

## Linear Algebra for Data Science, Machine Learning, and Signal Processing

Maximize student engagement and understanding of matrix methods in data-driven applications with this modern teaching package. Students are introduced to matrices in two preliminary chapters, before progressing to advanced topics such as the nuclear norm, proximal operators, and convex optimization. Highlighted applications include low-rank approximation, matrix completion, subspace learning, logistic regression for binary classification, robust principal component analysis, dimensionality reduction, and Procrustes problems.

Extensively tested in the classroom, the book includes over 200 multiple-choice questions suitable for in-class interactive learning or quizzes, as well as homework exercises (with solutions available for instructors). It encourages active learning with engaging “explore” questions, with answers at the back of each chapter, and Julia code examples to demonstrate how the math is actually used in practice. A suite of computational notebooks offers a hands-on learning experience for students. This is a perfect textbook for upper-level undergraduates and first-year graduate students who have taken a prior course in linear algebra basics.

**Jeffrey A. Fessler** is the William L. Root Professor of Electrical Engineering and Computer Science at the University of Michigan. He received the Edward Hoffman Medical Imaging Scientist Award in 2013, and an IEEE EMBS Technical Achievement Award in 2016. He received the 2023 Steven S. Attwood Award, the highest honor awarded to a faculty member by the College of Engineering at the University of Michigan. He is a fellow of the IEEE.

**Raj Rao Nadakuditi** is an Associate Professor of Electrical Engineering and Computer Science at the University of Michigan. He received the Jon R. and Beverly S. Holt Award for Excellence in Teaching in 2018 and the Ernest and Bettine Kuh Distinguished Faculty Award in 2021.

“The authors provide a comprehensive contemporary presentation of linear algebra, demonstrating its foundational and intrinsic value to modern subjects, such as machine/deep learning, data science, and signal processing. The presentation is fun, exciting, topic-diverse, classroom tested, and addresses practical implementation in ways that jump start students’ use.”

**Christ D. Richmond**, *Duke University*

“This is an excellent and timely text that addresses the specific needs of data science (DS), machine learning (ML), and signal processing (SP). Its nicely crafted coverage is designed to prepare students in the areas of DS/ML/SP, in particular, by drawing thoughtful examples from these fields. With increasing demands from data-based sciences, there is a pressing need for a book in ‘the new linear algebra,’ and this text fills this gap.”

**Yousef Saad**, *University of Minnesota*

“With the emergence of Graphics Processing Units (GPUs), the importance of linear algebra for machine learning cannot be overstated. This is a thoughtful and timely work on the topic of linear algebra for machine learning, which I anticipate will be one of the definitive textbooks in this field.”

**Vahid Tarokh**, *Duke University*

“To see the spirit of this book, just look at pages 1 and 2. A painting is deblurred by linear algebra. Great ideas and how to use them in real time – all on display!”

**Gilbert Strang**, *Massachusetts Institute of Technology*

# Linear Algebra for Data Science, Machine Learning, and Signal Processing

JEFFREY A. FESSLER

*University of Michigan, Ann Arbor*

RAJ RAO NADAKUDITI

*University of Michigan, Ann Arbor*



CAMBRIDGE  
UNIVERSITY PRESS

Cambridge University Press & Assessment

978-1-009-41814-0 — Linear Algebra for Data Science, Machine Learning, and Signal Processing

Jeffrey A. Fessler, Raj Rao Nadakuditi

Frontmatter

[More Information](#)



CAMBRIDGE  
UNIVERSITY PRESS

Shaftesbury Road, Cambridge CB2 8EA, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of education, learning and research at the highest international levels of excellence.

[www.cambridge.org](http://www.cambridge.org)

Information on this title: [www.cambridge.org/highereducation/isbn/9781009418140](http://www.cambridge.org/highereducation/isbn/9781009418140)

DOI: 10.1017/9781009418164

© Jeff Fessler and Raj Rao Nadakuditi 2024

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press & Assessment.

First published 2024

*A catalogue record for this publication is available from the British Library*

*Library of Congress Cataloging-in-Publication Data*

Names: Fessler, Jeffrey A., 1964- author. | Nadakuditi, Raj Rao, 1977- author.

