

Numerical Algorithms and Digital Representation

Knut Mørken

November 11, 2010

Preface

These lecture notes form part of the syllabus for the first-semester course MAT-INF1100 at the University of Oslo. The topics roughly cover two main areas: *Numerical algorithms*, and what can be termed *digital understanding*. Together with a thorough understanding of calculus and programming, this is knowledge that students in the mathematical sciences should gain as early as possible in their university career. As subjects such as physics, meteorology and statistics, as well as many parts of mathematics, become increasingly dependent on computer calculations, this training is essential.

Our aim is to train students who should not only be able *to use* a computer for mathematical calculations; they should also have a basic understanding of *how* the computational methods work. Such understanding is essential both in order to judge the quality of computational results, and in order to develop new computational methods when the need arises.

In these notes we cover the basic numerical algorithms such as interpolation, numerical root finding, differentiation and integration, as well as numerical solution of ordinary differential equations. In the area of digital understanding we discuss digital representation of numbers, text, sound and images. In particular, the basics of lossless compression algorithms with Huffman coding and arithmetic coding is included.

A basic assumption throughout the notes is that the reader either has attended a basic calculus course in advance or is attending such a course while studying this material. Basic familiarity with programming is also assumed. However, I have tried to quote theorems and other results on which the presentation rests. Provided you have an interest and curiosity in mathematics, it should therefore not be difficult to read most of the material with a good mathematics background from secondary school.

MAT-INF1100 is a central course in the project *Computers in Science Edu-*

cation (CSE) at the University of Oslo. The aim of this project is to make sure that students in the mathematical sciences get a unified introduction to computational methods as part of their undergraduate studies. The basic foundation is laid in the first semester with the calculus course, MAT1100, and the programming course INF1100, together with MAT-INF1100. The maths courses that follow continue in the same direction and discuss a number of numerical algorithms in linear algebra and related areas, as well as applications such as image compression and ranking of web pages.

Some fear that a thorough introduction of computational techniques in the mathematics curriculum will reduce the students' basic mathematical abilities. This could easily be true if the use of computations only amounted to running code written by others. However, deriving the central algorithms, programming them, and studying their convergence properties, should lead to a level of mathematical understanding that should certainly match that of a more traditional approach.

Many people have helped develop these notes which have matured over a period of ten years. Solveig Bruvoll, Marit Sandstad and Øyvind Ryan have helped directly with recent versions, while Pål Hermunn Johansen provided extensive programming help with an earlier version. For this latest version, Andreas Våvang Solbrå has provided important assistance and feedback. Geir Pedersen was my co-lecturer for four years. He was an extremely good discussion partner on all the facets of this material, and influenced the list of contents in several ways. I work at the Centre of Mathematics for Applications (CMA) at the University of Oslo, and I am grateful to the director, Ragnar Winther, for his enthusiastic support of the CSE project and my extensive undertakings in teaching. Over many years, my closest colleagues Geir Dahl, Michael Floater, and Tom Lyche have shaped my understanding of numerical analysis and allowed me to spend considerably more time than usual on elementary teaching. Another colleague, Sverre Holm, has been my source of information on signal processing. To all of you: thank you!

My academic home, the Department of Informatics and its chairman Morten Dæhlen, has been very supportive of this work by giving me the freedom to extend the Department's teaching duties, and by extensive support of the CSE-project. It has been a pleasure to work with the Department of Mathematics over the past eight years, and I have many times been amazed by how much confidence they seem to have in me. I have learnt a lot, and have thoroughly enjoyed teaching at the cross-section between mathematics and computing which is my scientific home. I can only say thank you, and I feel at home in both departments.

A course like MAT-INF1100 is completely dependent on support from other

courses. Tom Lindstrøm has done a tremendous job with the parallel calculus course MAT1100, and its sequel MAT1110 on multivariate analysis and linear algebra. Hans Petter Langtangen has done an equally impressive job with INF1100, the introductory programming course with a mathematical and scientific flavour, and I have benefited from many hours of discussions with both of them. Morten Hjorth-Jensen, Arnt-Inge Vistnes and Anders MaltheSørensen with colleagues have introduced a computational perspective in a number of physics courses, and discussions with them have convinced me of the importance of introducing computations for all students in the mathematical sciences. Thank you to all of you.

