6.18.4	The effect of a time shift on the Fourier transform	136
6.18.5	The effect of a frequency shift on the inverse Fourier transform	138
6.18.6	Differentiation and integration	138
6.19	A selection of some important Fourier transforms	139
6.20	Summary	140
6.21	Problems	140
7.	The sampling and digitization of signals	142
7.1	Preview	142
7.2	Overview of sampling and digitization	142
7.3	Binary codes	145
7.4	Analogue to digital and digital to analogue converters	145
7.4.1	The 'flash' ADC	146
7.4.2	The weighted-resistor-network DAC	149
7.4.3	The $R-2R$ DAC	150
7.4.4	The successive-approximation ADC	152
7.5	Distortion due to sampling, and the sampling theorem	153
7.6	Ideal sampling and the Fourier transform approach to the spectra of sampled signals	157
7.7	Anti-aliasing filters	159
7.8	The effect of holding sample amplitudes	160
7.9	Dynamic range and quantization errors	162
7.10	An introduction to the sampling of bandpass signals	165
7.11	Summary	166
7.12	Problems	166
8.	The discrete Fourier transform	167
8.1	Preview	167
8.2	An introduction to basis functions	167

8.3	Alternative sets of basis functions and orthogonality	169
8.4	Trigonometric basis functions for discrete signals	172
8.5	Digital frequency	176
8.6	The inverse discrete Fourier transform	178
8.7	Notation and symbols	180
8.8	Evaluating the coefficients of complex exponential basis functions	181
8.9	The discrete Fourier transform	183
8.10	The discrete Fourier transform pair	185
8.11	Interpretation of the DFT	190
8.11.1	Differences in the definitions of the continuous and discrete Fourier transforms	190
8.11.2	The effect of sampling and windowing the waveform	191
8.11.3	The effect of sampling and windowing the spectrum	192
8.11.4	Summary of the relationship between the DFT and the continuous Fourier transform	192
8.11.5	Reordering of DFT coefficients	193
8.11.6	An alternative interpretation scheme	193
8.12	Summary	197
8.13	Problems	197
9.	The fast Fourier transform and some applications	199
9.1	Preview	199
9.2	Computational demands of the discrete Fourier transform	199
9.3	Computational demands of the fast Fourier transform	200
9.4	Spectral analysis	206
9.4.1	Window functions for improving spectral estimates	210
9.4.2	Zero-padding	216
9.5	Spectrograms	221
9.6	Linear convolution and periodic (or circular) convolution using the FFT	222
9.7	Summary	234
9.8	Problems	234
10.	The steady state response of analogue systems by consideration of the excitation e^{st}	235
10.1	Preview	235
10.2	Impedance and admittance when the excitation is e^{st}	236
10.3	Transfer functions	237
10.4	Further justification for the use of e^{st}	238
10.5	Pole–zero plots	239
10.6	Frequency response from the pole–zero plot	241
10.7	Frequency response using MATLAB	245
10.8	The order of a transfer function, some examples of second-order systems, and Q	247
10.9	Conjugate pole and conjugate zero pairs	254
10.10	Impedance functions for non-electrical components and systems	255
10.11	Transfer functions for electromechanical components and systems	256
10.12	Summary	258
10.13	Problems	258

11. Natural responses, transients and stability	259
11.1 Preview	259
11.2 Natural response	259
11.3 The natural response of first-order systems	260
11.3.1 The natural current response of a series RC network	260
11.3.2 The natural current response of a series RL network	262
11.4 The complete response	263
11.5 Transients	269
11.6 Relationship between the natural response and the system poles and zeros	269
11.7 The natural response of second-order systems	271
11.8 The natural response of hybrid electrical and mechanical systems	274
11.9 Stability	275
11.10 Summary	278
11.11 Problems	278
12. The Laplace transform	280
12.1 Preview	280
12.2 The bilateral Laplace transform	281
12.3 The unilateral Laplace transform	282
12.4 Convergence	284
12.5 A physical interpretation of the Laplace transform	285
12.6 Applications of the Laplace transform to system analysis	289
12.7 Justification of the Laplace transform and a comparison with the Fourier transform	291
12.8 s -plane plots and some more Laplace transforms	292
12.8.1 The step function	292
12.8.2 The exponentially decaying step function	293
12.8.3 The ramp function At	293
12.8.4 Sine and cosine functions	294
12.8.5 Functions modified by an exponential decay	294
12.8.6 Exponentially decaying sines and cosines	295
12.8.7 Delayed functions	296
12.9 Inverse Laplace transformation	296
12.10 Interpreting signal waveforms from pole–zero plots	299
12.11 The step function response of first-order systems	303
12.12 The step function response of second-order systems	305
12.13 System responses using MATLAB	309
12.14 An introduction to automatic control	311
12.15 Laplace transforms used directly for the solution of differential equations	314
12.16 Laplace transforms used directly for the solution of integro-differential equations	317
12.17 A comparison of Laplace transforms and phasors for system analysis	318
12.18 Summary	320
12.19 Problems	320
13. Synthesis of analogue filters	322
13.1 Preview	322
13.2 Basic approach to filter design	323