Title: Linear algebra for data science, machine learning, and signal processing / Jeffrey A. Fessler, Raj Rao Nadakuditi.

Description: New York, NY : Cambridge University Press, 2024. | Includes bibliographical references and index.

Identifiers: LCCN 2023050878 (print) | LCCN 2023050879 (ebook) |

ISBN 9781009418140 (hardback) | ISBN 9781009418164 (epub)

Subjects: LCSH: Algebras, Linear–Textbooks. | Matrices–Textbooks. |

Machine learning–Mathematics–Textbooks. | Signal

processing–Mathematics–Textbooks. | Julia (Computer program language)–Textbooks.

Classification: LCC QA184.2 .F47 2024 (print) | LCC QA184.2 (ebook) |

DDC 512/.5–dc23/eng/20231117

LC record available at <https://lcn.loc.gov/2023050878>

LC ebook record available at <https://lcn.loc.gov/2023050879>

ISBN 978-1-009-41814-0 Hardback

Additional resources for this publication at [www.cambridge.org/FesslerNadakuditi](http://www.cambridge.org/FesslerNadakuditi)

Cambridge University Press & Assessment has no responsibility for the persistence or accuracy of URLs for external or third-party Internet websites referred to in this publication and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

Cambridge University Press & Assessment

978-1-009-41814-0 — Linear Algebra for Data Science, Machine Learning, and Signal Processing

Jeffrey A. Fessler , Raj Rao Nadakuditi

Frontmatter

[More Information](#)

---

**To Sue**

**To Tata, Bama, Dad, Reizo and Ms. Gracie**

Cambridge University Press & Assessment  
978-1-009-41814-0 — Linear Algebra for Data Science, Machine Learning, and Signal Processing  
Jeffrey A. Fessler , Raj Rao Nadakuditi  
Frontmatter  
[More Information](#)

---

Contents

<i>Preface</i>	<i>page</i> xv
<i>Acknowledgments</i>	xix
<b>1    Getting Started</b>	<b>1</b>
1.1    Introduction	1
1.2    Example Applications	1
1.2.1    Signal Processing Example: Image Deblurring	1
1.2.2    Computer Vision Applications	2
1.2.3    Machine Learning Example: Handwritten Digit Recognition	3
1.3    Formatting	3
1.4    Notation Preview	4
1.4.1    What the $\mathbb{F}$ Means	6
1.5    Julia	7
1.6    Fields, Vector Spaces, Linear Maps	7
1.6.1    Field of Scalars	8
1.6.2    Vector Spaces	9
1.6.3    Linear Maps and Linear Operators	11
<b>2    Introduction to Matrices</b>	<b>12</b>
2.1    Introduction	12
2.2    Basics of Vectors and Matrices	12
2.3    Matrix Structures	17
2.3.1    Common Matrix Shapes and Types	17
2.3.2    Matrix Transpose and Symmetry	20
2.4    Multiplication	23
2.4.1    Vector–Vector Multiplication	23
2.4.2    Matrix–Vector Multiplication	25
2.4.3    Matrix–Matrix Multiplication	29
2.4.4    Matrix Multiplication Properties	30
2.4.5    Kronecker and Hadamard Products, and the vec Operator	34
2.4.6    Using Matrix–Vector Operations	35
2.4.7    Invertibility	40
2.5    Orthogonality and the Euclidean Norm	42
2.5.1    Orthogonal Vectors	42
2.5.2    Euclidean Norm	42

