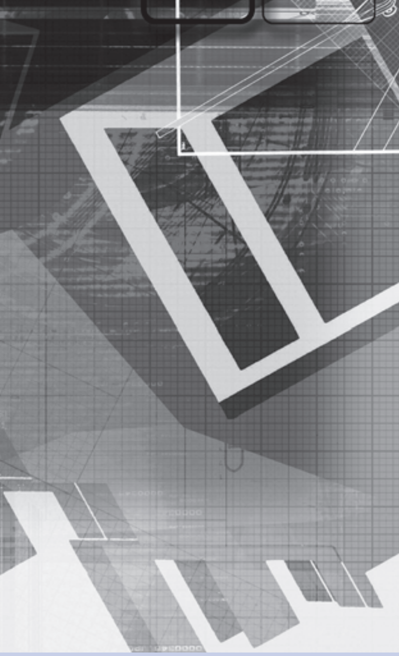
ADVANCED MODERN ENGINEERING MATHEMATICS

Fifth Edition

Glyn James & Phil Dyke

Pearson



Advanced Modern Engineering Mathematics

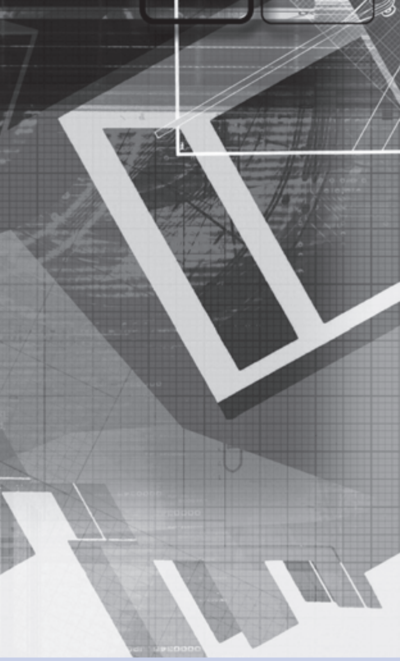
Fifth Edition

Pearson

We work with leading authors to develop the strongest educational materials in mathematics, bringing cutting-edge thinking and best learning practice to a global market.

Under a range of well-known imprints, including Prentice Hall, we craft high-quality print and electronic publications which help readers to understand and apply their content, whether studying or at work.

To find out more about the complete range of our publishing, please visit us on the World Wide Web at: www.pearsoned.co.uk



Advanced Modern Engineering Mathematics

Fifth Edition

Glyn James *Coventry University*
Phil Dyke *University of Plymouth*
and
David Burley *University of Sheffield*
Dick Clements *University of Bristol*
Matthew Craven *University of Plymouth*
Tim Reis *University of Greenwich*
John Searl *University of Edinburgh*
Julian Stander *University of Plymouth*
Nigel Steele *Coventry University*
Jerry Wright *AT&T*

Pearson

Harlow, England • London • New York • Boston • San Francisco • Toronto • Sydney • Dubai • Singapore • Hong Kong
Tokyo • Seoul • Taipei • New Delhi • Cape Town • São Paulo • Mexico City • Madrid • Amsterdam • Munich • Paris • Milan

Pearson Education Limited
KAO Two
KAO Park
Harlow CM17 9NA
United Kingdom
Tel: +44 (0)1279 623623
Web: www.pearson.com/uk

First published 1993 (print)
Second edition published 1999 (print)
Third edition published 2004 (print)
Fourth edition published 2011 (print)
Fifth edition published 2018 (print and electronic)

ISBN: 978-1-292-17434-1 (print)
978-1-292-17582-9 (PDF)
978-1-292-17583-6 (ePub)

British Library Cataloguing-in-Publication Data
A catalogue record for the print edition is available from the British Library

Library of Congress Cataloging-in-Publication Data

Names: James, Glyn, author.
Title: Advanced modern engineering mathematics / Glyn James (Coventry University) [and nine others].
Description: Fifth edition. | Harlow, United Kingdom : Pearson Education, 2018. | Includes index.
Identifiers: LCCN 2018008839| ISBN 9781292174341 (print) | ISBN 9781292175829 (pdf) | ISBN 9781292175836 (epub)
Subjects: LCSH: Engineering mathematics
Classification: LCC TA330 .A38 2018 | DDC 620.001/51--dc23
LC record available at https://urldefense.proofpoint.com/v2/url?u=https-3A__lccn.loc.gov_2018008839&d=DwIFAg&c=0YLnzTkWOdJlub_y7qAx8Q&r=eK0q0-QqUPIJD1OLTc7YiWdHxmNowNBMcvK9N3XeA-U&m=4G2pfE4jTE5-LBbM4fpokbAjBOaA-h39xC4FZ8wxPIU&s=fH6box9GavoU_1p97XMA38XG0TLT-fWxCM6hQneaghGY&e=

10 9 8 7 6 5 4 3 2 1
22 21 20 19 18

Print edition typeset in 10/12pt Times LT Std by Spi Global
Printed and bound in Malaysia
NOTE THAT ANY PAGE CROSS REFERENCES REFER TO THE PRINT EDITION

Contents

Preface xix
About the Authors xxi
Publisher's Acknowledgements xxiii

Chapter 1 Matrix Analysis 1

1.1 Introduction 2

1.2 Review of matrix algebra 2

1.2.1 Definitions 3
1.2.2 Basic operations on matrices 3
1.2.3 Determinants 5
1.2.4 Adjoint and inverse matrices 5
1.2.5 Linear equations 7
1.2.6 Rank of a matrix 8

1.3 Vector spaces 9

1.3.1 Linear independence 10
1.3.2 Transformations between bases 11
1.3.3 Exercises (1–4) 13

1.4 The eigenvalue problem 13

1.4.1 The characteristic equation 14
1.4.2 Eigenvalues and eigenvectors 16
1.4.3 Exercises (5–6) 22
1.4.4 Repeated eigenvalues 22
1.4.5 Exercises (7–9) 26
1.4.6 Some useful properties of eigenvalues 26
1.4.7 Symmetric matrices 28
1.4.8 Exercises (10–13) 29

1.5 Numerical methods 29

1.5.1 The power method 29
1.5.2 Exercises (14–18) 35

1.6 Reduction to canonical form 36

1.6.1 Reduction to diagonal form 36
1.6.2 The Jordan canonical form 39
1.6.3 Exercises (19–26) 43
1.6.4 Quadratic forms 44
1.6.5 Exercises (27–33) 50

