Scientific Computing and Programming Problems

by
Willi-Hans Steeb
International School for Scientific Computing
at
University of Johannesburg, South Africa

Yorick Hardy
Department of Mathematical Sciences
at
University of South Africa, South Africa

George Dori Anescu
email: george.anescu@gmail.com

Preface

The purpose of this book is to supply a collection of problems in matrix calculus.

Prescribed books for problems.

1) Matrix Calculus and Kronecker Product with Applications and C++ Programs

by Willi-Hans Steeb
World Scientific Publishing, Singapore 1997
ISBN 981 023 2411
http://www.worldscibooks.com/mathematics/3572.html

2) Problems and Solutions in Introductory and Advanced Matrix Calculus

by Willi-Hans Steeb
World Scientific Publishing, Singapore 2006
ISBN 981 256 916 2
http://www.worldscibooks.com/mathematics/6202.html

3) Continous Symmetries, Lie Algebras, Differential Equations and Computer Algebra, second edition

by Willi-Hans Steeb
World Scientific Publishing, Singapore 2007
ISBN 981-256-916-2
http://www.worldscibooks.com/physics/6515.html

4) Problems and Solutions in Quantum Computing and Quantum Information, second edition

by Willi-Hans Steeb and Yorick Hardy
World Scientific, Singapore, 2006
ISBN 981-256-916-2
http://www.worldscibooks.com/physics/6077.html

The International School for Scientific Computing (ISSC) provides certificate courses for this subject. Please contact the author if you want to do this course or other courses of the ISSC.

e-mail addresses of the author:

steebwilli@gmail.com
steeb_wh@yahoo.com

Home page of the author:

http://issc.uj.ac.za

Contents

Notation

$:=$	is defined as
$\in$	belongs to (a set)
$\notin$	does not belong to (a set)
$\cap$	intersection of sets
$\cup$	union of sets
$\emptyset$	empty set
$\mathbb{N}$	set of natural numbers
$\mathbb{Z}$	set of integers
$\mathbb{Q}$	set of rational numbers
$\mathbb{R}$	set of real numbers
$\mathbb{R}^+$	set of nonnegative real numbers
$\mathbb{C}$	set of complex numbers
$\mathbb{R}^n$	n-dimensional Euclidean space
	space of column vectors with n real components
$\mathbb{C}^n$	n-dimensional complex linear space
	space of column vectors with n complex components
$\mathcal{H}$	Hilbert space
i	$\sqrt{-1}$
$\Re z$	real part of the complex number z
$\Im z$	imaginary part of the complex number z
$\|z\|$	modulus of complex number z
	$\|x+iy\| = (x^2+y^2)^{1/2}, \;\; x, y \in \mathbb{R}$
$T \subset S$	subset T of set S
$S \cap T$	the intersection of the sets S and T
$S \cup T$	the union of the sets S and T
$f(S)$	image of set S under mapping f
$f \circ g$	composition of two mappings $(f \circ g)(x) = f(g(x))$
$\mathbf{x}$	column vector in $\mathbb{C}^n$
$\mathbf{x}^T$	transpose of $\mathbf{x}$ (row vector)
$\mathbf{0}$	zero (column) vector
$\|\,.\,\|$	norm
$\mathbf{x} \cdot \mathbf{y} \equiv \mathbf{x}^* \mathbf{y}$	scalar product (inner product) in $\mathbb{C}^n$
$\mathbf{x} \times \mathbf{y}$	vector product in $\mathbb{R}^3$
A, B, C	$m \times n$ matrices
$\det(A)$	determinant of a square matrix A
tr(A)	trace of a square matrix A
$\mathrm{rank}(A)$	rank of matrix A
A^T	transpose of matrix A
$\overline{A}$	conjugate of matrix A

A^*	conjugate transpose of matrix A
$A^\dagger$	conjugate transpose of matrix A (notation used in physics)
A^{-1}	inverse of square matrix A (if it exists)
I_n	$n \times n$ unit matrix
I	unit operator
0_n	$n \times n$ zero matrix
AB	matrix product of $m \times n$ matrix A and $n \times p$ matrix B
$A \bullet B$	Hadamard product (entry-wise product) of $m \times n$ matrices A and B
$[A, B] := AB - BA$	commutator for square matrices A and B
$[A, B]_+ := AB + BA$	anticommutator for square matrices A and B
$A \otimes B$	Kronecker product of matrices A and B
$A \oplus B$	Direct sum of matrices A and B
δ_{jk}	Kronecker delta with $\delta_{jk} = 1$ for $j = k$ and $\delta_{jk} = 0$ for $j \neq k$
λ	eigenvalue
ϵ	real parameter
t	time variable
$\hat{H}$	Hamilton operator

The Pauli spin matrices are used extensively in the book. They are given by

$$\sigma_1 := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

In some cases we will also use σ_x, σ_y and σ_z to denote σ_1, σ_2 and σ_3.

Chapter 1

Quickies

Problem 1. How would we calculate more efficiently the following functions $(x, x_1, x_2, x_3 \in \mathbb{R})$

$$\begin{aligned} f_1(x) &= 2\sinh(x)\cosh(x) \\ f_2(x) &= \cosh^2(x) - \sinh^2(x) \\ f_3(x) &= \cos^2(x) - \sin^2(x) \\ f_4(\mathbf{x}) &= e^{x_1}e^{x_2}e^{x_3} \\ f_5(x) &= 1 - \frac{1}{1+e^{-x}} \\ f_6(x) &= \frac{1}{\sqrt{1-\tanh^2(x)}}. \end{aligned}$$

Problem 2. (i) Let ω be the frequency and t the time. Simplify

$$e^{i\omega t} + e^{-i\omega t}, \qquad \frac{e^{i\omega t} - e^{-i\omega t}}{2i}.$$

(ii) Let $a, b \in \mathbb{R}$ and $a, b > 0$. Can the expression

$$(a+ib)^{1/3} + (a-ib)^{1/3}$$

be simplified? Show that the number is actually real. Hint. For the complex numbers $z = a + ib$ and $\bar{z} = a - ib$ set $z = re^{i\phi}$ and $\bar{z} = re^{-i\phi}$.
(iii) Let $a, b \in \mathbb{R}$ and $a^2 \neq b^2$. Simplify

$$\frac{1}{2ia}\left(\frac{i}{a-b} + \frac{i}{a+b}\right).$$

Problem 3. (i) Let n be a positive integer. Can the calculation of

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\cdots+\frac{1}{n(n+1)}$$

be simplified for computation?
(ii) Let n be a positive integer. Can the calculation of

$$1^3+2^3+\cdots+n^3$$

be simplified for computation? Hint. The ansatz

$$1^3+2^3+\cdots+n^3=an^4+bn^3+cn^2$$

could be helpful with the coefficients a, b, c to be determined.

Problem 4. Can the expression $\sqrt{3-2\sqrt{2}}$ be simplified for computation? Hint. Let $a>0$ and $b>0$. Calculate $(\sqrt{a}-b)(\sqrt{a}-b)$ and compare coefficients.

Problem 5. Let n be a positive integer and $x, y\in\mathbb{R}$. Can the calculation of

$$\sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k$$

be simplified for computation?

Problem 6. The square of the *Planck length* is defined as

$$\ell_P^2:=\frac{G\hbar}{c^3}$$

where G is the gravitational constant, $\hbar$ is the Planck constant divided by 2π and c is the speed of light. We have in the MKSA-system

$$\begin{aligned}G&=6.6732\cdot 10^{-11}\frac{\mathrm{m}^3}{\mathrm{sec}^2\mathrm{kg}}\\ \hbar&=1.0545919\cdot 10^{-34}\frac{\mathrm{kgm}^2}{\mathrm{sec}}\\ c&=2.9979250\cdot 10^{8}\frac{\mathrm{m}}{\mathrm{sec}}.\end{aligned}$$

Write a C++ program that calculates this quantity. Is the data type `double` sufficient? Extend the calculation to find the Planck time interval and the Planck mass

$$t_P=\sqrt{\frac{G\hbar}{c^5}},\qquad m_P=\sqrt{\frac{c\hbar}{G}}.$$

Problem 7. Let $n \in \mathbb{N}$. How can the calculation of

$$x(e^{-x} + e^{-2x} + \cdots + e^{-nx})$$

be simplified for n large?

Problem 8. Let $\mathbb{Z}$ be the integer numbers. Let $\mathbb{N}_0$ be the natural numbers including 0. Find a $1-1$ map

$$f : \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \to \mathbb{N}_0$$

with $f(0,0,0) = 0$ and $f(1,0,0) = 1$. The number of nearest neighbours are 6. Let $(j_1, j_2, j_3) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$. Then the six nearest neighbours are

$$(j_1+1, j_2, j_3), \quad (j_1, j_2+1, j_3), \quad (j_1, j_2, j_3+1)$$

$$(j_1-1, j_2, j_3), \quad (j_1, j_2-1, j_3), \quad (j_1, j_2, j_3-1).$$

Give a C++ implementation using the `Verylong` class of SymbolicC++. Give a Java implementation using the `BigInteger` class.

Problem 9. (i) Let $\epsilon \in \mathbb{R}$ and $\epsilon > 0$. Let $\mathbf{v}$ be a nonzero vector in $\mathbb{R}^n$. Assume that $\|\mathbf{v}\| \gg \epsilon$. Show that

$$\sqrt{\mathbf{v}^T\mathbf{v} \pm \epsilon} \approx \|\mathbf{v}\| \pm \frac{\epsilon}{2\mathbf{v}^T\mathbf{v}}.$$

(ii) Let $x, \ell \geq 0$ and $x \ll \ell$. Show that

$$\sqrt{\ell^2 + x^2} - \ell \approx \frac{x^2}{2\ell}.$$

Problem 10. (i) Let N_1, N_2 be given positive integers. Let $n_1 = 0, 1, \ldots, N_1 - 1$, $n_2 = 0, 1, \ldots, N_2 - 1$. There are $N_1 \cdot N_2$ points. The points (n_1, n_2) are a subset of $\mathbb{N}_0 \times \mathbb{N}_0$ and can be mapped one-to-one onto a subset of $\mathbb{N}_0$

$$j(n_1, n_2) = n_1 N_2 + n_2$$

where $j = 0, 1, \ldots, N_1 \cdot N_2 - 1$. Find the inverse of this map. Consider first the case $N_1 = N_2 = 2$.
(ii) Give a C++ implementation of the map and the inverse.

Problem 11. Let $n \in \mathbb{N}$. The map $f : [0, 2n] \to [0, 2n]$ on the integers defined by

$$\begin{aligned} f(0) &= n \\ f(k) &= 2n + 1 - k \text{ for } 0 < k \leq n \\ f(k) &= 2n - k \text{ for } n < k \leq 2n \end{aligned}$$

plays a role for the converse of Sarkovskii's theorem.
(i) Let $n = 2$. Starting with 1 find

$$f(1),\ \ f(f(1)),\ \ f(f(f(1))),\ \ f(f(f(f(1)))),\ \ f(f(f(f(f(1))))).$$

Discuss.
(ii) Give a C++ implementation of this map. The user provides the n.

Problem 12. Show that

$$\frac{1}{a+n-k+1}\left(\frac{1}{a-b+1}+\frac{1}{n-k+b}\right)=\frac{1}{(a-b+1)(n-k+b)}.$$

Problem 13. Given a vector of length n. Write a C++ program that checks whether all entries are pairwise different.

Problem 14. The *sinc function* $f : \mathbb{R} \to \mathbb{R}$

$$f(x) = \frac{\sin(\pi x)}{\pi x}$$

can be evaluated using the series expansion

$$f(x) = 1 - \frac{1}{3!}(\pi x)^2 + \frac{1}{5!}(\pi x)^4 - \cdots$$

However the sinc function could also be evaluated from

$$f(x) = \prod_{k=1}^{\infty}\left(1-\frac{x^2}{k^2}\right).$$

Compare the two methods.

Problem 15. The quadratic equation $x^2 = x + 1$ has the solutions

$$\tau = \frac{1}{2}(1+\sqrt{5}), \qquad \sigma = \frac{1}{2}(1-\sqrt{5})$$

(golden mean numbers). Let $k \in \mathbb{Z}$. Can the expressions

$$\tau^{k-1} + \tau^{k-2}, \qquad \sigma^{k-1} + \sigma^{k-2}$$

be simplified?

Problem 16. How would one calculate more efficiently (i.e. minimizing the number of multiplications) the analytic function $f : \mathbb{R} \to \mathbb{R}$

$$f(x) = x + 2x^2 + 3x^3$$

for a given x.

Problem 17. Consider the analytic function $f : \mathbb{R} \to \mathbb{R}$

$$f(x) = \frac{x}{1+x^2}.$$

Simplify the calculation of the integral

$$\int_{-1}^{2} f(x)dx.$$

Hint. First show that $f(x) = -f(-x)$.

Problem 18. (S) Solve the quadratic equation $\omega^2 + \omega + 1 = 0$ by multiplying this equation with ω and inserting the quadratic equation and then solving the resulting cubic equation. Select the solutions from the cubic equation which are also solutions of the quadratic equation. Note that $1 \equiv \exp(i2\pi)$.

Problem 19. Find numerically solutions of the transcendental equation

$$e^{-x} + \frac{x}{5} - 1 = 0$$

for $x \geq 0$.

Problem 20. (i) Let m be a mass, E an energy, a a length and $\hbar$ the Planck constant (divided by 2π). Show that

$$\epsilon := \frac{\sqrt{2m|E|}a}{\hbar}$$

be dimensionless.
(ii) Let e be the charge, E the electric field, m the mass, ω the frequency and c the speed of light (all in SI units). Is

$$\eta = \frac{eE}{m\omega c}$$

a dimensionless quantity so that cases such as $\eta \ll 1$ can be studied?

Problem 21. Let $n_0, n_1, n_2 \in \mathbb{N}_0$. Implement the function

$$f(n_0, n_1, n_2) = \frac{(n_0 + n_1 + n_2)!}{n_0! n_1! n_2!}$$

in SymbolicC++ utilizing the `Verylong` class and `Rational` class.

Problem 22. Calculate efficiently

$$\int_0^7 \sin(x)dx, \qquad \int_0^7 \cos(x)dx.$$

Problem 23. (S) Let $i, j, k \in \mathbb{N}_0$. Find all solutions of $i + j + k = 3$. Give the solution in lexicographical order.

Problem 24. The number π can be calculated from the expansion

$$\pi = \sum_{j=0}^{\infty} \frac{(j!)^2 2^{j+1}}{(2j+1)!}.$$

Let n be positive integer. Give an implementation of the sum

$$s = \sum_{j=0}^{n} \frac{(j!)^2 2^{j+1}}{(2j+1)!}$$

with SymbolicC++ using the `Verylong` and `Rational` class and so find an approximation of π.

Problem 25. Let $\mathbb{N}_0$ be the set of natural numbers including 0. Let $n_1, n_2, n_3 \in \mathbb{N}_0$. An invertible function $f : \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \mapsto \mathbb{N}_0$ is defined as follows. Let $n = n_1 + n_2 + n_3$. For a fixed n we have $f(n) < f(n+1)$ and within a fixed n a lexicographical ordering is assumed. For the first ten elements one has

$$(0,0,0) \to 0,\ (0,0,1) \to 1,\ (0,1,0) \to 2,\ (1,0,0) \to 3,$$

$$(0,0,2) \to 4,\ (0,1,1) \to 5,\ (1,0,1) \to 7,\ (1,1,0) \to 8,\ (2,0,0) \to 9$$

Give a C++ implementation of the function f and its inverse f^{-1} using templates so that the `Verylong` class of SymbolicC++ can be used. Give a Java implementation using the `BigInteger` class.

Problem 26. Let $f \in L_2(\mathbb{R})$. *Poisson's summation formula* in one dimension is given by

$$\sum_{n=-\infty}^{+\infty} f(n) = \sum_{q=-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x) e^{-2\pi i q x} dx.$$

Show that if f is an even function of x, then the summation formula can be written as

$$\sum_{n=1}^{+\infty} f(n) = -\frac{1}{2} f(0) + \int_0^{+\infty} f(x)dx + 2\sum_{q=1}^{+\infty} \int_0^{+\infty} f(x)\cos(2\pi q x)dx.$$

Apply it to the function $f(x) = e^{-|x|}$.

Problem 27. The *Catalan constant* is defined as

$$G = 1 - 3^{-2} + 5^{-2} - 7^{-2} + \cdots$$

Give a SymbolicC++ implementation using the `Verylong` and `Rational` class to find an approximation of the constant.

Problem 28. Let $n \in \mathbb{Z}$. Simplify $\sin(n\pi)$, $\cos(2n\pi)$, $\cos((2n+1)\pi)$.

Problem 29. Consider the normalized vectors

$$\mathbf{n}_j := \begin{pmatrix} \sin(\theta_j)\cos(\phi_j) \\ \sin(\theta_j)\sin(\phi_j) \\ \cos(\theta_j) \end{pmatrix}, \quad \mathbf{n}_k := \begin{pmatrix} \sin(\theta_k)\cos(\phi_k) \\ \sin(\theta_k)\sin(\phi_k) \\ \cos(\theta_k) \end{pmatrix}$$

in $\mathbb{R}^3$. Find the scalar product

$$\mathbf{n}_j \cdot \mathbf{n}_k$$

and simplify it. The scalar product is the angle between the two vectors.

Problem 30. (i) Let $\mathbf{u}$, $\mathbf{v}$ be (column) vectors in the Euclidean space $\mathbb{R}^n$. Now $\mathbf{u}^T$, $\mathbf{v}^T$ are the corresponding row vectors (^{T} denotes transpose) and thus $\mathbf{u}^T\mathbf{v}$ is the scalar product of $\mathbf{u}$ and $\mathbf{v}$. What does

$$A := \sqrt{|(\mathbf{u}^T\mathbf{u})(\mathbf{v}^T\mathbf{v}) - (\mathbf{u}^T\mathbf{v})^2|}$$

calculate?
(ii) Consider $\mathbb{R}^4$ and the vectors

$$\mathbf{u} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Calculate A.

Problem 31. (S) (i) Let $\mathbf{u}$, $\mathbf{v}$ be column vectors in $\mathbb{C}^n$ and thus $\mathbf{u}^*$, $\mathbf{v}^*$ (transpose and complex conjugate) are row vectors. Calculate efficiently

$$\mathrm{tr}(\mathbf{u}\mathbf{u}^*\mathbf{v}\mathbf{v}^*).$$

Note that $\mathbf{u}\mathbf{u}^*\mathbf{v}\mathbf{v}^*$ is an $n \times n$ matrix. Could one utilize that matrix multiplication is associative? Discuss. Is

$$\mathrm{tr}(\mathbf{u}\mathbf{u}^*\mathbf{v}\mathbf{v}^*) \geq 0\,?$$

Prove or disprove.
(ii) Let A be a 2×2 matrix. Calculate efficiently $\mathrm{tr}(A^2)$.

Problem 32. Consider the 2×2 matrices

$$C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{11} \end{pmatrix}, \qquad a_{11}, a_{12} \in \mathbb{R}.$$

Can the expression

$$A^3 + 3AC(A + C) + C^3$$

be simplified for computation?

Problem 33. Given two invertible $n \times n$ matrices A and B.
(i) How can we calculate $B^{-1}A^{-1}$ more efficiently?
(ii) Let $\otimes$ be the Kronecker product. How can we calculate $B^{-1} \otimes A^{-1}$ more efficiently?

Problem 34. (i) Let

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \in \mathbb{R}^3.$$

What does

$$\frac{1}{2} \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix}$$

calculate?
(ii) Consider the coordinates

$$\mathbf{p}_1 = (x_1, y_1, z_1)^T, \quad \mathbf{p}_2 = (x_2, y_2, z_2)^T, \quad \mathbf{p}_3 = (x_3, y_3, z_3)^T$$

with $\mathbf{p}_1 \neq \mathbf{p}_2$, $\mathbf{p}_2 \neq \mathbf{p}_3$, $\mathbf{p}_3 \neq \mathbf{p}_1$. We form the vectors

$$\mathbf{v}_{21} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix}, \quad \mathbf{v}_{31} = \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{pmatrix}.$$

Let $\times$ be the vector product. What does

$$\frac{1}{2} |\mathbf{v}_{21} \times \mathbf{v}_{31}|$$

calculate? Apply it to $\mathbf{p}_1 = (0, 0, 0)^T$, $\mathbf{p}_2 = (1, 0, 1)^T$, $\mathbf{p}_3 = (1, 1, 1)^T$.

Problem 35. What is the output of the following C++ program

```
// whileloop.cpp

#include <iostream>
using namespace std;

int main(void)
{
  int x = 0; int t = 0; int p = 0;
  while(t < 100) {
   if(p==0) x = x + 2;
   if(p==1) x = x - 1;
  p = 1 - p;
  t++;
  } // end while
  cout << "x = " << x << endl;
  cout << "p = " << p << endl;
  cout << "t = " << t << endl;
  return 0;
}
```

Problem 36. The surface area of a torus with inner radius a and outer radius b is a given by

$$A = \pi^2(b^2 - a^2).$$

The formula for the volume of a torus is given by

$$V = \frac{\pi^2}{4}(a+b)(b-a)^2.$$

Simplify the calculation of V given A.

Problem 37. Let a, b be non-negative integers.
(i) Simplify the expression

$$E_1 = \sqrt{a + \sqrt{-b}} + \sqrt{a - \sqrt{-b}}.$$

(ii) Simplify the expression

$$E_2 = \sqrt{a + \sqrt{-b}} - \sqrt{a - \sqrt{-b}}.$$

Problem 38. Let $x \in [-1, 1]$. Simplify $\arcsin(x) + \arccos(x)$.