2.5.3	Cauchy–Schwarz Inequality	43
2.5.4	Orthogonal Matrices	43
2.6	Determinant of a Matrix	45
2.6.1	Determinant Properties	46
2.6.2	Matrices with Units	47
2.6.3	Laplace’s Determinant Formula	48
2.6.4	Avoiding Computation	48
2.6.5	Small Matrices	49
2.7	Eigenvalues	50
2.7.1	Eigenvectors	51
2.7.2	Practical Implementation	51
2.7.3	Properties of Eigenvalues	52
2.8	Trace	54
2.9	Summary	55
<b>3</b>	<b>Matrix Factorization: Eigendecomposition and SVD</b>	<b>63</b>
3.1	Introduction	63
3.1.1	Matrix Factorizations	63
3.1.2	Square Matrices	64
3.2	Spectral Theorem for Symmetric Matrices	65
3.2.1	Normal Matrices	66
3.2.2	Square Asymmetric and Nonnormal Matrices	68
3.2.3	Geometry of Matrix Diagonalization	70
3.2.4	Matrix Powers and Matrix Exponential	73
3.3	Singular Value Decomposition	74
3.3.1	Singular Values and Singular Vectors	74
3.3.2	Existence of SVD	74
3.3.3	Geometry	75
3.3.4	Practical Implementation in JULIA	76
3.3.5	SVD Basic Properties	77
3.4	The Matrix 2-Norm or Spectral Norm	78
3.4.1	Optimization: min versus arg min	79
3.4.2	Eigenvalues as Optimization Problems	80
3.4.3	Smallest Singular Value	81
3.5	Relating SVDs and Eigendecompositions	82
3.5.1	When Does $U = V$ ?	84
3.5.2	SVD Computation Using Eigendecomposition	85
3.5.3	SVD Nonuniqueness Revisited	87
3.6	Positive Semidefinite Matrices	88
3.6.1	Relating Positive (Semi)Definiteness to Eigenvalues	89
3.7	Summary	89
<b>4</b>	<b>Subspaces, Rank, and Nearest-Subspace Classification</b>	<b>96</b>
4.1	Introduction	96
4.2	Subspaces	96
4.2.1	Span	98



	Contents	ix
4.2.2	Linear Independence	99
4.2.3	Basis	101
4.2.4	Dimension	103
4.2.5	Sums and Intersections of Subspaces	105
4.2.6	Direct Sum of Subspaces	106
4.2.7	Dimensions of Sums of Subspaces	106
4.2.8	Orthogonal Complement of a Subspace	107
4.2.9	Linear Transforms	108
4.2.10	Range of a Matrix	109
4.3	Rank of a Matrix	111
4.3.1	Practical Use in JULIA	112
4.3.2	Rank of a Matrix Product	112
4.3.3	Other Rank Properties	113
4.3.4	Spark	114
4.3.5	Unitary Invariance of Rank	114
4.4	Nullspace of a Matrix	115
4.4.1	Nullspace or Kernel	116
4.4.2	Properties of Null Space	116
4.4.3	Columns of Unitary Matrices	118
4.5	The Four Fundamental Spaces	119
4.5.1	Anatomy of the SVD	121
4.5.2	SVD of Finite Differences	123
4.5.3	Synthesis View of Matrix Decomposition	125
4.6	Orthogonal Bases	125
4.6.1	Finding Coordinates in an Orthogonal Basis	126
4.6.2	Stiefel Manifold of Orthogonal Bases	127
4.7	Spotting Decompositions	128
4.7.1	Matrix–Vector Products and the SVD	130
4.8	Application: Signal Classification	130
4.8.1	Projection onto a Set	130
4.8.2	Nearest Point in a Subspace	131
4.8.3	Signal Classification by Nearest Subspace	133
4.9	Optimization Preview	134
4.9.1	Convex Sets	135
4.9.2	Convex Functions	136
4.10	Summary	137
5	Linear Least-Squares Regression and Binary Classification	143
5.1	Introduction	143
5.2	Introduction to Linear Equations	143
5.2.1	Solving $A\mathbf{x} = \mathbf{y}$	144
5.2.2	Linear Regression and Machine Learning	144
5.2.3	Lifting for Nonlinear Regression	145
5.3	Linear Least-Squares Estimation	145
5.3.1	Minimization and Gradients	148
5.3.2	Solving LLS Using the Normal Equations	150

