Numerical Analysis

Rui Du, Zhizhong Sun

Contents

List of Figures **xi**

List of Tables **xiii**

1 Introduction **1**
1.1 Error . . . 2
1.1.1 Sources of Approximation . . . 2
1.1.2 Absolute Error and Relative Error . . . 3
1.1.3 Significant digits (or figures) . . . 4
1.1.4 Error Estimation of Function . . . 6
1.2 Computer Arithmetic . . . 9
1.3 Numerical Stability . . . 13
1.4 Ill-conditioned Problem . . . 16
1.5 Hornor's method . . . 18
1.6 Excercise . . . 19

2 Solutions of Equations in One Variable **21**
2.1 Introduction . . . 21
2.2 The Bisection Method . . . 22
2.3 Fixed-Point Iteration . . . 24
2.3.1 Fixed-Point Iteration . . . 24
2.3.2 Convergence . . . 26
2.3.3 Order of Convergence . . . 31
2.3.4 Aitken Method . . . 34
2.4 Newton's Method . . . 37
2.4.1 Local Convergence of Newton's method. . . . 38
2.4.2 Multiple Roots . . . 39
2.4.3 The Secant Method . . . 41
2.5 Zeros of Polynomials . . . 41
2.6 Excercise . . . 42

3 Numerical methods for linear system **43**
3.1 Direct Method . . . 45
3.1.1 Gaussian Elimination Method . . . 45
3.1.2 Gaussian Elimination Method with Partial Pivoting . 51
3.1.3 Thomas Algorithm for Tridiagonal System . . . 53
3.2 Norms and Error Analysis . . . 54

3.2.1 Norms of Vectors 54
3.2.2 Norms of Matrices 57
3.2.3 Error analysis 62
3.3 Iterative Methods 68
3.3.1 Jacobi iterative method 69
3.3.2 Gauss-Seidel Iterative method 71
3.3.3 Successive Over-relaxation method 72
3.3.4 Convergency of Iterative Method 73
3.3.5 Power Method 78
3.4 Exercise 84

4 Interpolation 87
4.1 Lagrange Interpolating Polynomials 88
4.1.1 Interpolation Error 92
4.2 Newton's Divided-Difference Formula 94
4.2.1 Divided difference 95
4.2.2 Forward Difference and Newton forward-difference formula 99
4.3 Hermite Interpolation 101
4.4 Piecewise-Polynomial Approximation 108
4.4.1 Error analysis of high-degree interpolating polynomials 108
4.4.2 Piecewise-Linear Interpolation 111
4.4.3 Hermite Piecewise Interpolation 112
4.5 Cubic Spline Interpolation 113
4.5.1 Cubic Splines 114
4.5.2 Construction of a Cubic Spline 115
4.5.3 Convergence of cubic spline 118
4.6 Exercise 119

5 Approximation Theory 121
5.1 Best Uniform Approximation 121
5.1.1 Normed Linear Space 121
5.1.2 Polynomial of Best Uniform Approximation 123
5.2 Least Square Approximation 127
5.2.1 Inner Product Space 127
5.3 Least Squares Approximation 129
5.3.1 Least Squares Approximation to continuous function 131
5.3.2 Least Squares Solution of Overdetermined Linear Equations 133
5.3.3 Discrete Least Square Approximation 134
5.4 Exercise 136

6 Numerical Integration and Differentiation **137**
6.1 Elements of Numerical Integration 137
6.1.1 Newton-Cotes Formula 138
6.1.2 Measuring Precision 141
6.1.3 Error of Trapezoidal Rule, Simpson's Rule and Boole's Rule 144
6.1.4 Stability 146
6.2 Composite Numerical Integration 146
6.2.1 Composite Trapezoidal Rule 147
6.2.2 Composite Simpson's Rule 149
6.2.3 Composite Boole's rule 151
6.3 Romberg Integration 153
6.4 Gaussian Quadrature 156
6.4.1 Orthogonal Polynomials 159
6.4.2 Truncation Error of Gaussian Quadrature 164
6.4.3 Stability and Convergence 165
6.5 Numerical Differentiation 166
6.6 Exercise 170

7 Initial-Value Problems for Ordinary Differential Equations **173**
7.1 Euler's Method 175
7.1.1 Euler's Explicit Method 175
7.1.2 Euler's Implicit Method 177
7.1.3 Trapezoidal method 178
7.1.4 Modified Euler's Method 179
7.1.5 Global Truncation Error 181
7.2 Runge-Kutta Methods 181
7.2.1 Second-order Runge-Kutta Methods 183
7.2.2 High-order Runge-Kutta method 185
7.2.3 Implicit Runge-Kutta Method 186
7.3 Stability and Convergence of One-step Method 187
7.4 Multistep Methods 188
7.4.1 Adams Methods 189
7.4.2 Taylor Methods 194
7.5 Exercise 199

8 Numerical Methods for Partial Differential Equations **201**
8.1 Parabolic Partial Differential Equations 201
8.1.1 Selecting a Grid 203
8.1.2 Forward Difference Method 204
8.1.3 Backward Difference Method 205
8.1.4 Richardson's Method 207
8.1.5 Crank-Nicolson Method 208
8.1.6 Stability and Convergence 212
8.2 Hyperbolic Partial Differential Equations 217

8.2.1 Explicit Method 217
8.2.2 Implicit Method 219
8.3 Elliptic Partial Differential Equations 223
8.4 Exercise 228

List of Figures

2.1 The plots of $y = x^2$ and $y = 1 + \sin x$ 22
2.2 Method 1 26
2.3 Method 2 27
2.4 The geometric of the Newton's method 38

4.1 $L_2(x)$ 91
4.2 $L_3(x)$ 92
4.3 Simplex on 1-dimensional space 104
4.4 Simplex on 3-dimensional space 104
4.5 $f(x)$ and $L_5(x)$ 109
4.6 $f(x)$ and $L_{10}(x)$ 109
4.7 $f(x)$ and $L_{16}(x)$ 109

5.1 Linear polynomial of best uniform approximation 127
5.2 127
5.3 136

6.1 Numerical Differentiation 167

7.1 D 174
7.2 The geometric interpretation of Euler's method 176

8.1 Grid 203
8.2 Grid points 205
8.3 Grid points 206
8.4 Grid points 208
8.5 Grid points 210
8.6 Grid points 217
8.7 Grid nodes 219
8.8 Grid points 222
8.9 Grid 224

List of Tables

1.1 Values of $\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon}$ 17

2.1 Computing results by the Bisection Method 24
2.2 Computing results by Newton's method 39
2.3 Computing results by Newton's method 40

3.1 Computing results 81

4.1 96
4.2 The divided differences 96
4.3 98
4.4 Divided Difference 98
4.5 The forward difference 100
4.6 Table of the divided difference 107
4.7 107
4.8 The values of functions $f(x)$ and $L_{10}(x)$ 110

6.1 The degree of precisionNewton-Cotes formula 144
6.2 Example of composite Trapezoidal rule 150
6.3 Example of composite Simpson's rule 152
6.4 Example of composite Boole's Rule 153
6.5 Example of Romberg integration 155
6.6 t_k and A_k of Gaussian quadrature on $[-1, 1]$ 162