1.7 Functions of a matrix 51

1.7.1 Exercises (34–41) 62

1.8 Singular value decomposition 63

1.8.1 Singular values 65
1.8.2 Singular value decomposition (SVD) 69
1.8.3 Pseudo inverse 72
1.8.4 Exercises (42–49) 78

1.9 State-space representation 79

1.9.1 Single-input–single-output (SISO) systems 79
1.9.2 Multi-input–multi-output (MIMO) systems 84
1.9.3 Exercises (50–54) 85

1.10 Solution of the state equation 86

1.10.1 Direct form of the solution 86
1.10.2 The transition matrix 88
1.10.3 Evaluating the transition matrix 89
1.10.4 Exercises (55–60) 91
1.10.5 Spectral representation of response 92
1.10.6 Canonical representation 95
1.10.7 Exercises (61–67) 100

1.11 Engineering application: Lyapunov stability analysis 101

1.11.1 Exercises (68–72) 103

1.12 Engineering application: capacitor microphone 104

1.13 Review exercises (1–19) 108

Chapter 2 Numerical Solution of Ordinary Differential Equations 113

2.1 Introduction 114

2.2 Engineering application: motion in a viscous fluid 114

2.3 Numerical solution of first-order ordinary differential equations 115

2.3.1 A simple solution method: Euler's method 116
2.3.2 Analysing Euler's method 120
2.3.3 Using numerical methods to solve engineering problems 123
2.3.4 Exercises (1–7) 125
2.3.5 More accurate solution methods: multistep methods 126
2.3.6 Local and global truncation errors 132
2.3.7 More accurate solution methods: predictor–corrector methods 134
2.3.8 More accurate solution methods: Runge–Kutta methods 139
2.3.9 Exercises (8–17) 143
2.3.10 Stiff equations 145
2.3.11 Computer software libraries 147

2.4 Numerical methods for systems of ordinary differential equations and higher-order differential equations 149

2.4.1 Numerical solution of coupled first-order equations 149
2.4.2 State-space representation of higher-order systems 154
2.4.3 Exercises (18–23) 158
2.4.4 Boundary-value problems 159
2.4.5 The method of shooting 160

2.5 Engineering application: oscillations of a pendulum 162

2.6 Engineering application: heating of an electrical fuse 167

2.7 Review exercises (1–12) 172

Chapter 3 Vector Calculus 175

3.1 Introduction 176

3.1.1 Basic concepts 177
3.1.2 Exercises (1–10) 184
3.1.3 Transformations 185

3.1.4 Exercises (11–17) 188
3.1.5 The total differential 188
3.1.6 Exercises (18–20) 192

3.2 Derivatives of a scalar point function 192

3.2.1 The gradient of a scalar point function 192
3.2.2 Exercises (21–30) 195

3.3 Derivatives of a vector point function 196

3.3.1 Divergence of a vector field 196
3.3.2 Exercises (31–37) 198
3.3.3 Curl of a vector field 199
3.3.4 Exercises (38–45) 202
3.3.5 Further properties of the vector operator ∇ 202
3.3.6 Exercises (46–55) 206

3.4 Topics in integration 206

3.4.1 Line integrals 207
3.4.2 Exercises (56–64) 210
3.4.3 Double integrals 211
3.4.4 Exercises (65–76) 216
3.4.5 Green's theorem in a plane 217
3.4.6 Exercises (77–82) 221
3.4.7 Surface integrals 222
3.4.8 Exercises (83–91) 229
3.4.9 Volume integrals 229
3.4.10 Exercises (92–102) 232
3.4.11 Gauss's divergence theorem 233
3.4.12 Stokes' theorem 236
3.4.13 Exercises (103–112) 239

3.5 Engineering application: streamlines in fluid dynamics 240

3.6 Engineering application: heat transfer 242

3.7 Review exercises (1–21) 246

Chapter 4 Functions of a Complex Variable 249

4.1 Introduction 250

4.2 Complex functions and mappings 251

4.2.1 Linear mappings 253
4.2.2 Exercises (1–8) 260

4.2.3 Inversion 260
4.2.4 Bilinear mappings 265
4.2.5 Exercises (9–19) 271
4.2.6 The mapping $w = z^2$ 272
4.2.7 Exercises (20–23) 274

4.3 Complex differentiation 274

4.3.1 Cauchy–Riemann equations 275
4.3.2 Conjugate and harmonic functions 280
4.3.3 Exercises (24–32) 282
4.3.4 Mappings revisited 282
4.3.5 Exercises (33–37) 286

4.4 Complex series 287

4.4.1 Power series 287
4.4.2 Exercises (38–39) 291
4.4.3 Taylor series 291
4.4.4 Exercises (40–43) 294
4.4.5 Laurent series 295
4.4.6 Exercises (44–46) 300

4.5 Singularities and zeros 300

4.5.1 Exercises (47–49) 303

4.6 Engineering application: analysing AC circuits 304

4.7 Engineering application: use of harmonic functions 305

4.7.1 A heat transfer problem 305
4.7.2 Current in a field-effect transistor 307
4.7.3 Exercises (50–56) 310

4.8 Review exercises (1–19) 311

Chapter 5 Laplace Transforms 315

5.1 Introduction 316

5.1.1 Definition and notation 316
5.1.2 Other results from MEM 318

5.2 Step and impulse functions 320

5.2.1 The Heaviside step function 320
5.2.2 Laplace transform of unit step function 323
5.2.3 The second shift theorem 325

5.2.4 Inversion using the second shift theorem 328
5.2.5 Differential equations 331
5.2.6 Periodic functions 335
5.2.7 Exercises (1–12) 339
5.2.8 The impulse function 341
5.2.9 The sifting property 342
5.2.10 Laplace transforms of impulse functions 343
5.2.11 Relationship between Heaviside step and impulse functions 346
5.2.12 Exercises (13–18) 351
5.2.13 Bending of beams 352
5.2.14 Exercises (19–21) 356

5.3 Transfer functions 356

5.3.1 Definitions 356
5.3.2 Stability 359
5.3.3 Impulse response 364
5.3.4 Initial- and final-value theorems 365
5.3.5 Exercises (22–33) 370
5.3.6 Convolution 371
5.3.7 System response to an arbitrary input 374
5.3.8 Exercises (34–38) 378

