B

Aleksandr A. Samarskii
Evgenii S. Nikolaev

Numerical Methods for Grid Equations

Volume I
Direct Methods

Translated from the Russian
by Stephen G. Nash

1989 Birkhäuser Verlag
Basel · Boston · Berlin

Authors' address:
Aleksandr A. Samarskii
Evgenii S. Nikolaev
Department of Computational
Mathematics and Cybernetics
Moscow University
Moscow 117234
USSR

Originally published as
Metody resheniya setochnykh uravnenii
by Nauka, Moscow 1978.

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Samarskij, Aleksandr A.:
Numerical methods for grid equations / Aleksandr A. Samarskii
; Evgenii S. Nikolaev. Transl. from the Russ. by Stephen G.
Nash. – Basel ; Boston ; Berlin : Birkhäuser.
Einheitssacht.: Metody rešenija setočnych uravnenij <engl.>

NE: Nikolaev, Evgenij S.:

Vol. 1. Direct methods. – 1989

Softcover reprint of the hardcover 1st edition 1989
Typesetting and Layout: *mathScreen online*, CH-4056 Basel

ISBN-13: 978-3-0348-9972-7 e-ISBN-13: 978-3-0348-9272-8
DOI: 10.1007/978-3-0348-9272-8

Volume I
Table of Contents

Volume II: Table of Contents ix

Preface xvii

Introduction xxi

Chapter 1
Direct Methods for Solving Difference Equations 1

1.1 Grid equations. Basic concepts 1
1.1.1 Grids and grid functions 1
1.1.2 Difference derivatives and various difference identities 4
1.1.3 Grid and difference equations 8
1.1.4 The Cauchy problem and boundary-value problems for difference equations 13

1.2 The general theory of linear difference equations 17
1.2.1 Properties of the solutions of homogeneous equations 17
1.2.2 Theorems about the solutions of linear equations 20
1.2.3 The method of variation of parameters 21
1.2.4 Examples 26

1.3 The solution of linear equations with constant coefficients 30
1.3.1 The characteristic equation. The simple-roots case 30
1.3.2 The multiple-root case 31
1.3.3 Examples 35

1.4 Second-order equations with constant coefficients 38
1.4.1 The general solution of a homogeneous equation 38
1.4.2 The Chebyshev polynomials 41
1.4.3 The general solution of a non-homogeneous equation 44

1.5 Eigenvalue difference problems 48
1.5.1 A boundary-value problem of the first kind 48
1.5.2 A boundary-value problem of the second kind 50
1.5.3 A mixed boundary-value problem 52
1.5.4 A periodic boundary-value problem 54

Chapter 2
The Elimination Method 61

2.1 The elimination method for three-point equations 61
2.1.1 The algorithm 61
2.1.2 Two-sided elimination 65
2.1.3 Justification of the elimination method 66
2.1.4 Sample applications of the elimination method 70

2.2 Variants of the elimination method 73
2.2.1 The flow variant of the elimination method 73
2.2.2 The cyclic elimination metod 77
2.2.3 The elimination method for complicated systems 81
2.2.4 The non-monotonic elimination method 86

2.3 The elimination method for five-point equations 90
2.3.1 The monotone elimination algorithm 90
2.3.2 Justification of the method 92
2.3.3 A variant of non-monotonic elimination 95

2.4 The block-elimination method 97
2.4.1 Systems of vector equations 97
2.4.2 Elimination for three-point vector equations 101
2.4.3 Elimination for two-point vector equations 104
2.4.4 Orthogonal elimination for two-point vector equations 107
2.4.5 Elimination for three-point equations with constant coefficents 112

Chapter 3
The Cyclic Reduction Method 117

3.1 Boundary-value problems for three-point vector equations 117
3.1.1 Statement of the boundary-value problems 117
3.1.2 A boundary-value problem of the first kind 119
3.1.3 Other boundary-value problems for difference equations 122
3.1.4 A high-accuracy Dirichlet difference problem 126

3.2 The cylic reduction method for a boundary-value problem of the first kind 128
3.2.1 The odd-even elimination process 128

3.2.2 Transformation of the right-hand side and inversion of the matrices 131
3.2.3 The algorithm for the method 136
3.2.4 The second algorithm of the method 139
3.3 Sample applications of the method 145
3.3.1 A Dirichlet difference problem for Poisson's equation in a rectangle 145
3.3.2 A high-accuracy Dirichlet difference problem 148
3.4 The cyclic reduction method for other boundary-value problems . 151
3.4.1 A boundary-value problem of the second kind 151
3.4.2 A periodic problem 157
3.4.3 A boundary-value problem of the third kind 161

Chapter 4
The Separation of Variables Method 171

4.1 The algorithm for the discrete Fourier transform 171
4.1.1 Statement of the problem 171
4.1.2 Expansion in sines and shifted sines 176
4.1.3 Expansion in cosines 185
4.1.4 Transforming a real-valued periodic grid function 188
4.1.5 Transforming a complex-valued periodic grid function 193
4.2 The solution of difference problems by the Fourier method 196
4.2.1 Eigenvalue difference problems for the Laplace operator in a rectangle 196
4.2.2 Poisson's equation in a rectangle; expansion in a double series 201
4.2.3 Expansion in a single series 205
4.3 The method of incomplete reduction 211
4.3.1 Combining the Fourier and reduction methods 211
4.3.2 The solution of boundary-value problems for Poisson's equation in a rectangle 219
4.3.3 A high-accuracy Dirichlet difference problem in a rectangle 223
4.4 The staircase algorithm and the reduction method for solving tridiagonal systems of equations 227
4.4.1 The staircase algorithm for the case of tridiagonal matrices with scalar elements 227
4.4.2 The staircase algorithm for the case of a block-tridiagonal matrix 230
4.4.3 Stability of the staircase algorithm 231
4.4.4 The reduction method for three-point scalar equations 236

Index 239

Volume II
Table of Contents

Chapter 5
The Mathematical Theory of Iterative Methods 1

5.1 Several results from functional analysis 1
5.1.1 Linear spaces 1
5.1.2 Operators in linear normed spaces 5
5.1.3 Operators in a Hilbert space 8
5.1.4 Functions of a bounded operator 14
5.1.5 Operators in a finite-dimensional space 15
5.1.6 The solubility of operator equations 18

5.2 Difference schemes as operator equations 21
5.2.1 Examples of grid-function spaces 21
5.2.2 Several difference identities 25
5.2.3 Bounds for the simplest difference operators 28
5.2.4 Lower bounds for certain difference operators 31
5.2.5 Upper bounds for difference operators 41
5.2.6 Difference schemes as operator equations in abstract spaces 42
5.2.7 Difference schemes for elliptic equations with constant coefficients 47
5.2.8 Equations with variable coefficients and with mixed derivatives 50

5.3 Basic concepts from the theory of iterative methods 55
5.3.1 The steady state method 55
5.3.2 Iterative schemes 56
5.3.3 Convergence and iteration counts 58
5.3.4 Classification of iterative methods 60

Chapter 6
Two-Level Iterative Methods 65

6.1 Choosing the iterative parameters 65
6.1.1 The initial family of iterative schemes 65
6.1.2 The problem for the error 66
6.1.3 The self-adjoint case 67

6.2 The Chebyshev two-level method 69
6.2.1 Construction of the set of iterative parameters 69
6.2.2 On the optimality of the *a priori* estimate 71
6.2.3 Sample choices for the operator D 72
6.2.4 On the computational stability of the method 75
6.2.5 Construction of the optimal sequence of iterative parameters 82

6.3 The simple iteration method 86
6.3.1 The choice of the iterative parameter 86
6.3.2 An estimate for the norm of the transformation operator .. 88

6.4 The non-self-adjoint case. The simple iteration method 90
6.4.1 Statement of the problem 90
6.4.2 Minimizing the norm of the transformation operator 91
6.4.3 Minimizing the norm of the resolving operator 98
6.4.4 The symmetrization method 104

6.5 Sample applications of the iterative methods 105
6.5.1 A Dirichlet difference problem for Poisson's equation in a rectangle 105
6.5.2 A Dirichlet difference problem for Poisson's equation in an arbitrary region 110
6.5.3 A Dirichlet difference problem for an elliptic equation with variable coefficients 116
6.5.4 A Dirichlet difference problem for an elliptic equation with mixed derivatives 122

Chapter 7
Three-Level Iterative Methods 125

7.1 An estimate of the convergence rate 125
7.1.1 The basic family of iterative schemes 125
7.1.2 An estimate for the norm of the error 126

7.2 The Chebyshev semi-iterative method 129
7.2.1 Formulas for the iterative parameters 129
7.2.2 Sample choices for the operator D 132
7.2.3 The algorithm of the method 132

7.3 The stationary three-level method 133
7.3.1 The choice of the iterative parameters 133
7.3.2 An estimate for the rate of convergence 134

7.4 The stability of two-level and three-level methods relative to *a priori* data 136
7.4.1 Statement of the problem 136
7.4.2 Estimates for the convergence rates of the methods 138

Chapter 8
Iterative Methods of Variational Type 145

8.1 Two-level gradient methods 145
8.1.1 The choice of the iterative parameters 145
8.1.2 A formula for the iterative parameters 147
8.1.3 An estimate of the convergence rate 149
8.1.4 Optimality of the estimate in the self-adjoint case 151
8.1.5 An asymptotic property of the gradient methods in the self-adjoint case 153

8.2 Examples of two-level gradient methods 156
8.2.1 The steepest-descent method 156
8.2.2 The minimal residual method 158
8.2.3 The minimal correction method 160
8.2.4 The minimal error method 161
8.2.5 A sample application of two-level methods 162

8.3 Three-level conjugate-direction methods 164
8.3.1 The choice of the iterative parameters. An estimate of the convergence rate 164
8.3.2 Formulas for the iterative parameters. The three-level iterative scheme 167
8.3.3 Variants of the computational formulas 172

8.4 Examples of the three-level methods 174
8.4.1 Special cases of the conjugate-direction methods 174
8.4.2 Locally optimal three-level methods 176

8.5 Accelerating the convergence of two-level methods in the self-adjoint case 181
8.5.1 An algorithm for the acceleration process 181
8.5.2 An estimate of the effectiveness 183
8.5.3 An example 184

Chapter 9
Triangular Iterative Methods 189

9.1 The Gauss-Seidel method 189
9.1.1 The iterative scheme for the method 189
9.1.2 Sample applications of the method 193
9.1.3 Sufficient conditions for convergence 196

9.2 The successive over-relaxation method 199
9.2.1 The iterative scheme. Sufficient conditions for covergence .. 199
9.2.2 The choice of the iterative parameter 200
9.2.3 An estimate of the spectral radius 204
9.2.4 A Dirichlet difference problem for Poisson's equation in a rectangle 207
9.2.5 A Dirichlet difference problem for an elliptic equation with variable coefficients 212

9.3 Triangular methods 215
9.3.1 The iterative scheme 215
9.3.2 An estimate of the convergence rate 217
9.3.3 The choice of the iterative parameter 219
9.3.4 An estimate for the convergence rates of the Gauss-Seidel and relaxation methods 220

Chapter 10
The Alternate-Triangular Method 225

10.1 The general theory of the method 225
10.1.1 The iterative scheme 225
10.1.2 Choice of the iterative parameters 228
10.1.3 A method for finding δ and Δ 231
10.1.4 A Dirichlet difference problem for Poisson's equation in a rectangle 233

10.2 Boundary-value difference problems for elliptic equations in a rectangle 241
10.2.1 A Dirichlet problem for an equation with variable coefficients 241
10.2.2 A modified alternate-triangular method 243
10.2.3 A comparison of the variants of the method 251
10.2.4 A boundary-value problem of the third kind 252
10.2.5 A Dirichlet difference problem for an equation with mixed derivatives 255

10.3 The alternate-triangular method for elliptic equations in arbitrary regions 258
10.3.1 The statement of the difference problem 258
10.3.2 The construction of an alternate-triangular method 259
10.3.3 A Dirichlet problem for Poisson's equation in an arbitrary region 264

Chapter 11
The Alternating-Directions Method 269

11.1 The alternating-directions method in the commutative case 269
11.1.1 The iterative scheme for the method 269
11.1.2 The choice of the parameters 271
11.1.3 A fractionally-linear transformation 273
11.1.4 The optimal set of parameters 276

11.2 Sample applications of the method 280
11.2.1 A Dirichlet difference problem for Poisson's equation in a rectangle 280
11.2.2 A boundary-value problem of the third kind for an elliptic equation with separable variables 285
11.2.3 A high-accuracy Dirichlet difference problem 289

11.3 The alternating-directions method in the general case 294
11.3.1 The case of non-commuting operators 294
11.3.2 A Dirichlet difference problem for an elliptic equation with variable coefficients 297

Chapter 12
Methods for Solving Equations with Indefinite and Singular Operators 303

12.1 Equations with real indefinite operators 303
12.1.1 The iterative scheme. The choice of the iterative parameters 303
12.1.2 Transforming the operator in the self-adjoint case 306
12.1.3 The iterative method with the Chebyshev parameters 309
12.1.4 Iterative methods of variational type 313
12.1.5 Examples 315