5.3.3	Solving LLS Problems Using the Compact SVD	151
5.3.4	Uniqueness of LLS Solution	154
5.4	Moore–Penrose Pseudoinverse	155
5.4.1	Pseudoinverse and Matrix Products	155
5.4.2	Pseudoinverse and SVD	156
5.5	LLS: Under-Determined Case	158
5.5.1	Orthogonality Principle	160
5.5.2	Minimum-Norm LS Solution via Pseudoinverse	162
5.6	Truncated SVD Solution	164
5.6.1	Condition Number	164
5.6.2	Practical Implementation of Truncated SVD Solution	165
5.6.3	Low-Rank Approximation Interpretation of Truncated SVD	165
5.6.4	Noise Effects and Perturbations	166
5.6.5	Tikhonov Regularization, or Ridge Regression	167
5.7	Summary of LLS Solution Methods in Terms of SVD	168
5.8	Frames and Tight Frames	168
5.8.1	Properties of a Frame	170
5.8.2	Tight Frame	170
5.8.3	Parseval Tight Frame	171
5.8.4	Properties of Parseval Tight Frames	171
5.8.5	Frame Summary	173
5.9	Projection and Orthogonal Projection	174
5.9.1	Idempotent Matrix	174
5.9.2	Orthogonal Projection Matrix	176
5.9.3	Projection onto a Subspace	177
5.9.4	Binary Classifier Design Using Least Squares	182
5.9.5	Empirical Risk Minimization	183
5.10	Recursive Least Squares	184
5.10.1	RLS with a Forgetting Factor	185
5.11	Summary	186
<b>6</b>	<b>Norms and Procrustes Problems</b>	197
6.1	Introduction	197
6.2	Vector Norms	197
6.2.1	Examples of Vector Norms	198
6.2.2	Practical Implementation	199
6.2.3	Properties of Vector Norms	200
6.2.4	Norm Notation	201
6.2.5	Robust Regression Application	201
6.2.6	Unitarily Invariant Vector Norms	202
6.3	Inner Products	203
6.3.1	Examples of Inner Products	203
6.3.2	Properties of Inner Products	204
6.3.3	More Inner Product Inequalities	205
6.3.4	Angle Between Vectors	205
6.3.5	Angle Between Subspaces	206

6.4	Matrix Norms and Operator Norms	206
6.4.1	Examples of Matrix Norms	207
6.4.2	Induced Matrix Norms	208
6.4.3	Norms Defined in Terms of Singular Values	210
6.4.4	Practical Implementation	214
6.4.5	Properties of Matrix Norms	214
6.4.6	Spectral Radius	217
6.4.7	Practical Step Size for Gradient Descent	218
6.5	Convergence of Sequences of Vectors and Matrices	219
6.6	Generalized Inverse of a Matrix	221
6.6.1	Minimum Frobenius Norm Generalized Inverse	221
6.7	Procrustes Analysis	222
6.7.1	Sanity Check and Scale Invariance	224
6.7.2	Procrustes Generalizations	225
6.7.3	Subspace/Span Comparisons	228
6.7.4	Weighted Procrustes Problems	228
6.7.5	Practical Implementation	229
6.8	Summary	229
<b>7</b>	<b>Low-Rank Approximation and Multidimensional Scaling</b>	<b>238</b>
7.1	Introduction	238
7.2	Low-Rank Approximation via Frobenius Norm	238
7.2.1	Eckart–Young–Mirsky Theorem	239
7.2.2	Subspace Approximation Perspective	240
7.2.3	Implementation	240
7.2.4	Choosing Rank via Permutation	242
7.2.5	Nonuniqueness of SVD and Low-Rank Approximation	243
7.2.6	One-Dimensional Example	244
7.2.7	Generalization to Other Norms	245
7.2.8	Bases for $\mathbb{R}^{M \times N}$	247
7.2.9	Low-Rank Approximation Summary	248
7.2.10	Rank and Stability	249
7.2.11	Example: Photometric Stereo	250
7.3	Sensor Localization Application: Multidimensional Scaling	250
7.3.1	Derivation (Analysis)	251
7.3.2	MDS Method	254
7.3.3	Practical Implementation	255
7.3.4	Extensions	256
7.4	Proximal Operators	257
7.4.1	Soft Thresholding	257
7.4.2	Hard Thresholding	259
7.5	Alternative Low-Rank Approximation Formulations	260
7.5.1	Unconstrained/Regularized Formulation	260
7.5.2	General Unitarily Invariant Formulations	260
7.5.3	Singular Value Hard Thresholding	261
7.5.4	Singular Value Soft Thresholding	262