7.1 Computed results by Euler's method 176
7.2 The computed results by modified Euler's method 180
7.3 The computing results by RK_4 186
7.4 The computing results 194

8.1 By Forward Difference method $(h = 1/10, \tau = 1/200)$ 211
8.2 By Forward Difference method $(h = 1/10, \tau = 1/100)$ 212
8.3 By Backward Difference method with $(h = 1/10, \tau = 1/200)$. 213
8.4 By Richardson's method with $(h = 1/10, \tau = 1/100)$ 213
8.5 By Crank-Nicolson method with $(h = 1/10, \tau = 1/10)$ 214
8.6 Numerical results with $(h = 1/100, \tau = 1/100)$ 220
8.7 Numerical results with $(h = 1/100, \tau = 1/80)$ 220
8.8 Numerical Results with $(h = 1/100, \tau = 1/100)$ 223

8.9 Numerical Results with($M = 10, N = 10$) 227
8.10 Num ($M = 100, N = 100$) 227

1

Introduction

CONTENTS

1.1 Error 2
  1.1.1 Sources of Approximation 2
  1.1.2 Absolute Error and Relative Error 3
  1.1.3 Significant digits (or figures) 4
  1.1.4 Error Estimation of Function 5
1.2 Computer Arithmetic 8
1.3 Numerical Stability 13
1.4 Ill-conditioned Problem 15
1.5 Hornor's method 18
1.6 Excercise 19

It is best to start this book with a question: What do we mean by "Numerical Methods and Analysis"? What kind of mathematics is this book about?

Generally and broadly speaking, this book covers the mathematics and methodologies that underlie the techniques of scientific computation. More prosaically, consider the button on your calculator that computes the sine of the number in the display. Exactly how does the calculator know that correct value? When we speak of using the computer to solve a complicated mathematics or engineering problem, exactly what is involved in making that happen? Are computers "born" with the knowledge of how to solve complicated mathematical and engineering problems? No, of course they are not. Mostly they are programmed to do it, and the programs implement algorithms that are based on the kinds of things we talk about in this book.

Textbooks and courses in this area generally follow one of two main themes: Those titled "Numerical Methods" tend to emphasize the implementation of the algorithms, perhaps at the expense of the underlying mathematical theory that explains why the methods work; those titled "Numerical Analysis" tend to emphasize this underlying mathematical theory, perhaps at the expense of some of the implementation issues. The best approach, of course, is to properly mix the study of the algorithms and their implementation ("methods") with the study of the mathematical theory ("analysis") that supports them. This is our goal in this book.

Whenever someone speaks of using a computer to design an airplane, predict the weather, or otherwise solve a complex science or engineering problem, that person is talking about using numerical methods and analysis. The problems and areas of endeavor that use these kinds of techniques are continually expanding. For example, computational mathematics another name for the material that we consider here- is now commonly used in the study of financial markets and investment structures, an area of study that does not ordinarily come to mind when we think of ”scientific” computation. Similarly, the increasingly frequent use of computer-generated animation in film production is based on a heavy dose of spline approximations. And modern weather prediction is based on using numerical methods and analysis to solve the very complicated equations governing fluid flow and heat transfer between and within the atmosphere, oceans, and ground.

There are a number of different ways to break the subject down into component parts. We will discuss the derivation and implementation of the algorithms, and we will also analyze the algorithms, mathematically, in order to learn how best to use them and how best to implement them. In our study of each technique, we will usually be concerned with two issues that often are in competition with each other:

1. Accuracy: Very few of our computations will yield the exact answer to a problem, so we will have to understand how much error is made, and how to control (or even diminish) that error.
2. Efficiency: Does the algorithm take an inordinate amount of computer time? This might seem to be an odd question to concern ourselves with - after all, computers are fast, right? - -but there are slow ways to do things and fast ways to do things. All else being equal (it rarely is), we prefer the fast ways.
3. Stability: Does the method produce similar results for similar data? If we change the data by a small amount, do we get vastly different results? If so, we say that the method is unstable, and unstable methods tend to produce unreliable results. It is entirely possible to have an accurate method that is efficiently implemented, yet is horribly unstable.

1.1 Error

1.1.1 Sources of Approximation

There are many sources of approximation or inexactness in computational science, such as

(1) Modeling Error: Some physical features of the problem or system under study may be simplified or omitted (e.g., friction, viscosity, air resistance).

(2) Measuring Error: In the mathematical model, some physical coefficients are obtained by measuring, such as voltage, current and temperature etc. Since laboratory instruments have finite precision, the deviation of a measurement from its true value always exists.

(3) Truncation Error: Some features of a mathematical model may be omitted or simplified (e.g., replacing derivatives by finite differences or using only a finite number of terms in an infinite series).

(4) Roundoff Error: Whether in hand computation, a calculator, or a digital computer, the representation of real numbers and arithmetic operations upon them is ultimately limited to some finite amount of precision and thus is generally inexact. For example, $\sqrt{2} = 1.41421356\cdots$, $\frac{1}{3} = 0.3333\cdots$ which would be approximated by finite decimal in the computation.

The accuracy of the final results of a computation may reflect a combination of any or all of these approximations, and the resulting perturbations may be amplified by the nature of the problem being solved or the algorithm being used, or both. In this book, the truncation error and roundoff error are considered.

1.1.2 Absolute Error and Relative Error

Definition 1. *If x is an approximation to x^*, the* **absolute error** *is*

$$e(x) = x^* - x,$$

and the **relative error** *is*

$$e_r(x) = \frac{x^* - x}{x^*} = \frac{e(x)}{x^*},$$

where we assume $x^* \neq 0$.

Why do we need two different measures of error? Consider the problem of approximating the number

$$x^* = e^{-16} = 0.1125351747... \times 10^{-6}.$$

Because x^* is so small, the absolute error in $x = 0$ as an approximation to x^* is also small. In fact, $|x^* - x| < 1.2 \times 10^{-7}$, which is decent accuracy in many settings. However, this "approximation" is clearly not a good one. On the other hand, consider the problem of approximating

$$z^* = e^{16} = 0.8886110521... \times 10^{7}.$$

Because z^* is so large, the absolute error in almost any approximation will be large, even though almost all of the digits are matched. For example, if we take $z = 0.8886110517\times\ 10^7$, then we have $|z^* - z| = 4 \times 10^{-3}$.

The point is that relative error gives a measure of the number of correct digits in the approximation. Thus,

$$\left|\frac{x^* - x}{x^*}\right| = 1$$

which tells us that not many digits are matched in that example, whereas

$$\left|\frac{z^* - z}{z^*}\right| = \frac{4 \times 10^{-3}}{0.8886110521 \times 10^7} = 0.4501 \times 10^{-9}$$

which shows that about nine digits are correct. Generally speaking, using a relative error protects us from misjudging the accuracy of an approximation because of scale extremes (very large or very small numbers).

As a practical matter, the absolute error cannot be obtained because x^* is unknown. Actually, if $\exists\ \varepsilon > 0$ with

$$|e(x)| = |x^* - x| \le \varepsilon,$$

ε is called the bound of absolute error. Sometimes we denote $x^* = x \pm \varepsilon$.