12.2 Equations with complex operators 317
12.2.1 The simple iteration method 317
12.2.2 The alternating-directions method 321

12.3 General iterative methods for equations with singular operators .. 326
12.3.1 Iterative schemes in the case of a non-singular operator B . 326
12.3.2 The minimum-residual iterative method 330
12.3.3 A method with the Chebyshev parameters 333

12.4 Special methods 339
12.4.1 A Neumann difference problem for Poisson's equation in a rectangle 339
12.4.2 A direct method for the Neumann problem 343
12.4.3 Iterative schemes with a singular operator B 346

Chapter 13
Iterative Methods for Solving Non-Linear Equations 351

13.1 Iterative methods. The general theory 351
13.1.1 The simple iteration method for equations with a monotone operator 351
13.1.2 Iterative methods for the case of a differentiable operator .. 354
13.1.3 The Newton-Kantorovich method 358
13.1.4 Two-stage iterative methods 362
13.1.5 Other iterative methods 365

13.2 Methods for solving non-linear difference schemes 368
13.2.1 A difference scheme for a one-dimensional elliptic quasi-linear equation 368
13.2.2 The simple iteration method 377
13.2.3 Iterative methods for quasi-linear elliptic difference equations in a rectangle 379
13.2.4 Iterative methods for weakly-nonlinear equations 385

Chapter 14
Example Solutions of Elliptic Grid Equations 389

14.1 Methods for constructing implicit iterative schemes 389
14.1.1 The regularizer principle in the general theory of iterative methods 389
14.1.2 Iterative schemes with a factored operator 393
14.1.3 A method for implicity inverting the operator B (a two-stage method) 399

14.2 Examples of solving elliptic boundary-value problems 401
14.2.1 Direct and iterative methods 401
14.2.2 A high-accuracy Dirichlet difference problem in the multi-dimensional case 407
14.2.3 A boundary-value problem of the third kind for an equation with mixed derivatives in a rectangle 409
14.2.4 Iterative methods for solving a difference problem 414

14.3 Systems of elliptic equations 423
14.3.1 A Dirichlet problem for systems of elliptic equations in a p-dimensional parallelipiped 423
14.3.2 A system of equations from elasticity theory 428

14.4 Methods for solving elliptic equations in irregular regions 431
14.4.1 Difference problems in regions of complex form, and methods for their solution 431
14.4.2 Decomposition of regions 433
14.4.3 An algorithm for the domain decomposition method 438
14.4.4 The method of domain augmentation to a rectangle 442

Chapter 15
Methods for Solving Elliptic Equations in Curvilinear Orthogonal Coordinates 447

15.1 Posing boundary-value problems for differential equations 447
15.1.1 Elliptic equations in a cylindrical system of coordinates ... 447
15.1.2 Boundary-value problems for equations in a cylindrical coordinate system 450

15.2 The solution of difference problems in cylindrical coordinates 454
15.2.1 Difference schemes without mixed derivatives in the axially-symmetric case 454
15.2.2 Direct methods 459
15.2.3 The alternating-directions method 461
15.2.4 The solution of equations defined on the surface of a cylinder 465

15.3 Solution of difference problems in polar coordinate systems 470
15.3.1 Difference schemes for equations in a circle or a ring 470
15.3.2 The solubility of the boundary-value difference problems .. 473
15.3.3 The superposition principle for a problem in a circle 476
15.3.4 Direct methods for solving equations in a circle or a ring .. 478
15.3.5 The alternating-directions method 479
15.3.6 Solution of difference problems in a ring sector 483
15.3.7 The general variable-coefficients case 485

Appendices

Construction of the minimax polynomial 489

Bibliography 495

Translator's note 497

Index 499

Preface

The finite-difference solution of mathematical-physics differential equations is carried out in two stages: 1) the writing of the difference scheme (a difference approximation to the differential equation on a grid), 2) the computer solution of the difference equations, which are written in the form of a high-order system of linear algebraic equations of special form (ill-conditioned, band-structured). Application of general linear algebra methods is not always appropriate for such systems because of the need to store a large volume of information, as well as because of the large amount of work required by these methods. For the solution of difference equations, special methods have been developed which, in one way or another, take into account special features of the problem, and which allow the solution to be found using less work than via the general methods.

This work is an extension of the book *Difference Methods for the Solution of Elliptic Equations* by A.A. Samarskii and V.B. Andreev which considered a whole set of questions connected with difference approximations, the construction of difference operators, and estimation of the convergence rate of difference schemes for typical elliptic boundary-value problems.

Here we consider only solution methods for difference equations. The book in fact consists of two volumes. The first volume (Chapters 1–4) deals with the application of direct methods to the solution of difference equations, the second volume (Chapters 5–15) considers the theory of iterative methods for solving general grid equations and their application to difference equations. The special form of the difference equations plays an important role when using direct methods. For solving one-dimensional 3-point equations, various forms of the elimination method are considered (monotone, non-monotone, cyclic, flow elimination, and others).

Chapters 3 and 4 present up-to-date efficient direct methods for solving Poisson difference equations in a rectangle with various boundary conditions.

These are the cyclic reduction method, the method of separation of variables (using the fast Fourier transform), and also combined methods.

For the study of iterative methods, the iterative method is considered as an operator-difference scheme, as was described in the books *An Introduction to the Theory of Difference Schemes* (1971) and *The Theory of Difference Schemes* (1977) by A.A. Samarskii. This concept allows us to present the theory of iterative methods as a part of the general stability theory for operator-difference schemes, without any assumptions about the structure of the matrix system (see also A.A. Samarskii and A.V. Gulin *Stability of Difference Schemes* (1973)). Writing the iterative schemes in a canonical form not only allows us to isolate the operator responsible for the convergence of the iteration, but also allows us to compare different iterative methods. Much attention is given to the study of the convergence rate of the iteration and to the choice of the optimal parameters, for which the convergence rate is maximal. The availability of convergence rate estimates, and also a study of the character of the computational stability, allows us to compare various iterative methods and make a choice in concrete situations. Although the reader is undoubtedly familiar with the basic theory of difference schemes and elementary functional analysis, Chapter 5 presents the basic mathematical apparatus from the theory of iterative schemes and shows how the difference approximations for elliptic equations lead to operator equations of the first kind $Au = f$ where the operators A are in a Hilbert space of grid functions.

The succeeding chapters investigate two-level iterative schemes with Chebyshev parameters (a stable method); three-level schemes; iterative methods of variable type (the steepest-descent, minimum residual, minimum correction, and conjugate-gradient methods, etc.); iterative methods for non-self-adjoint equations and for indefinite and singular operators; alternating direction methods; "triangular" methods (where a triangular matrix is inverted in order to define a new iterate) such as the Seidel method, successive over-relaxation, and others; iterative methods for solving nonlinear difference equations, for solving boundary-value difference problems for elliptic equations in curvilinear systems of coordinates, etc.

A fundamental place in the book is occupied by the universal alternate-triangular method, which was proposed and developed between 1964 and 1977, and which is particularly effective for solving the Dirichlet problem for Poisson's equations in an arbitrary region and the Dirichlet problem for the equation $div\,(k\,grad\,u) = -f(x)$, $x = (x_1, x_2)$ with a rapidly changing coefficient $k(x)$.

The book shows how to pass from the general theory to concrete problems, and mentions a great number of iterative algorithms for solving difference equations for elliptic equations and systems of equations. Estimates are given for the number of iterations required, and comparisons are made of various methods. In particular, it is shown that for the simplest problems, di-

rect methods are more economical that the alternating directions method. It should be emphasized that the linear algebra problems that arise in practice are constantly becoming more and more complex, and that they require new methods of solution as well as widening of the field of application of older methods.

For the writing of this book, the authors used their own lecture notes presented between 1961–1977 at the mathematical-mechanics faculty and the computational mathematics and cybernetics faculty of the Moscow State University, and also materials from their own published works.

The authors would like to take this opportunity to express gratitude to V.B. Andreev, I.V. Fryazinov, M.I. Bakirova, A.B. Kucherov, and I.E. Kaporin for their many useful comments on the text.

The authors also thank T.N. Galishnikova, A.A. Golubeva, and especially V.M. Marchenko for their help in preparing the manuscript for publication.

A.A. Samarskii, E.S. Nikolaev

Introduction

The application of various numerical methods (difference, variational-difference, projected-difference methods including the finite-element method) to the solution of differential equations leads to a system of linear algebraic equations of special form, the difference equations. This system possesses the following special features: 1) it is of high order, equal to the number of grid points; 2) the system is ill-conditioned (the ratio of the largest to the smallest eigenvalue is great; for the Laplace difference operator this ratio is inversely proportional to the square of the grid spacing); 3) the matrix of the system is sparse — only a few elements in each row are non-zero, and this is independent of the number of nodes; 4) the non-zero elements of the matrix are distributed in a special way — the matrix is banded.

In approximating integral and integral-differential equations on a grid, we obtain a system of equations relating to a function defined on the grid (the grid function). These equations are naturally called the grid equations:

$$\sum_{\xi\in\omega} a(x,\xi)y(\xi) = f(x), \quad x \in \omega \tag{1}$$

where the sum is taken over all points of the grid ω, i.e., over a discrete set of points. The matrix $(a(x,\xi))$ of the grid equation is, in the general case, full. If the grid points are enumerated, then the grid equation can be written in the form

$$\sum_{j=1}^{N} a_{ij}y_j = f_i, \quad i = 1,2,\ldots,N, \tag{2}$$

where i,j are the indices of the grid nodes, and N is the total number of nodes. It is obvious how to reverse the path of this reasoning. Thus, the

linear grid equation is a system of linear algebraic equations and, conversely, any linear system of algebraic equations can be expressed as a linear grid equation relative to a grid function defined on some grid with the number of nodes equal to the order of the system. We remark that variational methods (Ritz, Galerkin, etc.) for numerically solving differential equations usually lead to dense systems.

The difference equation is a particular case of the grid equation when the matrix (a_{ij}) is sparse. So, for example (2) is a difference equation of m^{th} order if, in row i, there are only $m+1$ non-zero elements $a_{ij}(j = i, i+1, \ldots, i+m)$.

From the above remarks it is clear that the solution of grid and, in particular, difference equations is a problem in linear algebra.

* * *

There exist many different numerical methods for solving linear algebra problems, and research continually leads to re-evaluations and reworkings of these methods, as well as to the discovery of new methods. Many of the existing methods have a specific set of problems to which they are best-suited. Thus, in order to solve a given problem on a computer, there arises the problem of choosing one method from a set of admissible methods for solving the given problem. This method must, obviously, display the best characteristics (or, as one would like to say, be an optimal method) so that the computer time is a minimum (or the number of arithmetic and logical operations for finding the solution is a minimum), and so that the computation is stable (i.e. stable in relation to the rounding error), etc.

It is natural to require that any computational algorithm in principle allow the solution to be obtained to any pre-specified accuracy $\epsilon > 0$ after a finite number of operations $Q(\epsilon)$. An infinite set of algorithms satisfies this requirement, so the algorithm should be found which minimizes $Q(\epsilon)$ for any $\epsilon > 0$. Such an algorithm is called economical. Finally, the search for an "optimal" or "best" method in general leads to a set of known (but not always admissible) methods, and so the term "optimal algorithm" has a limited and conditional meaning.

* * *

The problem for the theory of numerical methods consists in finding optimal algorithms for a given class of problems, and in establishing a hierarchy of methods. The notion of the best algorithm depends on the goal of the computation.

There are two ways of defining what is meant by a best method:

[a] require that it solve one concrete system of equations $Au = f$ with matrix $A = (a_{ij})$

[b] require that it solve several variants of some problem, for example, the equations $Au = f$ with several right-hand sides f.

For multi-variant computations, it is possible to reduce the average number of operations $\bar{Q}(\epsilon)$ for one variant if some quantities are saved and not computed anew for each variant (for example, preserving the inverse of the matrix).

From this it is clear that the choice of an algorithm must depend on the type of computation (single-variant or multi-variant), on the possibility of saving sufficient information in the computer memory (which is to some degree dependent on the type of computer), as well as on the order of the system of equations. For theoretical estimates the computational work is usually estimated by the number of arithmetic operations required to find the solution to a given accuracy; for this question, the parameters of the computer are as a rule not considered.

The stormy development in recent years of numerical methods for solving difference equations approximating elliptic differential equations, and the appearance of new economical algorithms, has necessitated the reconsideration of the applicability of existing methods.

* * *

The contents of this book to a considerable degree hinge on the need to give effective methods for solving difference equations corresponding to boundary-value problems for second-order elliptic equations. The classifiction of the boundary-value difference problems can be carried out according to the following rules:

[1] the form of the differential operator L in the equation

$$Lu = f(x), \quad x = (x_1, \ldots, x_p) \in G; \tag{3}$$

[2] the form of the region G in which the solution is to be found;

[3] the type of boundary conditions on the boundary Γ of the region G;

[4] the grid $\bar{\omega}$ in the region $\bar{G} = G + \Gamma$ and the difference scheme

$$\Lambda y = -\varphi(x), \quad x \in \omega, \tag{4}$$

i.e., the form of the difference operator Λ.