7.5.5	Other Extensions of Low-Rank Approximation	263
7.6	Choosing the Rank or Regularization Parameter	263
7.6.1	Stein's Unbiased Risk Estimate	264
7.6.2	OptShrink	265
7.7	Related Methods: Autoencoders and PCA	269
7.7.1	Relation to Autoencoder with Linear Layers	269
7.7.2	Relation to Principal Component Analysis	270
7.8	Subspace Learning for Classification	275
7.8.1	Subspace Clustering	277
7.9	Subspace Tracking and Streaming PCA	277
7.9.1	Incremental SVD	278
7.9.2	Streaming PCA	278
7.10	Summary	279
<b>8</b>	<b>Special Matrices, Markov Chains, and PageRank</b>	<b>283</b>
8.1	Introduction	283
8.2	Companion Matrices	283
8.2.1	Practical Implementation	285
8.2.2	Polynomial Matrix Functions	286
8.2.3	Eigenvectors of Companion Matrices	287
8.2.4	Vandermonde Matrices	288
8.2.5	Kronecker Sum and Polynomial Roots	289
8.3	Circulant Matrices	290
8.3.1	Relationship to DFT Properties from DSP	293
8.3.2	Practical Implementation	293
8.3.3	Spectral Properties of Circulant Matrices	294
8.3.4	Inverting a Circulant Matrix	294
8.4	Toeplitz Matrices	294
8.4.1	Toeplitz Matrix Multiplication with a Vector	295
8.4.2	Inverting a Toeplitz Matrix	295
8.4.3	Factoring a Toeplitz Matrix	295
8.5	Power Iteration	296
8.5.1	Convergence of the Power Iteration	297
8.5.2	Geršgorin Disk Theorem	298
8.6	Nonnegative Matrices and Graphs	301
8.6.1	Primitive Matrices	301
8.6.2	Weighted Directed Graphs	303
8.6.3	Strongly Connected Graphs	305
8.6.4	Irreducible Matrix	306
8.6.5	Matrix Period	307
8.7	Nonnegative Matrices and Perron–Frobenius Theorems	309
8.7.1	Perron–Frobenius for Square Nonnegative Matrices	309
8.7.2	Perron–Frobenius for Nonnegative Irreducible Matrices	311
8.7.3	Perron–Frobenius for Primitive Matrices	312
8.7.4	Perron–Frobenius for Stochastic Matrices	312
8.8	Markov Chains	313

8.8.1	Equilibrium Distribution(s) of a Markov Chain	315
8.8.2	Limiting Distribution(s) of a Markov Chain	316
8.8.3	Markov Chains with Strongly Connected Graphs	318
8.8.4	Google's PageRank Method	319
8.9	Graph Laplacian and Spectral Clustering	322
8.9.1	Clustering	322
8.9.2	Weighted Graph Based on Similarity	323
8.9.3	Connected Components	324
8.9.4	Graph Laplacian	324
8.9.5	Spectral Clustering Algorithm	325
8.9.6	Laplacian Eigenmaps	326
8.10	Summary	327
<b>9</b>	<b>Optimization Basics and Logistic Regression</b>	<b>335</b>
9.1	Introduction	335
9.2	Preconditioned Gradient Descent for LS	335
9.2.1	Tool: Matrix Square Root	336
9.2.2	Convergence Rate Analysis of PGD: First Steps	338
9.2.3	Classical GD: Step Size Bounds	339
9.2.4	Optimal Step Size for GD	340
9.2.5	Ideal Preconditioner for PGD	340
9.2.6	Tool: Positive (Semi)Definiteness Properties	341
9.2.7	General Preconditioners for PGD	342
9.2.8	Diagonal Majorizer	342
9.2.9	Preconditioning Illustration/Demo	344
9.2.10	Convergence Rates	345
9.2.11	Tool: Commuting (Square) Matrices	346
9.2.12	Monotonicity	347
9.3	Preconditioned Steepest Descent	349
9.4	Gradient Descent for Smooth Convex Functions	349
9.4.1	Lipschitz Continuity	350
9.4.2	Convexity and Hessian	351
9.4.3	GD Convergence Theorem	352
9.4.4	Nesterov's Fast Gradient Method	353
9.4.5	Optimized Gradient Method	354
9.4.6	Gradient Projection Method	355
9.5	Machine Learning via Logistic Regression for Binary Classification	356
9.5.1	Practical Implementation of Logistic Regression	360
9.5.2	Numerical Results: Logistic Regression	360
9.6	Stochastic Gradient Descent	360
9.7	Summary	361
<b>10</b>	<b>Matrix Completion and Recommender Systems</b>	<b>365</b>
10.1	Introduction	365
10.2	Measurement Model	366
10.2.1	Practical Implementation	366
10.2.2	Sampling Conditions for LRMC	366