Problem 39. Simplify

$$f(\alpha, \beta, \gamma) = \sin(\alpha + \beta - \gamma) + \sin(\beta + \gamma - \alpha) + \sin(\gamma + \alpha - \beta) - \sin(\alpha + \beta + \gamma).$$

Problem 40. Let $f : \mathbb{R} \to \mathbb{R}$ be an analytic function. Consider the central *difference operator* δ defined by

$$\delta(f(x)) := f(x + \frac{1}{2}h) - f(x - \frac{1}{2}h)$$

where $h > 0$ is the step length. The operator δ is linear. Find $\delta(\delta(f(x))$.

Problem 41. Consider the coordinates in $\mathbb{R}^3$

$$\mathbf{p}_1 = (x_1, y_1, z_1), \quad \mathbf{p}_2 = (x_2, y_2, z_2), \quad \mathbf{p}_3 = (x_3, y_3, z_3)$$

with $\mathbf{p}_1 \neq \mathbf{p}_2$, $\mathbf{p}_2 \neq \mathbf{p}_3$, $\mathbf{p}_3 \neq \mathbf{p}_1$. We form the two vectors

$$\mathbf{v} = \mathbf{p}_2 - \mathbf{p}_1, \qquad \mathbf{w} = \mathbf{p}_3 - \mathbf{p}_1.$$

What does $\frac{1}{2}|\mathbf{v} \times \mathbf{w}|$ calculate?

Problem 42. Let n be a positive integer. Give a C++ implementation of sum

$$\sum_{k=0}^{n} \sum_{\ell=0}^{n} \binom{k}{\ell}$$

using templates so that the `Verylong` class of SymbolicC++ can be used.

Problem 43. Let $x \in \mathbb{R}$. Show that

$$\frac{1}{e^{-x} + 1} \equiv 1 - \frac{1}{e^x + 1}.$$

Problem 44. Find a good approximation of $\sqrt{29}$ utilizing

$$29 \equiv 36 \left(1 - \frac{7}{36}\right) \equiv 6^2 \left(1 - \frac{7}{36}\right)$$

and an expansion.

Problem 45. Let $n \in \mathbb{Z}$. Show that

$$\cos((n+1)\alpha) + \cos((n-1)\alpha) \equiv 2\cos(\alpha)\cos(n\alpha)$$
$$\sin((n+1)\alpha) + \sin((n-1)\alpha) \equiv 2\cos(\alpha)\sin(n\alpha).$$

Problem 46. Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous function. Assume that $f(x) = -f(-x)$. Let $a < 0$ and $b > 0$. Simplify the calculation of

$$\int_a^b f(x)dx.$$

Problem 47. Calulate approximative $(72)^{1/6}$ utilizing the expression

$$(72)^{1/6} = 2\left(1 + \frac{1}{8}\right)^{1/6}.$$

Chapter 2

Bitwise Operations

Problem 1. Let $x, y \in \{0, 1\}$. Show that the NOT-gate, AND-gate, OR-gate, NAND-gate, NOR-gate and XOR-gate can be expressed using arithmetic operations (i.e. addition, subtraction, multiplication).

Problem 2. Consider a binary $n \times n$ matrix, where we count the entries from 0. We have $b_{00} = 1$ and $b_{n-1n-1} = 1$. The other 0-1 entries are generated randomly. An ant at entry $(0, 0)$ can only move to the right or down (not diagonal) when this entry contains a 1. Write a C++ program that checks whether the ant could reach the entry $(n - 1, n - 1)$. For example, consider the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

For this case one path

$$(0,0) \to (1,0) \to (1,1) \to (1,2) \to (2,2) \to (3,2) \to (3,3) \to (4,3) \to (4,4)$$

would be possible. Note that the ant could also get stuck at $(2, 0)$.

Problem 3. Linear feedback shift registers play a role in the generation of pseudo-random numbers. A linear feedback shift register is a shift register whose input is a linear function of its previous state. What is the output of the following C++ program?

```
// lfsr.cpp
```

```
#include <iostream>
using namespace std;

int main(void)
{
 unsigned short i = 0xACE1u; // hex notation
 cout << "i = " << i << endl;
 unsigned short b;
 unsigned short counter = 0u;

 do
 {
 b = ((i)^(i >> 2)^(i >> 3)^(i >> 5)) & 1;
 i = (i >> 1) | (b << 15);
 counter++;
 cout << "b = " << b << endl;
 cout << "i = " << i << endl;
 }
 while(i != 0xACE1u);
 cout << "leaving while-loop" << endl;
 cout << "b = " << b << endl;
 cout << "i = " << i << endl;
 cout << "counter = " << counter << endl;
 return 0;
}
```

Note that `^` is the XOR-operation, `|` is the OR-operation and `&` is the AND-operation. Note that `unsigned short` has 16 bits.

Problem 4. A boolean function $f : \{0,1\}^n \to \{0,1\}$ ($x_j \in \{0,1\}, j = 1,\ldots,n$) can be transformed from the domain $\{0,1\}$ into the spectral domain by a linear transformation

$$T\mathbf{y} = \mathbf{s}$$

where T is a $2^n \times 2^n$ orthogonal matrix, $\mathbf{y} = (y_0, y_1, \ldots, y_{2^n-1})^T$ is the two-valued ($\{+1,-1\}$ with $0 \leftrightarrow 1$, $1 \leftrightarrow -1$) truth table of the boolean function and spectral coefficients s_j. Since T is invertible we have

$$T^{-1}\mathbf{s} = \mathbf{y}.$$

For T we select the Hadamard matrix. The $(n+1) \times (n+1)$ Hadamad matrix is recursively defined as

$$H(n) = \begin{pmatrix} H(n-1) & H(n-1) \\ H(n-1) & -H(n-1) \end{pmatrix}, \quad n = 1, 2, \ldots$$

with $H(0) = (1)$ ((1×1) matrix). The inverse of $H(n)$ is given by

$$H(n)^{-1} = \frac{1}{2^n} H(n).$$

Now any boolean function can be expanded as the arithmetical polynomial

$$f(x_1,\dots,x_n)=\frac{1}{2^{n+1}}(2^n-s_0-s_1(-1)^{x_n}-s_2(-1)^{x_{n-1}}-\cdots-s_{2^n-1}(-1)^{x_1\oplus x_2\oplus\cdots\oplus x_n}$$

where $\oplus$ denotes the modulo-2 addition and the

$$(s_0,s_1,\dots,s_{2^n-1})=\mathbf{s}$$

are the spectral coefficients. Consider the boolean function $f:\{0,1\}^3\to\{0,1\}$

$$f(x_1,x_2,x_3)=\bar{x}_1\cdot\bar{x}_2\cdot\bar{x}_3+\bar{x}_1\cdot x_2\cdot\bar{x}_3+x_1\cdot x_2\cdot\bar{x}_3.$$

Find the truth table, the vector $\mathbf{y}$ and then using $H(3)$ calculate the spectral coefficients s_j $(j=0,1,\dots,7)$.

Problem 5. Analogously to the Hamming distance for finite sequences, a metric can be used to compute distances between infinite $u(j)$ and $v(j)$, where $j\in\mathbb{Z}$

$$d(u,v):=\sum_{j\in\mathbb{Z}}\frac{|u(j)-v(j)|}{2^{|j|}}.$$

Consider the infinite alternating sequences

```
...010101010...  = u
...101010101...  = v
       ^
       |
       0
```

Find the distance between u and v.

Problem 6. Let $x,y\in\{0,1\}$ and $\cdot$ the AND operation. Is the circuit

$$x'=x,\qquad y'=x\cdot y$$

a reversible gate?

Problem 7. Consider the unit cube with the vertices

$$(0,0,0),(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1)$$

The *majority gate* is given by

$$\begin{aligned}(0,0,0)&\mapsto 0,\quad (0,0,1)\mapsto 0\\ (0,1,0)&\mapsto 0,\quad (0,1,1)\mapsto 1\\ (1,0,0)&\mapsto 0,\quad (1,0,1)\mapsto 1\\ (1,1,0)&\mapsto 1,\quad (1,1,1)\mapsto 1.\end{aligned}$$

Find the boolean expression.

Problem 8. Consider the truth table

$$\begin{aligned}(0,0,0)&\mapsto 1, \quad (0,0,1)\mapsto 0\\ (0,1,0)&\mapsto 0, \quad (0,1,1)\mapsto 0\\ (1,0,0)&\mapsto 0, \quad (1,0,1)\mapsto 0\\ (1,1,0)&\mapsto 0, \quad (1,1,1)\mapsto 1.\end{aligned}$$

Find the boolean expression.

Problem 9. Let $x_0, y_0 \in \{0,1\}$. Solve the system of boolean equations

$$x_{t+1} = x_t \oplus y_t, \qquad y_{t+1} = x_t \cdot y_t$$

where $x_0 = 0$, $y_0 = 1$ and $t = 0, 1, \ldots$. Here $\oplus$ denotes the XOR operation and $\cdot$ denotes the AND operation. First find the fixed points, i.e. solve $x \oplus y = x$, $x \cdot y = y$. Does the sequence x_t, y_t tend to a fixed point?

Problem 10. (i) Let $s_1(0), s_2(0), s_3(0) \in \{+1, -1\}$. Study the time-evolution ($t = 0, 1, 2, \ldots$) of the coupled system of equations

$$\begin{aligned}s_1(t+1) &= s_2(t)s_3(t)\\ s_2(t+1) &= s_1(t)s_3(t)\\ s_3(t+1) &= s_1(t)s_2(t)\end{aligned}$$

for the eight possible initial conditions, i.e. (i) $s_1(0) = s_2(0) = s_3(0) = 1$, (ii) $s_1(0) = 1$, $s_2(0) = 1$, $s_3(0) = -1$, (iii) $s_1(0) = 1$, $s_2(0) = -1$, $s_3(0) = 1$, (iv) $s_1(0) = -1$, $s_2(0) = 1$, $s_3(0) = 1$, (v) $s_1(0) = 1$, $s_2(0) = -1$, $s_3(0) = -1$, (vi) $s_1(0) = -1$, $s_2(0) = 1$, $s_3(0) = -1$, (vii) $s_1(0) = -1$, $s_2(0) = -1$, $s_3(0) = 1$, (viii) $s_1(0) = -1$, $s_2(0) = -1$, $s_3(0) = -1$. Which of these initial conditions are fixed points?
(ii) Let $s_1(0), s_2(0), s_3(0) \in \{+1, -1\}$. Study the time-evolution ($t = 0, 1, 2, \ldots$) of the coupled system of equations

$$\begin{aligned}s_1(t+1) &= s_2(t)s_3(t)\\ s_2(t+1) &= s_1(t)s_2(t)s_3(t)\\ s_3(t+1) &= s_1(t)s_2(t)\end{aligned}$$

for the eight possible initial conditions, i.e. (i) $s_1(0) = s_2(0) = s_3(0) = 1$, (ii) $s_1(0) = 1$, $s_2(0) = 1$, $s_3(0) = -1$, (iii) $s_1(0) = 1$, $s_2(0) = -1$, $s_3(0) = 1$, (iv) $s_1(0) = -1$, $s_2(0) = 1$, $s_3(0) = 1$, (v) $s_1(0) = 1$, $s_2(0) = -1$, $s_3(0) = -1$, (vi) $s_1(0) = -1$, $s_2(0) = 1$, $s_3(0) = -1$, (vii) $s_1(0) = -1$, $s_2(0) = -1$, $s_3(0) = 1$, (viii) $s_1(0) = -1$, $s_2(0) = -1$, $s_3(0) = -1$. Which of these initial conditions are fixed points?

Problem 11. Let $x_1(0), x_2(0), x_3(0) \in \{0, 1\}$ and let $\oplus$ be the XOR-operation. Study the time-evolution ($t = 01, 2, \ldots$) of the coupled system of equations

$$\begin{aligned} x_1(t+1) &= x_2(t) \oplus x_3(t) \\ x_2(t+1) &= x_1(t) \oplus x_3(t) \\ x_3(t+1) &= x_1(t) \oplus x_2(t) \end{aligned}$$

for the eight possible initial conditions, i.e. (i) $x_1(0) = x_2(0) = x_3(0) = 0$, (ii) $x_1(0) = 0$, $x_2(0) = 0$, $x_3(0) = 1$, (iii) $x_1(0) = 0$, $x_2(0) = 1$, $x_3(0) = 0$, (iv) $x_1(0) = 1$, $x_2(0) = 0$, $x_3(0) = 0$, (v) $x_1(0) = 0$, $x_2(0) = 1$, $x_3(0) = 1$, (vi) $x_1(0) = 1$, $x_2(0) = 0$, $x_3(0) = 1$, (vii) $x_1(0) = 1$, $x_2(0) = 1$, $x_3(0) = 0$, (viii) $x_1(0) = 1$, $x_2(0) = 1$, $x_3(0) = 1$. Which of these initial conditions are fixed points?

Problem 12. Show that one Fredkin gate is sufficient to implement the XOR gate.

Problem 13. Show that $\overline{a \cdot b} = \overline{a} + \overline{b}$ using (i) truth tables and (ii) properties of boolean algebra (with $a + 1 = 1$).

Chapter 3

Maps and Functions

Problem 1. The straight line *Hough transform* maps a line in $\mathbb{R}^2$ into a point in the Hough transform space. The polar definition of the Hough transform is based on the representation of the lines by the parameters (ρ, θ) via the equation

$$\rho = x_j \cos(\theta) + y_j \sin(\theta).$$

with $\rho \geq 0$ and $\theta \in [0, 2\pi)$. All points (x_j, y_j) of a given line correspond to a point (ρ, θ) in the Hough transform space. Any point (x_j, y_j) is mapped to a sinusoidal curve in the Hough transform space. Consider the two points $(x_0, y_0) = (1, 0)$ and $(x_1, y_1) = (0, 1)$ on a line. Find ρ, θ.

Problem 2. Let $f, g : \mathbb{R} \to \mathbb{R}$ be analytic functions and $n \geq 1$. Then

$$\int_a^b f^{(n)} g dx = f^{(n-1)} g \Big|_a^b - f^{(n-2)} g' \Big|_a^b + f^{(n-3)} g'' \Big|_a^b - \cdots (-1)^n \int_a^b f g^{(n)} dx.$$

Here $f^{(n)}$ denotes the n-th derivative. This identity is called *generalized integration by parts*. Let $\epsilon > 0$. Find

$$\int_0^1 e^{\epsilon x} x^n dx$$

using generalized integration by parts.

Problem 3. Show that expanding the function

$$f(x) := \begin{cases} \sin(2\pi x) & x \in [0, 1] \\ 0 & \text{otherwise} \end{cases}$$

on the Hilbert space of square integrable functions $L_2(\mathbb{R})$ in terms of the Haar basis

$$\{\ \psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k)\ :\ j,k \in \mathbb{Z}\ \}$$

where

$$\psi(x) := \begin{cases} -1 & x \in [0, 1/2] \\ 1 & x \in (1/2, 1] \\ 0 & \text{otherwise} \end{cases}.$$

yields the expansion

$$f(x) = \frac{1}{2\pi}\sum_{j=0}^{\infty}\sum_{k=0}^{2^j-1} 2^{\frac{j}{2}}\left[2\cos\frac{2\pi\left(k+\frac{1}{2}\right)}{2^j} - \cos\frac{2\pi\,(k+1)}{2^j} - \cos\frac{2\pi k}{2^j}\right]$$

by considering $j \leq -1$ and $j \geq 0$ separately.

Problem 4. Find a polynomial

$$p(x) = ax^4 + bx^3 + cx^2 + dx + e$$

which satisfies the conditions

$$p(0) = 0, \quad p(1) = 0, \quad p(1/2) = 0$$
$$p(1/4) = 1, \qquad p(3/4) = 1/2.$$

Problem 5. Let A, B and C be arbitrary non-empty sets and let $f : A \to B$ and $g : B \to C$. The composite function of f and g is the function

$$g \circ f\ :\ A \to C, \quad (g \circ f)(x) = g(f(x)).$$

Notice that $g \circ f$ reads from right to left; it means first apply f, then apply g to the result. Note that function composition is associative.
(i) Let $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^2$, and $g : \mathbb{R} \to \mathbb{R}$, $g(x) = 3x - 1$. Find $g \circ f$ and $f \circ g$.
(ii) Write a C++ program which implements these compositions with x of data type `double`.

Problem 6. Let f_1, f_2 be continuous functions over an interval $[a, b]$. Then we have the identities

$$\min(f_1, f_2) \equiv \frac{1}{2}(f_1 + f_2 - |f_1 - f_2|)$$
$$\max(f_1, f_2) \equiv \frac{1}{2}(f_1 + f_2 + |f_1 - f_2|).$$

Write a C++ program that finds the min and max for two given continuous functions f_1 and f_2 using the function

```
void minmax(double (*f1)(double),double (*f2)(double),
            double x,double& min,double& max)
```

where x is the function parameter. Apply it to the sine function and cosine funtion in the interval $[0, 2]$.

Problem 7. Given a smooth surface in the Euclidean space $\mathbb{R}^3$ described by

$$\mathbf{x}(u, v) = \begin{pmatrix} x_1(u, v) \\ x_2(u, v) \\ x_3(u, v) \end{pmatrix}.$$

The *Gaussian curvature* is calculated as follows. First we calculate $E(u, v)$, $F(u, v)$, $G(u, v)$ of the first fundamental form

$$E(u, v) = \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial \mathbf{x}}{\partial u}, \qquad F(u, v) = \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial \mathbf{x}}{\partial v}, \qquad G(u, v) = \frac{\partial \mathbf{x}}{\partial v} \cdot \frac{\partial \mathbf{x}}{\partial v}$$

where $\cdot$ denotes the scalar product. Next we calculate

$$\mathbf{n}^+(u, v) := \frac{\frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v}}{\left| \frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v} \right|}$$

where $\times$ denotes the vector product. Using $\mathbf{n}^+$ we calculate $L(u, v)$, $M(u, v)$, $N(u, v)$ of the second fundamental form

$$L(u, v) = \mathbf{n}^+ \cdot \frac{\partial^2 \mathbf{x}}{\partial u^2}, \qquad M(u, v) = \mathbf{n}^+ \cdot \frac{\partial^2 \mathbf{x}}{\partial u \partial v}, \qquad N(u, v) = \mathbf{n}^+ \cdot \frac{\partial^2 \mathbf{x}}{\partial v^2}.$$

Then the Gaussian curvature $K(u, v)$ is given by

$$K := \frac{LN - M^2}{EG - F^2}.$$

Write a SymbolicC++ program that calculates K and apply it to the *Möbius band* given by

$$\mathbf{x}(u, v) = \begin{pmatrix} (2 - v \sin(u/2)) \sin(u) \\ (2 - v \sin(u/2)) \cos(u) \\ v \cos(u/2) \end{pmatrix}.$$

Problem 8. Consider the two membership functions $f : \mathbb{R} \to [0, 1]$, $g : \mathbb{R} \to [0, 1]$ in fuzzy logic

$$f(x) = e^{-x^2/2}, \qquad g(x) = 1/(1 + e^{-x}).$$

Write a C++ program that finds the algebraic sum (page 524, Nonlinear Workbook 5th edition).

Problem 9. What is the output of the following C++ program

```
// fcomposition.cpp

#include <iostream>
#include <cmath>
using namespace std;

double f(double x) { return x*x; }
double g(double x) { return 3.0*x-1.0; }

double comp(double (*f)(double),double (*g)(double),double x)
{ f(g(x)); }

int main(void)
{
  double x = 2.5;
  cout << "f(" << x << ") = " << f(x) << endl;
  cout << "g(" << x << ") = " << g(x) << endl;
  cout << comp(f,g,x) << endl;
  return 0;
}
```

Problem 10. Let $\mathbb{N}_0$ be the set of natural numbers including 0. The *Cantor pairing function* $f : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ is defined by

$$f(x, y) = y + \frac{1}{2}(x + y)(x + y + 1).$$

(i) Find the inverse function, i.e. given $s = f(x, y)$ find x and y. Set

$$a := x + y, \quad b := \frac{1}{2}(a^2 + a).$$

(ii) Give a C++ implementation utilizing `Verylong` of SymbolicC++.

Problem 11. Consider the mathematical expression

$$\sin(b) + a * b \underbrace{+}\ c * d + (a - b).$$

Write this mathematical expression as a binary tree with the root indicated by the brace. Then evaluate this binary tree from bottom to top with the values $a = 2$, $b = \pi/2$, $c = 4$, $d = 1$.

voluntary. An alternative to represent a mathematical expression as tree is multiexpression programming (see next page). Use multiexpression programming to evaluate the expression.

Problem 12. (i) Let $r > 0$ (fixed) and $x > 0$. Consider the map

$$f_r(x) = \frac{1}{2}\left(x + \frac{r}{x}\right)$$

or written as difference equation

$$x_{t+1} = \frac{1}{2}\left(x_t + \frac{r}{x_t}\right), \qquad t = 0, 1, 2, \ldots \quad x_0 > 0.$$

Find the fixed points of f_r. Are the fixed points stable?
(ii) Let $r = 3$ and $x_0 = 1$. Find $\lim_{t\to\infty} x_t$. Discuss.

Problem 13. Let A be a given 3×3 matrix over $\mathbb{R}$ with $\det(A) \neq 0$. Is the transformation

$$x'(x, y) = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + a_{33}}$$
$$y'(x, y) = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + a_{33}}$$

invertible? If so find the inverse.