Similarly because x^* is unknown,

$$\bar{e}_r(x) = \frac{x^* - x}{x}$$

can be seen as the approximation to the relative error generally.

Noting that

$$\bar{e}_r - e_r = \frac{\bar{e}_r^2}{1 + \bar{e}_r} = \frac{e_r^2}{1 - e_r},$$

we also use $\bar{e}_r$ instead of e_r since the difference of them is about $O(\bar{e}_r^2)$ or $O({e_r}^2)$. If $\exists\ \varepsilon_r > 0$, satisfying

$$|e_r(x)| \le \varepsilon_r \qquad \texttt{or} \qquad |\bar{e}_r(x)| \le \varepsilon_r,$$

ε_r is called the (upper) bound of relative error.

1.1.3 Significant digits (or figures)

Definition 2. *The number $x = 0.d_1 d_2 ... d_n ... \times 10^m$ is said to approximate x^* to n significant digits (or figures) if the n is the largest nonnegative integer for which*

$$|x^* - x| \le \frac{1}{2} \times 10^{m-n}.$$

For example, the approximation $x_1 = 3.14$ is approximated to π, then

$$|\pi - x_1| = 0.00159\cdots < 0.005 = \frac{1}{2} \times 10^{-2},$$

so we say x_1 is of 3 significant figures;

If $x_2 = 3.1416$ is another approximation to π, then

$$|\pi - x_2| < 0.00005 = \frac{1}{2} \times 10^{-4},$$

x_2 has 5 significant figures; If $x_3 = 3.1415$, then

$$|\pi - x_3| = 0.00009\cdots < 0.0005 = \frac{1}{2} \times 10^{-3},$$

so x_3 has 4 significant figures.

The number x is represented of n significant figures as follows

$$x = \pm 0, \alpha_1\alpha_2\cdots\alpha_n \times 10^m$$

i.e.

$$x = \pm\left(\alpha_1 \times 10^{-1} + \alpha_2 \times 10^{-2} + \cdots + \alpha_n \times 10^{-n}\right) \times 10^m$$

where m is integer, $\alpha_1, \alpha_2, \cdots, \alpha_n$ are integers from 0 to 9, and $\alpha_1 \neq 0$. Since

$$|x^* - x| \leqslant \frac{1}{2} \times 10^{m-n},$$

the upper bound of the absolute error of x is $\varepsilon = \frac{1}{2} \times 10^{m-n}$. Then the bigger n, the smaller error with the same m. From

$$\frac{|x - x^*|}{|x|} \leqslant \frac{\frac{1}{2} \times 10^{m-n}}{\alpha_1 \times 10^{-1} \times 10^m} = \frac{1}{2\alpha_1} \times 10^{-n+1},$$

the bound of the relative error of x is

$$\varepsilon_r = \frac{1}{2\alpha_1} \times 10^{-n+1}.$$

From Eq. (1.1), we can see that the more significant figures of an approximation, the smaller bound of the relative error. Since $\frac{1}{\alpha_1} \leqslant 1$, we also use

$$\varepsilon_r = \frac{1}{2} \times 10^{-n+1}$$

as the bound of the relative error of the approximation x.

1.1.4 Error Estimation of Function

In numerical operation, the error of the given data will inevitably lead to the error of the function value. The error estimation of functions can be obtained by Taylor's theorem.

For the function in one variable $f(x)$, suppose x^* is exact and$y^* = f(x^*)$. The approximation x is to x^*, then $y = f(x)$. From Taylor's theorem,

$$e(y) = y^* - y = f(x^*) - f(x) \approx f'(x)(x^* - x) = f'(x)e(x),$$

that is

$$e(y) \approx f'(x)e(x) \tag{1.1}$$

Then

$$e_r(y) = \frac{e(y)}{y} \approx \frac{xf'(x)}{f(x)} e_r(x). \tag{1.2}$$

For the function in two variables, suppose x_1^*, x_2^* are real numbers, $y^* = f(x_1^*, x_2^*)$. x_1, x_2 are approximations to x_1^*, x_2^* correspondingly, $y = f(x_1, x_2)$. From Taylor's theorem,

$$\begin{aligned} e(y) \quad &= y^* - y \\ &= f(x_1^*, x_2^*) - f(x_1, x_2) \\ &\approx \frac{\partial f(x_1, x_2)}{\partial x_1}(x_1^* - x_1) + \frac{\partial f(x_1, x_2)}{\partial x_2}(x_2^* - x_2), \end{aligned}$$

i.e.

$$e(y) \approx \frac{\partial f(x_1, x_2)}{\partial x_1} e(x_1) + \frac{\partial f(x_1, x_2)}{\partial x_2} e(x_2), \tag{1.3}$$

where $e(x_1) = x_1^* - x_1$ and $e(x_2) = x_2^* - x_2$.

Then we can obtain

$$\begin{aligned} e_r(y) = \frac{e(y)}{y} \quad &\approx \frac{\partial f(x_1, x_2)}{\partial x_1} \frac{x_1}{f(x_1, x_2)} e_r(x_1) \\ &+ \frac{\partial f(x_1, x_2)}{\partial x_2} \frac{x_2}{f(x_1, x_2)} e_r(x_2). \end{aligned} \tag{1.4}$$

where $e(x_1) = x_1^* - x_1$ and $e(x_2) = x_2^* - x_2$.

It is easy to get the following formulas from (1.3) and (1.4)

$$e(x_1 + x_2) \approx e(x_1) + e(x_2), \tag{1.5}$$
$$e(x_1 - x_2) \approx e(x_1) - e(x_2), \tag{1.6}$$
$$e(x_1 x_2) \approx x_2 e(x_1) + x_1 e(x_2), \tag{1.7}$$
$$e\left(\frac{x_1}{x_2}\right) \approx \frac{1}{x_2} e(x_1) - \frac{x_1}{x_2^2} e(x_2), \tag{1.8}$$
$$e_r(x_1 + x_2) \approx \frac{x_1}{x_1 + x_2} e_r(x_1) + \frac{x_2}{x_1 + x_2} e_r(x_2), \tag{1.9}$$
$$e_r(x_1 - x_2) \approx \frac{x_1}{x_1 - x_2} e_r(x_1) - \frac{x_2}{x_1 - x_2} e_r(x_2), \tag{1.10}$$
$$e_r(x_1 x_2) \approx e_r(x_1) + e_r(x_2), \tag{1.11}$$
$$e_r\left(\frac{x_1}{x_2}\right) \approx e_r(x_1) - e_r(x_2). \tag{1.12}$$

Exercise 1.1. *Suppose $x_1 = 1.021$, $x_2 = 2.134$ are approximations with 4 significant figures. Determine the upper bounds of the absolute error and relative error of $x_1 - x_2$, $x_1^2 - x_2^2$ and $x_1^2 x_2$.*

Solution Since x_1 and x_2 are of 4 significant figures,

$$|e(x_1)| \le \frac{1}{2} \times 10^{-3}, \quad |e(x_2)| \le \frac{1}{2} \times 10^{-3}.$$

