

Eric F. Van de Velde

Concurrent Scientific Computing

With 47 Illustrations



Springer Science+Business Media, LLC

Eric F. Van de Velde
Applied Mathematics 217-50
California Institute of Technology
Pasadena, CA 91125
USA

Series Editors

F. John
Courant Institute of
Mathematical Sciences
New York University
New York, NY 10012
USA

J.E. Marsden
Department of
Mathematics
University of California
Berkeley, CA 94720
USA

L. Sirovich
Division of Applied
Mathematics
Brown University
Providence, RI 02912
USA

M. Golubitsky
Department of Mathematics
University of Houston
Houston, TX 77204
USA

W. Jäger
Department of Applied Mathematics
Universität Heidelberg
Im Neuenheimer Feld 294
Heidelberg, FRG 6900

Mathematics Subject Classification (1992): 42A38, 65T20, 68Q10

Library of Congress Cataloging-in-Publication Data

Van de Velde, Eric F.

Concurrent scientific computing / Eric F. Van de Velde.

p. cm. — (Texts in applied mathematics ; 16)

Includes bibliographical references and index.

1. Parallel processing (Electronic computers) I. Title.

II. Series.

QA76.58.V35 1994

519.4'0285'52—dc20

93-43289

Printed on acid-free paper.

© 1994 Springer Science+Business Media New York

Originally published by Springer-Verlag New York Inc. in 1994

Softcover reprint of the hardcover 1st edition 1994

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not especially identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Production managed by Ellen Seham; manufacturing supervised by Genieve Shaw.

Photocomposed pages prepared from the author's LaTeX file.

9 8 7 6 5 4 3 2 1

ISBN 978-1-4612-6921-2

ISBN 978-1-4612-0849-5 (eBook)

DOI 10.1007/978-1-4612-0849-5

Contents

Series Preface	v
Preface	vii
Acknowledgments	ix
List of Figures	xvii
List of Tables	xxi
1 The Basics	1
1.1 Notation	2
1.2 Multicomputers	4
1.3 Data Distribution	6
1.4 Program Development	8
1.5 Communication	11
1.6 Performance Analysis	14
1.6.1 Computations and Execution Times	15
1.6.2 Speed-up: Theory	16
1.6.3 Speed-up: Practice	19
1.6.4 Performance Modeling	21
1.6.5 Vector Operations	22
1.7 The Recursive-Doubling Algorithm	26
1.7.1 Specification	26

1.7.2	Implementation	26
1.7.3	Performance Analysis	30
1.7.4	Numerical Aspects	32
1.8	Notes and References	33
1.8.1	Communicating Sequential Processes	33
1.8.2	UNITY	33
1.8.3	Stepwise Refinement	34
1.8.4	Speed-up and Efficiency	34
1.8.5	Other Performance Measures	36
	Exercises	38
2	Vector and Matrix Operations	39
2.1	Matrix-Vector Product	39
2.1.1	Specification	39
2.1.2	Implementation	40
2.1.3	Performance Analysis	44
2.1.4	Sparse Matrices	47
2.2	Vectors and Matrices	49
2.2.1	Vector Conversion	50
2.2.2	Inner Product	52
2.3	Practical Data Distributions	53
2.4	Vector Shift	55
2.4.1	Definition and Specification	55
2.4.2	Implementation	57
2.5	The Diagonal Form of Matrices	58
2.5.1	Definition	58
2.5.2	Matrix-Vector Product	60
2.5.3	Matrix Transpose	61
2.5.4	Vector-Matrix Product	62
2.5.5	Matrix-Matrix Product	62
2.5.6	Diagonal Form and Column Distribution	63
2.6	Notes and References	64
	Exercises	65
3	Iterative Methods	67
3.1	The Power Method	68
3.1.1	Numerics	68
3.1.2	Program	70
3.2	Relaxation Methods	73
3.3	The Conjugate-Gradient Method	76
3.3.1	Subspace Minimization	76
3.3.2	The Steepest-Descent Method	78
3.3.3	Krylov-Subspace Minimization	79
3.3.4	The Program	80
3.3.5	Preconditioning	83

3.3.6	Termination and Convergence	85
3.4	The Quasi-Minimal-Residual Method	86
3.4.1	Krylov-Space Bases	86
3.4.2	Quasi-Minimization in a Subspace	88
3.4.3	Updating	89
3.4.4	The Program	92
3.5	Notes and References	93
	Exercises	95
4	LU-Decomposition	97
4.1	LU-Decomposition Without Pivoting	97
4.1.1	Specification	97
4.1.2	Implementation	99
4.1.3	Performance Analysis	101
4.2	LU-Decomposition With Pivoting	103
4.2.1	Specification	103
4.2.2	Implementation	106
4.2.3	Pivot Strategies	107
4.3	Triangular Systems	111
4.3.1	Specification	112
4.3.2	Implementation	113
4.3.3	Performance Analysis	117
4.4	Linear-System Solver	118
4.5	Sparse Systems	121
4.6	Notes and References	122
	Exercises	123
5	QR-Decomposition	125
5.1	Introduction	125
5.1.1	Orthogonal Matrices	125
5.1.2	Gram-Schmidt Orthogonalization	126
5.1.3	QR-Decomposition	126
5.1.4	QR-Decomposition and Linear Systems	127
5.2	Householder QR-Decomposition	128
5.2.1	Householder Reflections	128
5.2.2	QR-Decomposition	130
5.3	Givens QR-Decomposition	131
5.3.1	Givens Rotations	131
5.3.2	QR-Decomposition	132
5.4	Multicomputer QR-Decomposition	133
5.5	Notes and References	140
	Exercises	140