Problem 14. Let N_1, N_2, N_3 be positive integers. Let $n_1 = 0, 1, \ldots, N_1 - 1$, $n_2 = 0, 1, \ldots, N_2 - 1$, $n_3 = 0, 1, \ldots, N_3 - 1$. There are $N_1 \cdot N_2 \cdot N_3$ points and the points (n_1, n_2, n_3) are a subset of $\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$ and can be mapped one-to-one onto a subset of $\mathbb{N}_0$

$$j(n_1, n_2, n_3) = (n_1 N_2 + n_2)N_3 + n_3$$

where $j = 0, 1, \ldots, N_1 \cdot N_2 \cdot N_3 - 1$. Consider first the case $N_1 = N_2 = N_3 = 2$.

Problem 15. Give an implementation of the function

$$\delta^{i_1 i_2 \ldots i_k}_{j_1 j_2 \ldots j_k} := \frac{1}{k!} \sum_{\pi \in S_k} \operatorname{sgn}(\pi) \delta^{i_{\pi(1)}}_{j_1} \delta^{i_{\pi(2)}}_{j_2} \cdots \delta^{i_{\pi(k)}}_{j_k}.$$

For example

$$\delta^{ij}_{k\ell} = \frac{1}{2}(\delta^i_k \delta^j_\ell - \delta^i_\ell \delta^j_k).$$

Problem 16. Let $i_0, i_1, \ldots, i_{k-1} \in \mathbb{N}_0$. Given a vector $(i_0, i_1, \ldots, i_{k-1})$. Give an implementation of the function

$$\zeta_{i_0, i_1, \ldots, i_{k-1}} = \begin{cases} -1 & \text{for} \quad i_0 > i_1 > \cdots > i_{k-1} \\ 1 & \text{for} \quad i_0 < i_1 < \cdots < i_{k-1} \\ 0 & \text{otherwise} \end{cases}$$

Problem 17. Let $n_1, n_2, n_3 \in \mathbb{N}$. Consider the equation

$$\frac{1}{n_1} + \frac{1}{n_2} + \frac{1}{n_3} = \frac{1}{2}.$$

Write a SymbolicC++ program utilizing the class Rational and Verylong to test whether there are solutions for $n_1, n_2, n_3 \in \{1, 2, \ldots, 50\}$.

Problem 18. Let N_1, N_2 be nonnegative integers and $N = N_1 + N_2$. Consider the function

$$f(N_1, N) = \frac{N!}{N_1! N_2!} \equiv \frac{N!}{N_1!(N - N_1)!}$$

Let $N = 10$. Find the minima and maxima of the function $f(N_1, 10)$.

Chapter 4

Number Manipulations

Problem 1. Let p be a *prime number* (base 10) with $p > 2$. Convert the prime number into binary. Then consider this number in base 10. Test whether this number is prime again. For example, consider the prime number 5. Then in binary we have 101. Now 101 (base 10) is a prime number.
(i) Study this question for the first 10 prime numbers.
(ii) Write a C++ program that could do the job.

Voluntary. Study the same question for base 3.

Problem 2. (i) Consider the first 10 prime twins, i.e.

$$(3,5), \quad (5,7), \quad (11,13), \quad (17,19), \quad (29,31),$$

$$(41,43), \quad (59,61), \quad (71,73), \quad (101,103), \quad (107,109)$$

Let (p_1, p_2) be a set of prime twins. Is

$$p_1 p_2 - (p_1 + p_2)$$

a prime numbers? Test for the first ten prime twins.
(ii) Write a C++ program that implements this test.

Problem 3. For *elliptic curve cryptography* we consider elliptic curves that are defined over a finite field, i.e the elliptic group mod p, where p is a prime number. This is defined as follows. Choose two nonnegative integers, a and b, less then p that satisfy

$$(4a^3 + 27b^2) \bmod p \neq 0. \tag{1}$$

Then $E_P(a,b)$ denotes the elliptic group mod p whose elements (x,y) are pairs of non-negative integers less than p satisfying

$$y^2 \equiv x^3 + ax + b \pmod{p}$$

together with the point at infinity. Let $p = 23$ and consider the elliptic curve $y^2 = x^3 + x + 1$.
(i) Show that the condition (1) is satisfied.
(ii) To find the points in $E_P(a,b)$ one proceeds as follows:
1. For each x such that $0 \le x < p$, calculate $x^3 + ax + b \,(\mathrm{mod}\, p)$.
2. For each x from step 1 determine if it has a square root mod p. If not, there are no points in $E_P(a,b)$ with this value of x. If so, there will be two values of y that satisfy the square root operation root operation (unless the value is the single y value of 0). These (x,y) values are points in $E_P(a,b)$. Find the points in $E_{23}(1,1)$.

Problem 4. Apply the *Chinese remainder theorem* to solve the set of equations

$$\begin{aligned} x &\equiv 7 \pmod{8} \\ x &\equiv 2 \pmod{9} \\ x &\equiv -1 \pmod{5}. \end{aligned}$$

Chinese remainder theorem. Suppose that the positive integers $m_1, m_2, \ldots, m_t$ are relatively prime in pairs; that is, $\gcd(m_i, m_j) = 1$ if $i \ne j$, $1 \le i, j \le t$. Let $b_1, b_2, \ldots, b_t$ be arbitrary integers. Then the congruences

$$\begin{aligned} x &\equiv b_1 \pmod{m_1} \\ x &\equiv b_2 \pmod{m_2} \\ &\vdots \\ x &\equiv b_t \pmod{m_t} \end{aligned}$$

have a simultaneous solution. Moreover, the simultaneous solution is unique modulo $m_1 m_2 \cdots m_t$. That is, if y is another solution, then $x \equiv y \pmod{m_1 m_2 \cdots m_t}$.

To find x we write it in the form

$$x = y_1 b_1 + \cdots + y_t b_t$$

where $y_1 \equiv 1 \,(\mathrm{mod}\, m_1$ and $y_1 \equiv 0 \,(\mathrm{mod}\, m_i$ $(2 \le i \le t)$, $y_2 \equiv 1 \,(\mathrm{mod}\, m_2)$ and $y_2 \equiv 0 \,(\mathrm{mod}\, m_i)$ $(i = 1, 3, 4, \ldots, t)$ and similarly for $y_3, \ldots, y_t$. To have $y_1 \equiv 0 \,(\mathrm{mod}\, m_i)$ $(2 \le i \le t)$ we must have $m_2 m_3 \cdots m_t | y_1$, since the m_i are relative prime in pairs. Thus, in general, set

$$m_i' = \frac{m_1 m_2 \cdots m_t}{m_i}.$$

Then $\gcd(m_i, m_i') = 1$ since $m_1, m_2, \ldots, m_t$ are relatively prime in pairs. Thus m_i' has an arithmetic inverse $m_i'^* \bmod m_i$, i.e.

$$m_i'^* m_i' = 1 \,(\mathrm{mod}\, m_i).$$

We set $y_i = m_i'^* m_i'$ and correspondingly set

$$x = m_1'^* m_1' b_1 + \cdots + m_t'^* m_t' b_t.$$

We have $x \equiv b_1 \,(\mathrm{mod}\, m_1)$ since for $2 \leq i \leq t$ we have $m_1 | m_i'$ so that $m_i'^* m_i' b_i \equiv 0 \,(\mathrm{mod}\, m_1)$ for $2 \leq i \leq t$. We also have $m_1'^* m_1' \equiv 1 \,(\mathrm{mod}\, m_1)$ so that $m_1'^* m_1' b_1 \equiv b_1 \,(\mathrm{mod}\, m_1)$. Thus

$$x \equiv b_1 + 0 + \cdots + 0 \equiv b_1 \,(\mathrm{mod}\, m_1).$$

It follows similarly that $x = b_i \,(\mathrm{mod}\, m_i)$ for all i, $1 \leq i \leq t$.

Problem 5. Let p and q be prime numbers with $p, q \geq 3$ and $p \neq q$. Let $n = pq$. Assume that $d, e \in \mathbb{N}$ be two integer numbers with the properties

$$\begin{aligned} 3 < e &< (p-1)(q-1) \\ de &= 1 \bmod (p-1)(q-1) \\ \gcd(e, n) &= \gcd(e, (p-1)(q-1)) = 1. \end{aligned}$$

With these properties we can prove that for $M \in \{\, 0, 1, 2, \ldots, n-1 \,\}$ the definition

$$C := M^e \bmod n$$

yields $M = C^d \bmod n$. Consequently

$$C = C^{ed} \bmod n$$

(i) Let $s \in \mathbb{N}$ such that $C = C^{ed} \bmod n$. Show that

$$M = C^s \bmod n.$$

(ii) Show how the order r of C under modulo n arithmetic can be used to obtain a linear diophantine equation for s.
(iii) Let $p = 3$, $q = 17$, $e = 5$ and $M = 10$. Find s and d.

Problem 6. Let $\mathbb{F}$ be a field ($\mathbb{F} = \mathbb{R}, \mathbb{C}$). Consider polynomials. The *division algorithm* is as follows. Let g be a nonzero polynomial in $\mathbb{F}[x]$. Then every p in $\mathbb{F}[x]$ can be written as

$$p = qg + r$$

where q anbd r are in $\mathbb{F}[x]$, and either $r = 0$ or $\deg(r) < \deg(g)$. Furthermore q and r are unique and can be found by the following algorithm

```
input:   p, g
output: q, r
q := 0; r := p
while(r <> 0) and LT(g) divides LT(r)) do
 q := q + LT(r)/LT(g)
 r := r - (LT(r)/LT(g))
```

where `LT` is the leading term, i.e. the term with the highest degree. Apply the division algorithm to

$$p(x) = x^4 - 1, \qquad g(x) = x^3 - x^2 + x - 1.$$

Write a C++ program using SymbolicC++ that implements the algorithm.

Problem 7. Consider the bijective spiral map on page 79 (problem 19). Can we find an explicit expression for f? Could a polynomial ansatz work

$$f(x,y) = \sum_{i,j=0}^{N} c_{ij} x^i y^j, \qquad (x,y) \in \mathbb{Z} \times \mathbb{Z}.$$

Problem 8. Let $x \in \mathbb{Z}$. Consider the equation

$$5x^2 + 9x + 11 \equiv 0 \pmod{13}.$$

Write a C++ program that checks whether there are solutions in the range $-20 \leq x \leq 20$.

Problem 9. What is the output of the following C++ program

```
// successor.cpp

#include <iostream>
using namespace std;

template<int n> class number
{
  private:
    number<n-1> predecessor;
  public:
    number<n+1> successor(void)
    { return number<n+1>(); }

  ostream& output(ostream& o)
  { o << "{" << predecessor << "}"; return o; }
}; // end class number
```

```
class number<0>
{
  public:
    number<1> successor(void)
    { return number<1>(); }
}; // end class number<0>

template <int n> ostream& operator << (ostream& o,number<n> n)
{  return n.output(o); }

ostream& operator << (ostream& o,number<0> n)
{ o << "{ }"; return o; }

int main(void)
{
  number<0> zero;
  cout << zero  << endl;
  cout << zero.successor() << endl;
  cout << zero.successor().successor() << endl;
  return 0;
}
```

Problem 10. The following program uses the `Verylong` class of SymbolicC++. What is the program doing?

```
// wormell.cpp

#include <iostream>
#include "verylong.h"
using namespace std;

int main(void)
{
 Verylong two("2");
 for(Verylong x("3");x<=Verylong("100");x+=2)
 {
 Verylong p("1");
 Verylong t = x/two;
 for(Verylong a("2");a<=t;a++)
 { for(Verylong b("2");b<=t;b++) { p = p*(x-a*b); } }
 cout << "x = " << x << " " << "p = " << p << endl;
 } // end x for loop
 return 0;
}
```

Problem 11. What is the output of the following program?

```
// cypher.cpp
```

```
#include <iostream>
#include <string>
using namespace std;

int main(void)
{
  string input = "PLEASE CONFIRM RECEIPT 471";
  string output;
  char t = 3;

  for(int i=0;i<input.length();i++)
  {
  if(('a' <= input[i]) && (input[i] <= 'z'))
       output += (input[i] - 'a' + t)%26 + 'a';
  else if(('A' <= input[i]) && (input[i] <= 'Z'))
       output += (input[i] - 'A' + t)%26 + 'A';
  else if(('0' <= input[i]) && (input[i] <= '9'))
       output += (input[i] - '0' + t)%10 + '0';
  else output += input[i];
  }
  cout << "output = " << output << endl;
  return 0;
}
```

Problem 12. Let n be a positive integer. We define the set $\mathbb{Z}_n$ as the set of nonnegative integers less than n

$$\mathbb{Z}_n := \{\, 0, 1, 2, \ldots, (n-1) \,\}.$$

This is referred to as the set of residues modulo n. If we perform modular arithmetic within this set the following properties hold

$$\begin{aligned}
\text{Commutative laws} \quad & (w + x) \bmod n = (x + w) \bmod n \\
& (w * x) \bmod n = (x * w) \bmod n \\
\text{Associative laws} \quad & ((w + x) + y) \bmod n = (w + (x + y)) \bmod n \\
& ((w * x) * y) \bmod n = (x * (w * y)) \bmod n \\
\text{Distributive law} \quad & (w * (x + y)) \bmod n = ((w * x) + (w * y)) \bmod n \\
\text{Identities} \quad & (0 + w) \bmod n = w \bmod n \\
& (1 * w) \bmod n = w \bmod n
\end{aligned}$$

and we have an additive inverse $(-w)$, i.e. for each $w \in \mathbb{Z}_n$ there exists a z such that $w + z = 0 \bmod n$. Note that

```
if (a*b) = (a*c) mod n then b = c mod n if a is relatively
prime to n
```

Write a C++ program using templates that implements modular arithmetic.

Problem 13. The change problem is as follows. Convert some amount of money M into given denominations, using the smallest possible number of coins. This means the input is an amount of money M and an array of d denominations $\mathbf{c} = (c_0, c_1, \ldots, c_{d-1})$ in decreasing order of value, i.e. $c_0 > c_1 > \cdots > c_{d-1}$, where $c_{d-1} = 1$. The output is an array of d integers $i_0, i_1, \ldots, i_{d-1}$ such that

$$c_0 i_0 + c_1 i_1 + \cdots + c_{d-1} i_{d-1} = M$$

and $i_0 + i_1 + \cdots + i_{d-1}$ is a small as possible. The pseudocode is as follows

smallestNumberOfCoins = M

for each $(i_0, \ldots, i_{d-1})$ from $(0, \ldots, 0)$ to $(M/c_0, \ldots, M/c_{d-1})$

  valueOfCoins = $\sum_{k=0}^{d-1} i_k c_k$

    if valueOfCoins = M

      numberOfcoins = $\sum_{k=0}^{d-1} i_k$

      if numberOfCoins < smallestNumberOfCoins

        smallestNumberOfCoins = numberOfCoins

        bestChange = $(i_0, i_1, \ldots, i_{d-1})$

return array bestChange

Write the pseudocode as a C++ program.

Problem 14. Write a C++ program using `Verylong` of SymbolicC++ that finds all integer solutions of

$$2437x + 51329y = 1 \qquad x, y \in \mathbb{Z}$$

in the range $x, y \in [-100000, 100000]$.

Problem 15. Let n be a positive integer to be tested for primality. Let a be an integer less than n. The following algorithm due to Miller and Rabin tests n for primality. Write the algorithm as a C++ program using templates so that also the `Verylong` class of SymbolicC++ can be used.

```
primality(a,n)
 1. let b_k b_{k-1}...b_0 be the binary representation of (n-1)
```

```
 2. d <- 1
 3. for i <- k downto 0
 4.    do x <- d
 5.    d <- (d*d) mod n
 6.    if d = 1 and x neq 1 and x neq n-1 then return true
 7.    if b_i = 1 then d <- (d*a) mod n
 8.   if d neq 1 then return true
 9.   return false
```

Problem 16. We define a mapping from the natural numbers $\mathbb{N}_0$ to sets as

$$0 \to \{\,\}, \qquad n+1 \to \{\,n\,\}.$$

Give a C++ implementation of this representation of natural numbers.

Problem 17. Let m and n be positive integers. We define the map $\mathbb{N}\times\mathbb{N} \to \mathbb{N}$

$$m\blacklozenge n := \frac{\text{the lowest common multiple of } m \text{ and } n}{\text{the highest common factor of } m \text{ and } n}.$$

For example

$$12\blacklozenge 30 = \frac{60}{6} = 10.$$

Is the composition associative, i.e.

$$(m\blacklozenge n)\blacklozenge p = m\blacklozenge(n\blacklozenge p)\,?$$

Write a C++ program using templates (so that `Verylong` of SymbolicC++ can also be used) that implements this composition.

Problem 18. Consider the following arithmetic problem

```
  # # * #
 --------
  # #
+ # #
 ----
  # #
```

where $*$ denotes multiplication and $+$ denotes addition. Each `#` should be one of the digit 1,2,3,4,5,6,7,8,9, where the condition is that each digit occurs only once. Write a C++ program that finds all the solutions.

Problem 19. Show that any positive integer n can be written uniquely as

$$n = 2^j + k$$

where $0 \le k < 2^j$. For example $10 = 2^3 + 2$. Write a C++ program that finds j and k for a given n. Use templates so that also the `Verylong` class of SymbolicC++ can be used.

Problem 20. Give a C++ implementation of modular arithmetic using the `Verylong` class of SymbolicC++. Let $a, b \in \mathbb{Z}$. Recall that

$$a \mod b$$

denotes the remainder obtained by dividing integer b into integer a, which is a number less than b. Thus $a = \mod b$ if $a = b + kn$ for some integer k. This is expressed by saying that a is congruent to b modulo n or that b is the residue of a modulo n. Modular arithmetic is commutative, associative, and distributive

$$\begin{aligned}(a \pm b) \bmod &= ((a \bmod n) \pm (b \bmod n)) \bmod n \\ (a * b) \bmod &= ((a \bmod n) * (b \bmod n)) \bmod n \\ (a * (b + c)) \bmod &= (((ab) \bmod n) + ((ac) \bmod n)) \bmod n\end{aligned}$$

Problem 21. Let n be a positive integer. There are exactly as many irreducible representations of the permutation group S_n (order of S_n is $n!$) as there are *partitions* $\{p_j\}$ of n

$$\sum_{j=1}^{n} p_j = n, \qquad p_1 \ge p_2 \ge \cdots \ge p_n \ge 0.$$

For example for $n = 4$ we have 5 partitions with

$$4000, \quad 3100, \quad 2200, \quad 2110, \quad 1111.$$

The number of partitions is given by $p(1, k)$, where $p(k, n)$ represents the number of partitions of n using only natural numbers at least as large as k. A recursion relation for $p(k, n)$ is given by

$$p(k,n) = \begin{cases} 0 & \text{if} \quad k > n \\ 1 & \text{if} \quad k = n \\ p(k+1,n) + p(k+1,n) + p(k,n-k) & \text{otherwise} \end{cases}$$

(i) Give a C++ implementation of this recursion.
(ii) Give a C++ implementation that finds the partitions for a given n without the trailing 0's.

Problem 22. Consider the sequence (*Ballot numbers*)

$$\begin{aligned}&B_0 = B_1 = 1, \\ &B_L = B_{L-1} + \sum_{\ell=0}^{L-2} B_\ell B_{L-2-\ell}, \qquad L = 2, 3, \ldots.\end{aligned}$$

(i) Give a C++ implementation of this recursion.
(ii) Show that the generating function

$$B(x) = \sum_{L=0}^{\infty} B_L x^L$$

is given by

$$x^2(B(x))^2 - (1-x)B(x) + 1 = 0.$$

Problem 23. A *perfect number* is a natural number with the properties that the sum of the factors gives twice the number, for example

$$2 \cdot 6 = 1 + 2 + 3 + 6$$

so that 6 is a perfect number. A *Mersenne prime* is a Mersenne number $2^n - 1$, where n is chosen such that $2^n - 1$ is prime (for example $n = 5$).
(i) Show that $2^{n-1}(2^n - 1)$ is perfect. For example for $n = 5$ we have 496 with

$$496 = 248 + 124 + 62 + 31 + 16 + 8 + 4 + 2 + 1.$$

(ii) Prove that if $2^{n-1}p$ is perfect for p prime then p is a Mersenne prime and $p = 2^n - 1$.
(iii) Prove that for any even perfect number q there exists $n \in \mathbb{N}$ such that $q = 2^{n-1}(2^n - 1)$.

Problem 24. Consider the one-dimensional map (logistic map) $f : [0,1] \to [0,1]$

$$f(x) = 4x(1-x).$$

A computational analysis using a finite state machine with base 2 arithmetic in fixed point operation provides one-dimensional maps with a lattice of 2^N sites labeled by numbers

$$x = \sum_{j=1}^{N} \frac{\epsilon_j}{2^j}, \qquad \epsilon_j \in \{\, 0, 1 \,\}$$

and N defines the machine's precision. Consider $N = 8$ bits and $x = 1/8$. Calculate the orbit $f(x)$, $f(f(x))$, $f(f(f(x)))$, ... with this precison. Discuss.

Problem 25. Find the solution of the system

$$\begin{aligned} 5x &= 2 \mod 3 \\ 4x &= 7 \mod 9 \\ 2x &= 4 \mod 10. \end{aligned}$$

Problem 26. Let n be a postive integer. A numerical partition of n is a sequence

$$p_1 \geq p_2 \geq \cdots \geq p_k \geq 1$$

such that

$$p_1 + p_2 + \cdots + p_k = n.$$

Each p_j is called a part. For example

$$18 = 7 + 4 + 4 + 1 + 1 + 1$$

is a partition of the integer 18 into 6 parts. The number of partitions of n into k parts is denoted by $p(n, k)$.
(i) Find $p(7, 3)$.
(ii) Show that the recurrence for $p(n, k)$ is given by

$$p(n, k) = p(n - 1, k - 1) + p(n - k, k)$$

with the initial conditions $p(n, 0) = 0$, $p(k, k) = 1$. Obviously we have $p(n, 1) = 1$.
(iii) Write a C++ or Java program that implements this recurrence.
(iv) Every numerical partition of n corresponds to a unique *Ferrer's diagram*. A Ferrer's diagram of a partition is an arrangement of n dots on a square grid, where a part j in the partion is represented by placing p_j dots in a row. Thus we represent each term of the partition by a row of dots, the terms in descending order with the largest at the top. Often it is more convenient to use squares instead of dots. In this case the diagram is called a Young diagram. Draw Ferrer's diagram for the partition of 18 given above. The partition we obtain by reading Ferrer's diagram by columns instead of rows is called the conjugate of the original partition. Find the conjugate of partion of 18 given above.