Using (1.3) and (1.4), there are

$$|e(x_1 - x_2)| \le |e(x_1)| + |e(x_2)| \le 10^{-3},$$
$$|e_r(x_1 - x_2)| = \left|\frac{e(x_1 - x_2)}{x_1 - x_2}\right| \le \frac{10^{-3}}{1.113} = 8.9847 \times 10^{-4}.$$

Similarly, we obtain

$$|e(x_1^2 - x_2^2)| \approx |2(x_1 e(x_1) - x_2 e(x_2))| \le 2(x_1 |e(x_1)| + x_2 |e(x_2)|)$$
$$\le 3.155 \times 10^{-3},$$
$$|e_r(x_1^2 - x_2^2)| = \left|\frac{e(x_1^2 - x_2^2)}{x_1^2 - x_2^2}\right| \le 8.985 \times 10^{-4},$$

and

$$|e(x_1^2 x_2)| \approx |2x_1 x_2 e(x_1) + x_1^2 e(x_2)| \le 2.7 \times 10^{-3},$$
$$|e_r(x_1^2 x_2)| = \left|\frac{e(x_1^2 x_2)}{x_1^2 x_2}\right| \le 1.2134 \times 10^{-3}.$$

□

Exercise 1.2. *Assume* $x_1^* = \sqrt{2001}$, $x_2^* = \sqrt{1999}$, $x_1 = 44.7325$, $x_2 = 44.7102$ *are approximations to* x_1^*, x_2^* *with 6 significant figures. There are two algorithms to calculate* $\sqrt{2001} - \sqrt{1999}$*:*

$$(1) \quad x_1^* - x_2^* \approx x_1 - x_2 = 44.7325 - 44.7102 = 0.0223.$$

$$\begin{aligned}(2) \quad x_1^* - x_2^* &= \frac{2}{x_1^* + x_2^*} \approx \frac{2}{x_1 + x_2} = \frac{2}{44.7325 + 44.7102} \\ &= 0.0223606845 \cdots .\end{aligned}$$

Try to analyze how many significant figures of the computed results by the above algorithms.

Solution We know

$$|e(x_1)| \le \frac{1}{2} \times 10^{-4}, \quad |e(x_2)| \le \frac{1}{2} \times 10^{-4}.$$

Then

$$\begin{aligned}|e(x_1 - x_2)| &\approx |e(x_1) - e(x_2)| \\ &\le |e(x_1)| + |e(x_2)| \\ &\le \frac{1}{2} \times 10^{-4} + \frac{1}{2} \times 10^{-4} = 10^{-4} < \frac{1}{2} \times 10^{-3}.\end{aligned}$$

So the result by the algorithm (1) has at least 2 significant figures. On the other hand,

$$\begin{aligned}|e(\frac{2}{x_1 + x_2})| &\approx |-\frac{2}{(x_1 + x_2)^2} e(x_1 + x_2)| \\ &\approx |-\frac{2}{(x_1 + x_2)^2} [e(x_1) + e(x_2)]| \\ &\le \frac{2}{(x_1 + x_2)^2} [|e(x_1)| + |e(x_2)|] \\ &\le \frac{2}{(44.7325 + 44.7102)^2} \left(\frac{1}{2} \times 10^{-4} + \frac{1}{2} \times 10^{-4} \right) \\ &= 0.25 \times 10^{-7} < \frac{1}{2} \times 10^{-7}\end{aligned}$$

The result by algorithm (2) has at least 6 significant figures. It is easy to get the result by algorithm (1) is of only 2 significant figures.

□

From this example, it is noted that the significant figures will be reduced when subtracting two closed numbers.

1.2 Computer Arithmetic

We need to spend some time reviewing how the computer actually does arithmetic. The reason for this is simple: computer arithmetic is generally inexact, and while the errors that are made are very small, they can accumulate under some circumstances and actually dominate the calculation. Thus, we need to understand computer arithmetic well enough to anticipate and deal with this phenomenon. Most computer languages use what is called floating-point arithmetic. Although the details differ from machine to machine, the basic idea is the same. Every number is represented using a (fixed, finite) number of binary digits, usually called bits. A typical implementation would represent the number in the form

$$x = \pm\left(0.\alpha_1\alpha_2\cdots\alpha_n\right)\beta^p. \tag{1.13}$$

i.e.

$$x = \pm\left(\frac{\alpha_1}{\beta} + \frac{\alpha_2}{\beta^2} + \cdots + \frac{\alpha_i}{\beta^i} + \cdots + \frac{\alpha_n}{\beta^n}\right)\beta^p$$

where α_i is integer and satisfies

$$0 \le \alpha_i \le \beta - 1 \quad (i = 1, 2, \cdots, n)$$

where $\alpha = \pm 0.\alpha_1\alpha_2\cdots\alpha_n$ is called **mantissa**, β is the **base** of the internal number system and p is the **exponent** $L \leqslant p \leqslant U$ (L and U are constant decided by the computer).

If $\alpha_1 \neq 0$, the floating-point form of x is said n-digit normalized machine number. The set of all normalized float number and zero is called **machine number system**.

Denote

$$F(\beta, n, L, U) = \{0\} \cup \{x \mid x = \pm\left(0.\alpha_1\alpha_2\cdots\alpha_n\right)\beta^p\}$$

where $1 \le \alpha_1 \le \beta - 1; 0 \le \alpha_i \le \beta - 1, i = 2, 3, \cdots, n; L \le p \le U$.

$F(\beta, n, L, U)$ is a discrete set of finite rational numbers which has $1+2(\beta-1)\beta^{n-1}(U-L+1)$ numbers. The number with the largest absolute value is $\pm\left(\frac{\beta-1}{\beta} + \frac{\beta-1}{\beta^2} + \cdots + \frac{\beta-1}{\beta^n}\right)\beta^U = \pm\left(1-\beta^{-n}\right)\beta^U$, and the nonzero number with minimum absolute value is $\pm\left(\frac{1}{\beta} + \frac{0}{\beta^2} + \cdots + \frac{0}{\beta^n}\right)\beta^L = \pm\beta^{-1+L}$.

Exercise 1.3. *Suppose there is a computer with $\beta = 2, n = 3, L = -1, U = 2$.*
(1) Confirm how many numbers in the set F;
(2) Show all the numbers of F in binary and decimal floating point form;
(3) Numbers in F are represented on the axis.