13.3	Amplitude and delay distortion	324
13.4	Selecting a frequency response	327
13.5	Transfer functions and frequency responses of analogue filters using MATLAB	329
13.6	Obtaining the transfer function corresponding to a desired frequency response	333
13.7	Translating a transfer function into a filter design	335
13.8	Design of passive lowpass filters using tables	336
13.9	Impedance scaling	337
13.10	Frequency scaling of lowpass filters	337
13.11	Lowpass to bandpass transformation	338
13.12	Lowpass to highpass transformation	343
13.13	Passive bandstop filters	344
13.14	Active lowpass filters	344
13.15	Active highpass filters	346
13.16	Active bandpass filters	346
13.17	Delay equalizers	348
13.18	Summary	350
13.19	Problems	350
14.	An introduction to digital networks and the z transform	351
14.1	Preview	351
14.2	Structure of digital networks and the difference equation	352
14.3	The impulse responses of digital networks – FIR and IIR systems	355
14.4	Other configurations of digital networks	357
14.5	The hardware realization of digital networks	359
14.6	An s -plane treatment of digital networks	360
14.7	A z -plane treatment of digital networks	363
14.8	Mapping between the s -plane and the z -plane	364
14.9	The frequency response of a digital network from a geometrical interpretation of z -plane poles and zeros	365
14.10	An analytical approach to the frequency response of a digital network	369
14.11	Frequency responses and pole–zero plots using MATLAB	372
14.12	The z transform	373
14.13	Some important z transforms	376
14.13.1	The unit step function	376
14.13.2	The unit ramp	377
14.13.3	The decaying exponential	377
14.13.4	The damped cosinusoid	377
14.13.5	The damped sinusoid	378
14.14	A direct derivation of the z transfer function	379
14.15	The inverse z transform	379
14.16	Discrete-time convolution using the z transform	383
14.17	The responses of digital systems using MATLAB	384
14.18	Transfer functions of sampled analogue systems	388
14.19	An introduction to digital control systems	391
14.20	The relationship between the z transform, the Fourier transform and the DFT	394

14.21	Summary	397
14.22	Problems	397
15	Synthesis of digital filters	399
15.1	Preview	399
15.2	Filter design by pole–zero placement	399
15.3	Design of recursive digital filters based on the frequency response characteristics of analogue filters	402
15.3.1	The impulse invariant technique	403
15.3.2	The bilinear transformation technique	407
15.3.3	Highpass, bandpass and bandstop filters	411
15.4	Designing recursive digital filters with MATLAB	412
15.5	Effect of coefficient quantization in IIR filters	416
15.6	Cascading low-order IIR filters for improved stability	420
15.7	Design of non-recursive digital filters	424
15.7.1	The Fourier (or window) method	426
15.7.2	The design of optimal linear phase FIR filters	434
15.8	Summary	436
15.9	Problems	436
16.	Correlation	438
16.1	Preview	438
16.2	Covariance and the correlation coefficient	438
16.3	The cross-correlation function for finite-duration signals	441
16.4	Correlation using MATLAB	445
16.5	The cross-correlation function for power signals	447
16.6	Autocorrelation functions	449
16.7	Some applications of cross-correlation and autocorrelation	454
16.7.1	Determination of time delays	454
16.7.2	System identification	455
16.7.3	Pattern recognition	456
16.7.4	Detection of signals in noise by cross-correlation	457
16.7.5	Detection of signals in noise by autocorrelation	459
16.8	The chirp pulse	460
16.9	Time domain correlators and matched filters	463
16.10	Frequency domain correlation	465
16.11	The output waveform from a matched filter, and the processing gain	468
16.12	Doppler effects in correlators	470
16.13	Summary	472
16.14	Problems	473
17.	Processing techniques for bandpass signals	474
17.1	Preview	474
17.2	Bandpass signals	474
17.3	Baseband representations of bandpass signals and the complex envelope	475
17.4	Generation of in-phase and quadrature components	478

17.5	Envelope detection using quadrature components	479
17.6	An example of quadrature channels for measuring Doppler shifts in radar	483
17.7	An example of quadrature channels in narrowband FFT beamforming	486
17.8	Filtering bandpass signals with lowpass filters	487
17.9	Correlators for bandpass signals	491
17.10	The physical reality of complex envelopes	494
17.11	Analytic signals	495
17.12	Summary	496
17.13	Problems	496
Appendix A Listings of selected MATLAB programs		498
Bibliography		504
Index		505

Trademark notice

The following are trademarks or registered trademarks of their respective companies:

Macintosh	Apple Computer, Inc.
MATLAB	The Mathworks, Inc.
Windows	Microsoft Corporation

Getting started in MATLAB and an introduction to systems and signal processing

1.1 ● Preview

This introductory chapter begins by presenting some of the main commands and capabilities of the powerful and widely used scientific software sold as MATLAB[®]. It is a very condensed version of what is described more fully in the manuals that accompany the student editions of MATLAB, constraining itself to the very minimum that is needed for getting started with its use in this book.