Some examples of second-order elliptic operators are

$$Lu = \Delta u = \sum_{\alpha=1}^{p} \frac{\partial^2 u}{\partial x_\alpha^2} \quad \text{the Laplace operator,} \tag{5}$$

$$Lu = \sum_{\alpha,\beta=1}^{p} \frac{\partial}{\partial x_\alpha}\left(k_{\alpha\beta}(x)\frac{\partial u}{\partial x_\beta}\right) - q(x)u, \tag{6}$$

where the coefficients $k_{\alpha\beta}(x)$ at each point $x = (x_1, \ldots, x_p)$ satisfy a strong ellipticity condition

$$c_1 \sum_{\alpha=1}^{p} \xi_\alpha^2 \le \sum_{\alpha,\beta=1}^{p} k_{\alpha\beta}(x)\xi_\alpha\xi_\beta \le c_2 \sum_{\alpha=1}^{p} \xi_\alpha^2, \qquad c_1, c_2 = \text{constant} > 0, \tag{7}$$

where $\xi = (\xi_1, \ldots, \xi_p)$ is an arbitrary vector. If $u(x) = (u^1(x), u^2(x), \ldots, u^m(x))$ is a vector function, then (3) is a system of equations and

$$(Lu)^i = \sum_{j=1}^{m} \sum_{\alpha,\beta=1}^{p} \frac{\partial}{\partial x_\alpha}\left(k_{\alpha\beta}^{ij}\frac{\partial u^j}{\partial x_\beta}\right), \quad i = 1, 2, \ldots, m,$$

where the condition of strong ellipticity has the form

$$c_1 \sum_{i=1}^{m}\sum_{\alpha=1}^{p} \left(\xi_\alpha^i\right)^2 \le \sum_{i,j=1}^{m}\sum_{\alpha,\beta=1}^{p} k_{\alpha\beta}^{ij}(x)\xi_\alpha^i\xi_\beta^i \le c_2 \sum_{i=1}^{m}\sum_{\alpha=1}^{p} \left(\xi_\alpha^i\right)^2, \quad c_1, c_2 = \text{constant} > 0.$$

* * *

The shape of the region strongly affects the properties of the difference matrix. We will consider separately regions where separation of variables can be used to solve the equation $Lu = 0$ with homogeneous boundary conditions. So, for example, separation of variables can be used for the Laplace equation in Euclidean coordinates (x_1, x_2)

$$Lu = \Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}$$

when G is a rectangle. A difference scheme on a rectangular grid (for example, the "cross" scheme) possesses an analogous property; in this case, the grid can be non-uniform in either direction.

In order to compare various methods for solving systems of algebraic equations, we will use as a *standard* or *model* problem the following boundary-value difference problem:

Poisson's equation, on a square region, with boundary conditions of the first kind on a square grid with steps $h_1 = h$ and $h_2 = h$ along x_1 and x_2, and with the five-point difference operator Λ.

The second group of boundary-value difference problems corresponds to the following data: L is an operator with variable coefficients of the form (6): a) without mixed derivatives, b) with mixed derivatives, and the region

$$G = \{0 \leq x_\alpha \leq l_\alpha, \quad \alpha = 1, 2\}$$

is a rectangle (a parallelipiped for $p \geq 3$).

The third group of problems has a region of complex form, where L is any Laplace operator or any operator of general form; here the degree of complexity of the problem is determined by the shape of the region, and by the choice of the grid and the difference operator near the boundary.

For the second and third groups of problems the difference operator is usually chosen so as to preserve the basic properties of the underlying problem (self-adjointness, definiteness, etc.) and in order to preserve the necessary order of approximation in relation to the grid spacing.

* * *

Direct and iterative methods are used to solve elliptic difference problems.

Direct methods are generally applied in the multidimensional case to problems from the first group (L is the Laplace operator, G is a rectangle for $p = 2$ and a parallelipiped for $p \geq 3$, Λ is a five- or a nine-point difference scheme for $p = 2$). For one-dimensional problems, where the difference equation is of second order (the matrix is tridiagonal), and where the equation may have variable coefficients, the elimination method (a variant of a method of Gauss, see Chapter 2) is used. There are a number of variants of the elimination method: monotone elimination, non-monotone elimination, flow elimination, cyclic elimination, etc. (see Chapter 2). For two-dimensional problems from the first group (see above), the following are effective: the cyclic reduction method (Chapter 3), the method of separation of variables using the fast Fourier transform (FFT), and also the combined method using incomplete reduction with the FFT (Chapter 4). In all cases, the elimination method is used to solve the second-order difference equation along each of the directions.

The direct methods indicated above for solving the Dirichlet difference problem for Poisson's equation in a rectangle

$$\bar{G} = (0 \leq x_\alpha \leq l_\alpha, \quad \alpha = 1, 2)$$

on the grid

$$\bar{\omega} = \{(i_1 h_1, i_2 h_2), i_\alpha = 0, 1 \ldots, N_\alpha, h_\alpha = l_\alpha / N_\alpha, \alpha = 1, 2\}$$

require $Q = O(N_1 N_2 \log_2 N_2)$ arithmetic operations, where $N_2 = 2^n, n > 0$ is an integer.

Direct methods are used for a very special class of problems.

* * *

Elliptic difference problems in the case of general operators L for complex regions are generally solved using iterative methods.

Grid equations can be treated as operator equations of first order

$$Au = f \tag{8}$$

with operators defined on spaces H of grid functions. In the space H, there is an inner product (,) and energy norm

$$\| u \|_D = \sqrt{(Du, u)}, \quad D = D^* > 0, \quad D : H \to H$$

where D is some linear operator in H.

Iterative methods for solving the operator equation $Au = f$ can be treated as operator-difference equations (differenced according to fictitious time or according to the index-number of the iteration) with operators in the Hilbert space H. If the new iteration y_{k+1} is computed using the m previous iterations

$$y_k, y_{k-1}, \cdots, y_{k-m+1}$$

then the iterative method (scheme) is called an $m+1$-level (m-step) method. From this, the analogy between iterative schemes and difference schemes for non-stationary problems is clear. In fact, it follows that the theory of iterative methods is a special case of the general stability theory for operator-difference schemes. We will limit our attention to two-level and, to a lesser extent, three-level schemes. Going to multi-level schemes gives no special advantages (since this also follows from the general stability theory, see [10]).

An important role is played by the writing of iterative methods in a special (canonical) form that allows us to separate out the operator (stabilizer) responsible for the stability and convergence of the iterations, and compare different iterative methods having the same form.

Any two-level (one-step) iterative method can be written in the following canonical form:

$$B \frac{y_{k+1} - y_k}{\tau_{k+1}} + A y_k = f, \quad k = 0, 1, \ldots, \quad y_0 \in H, \tag{9}$$

where $B : H \to H$ is a linear operator having inverse B^{-1}, $\tau_1, \tau_2, \ldots$ are iterative parameters, k is the iteration number, and y_k is the k^{th} iterative approximation. In the general case $B = B_{k+1}$ depends on k. In the general theory we will assume that B does not depend on k.

The parameters $\{\tau_k\}$ and operator B are arbitrary, and they should be chosen to minimize the number of iterations n required to insure that the solution y_n to equation (9) approximates in H_D the exact solution u to the equation $Au = f$ with accuracy $\epsilon > 0$:

$$\| y_n - u \|_D \leq \epsilon \| y_0 - u \|_D \ . \tag{10}$$

For the general theory presented in the book, the iterative methods do not require any assumptions on the structure of the operator A (the matrix (a_{ij})). All that is required are properties of the general form:

$$A = A^* > 0, \quad B = B^* > 0, \quad \gamma_1 B \leq A \leq \gamma_2 B, \quad \gamma_1 > 0. \tag{11}$$

The operator inequalities signify that there exist energy-equivalence constants γ_1, γ_2 for the operators A and B or bounds on the spectrum of the operator A in the space H_B (γ_1 and γ_2 are the smallest and largest eigenvalues, respectively, for the generalized eigenvalue problem: $Av = \lambda Bv$).

* * *

The solution $\tau_1, \tau_2, \ldots, \tau_n$ of the above minimization problem

$$\min_{\tau_1, \ldots, \tau_n} n_0(\epsilon)$$

for fixed γ_1, γ_2 and fixed B in the case $D = AB^{-1}A$ can be expressed via the zeroes of the n^{th} order Chebyshev polynomial (the Chebyshev iterative method). For these optimal values $\tau_1, \tau_2, \ldots, \tau_n$ and for any given $\epsilon > 0$, the number of iterations n required by the scheme (9) can be estimated using

$$n \geq \frac{\ln(2/\epsilon)}{\ln((1+\sqrt{\xi})/(1-\sqrt{\xi}))}$$

or

$$n \geq n_0(\epsilon) = \frac{\ln(2/\epsilon)}{2\sqrt{\xi}}, \quad \xi = \gamma_1/\gamma_2$$

and the following inequality is satisfied:

$$\| Ay_n - f \|_{B^{-1}} \leq \epsilon \| Ay_0 - f \|_{B^{-1}} \ .$$

The computational stability of the Chebyshev method is valid for a special ordering of the zeroes of the Chebyshev polynomial and the parameters $\tau_1^*, \tau_2^*, \ldots, \tau_n^*$; this ordering is indicated in Chapter 6.

For $B = E$ (E is the identity operator) the method (9) is called explicit, and for $B \neq E$, implicit. If the parameter τ_k is chosen as a constant

$$\tau_k = \tau_0 = 2/(\gamma_1 + \gamma_2), \quad k = 1, 2, \ldots, n$$

then we obtain the implicit simple-iteration scheme, for which

$$n \geq n_0(\epsilon) = \ln \frac{1}{\epsilon} \Big/ (2\xi).$$

The operator B (stabilizer) is chosen using an efficiency condition, i.e., in order to minimize the computational work to solve $Bv = F$ for a given right-hand side F, and, as was already mentioned, to minimize the number of iterations $n_0(\epsilon)$.

We will assume that we can efficiently solve the problem $Rv = f$ using $Q_R(\epsilon)$ operations, where

$$R : H \to H, \quad R = R^* > 0, \quad c_1 R \leq A \leq c_2 R, \quad c_1 > 0. \tag{12}$$

Then it is possible to set $B = R$ and find the solution to the problem $Au = f$ using the scheme (9) with parameters $\{\tau_k^*\}$ and with $\gamma_1 = c_1, \gamma_2 = c_2$ using

$$Q_A(\epsilon) \approx \frac{1}{2}\sqrt{c_2/c_1}\ln(2/\epsilon) Q_R(\epsilon)$$

operations.

If, for example, L is a general operator and G is a rectangle, then R can be taken as the five-point Laplace difference operator and the equation $Rv = f$ can be solved using a direct method.

It can be remarked that, if it is more advantageous to solve the equation $Rv = f$ iteratively, then $B \neq R$ and B is not written out in explicit form, but realized as the result of an iterative procedure.

* * *

The well-known Seidel and successive over-relaxation (SOR) methods are implicit and correspond to triangular matrices (operators) B. The convergence of these methods is proved using the general theory of difference schemes (see A.A. Samarskii, *Theory of Difference Schemes,* Moscow, 1977, or A.A. Samarskii and A.V. Gulin, *Stability of Difference Schemes*, Moscow, 1973).

However, for Seidel's method and for SOR, B is not self-adjoint, and it is not possible to use the Chebyshev method (9) with the optimal selection of the iterative parameters $\tau_1^*, \ldots, \tau_n^*$ in order to accelerate the convergence of the iteration. The operator B can made self-adjoint by setting it equal to the product of mutually adjoint operators

$$B = (E + \omega R_1)(E + \omega R_2), \quad R_2^* = R_1 \tag{13}$$

where $\omega > 0$ is a parameter, R_1 and R_2 can be taken as operators having triangular matrices (R_1 lower- and R_2 upper-triangular), so that

$$R_1 + R_2 = R : H \to H, R^* = R > 0.$$

In particular, it is possible to set

$$R_1 + R_2 = A, \quad R_2^* = R_1 . \tag{14}$$

It is typically assumed that

$$R \geq \delta E, \quad R_1 R_2 \leq \frac{\Delta}{4} A, \quad \delta > 0, \quad \Delta > 0. \tag{15}$$

Choosing $\omega = 2\sqrt{\delta\Delta}$ from the condition $\min n_0(\epsilon)$, we find the parameters γ_1, γ_2 and compute the parameters $\{\tau_k^*\}$. Determining y_{k+1} from y_k and f leads to the sequential solution of two systems of equations with lower and upper triangular matrices.