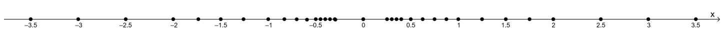
Solution (1) The set $F(2,3,-1,2)$ has

$$1+2(2-1)2^{3-1}[2-(-1)+1]=33$$

numbers. (2)The 33 numbers in binary and decimal floating point form are:

$$
\begin{array}{ll}
(0.000)_2=(0)_{10} & 0 \\
\left.\begin{array}{c}
\pm\left(0.100\times 2^{-1}\right)_2=\pm(0.25)_{10} \\
\pm\left(0.101\times 2^{-1}\right)_2=\pm(0.3125)_{10} \\
\pm\left(0.110\times 2^{-1}\right)_2=\pm(0.375)_{10} \\
\pm\left(0.111\times 2^{-1}\right)_2=\pm(0.4375)_{10}
\end{array}\right\} & p=-1 \\
\left.\begin{array}{c}
\pm\left(0.100\times 2^{0}\right)_2=\pm(0.5)_{10} \\
\pm\left(0.101\times 2^{0}\right)_2=\pm(0.625)_{10} \\
\pm\left(0.110\times 2^{0}\right)_2=\pm(0.75)_{10} \\
\pm\left(0.111\times 2^{0}\right)_2=\pm(0.875)_{10}
\end{array}\right\} & p=0 \\
\left.\begin{array}{c}
\pm\left(0.100\times 2^{1}\right)_2=\pm(1)_{10} \\
\pm\left(0.101\times 2^{1}\right)_2=\pm(1.25)_{10} \\
\pm\left(0.110\times 2^{1}\right)_2=\pm(1.5)_{10} \\
\pm\left(0.111\times 2^{1}\right)_2=\pm(1.75)_{10}
\end{array}\right\} & p=1 \\
\left.\begin{array}{c}
\pm\left(0.100\times 2^{2}\right)_2=\pm(2)_{10} \\
\pm\left(0.101\times 2^{2}\right)_2=\pm(2.5)_{10} \\
\pm\left(0.110\times 2^{2}\right)_2=\pm(3)_{10} \\
\pm\left(0.111\times 2^{2}\right)_2=\pm(3.5)_{10}
\end{array}\right\} & p=2
\end{array}
$$

(3) The numbers in F are shown in the following figure.



From the example, we can see that the 33 numbers in $F(2,3,-1,2)$ are rational points and are unevenly distributed on [-3.5,3.5].

□

Any real number x can be represented in machine number by terminating the mantissa of x at n digits i.e. $fl(x)$. There are two ways of performing this termination. One method, called **chopping**, is to simply chop off. The other method, called **rounding**, to round up or down.

Theorem 1. *Suppose $x\neq 0$ and $fl(x)$ is the representation of x in $F(\beta,n,L,U)$. The relative error of $fl(x)$ satisfy*

$$|e_r|=\left|\frac{x-fl(x)}{x}\right|\le\begin{cases}\frac{1}{2}\beta^{1-n}, & \textit{rounding}\\ \beta^{1-n}, & \textit{chopping}\end{cases}$$

Proof. Suppose the real number x is represented of the base β as $x = \alpha\beta^p$, where p is integer and $\beta^{-1} \leq |\alpha| < 1$. The mantissa α is

$$\alpha = \pm 0.\alpha_1\alpha_2 \cdots \alpha_n\alpha_{n+1} \cdots$$

where $1 \leq \alpha_1 \leq \beta - 1; 0 \leq \alpha_i \leq \beta - 1, i = 2, 3, \cdots$.

For rounding, set

$$\alpha' = \begin{cases} 0.\alpha_1\alpha_2 \cdots \alpha_n, & \text{if } 0 \leq \alpha_{n+1} \leq \frac{\beta}{2} - 1; \\ 0.\alpha_1\alpha_2 \cdots \alpha_n + \beta^{-n}, & \text{if } \alpha_{n+1} \geq \frac{\beta}{2} \end{cases}$$

and

$$fl(x) = \operatorname{sgn}(x)\alpha'\beta^p,$$

where

$$\operatorname{sgn}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -1, & \text{if } x < 0. \end{cases}$$

Then

$$|e_r| = \left|\frac{x - fl(x)}{x}\right| \leq \left|\frac{\frac{1}{2}\beta^{-n}\beta^p}{\alpha\beta^p}\right| \leq \frac{1}{2}\beta^{1-n}.$$

For chopping, set $\alpha' = 0.\alpha_1\alpha_2 \cdots \alpha_n$, and

$$fl(x) = \operatorname{sgn}(x)\alpha'\beta^p,$$

then

$$|e_r| = \left|\frac{x - fl(x)}{x}\right| \leq \left|\frac{\beta^{-n} \cdot \beta^p}{\alpha\beta^p}\right| \leq \frac{\beta^{-n}}{\beta^{-1}} = \beta^{1-n}$$

□

For decimal system, i.e. $\beta = 10$. The relative error is

$$|e_r| = \left|\frac{x - fl(x)}{x}\right| = \begin{cases} \frac{1}{2} \times 10^{1-n}, & \text{rounding} \\ 10^{1-n}, & \text{chopping} \end{cases}$$

The $\frac{1}{2}\beta^{1-n}$ or β^{1-n} are called computer precision which is denoted eps, i.e.

$$\text{eps} = \frac{1}{2}\beta^{1-n} \text{ or } \beta^{1-n}.$$

Denote

$$\varepsilon = \frac{fl(x) - x}{x}.$$

Then the relation between x and $fl(x)$ is as follows:

$$fl(x) = x(1 + \varepsilon), \quad |\varepsilon| \leq \text{eps}.$$

Assume x_1, x_2 are normalized floating numbers, and

$$fl\left(x_1+x_2\right)=\left(x_1+x_2\right)\left(1+\varepsilon_1\right), \tag{1.14}$$
$$fl\left(x_1-x_2\right)=\left(x_1-x_2\right)\left(1+\varepsilon_2\right), \tag{1.15}$$
$$fl\left(x_1\cdot x_2\right)=\left(x_1\cdot x_2\right)\left(1+\varepsilon_3\right), \tag{1.16}$$
$$fl\left(x_1/x_2\right)=\left(x_1/x_2\right)\left(1+\varepsilon_4\right), \tag{1.17}$$

where

$$|\varepsilon_i|<\ \text{eps}\quad (i=1,2,3,4).$$

Exercise 1.4. *Suppose x_1, x_2, x_3 are real number. Show the relative errors of $fl\left(fl\left(x_1+x_2\right)+x_3\right)$ and $fl\left(x_1+\ fl\left(x_2+x_3\right)\right)$.*

Solution From Eq. (1.14)

$$\begin{aligned} fl\left(fl\left(x_1+x_2\right)+x_3\right) &= fl\left(\left(x_1+x_2\right)\left(1+\varepsilon_1\right)+x_3\right)\\ &= \left[\left(x_1+x_2\right)\left(1+\varepsilon_1\right)+x_3\right]\left(1+\varepsilon_2\right)\\ &= \left[\left(x_1+x_2+x_3\right)+\left(x_1+x_2\right)\varepsilon_1\right]\left(1+\varepsilon_2\right)\\ &= \left(x_1+x_2+x_3\right)\left[1+\varepsilon_2+\frac{x_1+x_2}{x_1+x_2+x_3}\varepsilon_1\left(1+\varepsilon_2\right)\right]\\ &= \left(x_1+x_2+x_3\right)(1+\varepsilon), \end{aligned}$$

where

$$\varepsilon=\varepsilon_2+\frac{x_1+x_2}{x_1+x_2+x_3}\varepsilon_1\left(1+\varepsilon_2\right),\quad |\varepsilon_i|\le\ \text{eps}\quad (i=1,2). \tag{1.18}$$

Then absolute relative error of $fl\left(fl\left(x_1+x_2\right)+x_3\right)$ is $|\varepsilon|$.