5.4 Solution of state-space equations 378

5.4.1 SISO systems 378
5.4.2 Exercises (39–47) 382
5.4.3 MIMO systems 383
5.4.4 Exercises (48–50) 390

5.5 Engineering application: frequency response 390

5.6 Engineering application: pole placement 398

5.6.1 Poles and eigenvalues 398
5.6.2 The pole placement or eigenvalue location technique 398
5.6.3 Exercises (51–56) 400

5.7 Review exercises (1–18) 401

Chapter 6 The *z* Transform 407

6.1 Introduction 408

6.2 The *z* transform 409

6.2.1 Definition and notation 409
6.2.2 Sampling: a first introduction 413
6.2.3 Exercises (1–2) 414

6.3 Properties of the z transform 414

6.3.1 The linearity property 415
6.3.2 The first shift property (delaying) 416
6.3.3 The second shift property (advancing) 417
6.3.4 Some further properties 418
6.3.5 Table of z transforms 419
6.3.6 Exercises (3–10) 420

6.4 The inverse z transform 420

6.4.1 Inverse techniques 421
6.4.2 Exercises (11–13) 427

6.5 Discrete-time systems and difference equations 428

6.5.1 Difference equations 428
6.5.2 The solution of difference equations 430
6.5.3 Exercises (14–20) 434

6.6 Discrete linear systems: characterization 435

6.6.1 z transfer functions 435
6.6.2 The impulse response 441
6.6.3 Stability 444
6.6.4 Convolution 450
6.6.5 Exercises (21–29) 454

6.7 The relationship between Laplace and z transforms 455

6.8 Solution of discrete-time state-space equations 456

6.8.1 State-space model 456
6.8.2 Solution of the discrete-time state equation 459
6.8.3 Exercises (30–33) 463

6.9 Discretization of continuous-time state-space models 464

6.9.1 Euler's method 464
6.9.2 Step-invariant method 466
6.9.3 Exercises (34–37) 469

6.10 Engineering application: design of discrete-time systems 470

6.10.1 Analogue filters 471
6.10.2 Designing a digital replacement filter 472
6.10.3 Possible developments 473

6.11	Engineering application: the delta operator and the $\mathscr{D}$ transform	473
	6.11.1 Introduction	473
	6.11.2 The q or shift operator and the δ operator	474
	6.11.3 Constructing a discrete-time system model	475
	6.11.4 Implementing the design	477
	6.11.5 The $\mathscr{D}$ transform	479
	6.11.6 Exercises (38–41)	480
6.12	Review exercises (1–18)	480
Chapter 7	**Fourier Series**	**485**
7.1	Introduction	486
	7.1.1 Periodic functions	486
	7.1.2 Fourier's theorem	487
	7.1.3 Functions of period 2π	488
	7.1.4 Functions defined over a finite interval	492
	7.1.5 Exercises (1–10)	498
7.2	Fourier series of jumps at discontinuities	499
	7.2.1 Exercises (11–12)	502
7.3	Engineering application: frequency response and oscillating systems	502
	7.3.1 Response to periodic input	502
	7.3.2 Exercises (13–16)	507
7.4	Complex form of Fourier series	508
	7.4.1 Complex representation	508
	7.4.2 The multiplication theorem and Parseval's theorem	512
	7.4.3 Discrete frequency spectra	515
	7.4.4 Power spectrum	521
	7.4.5 Exercises (17–22)	523
7.5	Orthogonal functions	524
	7.5.1 Definitions	524
	7.5.2 Generalized Fourier series	526
	7.5.3 Convergence of generalized Fourier series	527
	7.5.4 Exercises (23–29)	529

7.6	Engineering application: describing functions	532
7.7	Review exercises (1–20)	533

Chapter 8	**The Fourier Transform**	**537**
8.1	Introduction	538
8.2	The Fourier transform	539
	8.2.1 The Fourier integral	539
	8.2.2 The Fourier transform pair	544
	8.2.3 The continuous Fourier spectra	548
	8.2.4 Exercises (1–10)	551
8.3	Properties of the Fourier transform	552
	8.3.1 The linearity property	552
	8.3.2 Time-differentiation property	552
	8.3.3 Time-shift property	553
	8.3.4 Frequency-shift property	554
	8.3.5 The symmetry property	555
	8.3.6 Exercises (11–16)	557
8.4	The frequency response	558
	8.4.1 Relationship between Fourier and Laplace transforms	558
	8.4.2 The frequency response	560
	8.4.3 Exercises (17–21)	563
8.5	Transforms of the step and impulse functions	563
	8.5.1 Energy and power	563
	8.5.2 Convolution	572
	8.5.3 Exercises (22–27)	574
8.6	The Fourier transform in discrete time	575
	8.6.1 Introduction	575
	8.6.2 A Fourier transform for sequences	575
	8.6.3 The discrete Fourier transform	579
	8.6.4 Estimation of the continuous Fourier transform	583
	8.6.5 The fast Fourier transform	592
	8.6.6 Exercises (28–31)	599

8.7 Engineering application: the design of analogue filters 599

8.8 Engineering application: direct design of digital filters and windows 602

8.8.1 Digital filters 602
8.8.2 Windows 607
8.8.3 Exercises (32–33) 611

8.9 Review exercises (1–25) 611

Chapter 9 Partial Differential Equations 615

9.1 Introduction 616

9.2 General discussion 617

9.2.1 Wave equation 617
9.2.2 Heat-conduction or diffusion equation 620
9.2.3 Laplace equation 623
9.2.4 Other and related equations 625
9.2.5 Arbitrary functions and first-order equations 627
9.2.6 Exercises (1–14) 632

9.3 Solution of the wave equation 634

9.3.1 D'Alembert solution and characteristics 634
9.3.2 Separation of variables 643
9.3.3 Laplace transform solution 648
9.3.4 Exercises (15–27) 651
9.3.5 Numerical solution 653
9.3.6 Exercises (28–31) 659

9.4 Solution of the heat-conduction/diffusion equation 660

9.4.1 Separation of variables 660
9.4.2 Laplace transform method 664
9.4.3 Exercises (32–40) 669
9.4.4 Numerical solution 671
9.4.5 Exercises (41–43) 677