The iterative method (9) with the operator B factorized in the form (13) is called the alternate-triangular method (ATM). ATM is clearly a universal method, since the representation of A in the form

$$R_1 + R_2 = A, \quad R_2^* = R_1$$

is always possible. Constructing R_1 and R_2 in the case of an elliptic difference problem presents no difficulty. Thus, for example

$$R_1 y \to \sum_{\alpha=1}^{p} \frac{y_{\bar{x}_\alpha}}{h_\alpha}, \quad R_2 y \to -\sum_{\alpha=1}^{p} \frac{y_{x_\alpha}}{h_\alpha}$$

if A is the $2p+1$-point Laplace difference operator, and

$$Ay \to -\sum_{\alpha=1}^{p} y_{\bar{x}_\alpha x_\alpha},$$

where h_α is the grid step in the direction x_α. This method converges quickly. If we take the Chebyshev parameters $\{\tau_k^*\}$ and use (14), (15), then the number of iterations for the ATM satisfies

$$n_0(\epsilon) \geq \frac{1}{2\sqrt{2}\sqrt[4]{\eta}} \ln \frac{2}{\epsilon}, \quad \eta = \frac{\delta}{\Delta}. \tag{16}$$

In particular, for the model problem we have

$$n \geq n_0(\epsilon) = 0.3 \ln \frac{2}{\epsilon} \Big/ \sqrt{h}.$$

In the case of an arbitrary region and with equations having variable coefficients, it is appropriate to use the modified alternate-triangular method (MATM), setting

$$B = (\mathcal{D} + \omega R_1)\mathcal{D}^{-1}(\mathcal{D} + \omega R_2), \quad R_2^* = R_1, \quad \mathcal{D} = \mathcal{D}^* > 0, \tag{17}$$

where $\mathcal{D}$ is an arbitrary operator. If in place of (15) we use

$$R \geq \delta \mathcal{D}, \quad R_1 \mathcal{D}^{-1} R_2 \leq \frac{\Delta}{4}\mathcal{D}, \quad \delta > 0, \Delta > 0, \tag{18}$$

then the estimate (16) remains valid.

Here, δ and Δ are given, and the operator $\mathcal{D}$ and the parameter ω are chosen so that the ratio $\xi = \gamma_1/\gamma_2$ is maximized. In practice, the matrix $\mathcal{D}$ can be taken to be diagonal.

We indicate here two effective applications of the MATM.

[1] The Dirichlet problem for Poisson's equation in an arbitrary two-dimensional region; the basic grid in the plane (x_1, x_2) is uniform with step h, and a five-point scheme is used. The MATM for some given $\mathcal{D}$ requires only 4–5% more work per iteration than for the same problem in a square with side equal to the diameter of the region.

[2] For elliptic equations with quickly changing coefficients (the ratio c_2/c_1 is large), the MATM with a corresponding choice of $\mathcal{D}$ weakens the dependence on c_2/c_1.

In practice, besides the one-step (two-level) methods (9), two-step (three-level) iterative schemes are also applied. With the optimal iterative parameters, they are comparable (in terms of the number of iterations) with the Chebyshev scheme with parameters $\{\tau_k^*\}$ as $\xi \to 0$, however they are more sensitive to errors in the definition of γ_1 and γ_2. With the conditions (11), it is appropriate to use the Chebyshev scheme (9) with parameters $\{\tau_k^*\}$.

* * *

For solving elliptic problems, a very important role is played by the alternating-directions iterative method (ADI), which has been developed, starting in 1955, by many authors. However, it appeared to be efficient only for a very narrow class of problems from the first group, those which satisfied the conditions

$$A = A_1 + A_2, \quad A_\alpha = A_\alpha^* \geq 0, \ \alpha = 1,2, \quad A = A^* > 0, \quad A_1 A_2 = A_2 A_1.$$

If A_1 and A_2 commute, then it is possible to choose the optimal parameters for the ADI. For the model problem with these parameters, the number of iterations satisfies $n_0(\epsilon) = O(\ln \frac{1}{h} \ln \frac{1}{\epsilon})$, and the number of operations $Q(\epsilon) = O(\frac{1}{h^2} \ln \frac{1}{h} \ln \frac{1}{\epsilon})$, whereas for direct methods $Q = O(\frac{1}{h^2} \ln \frac{1}{h})$. Direct methods in this case are more economical than the ADI. If A_1 and A_2 do not commute, then the ADI requires $O(\frac{1}{h} \ln \frac{1}{\epsilon})$ iterations whereas for the ATM $O(\frac{1}{\sqrt{h}} \ln \frac{1}{\epsilon})$ iterations are sufficient. For three-dimensional problems, when $A = A_1 + A_2 + A_3$, even with the assumption of pairwise commutativity, the ADI requires more operations than the ATM. Thus, the ADI to a great degree had little significance.

* * *

If the operator $A > 0$ is not self-adjoint, then it is not possible using the scheme (9) with any choice of parameters and self-adjoint operator $B = B^* > 0$ to construct an iterative process with the same convergence rate as for the Chebyshev method for $A = A^* > 0$. All known methods possess a slower convergence rate. Here we consider the simple-iterative method (Chapter 6) with *a priori* information of two types:

[a] parameters γ_1, γ_2 entering into the condition (for simplicity we assume $D = B = E$)

$$\gamma_1(x,x) \leq (Ax,x), \quad (Ax,Ax) \leq \gamma_2(Ax,x), \quad \gamma_1 > 0, \ \gamma_2 > 0; \qquad (19)$$

[b] three parameters γ_1, γ_2, γ_3, where γ_1 and γ_2 (for $D = B = E$) are bounds on the symmetric part of the operator A:

$$\gamma_1 E \leq A \leq \gamma_2 E, \quad \| A_1 \| \leq \gamma_3, \quad \gamma_1 > 0, \ \gamma_3 \geq 0, \qquad (20)$$

where $A_1 = 0.5(A - A^*)$ is the skew-symmetric part of A.

Choosing τ from the minimum norm condition for the transition or resolving operator, in all cases we obtain an increase in the number of iterations in comparison with the case $A = A^*$.

* * *

Any two-level method, constructed on the basis of the scheme (9), is characterized by the operators B and A, the energy space H_D in which the convergence of the method is proved, and the choice of parameters. If the operator B is fixed, then the basic problem is finding $\{\tau_k\}$.

A priori information about the operators of the scheme is used to choose the parameters $\{\tau_k\}$. The form of the information is determined by the properties of the operators A, B, and D. So for the Chebyshev scheme with $D = AB^{-1}A$, when A and B are self-adjoint operators, it is assumed that the constants γ_1 and γ_2 in (11) are given. In the general case, when $DB^{-1}A$ is self-adjoint in H, then in place of (11) it is sufficient to assume that

$$\gamma_1 D \leq DB^{-1}A \leq \gamma_2 D, \quad \gamma_1 > 0.$$

In the non-self-adjoint case, when $A \neq A^*$, but $B = B^* > 0$, we use either the two numbers γ_1, γ_2 or the three numbers γ_1, γ_2 (entering in (19)) and γ_3 (a constant, entering into the estimate of the skew-symmetric part of the operator A). In a number of cases, finding the constants γ_1, γ_2, and γ_3 with sufficient accuracy can lead to a separate complex problem, requiring special algorithms for its solution. If the *a priori* information can be obtained at low cost, or if several solutions of the equation $Au = f$ with different right-hand sides are required, then it is appropriate to once find the constants γ_1, γ_2 and γ_3 and then use the Chebyshev method or the ATM. If the problem $Au = f$ must only be solved once, or if there is a good initial approximation, and if the computation of the constants γ_1, γ_2 is time-consuming, then a variational method should be used.

In order to compute the computational parameters $\{\tau_k\}$ for a variational method, it is not necessary to know γ_1, γ_2. These methods only use information of general form

$$D = D^* > 0, \quad (DB^{-1}A)^* = DB^{-1}A. \tag{21}$$

In order to determine y_{k+1}, the same scheme (9) is used; only the formula for τ_{k+1} is changed. The parameter τ_{k+1} is found by minimizing the norm in H_D of the error

$$z_{k+1} = y_{k+1} - u,$$

i.e., minimizing the functional

$$I[y] = (D(y-u), y-u).$$

The parameter τ_{k+1} is computed using y_k. Choosing $D = A$, we obtain the steepest-descent method; for $D = A^*A$, the minimum-residual method; etc. These methods have the same convergence rate as the simple-iterative method (with accurate constants γ_1, γ_2). The convergence rate of the iterations can

be improved if local (per-step) minimization of $\| z_{k+1} \|_D$ is avoided and if the parameter τ_k is chosen by minimizing the norm of the error $\| z_n \|_D$ after n steps, i.e., after the passage from y_0 to y_n. This leads to a two-parameter (for each k), three-level iterative conjugate-direction scheme (conjugate gradient, residual, correction, or error), which possesses the same convergence rate as the Chebyshev method with parameters $\{\tau_k^*\}$ computed with accurate values of γ_1, γ_2. If $A = A^* > 0$, then it is possible to accelerate (approximately 1.5 to 2 times) the convergence rate of two-level gradient methods.

* * *

In the general theory of iterative methods, knowledge of the concrete structure of the operators is not required — only a minimum of information concerning the general character of the operators is used, for example, condition (11). The choice of the operator B in (9) is subject to the requirements: 1) securing the fastest possible convergence rate for the method (9), 2) the efficient inversion of B. To construct B it is possible to start with some operator $R = R^* > 0$ (the regularizer), and with some energy equivalence for $A = A^* > 0$, $B = B^* > 0$:

$$c_1 R \leq A \leq c_2 R, \; c_1 > 0, \quad \mathring{\gamma}_1 B \leq R \leq \mathring{\gamma}_2 B, \; \mathring{\gamma}_1 > 0. \tag{22}$$

Thus

$$\gamma_1 = c_1 \mathring{\gamma}_1, \quad \gamma_2 = c_2 \mathring{\gamma}_2.$$

For various A, it is possible to choose the same regularizer R. Most common is the case of a factorized operator B, for example,

$$B = (E + \omega R_1)(E + \omega R_2), \quad R_1 + R_2 = R, \tag{23}$$

where

$$R_1^* = R_2 > 0 \text{ for the ATM} \tag{24}$$

$$R_1^* = R_1 > 0, \; R_2^* = R_2 > 0, \quad R_1 R_2 = R_2 R_1 \text{ for the ADI.} \tag{25}$$

In order to apply the theory, it is necessary to find $\mathring{\gamma}_1$ and $\mathring{\gamma}_2$; the parameter $\omega > 0$ is found from the condition

$$\min(\mathring{\gamma}_1(\omega)/\mathring{\gamma}_2(\omega)).$$

If the equation $Rw = F$ can be solved efficiently by a direct method, then we set $B = R$ (for example, in the case when $(-R)$ is the Laplace difference operator, and the region is a rectangle). The operator B cannot be written

out explicitly, but is realized as the result of the iterative solution of the equation

$$Rw = r_k, \quad r_k = Ay_k - f$$

(a two-stage method).

* * *

For equations with indefinite, singular, and complex operators A, it is still possible to consider the same scheme (9). However, the choice of the optimal parameters is more complicated, and the convergence rate is slower. Application of the general theory in these particular cases requires prior "reworking" of the problem. It is possible to construct modified versions of the Chebyshev method, as well as the methods of variational type.

If A is a singular linear operator, i.e., the homogeneous equation $Au = 0$ has a non-trivial solution, then the problem (9) for $B = E$ and for any τ_k is always soluble. Let $H^{(0)}$ be the null space of the operator A, and $H^{(1)}$ its orthogonal complement in H. Any vector $y \in H^{(0)}$ satisfies the equation $Ay = 0$. If $f \in H^{(1)}$ and $y_0 \in H^{(1)}$, then at each iteration $y_k \in H^{(1)}$. If the following condition is satisfied

$$\gamma_1(y, y) \le (Ay, y) \le \gamma_2(y, y), \quad y \in H^{(1)}, \quad \gamma_1 > 0,$$

then it is possible to use the explicit scheme (9) with the Chebyshev parameters $\{\tau_k^*\}$ found using γ_1, γ_2. Under these assumptions, y_k converges to the solution of the normal equations having minimal norm.

If

$$f = f^{(0)} + f^{(1)} \text{ and } f^{(0)} \neq 0,$$

then the generalized normal solution of the equation $Au = f$ will be taken to be the solution of the equation

$$Au^{(1)} = f^{(1)}, \quad u^{(1)} \in H^{(1)}$$

having minimal norm. We then have the estimates

$$\| y_n - u^{(1)} \| \le \tilde{q}_n \| y_0 - u^{(1)} \|, \quad \tilde{q}_n = q_{n-1} \left(1 + (n-1)\sqrt{\frac{1 - q_{n-1}^2}{\xi}} \right),$$

$$q_n = \frac{2\rho_1^n}{1 + \rho_1^{2n}}, \ \rho_1 = \frac{1 - \sqrt{\xi}}{1 + \sqrt{\xi}}, \ \xi = \frac{\gamma_1}{\gamma_2}, \ y_n, y_0 \in H^{(1)},$$

if $\tau_1^*, \ldots, \tau_{n-1}^*$ are the Chebyshev parameters, and

$$\tau_n^* = -\sum_{j=1}^{n-1} \tau_j^*.$$

The convergence rate is diminished in comparison with the non-singular case with the same γ_1, γ_2. There is a corresponding modified method of variational type.

The general theory allows us to investigate the implicit simple-iterative scheme for the case when H is a complex Hilbert space, $A = \tilde{A} + qE$, $\tilde{A}$ is a Hermitian operator, $q = q_1 + iq_2$ is a complex number, and to choose the optimal value for the iterative parameter. The transfer to the alternating-direction method presents no difficulty.