Similarly,

$$fl\left(x_1+fl\left(x_2+x_3\right)\right)=\left(x_1+x_2+x_3\right)\left(1+\epsilon'\right),$$

where

$$\varepsilon'=\varepsilon_2'+\frac{x_2+x_3}{x_1+x_2+x_3}\varepsilon_1'\left(1+\varepsilon_2'\right),\quad |\varepsilon_i'|\le\ \text{eps}\quad (i=1,2), \tag{1.19}$$

i.e. the absolute relative error of $fl\left(x_1+f\left(x_2+x_3\right)\right)$ is $|\varepsilon'|$.

□

Because ε_i and ε_i' is random, it can be obtained from Eq.(1.18) and (1.19) : if $|x_1+x_2|<|x_2+x_3|$, then $|\varepsilon|<|\varepsilon'|$; if $|x_2+x_3|<|x_1+x_2|$, then $|\varepsilon'|<|\varepsilon|$. When some positive (or negative) numbers are added, the relative error would be smaller if the smaller absolute value of the numbers are added firstly. In practice, $(a+b)+c=a+(b+c)$ is not held in the computer arithmetic.

The four arithmetic operation are realized according to these rules:

(1) **Addition and subtraction** Firstly, the smaller exponent is to match the larger one. The two mantissas are added or subtracted and the result is rounded up or down to the normalized form.

(2) **Multiplication** The exponents are added and the mantissas are multiplied. The result is rounded up or down to the normalized form with the sign of the number (± 1) .

(3) **Division** The exponents are subtracted and the mantissas are divided. The result is rounded up or down to the normalized form with the sign of the number (± 1) .

Exercise 1.5. *For the rounding method,* $n = 3, L = -5, U = 5, x = 1.623, y = 0.184$ $z = 0.00362$*. Compute* $u = (x + y) + z$ *and* $v = x + (y + z)$*.*

Solution $fl(x) = 0.162 \times 10^1, fl(y) = 0.184 \times 10^0, fl(z) = 0.362 \times 10^{-2}$.

$$\begin{aligned}
fl(x) + fl(y) &= 0.162 \times 10^1 + 0.184 \times 10^0 = 0.162 \times 10^1 + 0.018 \times 10^1 \\
&= (0.162 + 0.018) \times 10^1 = 0.180 \times 10^1 \\
u &= (fl(x) + fl(y)) + fl(z) = 0.180 \times 10^1 + 0.362 \times 10^{-2} \\
&= 0.180 \times 10^1 + 0.000 \times 10^1 = (0.180 + 0.000) \times 10^1 \\
&= 0.180 \times 10^1 \\
fl(y) + fl(z) &= 0.184 \times 10^0 + 0.362 \times 10^{-2} = 0.184 \times 10^0 + 0.004 \times 10^0 \\
&= (0.184 + 0.004) \times 10^0 = 0.188 \times 10^0 \\
v &= fl(x) + (fl(y) + fl(z)) = 0.162 \times 10^1 + 0.188 \times 10^0 \\
&= 0.162 \times 10^1 + 0.019 \times 10^1 = (0.162 + 0.019) \times 10^1 \\
&= 0.181 \times 10^1.
\end{aligned}$$

□

From this example, the results by two algorithms are different which is not equal to the exact value 1.81062 because the rounding error of x and the error result in the addition and subtraction.

1.3 Numerical Stability

Definition 3. *A numerical algorithm is said to be* **numerical stable** *if the result it produces is relatively insensitive to perturbations due to approximations made during the computation; otherwise it is called* **unstable**.

Exercise 1.6. *Construct the recursive equation to compute*

$$I_n = \int_0^1 \frac{x^n}{x+5} \mathrm{d}x \quad (n = 0, 1, 2, \cdots, 10). \tag{1.20}$$

and analyze the error propagation.

Solution

$$\begin{aligned} I_n &= \int_0^1 \frac{x^n + 5x^{n-1} - 5x^{n-1}}{x+5}\mathrm{d}x = \int_0^1 x^{n-1}\mathrm{d}x - 5\int_0^1 \frac{x^{n-1}}{x+5}\mathrm{d}x \\ &= \frac{1}{n} - 5I_{n-1} \quad (n = 1, 2, \cdots, 10) \end{aligned}$$

and

$$I_0 = \int_0^1 \frac{1}{x+5}\mathrm{d}x = \ln 1.2.$$

We can obtain the recursive equation of I_n

$$\begin{cases} I_n = \frac{1}{n} - 5I_{n-1}, \quad (n = 1, 2, \cdots, 10), \\ I_0 = \ln 1.2. \end{cases} \tag{1.21}$$

In the computing, $\tilde{I}_0 = 0.182322$ is an approximation to I_0 with 6 significant figures.Assume $\tilde{I}_i$ is an approximation to I_i

$$\begin{array}{ll} \tilde{I}_1 = 1 - 5\tilde{I}_0 = 0.0883900, & \tilde{I}_2 = \frac{1}{2} - 5\tilde{I}_1 = 0.0580500, \\ \tilde{I}_3 = \frac{1}{3} - 5\tilde{I}_2 = 0.0430833, & \tilde{I}_4 = \frac{1}{4} - 5\tilde{I}_3 = 0.0345835, \\ \tilde{I}_5 = \frac{1}{5} - 5\tilde{I}_4 = 0.0270825, & \tilde{I}_6 = \frac{1}{6} - 5\tilde{I}_5 = 0.0312542, \\ \tilde{I}_7 = \frac{1}{7} - 5\tilde{I}_6 = -0.0134139, & \tilde{I}_8 = \frac{1}{8} - 5\tilde{I}_7 = 0.192070, \\ \tilde{I}_9 = \frac{1}{9} - 5\tilde{I}_8 = -0.849239, & \tilde{I}_{10} = \frac{1}{10} - 5\tilde{I}_9 = 4.34620. \end{array}$$

From Eq.(1.20), $I_n > 0$ for any n and $\{I_n\}_{n=0}^{\infty}$ is monotonically decreasing and tends to zero. In the computation, we can see $I_7 < 0$, and the computing results are not right. Now set $e_0 = I_0 - \tilde{I}_0$. In the recursive equation (1.21), the approximation of I_n is obtained by the approximation $\tilde{I}_{n-1}$:

$$\tilde{I}_n = \frac{1}{n} - 5\tilde{I}_{n-1} \tag{1.22}$$

Subtracting Eq.(1.21) and Eq.(1.22), it can be obtained

$$I_n - \tilde{I}_n = (-5)\left(I_{n-1} - \tilde{I}_{n-1}\right) \quad (n = 1, 2, \cdots, 10),$$

or

$$\left|I_n - \tilde{I}_n\right| = 5\left|I_{n-1} - \tilde{I}_{n-1}\right| \quad (n = 1, 2, \cdots, 10).$$

Denote

$$e_n = I_n - \tilde{I}_n.$$

We have

$$|e_n| = 5^n |e_0| \quad (n = 1, 2, \cdots, 10).$$

The error e_n is 5^n times as much as e_0. If n is much larger, the error would influent the I_n. The method is unstable.