9.5 Solution of the Laplace equation 677

9.5.1 Separation of variables 677
9.5.2 Exercises (44–54) 685
9.5.3 Numerical solution 686
9.5.4 Exercises (55–59) 693

9.6	Finite elements	694
	9.6.1 Exercises (60–62)	706
9.7	Integral solutions	707
	9.7.1 Separation of variables	707
	9.7.2 Use of singular solutions	709
	9.7.3 Sources and sinks for the heat-conduction equation	712
	9.7.4 Exercises (63–67)	715
9.8	General considerations	716
	9.8.1 Formal classification	716
	9.8.2 Boundary conditions	718
	9.8.3 Exercises (68–74)	723
9.9	Engineering application: wave propagation under a moving load	723
9.10	Engineering application: blood-flow model	726
9.11	Review exercises (1–21)	730
Chapter 10	**Optimization**	**735**
10.1	Introduction	736
10.2	Linear programming	739
	10.2.1 Introduction	739
	10.2.2 Simplex algorithm: an example	741
	10.2.3 Simplex algorithm: general theory	745
	10.2.4 Exercises (1–11)	752
	10.2.5 Two-phase method	753
	10.2.6 Equality constraints and variables that are unrestricted in sign	761
	10.2.7 Exercises (12–20)	762
10.3	Lagrange multipliers	764
	10.3.1 Equality constraints	764
	10.3.2 Inequality constraints	768
	10.3.3 Exercises (21–28)	768
10.4	Hill climbing	769
	10.4.1 Single-variable search	769
	10.4.2 Exercises (29–34)	775

10.4.3 Simple multivariable searches: steepest ascent and Newton's method 775
10.4.4 Exercises (35–39) 781
10.4.5 Advanced multivariable searches 782
10.4.6 Least squares 786
10.4.7 Exercises (40–43) 789

10.5 Engineering application: chemical processing plant 790

10.6 Engineering application: heating fin 792

10.7 Review exercises (1–26) 795

Chapter 11 Applied Probability and Statistics 799

11.1 Introduction 800

11.2 Review of basic probability theory 801

11.2.1 The rules of probability 801
11.2.2 Random variables 802
11.2.3 The Bernoulli, binomial and Poisson distributions 804
11.2.4 The normal distribution 805
11.2.5 Sample measures 808

11.3 Estimating parameters 810

11.3.1 Interval estimates and hypothesis tests 810
11.3.2 Distribution of the sample average 810
11.3.3 Confidence interval for the mean 812
11.3.4 Testing simple hypotheses 815
11.3.5 Other confidence intervals and tests concerning means 817
11.3.6 Interval and test for proportion 821
11.3.7 Exercises (1–13) 824

11.4 Joint distributions and correlation 825

11.4.1 Joint and marginal distributions 825
11.4.2 Independence 828
11.4.3 Covariance and correlation 829
11.4.4 Sample correlation 833
11.4.5 Interval and test for correlation 835
11.4.6 Rank correlation 838
11.4.7 Exercises (14–24) 840

11.5 Regression 841

11.5.1 The method of least squares 842
11.5.2 Residuals 852
11.5.3 Regression and correlation 856
11.5.4 Nonlinear regression 856
11.5.5 Exercises (25–33) 861

11.6 Goodness-of-fit tests 863

11.6.1 Chi-square distribution and test 863
11.6.2 Contingency tables 867
11.6.3 Exercises (34–42) 873

11.7 Engineering application: analysis of engine performance data 874

11.7.1 Introduction 874
11.7.2 Difference in mean running times and temperatures 877
11.7.3 Dependence of running time on temperature 880
11.7.4 Test for normality 888
11.7.5 Conclusions 890

11.8 Engineering application: statistical quality control 891

11.8.1 Introduction 891
11.8.2 Shewhart attribute control charts 891
11.8.3 Shewhart variable control charts 894
11.8.4 Cusum control charts 898
11.8.5 Moving-average control charts 901
11.8.6 Range charts 905
11.8.7 Exercises (43–54) 907

11.9 Poisson processes and the theory of queues 908

11.9.1 Typical queueing problems 909
11.9.2 Poisson processes 909
11.9.3 Single service channel queue 916
11.9.4 Queues with multiple service channels 921
11.9.5 Queueing system simulation 923
11.9.6 Exercises (55–62) 929

11.10 Bayes' theorem and its applications 930

11.10.1 Derivation and simple examples 930
11.10.2 Applications in probabilistic inference 933
11.10.3 Bayesian statistical inference 935
11.10.4 Exercises (63–74) 944

11.11 Review exercises (1–10) 945

Answers to Exercises 949

Index 975

Preface

The first edition of this book appeared in 1993, and it could be assumed, wrongly, that its time has passed as 24 years have now elapsed. It is true that all the original authors apart from myself have retired but, in the intervening years the text has been regularly updated and we have now reached the fifth edition. The words of my colleague and predecessor as editor, Professor Glyn James, still ring true. Here is an excerpt from his preface to the fourth edition (2011):

> Throughout the course of history, engineering and mathematics have developed in parallel. All branches of engineering depend on mathematics for their description and there has been a steady flow of ideas and problems from engineering that has stimulated and sometimes initiated branches of mathematics. Thus, it is vital that engineering students receive a thorough grounding in mathematics, with the treatment related to their interests and problems. As with the previous editions, this has been the motivation for the production of this latest edition – a companion text to the fifth edition of *Modern Engineering Mathematics*, this being designed to provide a first-level core studies course in mathematics for undergraduate programmes in all engineering disciplines. Building on the foundations laid in the companion text, this book gives an extensive treatment of some of the more advanced areas of mathematics that have applications in various fields of engineering, particularly as tools for computer-based system modelling, analysis and design. Feedback, from users of the previous editions, on subject content has been highly positive indicating that it is sufficiently broad to provide the necessary second-level, or optional, studies for most engineering programmes, where in each case a selection of the material may be made. Whilst designed primarily for use by engineering students, it is believed that the book is also suitable for use by students of applied mathematics and the physical sciences.
>
> Although the pace of the book is at a somewhat more advanced level than the companion text, the philosophy of learning by doing is retained with continuing emphasis on the development of students' ability to use mathematics with understanding to solve engineering problems. Recognizing the increasing importance of mathematical modelling in engineering practice, many of the worked examples and exercises incorporate mathematical models that are designed both to provide relevance and to reinforce the role of mathematics in various branches of engineering. In addition, each chapter contains specific sections on engineering applications, and these form an ideal framework for individual, or group, study assignments, thereby helping to reinforce the skills of mathematical modelling, which are seen as essential if engineers are to tackle the increasingly complex systems they are being called upon to analyse and design. The importance of numerical methods in problem solving is also recognized, and its treatment is integrated with the analytical work throughout the book.