The CSE project is run by a group of people: Morten Hjorth-Jensen and Anders Malthe-Sørensen from the Physics Department, Hans Petter Langtangen from the Simula Research Lab and the Department of Informatics, Øyvind Ryan from the CMA, Annik Myhre (Dean of Education at the MN-faculty¹), Hanne Sølna (Head of the teaching section at the MN-faculty¹), Helge Galdal (Administrative Leader of the CMA), and myself. This group of people has been the main source of inspiration for this work, and without you, there would still only be uncoordinated attempts at including computations in our elementary courses. Thank you for all the fun we have had.

The CSE project has become much more than I could ever imagine, and the reason is that there seems to be a genuine collegial atmosphere at the University of Oslo in the mathematical sciences. This means that it has been possible to build momentum in a common direction not only within a research group, but across several departments, which seems to be quite unusual in the academic world. Everybody involved in the CSE project is responsible for this, and I can only thank you all.

Finally, as in all teaching endeavours, the main source of inspiration is the students, without whom there would be no teaching. Many students become frustrated when their understanding of the nature of mathematics is challenged, but the joy of seeing the excitement in their eyes when they understand something new is a constant source of satisfaction.

Tønsberg, September 2010

Knut Mørken

¹The faculty of Mathematics and Natural Sciences.

Contents

1	Introduction	1
1.1	A bit of history	1
1.2	Computers and different types of information	3
1.2.1	Text	4
1.2.2	Sound	4
1.2.3	Images	4
1.2.4	Film	4
1.2.5	Geometric form	5
1.2.6	Laws of nature	5
1.2.7	Virtual worlds	5
1.2.8	Summary	6
1.3	Computation by hand and by computer	7
1.4	Algorithms	8
1.4.1	Statements	10
1.4.2	Variables and assignment	11
1.4.3	For-loops	12
1.4.4	If-tests	13
1.4.5	While-loops	14
1.4.6	Print statement	14
1.5	Doing computations on a computer	15
1.5.1	How can computers be used for calculations?	15
1.5.2	What do you need to know?	16
1.5.3	Different computing environments	17

I	Numbers	21
2	0 and 1	23
2.1	Robust communication	23
2.2	Why 0 and 1 in computers?	24
2.3	True and False	25
2.3.1	Logical variables and logical operators	26
2.3.2	Combinations of logical operators	29
3	Numbers and Numeral Systems	31
3.1	Terminology and Notation	31
3.2	Natural Numbers in Different Numeral Systems	33
3.2.1	Alternative Numeral Systems	33
3.2.2	Conversion to the Base- β Numeral System	36
3.2.3	Conversion between base-2 and base-16	39
3.3	Representation of Fractional Numbers	42
3.3.1	Rational and Irrational Numbers in Base- β	42
3.3.2	An Algorithm for Converting Fractional Numbers	46
3.3.3	Conversion between binary and hexadecimal	47
3.3.4	Properties of Fractional Numbers in Base- β	49
3.4	Arithmetic in Base β	53
3.4.1	Addition	53
3.4.2	Subtraction	54
3.4.3	Multiplication	54
4	Computers, Numbers, and Text	57
4.1	Representation of Integers	57
4.1.1	Bits, bytes and numbers	58
4.1.2	Fixed size integers	59
4.1.3	Two's complement	60
4.1.4	Integers in Java	61
4.1.5	Integers in Python	62
4.1.6	Division by zero	63
4.2	Computers and real numbers	63
4.2.1	The challenge of real numbers	63
4.2.2	The normal form of real numbers	64
4.2.3	32-bit floating-point numbers	66
4.2.4	Special bit combinations	67
4.2.5	64-bit floating-point numbers	67
4.2.6	Floating point numbers in Java	68

