Contents

Rules ix

Preface xi

Prologue xv

Mathematical Symbols xix

1 Introduction 1

I Tangents and Gradients 13

2 A Framework for Evaluating Functions 15
2.1 The Lighthouse Example 16
2.2 Three-Part Evaluation Procedures 18
2.3 Elemental Differentiability 23
2.4 Generalizations and Connections 25
2.5 Examples and Exercises 29

3 Fundamentals of Forward and Reverse 31
3.1 Forward Propagation of Tangents 32
3.2 Reverse Propagation of Gradients 37
3.3 Cheap Gradient Principle with Examples 44
3.4 Approximate Error Analysis 50
3.5 Summary and Discussion 52
3.6 Examples and Exercises 56

4 Memory Issues and Complexity Bounds 61
4.1 Memory Allocation and Overwrites 61
4.2 Recording Prevalues on the Tape 64
4.3 Selective Saves and Restores 70
4.4 A Temporal Complexity Model 73
4.5 Complexity of Tangent Propagation 80
4.6 Complexity of Gradient Propagation 83
4.7 Examples and Exercises 88

5 Repeating and Extending Reverse **91**
5.1 Adjoining Iterative Assignments 93
5.2 Adjoints of Adjoints 95
5.3 Tangents of Adjoints 98
5.4 Complexity of Second-Order Adjoints 102
5.5 Examples and Exercises 105

6 Implementation and Software **107**
6.1 Operator Overloading 110
6.2 Source Transformation 120
6.3 AD for Parallel Programs 129
6.4 Summary and Outlook 138
6.5 Examples and Exercises 139

II Jacobians and Hessians **143**

7 Sparse Forward and Reverse **145**
7.1 Quantifying Structure and Sparsity 147
7.2 Sparse Derivative Propagation 151
7.3 Sparse Second Derivatives 154
7.4 Examples and Exercises 159

8 Exploiting Sparsity by Compression **161**
8.1 Curtis–Powell–Reid Seeding 164
8.2 Newsam–Ramsdell Seeding 168
8.3 Column Compression 171
8.4 Combined Column and Row Compression 173
8.5 Second Derivatives, Two-Sided Compression 176
8.6 Summary 181
8.7 Examples and Exercises 182

9 Going beyond Forward and Reverse **185**
9.1 Jacobians as Schur Complements 188
9.2 Accumulation by Edge-Elimination 193
9.3 Accumulation by Vertex-Elimination 200
9.4 Face-Elimination on the Line-Graph 204
9.5 NP-hardness via Ensemble Computation 207
9.6 Summary and Outlook 209
9.7 Examples and Exercises 209

10 Jacobian and Hessian Accumulation **211**
10.1 Greedy and Other Heuristics 211
10.2 Local Preaccumulations 220
10.3 Scarcity and Vector Products 225
10.4 Hessians and Their Computational Graph 236

10.5 Examples and Exercises 241

11 Observations on Efficiency **245**
11.1 Ramification of the Rank-One Example 245
11.2 Remarks on Partial Separability 248
11.3 Advice on Problem Preparation 255
11.4 Examples and Exercises 257

III Advances and Reversals **259**

12 Reversal Schedules and Checkpointing **261**
12.1 Reversal with Recording or Recalculation 262
12.2 Reversal of Call Trees 265
12.3 Reversals of Evolutions by Checkpointing 278
12.4 Parallel Reversal Schedules 295
12.5 Examples and Exercises 298

13 Taylor and Tensor Coefficients **299**
13.1 Higher-Order Derivative Vectors 300
13.2 Taylor Polynomial Propagation 303
13.3 Multivariate Tensor Evaluation 311
13.4 Higher-Order Gradients and Jacobians 317
13.5 Special Relations in Dynamical Systems 326
13.6 Summary and Outlook 332
13.7 Examples and Exercises 333

14 Differentiation without Differentiability **335**
14.1 Another Look at the Lighthouse Example 337
14.2 Putting the Pieces Together 341
14.3 One-Sided Laurent Expansions 350
14.4 Summary and Conclusion 362
14.5 Examples and Exercises 363

15 Implicit and Iterative Differentiation **367**
15.1 Results of the Implicit Function Theorem 370
15.2 Iterations and the Newton Scenario 375
15.3 Direct Derivative Recurrences 381
15.4 Adjoint Recurrences and Their Convergence 386
15.5 Second-Order Adjoints 390
15.6 Summary and Outlook 393
15.7 Examples and Exercises 394

