

# Contents

## 1 Introduction

1.1	Introduction .....	1
1.2	Graph theory .....	2
1.3	Example .....	4
1.4	Advanced computer architectures.....	9
1.5	Problem formulation .....	11
1.6	A sparse matrix test collection.....	14

## 2 Sparse matrices: storage schemes and simple operations

2.1	Introduction .....	19
2.2	Sparse vector storage.....	19
2.3	Inner product of two packed vectors .....	20
2.4	Adding packed vectors .....	21
2.5	Use of full-sized arrays .....	22
2.6	Coordinate scheme for storing sparse matrices.....	23
2.7	Sparse matrix as a collection of sparse vectors .....	24
2.8	Linked lists .....	25
2.9	Sparse matrix in row-linked list .....	28
2.10	Generating collections of sparse vectors .....	30
2.11	Access by rows and columns .....	31
2.12	Matrix by vector products.....	32
2.13	Matrix by matrix products .....	33
2.14	Permutation matrices.....	34
2.15	Clique (or finite-element) storage.....	36
2.16	Comparisons between sparse matrix structures.....	37

## 3 Gaussian elimination for dense matrices: the algebraic problem

3.1	Introduction .....	41
3.2	Solution of triangular systems .....	41
3.3	Gaussian elimination.....	43
3.4	Required row interchanges.....	45
3.5	Relationship with $LU$ factorization.....	46
3.6	Equivalent factorization methods, including Doolittle and Crout .....	49
3.7	Computational sequences .....	51
3.8	Symmetric matrices .....	53
3.9	Gauss-Jordan elimination .....	55
3.10	Multiple right-hand sides and inverses .....	56
3.11	Computational cost .....	57
3.12	Block factorization.....	59

## 4 Gaussian elimination for dense matrices: numerical considerations

4.1	Introduction .....	65
4.2	Computer arithmetic error .....	66
4.3	Algorithm instability .....	67
4.4	Controlling algorithm stability .....	69
4.5	Monitoring the stability .....	72
4.6	Special stability considerations .....	76
4.7	Solving indefinite symmetric systems .....	77
4.8	Ill-conditioning: introduction .....	78
4.9	Ill-conditioning: theoretical discussion .....	79
4.10	Ill-conditioning: automatic detection .....	81
4.11	Iterative refinement .....	85
4.12	Scaling .....	86
4.13	Prevention of ill-conditioning .....	88
4.14	Overall error bound .....	89

## 5 Gaussian elimination for sparse matrices: an introduction

5.1	Introduction .....	93
5.2	Features of a sparse matrix code .....	94
5.3	Orderings .....	96
5.4	Numerical stability in sparse computation .....	98
5.5	Estimating condition numbers in sparse computation .....	100
5.6	The ANALYSE phase .....	101
5.7	The FACTORIZE and SOLVE phases .....	102
5.8	Writing compared with using sparse matrix software .....	103

## 6 Reduction to block triangular form

6.1	Introduction .....	105
6.2	Background for finding a transversal .....	107
6.3	A depth-first search algorithm for transversal extension .....	109
6.4	Analysis of the depth-first search transversal algorithm .....	111
6.5	Implementation of the transversal algorithm .....	112
6.6	Symmetric permutations to block triangular form .....	113
6.7	The algorithm of Sargent and Westerberg .....	114
6.8	Tarjan's algorithm .....	117
6.9	Implementation of Tarjan's algorithm .....	120
6.10	Essential uniqueness of the block triangular form .....	121
6.11	Experience with block triangular forms .....	122
6.12	Maximum transversals .....	124

**7 Local pivotal strategies for sparse matrices**

7.1	Introduction .....	127
7.2	The Markowitz criterion.....	128
7.3	Minimum degree (Tinney scheme 2) .....	129
7.4	Simpler strategies.....	130
7.5	Local minimum fill-in .....	133
7.6	Other local strategies.....	135
7.7	The effect of tie-breaking on the minimum degree algorithm .....	136
7.8	Numerical pivoting .....	139
7.9	Sparsity in the right-hand side and partial solution .....	140
7.10	Variability-type ordering.....	143
7.11	The symmetric indefinite case .....	145
7.12	Solution methods based on orthogonalization.....	145