4.2.7	Floating point numbers in Python	68
4.3	Representation of letters and other characters	69
4.3.1	The ASCII table	70
4.3.2	ISO latin character sets	73
4.3.3	Unicode	73
4.3.4	UTF-8	75
4.3.5	UTF-16	77
4.3.6	UTF-32	78
4.3.7	Text in Java	78
4.3.8	Text in Python	79
4.4	Representation of general information	81
4.4.1	Text	81
4.4.2	Numbers	82
4.4.3	General information	83
4.4.4	Computer programs	83
4.5	A fundamental principle of computing	83
5	Computer arithmetic and round-off errors	87
5.1	Integer arithmetic and errors	88
5.2	Floating-point arithmetic and round-off error	88
5.2.1	Truncation and rounding	89
5.2.2	A simplified model for computer arithmetic	90
5.2.3	An algorithm for floating-point addition	91
5.2.4	Observations on round-off errors in addition/subtraction . .	93
5.2.5	Multiplication and division of floating-point numbers . . .	95
5.2.6	The IEEE standard for floating-point arithmetic	97
5.3	Measuring the error	99
5.3.1	Absolute error	99
5.3.2	Relative error	99
5.3.3	Properties of the relative error	100
5.3.4	Errors in floating-point representation	102
5.4	Rewriting formulas to avoid rounding errors	104
II	Sequences of Numbers	109
6	Numerical Simulation of Difference Equations	111
6.1	Why equations?	111
6.2	Difference equations defined	112

6.2.1	Initial conditions	115
6.2.2	Linear difference equations	116
6.2.3	Solving difference equations	116
6.3	Simulating difference equations	118
6.4	Review of the theory for linear equations	121
6.4.1	First-order homogenous equations	121
6.4.2	Second-order homogenous equations	122
6.4.3	Linear homogenous equations of general order	126
6.4.4	Inhomogenous equations	127
6.5	Simulation of difference equations and round-off errors	130
6.5.1	Explanation of example 6.27	133
6.5.2	Round-off errors for linear equations of general order	136
6.6	Summary	139
7	Lossless Compression	141
7.1	Introduction	142
7.1.1	Run-length coding	143
7.2	Huffman coding	144
7.2.1	Binary trees	146
7.2.2	Huffman trees	146
7.2.3	The Huffman algorithm	148
7.2.4	Properties of Huffman trees	152
7.3	Probabilities and information entropy	154
7.3.1	Probabilities rather than frequencies	154
7.3.2	Information entropy	155
7.4	Arithmetic coding	157
7.4.1	Arithmetic coding basics	157
7.4.2	An algorithm for arithmetic coding	159
7.4.3	Properties of arithmetic coding	163
7.4.4	A decoding algorithm	165
7.4.5	Arithmetic coding in practice	167
7.5	Lempel-Ziv-Welch algorithm	169
7.6	Lossless compression programs	169
7.6.1	Compress	170
7.6.2	gzip	170
8	Digital Sound	171
8.1	Sound	171
8.1.1	Loudness: Sound pressure and decibels	172
8.1.2	The pitch of a sound	175

8.1.3	Any function is a sum of sin and cos	177
8.2	Digital sound	180
8.2.1	Sampling	180
8.2.2	Limitations of digital audio: The sampling theorem	181
8.2.3	Reconstructing the original signal	182
8.3	Simple operations on digital sound	184
8.4	More advanced sound processing	189
8.4.1	The Discrete Cosine Transform	189
8.5	Lossy compression of digital sound	190
8.6	Psycho-acoustic models	193
8.7	Digital audio formats	194
8.7.1	Audio sampling — PCM	194
8.7.2	Lossless formats	195
8.7.3	Lossy formats	196
9	Polynomial Interpolation	199
9.1	The Taylor polynomial with remainder	200
9.1.1	The Taylor polynomial	200
9.1.2	The remainder	202
9.2	Interpolation and the Newton form	207
9.2.1	The interpolation problem	208
9.2.2	The Newton form of the interpolating polynomial	210
9.3	Divided differences	214
9.4	Computing with the Newton form	219
9.5	Interpolation error	221
9.6	Summary	223
10	Zeros of Functions	225
10.1	The need for numerical root finding	226
10.1.1	Analysing difference equations	226
10.1.2	Labelling plots	227
10.2	The Bisection method	228
10.2.1	The intermediate value theorem	228
10.2.2	Derivation of the Bisection method	229
10.2.3	Error analysis	232
10.2.4	Revised algorithm	234
10.3	The Secant method	236
10.3.1	Basic idea	237
10.3.2	Testing for convergence	239
10.3.3	Revised algorithm	240