The position of software use is an important aspect of engineering education. The decision has been taken to use mainly MATLAB but also MAPLE. Students are encouraged to make intelligent use of software and, where appropriate, codes are included, but there is a health warning. The pace of technology shows little signs of lessening, and so in the space of six years, the likely time lapse before a new edition of this text, it is probable that software will continue to be updated, probably annually. There is therefore a real risk that much coding though correct and working at the time of publication could be broken by these updates. Therefore, in this edition the decision has been made not to over-emphasise specific code but to direct students to the companion website or to general principles instead. The software packages, particularly MAPLE, have become easier to use without the need for programming skills. Much is menu driven these days. Here's more from Glyn on the subject that is still true:

> Much of the feedback from users relates to the role and use of software packages, particularly symbolic algebra packages. Without making it an essential requirement the authors have attempted to highlight throughout the text situations where the user could make effective use of software. This also applies to exercises and, indeed, a limited number have been introduced for which the use of such a package is essential. Whilst any appropriate piece of software can be used, the authors recommend the use of MATLAB and/or MAPLE. In this edition reference to the use of these two packages is made throughout the text, with commands or codes introduced and illustrated. When indicated, students are strongly recommended to use these packages to check their solutions to exercises. This is not only to help develop proficiency in their use, but also to enable students to appreciate the necessity of having a sound knowledge of the underpinning mathematics if such packages are to be used effectively. Throughout the book two icons are used:
>
> - An open screen indicates that the use of a software package would be useful (e.g. for checking solutions) but not essential.
> - A closed screen indicates that the use of a software package is essential or highly desirable.

Specific changes in this fifth edition are an improvement in many of the diagrams, taking advantage of present day software, and modernization of the examples and language. Also, the chapter on Applied Probability and Statistics has been significantly modernized by interfacing the presentation with the very powerful software package R. Simply search for 'R Software' and it is a free download. I have been much aided in getting this edition ready for publication by my hardworking colleagues Matthew, Tim and Julian who have joined the editorial team.

Acknowledgements

The authoring team is extremely grateful to all the reviewers and users of the text who have provided valuable comments on previous editions of this book. Most of this has been highly constructive and very much appreciated. The team has continued to enjoy the full support of a very enthusiastic production team at Pearson Education and wishes to thank all those concerned.

Phil Dyke and Glyn James

About the Authors

A new set of authors, Matthew Craven, Tim Reis and Julian Stander under the new editor, one of the original authors, Phil Dyke, have taken on the task of producing this, the fifth edition of *Advanced Modern Engineering Mathematics*.

Phil Dyke is Professor of Applied Mathematics at the University of Plymouth. He was a Head of School for 22 years, 18 of these as Head of Mathematics and Statistics. He has over 45 years teaching and research experience in higher education, much of this teaching engineering students not only mathematics but also marine and coastal engineering. Apart from his contribution to both *Modern Engineering Mathematics* and *Advanced Modern Engineering Mathematics* he is the author of 11 other textbooks ranging in topic from mathematical methods to mechanics and marine physics. He is now semi-retired, but still teaches and writes.

Matthew Craven is a Lecturer in Applied Mathematics at the University of Plymouth. For fifteen years, he has taught foundation year, postgraduate and everything in between. He has research interests in computational simulation, operational research, high performance computing and algebraic systems.

Tim Reis is a Senior Lecturer in Mathematics at the University of Greenwich. He is an applied mathematician with interests in fluid dynamics, numerical analysis, and mathematical modelling. His doctoral thesis was awarded the Vernon Harrison prize by the British Society of Rheology and he continues to conduct research into the modelling and simulation of complex flows. Tim teaches a range of mathematical subjects at undergraduate and postgraduate level and he is actively involved in promoting mathematics to the wider community. He is a fellow of the Institute of Mathematics and its Applications (IMA).

Julian Stander is Associate Professor (Reader) in Mathematics and Statistics at the University of Plymouth. He has over twenty years' cross-disciplinary teaching experience in mathematics and statistics at both the undergraduate and postgraduate level. His research and consultancy interests are in computational statistics, data science, extreme value and dependence modelling, and social media information extraction.

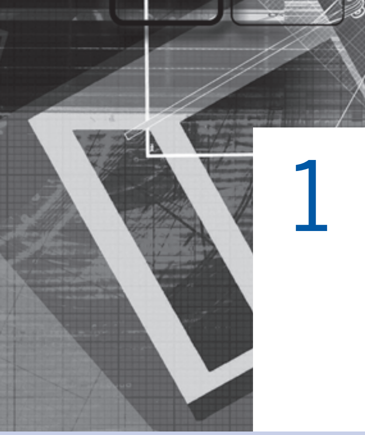
The original editor is **Glyn James** who retired as Dean of the School of Mathematical and Information Sciences at Coventry University in 2001 and is now Emeritus Professor in Mathematics at the University. He graduated from the University College of Wales, Cardiff in the late 1950s, obtaining first class honours degrees in both Mathematics and

Chemistry. He obtained a PhD in Engineering Science in 1971 as an external student of the University of Warwick. He has been employed at Coventry since 1964 and held the position of the Head of Mathematics Department prior to his appointment as Dean in 1992. His research interests are in control theory and its applications to industrial problems. He also has a keen interest in mathematical education, particularly in relation to the teaching of engineering mathematics and mathematical modelling. He was co-chairman of the European Mathematics Working Group established by the European Society for Engineering Education (SEFI) in 1982, a past chairman of the Education Committee of the Institute of Mathematics and its Applications (IMA), and a member of the Royal Society Mathematics Education Subcommittee. In 1995 he was chairman of the Working Group that produced the report 'Mathematics Matters in Engineering' on behalf of the professional bodies in engineering and mathematics within the UK. He is also a member of the editorial/advisory board of three international journals. He has published numerous papers and is co-editor of five books on various aspects of mathematical modelling. He is a past Vice-President of the IMA and has also served a period as Honorary Secretary of the Institute. He is a Chartered Mathematician and a Fellow of the IMA.

The original authors are David Burley, Dick Clements, John Searl, Nigel Steele, Jerry Wright together with Phil Dyke. Their short biographies can be found in the previous editions.

Publisher's Acknowledgements

Extracts in section 7.4.1, 8.5, 8.6 from Signal Processing Communication, ISBN 1898563233, 1 ed., Woodhead Publishing Ltd (Chapman, N, Goodhall, D, Steele, N).