* * *

It is not difficult to use results from the general theory to solve difference equations approximating boundary-value problems for elliptic equations. It is easy to formulate general rules for the solution of difference problems. Let $Au = f$ be a difference equation, where $A : H \to H$ is a difference operator defined in the space H of grid functions defined on the grid ω. Initially we study the general properties of the operator A and establish, for example, that it is self-adjoint and positive, $A = A^* > 0$, then we construct the operator $B = B^* > 0$ and compute the constants γ_1, γ_2 and, finally, find $n = n_0(\epsilon)$ and the parameters $\{\tau_k^*\}$.

If we are using the ATM with factored operator

$$B = (\mathcal{D} + \omega R_1)\mathcal{D}^{-1}(\mathcal{D} + \omega R_2)$$

then it is necessary to choose the matrix $\mathcal{D}$ and the constants δ, Δ (see Chapter 10), and knowing δ, Δ, we determine ω, γ_1, γ_2, and so forth.

In the book, many examples are introduced which apply direct and iterative methods to solve concrete difference equations. In Chapter 15, in particular, methods for solving elliptic difference equations in curvilinear coordinates (both cylindrical (r, z) and polar (r, φ)) are considered.

In Chapter 14, we consider multi-dimensional problems, schemes for elasticity theory equations, etc.

It is important to note that, independent of the method which is being used to solve the given boundary-value difference problem, the preliminary work follows the same formula: initially formulate the operator A, then study it as an operator in the space H of grid functions. After this "harvest" of information about the problem is completed, consider the problem of choosing a method, taking into account all the particular circumstances (the machine, the availability of software, etc.).

Chapter 1

Direct Methods for Solving Difference Equations

In this chapter we study the general theory of linear difference equations, as well as direct methods for solving equations with constant coefficients, which give the solution in a closed form. In Section 1 general concepts about grid equations are introduced. Section 2 is devoted to the general theory of m^{th} order linear difference equations. In Section 3 methods for solving constant-coefficient equations are considered, and in Section 4 these methods are used to solve second-order equations. Solving grid eigenvalue problems for the simplest difference operators is discussed in Section 5.

1.1 Grid equations. Basic concepts

1.1.1 Grids and grid functions. A significant number of physics and engineering problems lead to differential equations with partial derivatives (mathematical-physics equations). A great variety of physical processes can be described by equations of elliptic type.

Explicit solutions of elliptic boundary-value problems are obtainable only in special cases. Therefore these problems are generally solved approximately. One of the most universal and effective methods in wide use today for approximately solving mathematical-physics equations is the method of finite differences or the method of grids.

The essence of the method is as follows. The continuous domain region (for example, an interval, a rectangle, etc.) is replaced by a discrete set of points (nodes), called the *grid* or *lattice*. In place of a function of continuous arguments we consider a function of discrete arguments, defined at the nodes of the grid and called the *grid function*. The derivatives entering into the differential equation and the boundary conditions are changed into difference

derivatives; thus the boundary-value problem for a differential equation is changed into a system of linear or non-linear algebraic equations (grid or difference equations). Such a system is often called a *difference scheme.*

We will expand in more detail on the basic concepts of the grid method. We first consider the simplest examples of grids.

Example 1. *Grids in a one-dimensional region.* Let the domain of the variable x be the interval $0 \le x \le l$. We split this interval into N equal parts of length $h = l/N$ using points $x_i = ih$, $i = 0, 1, \ldots, N$. This set of points is called the *uniform grid* on the interval $[0, l]$ and is denoted $\bar{\omega} = \{x_i = ih,\ i = 0, 1, \ldots, N\ hN = l\}$; the number h — the distance between points (nodes) of the grid $\bar{\omega}$ — is called the grid *step.*

To subdivide the grid $\bar{\omega}$ we will also use the following definitions:

$$\begin{aligned}
\omega &= \{x_i = ih, \quad i = 1, 2, \ldots, N-1, Nh = l\} \\
\omega^+ &= \{x_i = ih, \quad i = 1, 2, \ldots, N, \quad Nh = l\} \\
\omega^- &= \{x_i = ih, \quad i = 0, 1, \ldots, N-1, Nh = l\} \\
\gamma &= \{x_0 = 0, \quad x_N = l\}
\end{aligned}$$

The interval $[0, l]$ can be split into N parts using arbitrary points $0 = x_0 < x_1 < \ldots < x_i < x_{i+1} < \ldots < x_{N-1} < x_N = l$. In this case, we obtain the grid $\bar{\omega} = \{x_i, i = 0, 1, \ldots, N, x_0 = 0, x_N = l\}$ with step $h_i = x_i - x_{i-1}$ at the point $x_i, i = 1, 2, \ldots, N$, which depends on the index i of the node x_i, i.e., $h_i = h(i)$ is a grid function.

If $h_i \neq h_{i+1}$ for even one index i, then the grid $\bar{\omega}$ is called *non-uniform.* If $h_i = l/N$, then we obtain the uniform grid constructed above. For a non-uniform grid, we define the average step $\hbar_i = \hbar(i)$ at the node x_i,

$$\hbar_i = 0.5(h_i + h_{i+1}), \quad 1 \le i \le N-1, \quad \hbar_0 = 0.5h_1, \quad \hbar_N = 0.5h_N.$$

On the infinite line

$$-\infty < x < \infty$$

it is possible to consider the grid

$$\Omega = \{x_i = a + ih,\ i = 0, \pm 1, \pm 2, \ldots\}$$

beginning at any point $x = a$ and with step h, consisting of an infinite number of nodes.

Example 2. *A grid in a two-dimensional region.* Let the domain of the variables $x = (x_1, x_2)$ be the rectangle

$$\bar{G} = \{0 \le x_\alpha \le l_\alpha,\ \alpha = 1, 2\}$$

with boundary Γ. On the intervals $0 \le x_\alpha \le l_\alpha$ we construct the uniform grid $\bar{\omega}_\alpha$ with steps h_α:

$$\bar{\omega}_1 = \{x_1(i) = ih_1, i = 0, 1, \ldots, M,\ h_1 M = l_1\},$$
$$\bar{\omega}_1 = \{x_2(j) = jh_2, j = 0, 1, \ldots, N,\ h_2 N = l_2\}.$$

The set of nodes $x_{ij} = (x_1(i), x_2(j))$, having coordinates on the plane $x_1(i)$ and $x_2(j)$, is called the *grid* in the *rectangle* $\bar{G}$ and is denoted

$$\bar{\omega} = \{x_{ij} = (ih_1, jh_2),\ i = 0, 1, \ldots, M,\ j = 0, 1, \ldots, N, h_1 M = l_l,\ h_2 N = l_2\}.$$

Clearly, the grid $\bar{\omega}$ consists of points intersecting the lines $x_1 = x_1(i)$ and $x_2 = x_2(j)$.

The constructed grid $\bar{\omega}$ is uniform for each of the variables x_1 and x_2. However, if one of the grids $\bar{\omega}_\alpha$ is non-uniform, then the grid $\bar{\omega}$ is called *non-uniform*. If $h_1 = h_2$ then the grid is called *square*, otherwise it is *rectangular*.

The points of $\bar{\omega}$ belonging to Γ are called *boundary points* and their union forms the boundary of the grid: $\gamma = \{x_{ij} \in \Gamma\}$.

In order to describe the structure of the grid $\bar{\omega}$, it is convenient to use the notation $\bar{\omega} = \bar{\omega}_1 \times \bar{\omega}_2$, i.e., to represent $\bar{\omega}$ as the topological product of the grids $\bar{\omega}_1$ and $\bar{\omega}_2$. Using the definitions of ω^+, ω^- and ω introduced in example 1, it is possible to subdivide the grid $\bar{\omega}$ in the rectangle, for example:

$$\omega_1 \times \omega_2^+ = \{x_{ij} = (ih_1, jh_2),\ i = 1, 2, \ldots, M-1,\ j = 1, 2, \ldots, N\}$$
$$\omega_1^- \times \bar{\omega}_2 = \{x_{ij} = (ih_1, jh_2),\ i = 0, 1, \ldots, M-1,\ j = 0, 1, \ldots, N\}.$$

We now consider the concept of a grid function. Let $\bar{\omega}$ be a grid introduced in a one-dimensional region, and let x_i be the nodes of the grid. A function $y = y(x_i)$ of the discrete variable x_i is called a *grid function* defined on the grid $\bar{\omega}$. Analogously, we define a grid function on any grid $\bar{\omega}$ in a domain. For example, if x_{ij} is a node of the grid $\bar{\omega}$ in a two-dimensional region, then $y = y(x_{ij})$. Obviously, grid functions can also be considered as functions of an integer variable, the node-number of the grid point. So, we can write $y = y(x_i) = y(i), y = y(x_{ij}) = y(i, j)$. We will sometimes use the following notation for grid functions: $y(x_i) = y_i, y(x_{ij}) = y_{ij}$.

The grid function y_i can be represented as a vector if we consider the values of the function as components of the vector $Y = (y_0, y_1, \ldots, y_N)^T$. In

this example, y_i is defined on the grid $\bar{\omega} = \{x_i, i = 0, 1, \ldots, N\}$ containing $N + 1$ nodes, and the vector Y has dimension $N + 1$. If $\bar{\omega}$ is a grid in the rectangle

$$\bar{\omega} = \{x_{ij} = (ih_1, jh_2),\ i = 0, 1, \ldots, M,\ j = 0, 1, \ldots, N\},$$

then the grid function y_{ij}, defined on $\bar{\omega}$, corresponds to the vector $Y = (y_{00}, \ldots, y_{M0}, y_{01}, \ldots, y_{M1}, \ldots, y_{0N}, \ldots, y_{MN})^T$ of dimension $(M+1)(N+1)$. The nodes of the grid $\bar{\omega}$ will be ordered according to the rows of the grid.

We have considered scalar grid functions, i.e., those functions for which the value at each node is a number. We now introduce examples of *vector grid functions*, which are vector-valued at each node. If in the example above we denote by $Y(x_2(j)) = Y_j$ the vector consisting of the value of the grid function y_{ij} at the nodes $x_{0j}, x_{1j}, \ldots, x_{Mj}$ of the j^{th} row of the grid $\bar{\omega} : Y_j = (y_{0j}, y_{1j}, \ldots, y_{Mj})^T$, $j = 0, 1, \ldots, N$, then we obtain the vector grid function Y_j defined on the grid $\bar{\omega}_2 = \{x_2(j) = jh_2,\ j = 0, 1, \ldots, N\}$. If the function defined on the grid has complex values, then the grid function will be called *complex.*

1.1.2 Difference derivatives and various difference identities. Let $\bar{\omega}$ be a given grid. The set of all grid functions defined on $\bar{\omega}$ forms a vector space with the obvious definitions of addition and multiplication by a scalar. It is possible to define difference or grid operators on the space of grid functions. An operator Λ, mapping a grid function y into a grid function $f = \Lambda y$, is called a *grid* or *difference* operator. The set of nodes used to write the difference operator at a node of the grid is called the *stencil* of this operator.

The simplest difference operator is the difference differentiation operator for a grid function, which gives rise to difference derivatives. We will now define difference derivatives.

Let Ω be a uniform grid with step h, defined on the line $-\infty < x < \infty$:

$$\Omega = \{x_i = a + ih, \quad i = 0, \pm 1, \pm 2, \ldots\}.$$

Difference derivatives of first order for the grid functions $y_i = y(x_i), x_i \in \Omega$ are defined by the formulas

$$\Lambda_1 y_i = y_{\bar{x},i} = \frac{y_i - y_{i-1}}{h}, \quad \Lambda_2 y_i = y_{x,i} = \frac{y_{i+1} - y_i}{h} \tag{1}$$

and are called *left* and *right derivatives*, respectively. We shall also use *central derivatives*

$$\Lambda_3 y_i = y_{\mathring{x},i} = \frac{y_{i+1} - y_{i-1}}{2h} = 0.5(\Lambda_1 + \Lambda_2) y_i. \tag{2}$$

If the grid is non-uniform, then the following definitions are used for first-order difference derivatives:

$$y_{\bar{x},i} = \frac{y_i - y_{i-1}}{h_i}, \quad y_{x,i} = \frac{y_{i+1} - y_i}{h_{i+1}}, \quad y_{\hat{x},i} = \frac{y_{i+1} - y_i}{\hbar_i}$$
$$y_{\mathring{x},i} = 0.5(y_{\bar{x},i} + y_{x,i}), \quad \hbar_i = 0.5(h_i + h_{i+1}). \tag{3}$$

From definitions (1) and (3), we obtain the following relations:

$$y_{x,i} = y_{\bar{x},i+1} \tag{4}$$

$$y_{x,i} = \frac{\hbar_i}{h_{i+1}} y_{\hat{x},i}, \tag{5}$$

and also the equalities

$$y_i = y_{i+1} - h_{i+1} y_{x,i} = y_{i-1} + h_i y_{\bar{x},i}. \tag{6}$$

The difference operators Λ_1, Λ_2, and Λ_3 have stencils consisiting of two points, and are used to approximate the first derivative $Lu = u'$ of the one-variable function $u = u(x)$. For smooth functions, the operators Λ_1 and Λ_2 approximate the operator L with error $O(h)$, and Λ_3 with error $O(h^2)$.

