Jean-Michel Muller

Elementary Functions

Algorithms and Implementation

Second Edition

Birkhäuser
Boston • Basel • Berlin

Contents

List of Figures xi

List of Tables xv

Preface to the Second Edition xix

Preface to the First Edition xxi

1 Introduction 1

2 Some Basic Things About Computer Arithmetic 9

2.1 Floating-Point Arithmetic 9
- 2.1.1 Floating-point formats 9
- 2.1.2 Rounding modes 11
- 2.1.3 Subnormal numbers and exceptions 13
- 2.1.4 ULPs 14
- 2.1.5 Fused multiply-add operations 15
- 2.1.6 Testing your computational environment 16
- 2.1.7 Floating-point arithmetic and proofs 17
- 2.1.8 Maple programs that compute double-precision approximations 17

2.2 Redundant Number Systems 19
- 2.2.1 Signed-digit number systems 19
- 2.2.2 Radix-2 redundant number systems 21

I Algorithms Based on Polynomial Approximation and/or Table Lookup, Multiple-Precision Evaluation of Functions 25

3 Polynomial or Rational Approximations 27

3.1 Least Squares Polynomial Approximations 28
- 3.1.1 Legendre polynomials 29
- 3.1.2 Chebyshev polynomials 29
- 3.1.3 Jacobi polynomials 31

3.1.4 Laguerre polynomials 31
3.1.5 Using these orthogonal polynomials in any interval 31
3.2 Least Maximum Polynomial Approximations 32
3.3 Some Examples 33
3.4 Speed of Convergence 39
3.5 Remez's Algorithm 41
3.6 Rational Approximations 46
3.7 Actual Computation of Approximations 50
3.7.1 Getting "general" approximations 50
3.7.2 Getting approximations with special constraints 51
3.8 Algorithms and Architectures for the Evaluation of Polynomials . 54
3.8.1 The E-method 57
3.8.2 Estrin's method 58
3.9 Evaluation Error Assuming Horner's Scheme is Used 59
3.9.1 Evaluation using floating-point additions and multiplications 60
3.9.2 Evaluation using fused multiply-accumulate instructions 64
3.10 Miscellaneous 66

4 Table-Based Methods 67
4.1 Introduction 67
4.2 Table-Driven Algorithms 70
4.2.1 Tang's algorithm for $\exp(x)$ in IEEE floating-point arithmetic 71
4.2.2 $\ln(x)$ on $[1, 2]$ 72
4.2.3 $\sin(x)$ on $[0, \pi/4]$ 73
4.3 Gal's Accurate Tables Method 73
4.4 Table Methods Requiring Specialized Hardware 77
4.4.1 Wong and Goto's algorithm for computing logarithms 78
4.4.2 Wong and Goto's algorithm for computing exponentials 81
4.4.3 Ercegovac et al.'s algorithm 82
4.4.4 Bipartite and multipartite methods 83
4.4.5 Miscellaneous 87

5 Multiple-Precision Evaluation of Functions 89
5.1 Introduction 89
5.2 Just a Few Words on Multiple-Precision Multiplication 90
5.2.1 Karatsuba's method 91
5.2.2 FFT-based methods 92
5.3 Multiple-Precision Division and Square-Root 92
5.3.1 Newton–Raphson iteration 92

5.4 Algorithms Based on the Evaluation of Power Series . . . 94
5.5 The Arithmetic-Geometric (AGM) Mean . . . 95
5.5.1 Presentation of the AGM . . . 95
5.5.2 Computing logarithms with the AGM . . . 95
5.5.3 Computing exponentials with the AGM . . . 98
5.5.4 Very fast computation of trigonometric functions . . . 98

II Shift-and-Add Algorithms 101

6 Introduction to Shift-and-Add Algorithms 103
6.1 The Restoring and Nonrestoring Algorithms . . . 105
6.2 Simple Algorithms for Exponentials and Logarithms . . . 109
6.2.1 The restoring algorithm for exponentials . . . 109
6.2.2 The restoring algorithm for logarithms . . . 111
6.3 Faster Shift-and-Add Algorithms . . . 113
6.3.1 Faster computation of exponentials . . . 113
6.3.2 Faster computation of logarithms . . . 119
6.4 Baker's Predictive Algorithm . . . 122
6.5 Bibliographic Notes . . . 131

7 The CORDIC Algorithm 133
7.1 Introduction . . . 133
7.2 The Conventional CORDIC Iteration . . . 134
7.3 Scale Factor Compensation . . . 139
7.4 CORDIC With Redundant Number Systems and a Variable Factor 141
7.4.1 Signed-digit implementation . . . 142
7.4.2 Carry-save implementation . . . 143
7.4.3 The variable scale factor problem . . . 143
7.5 The Double Rotation Method . . . 144
7.6 The Branching CORDIC Algorithm . . . 146
7.7 The Differential CORDIC Algorithm . . . 150
7.8 Computation of $\cos^{-1}$ and $\sin^{-1}$ Using CORDIC . . . 153
7.9 Variations on CORDIC . . . 156

8 Some Other Shift-and-Add Algorithms 157
8.1 High-Radix Algorithms . . . 157
8.1.1 Ercegovac's radix-16 algorithms . . . 157
8.2 The BKM Algorithm . . . 162
8.2.1 The BKM iteration . . . 162
8.2.2 Computation of the exponential function (E-mode) . . . 162
8.2.3 Computation of the logarithm function (L-mode) . . . 166

8.2.4 Application to the computation of elementary functions 167

III Range Reduction, Final Rounding and Exceptions 171

9 Range Reduction 173
9.1 Introduction 173
9.2 Cody and Waite's Method for Range Reduction 177
9.3 Finding Worst Cases for Range Reduction? 179
9.3.1 A few basic notions on continued fractions 179
9.3.2 Finding worst cases using continued fractions 180
9.4 The Payne and Hanek Reduction Algorithm 184
9.5 The Modular Range Reduction Algorithm 187
9.5.1 Fixed-point reduction 188
9.5.2 Floating-point reduction 190
9.5.3 Architectures for modular reduction 190
9.6 Alternate Methods 191

10 Final Rounding 193
10.1 Introduction 193
10.2 Monotonicity 194
10.3 Correct Rounding: Presentation of the Problem 195
10.4 Some Experiments 198
10.5 A "Probabilistic" Approach to the Problem 198
10.6 Upper Bounds on m 202
10.7 Obtained Worst Cases for Double-Precision 203
10.7.1 Special input values 203
10.7.2 Lefèvre's experiment 203

11 Miscellaneous 217
11.1 Exceptions 217
11.1.1 NaNs 218
11.1.2 Exact results 218
11.2 Notes on x^y 220
11.3 Special Functions, Functions of Complex Numbers 222

12 Examples of Implementation 225
12.1 Example 1: The Cyrix FastMath Processor 225
12.2 The INTEL Functions Designed for the Itanium Processor 226
12.2.1 Sine and cosine 227
12.2.2 Arctangent 228
12.3 The LIBULTIM Library 229
12.4 The CRLIBM Library 229
12.4.1 Computation of $\sin(x)$ or $\cos(x)$ (quick phase) 230

12.4.2 Computation of $\ln(x)$ 230

12.5 SUN's LIBMCR Library 231

12.6 The HP-UX Compiler for the Itanium Processor 231

Bibliography **233**

Index **261**