Problem 27. The *Bernoulli numbers* B_0, B_1, B_2, ... are defined by the series

$$\frac{x}{e^x - 1} = \sum_{j=0}^{\infty} \frac{B_j x^j}{j!}$$

where $B_0 = 1$, $B_1 = -1/2$, $B_2 = 1/6$, $B_3 = 0$, $B_4 = -1/30$. Note that $B_{2j+1} = 0$ for all $j = 1, 2, \ldots$. Give an efficient way to calculate the Bernoulli numbers. Then write a C++ program using the `Verylong` and `Rational` class of SymbolicC++.

Problem 28. In the following arithmetic problem with one multiplication and one sum each digit (except one with the digit 0) has been replaced by an asterisks.

```
* * * x
```

```
    * *
-------
* * * *
* * * 0 +
-------
* * * *
```

Solve the problem when the asterisks can only be one of the prime numbers 2, 3, 5, 7.

Problem 29. Consider the nine numbers

$$111,\ 222,\ 333,\ 444,\ 555,\ 666,\ 777,\ 888,\ 999.$$

Find all the postive integers larger 1 that divide these numbers without remainder.

Problem 30. Let n be a positive integer. Any partition of n into parts $n = p_1 + p_2 + \cdots + p_n$, $0 \le p_1 \le p_2 \le \cdots \le p_n \le n$ can be rearranged in $2 \times r$ matrix

$$\begin{pmatrix} a_1 & a_2 & \cdots & a_r \\ b_1 & b_2 & \cdots & b_r \end{pmatrix}$$

called the *Frobenius symbol*, such that

$$0 \le a_1 < a_2 < \cdots < a_r, \quad 0 \le b_1 < b_2 < \cdots < b_r, \quad n = r + \sum_{j=1}^{r} a_j + \sum_{j=1}^{r} b_j.$$

Each partition has a unique representation as a Frobenius symbol.
(i) Find the Frobenius symbol for $n = 20$.
(ii) Write a C++ program that finds the Frobenius symbol for a given n.

Problem 31. Show that (continued-fraction representation)

$$\frac{1}{\sqrt{3}} = [1, 2, 1, 2, 1, 2 \ldots].$$

Problem 32. The Möbius arithmetic functions $\mu : \mathbb{N} \mapsto \{0, \pm 1\}$ is defined by

$$\mu(n) := \begin{cases} 1 & \text{if} \quad n = 1 \\ (-1)^m & \text{if} \quad n \text{ is the product of } m \text{ distinct primes} \\ 0 & \text{otherwise} \end{cases}$$

Give a C++ implementation using `Verylong` of SymbolicC++ and `BigInteger` using Java. For $m \in \mathbb{N}$ one has

$$\sum_{n|m} \mu(n) = \delta_{m,1}$$

where δ denotes the Kronecker delta.

Problem 33. Let x, y be integers. Assume they satisfy the equation

$$x^2 - 8y^2 = 17.$$

(i) Show that $(x_0, y_0) = (5, 1)$ is a solution. Show that $(x_0, y_0) = (7, 2)$ is a solution.
(ii) Let $n = 0, 1, 2, \ldots$. Show that if (x_n, y_n) satisfy the equation, then

$$x_{n+1} = 3x_n + 8y_n, \qquad y_{n+1} = x_n + 3y_n$$

also satisfy the equation.

Problem 34. *Nobles* are irrationals with continued fraction representations of the form

$$\omega = a_0 + 1/(a_1 + 1/(a_2 + \cdots + 1/(1 + 1/(1 + \cdots$$

which eventually have elements all equal to one. Find the number with all a_j's equal to one.

Problem 35. Let p and q be prime with $p \leq q$. Solve for p and q such that $p + q$ is prime.

Problem 36. Let $n = 0, 1, 2, \ldots$. The *Fermat numbers* F_n are given by

$$F_n = 2^{(2^n)} + 1$$

(i) Show that the Fermat numerbers satisfy the recurrence relation

$$F_{n+1} = (F_n - 1)^2 + 1$$

with $F_0 = 3$.
(ii) Calculate the first ten Fermat numbers using this recurrence relation and the `Verylong` class of SymbolicC++.
(iii) Calculate the first ten Fermat numbers using this recurrence relation and the `BigInteger` class of Java.

Problem 37. Consider a palindrome consisting of digits. A *palindrome* is a string of characters that reads the same from left to right or from right to left. Show that if $n \in \mathbb{N}$ is a palindrome with an even number of digits, then n is a multiple of 11. For example $3223 : 11 = 293$. Write a C++ program utilizing the string class to find out whether a string with an even number of digits is a palindrome.

Problem 38. Consider the 6 digit number 142857 with the digits 1, 4, 2, 8, 5, 7. Multiply the number by 2,3,4,5,6. Discuss. What is the connection with 6×6 permutation matrices? Write a C++ program that can do the job.

Problem 39. Can one find two positive integers m and n such that

$$m \cdot n \cdot (m+n) = 29400.$$

Problem 40. Let $n \in \mathbb{N}$. Show that $7^n - 2^n$ is divisible by 5. Is $7^n - 3^n$ divisible by 4?

Problem 41. Let $n \in \mathbb{N}$ and $x, y \in \mathbb{R}$. Show that $x^n - y^n$ is divisible by $x - y$.

Problem 42. We know that there is an positive integer n such that

$$n^5 = 5277319168$$

Find n using that $0^5 = 0$, $1^5 = 1$, $2^5 = ...2$, $3^5 = ...3$, $4^5 = ...4$, $5^5 = ...5$, $6^5 = ...6$, $6^5 = ...6$, $7^5 = ...7$, $8^5 = ...8$, $9^5 = ...9$, i.e. the last digit of the power 5 of the numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 is the number themselve.

Chapter 5

Combinatorical Problems

Problem 1. Consider n distinct objects. Then there are $n!$ permutations. Without loss of generality we consider the integer numbers $1, 2, \ldots, n$. Write a C++ program that uses recursion to find all permutations of these numbers.

Problem 2. Let $X := \{\, x_0 = 0,\, x_1, x_2, \ldots, x_{n-1} \,\}$ be a set of n points on the real axis with $x_j < x_{j+1}$ for $j = 0, 1, \ldots, n-2$. Let ΔX denote the sequence of all

$$\binom{n}{2} \equiv \frac{n!}{(n-2)!2!}$$

pairwise distance between points in X with the ordering $\Delta x_j \leq \Delta x_{j+1}$ for $j = 0, 1, \ldots, \binom{n}{2} - 1$. For example, let $X = \{\, 0, 2, 4, 7, 10\}$. Then

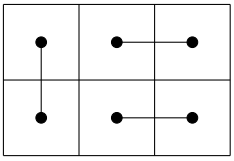
$$\Delta X = \{2, 2, 3, 3, 4, 5, 6, 7, 8, 10\}.$$

Given X write a C++ program that generates ΔX. Can one reconstruct X from ΔX? For the given example look at

```
    0 2 4 7 10
 0    2 4 7 10
 2      2 5  8
 4        3  6
 7           3
10
```

Problem 3. Let n be an integer $n \geq 1$. Consider a $2 \times n$ rectangular array of boxes (a lattice space) and r dumbbell-shaped objects $\bullet \leftrightarrow \bullet$. Let $A(r, n)$ be the number of ways in which the r dumbbells may be placed in the $2 \times n$

rectangular array such that the two ends of each dumbbel are in two horizontally or vertically adjacent boxes and no two *dumbells* have ends which share a box. Obviously, $A(r,n) = 0$ if $r > n$. For example a configuration with $n = 3$ and $r = 3$ is



(i) Find $A(1,1)$, $A(1,2)$, $A(2,2)$.
(ii) A recurrence is given by

$$A(r,n) = A(r,n-1) + 2A(r-1,n-1) + A(r-1,n-2) - A(r-3,n-3)$$

with the initial conditions $A(1,1)$, $A(1,2)$, $A(2,2)$ and $A(0,n) = 1$. Use this implementation to find $A(1,5)$, $A(2,5)$, $A(3,5)$, $A(4,5)$, $A(5,5)$.
(iii) Give a C++ implementation of this recurrence together with the initial conditions.

Problem 4. The rank of an element in a sequence (one-dimensional array) of numbers is the number of smaller elements in the sequence plus the number of equal elements that appear to its left. For example, if the sequence is given as the one-dimensional array $a = [4,3,9,3,7]$, then the ranks are $r = [2,0,4,1,3]$. Write a C++ program with a function `void rank(T* a,int n,int* r)` that computes the ranks of the elements of the array $a[0:n-1]$. Once the elements have been ranked using the function `rank()` write a function `rearrange()` that rearrange them in nondecreasing order so that $a[0] \le a[1] \le \cdots \le a[n-1]$ by moving elements to positions corresponding to their ranks.

Problem 5. (i) Let $\mathbb{N}_0$ be the set of all positive integers including 0. Let $i,j,k \in \mathbb{N}_0$. Find all solutions of $i + j + k = 3$. Write down the solution in lexigographical order.
(ii) Write a C++ program that finds all solutions of $i + j + k = n$ for a given $n \in \mathbb{N}_0$.

Problem 6. Suppose we list all the $2^n - 1$ nonempty subsets of the set of numbers $\{1,2,\ldots,n\}$. Then, for each subset, we write down the product of its elements. Finally, we add these $2^n - 1$ numbers to obtain the number s_n. Obviously $s_1 = 1$. For $n = 2$ we have $\{1\}$, $\{2\}$, $\{1,2\}$. Thus $s_2 = 5 = 1 + 2 + 2$. For $n = 3$ the seven products one obtains are 1, 2, 3, 1×2, 1×3, 2×3 and $1 \times 2 \times 3$. Thus $s_3 = 23$.
(i) Find s_4.
(ii) Find a recursion $s_{n+1} = f(n, s_n)$ for s_{n+1} with $s_{=}1$.

(iii) Write a C++ program that implements this recurrence relation with the initial value $s_1 = 1$.

Problem 7. How many arrangements of a, a, a, b, b, b, c, c, c are there so that no 2 letters of the same type appear consecutively? For example *abcabcabc* would be such an arrangement. Could a tree structure be used to find the solution? Write a C++ program that finds all sequences.

Problem 8. A coin is tossed eight times in a row.
(i) What is the probability of getting exactly four heads in a row?
(ii) What is the probability of getting at least four heads in a row?

Problem 9. Show that the number of ways in which n different objects can be arranged in a a ring, if only relative order matters, is $(n-1)!$. First give an example with 3 objects.

Problem 10. (i) How many n-digit ternary sequences (using only 0,1,2) are there with k 1's?
(ii) Find the sequences for $n = 3$ and $k = 2$.
(iii) Write a C++ program that finds these sequences for given n and k in lexicographical order.

Problem 11. Suppose that we list all the $2^n - 1$ nonempty subsets of the set of numbers $\{1, 2, \ldots, n\}$. Then, for each subset, we write down the product of its elements. Finally, we add these $2^n - 1$ numbers to obtain the number s_n. Obviously, $s_1 = 1$. For $n = 3$ the seven products we obtain are $1, 2, 3, 1 \times 2, 1 \times 3, 2 \times 3$ and $1 \times 2 \times 3$. Thus $s_3 = 1+2+3+2+3+6+6 = 23$. Find a recurrence relation for s_n. Write a C++ program that implements this recurrence relation with the initial value $s_1 = 1$.

Problem 12. How many arrangements of $a, a, a, b, b, b, c, c, c$ are there so that no 2 letters of the same type appear consecutively? For example, *abcabcabc* would be such an arrangement.

Problem 13. Let n be a positive integer. Use the inclusion-exclusion principle to prove that

$$n = \sum_{k=1}^{n} (-1)^{k-1} k \binom{n}{k} 2^{n-k}.$$

Problem 14. Show that the number of ways of writing the positive integer n as a sum of positive integers, where the order of the summands is significant, is 2^{n-1} for $n \geq 1$. For example, for $n = 3$ we have $3 = 3$, $3 = 2 + 1$, $3 = 1 + 2$,

$3 = 1 + 1 + 1$.

Problem 15. The number x_n of steps required to solve the *Chinese rings puzzle* with n rings satisfies $x_1 = 1$ and the recurrence relation

$$x_{n+1} = \begin{cases} 2x_n & n \text{ odd} \\ 2x_n + 1 & n \text{ even} \end{cases}$$

where $n = 1, 2, \ldots$.
(i) Prove that $x_{n+2} = x_{n+1} + 2x_n + 1$.
(ii) Find a formula for x_n.

Problem 16. A derangement (or fixed-point-free permutation) of $\{ 1, 2, \ldots, n \}$ is a permutation f such that $f(j) \neq j$ for all $j = 1, 2, \ldots, n$. What is the number of derangements of n objects?

Problem 17. A numerical partition of a positive integer n is a sequence

$$p_1 \geq p_2 \geq \cdots \geq p_k \geq 1$$

such that

$$p_1 + p_2 + \cdots + p_k = n.$$

Each p_j is called a part. For example, $18 = 7+4+4+1+1+1$ is a partition of 18 into 6 parts. The number of partitions of n into k parts is denoted by $p(n, k)$.
(i) Find $p(7, 3)$.
(ii) Show that the recurrence for $p(n, k)$ is given by

$$p(n, k) = p(n-1, k-1) + p(n-k, k)$$

with the initial conditions $p(n, 0) = 0$, $p(k, k) = 1$. Obviously, we have $p(n, 1) = 1$.
(iii) Write a C++ program that implements this recurrence..
(iv) Every numerical partition of a positive integer n corresponds to a unique *Ferrer's diagram.* A Ferrer's diagram of a partition is an arrangement of n dots on a square grid, where a part j in the partition is represented by placing p_j dots in a row. This means we represent each term of the partition by a row of dots, the terms in descending order with the largest at the top. Sometimes it is more convenient to use squares instead of dots (in this case the diagram is called a Young diagram). Draw the Ferrer's diagram for the partition $18 = 7+4+4+1+1+1$. The partition we obtain by reading the Ferrer's diagram by columns instead of rows is called the conjugate of the original partition. Find the conjugate of the partition $18 = 7 + 4 + 4 + 1 + 1 + 1$.

Problem 18. The *Bell numbers* count (starting from 0) the ways that n distinguishable objects can be grouped into sets if no set can be empty. Thus

the Bell numbers are given by the sequence

$$\{\,1, 1, 2, 5, 15, 52, 203, 877, 4140, \ldots\,\}.$$

For example the numbers 1, 2, 3 can be grouped into sets so that

1) 1, 2, 3 are in three separate sets: $\{1\}$, $\{2\}$, $\{3\}$
2) 1 and 2 are together and 3 is separate: $\{1, 2\}$, $\{3\}$
3) 1 and 3 together and 2 separate: $\{1, 3\}$, $\{2\}$
4) 2 and 3 together and 1 separate: $\{2, 3\}$, $\{1\}$
5) 1, 2, 3 are all together in a single set: $\{1, 2, 3\}$

Hence for $n = 3$ there are five partitions and thus the third Bell number is 5.
(i) Let P_n denote the n^{th} Bell number, i.e. the number of all partitions of n objects. Then we have

$$P_n = \frac{1}{e}\sum_{k=0}^{\infty}\frac{k^n}{k!}.$$

(i) Find a recurrence relation for P_n.
(ii) Write a C++ program that implements this recursion. Apply `Verylong` of SymbolicC++.
(iii) Find the MacLaurin expansion (expansion around 0) of $\exp(\exp(x))$ and establish a connection with the Bell numbers.

Problem 19. Let N be a positive integer. Consider the set of numbers

$$S = \{\,0,\, 1,\, 2, \ldots, N\,\}$$

How many pairs (m, n) $(m, n \in S)$ can be formed with the condition that $m < n$.

Problem 20. Let n be the number of discrete symbols s_1, s_2, ...,s_n that can be used. Let m be the length of the message string. Find the number M of messages. Then consider the special case $n = m = 2$.

Problem 21. Consider a bitstring of length m which has exactly m_1 ones and m_2 zeros $(m_1 + m_2 = m)$.
(i) Find the number of different possible bitstrings.
(ii) Consider the special case $m = 4$, $m_1 = m_2 = 2$. Write down the bitstrings in lexicographical order.

Problem 22. Consider the n-dimensional unit cube in $\mathbb{R}^n$. How many k-dimensional surfaces $(1 \leq k < n)$ does the n-dimensional unit cube has?

Problem 23. Let n_1, n_2, n_3 be nonnegative integers. Let $n = n_1 + n_2 + n_3$. To what combinatorical problem can

$$\frac{n!}{n_1!n_2!n_3!}$$

be associated?

Problem 24. A dice is thrown twice. The first throw determines the tens digit and the second throw the ones digit of the two-digit number. Find the probability that this two-digit number is a perfect square.

Problem 25. How many different numbers of 7 digits can be formed with the digits 1122334 ?

Problem 26. Let $i, j, k \in \mathbb{N}_0$. Find all solutions of

$$i + j + k = 3.$$

Give the solution in lexicographical order.

Problem 27. Suppose three fair coins are flipped. Let X be the event that they same face. Let Y be the event that there is at most one head. Show that X and Y are independent.

Problem 28. The *Stirling number* of the second kind $S(n, k)$ is the number of partitions of a set with n elements into k classes. Let $b^\dagger$, b be Bose creation and annihilation operators with the commutation relations

$$[b, b^\dagger] = bb^\dagger - b^\dagger b = I$$

where I is the identity operator. Then $S(n, k)$ can be defined by

$$(b^\dagger b)^n = \sum_{k=1}^{n} S(n, k)(b^\dagger)^k b^k.$$

Let $n = 3$. Use this definition to find $S(3, 1)$, $S(3, 2)$, $S(3, 3)$.

Problem 29. Let $n \geq 1$. Let a_1, a_2, ...,a_n be real numbers. How many terms are there in the sum

$$\sum \sum_{1 \leq j_1 < j_2 < j_3 \leq n} \sum a_{j_1} a_{j_2} a_{j_3}.$$

Problem 30. Let $n \geq 1$. Let $c_1^\dagger$, $c_2^\dagger$, ..., $c_n^\dagger$ be spin-less Fermi creation operators and c_1, c_2, ..., c_n be spin-less Fermi annihilation operators. Thus

$$[c_j^\dagger, c_k]_+ = \delta_{jk} I, \quad [c_j, c_k]_+ = 0, \quad [c_j^\dagger, c_k^\dagger]_+ = 0$$

where 0 is the zero operator, $[\,,\,]_+$ denotes the anti-commutator and $j,k = 1,\ldots,n$. Then $(c_j)^2 = 0$ and $(c_j^\dagger)^2 = 0$. Let $|0\rangle$ be the vacuum state with $c_j|0\rangle = 0|0\rangle$ and $j = 1,\ldots,n$. Then for $n = 1$ we can form the two-dimensional basis $|0\rangle$, $c_1^\dagger|0\rangle$. For $n = 2$ we can form the four dimensional basis

$$|0\rangle, \quad c_1^\dagger|0\rangle, \quad c_2^\dagger|0\rangle, \quad c_2^\dagger c_1^\dagger|0\rangle.$$

(i) Find the basis for $n = 3$.
(ii) Find the basis for $n = 4$.
(iii) Extend to arbitrary n.

Chapter 6

Matrix Calculus

Problem 1. Let A be an $n \times n$ matrix over $\mathbb{C}$. We define $\sin(A)$ as

$$\sin(A) := \sum_{j=0}^{\infty} \frac{(-1)^j}{(2j+1)!} A^{2j+1}.$$

Consider the 2×2 matrix ($x, y \in \mathbb{R}$)

$$B = \begin{pmatrix} x & y \\ 0 & x \end{pmatrix}.$$

Calculate $\sin(B)$ efficiently. Decompose the matrix B as $B = C + D$, where

$$C = \begin{pmatrix} x & 0 \\ 0 & x \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & y \\ 0 & 0 \end{pmatrix}.$$

Problem 2. Let A be a square matrix over $\mathbb{R}$. We define the $n-m$ approximant of e^A as

$$f_{n,m}(A) := \left(\sum_{k=0}^{m} \frac{1}{k!} \left(\frac{A}{n} \right)^k \right)^n.$$

We have the error estimation

$$\|e^A - f_{n,m}(A)\| \leq \frac{1}{n^m (m+1)!} \|A\|^{m+1} e^{\|A\|}$$

and $f_{n,m}(A)$ converges to e^A

$$\lim_{n \to \infty} f_{n,m}(A) = \lim_{m \to \infty} f_{n,m}(A) = e^A.$$

Consider

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

(i) Calculate e^A.
(ii) Calculate $f_{2,2}(A)$, the norm of A and the error estimation. The norm is given by

$$\|A\| := \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|$$

where $\mathbf{x} \in \mathbb{R}^2$ and $\|A\mathbf{x}\|$ denotes the Euclidean norm.