Difference derivatives of n^{th} *order* are defined as the grid functions obtained by computing the first difference derivative of a difference derivative of $n - 1^{\text{st}}$ order. We now introduce examples of second-order difference derivatives:

$$y_{\bar{x}x,i} = \frac{y_{\bar{x},i+1} - y_{\bar{x},i}}{h} = \frac{1}{h^2}(y_{i-1} - 2y_i + y_{i+1}),$$

$$y_{\mathring{x}\mathring{x},i} = \frac{y_{\mathring{x},i+1} - y_{\mathring{x},i-1}}{2h} = \frac{1}{4h^2}(y_{i-2} - 2y_i + y_{i+2}),$$

$$y_{\bar{x}\hat{x},i} = \frac{1}{\hbar_i}(y_{\bar{x},i+1} - y_{\bar{x},i}) = \frac{1}{\hbar_i}(y_{x,i} - y_{\bar{x},i})$$
$$= \frac{1}{\hbar_i}\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right),$$

which are used to approximate the second derivative $Lu = u''$ of the function $u = u(x)$. In the case of a uniform grid, the error of approximation is $O(h^2)$. The corresponding difference operators have three-point stencils. To approximate the fourth derivative $Lu = u^{IV}$, we use a fourth-order difference derivative

$$y_{\bar{x}x\bar{x}x,i} = \frac{1}{h^4}(y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}).$$

Analogously, we use an n^{th} order difference derivative to approximate the n^{th} derivative.

There is no difficulty in defining difference derivatives for grid functions of several variables.

To transform expressions containing difference derivatives of grid functions, we need formulas for the difference derivative of the product of grid functions and formulas for summation by parts. These formulas are analogous to the corresponding formulas in differential calculus.

[1] *Formulas for the Difference Derivative of a Product.*

Using the definitions of difference derivatives (3), it is not difficult to verify the identities:

$$(uv)_{\bar{x},i} = u_{\bar{x},i}v_{i-1} + u_i v_{\bar{x},i} = u_{\bar{x},i}v_i + u_{i-1}v_{\bar{x},i} = u_{\bar{x},i}v_i + u_i v_{\bar{x},i} - h_i u_{\bar{x},i}v_{\bar{x},i},$$
$$(uv)_{x,i} = u_{x,i}v_{i+1} + u_i v_{x,i} = u_{x,i}v_i + u_{i+1}v_{x,i} = u_{x,i}v_i + u_i v_{x,i} + h_{i+1}u_{x,i}v_{x,i},$$
$$(uv)_{\hat{x},i} = u_{\hat{x},i}v_{i+1} + u_i v_{\hat{x},i} = u_{\hat{x},i}v_i + u_{i+1}v_{\hat{x},i} = u_{\hat{x},i}v_i + u_i v_{\hat{x},i} + \hbar_i u_{\hat{x},i}v_{\hat{x},i}.$$

Using (4), (5), the last identity can be written in the form

$$(uv)_{\hat{x},i} = u_{\hat{x},i}v_i + \frac{h_{i+1}}{\hbar_i}u_{i+1}v_{\bar{x},i+1}. \tag{7}$$

[2] *Formulas for Summation by Parts.*

Multiplying (7) by $\hbar_i$, and summing the resulting relation for i between $m+1$ and $n-1$, we find that

$$\sum_{i=m+1}^{n-1}(uv)_{\hat{x},i}\hbar_i = u_n v_n - u_{m+1}v_{m+1}$$
$$= \sum_{i=m+1}^{n-1} u_{\hat{x},i}v_i\hbar_i + \sum_{i=m+1}^{n-1} u_{i+1}v_{\bar{x},i+1}h_{i+1}.$$

Using (6), we obtain the relation

$$v_{m+1} = v_m + h_{m+1}v_{x,m} = v_m + h_{m+1}v_{\bar{x},m+1},$$

which we substitute in the above equality. As a result we have

$$u_n v_n - u_{m+1}v_m = \sum_{i=m+1}^{n-1} u_{\hat{x},i}v_i\hbar_i + \sum_{i=m}^{n-1} u_{i+1}v_{\bar{x},i+1}h_{i+1}.$$

Changing the index of summation to $i' = i - 1$ in the second sum on the right-hand side gives the following formula for summation by parts:

$$\sum_{i=m+1}^{n-1} u_{\hat{x},i} v_i \hbar_i = - \sum_{i=m+1}^{n} u_i v_{\bar{x},i} h_i + u_n v_n - u_{m+1} v_m. \tag{8}$$

Using (6), it is easy to obtain from (8) another formula for summation by parts

$$\sum_{i=m+1}^{n-1} u_{\bar{x},i} v_i h_i = - \sum_{i=m}^{n-1} u_i v_{\hat{x},i} \hbar_i + u_{n-1} v_n - u_m v_m. \tag{9}$$

From the formula (8) it follows that the function u_i must be defined for $m+1 \le i \le n$, and the function v_i for $m \le i \le n$. Suppose now that y_i is a grid function defined for $m \le i \le n$. Then the function $u_i = y_{\bar{x},i}$ is defined for $m+1 \le i \le n$. Substituting u_i in (8), we obtain the following identity:

$$\sum_{i=m+1}^{n-1} y_{\bar{x}\hat{x},i} v_i \hbar_i = - \sum_{i=m+1}^{n} y_{\bar{x},i} v_{\bar{x},i} h_i + y_{\bar{x},n} v_n - y_{x,m} v_m. \tag{10}$$

The following is valid

Lemma 1. *Suppose that the grid function y_i is defined on the arbitrary non-uniform grid*

$$\bar{\omega} = \{x_i, i = 0, 1, \ldots, N,\ x_0 = 0\ x_N = l\}$$

and that y_i is zero for $i = 0$, $i = N$. For this function, the following equality is valid

$$\sum_{i=1}^{N-1} y_{\bar{x}\hat{x},i} y_i \hbar_i = - \sum_{i=1}^{N} (y_{\bar{x},i})^2 h_i.$$

The proof of lemma 1 follows in an obvious manner from (10).

Corollary. *If $\bar{\omega}$ is a uniform grid, $y_0 = y_N = 0$ and $y_i \not\equiv 0$, then*

$$\sum_{i=1}^{N-1} y_{\bar{x}x,i} y_i h = - \sum_{i=1}^{N} y_{\bar{x},i}^2 h < 0.$$

With this we conclude our discussion of difference formulas. Several other formulas will be considered in Chapter 5.

These identities are not just used for transforming difference expressions. They are often applied, for example, to compute alternate forms of finite sums and series.

We mention here an example. We wish to compute the sum

$$S_n = \sum_{i=1}^{n-1} ia^i, \quad a \neq 1.$$

We introduce the following grid functions, defined on the uniform grid $\bar{\omega} = \{x_i, i = 0, 1, \ldots, N, h = 1\}$:

$$v_i = i, \quad u_i = (a^i - a^n)/(a-1). \tag{11}$$

On this grid, the summation by parts formula (8) for any grid functions has the form ($m = 0$)

$$\sum_{i=1}^{n-1} u_{x,i} v_i = -\sum_{i=1}^{n} u_i v_{\bar{x},i} + u_n v_n - u_1 v_0.$$

Taking into account that the function (11) satisfies the relations

$$v_0 = u_n = 0, \quad v_{\bar{x},i} = 1, \quad u_{x,i} = a^i$$

we obtain

$$S_n = \sum_{i=1}^{n-1} ia^i = -\sum_{i=1}^{n} \frac{a^i - a^n}{a-1} = \frac{a^n(n(a-1)-a)+a}{(a-1)^2}.$$

The desired sum has been found.

1.1.3. Grid and difference equations. Let $y_i = y(i)$ be a grid function of the discrete variable i. The values of the grid function $y(i)$ in turn form a discrete set. On this set it is possible to define a grid function, and equating this function to zero we obtain an equation related to the grid function $y(i)$ — the *grid equation.* A special case of the grid equation is the *difference equation.* Difference equations will be the basic object of study in our book.

Grid equations are obtained when approximating integral and differential equations on a grid.

We will first mention difference approximations of ordinary differential equations.

We will transform first-order differential equations

$$\frac{du}{dx} = f(x), \quad x > 0$$

into first-order difference equations

$$\frac{y_{i+1} - y_i}{h} = f(x_i), \; x_i = ih, \; i = 0, 1, \ldots$$

or $y_{i+1} = y_i + hf(x_i)$, where h is the step of the grid $\omega = \{x_i = ih, \; i = 0, 1, \ldots\}$. The desired function is the grid function $y_i = y(i)$.

To approximate the second-order equation

$$\frac{d^2u}{dx^2} = f(x),$$

we use a second-order difference equation

$$y_{i+1} - 2y_i + y_{i-1} = \varphi_i, \quad \varphi_i = h^2 f_i, \quad f_i = f(x_i), \quad x_i = ih.$$

If the equation of general form

$$(ku')' + ru' - qu = -f(x)$$

is approximated on the three-point stencil (x_{i-1}, x_i, x_{i+1}), then we obtain a second-order difference equation with variable coefficients of the form

$$a_i y_{i-1} - c_i y_i + b_i y_{i+1} = -\varphi_i, \; i = 0, 1, \ldots$$

where a_i, c_i, b_i, and φ_i are given grid functions, and y_i is the desired grid function.

Approximating the fourth-order equation

$$(ku'')'' = f(x)$$

on a grid leads to a fourth-order difference equation; it has the form

$$a_i^{(2)} y_{i-2} + a_i^{(1)} y_{i-1} + c_i y_i + b_i^{(1)} y_{i+1} + b_i^{(2)} y_{i+2} = \varphi_i.$$

For difference approximations to the derivatives u', u'', and u''', it is possible to use stencils with a large number of points. This leads to higher-order difference equations.

The linear equation related to the grid function $y(i)$ (a function of the integer variable i)

$$a_0(i)y(i) + a_1(i)y(i+1) + \ldots + a_m(i)y(i+m) = f(i), \tag{12}$$

where $a_0(i) \neq 0$ and $a_m(i) \neq 0$, and $f(i)$ is a given grid function, is called an m^{th} *order difference equation.*

If (12) does not contain $y(i)$, but contains $y(i+1)$, then changing the independent variable from $i+1$ to i' transforms this equation into an equation of order $m-1$.

This is one difference between grid equations and differential equations, where a change in the independent variable does not lead to a change in the order of the equation.

Let $F(i, y(i), y(i+1), \ldots, y(i+m))$ be a non-linear grid function. Then $F(i, y(i), y(i+1), \ldots, y(i+m)) = 0$ is a non-linear m^{th} *order difference equation* if F explicitly depends on $y(i)$ and $y(i+m)$.

For convenience when comparing with differential equations, we introduce *(right) grid function differences*:

$$\Delta y_i = y_{i+1} - y_i, \quad \Delta^2 y_i = \Delta(\Delta y_i), \ldots, \Delta^{k+1} y_i = \Delta(\Delta^k y_i), \quad k = 1, 2, \ldots .$$

Then (12) can be written in the form

$$\alpha_0(i)y_i + \alpha_1(i)\Delta y_i + \ldots + \alpha_m(i)\Delta^m y_i = f_i, \tag{12'}$$

where $\alpha_m(i) = a_m(i) \neq 0$.

The difference equation (12') is a formal analog of the m^{th} order differential equation

$$\alpha_0 u + \alpha_1 \frac{du}{dx} + \ldots + \alpha_{m-1} \frac{d^{m-1}u}{dx^{m-1}} + \alpha_m \frac{d^m u}{dx^m} = f(x),$$

where $\alpha_m \neq 0$, $\alpha_k = \alpha_k(x)$, $k = 0, 1, \ldots, m$. Let $\omega = \{x_i = ih, i = 0, 1, \ldots\}$ be some grid. If we designate

$$y_{x,i} = \frac{y_{i+1} - y_i}{n}, \; y_{xx,i} = (y_x)_{x,i}, \ldots, y_x^{(k)} = y\underbrace{x \ldots x}_{k\text{times}},i$$

so that

$$y_x^{(k)} = \left(y_x^{(k-1)}\right)_x, \quad k \geq 1, \quad y_{x,i}^{(0)} = y(i),$$

then $y(i+k)$ is expressed in terms of

$$y(i), \; y_x^{(1)}, \ldots, y_x^{(k-1)}$$

for example

$$y(i+3) = y(i) + 3hy_{x,i} + 3h^2 y_{xx,i} + h^3 y_{xxx,i}.$$

Then equation (12) will be written in the form

$$\bar{\alpha}_0 y(i) + \bar{\alpha}_1(i) y_x(i) + \ldots + \bar{\alpha}_{m-1} y_x^{(m-1)}(i) + \bar{\alpha}_m y_x^{(m)}(i) = f_i,$$

where $\bar{\alpha}_m = a_m \neq 0$. Here the analogy with the m^{th} order differential equation is obvious.