6	Tridiagonal Solvers	141
6.1	Full Recursive Doubling	141
6.1.1	Specification	142
6.1.2	Implementation	144
6.1.3	Performance Analysis	145
6.2	Rational Recursion	146
6.3	Tridiagonal LU-Decomposition	147
6.3.1	Specification	147
6.3.2	Implementation	148
6.3.3	Performance Analysis	151
6.4	An Iterative Tridiagonal Solver	152
6.5	Notes and References	153
	Exercises	154
7	The Fast Fourier Transform	155
7.1	Fourier Analysis	155
7.1.1	Fourier Series	155
7.1.2	The Discrete Fourier Transform	156
7.1.3	Aliasing	158
7.2	Divide and Conquer	159
7.3	The Fast-Fourier-Transform Algorithm	161
7.4	Multicomputer Implementation	167
7.5	Generalizations	172
7.5.1	Bit-Reversal Maps	172
7.5.2	General Dimensions	172
7.5.3	The Discrete Real Fourier Transform	173
7.5.4	The Discrete Fourier-Sine Transform	174
7.5.5	The Discrete Fourier-Cosine Transform	176
7.5.6	Multivariate Discrete Fourier Transforms	177
7.6	Notes and References	181
	Exercises	181
8	Poisson Solvers	183
8.1	The Poisson Problem	183
8.2	Jacobi Relaxation	187
8.2.1	Specification	187
8.2.2	Implementation	188
8.2.3	Performance Analysis	195
8.3	Gauss-Seidel Relaxation	197
8.3.1	Specification	197
8.3.2	Implementation	198
8.3.3	Performance Analysis	199
8.3.4	Gauss-Seidel Variants	200
8.4	Line Relaxation	202
8.4.1	Specification	202

8.4.2	Implementation	204
8.5	Domain-Decomposed Relaxations	207
8.6	The Conjugate-Gradient Method	208
8.7	FFT-Based Fast Poisson Solvers	210
8.7.1	The Poisson Problem in Fourier Space	210
8.7.2	Fast Poisson Solvers	212
8.7.3	Variants	213
8.8	Three-Dimensional Problems	214
8.9	Notes and References	215
	Exercises	215
9	Multigrid Methods	217
9.1	Relaxation Methods	218
9.1.1	Model Problem	218
9.1.2	Model-Problem Analysis	219
9.1.3	Convergence Factor	220
9.1.4	Smoothing Factor	220
9.1.5	Smoothing vs. Convergence	221
9.2	Two-Grid Methods	222
9.2.1	Intuitive Development	222
9.2.2	Model-Problem Analysis	224
9.3	Multigrid Methods	227
9.3.1	Elementary Multigrid	227
9.3.2	Performance Analysis	229
9.3.3	Full Multigrid	231
9.4	Multicomputer Multigrid	232
9.4.1	Specification Programs	232
9.4.2	Multicomputer Relaxation	235
9.4.3	Multicomputer Restriction	236
9.4.4	Multicomputer Interpolation	237
9.4.5	Multicomputer Multigrid	239
9.5	Notes and References	239
	Exercises	240
10	Domain Decomposition	241
10.1	Composite Domains	242
10.1.1	The Capacitance System	242
10.1.2	Capacitance-System Solvers	244
10.2	Domain Decomposition for Concurrency	246
10.2.1	Domain Decomposition	247
10.2.2	Data Distribution	249
10.2.3	Vector Operations	251
10.2.4	Sequential Poisson Solver	254
10.2.5	Matrix Operations	254
10.2.6	The Domain-Decomposition Program	257

10.3 Performance Analysis	260
10.4 Notes and References	263
Exercises	263
11 Particle Methods	265
11.1 The Astrophysical N -Body Problem	266
11.2 The N^2 Algorithm	267
11.3 Fast Particle Methods	270
11.4 Concurrent Fast Particle Methods	275
11.5 Notes and References	280
Exercises	281
12 Computer Dependency	283
12.1 Pipelined Processors	284
12.2 Hierarchical Memories	286
12.3 Communication Networks	290
12.3.1 Hypercubes and Meshes	291
12.3.2 Process Placement	292
12.3.3 A Transpose Algorithm	295
12.3.4 Communication Hiding	298
12.4 Multiprocessors	301
12.5 Synchronous Arrays of Processors	304
12.6 Benchmarking	306
12.7 Notes and References	308
Exercises	308
Bibliography	311
Index	319

11.1	The field at $\vec{x}$ generated by a point mass of size w_m at $\vec{x}_m(t)$	266
11.2	The body data structure.	270
11.3	The cell data structure.	271
11.4	Clustering a cell.	272
11.5	A domain split orthogonal to the x -axis followed by a domain split orthogonal to the y -axis.	278
12.1	Prototypical Von Neumann design of a sequential computer.	284
12.2	Prototypical floating-point pipeline with eight stages at the start ($t = 2\tau$), in the middle ($t = n\tau$), and at the end ($t = (L + 7)\tau$) of a vector operation.	285
12.3	Prototypical cache design.	287
12.4	Hypercube graphs of dimension 0 through 3.	291
12.5	Process placement on a hypercube multicomputer according to the binary-reflected Gray code and network channels used to communicate between processes p and $(p + 2^d) \bmod 2^D$	294
12.6	Matrix transpose by subblocks with $P = 4$ processes.	296
12.7	Three stages of a matrix transpose by recursive doubling.	297
12.8	Prototypical multiprocessor.	300
12.9	Prototypical multiprocessor, but local memory is associated with every processor.	302
12.10	Prototypical synchronous array of processors.	305

List of Tables

2.1	A comparison of linear and scatter distributions for $M = 11$ and $P = 4$	55
12.1	Derivation of the binary-reflected Gray-code sequence. . . .	293