10.3.4 Convergence and convergence order of the Secant method	241
10.4 Newton's method	243
10.4.1 Basic idea	243
10.4.2 Algorithm	244
10.4.3 Convergence and convergence order	245
10.5 Summary	249
11 Numerical Differentiation	251
11.1 Newton's difference quotient	252
11.1.1 The basic idea	253
11.1.2 The truncation error	254
11.1.3 The round-off error	256
11.1.4 Optimal choice of h	262
11.2 Summary of the general strategy	264
11.3 A symmetric version of Newton's quotient	265
11.3.1 Derivation of the method	265
11.3.2 The error	267
11.3.3 Optimal choice of h	269
11.4 A four-point differentiation method	271
11.4.1 Derivation of the method	271
11.4.2 The error	272
11.5 Numerical approximation of the second derivative	274
11.5.1 Derivation of the method	274
11.5.2 The error	275
11.5.3 Optimal choice of h	277
12 Numerical Integration	279
12.1 General background on integration	280
12.2 The midpoint rule for numerical integration	283
12.2.1 A detailed algorithm	284
12.2.2 The error	286
12.2.3 Estimating the step length	290
12.3 The trapezoidal rule	291
12.3.1 The error	293
12.4 Simpson's rule	296
12.4.1 Derivation of Simpson's rule	296
12.4.2 Composite Simpson's rule	298
12.4.3 The error	300
12.5 Summary	302

13 Numerical Solution of Differential Equations	303
13.1 What are differential equations?	303
13.1.1 An example from physics	303
13.1.2 General use of differential equations	305
13.1.3 Different types of differential equations	306
13.2 First order differential equations	306
13.2.1 Initial conditions	307
13.2.2 A geometric interpretation of first order differential equations	308
13.2.3 Conditions that guarantee existence of one solution	310
13.2.4 What is a numerical solution of a differential equation?	311
13.3 Euler's method	313
13.3.1 Basic idea and algorithm	313
13.3.2 Geometric interpretation	315
13.4 Error analysis for Euler's method	317
13.4.1 Round-off error	318
13.4.2 Local and global error	318
13.4.3 Untangling the local errors	320
13.5 Differentiating the differential equation	323
13.6 Taylor methods	326
13.6.1 The quadratic Taylor method	326
13.6.2 Taylor methods of higher degree	329
13.6.3 Error in Taylor methods	330
13.7 Midpoint Euler and other Runge-Kutta methods	332
13.7.1 Euler's midpoint method	332
13.7.2 The error	334
13.7.3 Runge-Kutta methods	335
13.8 Systems of differential equations	338
13.8.1 Vector notation and existence of solution	338
13.8.2 Numerical methods for systems of first order equations	341
13.8.3 Higher order equations as systems of first order equations	342
13.9 Final comments	345
III Functions of two variables	349
14 Functions of two variables	351
14.1 Basics	351
14.1.1 Basic definitions	351
14.1.2 Differentiation	353
14.1.3 Vector functions of several variables	355

14.2 Numerical differentiation	356
15 Digital images and image formats	361
15.1 What is an image?	361
15.1.1 Light	361
15.1.2 Digital output media	362
15.1.3 Digital input media	363
15.1.4 Definition of digital image	363
15.1.5 Images as surfaces	365
15.2 Operations on images	367
15.2.1 Normalising the intensities	367
15.2.2 Extracting the different colours	368
15.2.3 Converting from colour to grey-level	369
15.2.4 Computing the negative image	370
15.2.5 Increasing the contrast	370
15.2.6 Smoothing an image	372
15.2.7 Detecting edges	374
15.2.8 Comparing the first derivatives	377
15.2.9 Second-order derivatives	378
15.3 Image formats	379
15.3.1 Raster graphics and vector graphics	379
15.3.2 Vector graphics formats	381
15.3.3 Raster graphics formats	382
Answers to some exercises	385