On the other hand,

$$I_{n-1} = \frac{1}{5}\left(\frac{1}{n} - I_n\right) \quad (n = 10, 9, \cdots, 1). \tag{1.23}$$

If $\tilde{I}_{10}$ approximated to I_0 is known, $\tilde{I}_9, \tilde{I}_8, \cdots, \tilde{I}_0$ can be obtained.

$$\tilde{I}_{n-1} = \frac{1}{5}\left(\frac{1}{n} - \tilde{I}_n\right) \quad (n = 10, 9, \cdots, 1).$$

Similarly,

$$|e_{n-1}| = \frac{1}{5}|e_n| \quad (n = 10, 9, \cdots, 1),$$

or

$$|e_{10-k}| = \left(\frac{1}{5}\right)^k |e_{10}| \quad (k = 1, 2, \cdots, 10).$$

The error is $\frac{1}{5}$ times as much as that in the previous step. This algorithm (1.23) is numerical stable.

From Weighted Mean Value Theorem for Integrals,

$$I_n = \frac{1}{\xi_n + 5}\int_0^1 x^n \mathrm{d}x = \frac{1}{\xi_n + 5} \cdot \frac{1}{n+1} \quad (0 < \xi_n < 1)$$

so

$$\frac{1}{6} \cdot \frac{1}{n+1} < I_n < \frac{1}{5} \cdot \frac{1}{n+1}.$$

Choose

$$\tilde{I}_{10} = \frac{1}{2}\left(\frac{1}{6} \cdot \frac{1}{10+1} + \frac{1}{5} \cdot \frac{1}{10+1}\right) = \frac{1}{60}$$

and

$$\left|I_{10} - \tilde{I}_{10}\right| \le \frac{1}{2}\left(\frac{1}{5} \cdot \frac{1}{10+1} - \frac{1}{6} \cdot \frac{1}{10+1}\right) = \frac{1}{660}$$

The computing results by (1.23) are listed as follows:

$$\begin{aligned}
&\tilde{I}_9 = \tfrac{1}{5}\left(\tfrac{1}{10} - \tilde{I}_{10}\right) = 0.0166667, &&\tilde{I}_8 = \tfrac{1}{5}\left(\tfrac{1}{9} - \tilde{I}_9\right) = 0.0188889\\
&\tilde{I}_7 = \tfrac{1}{5}\left(\tfrac{1}{8} - \tilde{I}_8\right) = 0.0212222, &&\tilde{I}_6 = \tfrac{1}{5}\left(\tfrac{1}{7} - \tilde{I}_7\right) = 0.0243270\\
&\tilde{I}_5 = \tfrac{1}{5}\left(\tfrac{1}{6} - \tilde{I}_6\right) = 0.0284679, &&\tilde{I}_4 = \tfrac{1}{5}\left(\tfrac{1}{5} - \tilde{I}_5\right) = 0.0343064\\
&\tilde{I}_3 = \tfrac{1}{5}\left(\tfrac{1}{4} - \tilde{I}_4\right) = 0.0431387, &&\tilde{I}_2 = \tfrac{1}{5}\left(\tfrac{1}{3} - \tilde{I}_3\right) = 0.0580389\\
&\tilde{I}_1 = \tfrac{1}{5}\left(\tfrac{1}{2} - \tilde{I}_2\right) = 0.0883922, &&\tilde{I}_0 = \tfrac{1}{5}\left(1 - \tilde{I}_1\right) = 0.1823216.
\end{aligned}$$

□

From the example, numerical stable algorithm is better in practice.

1.4 Ill-conditioned Problem

If a problem in which a small error in the data or in subsequent calculation results in much larger errors in the answers, the problem is said to be ill conditioned.

Exercise 1.7. *Analyze the roots of the following equation:*

$$\begin{aligned} p(x) &= (x-1)(x-2)\cdots(x-20) \\ &= x^{20} - 210x^{19} + \cdots = 0, \end{aligned}$$

if the coefficient -210 *is changed to* $-210 + 2^{-23}$.

Solution: If the coefficient -210 is changed to $-210 + 2^{-23}$, then the equation becomes:

$$p(x) + 2^{-23}x^{19} = 0.$$

The roots of the above equation are

$$\begin{array}{ccc} 1.000000000, & 2.000000000, & 3.000000000 \\ 4.000000000, & 4.999999928, & 6.000006944 \\ 6.999697234, & 8.007267603, & 8.917250249 \end{array}$$

$$10.095266145 \pm 0.643500904\mathrm{i}, \quad 11.793633881 \pm 1.652329728\mathrm{i}$$

$$13.992358137 \pm 2.518830070\mathrm{i}, \quad 16.730737466 \pm 2.812624894\mathrm{i}$$

$$19.502439400 \pm 1.940330347\mathrm{i}, \quad 20.846908101.$$

where some of the roots are complex.

Let's analyze the reasons for this phenomenon. Denote

$$p(x,\varepsilon) = p(x) + \varepsilon x^{19} = x^{20} + (-210+\varepsilon)x^{19} + \cdots$$

Then the zeros of $p(x,\varepsilon)$ are all functions of ε, which are denoted as

$$x_i(\varepsilon) \quad (i = 1, 2, \cdots, 20).$$

When $\varepsilon \to 0$, $x_i(\varepsilon) \to i (i = 1, 2, \cdots, 20)$, we have

$$\begin{aligned} p(x,\varepsilon) &= (x - x_1(\varepsilon))(x - x_2(\varepsilon)) \cdots (x - x_{20}(\varepsilon)) \\ &= (x - x_i(\varepsilon)) \prod_{\substack{j=1 \\ j \neq i}}^{20} (x - x_j(\varepsilon)). \end{aligned}$$

TABLE 1.1
Values of $\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon}$

$x_i(0)$	$\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon}\big\vert_{\varepsilon=0}$	$x_i(0)$	$\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon}\big\vert_{\varepsilon=0}$
1	8.2×10^{-18}	11	4.6×10^{7}
2	-8.2×10^{-11}	12	-2.0×10^{8}
3	1.6×10^{-6}	13	6.1×10^{8}
4	-2.2×10^{-3}	14	-1.3×10^{9}
5	6.1×10^{-1}	15	2.1×10^{9}
6	-5.8×10^{1}	16	-2.4×10^{9}
7	2.5×10^{3}	17	1.9×10^{9}
8	-6.0×10^{4}	18	-1.0×10^{9}
9	8.3×10^{5}	19	3.1×10^{8}
10	-7.6×10^{6}	20	-4.3×10^{7}

Let us find the value of $\frac{\mathrm{d}x_i(\varepsilon)}{\mathrm{d}\varepsilon}\Big|_{\epsilon=0}$. Deriving both sides of the above formula with respect to ε, we get

$$x^{19} = \left[-\frac{\mathrm{d}x_i(\varepsilon)}{\mathrm{d}\varepsilon}\right] \prod_{\substack{j=1\\ j\neq i}}^{20} (x - x_j(\varepsilon)) + (x - x_i(\varepsilon)) \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \prod_{\substack{j=1\\ j\neq i}}^{20} (x - x_j(\varepsilon)) .$$