Problem 3. Consider an 8×8 matrix numbered as follows

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 \\ 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 \\ 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 \\ 40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 \\ 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 \\ 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 \end{pmatrix}.$$

Let (j, k) be the matrix elements with $j, k = 0, 1, \ldots, 7$. Write a C++ program that writes the matrix elements given as floating point numbers into a vector with 64 elements using the numbering given above. Use the `vector` class of the Standard Template Library for the vector and the matrix.

Problem 4. An $n \times n$ matrix T is called a *Toeplitz matrix* if it satisfies the relation $T(k, j) = T(k-1, j-1)$, for $2 \leq j, k \leq n$. In other words, the entries on each diagonal of T are all equal. Hence, such a matrix is determined by the $2n-1$ entries appearing in the first row and first column. We denote these entries by $t_0, t_1, \ldots, t_{2n-2}$ such that

$$T = \begin{pmatrix} t_{n-1} & t_{n-2} & \cdots & & t_2 & t_1 & t_0 \\ t_n & t_{n-1} & t_{n-2} & \cdots & & t_2 & t_1 \\ t_{n+1} & t_n & t_{n-1} & t_{n-2} & \cdots & & t_2 \\ \vdots & & & \ddots & & & \vdots \\ \vdots & & & & \ddots & & \vdots \\ t_{2n-3} & t_{2n-2} & \cdots & & & t_{n-1} & t_{n-2} \\ t_{2n-2} & t_{2n-3} & \cdots & & t_{n+1} & t_n & t_{n-1} \end{pmatrix}.$$

We say that the vector $\mathbf{t} = (t_0, t_1, \ldots, t_{2n-2})$ defines the Toeplitz matrix T. Thus the 4×4 Toeplitz matrix defined by the vector $\mathbf{t} = (t_0, t_1, t_2, t_3, t_4, t_5, t_6)$

is

$$T = \begin{pmatrix} t_3 & t_2 & t_1 & t_0 \\ t_4 & t_3 & t_2 & t_1 \\ t_5 & t_4 & t_3 & t_2 \\ t_6 & t_5 & t_4 & t_3 \end{pmatrix}.$$

Write a C++ program that generates the Toeplitz matrix T from a given vector $\mathbf{t}$. Vice versa given a Toeplitz matrix T find the vector $\mathbf{t}$.

Problem 5. Let $A_1, A_2, \ldots, A_p$ be square matrices of the same size. Let

$$f_{n,1}(A_1, A_2, \ldots, A_p) := (e^{A_1/n} e^{A_2/n} \cdots e^{A_p/n})^n.$$

Then

$$\| \exp(\sum_{j=1}^{p} A_j) - f_{n,1}(A_1, A_2, \ldots, A_p)\| \leq \frac{2}{n} \left(\sum_{j=1}^{p} \|A_j\| \right)^2 \exp\left(\frac{n+2}{n} \sum_{j=1}^{p} \|A_j\| \right)$$

and

$$\lim_{n\to\infty} f_{n,1}(A_1, A_2, \ldots, A_p) = \exp\left(\sum_{j=1}^{p} A_j \right).$$

For $p = 2$ we obtain

$$e^{A_1+A_2} = \lim_{n\to\infty} (e^{A_1/n} e^{A_2/n})^n. \tag{1}$$

Let

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \qquad A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Then $A = A_1 + A_2$. Note that $[A_1, A_2] \neq 0$.
(i) Calculate e^A by diagonalizing A.
(ii) Calculate e^A using equation (1).
(iii) Calculate $f_{2,1}(A_1, A_2)$ and the error estimation.

Problem 6. The discrete Fourier transform over n points can be written in matrix form

$$F_n := \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & w & w^2 & \ldots & w^{n-1} \\ 1 & w^2 & w^4 & \ldots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \ldots & w^{(n-1)^2} \end{pmatrix}$$

where $w := e^{2\pi i/n}$ is the n-th root of unity. We obtain the discrete Fourier transform from

$$(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n)^T = F_n (x_1, x_2, \ldots, x_n)^T.$$

Apply F_4 to the data $(1,0,0,1)^T$ and $(0,1,1,0)^T$ and interpret the results to find the underlying periodicity.

Problem 7. (i) The discrete Fourier transform over n points can be written in matrix form

$$F_n := \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & \dots & w^{n-1} \\ 1 & w^2 & w^4 & \dots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \dots & w^{(n-1)^2} \end{pmatrix}$$

where $w = e^{2\pi i/n}$ is the n-th root of unity. Show that the matrix $\frac{1}{\sqrt{n}}F_n$ is unitary.
(ii) Use trigonometric interpolation to find

$$f(x) = c_0 + c_1 e^{ix} + c_2 e^{2ix} + c_3 e^{3ix}$$

which interpolates the points

$$x_0 = 0, \quad y_0 = 0, \qquad x_1 = \frac{\pi}{2}, \quad y_1 = 1,$$

$$x_2 = \pi, \quad y_2 = 1, \qquad x_3 = \frac{3\pi}{2}, \quad y_3 = 0$$

i.e. solve the equation $F_4(c_0, c_1, c_2, c_3)^T = (y_0, y_1, y_2, y_3)^T$.

Problem 8. A *1-inverse* of the $m \times n$ matrix A is an $n \times m$ matrix A^- such that $AA^-A = A$.

(i) Suppose that $m = n$ and that A^{-1} exists, find A^-.
(ii) Is the 1-inverse unique? Prove or disprove.
(iii) The *Moore-Penrose pseudoinverse* of the $m \times n$ matrix A is the 1-inverse A^- of A which additionally satisfies

$$A^-AA^- = A^- \qquad (AA^-)^* = AA^- \qquad (A^-A)^* = A^-A.$$

Let $A = U\Sigma V^*$ be the singular value decomposition of A. Show that $A^- = V\Sigma^- U^*$ is a Moore-Penrose pseudoinverse of A, where

$$(\Sigma^-)_{jk} = \begin{cases} \frac{1}{(\Sigma)_{kj}} & (\Sigma)_{kj} \neq 0 \\ 0 & (\Sigma)_{kj} = 0 \end{cases}$$

Hint: First show that Σ^- is the Moore-Penrose pseudoinverse of Σ. Find the Moore-Penrose pseudoinverse of

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Problem 9. Find the eigenvectors and generalized eigenvectors of

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}.$$

Problem 10. Let

$$A = \begin{pmatrix} 0 & i\pi & 0 \\ 0 & 0 & 0 \\ 0 & -i\pi & 0 \end{pmatrix}.$$

Calculate $\exp(A)$, $\sinh(A)$, $\cosh(A)$, $\sin(A)$, $\cos(A)$ efficiently.

Problem 11. Find the Cosine-Sine decomposition of

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix}.$$

Problem 12. Solve the system of linear equations

$$\begin{pmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

using the Jacobi method. Start from the initial guess $x_0 = y_0 = z_0 = 0$.

Problem 13. How would one store a matrix using a linked list?

Problem 14. Given an $n \times n$ matrix over $\mathbb{C}$. How would we efficiently test whether the matrix is unitary? Apply your approach to

$$U = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ i & i & -i \\ i & -i & -i \end{pmatrix}.$$

Problem 15. The sum $1^2 + 2^2 + 3^2 + \cdots + n^2$ can be written as

$$1^2 + 2^2 + 3^2 + \cdots + n^2 = an^3 + bn^2 + cn$$

where the unknown coefficients a, b, c can be determined from a system of linear equations obtained from $n = 1$, $n = 2$, $n = 3$. Find this system of linear

equations and write a C++ program using *Gauss elimination* that finds the solution.

Problem 16. The sum $1^3 + 2^3 + 3^3 + \cdots + n^3$ can be written as

$$1^3 + 2^3 + 3^3 + \cdots + n^3 = an^4 + bn^3 + cn^2$$

where the unknown coefficients a, b, c can be determined from a system of linear equations obtained from $n = 1$, $n = 2$, $n = 3$. Find this system of linear equations and write a C++ program using *Gauss elimination* that finds the solution.

Problem 17. Let A, B be $m \times n$ matrices over $\mathbb{C}$. The *Hadamard product* $A \bullet B$ is defined as the $m \times n$ matrix (entrywise multiplication)

$$A \bullet B = (a_{jk} b_{jk}).$$

Write a C++ program using `vector<vector<complex> >` that implements the Hadamard product.

Problem 18. Consider the set, M, containing all matrices of $M_2(\mathbb{R})$ of the form

$$\begin{pmatrix} a & -b \\ b & a \end{pmatrix}$$

where $a, b \in \mathbb{R}$. Mapping this matrix onto $a + i * b$ we obtain a field isomorphism between M and $\mathbb{C}$. Write a C++ program using `complex<T>` and `vector<vector<T> >` and the function

```
void isomorphism(complex<T>& z,vector<vector<T> >& m)
```

that maps a complex number to the corresponding 2×2 matrix.

Problem 19. Let A be an $n \times n$ symmetric matrix over $\mathbb{R}$. Write a C++ program that uses *Givens transform* to cast the matrix into *tridiagonal form*.

Problem 20. Given an $n \times n$ tridiagonal matrix over $\mathbb{R}$ with $n \geq 3$. Write a C++ program that finds the characteristic polynomial.

Problem 21. Let A be an $n \times n$ matrix over $\mathbb{R}$. Let $\mathbf{u}$ be a nonzero column vector in $\mathbb{R}^n$. Computing the $n \times n$ matrix

$$\left(I_n - \frac{2\mathbf{u}\mathbf{u}^T}{\mathbf{u}^T\mathbf{u}} \right) A$$

can be done as follows.

Step 1. Compute the number $\beta = 2/(\mathbf{u}^T\mathbf{u})$.

Step 2. for $j = 1, 2, \ldots, n$ do
$\alpha = u_1 a_{1j} + u_2 a_{2j} + \cdots + u_n a_{nj}$
$\alpha = \beta \cdot \alpha$
for $i = 1, 2, \ldots, n$ do $a_{ij} = a_{ij} - \alpha u_i$

Write a C++ program that implements this algorithm.

Problem 22. Consider the two polynomials

$$p_1(x) = a_0 + a_1 x + \cdots + a_n x^n, \qquad p_2(x) = b_0 + b_1 x + \cdots + b_m x^m$$

where $n = \deg(p_1)$ and $m = \deg(p_2)$. Assume that $n > m$. Let $r(x) = p_2(x)/p_1(x)$. We expand $r(x)$ in powers of $1/x$, i.e.

$$r(x) = \frac{c_1}{x} + \frac{c_2}{x^2} + \cdots$$

From the coefficients c_1, c_2, …, c_{2n-1} we can form an $n \times n$ *Hankel matrix*

$$H_n = \begin{pmatrix} c_1 & c_2 & \cdots & c_n \\ c_2 & c_3 & \cdots & c_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n+1} & \cdots & c_{2n-1} \end{pmatrix}.$$

The determinant of this matrix is proportional to the *resultant* of the two polynomials. If the resultant vanishes, then the two polynomials have a non-trivial greates common divisor. Implement this algorithm in SymbolicC++ where $a_j, b_j \in \mathbb{Q}$ and apply it to the polynomials

$$p_1(x) = x^3 + 6x^2 + 11x + 6, \qquad p_2(x) = x^2 + 4x + 3.$$

Problem 23. Given an $m \times n$ matrix $A = (a_{jk})$. We define a norm as

$$\|A\| := \max_{1 \le j \le m} \left(\sum_{k=1}^{n} |a_{jk}| \right).$$

Give a C++ implementation using templates.

Problem 24. Write a C++ program that transposes a square matrix in-place.

Problem 25. Consider a binary $n \times n$ matrix, where we count the entries from 0. We have $b_{00} = 1$ and $b_{n-1n-1} = 1$. The other 0-1 entries are generated

randomly. An ant at entry $(0,0)$ can only move to the right or down (not diagonal) when this entry contains a 1. Write a C++ program that check whether the ant could reach the entry $(n-1,n-1)$. For example, consider the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

For this case one path

$$(0,0) \to (1,0) \to (1,1) \to (1,2) \to (2,2) \to (3,2) \to (3,3) \to (4,3) \to (4,4)$$

would be possible. Note that the ant could also get stuck at $(2,0)$.

Problem 26. Let $z \in \mathbb{C}$ and A be an $n \times n$ matrix over $\mathbb{C}$ with $A^2 = I_n$. Calculate $\exp(zA)$.

Problem 27. Let $z \in \mathbb{C}$ and A be an $n \times n$ matrix over $\mathbb{C}$ with $A^2 = A$. Calculate $\exp(zA)$.

Problem 28. Apply the Leverrier method to find the determinant of the matrix

$$A(\epsilon) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & \epsilon \end{pmatrix}$$

where $\epsilon \in \mathbb{R}$. What is the condition on ϵ such that the inverse of $A(\epsilon)$ exists?

Problem 29. Let

$$A = \begin{pmatrix} 1/4 & 1/2 \\ 1/2 & 1/4 \end{pmatrix}.$$

Let

$$\rho(A) = \max_{1 \le j \le 2} |\lambda_j|$$

where λ_j are the eigenvalues of A.
(i) Check that $\rho(A) < 1$.
(ii) If $\rho(A) < 1$, then

$$(I_2 - A)^{-1} = I_2 + A + A^2 + \cdots .$$

Calculate $(I_2 - A)^{-1}$.
(iii) Calculate

$$(I_2 - A)(I_2 + A + A^2 + \cdots + A^k).$$

Problem 30. Let A be an $n \times n$ matrix over $\mathbb{R}$. Consider the system of linear equations

$$A\mathbf{x} = \mathbf{b}$$

or

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \qquad i = 1, 2, \ldots, n.$$

We assume that A is invertible. Let $A = C - R$. This is called a splitting of the matrix A and R is the defect matrix of the splitting. Consider the iteration

$$C\mathbf{x}(t+1) = R\mathbf{x}(t) + \mathbf{b}, \quad t = 0, 1, \ldots$$

with a given $\mathbf{x}(0)$.
Let

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -2 & 4 \end{pmatrix}, \quad C = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 3 \\ 2 \\ 2 \end{pmatrix}, \quad \mathbf{x}(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Perform the iteration. Does $\mathbf{x}(k)$ $(k = 0, 1, \ldots)$ converge to the solution of $A\mathbf{x} = \mathbf{b}$. Write a C++ program that implements this iteration.

Problem 31. Let A be an $n \times n$ matrix over $\mathbb{C}$. Then any eigenvalue of A satisfies the inequality

$$|\lambda| \leq \max_{1 \leq j \leq n} \sum_{k=1}^{n} |a_{jk}|.$$

Write a C++ program that calculates the right-hand side of the inequality for a given matrix. Apply the complex class of STL. Apply it to the matrix

$$A = \begin{pmatrix} i & 0 & 0 & i \\ 0 & 2i & 2i & 0 \\ 0 & 3i & 3i & 0 \\ 4i & 0 & 0 & 4i \end{pmatrix}.$$

Problem 32. We count the entries of the $n \times n$ matrix from $(0, 0)$ to $(n - 1, n - 1)$. Let $A = (a_{jk})$ be a real $n \times n$ matrix $(j, k = 0, 1, \ldots, n - 1)$. The *permanent* of A is defined to be the real number

$$\operatorname{perm}(A) := \sum_{\sigma \in S_n} \left(\prod_{j \in [n]} A_{j,\sigma(j)} \right)$$

where the summation is over all $n!$ permutations of the set $[n] := \{0, 1, \ldots, n-1\}$. Give an implementation with SymbolicC++ to find the permanent of a given

matrix A.

Note that the definition of the determinant for the matrix A is

$$\det(A) := \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \left(\prod_{j \in [n]} A_{j,\sigma(j)} \right)$$

where the summation is over all $n!$ permutations of the set $[n] := \{0, 1, \ldots, n-1\}$ and $\operatorname{sgn}(\sigma)$ equals $+1$ if σ is an even permutation and equals -1 if σ is an odd permutation.

Problem 33. We count the entries of the $n \times n$ matrix F from $(0,0)$ to $(n-1, n-1)$. We define

$$\omega_n = \exp(i2\pi/n).$$

Now the $n \times n$ matrix F is defined by

$$F = \omega_n^{jk}, \quad j, k = 0, 1, \ldots, n-1.$$

Give a SymbolicC++ implementation for F.

Problem 34. Consider the linear equation written in matrix form

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}.$$

First show that the determinant of the 3×3 matrix is nonzero. Apply two different methods (Gauss elimination and the Leverrier's method) to find the solution. Compare the two methods and discuss.

Problem 35. Consider the two 3×3 permutation matrices (which are of course then also unitary matrices)

$$U_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad U_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

We want to find efficiently K_1 and K_2 such that $U_1 = e^{K_1}$ and $U_2 = e^{K_2}$. We would apply the spectral decomposition theorem to find K_1, i.e.

$$K_1 = \sum_{j=1}^{3} \ln(\lambda_j) \mathbf{v}_j \mathbf{v}_j^*$$

where λ_j are the eigenvalues of U_1 and $\mathbf{v}_j$ are the corresponding normalized eigenvectors. But then to find K_2 we would apply the property that $U_1^2 = U_2$. Or

could we actually apply that $U_2 = U_1^T$? Note that U_1, U_2, I_3 form a commutative subgroup of the group of 3×3 permutation under matrix multiplication.

Problem 36. Let A, B, C be $n \times n$ matrices. Simplify

$$(A \otimes I_n \otimes I_n)(I_n \otimes B \otimes I_n(I_n \otimes I_n \otimes C)\,.$$

Problem 37. Consider the 3×3 normal matrix

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Study the following four methods to calculate $\exp(A)$. Discuss.
(i) Apply the definition

$$\exp(A) := \sum_{j=0}^{\infty} \frac{A^j}{j!}.$$

(ii) Apply the definition

$$\exp(A) := \lim_{n\to\infty} \left(I_3 + \frac{A}{n}\right)^n.$$

(iii) Apply the spectral theorem, i.e. use the eigenvalues and normalized eigenvectors of A.
(iv) Apply the Cayley-Hamilton theorem which also needs the eigenvalues of A. Keep in mind that one eigenvalue is degenerate.

Problem 38. Consider an $n \times n$ symmetric tridiagonal matrix over $\mathbb{R}$. Let $f_n(\lambda) := \det(A - \lambda I_n)$ and

$$f_k(\lambda) = \det \begin{pmatrix} \alpha_1 - \lambda & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 - \lambda & \beta_2 & \cdots & 0 \\ 0 & \beta_2 & \ddots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \alpha_{k-1} - \lambda & \beta_{k-1} \\ 0 & \cdots & 0 & \beta_{k-1} & \alpha_k - \lambda \end{pmatrix}$$

for $k = 1, 2, \ldots, n$ and $f_0(\lambda) = 1$, $f_{-1}(\lambda) = 0$. Then

$$f_k(\lambda) = (\alpha - \lambda) f_{k-1}(\lambda) - \beta_{k-1}^2 f_{k-2}(\lambda)$$

for $k = 2, 3, \ldots, n$. Find $f_4(\lambda)$ for the 4×4 matrix

$$\begin{pmatrix} 0 & \sqrt{1} & 0 & 0 \\ \sqrt{1} & 0 & \sqrt{2} & 0 \\ 0 & \sqrt{2} & 0 & \sqrt{3} \\ 0 & 0 & \sqrt{3} & 0 \end{pmatrix}.$$

Chapter 7

Recursion

Problem 1. Let a, b be real positive numbers. If $A = (a + b)/2$ denotes the *arithmetic mean* and $B = \sqrt{ab}$ denotes the *geometric mean*, then $A \geq B$ with equality precisely when $a = b$. Given two positive real numbers, a_0 and b_0, where we suppose that $a_0 \geq b_0$, we consider the recursion

$$a_{j+1} = \frac{1}{2}(a_j + b_j), \qquad b_{j+1} = \sqrt{a_j b_j}$$

so that a_{j+1} is the arithmetic mean of a_j and b_j, while b_{j+1} is their geometric mean. Write a C++ program that implements this recursion with $a_0 = 2.0$ and $b_0 = 1.0$.

Problem 2. The *Catalan recurrence* is given by

$$h[n] = \sum_{k=1}^{n} h[k-1] * h[n-k]$$

with $n \geq 0$ and the initial value $h[0] = 1$. The Catalan number $h[1], h[2], \ldots$ arise in a number of problems in combinatorics. Write a C++ program that finds the *Catalan numbers* $h[1], h[2], \ldots, h[6]$ using the Catalan recurrence.

Problem 3. For all $p > 1$, the iteration $(k = 0, 1, 2, \ldots)$

$$x_{k+1} = \frac{1}{p}((p-1)x_k + x_k^{1-p} a), \qquad x_0 = 1$$

converges quadratically to $a^{1/p}$ if a belongs to

$$a \in \{\, z \in \mathbb{C} : \Re z > 0 \text{ and } |z| \leq 1 \,\} \cup \mathbb{R}^+.$$

Write a C++ program that implements this iteration for $p = 3$ and $a = 27$.

Problem 4. Let $0 < x < 2$. The computation of $1/x$ can be done with addition and multiplication using the following recurrence relation

$$a_{j+1} = a_j(1 + c_j), \qquad c_{j+1} = c_j^2 \tag{1}$$

with $j = 0, 1, 2, \ldots$ and the initial values $a_0 = 1$, $c_0 = 1 - x$.
(i) Show that

$$a_j = \frac{1 - c_j}{x} \tag{2}$$

and since $c_j = c_0^{2^j}$ with $|c_0| < 1$, it follows that

$$\lim_{j\to\infty} a_j = \frac{1}{x}.$$

(ii) Write a C++ program that implements this recurrence relation.

