

Contents

Preface	<i>page xi</i>
PART I MACHINES AND COMPUTATION	1
1 Introduction – The Nature of High-Performance Computation	3
1.1 Computing Hardware Has Undergone Vast Improvement	4
1.2 SIMD–Vector Computers	9
1.3 MIMD – True, Coarse-Grain Parallel Computers	14
1.4 Classification of Distributed Memory Computers	16
1.5 Amdahl’s Law and Profiling	20
Exercises	23
2 Theoretical Considerations – Complexity	27
2.1 Directed Acyclic Graph Representation	27
2.2 Some Basic Complexity Calculations	33
2.3 Data Dependencies	37
2.4 Programming Aids	41
Exercises	41
3 Machine Implementations	44
3.1 Early Underpinnings of Parallelization – Multiple Processes	45
3.2 Threads Programming	50
3.3 Java Threads	56
3.4 SGI Threads and OpenMP	63
3.5 MPI	67

viii	<i>Contents</i>	
3.6	Vector Parallelization on the Cray	80
3.7	Quantum Computation	89
	Exercises	97
PART II	LINEAR SYSTEMS	101
4	Building Blocks – Floating Point Numbers and Basic Linear Algebra	103
4.1	Floating Point Numbers and Numerical Error	104
4.2	Round-off Error Propagation	110
4.3	Basic Matrix Arithmetic	113
4.4	Operations with Banded Matrices	120
	Exercises	124
5	Direct Methods for Linear Systems and LU Decomposition	126
5.1	Triangular Systems	126
5.2	Gaussian Elimination	134
5.3	<i>ijk</i> -Forms for LU Decomposition	150
5.4	Bordering Algorithm for LU Decomposition	155
5.5	Algorithm for Matrix Inversion in $\log^2 n$ Time	156
	Exercises	158
6	Direct Methods for Systems with Special Structure	162
6.1	Tridiagonal Systems – Thompson’s Algorithm	162
6.2	Tridiagonal Systems – Odd–Even Reduction	163
6.3	Symmetric Systems – Cholesky Decomposition	166
	Exercises	170
7	Error Analysis and QR Decomposition	172
7.1	Error and Residual – Matrix Norms	172
7.2	Givens Rotations	180
	Exercises	184
8	Iterative Methods for Linear Systems	186
8.1	Jacobi Iteration or the Method of Simultaneous Displacements	186
8.2	Gauss–Seidel Iteration or the Method of Successive Displacements	189
8.3	Fixed-Point Iteration	191

<i>Contents</i>	<i>ix</i>
8.4 Relaxation Methods	193
8.5 Application to Poisson’s Equation	194
8.6 Parallelizing Gauss–Seidel Iteration	198
8.7 Conjugate Gradient Method	200
Exercises	204
9 Finding Eigenvalues and Eigenvectors	206
9.1 Eigenvalues and Eigenvectors	206
9.2 The Power Method	209
9.3 Jordan Canonical Form	210
9.4 Extensions of the Power Method	215
9.5 Parallelization of the Power Method	217
9.6 The QR Method for Eigenvalues	217
9.7 Householder Transformations	221
9.8 Hessenberg Form	226
Exercises	227
PART III MONTE CARLO METHODS	231
10 Monte Carlo Simulation	233
10.1 Quadrature (Numerical Integration)	233
Exercises	242
11 Monte Carlo Optimization	244
11.1 Monte Carlo Methods for Optimization	244
11.2 IIP Parallel Search	249
11.3 Simulated Annealing	251
11.4 Genetic Algorithms	255
11.5 Iterated Improvement Plus Random Restart	258
Exercises	262
APPENDIX: PROGRAMMING EXAMPLES	265
MPI Examples	267
Send a Message Sequentially to All Processors (C)	267
Send and Receive Messages (Fortran)	268
Fork Example	270
Polynomial Evaluation(C)	270

x	<i>Contents</i>	
Lan Example		275
Distributed Execution on a LAN(C)		275
Threads Example		280
Dynamic Scheduling(C)		280
SGL Example		282
Backsub.f(Fortran)		282
References		285
Index		286