1 Matrix Analysis

Chapter 1 Contents

1.1	Introduction	2
1.2	Review of matrix algebra	2
1.3	Vector spaces	9
1.4	The eigenvalue problem	13
1.5	Numerical methods	29
1.6	Reduction to canonical form	36
1.7	Functions of a matrix	51
1.8	Singular value decomposition	63
1.9	State-space representation	79
1.10	Solution of the state equation	86
1.11	Engineering application: Lyapunov stability analysis	101
1.12	Engineering application: capacitor microphone	104
1.13	Review exercises (1–19)	108

1.1 Introduction

In this chapter we turn our attention again to matrices, first considered in Chapter 5 of *Modern Engineering Mathematics* (MEM), and their applications in engineering. At the outset of the chapter we review the basic results of matrix algebra and briefly introduce vector spaces.

As the reader will be aware, matrices are arrays of real or complex numbers, and have a special, but not exclusive, relationship with systems of linear equations. Such systems occur quite naturally in the process of numerical solution of ordinary differential equations used to model everyday engineering processes. In Chapter 9 we shall see that they also occur in numerical methods for the solution of partial differential equations, for example those modelling the flow of a fluid or the transfer of heat. Systems of linear first-order differential equations with constant coefficients are at the core of the **state-space** representation of linear system models. Identification, analysis and indeed design of such systems can conveniently be performed in the state-space representation, with this form assuming a particular importance in the case of multivariable systems.

In all these areas it is convenient to use a matrix representation for the systems under consideration, since this allows the system model to be manipulated following the rules of matrix algebra. A particularly valuable type of manipulation is **simplification** in some sense. Such a simplification process is an example of a system transformation, carried out by the process of matrix multiplication. At the heart of many transformations are the **eigenvalues** and **eigenvectors** of a square matrix. In addition to providing the means by which simplifying transformations can be deduced, system eigenvalues provide vital information on system stability, fundamental frequencies, speed of decay and long-term system behaviour. For this reason, we devote a substantial amount of space to the process of their calculation, both by hand and by numerical means when necessary. Our treatment of numerical methods is intended to be purely indicative rather than complete, because a comprehensive matrix algebra computational tool kit, such as MATLAB, is now part of the essential armoury of all serious users of mathematics.

In addition to developing the use of matrix algebra techniques, we also demonstrate the techniques and applications of matrix analysis, focusing on the state-space system model widely used in control and systems engineering. Here we encounter the idea of a function of a matrix, in particular the matrix exponential, and we see again the role of the eigenvalues in its calculation. This edition also includes a section on singular value decomposition and the pseudo inverse, together with a brief section on Lyapunov stability of linear systems using quadratic forms.

1.2 Review of matrix algebra

This section contains a summary of the definitions and properties associated with matrices and determinants. A full account can be found in chapters of MEM or elsewhere. It is assumed that readers, prior to embarking on this chapter, have a fairly thorough understanding of the material summarized in this section.

1.2.1 Definitions

(a) An array of real numbers

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

is called an $m \times n$ **matrix** with m rows and n columns. The a_{ij} is referred to as the (ij)th **element** and denotes the element in the ith row and jth column. If $m = n$ then $\boldsymbol{A}$ is called a **square matrix** of order n. If the matrix has one column or one row then it is called a **column vector** or a **row vector** respectively.

(b) In a square matrix $\boldsymbol{A}$ of order n the diagonal containing the elements $a_{11}, a_{22}, \dots, a_{nn}$ is called the **principal** or **leading** diagonal. The sum of the elements in this diagonal is called the **trace** of $\boldsymbol{A}$, that is

$$\text{trace}\,\boldsymbol{A} = \sum_{i=1}^{n} a_{ii}$$

(c) A **diagonal matrix** is a square matrix that has its only non-zero elements along the leading diagonal. A special case of a diagonal matrix is the **unit** or **identity** matrix $\boldsymbol{I}$ for which $a_{11} = a_{22} = \dots = a_{nn} = 1$.

(d) A **zero** or **null** matrix $\mathbf{0}$ is a matrix with every element zero.

(e) The **transposed matrix** $\boldsymbol{A}^{\mathrm{T}}$ is the matrix $\boldsymbol{A}$ with rows and columns interchanged, its i, jth element being a_{ji}.

(f) A square matrix $\boldsymbol{A}$ is called a **symmetric matrix** if $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$. It is called **skew symmetric** if $\boldsymbol{A}^{\mathrm{T}} = -\boldsymbol{A}$.

1.2.2 Basic operations on matrices

In what follows the matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are assumed to have the i, jth elements a_{ij}, b_{ij} and c_{ij} respectively.

Equality

The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are **equal**, that is $\boldsymbol{A} = \boldsymbol{B}$, if they are of the same order $m \times n$ and

$$a_{ij} = b_{ij}, \quad 1 \leqslant i \leqslant m, \quad 1 \leqslant j \leqslant n$$

Multiplication by a scalar

If λ is a scalar then the matrix $\lambda\boldsymbol{A}$ has elements λa_{ij}.

Addition

We can only add an $m \times n$ matrix $\boldsymbol{A}$ to another $m \times n$ matrix $\boldsymbol{B}$ and the elements of the sum $\boldsymbol{A} + \boldsymbol{B}$ are

$$a_{ij} + b_{ij}, \quad 1 \leqslant i \leqslant m; \quad 1 \leqslant j \leqslant n$$

Properties of addition

(i) commutative law: $\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A}$

(ii) associative law: $(\boldsymbol{A} + \boldsymbol{B}) + \boldsymbol{C} = \boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C})$

(iii) distributive law: $\lambda(\boldsymbol{A} + \boldsymbol{B}) = \lambda\boldsymbol{A} + \lambda\boldsymbol{B}$, λ scalar

Matrix multiplication

If $\boldsymbol{A}$ is an $m \times p$ matrix and $\boldsymbol{B}$ a $p \times n$ matrix then we define the product $\boldsymbol{C} = \boldsymbol{AB}$ as the $m \times n$ matrix with elements

$$c_{ij} = \sum_{k=1}^{p} a_{ik}b_{kj}, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n$$

Properties of multiplication

(i) The commutative law is **not satisfied** in general; that is, in general $\boldsymbol{AB} \neq \boldsymbol{BA}$. Order does matter and we distinguish between $\boldsymbol{AB}$ and $\boldsymbol{BA}$ by the terminology: **pre**-multiplication of $\boldsymbol{B}$ by $\boldsymbol{A}$ to form $\boldsymbol{AB}$ and **post**-multiplication of $\boldsymbol{B}$ by $\boldsymbol{A}$ to form $\boldsymbol{BA}$.