Problem 5. Let $n = 1, 2, \ldots$. Consider the definite integral

$$I_n = \int_0^{\pi/2} \cos^n(x)dx.$$

Apply *integration by parts* to find a recursion relation for I_n, i.e. express I_n with I_{n-1}, I_{n-2}. Note that

$$I_1 = \int_0^{\pi/2} \cos(x)dx = 1, \qquad I_2 = \int_0^{\pi/2} \cos^2(x)dx = \frac{\pi}{4}.$$

Problem 6. Find a recursion for

$$I_n = \int_0^{\pi/4} \tan^n(x)dx$$

where $n = 0, 1, \ldots$ and

$$I_0 = \int_0^{\pi/4} dx = \frac{\pi}{4}$$

$$I_1 = \int_0^{\pi/4} \tan(x)dx = -\ln(\cos(x))|_0^{\pi/4} = -\ln(\cos(\pi/4)) + \ln(\cos(0)) = \ln(\sqrt{2}).$$

Problem 7. Let $x \in [0, 1]$. Then $\sqrt{x}$ can be approximated by the sequence of polynomials

$$p_{k+1}(x) = p_k(x) + \frac{1}{2}(x - (p_k(x))^2), \qquad k = 0, 1, 2, \ldots$$

and $k \to \infty$ the sequences converges pointwise to $\sqrt{x}$ with $p_0(x) = x$. Write a C++ program that implements this sequence to find an approximation for $\sqrt{x}$.

Problem 8. The number $\pi/2$ can be calculated using the iteration

$$x_{k+1} = x_k y_k, \qquad y_{k+1} = \sqrt{2y_k/(y_k+1)}, \quad k = 0, 1, 2, \ldots$$

where

$$x_0 = 1, \qquad y_0 = \sqrt{2}.$$

Then

$$\lim_{k\to\infty} x_k = \frac{\pi}{2}.$$

Write a C++ program that implements this iteration and thus finds an approximation of $\pi/2$.

Problem 9. The number π can be calculated using the iteration

$$x_{k+1} = \frac{2x_k y_k}{x_k + y_k}, \qquad y_{k+1} = \sqrt{x_{k+1} y_k}, \quad k = 0, 1, 2, \ldots$$

where

$$x_0 = 2\sqrt{3}, \qquad y_0 = 3.$$

Then

$$\lim_{k\to\infty} x_k = \lim_{k\to\infty} y_k = \pi.$$

Write a C++ program that implements this iteration and thus finds an approximation of π.

Problem 10. (i) Let r be a real nonzero number. Then $1/r$ can be calculated to ever increasing accuracy applying the map

$$x_{t+1} = x_t(2 - rx_t), \qquad t = 0, 1, \ldots$$

provided the initial estimate x_0 is sufficientlty close to $1/r$. The number of digits of accuracy approximately doubles with each iteration. First find the fixed points of the map. Let $r = 3$ and $x_0 = 2$. Find x_1, x_2, ... Discuss.
(ii) The same iteration can be applied to find the multiplicative inverse modulo any power of 2. For example, to find the multiplicative inverse of 5, modulo 256, we start with $x_0 = 1$ (any odd number will do). Then

$$\begin{aligned} x_1 &= x_0(2 - 5 \cdot x_0) = -3 \\ x_2 &= -3(2 - 5 \cdot (-3)) = -51 \\ x_3 &= -51(2 - 5 \cdot (-51)) = -13107. \end{aligned}$$

Thus $-13107 \bmod 256 = 205$. Thus the multiplicative inverse of 5 (modulo 256) is 205. Write a C++ program that implements this algorithm.

Problem 11. According to Gauss, the elliptic integral

$$I = \frac{2}{\pi} \int_0^{\pi/2} \frac{dx}{(a^2 \cos^2(x) + b^2 \sin^2(x))^{1/2}}$$

is equal to the limit of any of the two convergent sequences

$$s_0, s_1, s_2, \ldots \quad \text{or} \quad t_0, t_1, t_2 \ldots$$

as defined by the recurrence relations for $j > 0$

$$\begin{aligned} s_{j+1} &= (s_j + t_j)/2 \\ t_{j+1} &= \sqrt{s_j t_j} \end{aligned}$$

and $s_0 = a$, $t_0 = b$. The calculation of the two sequences is called the arithmetic-geometric mean method. Write a C++ program that implements this method.

Problem 12. Given the sequence of terms

$$t_0,\ t_1,\ t_2,\ \ldots$$

the series of partial sums

$$s_0,\ s_1,\ s_2,\ \ldots$$

is defined such that

$$s_j := t_0 + t_1 + \cdots + t_j \qquad j = 0, 1, 2, \ldots$$

If the sequence is given by the recurrence relation

$$t_{j+1} = f(t_j) \quad \text{for} \quad j \geq 0$$

then the series s_j is determined by

$$\begin{aligned} s_{j+1} &= s_j + t_{j+1} \quad \text{for } j \geq 0 \\ s_0 &= t_0. \end{aligned}$$

Give a C++ implementation for the sequence s_j and the recursion

$$t_{j+1} = f(t_j) = \frac{t_j}{j+1}.$$

Problem 13. Let m be a positive integer and x be a fixed real number. Then we can calculate $\cos(mx)$ using the recursion

$$\cos((m+1)x) = 2\cos(x)\cos(mx) - \cos((m-1)x)$$

Write a C++ program that implements this recursion. Use the values $m = 10$ and $x = 0.1$.

Problem 14. John McCarthy introduced the following recurrence equation ($n \in \mathbb{N}$)

$$f(n) = \textbf{if } n > 100 \textbf{ then } n - 10 \textbf{ else } f(f(n + 11)).$$

Write a C++ program that implements the recurrence equation. What is the output if $n \leq 101$?

Problem 15. Let k be a positive integer $k \geq 2$. Consider the recursion

$$x_{t+1} = x_t + ky_t, \qquad y_{t+1} = x_t + y_t$$

where $t = 0, 1, 2, \ldots$ and $x_0 = y_0 = 1$. Study x_{t+1}/y_{t+1} for $t \to \infty$ and $k = 5$.

Problem 16. Give a C++ implementation of the recursion

$$c_0 = 1, \quad c_1 = -1, \quad c_{n+1} = -\sum_{k=1}^{n} \sum_{i=0}^{k} c_i c_{k-i} c_{n-k+1}.$$

Problem 17. Give a C++ implementation of the recursion

$$x_j(t + 1) = (1 - \epsilon)x_j(t) + \frac{1}{2}(x_{j-1}(t) + x_{j+1}(t)), \quad t = 0, 1, 2, \ldots$$

where $j = 0, 1, 2, 3$ and $-1 \equiv 3$, $4 \equiv 0$.

Problem 18. Consider the function

$$f_n(x) = \int_0^\infty y^n \exp(-y^4 - xy^2) dy, \quad n = 0, 1, \ldots$$

(i) Show that

$$f_{n+4}(x) = \frac{n + 1}{4} f_n(x) - \frac{x}{2} f_{n+2}(x)$$

using integration by parts.
(ii) Show that

$$\frac{df_n(x)}{dx} = -f_{n+2}(x).$$

Problem 19. (i) Consider the recursion

$$x_{t+1} = x_t + 5y_t$$
$$y_{t+1} = x_t + y_t$$

where $t = 0, 1, \ldots$. Let $x_0 = y_0 = 1$. Calculate x_1, y_1, x_2, y_2, x_3, y_3 and x_0/y_0, x_1/y_1, x_2/y_2, x_3/y_3.
(ii) Define

$$z_t := \frac{x_t}{y_t}.$$

Find the recursion for z_t. Find the fixed points of this recursion. Are the fixed point stable? Find

$$\lim_{t\to\infty} z_t$$

with $z_0 = x_0/y_0 = 1$.

Problem 20. Let n be a positive integer. A *Dyk word* is a string of length $2n$ with n x's and n y's such that no initial segment of the string of length $2n$ has more y's than x's.
(i) Give the Dyk words for $n = 1$, $n = 2$ and $n = 3$.
(ii) Describe an algorithm to generate the Dyk words for a given n. Give a recursion.

Problem 21. Let k and n be positive integers. Implement in C++ the recursive function

$$p(k,n) = \begin{cases} 0 & \text{if} \quad k > n \\ 1 & \text{if} \quad k = n \\ p(k+1,n) + p(k,n-k) & \text{otherwise} \end{cases}$$

where $p(1,1) = 1$.

Problem 22. Consider the alphabet $\{A, B, C\}$ and the Fredholm subsitution

$$A \mapsto AB, \quad B \mapsto BC, \quad C \mapsto CC.$$

Start of with A and find the sequence. Then set $A = C = 0$ and $B = 1$ and find the bitstring.

Problem 23. Let $k = 0, 1, \ldots$ and $\phi \in \mathbb{R}$. Consider the integral defined by

$$I_k(\phi) := \int_0^\pi \frac{\cos(k\theta) - \cos(k\phi)}{\cos(\theta) - \cos(\phi)} d\theta \,.$$

Show that $I_0(\phi) = 0$ and $I_1(\phi) = \pi$. Show that $I_k(\phi)$ satisfies the second order difference equation

$$I_{k+2}(\phi) - 2\cos(\phi) I_{k+1}(\phi) + I_k(\phi) = 0, \quad k = 0, 1, \ldots$$

with $I_0(\phi) = 0$ and $I_1(\phi) = \pi$. Solve the second order difference equation.

Problem 24. Let $m = 1, 2, \ldots$. Consider the recursion

$$c_m = \sum_{k=1}^{m} c_{k-1} c_{m-k}$$

with $c_0 = 1$. Define a *generating function*

$$P(x) = \sum_{m=0}^{\infty} c_m x^m = c_0 + c_1 x^1 + c_2 x^2 + \cdots \equiv 1 + c_1 x + c_2 x^2 + \cdots .$$

Then

$$P(x) = 1 + \sum_{m=1}^{\infty} \sum_{k=1}^{m} c_{k-1} c_{m-k} x^m = 1 + x \sum_{k=1}^{\infty} \sum_{m=k}^{\infty} c_{m-k} x^{m-k} c_{k-1} x^{k-1} = 1 + x P^2(x)$$

with the solution for $P(x)$

$$P(x) = \frac{1 - \sqrt{1 - 4x}}{2x}$$

with $P(0) = 1$. Taylor expansion of $P(x)$ provides

$$c_m = \frac{4n - 2}{n + 1} c_{m-1}, \quad m = 1, 2, \ldots$$

and thus ($c_0 = 1$)

$$c_m = \frac{(2m)!}{m!(m+1)!}.$$

Give an implementation of the recursion for c_m and of the two last equations for c_m using SymbolicC++ and the `Verylong` of SymbolicC++.

Problem 25. Consider the tridiagonal $n \times n$ matrix

$$A = \begin{pmatrix} a_1 & b_2 & 0 & \ldots & 0 & 0 \\ c_2 & a_2 & b_3 & \ldots & 0 & 0 \\ 0 & c_3 & a_3 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & a_{n-1} & b_n \\ 0 & 0 & 0 & \ldots & c_n & a_n \end{pmatrix}$$

with $a_1, a_j, b_j, c_j \in \mathbb{C}$ ($j = 1, 2, \ldots, n$). It has in general n complex eigenvalues the n roots of the characteristic polynomial $p(\lambda)$. Show that this polynomial can be evaluated by the recursive formula

$$\begin{aligned} p_k(\lambda) &= (\lambda - a_k) p_{k-1} - b_k c_k p_{k-2}(\lambda), \quad k = 2, 3, \ldots, n \\ p_1(\lambda) &= \lambda - a_1 \\ p_0(\lambda) &= 1. \end{aligned}$$

Problem 26. Let $I_\nu(x)$ be the modified Bessel function. For the asymptotic expansion we have

$$\frac{I_{\nu+1}(x)}{I_\nu(x)} \sim \sum_{j=0}^{\infty} c_j x^{-j}$$

The expansion coefficients c_j obey the quadratic recursive relation

$$c_0 = 1, \quad c_1 = -\left(\nu + \frac{1}{2}\right), \quad c_2 = c_3 = \frac{1}{2}\left(\nu^2 - \frac{1}{4}\right),$$

$$2c_j = (j-1)c_{j-1} - \sum_{\ell=2}^{j-2} c_\ell c_{j-\ell}, \quad j \geq 4.$$

Give a SymbolicC++ implementation of this recursion applying the `Verylong` class. Then apply it to $\nu = 1/2$ and $\nu = -1/2$.

Problem 27. Let $a > 0$. Give a SymbolicC++ implementation of the recursion

$$\begin{aligned} c_0 &= 1 \\ c_1 &= -(a + 1/2) \\ c_2 &= c_3 = \frac{1}{2}(a^2 - 1/4) \\ c_n &= \frac{1}{2}(n-1)c_{n-1} - \frac{1}{2}\sum_{\ell=2}^{n-2} c_\ell c_{n-\ell}, \quad n \geq 4. \end{aligned}$$

Problem 28. Let $j = 2, 3, \ldots$ and

$$P_{j+1} = \frac{P_j^2}{P_{j-1}} + P_j$$

with $P_1 = 1$ and $P_2 = 2$. Give a SymbolicC++ implementation utilizing the class `Verylong`.

Problem 29. Let $x_1 = \sqrt{1} = 1$, $x_2 = \sqrt{1+\sqrt{2}}$. Study the recurrence relation

$$x_{t+2} = \sqrt{1 + \sqrt{2 + x_t}}, \quad t = 1, 2, \ldots$$

Find $\lim_{t\to\infty} x_t$.

Chapter 8

Numerical Techniques

Problem 1. The exponential function e^x ($x \in \mathbb{R}$) can be defined as

$$e^x := \lim_{n\to\infty} (1 + x/n)^n \tag{1}$$

or

$$e^x := \sum_{k=0}^{\infty} \frac{x^k}{k!}. \tag{2}$$

The above two formulae give methods to calculate e^x numerically. The second expression is more convenient for such a purpose, because the convergence of (2) is better than of (1). Is there a way to combine the two definitions for a more rapidly converging expression for e^x?

Problem 2. Calculating $\sqrt{2}$ an elementary and ancient recursion consists of the double sequence

$$p_{j+1} = p_j + 2q_j, \qquad q_{j+1} = p_j + q_j$$

with $j = 0, 1, \ldots$ and

$$\lim_{j\to\infty} \frac{p_j}{q_j} = \sqrt{2}.$$

(i) Calculate the first three terms with the initial values $p_0 = q_0 = 1$.
(ii) Give an error estimation with $\epsilon_j := |\sqrt{2} - p_j/q_j|$.
(iii) Write the problem in matrix notation and solve it.

Problem 3. Let f be an analytic function. *König's root-finding algorithm* is given by the iteration

$$x_{j+1} = x_j + (n-1)\frac{(1/f)^{[n-2]}(x_j)}{(1/f)^{[n-1]}(x_j)}$$

where $n \geq 2$, $(1/f)^{[k]}$ denotes the kth derivative of $1/f$ and $j = 0, 1, 2, \ldots$ with x_0 to be the initial value. Show that for $n = 2$ we obtain *Newton's method* and for $n = 3$ we obtain *Halley's method*. Derive the iteration for the case $n = 4$. Write a C++ program for this case with the analytic function $f(x) = \sin(2x) - \sinh(x)$ and the initial condition $x_0 = 1$.

Problem 4. Consider the case of solving the *quadratic equation*

$$ax^2 + bx + c = 0, \qquad a \neq 0.$$

The usual way the two roots x_1 and x_2 are computed is

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

If a, b, and c numbers such that $-b$ is about the same size as $\sqrt{b^2 - 4ac}$ (with respect to the arithmetic used), then a *catastrophic cancellation* will occur in computing x_2. As a result, the computed value of x_2 can be completely erroneous.
(i) Consider the case with $a = 1$, $b = -10^5$ and $c = 1$ and eight-digit arithmetic.
(ii) How can this problem be avoided?

Problem 5. Let $a > 0$. Find an iteration to approximate $\sqrt{a}$. Use the fact that if x ($x > 0$) is the actual square root of a, then $x = a/x$; that is, the two factors x and a/x are equal.

Problem 6. For any positive number h we define an operator S_h which replaces a continuous function $f : \mathbb{R} \to \mathbb{R}$ by its average over an interval with length h and centre x

$$(S_h f)(x) := \frac{1}{h}\int_{x-h/2}^{x+h/2} f(t)dt.$$

We find that

$$\lim_{h\to 0}(S_h f)(x) = f(x).$$

(i) Show that the operator S_h is linear.
(ii) Show that the operator S_h leaves linear functions unchanged.
(iii) Calculate $S_h f$ for $f(t) = e^{-|t|}\sin(t)$ with $h = 0.1$.

Problem 7. (i) Provide a fast algorithm to calculate $\sqrt{2}$.

(ii) Provide a fast algorithm to calculate the golden mean number $\varphi = (1+\sqrt{5})/2$.

Problem 8. If a beam runs into an obstacle a part of the signal is transmitted the rest reflected. The difference between the frequency of the reflected part η and the initial frequency η_0 is the so-called *Doppler shift* $\Delta\eta$ caused by a particle moving with velocity ν in the direction opposite to the transmitted signal. It can be calculated as

$$\Delta\eta = \eta - \eta_0 = \frac{2c\eta_0\nu}{c^2 - \nu^2}$$

where c denotes the velocity of sound within the medium. We have $\nu \ll c$. Can the calculation be simplified?

Problem 9. Given pairs of single precision numbers (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4). Decide whether the line segement $(x_1, y_1) - (x_2, y_2)$ intersects the line determined by $(x_3, y_3) - (x_4, y_4)$ at a unique point. If so, compute the coordinates (x, y) of the intersection point accurate to single precision. That is, if the exact intersection point is (x^*, y^*), then return a point (x, y) such that x is the nearest single precision number to x' and y is the nearest single precision number to y'.

Problem 10. The *harmonic series* can be approximated by

$$\sum_{j=1}^{n} \frac{1}{j} \approx 0.5772 + \ln(n) + \frac{1}{2n}.$$

Calculate the left and right hand side for $n = 1$ and $n = 10$.

Problem 11. Consider the linear first order delay-differential equation

$$\frac{du}{dt} = -u(t-1).$$

We can find the solution with the ansatz

$$u(t) = Ce^{\lambda t}.$$

Since $du/dt = C\lambda e^{\lambda t}$ and $u(t-1) = Ce^{\lambda(t-1)}$ we obtain

$$\lambda = -e^{-\lambda}.$$

There is no solution if λ is real. If λ is complex then there are an infinite number of solutions. They are given by the *Lambert W function*. Write a C++ program that finds some of the solutions.

Problem 12. Let $a, b, c, d > 0$ and $a + b + c > d$. Consider the problem of relating the input and output crank angles of a four-bar mechanism. The angles

θ and ϕ, respectively, are measured from the line of the fixed pivots. The moving links of fixed length are a, b, c. The fixed link is d. This provides us with the equation

$$b^2 = c^2 + d^2 + a^2 - 2dc\cos(\phi) - 2ac\cos(\phi)\cos(\theta) - 2ac\sin(\phi)\sin(\theta) + 2ad\cos(\theta).$$

With $C_1 = d/c$, $C_2 = d/a$, $C_3 = (d^2 + a^2 - b^2 + c^2)/(2ac)$ we obtain the *Freudenstein equation*

$$C_1\cos(\theta) - C_2\cos(\phi) + C_3 - \cos(\theta - \phi) = 0.$$

Set $a = 1$, $b = c = d = 2$. Solve this transcendental equation with the Newton method for different fixed θ and solve for ϕ.

Problem 13. Let $f : \mathbb{R} \to \mathbb{R}$ be an analytic function. We define

$$\Delta f(x) := \frac{f(x-\epsilon) + f(x+\epsilon) - 2f(x)}{\epsilon^2}.$$

(i) Let $f(x) = x^2$. Find $\Delta f(x)$. Study $\epsilon \to 0$.
(ii) Give a C++ implementation for $\Delta f(x)$.
(iii) Calculate the derivative of $f(x) = x^2$ using

$$f'(x) := \lim_{\epsilon \to 0} \frac{-11f(x) + 18f(x+\epsilon) - 9f(x+2\epsilon) + 2f(x+3h)}{6\epsilon}.$$

Problem 14. Given a sequence of ordered parameters (knots): $(x_0, x_1, \ldots, x_m)$, the ith normalized *B-spline basis function* (B-function) $N_{i,k}$ of order k is defined recursively as

$$N_{i,k}(x) = \begin{cases} 1 & \text{if} \quad x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \qquad \text{if } k = 1$$

$$N_{i,k}(x) = \frac{x - x_i}{x_{i+k-1} - x_i} N_{i,k-1}(x) + \frac{x_{i+k} - x}{x_{i+k} - x_{i+1}} N_{i+1,k-1}(x) \qquad \text{if } k > 1$$

with $i = 0, 1, \ldots, k$ and $k < m$. The properties of the B-spline basis functions are:
1) *Partition of unity*, i.e.

$$\sum_{i=0}^{m-k} N_{i,k}(x) = 1.$$

2) Positivity

$$N_{i,k}(x) \ge 0.$$

3) Local support

$$N_{i,k}(x) = 0 \text{ for } x \notin [x_i, x_{i+k}].$$

4) C^{k-2} continuity. If the knots $\{ x_i \}$ are pairwise different from each other, then $N_{i,k}(x) \in C^{k-2}$, i.e., the function $N_{i,k}(x)$ is $(k-2)$ times continuously differentiable.

Let $m = 4$ and

$$x_0 = 0,\ x_1 = 0.5,\ x_2 = 1.0,\ x_3 = 1.5,\ x_4 = 2.0.$$

Let $k = 2$. Find $N_{0,2}(x)$, $N_{1,2}(x)$ and $N_{2,2}(x)$. Draw the functions.