Let $\varepsilon \to 0$, we have

$$x^{19} = -\frac{\mathrm{d}x_i(0)}{\mathrm{d}\epsilon} \prod_{\substack{j=1\\ j\neq i}}^{20} (x-j) + (x-i) \left[\frac{\mathrm{d}}{\mathrm{d}\varepsilon} \prod_{j=1}^{20} (x - x_j(\varepsilon))\right]\Bigg|_{t=0}$$

and let $x \to i$, we have

$$i^{19} = -\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon} \prod_{\substack{j=1\\ j\neq i}}^{20} (i-j).$$

We get

$$\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon} = -\frac{i^{19}}{\prod_{\substack{j=1\\ j\neq i}}^{20}} \quad (i = 1, 2, \cdots, 20).$$

Their values are shown in the Table 1.1 . From

$$x_i(\varepsilon) - x_i(0) \approx \frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon}(\varepsilon - 0),$$

we get

$$x_i(\varepsilon) - x_i(0)| \approx |\frac{\mathrm{d}x_i(0)}{\mathrm{d}\varepsilon} | \cdot\varepsilon > 10^6 \cdot \varepsilon \quad (i = 10, 11, \cdots, 20).$$

It can be seen that the big change on the solution is caused the miner error of the parameter of x^{19}. Solving an algebraic equation of degree 20 is an ill-conditioned problem.

□

1.5 Hornor's method

Exercise 1.8. *Compute x^{22}.*

Solution: If x is multiplied one by one, 21 multiplications are required. On the other hand,

$$x^{22} = x \cdot x^3 \cdot x^6 \cdot x^{12} = x \cdot u \cdot v \cdot w$$

where $u = x \cdot x \cdot x, v = u \cdot u, w = v \cdot v$ and only 7 multiplications are needed.

Compute

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n.$$

If it is computed directly,

$$n + (n-1) + \cdots + 2 + 1 = \frac{n(n+1)}{2}$$

multiplications are required. We rewrite the the polynomial by nesting technique:

$$f(x) = \left(a_0x^{n-1} + a_1x^{n-2} + \cdots + a_{n-1}\right) x + a_n,$$

$$f(x) = \left(\left(a_0x^{n-2} + a_1x^{n-3} + \cdots + a_{n-2}\right) x + a_{n-1}\right) x + a_n,$$

and

$$f(x) = (\cdots((a_0x + a_1) x + a_2) x + \cdots + a_{n-1}) x + a_n.$$

Let

$$\begin{aligned} b_0 &= a_0 \\ b_1 &= b_0x + a_1 \\ b_2 &= b_1x + a_2 \\ &\vdots \\ b_n &= b_{n-1}x + a_n, \end{aligned}$$

then we have

$$\begin{cases} b_k = b_{k-1}x + a_k \quad (k = 1, 2, \cdots, n), \\ b_0 = a_0. \end{cases}$$

This is Hornor's method which can be computed in the simple way:

$$\begin{array}{c|cccccc} & a_0 & a_1 & a_2 & \cdots & a_{n-1} & a_n \\ x = x_0 & & b_0x_0 & b_1x_0 & \cdots & b_{n-2}x_0 & b_{n-1}x_0 \\ \hline & b_0 & b_1 & b_2 & \cdots & b_{n-1} & \boxed{b_n} = f(x_0) \end{array}$$

Exercise 1.9. *Let* $f(x) = 8x^5 + 4x^3 - 9x + 1$, *and compute* $f(3)$ *by Hornor's method.*

Solution

$$\begin{array}{r|cccccc}
 & 8 & 0 & 4 & 0 & -9 & 1 \\
x_0 = 3 & & 24 & 72 & 228 & 684 & 2025 \\
\hline
 & 8 & 24 & 76 & 228 & 675 & \boxed{2026} = f(3)
\end{array}$$

$f(3) = 2026$ is obtained.

Exercise 1.10. *Let* $f(x) = 2(x-5)^4 - 3(x-5)^3 + (x-5) + 3$ *and compute* $f(4.9)$ *by Hornor's method.*

Solution: Let $z = x - 5$. When $x_0 = 4.9$, $z_0 = x_0 - 5 = -0.1$.

$$\begin{array}{r|ccccc}
 & 2 & -3 & 0 & 1 & 3 \\
z_0 = -0.1 & & -0.2 & 0.32 & -0.032 & -0.0968 \\
\hline
 & 2 & -3.2 & 0.32 & 0.968 & \boxed{2.9032} = f(4.9)
\end{array}$$

$f(4.9) = 2.9032$ is obtained directly.

1.6 Excercise

1. x_i is to approximate to x_i^*. Show the significant figures of x_i.
   (1) $x_1^* = 451.023, \quad x_1 = 451.01$;
   (2) $x_2^* = 96 \times 10^5, \quad x_2 = 96.1 \times 10^5$;
   (3) $x_3^* = 0.00096, \quad x_3 = 0.96 \times 10^{-3}$.
2. If $x_1 = 0.973$ has 3 significant figures, compute the relative error of x_1. Let $f(x) = \sqrt{1-x}$, try to compute the bounds of absolute error and relative error of $f(x_1)$.
3. 1.42 and 1.41 are the approximations to $\sqrt{2.01}$ and $\sqrt{2.00}$ which are of 3 significant figures. There are two algorithms: $A^* = \sqrt{2.01} - \sqrt{2.00}$ and $A^* = 0.01/(\sqrt{2.01} + \sqrt{2.00})$ to compute A^*. Try to compute the upper limits of absolute error and relative error of the approximation to A^* by the two methods and show the signifiant figures of the two computing results.
4. Try to change the following expression to make the computing results more accuracy.
   (1) $\left(\frac{1-\cos x}{1+\cos x}\right)^{\frac{1}{2}}$, when $|x| \ll 1$;
   (2) $\sqrt{x+1} - \sqrt{x}$, when $x \gg 1$;
   (3) $\frac{1}{1+2x} - \frac{1-x}{1+x}$, when $|x| \ll 1$;
   (4) $\frac{1-\cos x}{\sin x}$, when $|x| \ll 1$.

5. Consider the sequence $1, \frac{1}{3}, \frac{1}{9}, \frac{1}{27}, \frac{1}{81}, \cdots$. Let $p_0 = 1,\ p_1 = \frac{1}{3}$. If using the following recursive equation

$$p_n = \frac{10}{3} p_{n-1} - p_{n-2}, (n = 2, 3, \cdots)$$

to compute the above sequence, try to analyze the method is stable or not.

6. Let $p(x) = 125x^5+230x^3-11x^2+3x-47$. Compute $p(5)$ by Hornor's method.

7. Let $S_N = \sum_{j=2}^{N} \frac{1}{j^2-1}$ which has exact value $\frac{1}{2}(\frac{3}{2} - \frac{1}{N} - \frac{1}{N+1})$. There are two methods to compute S_N:

   (1) $S_N = \frac{1}{2^2-1} + \frac{1}{3^2-1} + \cdots + \frac{1}{N^2-1}$;

   (2) $S_N = \frac{1}{N^2-1} + \frac{1}{(N-1)^2-1} + \cdots + \frac{1}{2^2-1}$.

   Try to compute $S_{10^2}, S_{10^4}, S_{10^6}$, and compare the computing results and significant figures. (Operation by single floating-point number on computer.)