Epilogue **397**

List of Figures **399**

List of Tables 403

Assumptions and Definitions 407

Propositions, Corollaries, and Lemmas 409

Bibliography 411

Index 433

Rules

0	ALGORITHMIC DIFFERENTIATION DOES NOT INCUR TRUNCATION ERRORS.	2
1	DIFFERENCE QUOTIENTS MAY SOMETIMES BE USEFUL TOO.	4
2	WHAT'S GOOD FOR FUNCTION VALUES IS GOOD FOR THEIR DERIVATIVES.	20
3	THE JACOBIAN-VECTOR AND JACOBIAN TRANSPOSED VECTOR PRODUCTS CALCULATED IN THE FORWARD AND REVERSE MODE, RESPECTIVELY, CORRESPOND TO THE EXACT VALUES FOR AN *EVALUATION PROCEDURE WHOSE ELEMENTALS ARE PERTURBED* AT THE LEVEL OF THE MACHINE PRECISION.	51
4	PRODUCTS OF TRANSPOSE-JACOBIANS WITH VECTORS ARE CHEAP, BUT THEIR TRACE IS AS EXPENSIVE AS THE WHOLE JACOBIAN.	56
5	ADJOINING CAN BE PERFORMED LINE BY LINE IRRESPECTIVE OF DEPENDENCIES AND ALIASING.	69
6	SAVE ONLY THE VALUES OF POWERS OR EXPONENTIALS AND THE ARGUMENTS OF OTHER NONLINEAR ELEMENTALS.	71
7	ADDITIVE TASKS WITH BOUNDED COMPLEXITY ON ELEMENTALS ARE ALSO BOUNDED ON COMPOSITE FUNCTIONS.	79
8	OPERATION COUNTS AND RANDOM ACCESS MEMORY REQUIREMENTS FOR THEIR GRADIENTS ARE BOUNDED MULTIPLES OF THOSE FOR THE FUNCTIONS.	88
9	THE SEQUENTIAL ACCESS MEMORY REQUIREMENT OF THE BASIC REVERSE MODE IS PROPORTIONAL TO THE TEMPORAL COMPLEXITY OF THE FUNCTION.	88
10	NEVER GO BACK MORE THAN ONCE (AFTER ALL WEIGHT VECTORS ARE KNOWN).	92
11	INCREMENTAL ADJOINTS OF LINEAR PROCEDURES ARE REFLEXIVE IRRESPECTIVE OF ALLOCATION.	98
12	DO AS MUCH AS POSSIBLE AT COMPILE-TIME, AS MUCH AS NECESSARY AT RUNTIME.	139
13	FACE-ELIMINATION ALLOWS LOWER OPERATIONS COUNTS THAN EDGE-ELIMINATION, WHICH IS IN TURN CHEAPER THAN VERTEX-ELIMINATION.	207

14 DO NOT USE GREEDY HEURISTICS BASED ON THE MARKOWITZ DEGREE FOR COMPUTING THE JACOBIANS OF FINAL STATES WITH RESPECT TO INITIAL STATES IN EVOLUTIONS. 218

15 LOCAL JACOBIANS SHOULD BE PREACCUMULATED IF THEIR GENERIC RANK IS SIGNIFICANTLY SMALLER THAN THE NUMBER OF DERIVATIVE OR ADJOINT COMPONENTS BEING PROPAGATED GLOBALLY. 222

16 THE CALCULATION OF GRADIENTS BY NONINCREMENTAL REVERSE MAKES THE CORRESPONDING COMPUTATIONAL GRAPH SYMMETRIC, A PROPERTY THAT SHOULD BE EXPLOITED AND MAINTAINED IN ACCUMULATING HESSIANS. 240

17 THE CHEAP GRADIENT RESULT DOES NOT YIELD CHEAP JACOBIANS AS WELL. 247

18 REVERSE BEATS FORWARD ONLY WHEN THE NONLINEAR WIDTH IS MUCH LARGER THAN THE NONLINEAR HEIGHT. 255

19 JOINT REVERSALS SAVE MEMORY, SPLIT REVERSALS SAVE OPERATIONS. 270

20 DURING CALL TREE REVERSALS ALL TAPING OCCURS IN A LAST-IN-FIRST-OUT FASHION. 275

21 THE RELATIVE REVERSAL COST (r) GROWS LIKE THE CHAIN LENGTH (l) RAISED TO THE RECIPROCAL OF THE NUMBER OF CHECKPOINTS AVAILABLE (c). 286

22 FORWARD DIFFERENTIATION CAN BE ENCAPSULATED IN TAYLOR ARITHMETIC. 310

23 WHAT WORKS FOR FIRST DERIVATIVES IS FIT TO YIELD HIGHER DERIVATIVES BY EXTENSION TO TAYLOR ARITHMETIC. 326

24 WHEN PROGRAM BRANCHES APPLY ON OPEN SUBDOMAINS, ALGORITHMIC DIFFERENTIATION YIELDS USEFUL DERIVATIVES. 343

25 FUNCTIONS GIVEN BY EVALUATION PROCEDURES ARE ALMOST EVERYWHERE REAL ANALYTIC OR STABLY UNDEFINED. 349

26 FIXED POINT ITERATIONS CAN AND SHOULD BE ADJOINED BY RECORDING ONLY SINGLE STEPS. 389