(ii) Associative law: $\boldsymbol{A}(\boldsymbol{BC}) = (\boldsymbol{AB})\boldsymbol{C}$

(iii) If λ is a scalar then

$$(\lambda\boldsymbol{A})\boldsymbol{B} = \boldsymbol{A}(\lambda\boldsymbol{B}) = \lambda\boldsymbol{AB}$$

(iv) Distributive law over addition:

$$(\boldsymbol{A} + \boldsymbol{B})\boldsymbol{C} = \boldsymbol{AC} + \boldsymbol{BC} \qquad \boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{AB} + \boldsymbol{AC}$$

Note the importance of maintaining order of multiplication as in property (*i*).

(v) If $\boldsymbol{A}$ is an $m \times n$ matrix and if $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$ are the unit matrices of order m and n respectively then

$$\boldsymbol{I}_m\boldsymbol{A} = \boldsymbol{AI}_n = \boldsymbol{A}$$

Properties of the transpose

If $\boldsymbol{A}^{\mathrm{T}}$ is the transposed matrix of $\boldsymbol{A}$ then

(i) $(\boldsymbol{A} + \boldsymbol{B})^{\mathrm{T}} = \boldsymbol{A}^{\mathrm{T}} + \boldsymbol{B}^{\mathrm{T}}$

(ii) $(\boldsymbol{A}^{\mathrm{T}})^{\mathrm{T}} = \boldsymbol{A}$

(iii) $(\boldsymbol{AB})^{\mathrm{T}} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}$

1.2.3 Determinants

The determinant of a square $n \times n$ matrix $\boldsymbol{A}$ is denoted by det $\boldsymbol{A}$ or $|\boldsymbol{A}|$.

If we take a determinant of a matrix and delete row i and column j then the determinant remaining is called the **minor** M_{ij} of the (ij)th element. In general we can take any row i (or column) and evaluate an $n \times n$ determinant $|\boldsymbol{A}|$ as

$$|\boldsymbol{A}| = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} M_{ij}$$

A minor multiplied by the appropriate sign is called the **cofactor** A_{ij} of the (ij)th element so $A_{ij} = (-1)^{i+j} M_{ij}$ and thus

$$|\boldsymbol{A}| = \sum_{j=1}^{n} a_{ij} A_{ij}$$

Some useful properties

(i) $|\boldsymbol{A}^{\mathrm{T}}| = |\boldsymbol{A}|$

(ii) $|\boldsymbol{AB}| = |\boldsymbol{A}||\boldsymbol{B}|$

(iii) A square matrix $\boldsymbol{A}$ is said to be **non-singular** if $|\boldsymbol{A}| \neq 0$ and **singular** if $|\boldsymbol{A}| = 0$.

1.2.4 Adjoint and inverse matrices

Adjoint matrix

The **adjoint** of a square matrix $\boldsymbol{A}$ is the transpose of the matrix of cofactors, so for a 3×3 matrix $\boldsymbol{A}$

$$\text{adj}\,\boldsymbol{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}^{\mathrm{T}}$$

Properties

(i) $\boldsymbol{A}\,(\text{adj}\,\boldsymbol{A}) = |\boldsymbol{A}|\boldsymbol{I}$

(ii) $|\text{adj}\,\boldsymbol{A}| = |\boldsymbol{A}|^{n-1}$, where n is the order of $\boldsymbol{A}$

(iii) $\text{adj}\,(\boldsymbol{AB}) = (\text{adj}\,\boldsymbol{B})(\text{adj}\,\boldsymbol{A})$

Inverse matrix

Given a square matrix $\boldsymbol{A}$ if we can construct a square matrix $\boldsymbol{B}$ such that

$$\boldsymbol{BA} = \boldsymbol{AB} = \boldsymbol{I}$$

then we call $\boldsymbol{B}$ the inverse of $\boldsymbol{A}$ and write it as $\boldsymbol{A}^{-1}$.

Properties

(i) If $\boldsymbol{A}$ is non-singular then $|\boldsymbol{A}| \neq 0$ and $\boldsymbol{A}^{-1} = (\text{adj } \boldsymbol{A})/|\boldsymbol{A}|$.

(ii) If $\boldsymbol{A}$ is singular then $|\boldsymbol{A}| = 0$ and $\boldsymbol{A}^{-1}$ does not exist.

(iii) $(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$.

All the basic matrix operations may be implemented in MATLAB using simple commands. In MATLAB a matrix is entered as an array, with row elements separated by spaces (or commas) and each row of elements separated by a semicolon(;), or the return key to go to a new line. Thus, for example,

```
A=[1 2 3; 4 0 5; 7 4 2]
```

gives

```
A=
   1 2 3
   4 0 5
   7 4 2
```

Having specified the two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ the operations of addition, subtraction and multiplication are implemented using respectively the commands

```
C=A+B, C=A-B, C=A*B
```

The trace of the matrix $\boldsymbol{A}$ is determined by the command `trace(A)`, and its determinant by `det(A)`.

Multiplication of a matrix $\boldsymbol{A}$ by a scalar is carried out using the command `*`, while raising $\boldsymbol{A}$ to a given power is carried out using the command `^` . Thus, for example, $3\boldsymbol{A}^2$ is determined using the command `C=3*A^2`.

The transpose of a real matrix $\boldsymbol{A}$ is determined using the apostrophe `'` key; that is `C=A'` (to accommodate complex matrices the command `C=A.'` should be used). The inverse of $\boldsymbol{A}$ is determined by `C=inv(A)`.

For matrices involving algebraic quantities, or when exact arithmetic is desirable use of the Symbolic Math Toolbox is required; in which matrices must be expressed in symbolic form using the `sym` command. The command `A=sym(A)` generates the symbolic form of $\boldsymbol{A}$. For example, for the matrix

$$\boldsymbol{A} = \begin{bmatrix} 2.1 & 3.2 & 0.6 \\ 1.2 & 0.5 & 3.3 \\ 5.2 & 1.1 & 0 \end{bmatrix}$$

the commands

```
A=[2.1 3.2 0.6; 1.2 0.5 3.3; 5.2 1.1 0];
A=sym(A)
```

generate

```
A=
  [21/10, 16/5, 3/5]
  [6/5, 1/2, 33/10]
  [26/5, 11/10, 0]
```

Symbolic manipulation can also be undertaken in MATLAB using the MuPAD version of Symbolic Math Toolbox.