10.2.3	Sampling Mask	367
10.3	LRMC: Noiseless Case	368
10.3.1	Noiseless Problem Statement	368
10.3.2	Alternating Projection Approach to LRMC	368
10.4	LRMC: Noisy Case	371
10.4.1	Noisy Problem Statement	371
10.4.2	Majorize–Minimize (MM) Iterations	372
10.4.3	MM Methods for LRMC	373
10.4.4	LRMC by Iterative Low-Rank Approximation	373
10.4.5	LRMC by Iterative Singular Value Hard Thresholding	374
10.4.6	LRMC by Iterative Singular Value Soft Thresholding	374
10.4.7	Iterative Soft-Thresholding Algorithm	375
10.4.8	Debiasing the Nuclear Norm Effects	376
10.4.9	Factorization Approaches	377
10.4.10	Demo	378
10.5	Robust PCA and Video Foreground/Background Separation	378
10.5.1	Robust PCA	378
10.5.2	Video Foreground/Background Separation	378
10.6	Nonnegative Matrix Factorization	379
10.7	Summary	380
<b>11</b>	<b>Neural Network Models</b>	<b>381</b>
11.1	Introduction	381
11.2	The Importance of Nonlinearity	381
11.3	Fully Connected NN Models	383
11.3.1	Perceptron Model	383
11.3.2	Multilayer Perceptron NN Models	384
11.3.3	Model Expressiveness	385
11.4	Training NN Models	385
11.4.1	Weight Regularization	386
11.5	CNN Models	387
11.5.1	Matrix Representations	388
11.5.2	CNN Architectures	388
11.6	Summary	389
<b>12</b>	<b>Random Matrix Theory, Signal + Noise Matrices, and Phase Transitions</b>	<b>390</b>
12.1	Introduction	390
12.1.1	Perturbation Bounds	390
12.2	Roundoff Error	391
12.2.1	RMT for Roundoff Analysis	393
12.3	Additive Noise	396
12.4	Outliers	400
12.5	Matrix Completion	401
12.6	Summary	403
	<i>References</i>	405
	<i>Index</i>	423

## Preface

### Overview

Modern methods in data science, machine learning, and signal processing (DS–ML–SP) all build extensively on matrix methods and linear algebra. Often students who are interested in DS–ML–SP are advised to “go take a linear algebra course” with the promise that the material learned there will be useful later in more advanced courses. The content in this book is designed to teach important linear algebra ideas in an integrated way with computational methods in the context of DS–ML–SP applications. The focus here is on using matrix methods to pose and solve DS–ML–SP problems, rather than to provide rigorous proofs of linear algebra theorems. Traditional linear algebra and numerical linear algebra courses spend considerable time focusing on solving  $\mathbf{Ax} = \mathbf{b}$ . Solving systems of equations is essential for physics-based applications described by partial differential equations, whereas modern DS–ML–SP applications are data-driven and rarely reduce to solving  $\mathbf{Ax} = \mathbf{b}$ . Thus, this book treats the topic of solving linear systems only very briefly so that we can get to “the good stuff” like regularized least squares regression (Chapter 5), multidimensional scaling (Chapter 6), low-rank matrix approximation and Procrustes analysis (Chapter 7), Markov chains and the PageRank method (Chapter 8), logistic regression for binary classification, (Chapter 9), and matrix completion (Chapter 10). Put another way, after establishing a foundation in Chapters 1–3, every chapter that follows has further mathematical methods and models, along with one or more compelling applications to motivate and illustrate them. The goal of this book is to provide mathematical foundations for subsequent DS–ML–SP courses while also introducing matrix-based DS–ML–SP methods and applications that are useful in their own right.