**8 Ordering sparse matrices to special forms**

8.1	Introduction .....	149
8.2	Desirable forms .....	149
8.3	Band and variable-band matrices .....	151
8.4	Ordering for small bandwidth .....	153
8.5	Refined quotient trees.....	157
8.6	One-way dissection .....	161
8.7	Nested dissection .....	163
8.8	The Boeing version of the Hellerman-Rarick algorithm .....	168
8.9	The Hellerman-Rarick ordering .....	172

**9 Implementing Gaussian elimination: ANALYSE with numerical values**

9.1	Introduction .....	177
9.2	Markowitz ANALYSE.....	178
9.3	FACTORIZE .....	183
9.4	SOLVE .....	186
9.5	Switching to full form .....	188
9.6	Loop-free code.....	191
9.7	Interpretative code .....	192
9.8	A posteriori ordering for stability .....	193
9.9	The use of drop tolerances to preserve sparsity .....	195

**10 Implementing Gaussian elimination with symbolic ANALYSE**

10.1	Introduction .....	199
10.2	Band methods.....	200
10.3	Variable-band (profile) methods.....	204
10.4	Special methods for banded systems.....	205
10.5	Frontal methods for finite-element problems .....	210
10.6	Frontal methods for non-element problems .....	216
10.7	Multifrontal (substructuring) methods for finite-element problems .....	218
10.8	Multifrontal methods for non-element problems .....	222
10.9	Minimum degree ordering .....	226
10.10	Generating static data structures for FACTORIZE .....	229
10.11	Numerical FACTORIZE and SOLVE using static data structures.....	232
10.12	Accommodating numerical pivoting within static data structures.....	233
10.13	Nested dissection, refined quotient tree, and one-way dissection.....	234
10.14	Effects of computer architecture .....	234

**11 Partitioning, matrix modification, and tearing**

11.1	Introduction .....	239
11.2	Exploiting the partitioned form.....	240
11.3	Using partitioning to handle low-rank perturbations .....	242
11.4	The matrix modification formula.....	243
11.5	Low-rank modifications .....	244
11.6	Use of the modification formula to solve equations.....	245
11.7	Perturbations that are full-rank submatrices .....	247
11.8	The equivalence of modification and partitioning .....	249
11.9	Modifications that change the matrix order.....	250
11.10	Artificial perturbations .....	251
11.11	Branch tearing .....	252
11.12	Node tearing.....	254
11.13	Implementation of node and branch tearing .....	255
11.14	Manual choice of partitions and tear sets.....	255
11.15	Automatic choice of partitions and tear sets .....	257
11.16	Nonlinear tearing.....	259
11.17	Conclusions.....	260

<b>12 Other sparsity-oriented issues</b>	
12.1 Introduction .....	265
12.2 Sparsity in nonlinear computations .....	266
12.3 Estimating a sparse Jacobian matrix .....	267
12.4 Updating a sparse Hessian matrix .....	269
12.5 Approximating a sparse matrix by a positive-definite one	270
12.6 The inverse of a sparse matrix .....	271
12.7 Computing entries of the inverse of a sparse matrix .....	273
12.8 Sparsity constrained backward error analysis .....	275
12.9 Assembling large finite-element matrices to band form .....	276
12.10 Hypermatrices .....	278
<b>Appendix A: Matrix and vector norms</b> .....	281
<b>Appendix B: The LINPACK condition number estimate</b> .....	285
<b>Appendix C: Fortran conventions</b> .....	287
<b>Appendix D: Pictures of sparse matrices</b> .....	289
<b>Solutions to selected exercises</b> .....	297
<b>References</b> .....	313
<b>Author index</b> .....	327
<b>Subject index</b> .....	331