Problem 15. Let f be a continuos function in the interval $[a, b]$ $(b > a)$. Then

$$\lim_{n\to\infty} \frac{b-a}{n} \sum_{k=1}^{n} f\left(a + \frac{k(b-a)}{n}\right) = \int_a^b f(x)dx.$$

Implement in C++ the left-hand side for a given f, a, b and n with

```
double integrate(double (*f)(double),double a,double b,int n)
```

Apply it to the function

$$f(x) = \sin(x)$$

and the interval $[0, \pi]$ and to the function

$$f(x) = \exp(x * \ln(x))$$

with the interval $[0, 1]$. Choose $n = 10, 100, 1000000$.

Problem 16. (i) Use the class `Derive` of SymbolicC++ to find the derivative of

$$y = 2x^3 - 5x - 1$$

at the point $x = 2$. Use the data type `double` for x.
(ii) Use the class `Derive` of SymbolicC++ and the class `complex` over `double` to find the derivative of the complex-valued function

$$w = 2z^2 - 5z - 1$$

at the point $z = i$.

Problem 17. Let r be a real number with $r \neq 0$. Then $1/r$ can be calculated to ever increasing precision by using the iteration

$$x_{t+1} = x_t(2 - rx_t), \qquad t = 0, 1, 2, \ldots$$

provided the initial value x_0 is sufficiently close to $1/r$. The number of digits of precision approximately doubles with each iteration. Write a C++ program to

find the inverse of $r = 2$ with the initial value $x_0 = 0.8$. Why does the initial value $x_0 = 1$ not work?

Problem 18. In C and C++ the function `fabs` finds the absolute value of a floating point number. How can `fabs` be replaced by an `if` condition and multiplication by -1.0?

Problem 19. What is the following code doing

```
// InvSqrt.cpp

#include <iostream>
using namespace std;

float invSqrt(float x)
{
  float xhalf = 0.5f*x;
  int i = *(int*)& x;         // get bits for floating value
  i = 0x5f3759df - (i >> 1); // initial guess
  x = *(float*)& i;           // convert bits back to float
  x *= 1.5f - xhalf*x*x;
  x *= 1.5f - xhalf*x*x;
  return x;
}

int main(void)
{
 float x1 = 10.0f;
 float r1 = invSqrt(x1);
 cout << "r1 = " << r1 << endl;

 float x2 = 100.0f;
 float r2 = invSqrt(x2);
 cout << "r2 = " << r2 << endl;
 return 0;
}
```

Problem 20. Consider the function $f : \mathbb{R}^n \to \mathbb{R}$, where it is assumed that the function f is at least twice continuous differentiable. We want to find the minimum of the function f. Let

$$H(\mathbf{x}) := \left(\frac{\partial^2 f(\mathbf{x})}{\partial x_j \partial x_k} \right)$$

be the *Hessian matrix* and $j, k = 1, 2, \ldots, n$. In the *Levenberg-Marquardt algorithm* we apply

$$\mathbf{x}_{t+1} = \mathbf{x}_t - (H(\mathbf{x}_t) + \lambda \mathrm{diag}(H(\mathbf{x}_t))^{-1} \nabla f(\mathbf{x}_t)$$

where $t = 0, 1, 2, \ldots$, λ is the step length and given initial values. We need matrix inversion as part of the update. Since the determinant of the Hessian matrix is proportional to the curvature of f, the iteration implies a large step in the direction with low curvature (i.e., an almost flat terrain) and a small step in the direction with hight curvature (i.e. a steep incline). Write a C++ program that applies this method to solve the system of equation

$$x_1 = \sin(x_1 + x_2), \qquad x_2 = \cos(x_1 - x_2)$$

by finding the minimum the function

$$f(x_1, x_2) = (x_1 - \sin(x_1 + x_2))^2 + (x_2 - \cos(x_1 - x_2))^2.$$

Use the initial values $x_{1,0} = x_{2,0} = 0$.

Problem 21. Given a polynomial of degree n which admits n real roots. Write a C++ program using the Newton method that finds all real roots. Apply the program to the polynomial

$$p(x) = x^4 - 7x^3 + 8x^2 + 2x - 1.$$

Problem 22. A *polygon* is a closed plane figure with n sides. If all sides and angles are equivalent the polygon is called regular. The area of a planar convex polygon with vertices

$$(x_0, y_0), \ (x_1, y_1), \ \ldots, \ (x_{n-1}, y_{n-1})$$

is given by

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i), \qquad x_n \equiv x_0, \ y_n \equiv y_0.$$

A *polygon* in the plane $\mathbb{R}^2$ is a closed figure with n sides. If all sides and angles are equal the polygon is called regular. The area of a planar convex polygon with vertices

$$(x_0, y_0), \ (x_1, y_1), \ \ldots, \ (x_{n-1}, y_{n-1})$$

is given by

$$A = \frac{1}{2} \left| \sum_{j=1}^{n-1} (x_j y_{j+1} - x_{j+1} y_j) \right|, \quad x_n \equiv x_0, \ y_n \equiv y_0.$$

(i) Write a C++ program that finds the area of a given planar convex polygon. Apply the modulus operator `%` to identify n and 0.

(ii) Write a Java program that finds the area of a given planar convex polygon. Apply the *modulus operator* `%` to identify n and 0.

Problem 23. Given a time series $\{ x_i : i = 0, 1, \ldots, N-1 \}$, the linear correlation of an epoch consisting of K points $\{ x_i : i = 0, 1, \ldots, K-1 \}$ and another epoch of the same length K covering a different time interval $\{ x_k : k = j, j+1, \ldots, K+j-1 \}$, $K + j \leq N$, is given by the *cross-correlation coefficient*

$$r_j := \sum_{i=0}^{K-1} \frac{(x_i - \langle x_0 \rangle)(x_{i+j} - \langle x_j \rangle)}{\sigma_0 \sigma_j}$$

where $\langle x_0 \rangle$ and $\langle x_j \rangle$ are, respectively, the mean values of the epochs starting at x_0 and x_j and σ_0 and σ_j the corresponding standard deviations. Write a C++ program that find the correlation coefficient for the logistic map

$$x_{t+1} = 4x_t(1 - x_t), \qquad t = 0, 1, 2, \ldots$$

and $x_0 = 1/3$. Let $N = 2048$ and $K = 1024$.

Problem 24. The standard Hermite polynomial satisfy the recursion relations

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \qquad \frac{dH_n(x)}{dx} = 2nH_{n-1}(x)$$

with $H_0(x) = 1$. Combining these two relations we get the recursion relation

$$H_{n+1}(x) = \left(2x - \frac{d}{dx} \right) H_n(x).$$

Write a C++ program using SymbolicC++ which implement this recursion relation.

Problem 25. Write a C++ program to implement an approximation to the integral

$$\int_0^x \exp(-s^2) ds = x - \frac{x^3}{3 \cdot 1!} + \frac{x^5}{5 \cdot 2!} - \frac{x^7}{7 \cdot 3!} + \cdots$$

Problem 26. Consider the 3×3 matrix

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Find A^2 and A^3. We know that

$$\mathrm{tr}(A) = \lambda_1 + \lambda_2 + \lambda_3, \quad \mathrm{tr}(A^2) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \quad \mathrm{tr}(A^3) = \lambda_1^3 + \lambda_2^3 + \lambda_3^3.$$

Use Newton's method to solve this system of equations to find the eigenvalues of A.

Problem 27. What is the output of the following C++ code

```
// epsilon.cpp

#include <iostream>
using namespace std;

int main(void)
{
 double eps = 1.0, x = 2.0, y = 1.0;
 while(y < x) { eps *= 0.5; x = 1.0 + eps; }
 eps *= 2.0;
 cout << "eps = " << eps;
}
```

Problem 28. Let $x > 1$. Then $\ln(x)$ can be calculated using the iteration $x_0 = x$

$$x_{j+1} = \frac{2x_j}{1 + \sqrt{1 + 2^{-j}x_j}}, \quad j = 0, 1, 2, \ldots$$

Write a C++ program that implements this iteration. Apply it to $x = 10$ and $x = e$.

Problem 29. The junction between p- and n-type semiconductors has properties which make it the basis of many electronic devices. For a p-n junction diode we have the equation

$$d = \left(\frac{2\epsilon_0 \epsilon_r (n_A + n_D)(V_D - U)}{e n_A n_D} \right)^{1/2}$$

where d has the dimension of a length (thickness of the deplection layer) and

$$\begin{aligned} e &= 1.6021 \cdot 10^{-19}\, As \quad \text{elementary charge} \\ \epsilon_0 &= 8.8542 \cdot 10^{-12}\, As/Vm \quad \text{permittivity of the vacuum} \\ \epsilon_r &= 16 \quad \text{for Germanium} \\ n_A = n_D &= 10^{22}\, m^{-3} \quad \text{donor and acceptor concentrations} \\ V_D &= 0.358\, V \quad \text{diffusion voltage} \\ a &= 10^{-6}\, m^2 \quad \text{area of the junction} \end{aligned}$$

Write a C++ program that finds d in dependence of the voltage U with $-10V \leq U \leq 0V$ and step size $0.5V$ and the capacity

$$C(d) = \epsilon_0 \epsilon_r a / d.$$

All the units are in the MKSA system. Thus the result for d will be in meters and the result for the capacity C will be in Farad ($= s^4A^2/m^2kg$).

Problem 30. Use numerical integration to show that

$$\int_0^1 \frac{x}{1+x^2}\ln(1+x)dx = 0.162865007.$$

Problem 31. Consider the one-dimensional Schrödinger equation (eigenvalue problem)

$$-\frac{\hbar^2}{2m}\frac{d^2u(x)}{dx^2} + V(x)u(x) = Eu(x).$$

For numerical studies the eigenvalue equation is replaced by the difference equation

$$-\frac{\hbar^2}{2m}\left(\frac{u_{n+1}+u_{n-1}-2u_n}{\delta^2}\right) + V_nu_n = Eu_n$$

where $\delta := x_{n+1}-x_n$ (step size), $u_n := u(x_n)$, $V_n = V(x_n)$. Imposing the boundary conditions the set of linear equations can be solved as eigenvalue problem of a symmetric matrix over the real numbers. Show that the error in the representation relative to $(2mE/\hbar^2)u_n$ is

$$\frac{1}{12}\delta^2u_n^{(4)}.$$

The Numerov-Cooley approximation uses the second order difference equation

$$-\frac{\hbar^2}{2m}\left(\frac{u_{n+1}+u_{n-1}-2u_n}{\delta^2}\right) = \frac{5}{6}(E-V_n)u_n+\frac{1}{12}(E-V_{n+1})u_{n+1}+\frac{1}{12}(E-V_{n-1})u_{n-1}.$$

Show that relative error using this approximation is

$$\frac{29}{300}\delta^4u_n^{(6)}.$$

Note that this approximation gives an asymmetric matrix equation.

Problem 32. Let $x > 0$. Find the solution of the equation

$$(x-2)^2 = \ln(x).$$

First show that the equation has a root in $[1, 2]$.

Problem 33. Consider the mathematical expression

$$\sin(b) + a * b \underbrace{+}\ c * d + (a-b).$$

(i) Write this mathematical expression as a binary tree with the root indicated by the underbrace. Then evaluate this binary tree from bottom to top with the values $a = 2$, $b = \pi/2$, $c = 4$, $d = 1$.
(ii) Am alternative to represent a mathematical expression as tree is multiexpression programming. Use multiexpression programming to evaluate the mathematical expression given above.

Problem 34. Find an approximation of $\sqrt{30}$ utilizing

$$\sqrt{30} = \sqrt{25 + 5} = 5\sqrt{1 + 0.2}.$$

Chapter 9

Random Numbers

Problem 1. Write a C++ program that implements the *Durstenfeld algorithm* for randomly shuffling a one-dimensional array of integers.

Problem 2. Calculate the integral

$$\int_0^1 |\cos(2\pi x)| dx$$

using the random numnber generator described in problem 7, chapter 10, page 250, Problems and Solutions in Scientific Computing. Compare to the exact result by solving the integral.

Chapter 10

Optimization Problems

Problem 1. Consider an overdetermined linear system $A\mathbf{x} = \mathbf{b}$, where A is an $m \times n$ matrix with $m > n$. Thus $\mathbf{x}$ is a column vector with n rows and $\mathbf{b}$ is a column vector with m rows. Write a genetic algorithm program that finds the Chebyshev or *minmax solution* to set of overdetermined linear equations $A\mathbf{x} = \mathbf{b}$, i.e. the column vector $\mathbf{x}$ which minimizes

$$c = \max_{1 \le i \le m} c_i \equiv \max_{1 \le i \le m} \left| b_i - \sum_{j=1}^{n} a_{ij} x_j \right| .$$

Apply the program to the overdetermined linear system

$$\begin{pmatrix} 1 & -1 & 1 \\ 1 & -0.5 & 0.25 \\ 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0.5 \\ 0 \\ 0.5 \\ 2.0 \end{pmatrix} .$$

Chapter 11

String Manipulations

Problem 1. A protein sequence can be considered as a sequence from a 20-letter alphabet $A = \{ a_1, a_2, \ldots, a_n \}$ where $n = 20$ and a_i is called the letter of type i $(1 \leq i \leq n)$. Let A^ℓ be the set of sequences of length ℓ over A for a positive integer ℓ. Given a sequence S in A^ℓ, we denote by v_i the number of occurrences of a_i in S $(1 \leq i \leq n)$. We have $v_1 + v_2 + \cdots + v_n = \ell$. The vector $V = (v_1, v_2, \ldots, v_n)$ is called the frequency vector of S. For a sequence $S = s_1 s_2 \ldots s_{\ell-1} s_\ell$ of length ℓ, the *Shannon entropy* of S is defined as

$$H(S) = -\sum_{i=1}^{n} \frac{v_i}{\ell} \log \left(\frac{v_i}{\ell} \right).$$

This can be regarded as the average information per position in the sequences. Write a C++ program that implements $H(S)$.

Problem 2. Given a string of digits with or without a decimal point, for example "345678" or "12.3456". Write a C++ program that converts the string to a floating point number `double`.

Problem 3. Write a C++ program that implements the *Levenshtein distance* (also called the edit distance).

Problem 4. Write a C++ program that find the first character in an ASCII string that occurs only once. Find a solution that minimizes the number of comparisons between characters. Note that `'\0'` denotes the null character.

Problem 5. Given a totally ordered infinite alphabet A, represented by $\{1, 2, 3, \dots\}$. The free associative algebra over an alphabet A, i.e., the algebra spanned by words, the product being the concatenation, is denoted by $\mathbf{K}\langle A\rangle$, and its unity (the empty word) is denoted by ϵ. Here, $\mathbf{K}$ is some field of characteristic zero. An algebraic structure on $\mathbf{K}\langle A\rangle$ is known as the *shuffle product*. Let w_1 and w_2 be two words. Then the shuffle product $w_1 \sqcup w_2$ is recursively defined by

$$\begin{aligned} w_1 \sqcup \epsilon &= w_1 \\ \epsilon \sqcup w_2 &= w_2 \\ au \sqcup bv &= a(u \sqcup bv) + b(au \sqcup v) \end{aligned}$$

where a, b are letters and u, v are words. For example

$$\begin{aligned} 12 \sqcup 43 &= 1(2 \sqcup 43) + 4(12 \sqcup 3) \\ &= 12(\epsilon \sqcup 43) + 14(2 \sqcup 3) + 41(2 \sqcup 3) + 43(12 \sqcup \epsilon) \\ &= 1243 + 142(\epsilon \sqcup 3) + 143(2 \sqcup \epsilon) + 412(\epsilon \sqcup 3) + 413(2 \sqcup \epsilon) + 4312 \\ &= 1243 + 1423 + 1432 + 4123 + 4132 + 4312. \end{aligned}$$

Write a C++ program that implements this product.

Problem 6. (i) Write a C++ program that uses the string class and the line

```
string* sa = new string[N];
```

and then concatentes the strings.
(ii) Write a C++ program that uses the string class and the line

```
string* sa = new string(6,' ');
```

and then concatentes the characters.

Chapter 12

Programming Problems

Problem 1. The rank of an element in a sequence (one-dimensional array) of numbers is the number of smaller elements in the sequence plus the number of equal elements that appear to its left. For example, if the sequence is given as the one-dimensional array $a = [4, 3, 9, 3, 7]$, then the ranks are $r = [2, 0, 4, 1, 3]$. Write a C++ program with a function `void rank(T* a,int n,int* r)` that computes the ranks of the elements of the array $a[0 : n-1]$. Once the elements have been ranked using the function `rank()` write a function `rearrange()` that rearrange them in nondecreasing order so that $a[0] \leq a[1] \leq \cdots \leq a[n-1]$ by moving elements to positions corresponding to their ranks.

Problem 2. Write a C++ program that transposes a square matrix in-place.

Problem 3. Given an array of 20 integer numbers. We would like to assign the numbers $0, 4, 8, \ldots, 76$ to the array, i.e.,

```
array[0] = 0, array[1] = 4, ... , array[19] = 76.
```

Can the multiplication `i*4` in the following C++ program `forloop1.cpp` be avoided and replaced by addition? Can the C++ program be even more efficient?

```
// forloop1.cpp

#include <iostream>
using namespace std;

int main(void)
{
  int* array = new int[20];
```

```
  for(int i=0;i<20;i++) { array[i] = i*4; }
  delete[] array;
  return 0;
}
```

Problem 4. In *wavelet theory* we start from a function (the so-called mother wavelet) and then by scaling and translation of the dependent variable x we generate a set of new functions. The most studied mother wavelet is the Haar function implemented in the following C++ program.

```
// wavelets.cpp

#include <iostream>
using namespace std;

template <class T> T transform(T (*f)(T),T x,T a,T b)
{ return f(a*x + b); }

double H(double x)
{
  if((x >= 0.0) && (x <= 0.5)) return 1.0;
  if((x > 0.5) && (x <= 1.0)) return -1.0;
  return 0.0;
}

int main(void)
{
  double r1 = transform(H,-4.0,0.5,4.0);
  cout << "r1 = " << r1 << endl;
  double r2 = transform(H,-1.6,2.0,4.0);
  cout << "r2 = " << r2 << endl;
  double r3 = transform(H,2.6,4.0,-10.0);
  cout << "r3 = " << r3 << endl;
  return 0;
}
```

What is the output of the program?

Problem 5. Let n be a positive integer. Then the solutions of $z^n = 1$ are called the roots of unity. For example, if $n = 4$ we have $z_1 = 1$, $z_2 = i$, $z_3 = -1$, $z_4 = -i$. Write a C++ program using the class `complex<double>` that stores the unit roots in an array for a given n.

Problem 6. Extend the C++ program `Gauss0.cpp` to a template basis so that it can also be used for other data types such as `Rational<int>` and `Rational<Verylong>`.

Problem 7. Consider the system of nonlinear equations

$$3x^2 - 2y^2 - 4z^2 + 54 = 0, \qquad 5x^2 - 3y^2 - 7z^2 + 74 = 0.$$

Write a C++ program that finds all integer solutions in the range $0 \leq x \leq 100$, $0 \leq y \leq 100$, $0 \leq z \leq 100$.

Problem 8. Let $a_0, a_1, a_2, \ldots, a_{n-1}$ be a finite sequence of numbers. Its *Cesáro sum* is defined as

$$\frac{1}{n}(s_0 + s_1 + s_2 + \cdots + s_{n-1})$$

where

$$s_k = a_0 + a_1 + \cdots + a_k$$

for each k, $0 \leq k \leq (n-1)$. Write a C++ program that finds the Cesáro sum for a given finite sequence of numbers. Use templates so that the program can also be used for complex numbers, rational numbers etc.

Problem 9. Write a C++ program that finds the maximum of the function

$$f(\mathbf{x}) = x_1^2 - 2x_1 + x_2^2 + 3x_2 + x_3^2 + 4x_3$$

subject to the constraint

$$x_1^2 + 2x_2^2 + 3x_3^2 \leq 17$$

where x_1, x_2, x_3 are non-negative integers.

Problem 10. Let $x, y \in \mathbb{Z}$. Consider the equation

$$x^4 + y^4 + 79 = 48xy.$$

Write a C++ program that finds all solutions in the range $-10 \leq x \leq 10$ and $-10 \leq y \leq 10$.

Problem 11. LISP is latently typed, i.e. does not explicitly specify the underlying data type for variables. Is it possible to achieve the same with C++? In other words, can be construct an abstract datatype in C++ such that the underlying data type can be stored and used in a form that does not explicitly state the underlying data type? Give a possible implementation in standard C++.

Problem 12. Write a C++ program that finds the numbers of integer solutions of the equation $i_1 + i_2 + i_3 = 12$ satisfying the following constraints

$$0 \leq i_1 \leq 6, \quad 0 \leq i_2 \leq 6, \quad 0 \leq i_3 \leq 3.$$

Problem 13. The `vector` class of the Standard Template Library in C++ is a container class. Arithmetic operation such as addition of vectors are not implemented. Write a C++ program that overloads `+` so that one can add two vectors.

Problem 14. Let x be the number of man, y be the number of woman and z be the number of children. Altogether there are 100 persons. Given 100 kg of potatos. Every man gets 3 portions, a woman gets 2 portions and a child gets 1/2 a portions. Find all (integer) solutions. We have two linear equations with three unkowns, however we have the constraint that x, y, z are nonnegative integers. Write a C++ program that finds all these integer solutions.

Problem 15. Consider a linked list. Determine if the linked list loops using only two pointers.

Problem 16. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Consider the program (page 25) `au.cpp`. Where is the amplitude? Extend the program to two or more sine waves (for example frequency 880 besides 440).

Problem 17. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Consider the program (page 8) `SineSound.java`. Run the Java program with the first line (page 10) changed to

```
...new File("sine.au"));
```

with `WAVE` also changed. Compare to the previous problem. Extend the program to get in more frequencies.