### Software

Nearly every mathematical concept in this book has corresponding operations in software, and this book describes those operations using the JULIA language [1]. This relatively new language has many benefits. JULIA is designed in a way that allows the code to look very similar to the math, facilitating the translation of algorithm ideas

to working software. JULIA uses dynamic typing so it is suitable for interactive and educational use, yet it is very fast because it is compiled. JULIA is open source and its git-based package manager greatly facilitates reproducible research. Readers do not need to know JULIA to begin using this book; the language borrows a lot of ideas from MATLAB and Python (among others), so readers familiar with those tools will be able to follow the examples easily. Readers can view the JULIA code examples as pseudocode even if they prefer to use other languages, learning some JULIA along the way. There are many tutorials online as well as other books based on JULIA that provide useful references, for example, [2]. Every figure in this book was generated using JULIA.

## Textbook Use

The content in this book has been used in two first-year graduate courses (EECS 505 and EECS 551) at the University of Michigan since at least 2016, taken by several thousand students over that time. Since 2017, those courses have used JULIA as the primary (505) or only (551) language for illustrating and implementing the ideas. Senior-level undergraduates with mathematical maturity have also taken these courses. The courses include weekly discussion sections where students apply the techniques to real data (like handwritten digit classification) using Jupyter notebooks. (The Ju in Jupyter is for JULIA.)

Several methods in the book are illustrated in JULIA demos at the website <https://github.com/JeffFessler/book-la-demo>. These demos were created using the convenient `Literate.jl` and `Documenter.jl` tools in JULIA. Those tools generate HTML output that is easily viewed in a browser, as well as Jupyter notebooks that students can use and modify.

A prior undergraduate-level course in linear algebra is likely to be helpful as background for getting the most out of this book. A prior undergraduate-level course in digital signal processing is helpful for understanding a few of the examples, but is not essential for most of the book.

## Instructor Resources

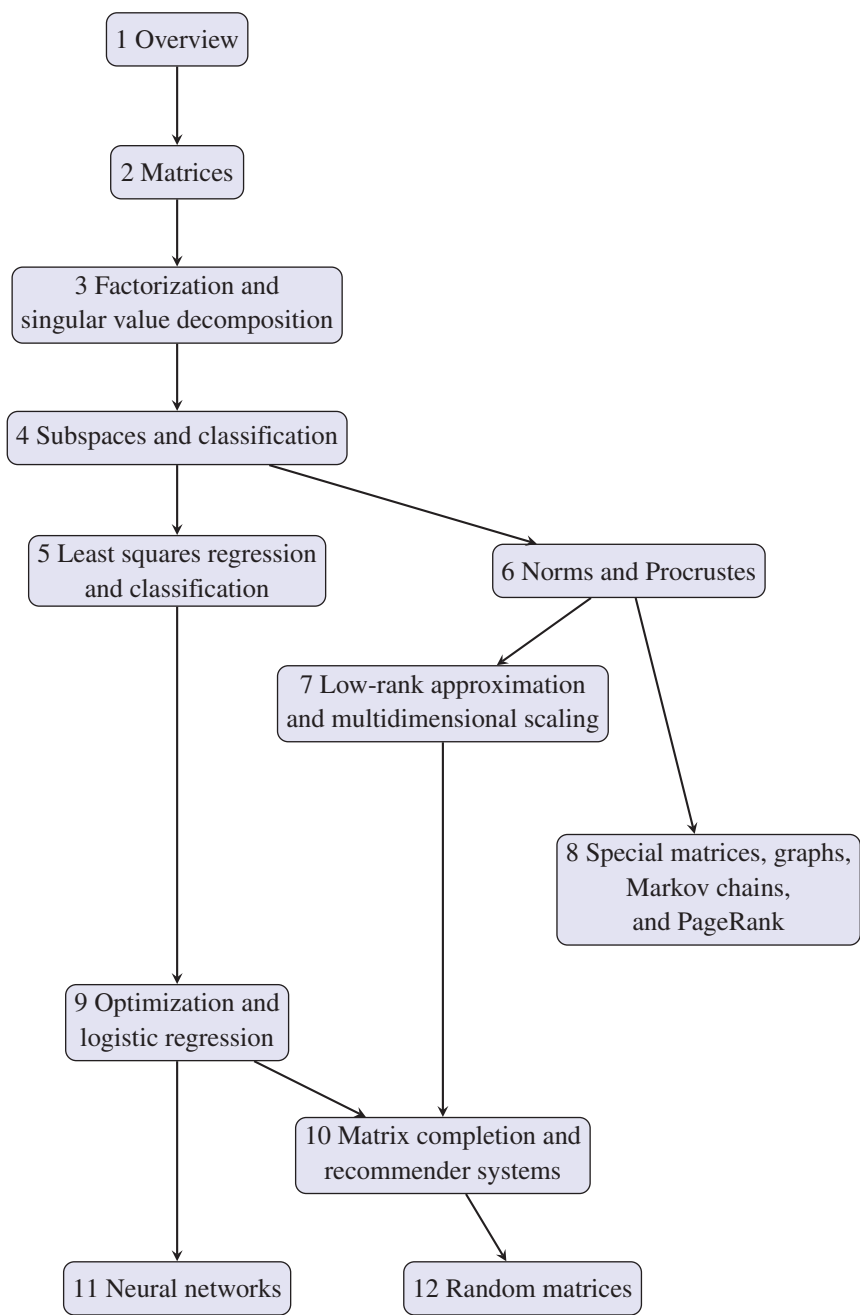
Embedded in the chapters are over 200 multiple-choice questions that instructors can use for in-class active learning exercises, or for self study by readers.