Analogously, we define the difference equation relative to the grid function

$$y_{i_1,i_2} = y(i_1, i_2)$$

of two discrete variables, and in general of any number of variables. For example, the five-point "cross" scheme for Poisson's equation

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = -f(x_1, x_2)$$

on the grid $\omega = \{x_i = (i_1 h_1, i_2 h_2), i_1, i_2 = 0, 1, \ldots\}$ has the form

$$\frac{y(i_1-1, i_2) - 2y(i_1, i_2) + y(i_1+1, i_2)}{h_1^2} + \frac{y(i, i_2-1) - 2y(i_1, i_2) + y(i_1, i_2+1)}{h_2^2} = -f_{i_1 i_2}$$

and is represented as a second-order difference equation in each of the discrete variables i_1 and i_2.

The grid equation of *general* form is obtained by approximating the integral equation

$$u(x) = \int_0^1 K(x,s)u(s)ds + f(x), \quad 0 \leq x \leq 1$$

on the grid $\bar{\omega} = \{x_i = ih, i = 0, 1, \ldots, N, hN = 1\}$. We replace the integral by a sum

$$\int_0^1 K(x,s)u(s)ds \approx h \sum_{j=0}^{N} \alpha_j K(x, jh)\, u(jh),$$

where α_j is a coefficient from the quadrature formula, and in place of the integral equation we write the grid equation

$$y_i = \sum_{j=0}^{N} \alpha_j \, K(ih, jh) y_j + f_i, \quad i = 0, 1, \ldots, N,$$

where the summation is taken over all nodes of the grid $\bar{\omega}$, and the unknown is the grid function y_i.

The grid equation can be written in the form

$$\sum_{j=0}^{N} c_{ij} y_j = f_i, \quad i = 0, 1, \ldots, N. \tag{13}$$

It contains all the values $y_0, y_1, \ldots, y_N$ of the grid function. It can be expressed as a difference equation of order N, equal to the number of grid nodes minus one.

The m^{th} order difference equation (12) is a special form of the grid equation where the matrix (c_{ij}) is only non-zero along the m diagonals parallel to the main diagonal.

In the general case, we can take i to be not only an index $i = 0, 1, \ldots$, but also a multi-index, i.e., a vector $i = (i_1, i_2, \ldots, i_p)$ with integer components $i_\alpha = 0, 1, 2, \ldots$, $\alpha = 1, 2, \ldots, p$, where $i \in \omega$, and ω is a grid.

The linear grid equation has the form

$$\sum_{j \in \omega} c_{ij} y_j = f_i, \qquad i \in \omega$$

where the summation is taken over all nodes of the grid ω, f_i is given, and y_i is the desired grid function.

If we renumber all the nodes of the grid, then we can write $y_i = y(i)$, where i is the number of a node, $i = 0, 1, 2, \ldots, N$. Then the grid equation (14) has the form (13).

Obviously, this is a system of linear algebraic equations of order $N + 1$ with matrix (c_{ij}). Thus, any system of linear equations can be expressed as a grid equation, and *vice versa.*

If $y(i)$ is a *vector* grid function, we speak of m^{th} *order grid (difference) equations.*

Let $F(i, y_0, y_1, \ldots, y_N)$ be a given function (generally speaking, non-linear) of the $N + 2$ variables $i, y_0, y_1, \ldots, y_N$. Setting it equal to zero, we obtain the non-linear grid equation $F(i, y_0, y_1, \ldots, y_N) = 0$, $i = 0, 1, \ldots, N$,

the solution of which is the grid function $y(i)$ which turns this equation into an identity.

We now consider the grid function $\mathcal{F}(i) = F(i, y_0, y_1, \ldots, y_N)$, $i = 0, 1, \ldots, N$. From this it is clear that the function F is a grid operator which maps the grid function $y(i)$ into the grid function $\mathcal{F}(i)$.

If F is a linear function, then we obtain equation (14), which clearly can be written in the operator form $Ay = f$, where A is the linear operator with matrix (a_{ij}), and y is a vector in the space of grid functions.

If the coefficients a_{ij} do not depend on $i - j$, then (14) is called a *grid equation with constant coefficients.*

Although in this book the basic focus is on the numerical solution of difference equations obtained from difference approximations to elliptic differential equations, iterative methods are applicable to any linear grid equation, i.e., to any system of linear equations. Therefore, the theory of iterative methods presented here has a general character. What is specific to grid equations is that this system is of high order, since refinement of the grid increases the order of the equations (the number of unknowns is equal to the number N of grid points, $N = O(1/h^p)$ in the p-dimensional case, where h is the mesh size).

1.1.4 The Cauchy problem and boundary-value problems for difference equations. We will now mention several further examples of difference equations and dwell on the posing of problems for difference equations.

Notice that the simplest examples of first-order difference equations are the formulas for the terms of an arithmetic or a geometric progression:

$$y_{i+1} = y_i + d, \quad y_{i+1} = qy_i, \quad i = 0, 1, \ldots .$$

The solution of a first-order equation can be found if an initial condition is given for $i = 0$ (the Cauchy problem).

The solution $y(i + m)$ of an m^{th} order difference equation is fully determined by the values $y(i)$ at m arbitrary but sequential points $i_0, i_0 + 1, \ldots, i_0 + m - 1$. In fact, since $a_m(i) \neq 0$, from (12) we find that

$$y(i + m) = b_{m-1}(i)y(i + m - 1) + \ldots + b_0(i)y(i) + \varphi(i).$$

Setting $i = i_0, i_0 + 1, \ldots$, we can find the values of $y(i)$ for $i \geq i_0$. Analogously, using (12) to express $y(i)$ in terms of $y(i + 1), \ldots, y(i + m)$ and setting $i = i_0 - 1, i_0 - 2, \ldots$, we can find $y(i)$ for $i \leq i_0 - 1$. If in equation (12) it is necessary to determine $y(i)$ for $i \geq 0$, then it is sufficient to give the values at the m points $y(0) = y_0, y(1) = y_1, \ldots, y(m - 1) = y_{m-1}$ (the initial conditions).

Adjoining these conditions to equation (12), we obtain the Cauchy problem or the initial value problem for an m^{th} order difference equation.

As we saw, for first-order equations ($m = 1$) it is sufficient to give one initial condition.

Non-linear difference equations are obtained when solving non-linear differential equations. Consider, for example, the differential equation

$$\frac{du}{dx} = f(x, u), \quad x > 0, \quad u(0) = \mu_1$$

(a Cauchy problem). Applying the Euler scheme (an explicit scheme), we obtain a first-order difference equation $y_{i+1} = y_i + hf(x_i, y_i), i \geq 0, y_0 = \mu_1$.

If the derivative du/dx at $x = x_i = ih$ is changed to the left difference quotient, then we obtain for y_i a non-linear first-order equation, $y_i = y_{i-1} + hf(x_i, y_i), i > 0, y_0 = \mu_1$. To determine y_i it is necessary to solve the non-linear equation $\varphi(y_i) = y_i - hf(x_i, y_i) = y_{i-1}$.

We now consider an example of a second-order difference equation. Suppose it is necessary to compute the integral

$$I_k(\varphi) = \int_0^\pi \frac{\cos k\Psi - \cos k\varphi}{\cos \Psi - \cos \varphi} d\Psi, \qquad k = 0, 1, 2, \ldots .$$

First of all, notice that $I_0(\varphi) = 0$, $I_1(\varphi) = \pi$. Consider the identity

$$\begin{aligned}[\cos(k+1)\Psi - \cos(k+1)\varphi] + [\cos(k-1)\Psi - \cos(k-1)\varphi] \\ = 2\cos k\Psi \cos \Psi - 2\cos k\varphi \cos \varphi \\ = 2(\cos k\Psi - \cos k\varphi)\cos \varphi + 2(\cos \Psi - \cos \varphi)\cos k\Psi.\end{aligned}$$

Using this, we obtain

$$I_{k+1}(\varphi) + I_{k-1}(\varphi) = 2\cos \varphi I_k(\varphi) + 2\int_0^\pi \cos k\Psi \, d\Psi = 2\cos \varphi I_k(\varphi), \; k \geq 1.$$

Thus, the computation of the integral $I_k(\varphi)$ leads to the solution of a Cauchy problem for the second-order difference equation

$$I_{k+1}(\varphi) - 2\cos \varphi I_k(\varphi) + I_{k-1}(\varphi) = 0, \; k \geq 1, \; I_0(\varphi) = 0, \; I_1(\varphi) = \pi. \quad (15)$$

We will consider one further example. We are required to find the solution to a boundary-value problem for a system of first-order ordinary differential equations

$$\frac{du}{dx} = Au + f(x), \qquad 0 < x < l, \quad (16)$$

where $Bu = \mu_1$ for $x = 0$, $Cu = \mu_2$ for $x = l$. Here $u(x) = (u_1(x), u_2(x), \ldots, u_M(x))^T$ is a vector function of dimension M, $A = A(x)$ is a square matrix of size $M \times M$, and B and C are rectangular matrices of sizes $M_1 \times M$ and $M_2 \times M$ respectively, where $M_1 + M_2 = M$. The vectors $f(x)$, μ_1, and μ_2 are given and have dimensions M, M_1, and M_2 respectively.

Introducing the uniform grid $\bar{\omega} = \{x_i = ih, i = 0, 1, \ldots, N, h = l/N\}$ onto the interval $0 \le x \le l$ and defining on it the grid vector-function $Y_i = (y_1(i), y_2(i), \ldots, y_M(i))^T$, we obtain from (16) the simple difference scheme

$$\begin{gathered} Y_{i+1} - (E + hA_i)Y_i = F_i \quad 0 \le i \le N-1, \\ BY_0 = \mu_1, \quad CY_N = \mu_2, \end{gathered} \tag{17}$$

where $F_i = hf(x_i)$. This is an example of a first-order linear vector difference equation with M_1 conditions at $i = 0$ and M_2 conditions at $i = N$. Thus, we have a boundary-value problem for systems of first-order difference equations.

Boundary-value problems are more typical for second-order equations. Let us consider, for example, the boundary-value problem

$$\frac{d^2u}{dx^2} - q(x)u = -f(x),\ 0 < x < l,\ u(0) = \mu_1,\ u(l) = \mu_2,\ q(x) \ge 0. \tag{18}$$

Choosing the grid $\bar{\omega} = \{x_i = ih, i = 0, 1, \ldots, N, h = l/N\}$, we obtain from (18) the corresponding boundary-value difference problem

$$y_{\bar{x}x,i} - d_i y_i = -\varphi_i, \quad 0 < i < N, \quad y_0 = \mu_1, \quad y_N = \mu_2, \tag{19}$$

where $d_i = q(x_i)$, $\varphi_i = f(x_i)$ for smooth $q(x)$, $f(x)$. This problem is a special case of a boundary-value problem for second-order difference equations

$$-a_i y_{i-1} + c_i y_i - b_i y_{i+1} = \varphi_i, \quad 1 \le i \le N-1, \quad y_0 = \mu_1, \quad y_N = \mu_2, \tag{20}$$

with $a_i = b_i = 1/h^2$, $c_i = d_i + 2/h^2$.

The difference problem (20) can be written in the form

$$\mathcal{A}Y = F, \tag{21}$$

where $Y = (y_1, y_2, \ldots, y_{N-1})^T$ is unknown,

$$F = \left(\varphi_1 + \frac{1}{h^2}\mu_1, \varphi_2, \ldots, \varphi_{N-2}, \varphi_{N-1} + \frac{1}{h^2}\mu_2\right)^T$$

is a known vector of dimension $N-1$, and $\mathcal{A}$ is a square tridiagonal matrix

of the form

$$\mathcal{A} = \begin{Vmatrix} c_1 & -b_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -a_2 & c_2 & -b_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & -a_3 & c_3 & -b_3 & \dots & 0 & 0 & 0 \\ . & . & . & . & \dots & . & . & . \\ 0 & 0 & 0 & 0 & \dots & c_{N-3} & -b_{N-3} & 0 \\ 0 & 0 & 0 & 0 & \dots & -a_{N-2} & c_{N-2} & -b_{N-2} \\ 0 & 0 & 0 & 0 & \dots & 0 & -a_{N-1} & c_{N-1} \end{Vmatrix} \tag{22}$$

From this it is clear that a boundary-value problem for the second-order difference equation (20) represents a system of linear algebraic equations of special form. If the Cauchy problem for the second-order difference equation is soluble everywhere, then the first boundary-value problem (20) is soluble for any right-hand side whenever the matrix $\mathcal{A}$ of the system (21) is non-singular.

Boundary-value problems for m^{th} order difference equations lead to systems of linear algebraic equations with matrices which have no more than $m + 1$ non-zero elements in any row.

To approximate equations with partial derivatives, we also arrive at a system of difference or simply algebraic equations with a special matrix. Since the number of unknowns in such a system is usually equal to the number of nodes in the grid, in practice we encounter systems of very high order (having tens or even hundreds of thousands of unknowns). Other features of such systems are the sparsity of the matrices and the band structure, i.e., the special distribution of the non-zero elements. These features, on the one hand, make the problems easier to solve, but on the other hand, demand the invention of special solution methods which take into account the specifics of the problem. Thus it is not surprising that classical linear algebra methods are often ineffective for solving difference equations, and that there is no universal method which effectively solves every difference equation.