Such operations may be performed in Python. Details are not given here, but the interested reader is directed to, for example, *Beginning Python* by Lie Hethand (Springer, 2005). The `numPy` package should be loaded.

1.2.5 Linear equations

In this section we reiterate some definitive statements about the solution of the system of simultaneous linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad\qquad & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

or, in matrix notation,

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

that is,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1.1}$$

where $\boldsymbol{A}$ is the matrix of coefficients and $\boldsymbol{x}$ is the vector of unknowns. If $\boldsymbol{b} = \boldsymbol{0}$ the equations are called **homogeneous**, while if $\boldsymbol{b} \neq \boldsymbol{0}$ they are called **nonhomogeneous** (or **inhomogeneous**). Considering individual cases:

Case (i): If $\boldsymbol{b} \neq \boldsymbol{0}$ and $|\boldsymbol{A}| \neq 0$ then we have a unique solution $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$.

Case (ii): If $\boldsymbol{b} = \boldsymbol{0}$ and $|\boldsymbol{A}| \neq 0$ we have the trivial solution $\boldsymbol{x} = 0$.

Case (iii): If $\boldsymbol{b} \neq \boldsymbol{0}$ and $|\boldsymbol{A}| = 0$ then we have two possibilities: **either** the equations are inconsistent and we have no solution **or** we have infinitely many solutions.

Case (iv): If $\boldsymbol{b} = \boldsymbol{0}$ and $|\boldsymbol{A}| = 0$ then we have infinitely many solutions.

Case (iv) is one of the most important, since from it we can deduce the important result that **the homogeneous equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ has a non-trivial solution if and only if** $|\boldsymbol{A}| = 0$.

Provided that a solution to (1.1) exists it may be determined in MATLAB using the command `x=A\b`. For example, the system of simultaneous equations

$$x + y + z = 6, \quad x + 2y + 3z = 14, \quad x + 4y + 9z = 36$$

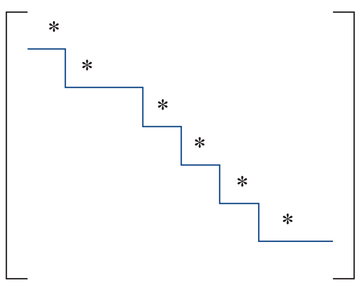
may be written in the matrix form

$$\underset{\boldsymbol{A}}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 9 \end{bmatrix}} \underset{\boldsymbol{x}}{\begin{bmatrix} x \\ y \\ z \end{bmatrix}} = \underset{\boldsymbol{b}}{\begin{bmatrix} 6 \\ 14 \\ 36 \end{bmatrix}}$$

Entering `A` and `b` and using the command `x = A\b` provides the answer $x = 1, y = 2, z = 3$.

1.2.6 Rank of a matrix

We adopt the following constructive definition of the **rank**, rank $\boldsymbol{A}$ of a matrix $\boldsymbol{A}$. First, using elementary row operations, the matrix $\boldsymbol{A}$ is reduced to **echelon form**



in which all the entries below the line are zero, and the leading element, marked *, in each row above the line is non-zero. Then the number of non-zero rows in the echelon form is equal to rank $\boldsymbol{A}$. These are equivalent definitions.

When considering the solution of (1.1) we saw that provided the determinant of the matrix $\boldsymbol{A}$ was not zero we could obtain explicit solutions in terms of the inverse matrix. However, when we looked at cases with zero determinant the results were much less clear. The idea of the rank of a matrix helps to make these results more precise. Defining the **augmented matrix** $(\boldsymbol{A} : \boldsymbol{b})$ for (1.1) as the matrix $\boldsymbol{A}$ with the column $\boldsymbol{b}$ added to it then we can state the results of cases (iii) and (iv) of Section 1.2.5 more clearly as follows:

If $\boldsymbol{A}$ and $(\boldsymbol{A} : \boldsymbol{b})$ have different rank then we have no solution to (1.1). If the two matrices have the same rank then a solution exists, and furthermore the solution will contain $n -$ rank $\boldsymbol{A}$ free parameters.

In MATLAB the rank of the matrix $\boldsymbol{A}$ is generated using the command `rank(A)`. For example, if

$$\boldsymbol{A} = \begin{bmatrix} -1 & 2 & 2 \\ 0 & 0 & 1 \\ -1 & 2 & 0 \end{bmatrix}$$

the commands

```
A=[-1 2 2; 0 0 1; -1 2 0];
rank(A)
```

generate

```
ans=2
```

In MAPLE the command is also `rank(A)`.

1.3 Vector spaces

Vectors and matrices form part of a more extensive formal structure called a **vector space**. The theory of vector spaces underpins many approaches to numerical methods and the approximate solution of many equations that arise in engineering analysis. In this section we shall, briefly, introduce some basic ideas of vector spaces necessary for later work in this chapter.

Definition

A **real vector space** V is a set of objects called **vectors** together with rules for addition and multiplication by real numbers. For any three vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ in V and any real numbers α and β the sum $\boldsymbol{a} + \boldsymbol{b}$ and the product $\alpha\boldsymbol{a}$ also belong to V and satisfy the following axioms:

(a) $\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{b} + \boldsymbol{a}$

(b) $\boldsymbol{a} + (\boldsymbol{b} + \boldsymbol{c}) = (\boldsymbol{a} + \boldsymbol{b}) + \boldsymbol{c}$

(c) there exists a zero vector $\boldsymbol{0}$ such that

$\boldsymbol{a} + \boldsymbol{0} = \boldsymbol{a}$

(d) for each $\boldsymbol{a}$ in V there is an element $-\boldsymbol{a}$ in V such that

$\boldsymbol{a} + (-\boldsymbol{a}) = \boldsymbol{0}$

(e) $\alpha(\boldsymbol{a} + \boldsymbol{b}) = \alpha\boldsymbol{a} + \alpha\boldsymbol{b}$

(f) $(\alpha + \beta)\boldsymbol{a} = \alpha\boldsymbol{a} + \beta\boldsymbol{a}$

(g) $(\alpha\beta)\boldsymbol{a} = \alpha(\beta\boldsymbol{a})$

(h) $1\boldsymbol{a} = \boldsymbol{a}$