Problem 18. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Consider the program (page 31) `LGB.java`. Rewrite the program in C++ either with the `vector` class of STL or "plain" (just functions).

Problem 19. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Consider the program (page 44) `Noise.java`. Rewrite the program into C++.

Problem 20. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Consider the Matlab Filter Implementation (page 49). Rewrite the code in C++.

Problem 21. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Consider the program (page 60)

`NonCircular.java`. Rewrite the code in C++.

Problem 22. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Give a Java implementation of the two-dimensional convolution (page 61).

Problem 23. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Implement the two-dimensional Fourier transform in C++ (page 72).

Problem 24. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Implement the two-dimensional Cosine transform (page 76).

Problem 25. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Use the `complex` class of STL and do someting useful with the z-transform (chapter 8).

Problem 26. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Implement two-dimensional wavelets (page 89).

Problem 27. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". In the C++ program (page 138) the total number of distinct observations is 2. Extend the program to more observations.

Problem 28. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Extend the C++ code fragment (page 205) to a complete C++ program.

Problem 29. The problem refers to the book "Mathematical Tools in Signal Processing with C++ and Java Simulations". Implement the decompression procedure (page 216) in C++.

Problem 30. Write a Java program `Gauss.java` that implements Gauss elimination to solve linear equation with n equations and n unkowns. Apply the program to the system

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Problem 31. Which of the following C++ program fragments will loop forever?

```
int i = 1;
  while(i != 0) i = i + 1;

int j = 1;
  while(j != 0) j = 2*j + 1;

double x = 1.0;
  while(x != 0) x = x/2.0;
```

Problem 32. Write down the tree expression for

$$((a/b) * 4) + ((1 + c) * (2 - d)).$$

Then evaluate the tree expression for $a = 1$, $b = 2$, $c = 3$, $d = 4$.

Problem 33. How would one store a matrix using a linked list?

Chapter 13

Applications of STL in C++

Problem 1. Given two sets of strings, for example,

```
{ "Jones", "Miller", "Steeb", "Smith" }
{ "Hardy", "Copper", "Steeb" }
```

Apply the `set` class of STL to find the union (method `set_union`), the intersection (method `set_intersection`), and the difference (method `set_difference`) of the two sets. Also apply the method `includes` to test for subsets. Finally use the method `size()` to find the number of elements in the sets.

Problem 2. Given a set of n strings, say

```
"ballon", "tree", "fish", "dog", "cat"
```

with $n = 5$. We want to display all possible $m = 2^n$ subsets of S (including the set itself and the empty set represented by `{ }`). Use recursion.

Problem 3. Given a set of positive integers, for example,

```
{ 2, 3, 11, 7, 4, 28, 41, 16 }
```

Write a C++ program using `set<int>` that partitions the set into two subsets of even and odd numbers, respectively.

Problem 4. Write a useful C++ program that uses the function `find_if()`. The function `find_if` takes a predicate (function object or function) as parameter.

Problem 5. A function f is a set of pairs (a, b) where $a \in A$ and $b \in B$ are elements of the sets A and B, respectively, such that for each $a \in A$ there is exactly one $b \in B$ such that $(a, b) \in f$.

Example. Consider the sets

$$A = \{one,\, two,\, three,\, four,\, five\}.$$

and

$$B = \{1,\, 2,\, 3,\, 4,\, 5\}.$$

Let

$$f = \{(one, 1),\, (two, 2),\, (three, 3),\, (four, 4),\, (five, 5)\}$$

then $f(one) = 1$, $f(two) = 2$ etc. The set f denotes a function since each element of A appears as the first of a pair exactly once in f. The set A is called the domain of f and the set B is called the range of f.

We usually write $f : A \to B$ and $f(a) = b$ where $(a, b) \in f$.

Example. Consider

$$A = \{1,\, 2,\, 3,\, 4,\, 5\}$$

and

$$B = \{true,\, false\}.$$

Let

$$g = \{(1, false),\, (2, true),\, (3, true),\, (4, false),\, (5, true)\}$$

then $g(2) = true$, $g(4) = false$ etc. The function g associates with each number in A the value in B which indicates whether the number is prime.

Let $f : A \to B$ and $g : B \to C$ where A, B and C are sets. Then, by convention, $a \in A \Rightarrow f(a) \in B$. Consequently

$$a \in A \Rightarrow f(a) \in B \Rightarrow g(f(a)) \in C.$$

Thus we define function composition

$$g \circ f : A \to C, \quad g \circ f = \{(a, g(f(a))) \,|\, a \in A\}.$$

Note that the functions f and g need to be compatible for function composition, i.e. the range of f must be contained in (subset of) the domain of g.

Example. Let f and g denote the two functions from the examples above and (we rename the sets for clarity)

$$A = \{one,\, two,\, three,\, four,\, five\},$$
$$B = \{1,\, 2,\, 3,\, 4,\, 5\},$$
$$C = \{true,\, false\}.$$

Thus $f : A \to B$ and $g : B \to C$. Consequently

$$\begin{aligned} g \circ f = &\{(one, g(f(one))),\ (two, g(f(two))),\ (three, g(f(three))),\\ &\quad (four, g(f(four))),\ (five, g(f(five)))\}\\ f = &\{(one, g(1)),\ (two, g(2)),\ (three, g(3)),\ (four, g(4)),\ (five, g(5))\}\\ f = &\{(one, false),\ (two, true),\ (three, true),\ (four, false),\ (five, true)\}. \end{aligned}$$

In other words $(g \circ f)(two) = g(f(two)) = g(2) = true$ and $(g \circ f)(four) = false$.

Finite sets of a homogeneous nature, such as integers (represented by `int` in C++) allow a simple representation of functions in C++ using the STL data type `map`. The template class `map<A,B>` associates pairs of type `A` and `B`. Implement the examples above using the `map` data type. Implement function composition for two arbitrary (compatible) `map`s.

Problem 6. The spiral map (Problems and Solutions in Scientific Computing, page 79) is a 1 to 1 map (i.e. invertible) which maps $\mathbb{Z}^2 \to \mathbb{Z}$. Write a C++ program using the `map` class and `pair` that stores the elements of the map as

```
map<int,pair<int,int> >
```

Problem 7. The group table for a group with three elements `a,b,e` is given by

$*$	e	a	b
e	e	a	b
a	a	b	e
b	b	e	a

where `e` is the neutral element (identity) of the group. Write a C++ program that implements this group table using the `map` class and the `string` class. For example, the user should enter `a*b` and the output should be `e`.

Problem 8. The STL set class is a template class. Thus we may construct for example `set<int>`, `set<double>`, and `set<string>`. In mathematics sets are not required to be of a homogeneous type. We should be able to store for example both integers and strings in a mathematical set. Implement a class `AnyData` such that `set<AnyData>` can contain elements of any type. For example, it should be possible to do the following:

```
set<AnyData> s;
s.insert(2);
s.insert(string("string"));
s.insert(3.5);
set.insert(2);
```

Problem 9. Let $\mathbb{C}$ be the complex plane. Let $c \in \mathbb{C}$. The *Mandelbrot set* M is defined as

$$M := \{\, c \in \mathbb{C} : c,\, c^2 + c,\, (c^2 + c)^2 + c, \ldots, \not\to \infty \,\}.$$

To find the Mandelbrot set we study the recursion relation

$$z_{t+1} = z_t^2 + c, \qquad t = 0, 1, 2, \ldots$$

with the initial value $z_0 = 0$ and whether z_t escapes to infity. For example $c = 0$ and $c = 1/4 + i/4$ belong to the Mandelbrot set. The point $c = 1/2$ does not belong to the Mandelbrot set. Write a C++ program using the complex class of STL to find the Mandelbrot set. The output should be written to a file `Mandel.pnm` (portable anymap utilities). This file can then be used to display the fractal.

Problem 10. The *spiral map* is a $1 : 1$ map from $\mathbb{Z} \to \mathbb{Z}^2$. Write a C++ program using the `map` class and `pair` that stores the elements of the map as

```
map<int,pair<int,int> >
```

Problem 11. A priority queue is a type of queue that assigns a priority to every element that it stores. New elements are added to the queue using the `push()` function. Thus it is a set for which two operations are defined:
1) Adding an item (using `push()`)
2) Extracting the item that has the highest priority using `top()` and `pop()`.

We may think of a priority queue as a set of tasks with priorities. At any time a new task can be added. A task can also be removed from the priority queue, but this can only be the one with the highest priority. If this highest priority is shared by more than one task, we do not care which one is taken. Write a C++ program that uses the priority queue from STL. Apply it to floating point numbers so that bigger numbers get a higher priority. Apply it to strings so that strings lexicographicaly higher get a higher priority (case sensitive).

Problem 12. The ancient puzzle of the Tower of Hanoi consists of a number of wooden disks mounted on three poles, which are in turn attached to a baseboard. The disks each have different diameters and a hole in the middle large enough for the poles to pass through. At the beginning all disks are on the left pole with the smallest at the top, the second smallest one down etc. The object of the puzzle is to move all the disks over to the right pole, one at the time, so that they end up in the original order on that pole. One uses the middle pole as a temporary resting place for the disks. However it is allowed for a larger disk to be on top of a smaller one. For example if we have three disks then the moves are

```
move disk A from pole 1 to 3
move disk B from pole 1 to 2
move disk A from pole 3 to 2
move disk C from pole 1 to 3
move disk A from pole 2 to 1
move disk B from pole 2 to 3
move disk A from pole 1 to 3
total number of moves: 7
```

(i) Write a C++ program using recursion to implement the Tower of Hanoi.
(ii) Write a C++ program using the `stack` class of the standard template libraray to implement the Tower of Hanoi.

Problem 13. Using the `Verylong` class of SymbolicC++ and the `complex` class (of STL) that finds positive integer solutions (a, b, c) of the equation

$$c = (a + bi)^3 - 107i$$

where $i^2 = -1$.

Problem 14. Let $i = \sqrt{-1}$. Calculate i^i. Use the `complex` class of the standard template library of C++ to calculate i^i. Discuss.

Chapter 14

Particle Swarm Optimization

Problem 1. Particle Swarm Optimization (PSO) is based on the behavior of a colony or swarm of insects, such as ants, termites, bees, and wasps; a flock of birds; or a school of fish. The particle swarm optimization algorithm mimics the behavior of these social organisms. The word particle denotes, for example, a bee in a colony or a bird in a flock. Each individual or particle in a swarm behaves in a distributed way using its own intelligence and the collective or group intelligence of the swarm. As such, if one particle discovers a good path to food, the rest of the swarm will also be able to follow the good path instantly even if their location is far away in the swarm. Optimization methods based on swarm intelligence are called behaviorally inspired algorithms as opposed to the genetic algorithms, which are called evolution-based procedures. The PSO algorithm was originally proposed by Kennedy and Eberhart in 1995.

In the context of multivariable optimization, the swarm is assumed to be of specified or fixed size with each particle located initially at random locations in the multidimensional design space. Each particle is assumed to have two characteristics: a position and a velocity. Each particle wanders around in the design space and remembers the best position (objective function value) it has discovered. The particles communicate information or good positions to each other and adjust their individual positions and velocities based on the information received on the good positions.
The PSO is developed based on the following model:

1. When one particle locates an extremum point of the objective function, it

instantaneously transmits the information to all other particules.
2. All other particles gravitate to the extremum point of the objective function,

but not directly.
3. There is a component of each particle's own independent thinking as well as

its past memory.
Thus the model simulates a random search in the design space for the extremum

points of the objective function. As such, gradually over many iterations, the particles go to the target (the extremum point of the objective function).
The algorithm for determining the maximum of a function $f(\mathbf{x})$ (with $\mathbf{x}$ an n

dimensional vector) is as follows:

1) Initialize the number of particles N, the search intervals for each dimension (a_i, b_i), $i = 1, \ldots, n$ (n being the dimension of the search space), the search precision for each dimension ϵ_i, the maximum number of iterations i_{max}.
Initialize the positions of the particles $\mathbf{x}_j(0) = rand()$, $j = 1, \ldots, N$ randomly in the search domain.
Initialize the speeds of the particles $\mathbf{v}_j(0) = 0$, $j = 1, \ldots, N$.
Initialize the individual best positions of the particles $\mathbf{x}_{best,j}(0) = \mathbf{x}_j(0)$, $j = 1, \ldots, N$.
Initialize the iteration count $k = 0$.

2) Check the stop conditions: the diameter of the swarm in each dimension is less then the dimension's precision ϵ_i, or the maximum number of iterations i_{max} was reached. If yes then terminate else continue to step 3).

3) Calculate the values of the function $f(\mathbf{x})$ in the current positions of the particles, $f(\mathbf{x}_j(k))$, $j = 1, \ldots, N$. Update the values of the best individual points for each particle $\mathbf{x}_{best,j}(k)$, $j = 1, \ldots, N$ and the value of the global best point $\mathbf{x}_{best}(k)$. Continue to 4).

4) Update the particles' speeds by applying the formula:

$$\mathbf{v}_j(k) = \theta(k)\mathbf{v}_j(k-1) + c_1 r_1 \left[\mathbf{x}_{best,j}(k) - \mathbf{x}_j(k-1)\right] + c_2 r_2 \left[\mathbf{x}_{best}(k) - \mathbf{x}_j(k-1)\right]$$

where $j = 1, \ldots, N$ and r_1 and r_2 are random number between 0 and 1. The parameters c_1 and c_2 have usually the value 2 so that the particles would overfly the target about half of the time. $\theta(k)$ is the inertia weight dependent of the iteration count according to the formula:

$$\theta(k) = \theta_{max} - \left(\frac{\theta_{max} - \theta_{min}}{i_{max}}\right) k$$

where θ_{max} is a maximum value and θ_{min} is a minimum value, typically $\theta_{max} = 0.9$ and $\theta_{min} = 0.4$. Continue to 5).
5) Update the particles' positions by applying the formula:

$$\mathbf{x}_j(k) = \mathbf{x}_j(k-1) + \mathbf{v}_j(k), \quad j = 1, \ldots, N$$

6) Increment the iteration count k and go to 2).

Determine the maximum of the function

$$f(x) = -x^2 + 2x + 11, \ -2 \leq x \leq 2$$

by applying the PSO (Particle Swarm Optimization) method. The required precision is 10^{-4}.

Problem 2. Minimize

$$f(x_1, x_2) = x_1^2 + x_2^2 - 2x_1 - 4x_2$$

subject to the constraints

$$x_1 + 4x_2 - 5 \leq 0, \quad 2x_1 + 3x_2 - 6 \leq 0, \quad x_1 \geq 0, \quad x_2 \geq 0$$

by applying the Particle Swarm Optimization (PSO) method. The required precision is 10^{-3}.

Problem 3. Find the global minimum of the De Jong's function (or sphere model)

$$f(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{x}_i^2, \ n \geq 2$$

by applying the Particle Swarm Optimization (PSO) method. The required precision is 10^{-3}.

Problem 4. Differntial evolution (DE) is a population-based optimization method that attacks the starting point problem by sampling the objective function at multiple, randomly chosen initial points. At initialization a vector population of dimension N_p is generated such that the allowed parameter region is entirely covered. Each vector is indexed with a number from 0 to $N_p - 1$ for bookkeeping because each of them has to enter a competition. Like other Evolutionary Strategy population-based methods, DE generates new points that are perturbations of existing points, but unlike other Evolutionary Strategy methods, DE perturbs population vectors with the scaled difference of two randomly

selected population vectors to produce the trial vectors. Assume that we produce the trial vector with index 0, $\mathbf{u}_0$. DE selects randomly two distinct vectors $\mathbf{x}_{r1}$ and $\mathbf{x}_{r2}$ from the population and adds the scaled perturbation $\mathbf{x}_{r1} - \mathbf{x}_{r2}$ to a third vector also randomly selected from the population $\mathbf{x}_{r3}$, distinct from the first two vectors. The procedure is repeated in order to generate all the set of trial vectors $\mathbf{u}_0$, $\mathbf{u}_1$, ..., $\mathbf{u}_{N_p}$. In the selection stage, each trial vector competes against the vector in the population vectors with the same index. The vector with the lower objective function value is selected as a member of the next generation. The survivors of the N_p pairwise competitions become parents for the next generation in the evolutionary cycle. The evolutionary cycles are repeated until a termination criteria is meet, for example the diameter of the population becomes less than a small limit value ϵ or a maximum number of population generations were generared.
The more detailed algorithm for determining the global minimum of a function

$f(\mathbf{x})$ (with $\mathbf{x}$ an n dimensional vector) is as follows:
1) *Initialization* - Initialize the number of vectors N_p, the search intervals for

each dimension (a_i, b_i), $i = 1, \ldots, n$, the precision for the termination criteria ϵ, the maximum number of iterations i_{max}, the scale factor $F \in (0, 1+)$, the crossover probability $C_r \in [0, 1)$. Initialize the positions of the particles $\mathbf{x}_{i,j}^{(0)} = a_j + rand_{i,j}()(b_j - a_j)$, $i = 1, \ldots, N_p$ $j = 1, \ldots, n$ randomly in the search domain (the random number generator, $rand()$, returns a uniformly distributed random number from within the range $[0, 1)$). Initialize the iteration count $k = 0$.
2) *Mutation* - differential mutation adds a scaled, randomly sampled, vector

difference to a third randomly sampled vector to create a mutant vector

$$\mathbf{v}_i^{(k)} = \mathbf{x}_{r3}^{(k)} + F(\mathbf{x}_{r1}^{(k)} - \mathbf{x}_{r2}^{(k)}) \;\; i = 1, \ldots N_p$$

The scale factor, $F \in (0, 1+)$, is a positive real number that controls the rate at which the population evolves. While there is no upper limit on F, effective values are seldom greater than 1. The base vector index, $r3$, can be determined in a variety of ways, but for now it is assumed to be a randomly chosen vector index that is different from the target vector index, i. Except for being distinct from each other and from both the base and target vector indices, the difference vector indices, $r1$ and $r2$, are also randomly selected once per mutant.
3) *Crossover* - to complement the differential mutation search strategy, DE also

employs uniform crossover. Sometimes referred to as discrete recombination, (dual) crossover builds trial vectors out of parameter values that have been copied from two different vectors. In particular, DE crosses each vector with a mutant vector:

$$\mathbf{u}_{i,j}^{(k)} = \begin{cases} \mathbf{v}_{i,j}^{(k)} & \text{if } rand()_{i,j} < C_r \text{ or } j = j_{rand} \\ \mathbf{x}_{i,j}^{(k)} & \text{otherwise} \end{cases} \quad i = 1, \ldots, N_p, \;\; j = 1, \ldots, n$$

The crossover probability, $C_r \in [0, 1)$, is a user-defined value that controls the fraction of parameter values that are copied from the mutant. To determine which source contributes a given parameter, uniform crossover compares C_r to the output of a uniform random number generator. If the random number is less than or equal to C_r, the trial vector component is inherited from the mutant $\mathbf{v}_i^{(k)}$; otherwise, the trial vector component is copied from the vector, $\mathbf{x}_i^{(k)}$. In addition, the trial vector component with randomly chosen index, j_{rand}, is taken from the mutant to ensure that the trial vector does not duplicate $\mathbf{x}_i^{(k)}$. Because of this additional demand, C_r only approximates the true probability, p_{C_r}, that a trial parameter will be inherited from the mutant.

4) *Selection* - if the trial vector, $\mathbf{u}_i^{(k)}$, has an equal or lower objective function value than that of its target vector, $\mathbf{x}_i^{(k)}$, it replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least one more generation:

$$\mathbf{x}_i^{(k)} = \begin{cases} \mathbf{u}_i^{(k)} & \text{if } f(\mathbf{u}_i^{(k)}) < f(\mathbf{x}_i^{(k)}) \\ \mathbf{x}_i^{(k)} & \text{otherwise} \end{cases} \quad i = 1, \ldots, N_p$$

By comparing each trial vector with the target vector from which it inherits parameters, DE more tightly integrates recombination and selection than do other Evolutionary Algorithms.

5) *Termination* - check the termination conditions: the diameter of the population is less then the preset precision ϵ, or the maximum preset number of generations i_{max} was reached. If yes then terminate, else increment the iteration count k and go to 2).

Determine the global minimum of the function

$$f(x, y) = 3(1-x)^2 e^{-(x^2+(y+1)^2)} - 10(\frac{x}{5} - x^3 - y^5)e^{-(x^2+y^2)} - \frac{1}{3}e^{-((x+1)^2+y^2)}$$

where

$$-4 \leq x \leq 4, \quad -4 \leq y \leq 4$$

by applying the DE method. The required precision is 10^{-4}.

Bibliography

Steeb W.-H., Hardy Y., Hardy A. and Stoop R.
Problems and Solutions in Scientific Computing with C++ and Java Simulations
World Scientific Publishing, Singapore (2004)

Steeb W.-H.
The Nonlinear Workbook: Chaos, Fractals, Cellular Automata, Neural Networks, Genetic Algorithm, Gene Expression Programming, Wavelets, Fuzzy Logic, fifth edition
World Scientific Publishing, Singapore 2011
ISBN 978-981-4335-77-5
`http://www.worldscibooks.com/chaos/8050.html`

Hardy Y., Kiat Shi Tan and Steeb W.-H.
Computer Algebra with SymbolicC++

World Scientific Publishing, Singapore 2008
ISBN-13: 978-981-283-360-0
`http://www.worldscibooks.com/mathematics/6966.html`

Steeb W.-H.,
Mathematical Tools in Signal Processing with C++ and Java Simulations
World Scientific Publishing, Singapore 2005
ISBN 981 256 500 0
`http://www.worldscibooks.com/engineering/5939.html`

Index