The chapter continues by classifying various types of signal and defining the linear time-invariant systems that form the subject of the book. Some simple examples of analogue and digital processing are given and reasons are presented for why digital processing is increasingly the preferred option. Finally, the special importance of sinusoidal, impulse function and step function signals is justified.

- 1.2 ● Getting started in MATLAB
- 1.3 ● Some signals and systems terminology
- 1.4 ● Some examples of analogue systems and analogue signal processing

- 1.5 ● Some examples of digital systems and digital signal processing
- 1.6 ● Justification for the digital processing signals
- 1.7 ● Some signals of special importance
- 1.8 ● Summary
- 1.9 ● Problems

1.2 ● Getting started in MATLAB

MATLAB has become an invaluable and widely accepted software tool whose existence has affected signal processing design procedures; as such, an acquaintance with its power has become an important part of a signal processing syllabus. This section is intended as a very brief tutorial on some of its most important features. A particularly useful feature of MATLAB is that student editions are available from scientific bookshops for no more than the price of a typical hardback textbook, and yet they retain many of the most important capabilities of the full version. The

relevant student edition for use with Microsoft Windows 3.1 is Version 4 and is available as a diskette plus manual (MATLAB 1995). This can also be used on Microsoft Windows 95 and Windows NT, but the updated student edition for these is Version 5, and this is available as a CD-ROM plus manual (MATLAB 1997). Versions 4 and 5 are also available for the appropriate Macintosh computers. Although fully compatible with the professional versions of MATLAB which may well be available to students through a copy on their university network, *all the MATLAB commands in this book are restricted to commands that are included in both of these student editions.*

Many sections of this book use MATLAB to demonstrate signal processing, and at these times the code will often be included in the main text. This is partly to demonstrate the power of the software and partly to provide further guidance in its use. In other sections MATLAB is used solely for producing figures, and on these occasions the code is usually omitted. Sometimes however, when there is something new to be learnt about the use of MATLAB for visualization, the figure is marked with an asterisk and the code is included in Appendix A.

The purpose of this present section is to describe some of the most basic features of MATLAB code so that the listings included later in the book will not need too much explanation to be understood.

1.2.1 Variables

When a number or a sequence of numbers is given a name it is termed a variable. A variable name can contain up to 19 characters, excluding punctuation characters. The first character must be a letter, but the remainder can be letters, digits or underscores and MATLAB distinguishes between lower case and upper case letters. An example of allocating a specific value to a variable is given by the command

```
b = 7
```

This causes the variable *b* to have the value 7. After entering this command the following appears on the screen

```
b =  
    7
```

If it is wished to suppress such a message one follows the command with a semicolon. For example

```
b = 7;
```

results in no such message on the screen.

A calculation can be involved when defining a variable. For example, the command

```
my_fingers = 2*5;
```

introduces a variable *my_fingers* that has the value 10.

A few variables are built into MATLAB, namely *pi*, *i*, *j*, *ans*, *eps*, *inf*, *NaN*, *realmin* and *realmax*. Only *pi*, *j* and *ans* are used in this text. The variable *pi* is π . The variable *j* (which is the same as *i*) is $\sqrt{-1}$. The variable *ans* is a default variable name used for

results. If, for example, we use MATLAB as a simple calculator by entering

```
4*5
```

we see

```
ans =  
20
```

1.2.2 Sequences, arrays and vectors

The previous subsection considered the situation where the variable had a single numerical value (it can be termed a *scalar* variable). Very often the variable is a *sequence* of numbers and one example of a four-element sequence might be noon temperatures over a four-day period. A set of one or more sequences can be termed an *array* and an example would be the daily noon temperatures over a four-day period at three locations, say London, New York and Tokyo. A hypothetical set of such temperatures could be entered as a MATLAB variable by encasing the values in square brackets separated by spaces and with the different number sequences separated by semicolons. Thus the command

```
x = [12.3 13.6 14.5 16.2; 18.1 18.8 19.2 18.4; 20.9 21.7 19.0 19.4]
```

results in the display

```
x =  
  
12.3000 13.6000 14.5000 16.2000  
18.1000 18.8000 19.2000 18.4000  
20.9000 21.7000 19.0000 19.4000
```

where x is an array in which the top row represents the London data, the second row the New York data, and the third row the Tokyo data. This array can also be thought of as a *matrix* of numbers. In this example it is a 3×4 matrix (3 rows by 4 columns). Generalizing this, an $m \times n$ matrix A is an array of numbers given by

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

This matrix A contains m rows each of n elements. Alternatively it can be considered as containing n columns each of m elements.

When the sequence is a one-dimensional array of numbers it is commonly termed a *vector*. In the example of temperatures given by the x array variable above, each set of numbers representing the temperatures at a single location is a *row vector*. If on the other hand we consider the temperatures at all three locations, but on one specific day, we would have a *column vector*. This book will deal mainly with vectors, and particularly with row vectors.