There are over 150 exercises at the ends of the chapters. Typeset solutions to these problems are available for instructors on the book's web page <https://doi.org/10.1017/9781009418164>. Also available there are slides of the material for classroom use. One version of the slides is a skeleton format with key equations omitted that an instructor can complete interactively during a lecture. (This is how the first author teaches this material.)



Organization

The following diagram illustrates how the book chapters are related. The first few chapters provide a foundation that should be read in sequence. There is more flexibility in the ordering of the subsequent chapters.



## Related Books

Books we used as references when preparing this material include [3], [4], [5], and [6]. None of those books use JULIA to illustrate the ideas.

Other books that provide useful linear algebra background, also using JULIA, are [2], [7], and [8]. Those books have less depth in DS–ML–SP applications. [9] describes JULIA and uses it for some ML applications with less matrix fundamentals. Other books using JULIA for related topics include [10], [11], and [12].

There are many graduate-level books on DS–ML–SP topics that give a brief review of linear algebra concepts before delving into more advanced material. The material in this book will provide the reader with a more thorough foundation in preparation for more advanced DS–ML–SP courses.

## Acknowledgments

The authors donate a portion of their royalties to organizations that empower groups that have been historically disadvantaged in science, technology, engineering, and mathematics fields, including [ostem.org](#).

### From JF

Thanks to Prof. Zhongming Liu, Prof. Yong Long, graduate student instructors Caroline Crockett, David Hong, Steven Whitaker, Haowei Xiang, and postdocs Rodrigo Lobos, Greg Ongie, and Dan Weller, and numerous past students, including Yongli He, Winston Wang, Emma Shedden, Zicheng Jin, Ege Taga and Yixuan (Isaac) Jia, for many corrections and suggestions. Special thanks to Matt Raymond for particularly detailed and insightful suggestions that refined and clarified the content of each chapter. Thanks to Raj for starting me on this journey with his handwritten notes. The best way to learn is to teach from a colleague's insightful lecture notes!

### From RN

Special thanks to Jonas Kersulis and Brian Moore for helping create, edit, and test the many computationally centered homework problems.

This book would not be possible without them and the various graduate student instructors (Arvind Prasad, David Hong, Hao Wu, Rishi Sonthalia, Dipak Narayan, Yash Sanjay Bhargat, and Raj Tejas Suryaprakash) who helped edit, test, and refine the many homework problems, often right before they were about to go live to hundreds of students. Thanks to Simon Danisch for helping start this journey in 2017 by porting my MATLAB demos to JULIA.

Thanks in particular to Gil Strang for his encouragement, feedback, and support, and for his inspiration during the very special semester of Spring 2017 when we launched and taught 18.065 at MIT using some of the material in this book.

Multiple thanks to Alan Edelman for years of encouragement and inspiration, and for teaching me so much (including JULIA). A reader might sometimes recognize Alan and Gil's style in the way the math and code are presented. That's no accident. This book is infused with their DNA and years of me soaking in their thoughts and ideas on so many matters, particularly on how elegant math produces elegant code and vice versa. All they taught me about how to see math and linear algebra makes me love it, and to want to share it with you in this book, even more.