At the present time, two types of methods are used to solve systems of linear equations: 1) direct methods; 2) iterative or successive-approximation methods. As a rule, direct methods are oriented to solving a narrow class of grid equations, but they allow us to find the solution at very little computational expense. Iterative methods allow us to solve more complex equations and often contain direct methods as a basic step in the algorithm for solving special difference equations. The fact that difference equations are ill-conditioned requires us to develop rapidly-convergent iterative processes, and to isolate the region of applicability for each method.

In a number of cases, for example for linear equations with constant coefficients related to grid functions of one argument, the solution can be found in closed form. Such methods for solving grid equations will be examined in Section 3 of this chapter.

1.2 The general theory of linear difference equations

1.2.1 Properties of the solutions of homogeneous equations. In this section, we will consider the general theory of m^{th} order linear difference equations with variable coefficients

$$a_m(i)y(i+m) + \ldots + a_0(i)y(i) = f_i,$$

where $a_m(i)$ and $a_0(i)$ are non-zero for any i. First of all, let us investigate the homogeneous equation

$$a_m(i)y(i+m) + \ldots + a_0(i)y_i = \sum_{k=0}^{m} a_k(i)y(i+k) = 0. \tag{1}$$

We will assume that the coefficients $a_k(i)$, $i = 0, 1, \ldots, m$, are finite for all values of i.

Each particular solution of (1) is determined by the values of the function $y(i)$ at m arbitrary but sequential points $i_0, i_0 + 1, \ldots, i_0 + m - 1$.

Theorem 1. *If $v_1(i), v_2(i), \ldots, v_p(i)$ are solutions of equation (1), then the function*

$$y(i) = c_1 v_1(i) + c_2 v_2(i) + \ldots c_p v_p(i), \tag{2}$$

where $c_1, c_2, \ldots, c_p$ are arbitrary constants, is also a solution of equation (1).

Proof. In fact, the condition of the theorem guarantees that

$$\sum_{k=0}^{m} a_k(i)v_l(i+k) = 0, \quad l = 1, 2, \ldots, p. \tag{3}$$

We substitute (2) in (1):

$$\sum_{k=0}^{m} a_k(i)y(i+k) = \sum_{k=0}^{m} a_k(i) \sum_{l=1}^{p} c_l v_l(i+k)$$

and change the order of summation on the right-hand side of the equality. Using (3), we obtain

$$\sum_{k=0}^{m} a_k(i)y(i+k) = \sum_{l=1}^{p} c_l \sum_{k=0}^{m} a_k(i)v_l(i+k) = 0$$

and consequently the function $y(i)$ defined by (2) is also a solution of equation (1). The theorem is proved. □

We will denote by $\Delta_i(v_1, \ldots, v_p)$ the determinant

$$\Delta_i(v_1, v_2, \ldots, v_p) = \begin{vmatrix} v_1(i) & v_1(i+1) & \ldots & v_1(i+p-1) \\ v_2(i) & v_2(i+1) & \ldots & v_2(i+p-1) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ v(i) & v_p(i+1) & \ldots & v_p(i+p-1) \end{vmatrix}$$

Then

Lemma 2. *Let $v_1(i), v_2(i), \ldots, v_m(i)$ be solutions of equation (1). Then the determinant $\Delta_i(v_1, \ldots, v_m)$ is either identically zero for all i, or is non-zero for all possible values of i.*

Proof. Actually, since $v_1(i), \ldots, v_m(i)$ are solutions of equation (1), the following equalities are valid:

$$a_0(i)v_1(i) + a_1(i)v_1(i+1) + \ldots + a_{m-1}(i)v_1(i+m-1) = -a_m(i)v_1(i+m),$$
$$a_0(i)v_2(i) + a_2(i)v_2(i+1) + \ldots + a_{m-1}(i)v_2(i+m-1) = -a_m(i)v_2(i+m),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$a_0(i)v_m(i) + a_2(i)v_m(i+1) + \ldots + a_{m-1}(i)v_m(i+m-1) = -a_m(i)v_m(i+m),$$

Solving this system for $a_0(i)$ and for fixed i using Cramer's rule, we obtain

$$a_0(i)\Delta_i(v_1,\ldots,v_m) = -a_m(i)\begin{vmatrix} v_1(i+m) & v_1(i+1) & \ldots & v_1(i+m-1) \\ v_2(i+m) & v_2(i+1) & \ldots & v_2(i+m-1) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ v_m(i+m) & v_m(i+1) & \ldots & v_m(i+m-1) \end{vmatrix}.$$

After an appropriate rearrangement of the columns of the determinant on the right-hand side of this last equation, we have the relation

$$a_0(i)\Delta_i(v_1, \ldots, v_m) = (-1)^m a_m(i)\Delta_{i+1}(v_1, \ldots, v_m).$$

Since $a_0(i)$ and $a_m(i)$ are non-zero for all possible values of i, this proves the lemma. □

We now introduce the concept of linearly independent solutions of equation (1). The grid functions $v_1(i), v_2(i), \ldots, v_m(i)$ are called *linearly independent solutions of equation* (1) if: 1) they assume finite values and satisfy equation (1); 2) the relation

$$c_1v_1(i) + c_2v_2(i) + \ldots + c_mv_m(i) = 0 \tag{4}$$

for constants $c_1, c_2, \ldots, c_m$, not all zero, is not satisfied for any i.

For linearly independent solutions we have

Lemma 3. *If $v_1(i), v_2(i), \ldots, v_m(i)$ are linearly independent solutions of equation (1), then the determinant $\Delta_i(v_1, \ldots, v_m)$ is non-zero for all possible values of i. Conversely, if for solutions $v_1(i), \ldots, v_m(i)$ of equation (1) the determinant $\Delta_i(v_1, \ldots, v_m)$ is non-zero for some i, then $v_1(i), \ldots, v_m(i)$ are linearly independent solutions of equation (1).*

Proof. By lemma 2, the determinant $\Delta_i(v_1, \ldots, v_m)$ is either identically zero, or non-zero for all i. Let $v_1(i), \ldots, v_m(i)$ be linearly independent solutions to equation (1), and assume that $\Delta_i(v_1, \ldots, v_m) \equiv 0$. Consider the system of equations

$$\begin{aligned} &c_1 v_1(i_0) + c_2 v_2(i_0) + \ldots + c_m v_m(i_0) = 0. \\ &c_1 v_1(i_0+1) + c_2 v_2(i_0+1) + \ldots + c_m v_m(i_0+1) = 0. \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &c_1 v_1(i_0+m-1) + c_2 v_2(i_0+m-1) + \ldots + c_m v_m(i_0+m-1) = 0. \end{aligned} \tag{5}$$

Since the determinant $\Delta_i(v_1, \ldots, v_m)$ of this system is, by assumption, equal to zero, there exists a non-zero solution $c_1, c_2, \ldots, c_m$ to this system. Consequently, for these $c_1, c_2, \ldots, c_m$, equation (4) is valid for $i = i_0, i_0+1, \ldots, i_0+m-1$. We now show that (4) is valid for $i = i_0+m$. For this, we take equation (1) with $l = 1, 2, \ldots, m$

$$\sum_{k=0}^{m} a_k(i_0) v_l(i_0+k) = 0,$$

multiply it by c_l and sum for $l = 1, 2, \ldots, m$. Using (5) we obtain

$$\begin{aligned} 0 &= a_m(i_0) \sum_{l=1}^{m} c_l v_l(i_0+m) + \sum_{k=0}^{m-1} a_k(i_0) \sum_{l=1}^{m} c_l v_l(i_0+k) \\ &= a_m(i_0) \sum_{l=1}^{m} c_l v_l(i_0+m). \end{aligned}$$

Thus we have shown the validity of (4) for $i = i_0 + m$. Proceeding in the same fashion, we find that, for above choice of $c_1, c_2, \ldots, c_m$, equation (4) is satisfied for all $i \geq i_0$. The validity of (4) is analogously proved for $i \leq i_0$. Consequently, (4) is satisfied for all i with non-zero $c_1, c_2, \ldots, c_m$, contradicting the linear independence of $v_1(i), \ldots, v_m(i)$. Therefore the assumption that the determinant $\Delta_i(v_1, \ldots, v_m)$ is identically zero is false.

We will now prove the second part of lemma 3. Suppose that the determinant $\Delta_i(v_1, \ldots, v_m)$ is non-zero for some $i = i_0$. Then let us assume that $v_1(i), v_2(i), \ldots, v_m(i)$ is a system of linearly independent solutions to (1). This implies that we can find constants $c_1, c_2, \ldots, c_m$, not all zero, so that equation

(4) is an identity for all i. Then we write (4) for $i = i_0, i_0 + 1, \dots, i_0 + m - 1$ in the form of the system (5). By force of the assumption in the lemma, the determinant $\Delta_i(v_1, \dots, v_m)$ of this system is non-zero. Therefore, all of the $c_1, c_2, \dots, c_m$ must be equal to zero. We have arrived at a contradiction. The lemma is proved. □

1.2.2 Theorems about the solutions of linear equations. First we will prove a theorem about the general solution of the homogeneous linear equation (1).

Theorem 2. *If $v_1(i), v_2(i), \dots, v_m(i)$ are linearly independent solutions of equation (1), then the general solution of this equation has the form*

$$y(i) = c_1 v_1(i) + c_2 v_2(i) + \dots + c_m v_m(i), \tag{6}$$

where $c_1, c_2, \dots, c_m$ are arbitrary constants.

Proof. In fact, by theorem 1 the function $y(i)$ defined by formula (6) is a solution to equation (1). We will now show that all solutions of equation (1) are of this form. Let $u(i)$ be an arbitrary solution of equation (1). It is fully determined by the initial values given at the m points $u(i_0), u(i_0 + 1), \dots, u(i_0 + m - 1)$. From the set of all functions of the form (6), we choose the one which has these same initial values. To do this, it is sufficient to find the constants $c_1, c_2, \dots, c_m$ which satisfy the m equations

$$\begin{aligned}
&c_1 v_1(i_0) + c_2 v_2(i_0) + \dots + c_m v_m(i_0) = u(i_0),\\
&c_1 v_1(i_0+1) + c_2 v_2(i_0+1) + \dots + c_m v_m(i_0+1) = u(i_0+1),\\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\\
&c_1 v_1(i_0+m-1) + c_2 v_2(i_0+m-1) + \dots + c_m v_m(i_0+m-1) = u(i_0+m-1).
\end{aligned}$$

Since $v_1(i), v_2(i), \dots, v_m(i)$ are linearly independent solutions of (1), by lemma 3 the determinant $\Delta_i(v_1, \dots, v_m)$ of this system is non-zero. Having solved this system for $c_1, c_2, \dots, c_m$, we obtain the function $y(i)$ having the same initial values as $u(i)$. But since the initial values fully determine the solution to equation (1), $y(i) \equiv u(i)$. The theorem is proved. □

We now consider the solution of the non-homogeneous equation

$$a_m(i) y(i + m) + \dots + a_0(i) y(i) = f(i). \tag{7}$$

We have

Theorem 3. *The general solution to equation (7) is the sum of a particular solution to (7) and the general solution to the homogeneous equation (1).*

Proof. We will show that any solution to equation (7) can be represented in the form

$$y(i) = \bar{y}(i) + \overline{\overline{y}}(i), \tag{8}$$

where $\bar{y}(i)$ is some solution to equation (7), and $\overline{\overline{y}}(i)$ is the general solution to equation (1). Suppose

$$a_m(i)\bar{y}(i+m) + \ldots + a_0(i)\bar{y}(i) = f(i). \tag{9}$$

Substituting (8) in (7) and using (9), we obtain

$$a_m(i)\overline{\overline{y}}(i+m) + \ldots + a_0\overline{\overline{y}}(i) = 0.$$

Consequently, $\overline{\overline{y}}(i)$ is the general solution to the homogeneous equation (1). The theorem is proved. □

Corollary 1. *From theorems 2 and 3 it follows that the general solution to the non-homogeneous equation (7) has the form*

$$y(i) = \bar{y}(i) + c_1 v_1(i) + \ldots + c_m v_m(i), \tag{10}$$

where $\bar{y}(i)$ is a particular solution to equation (7), $v_1(i), v_2(i), \ldots, v_m(i)$ are linearly independent solutions to equation (1), and $c_1, \ldots, c_m$ are arbitrary constants.

Corollary 2. *Using lemma 3, Corollary 1 can be reformulated as: the solution to equation (7) has the form (10) where the particular solutions $v_1(i)$, $\ldots, v_m(i)$ of the homogeneous equation are such that $\Delta_i(v_1, \ldots, v_m) \neq 0$ for some value i.*

Corollary 3. *If the right-hand side $f(i)$ of equation (7)is the sum of two functions $f(i) = f^{(1)}(i) + f^{(2)}(i)$, then a particular solution of equation (7) can be written in the form $y(i) = \bar{y}^{(1)}(i) + \bar{y}^{(2)}(i)$, where $\bar{y}^{(\alpha)}(i)$ is a particular solution to equation (7) with right-hand side $f^{(\alpha)}(i)$, $\alpha = 1, 2$.*

1.2.3 The method of variation of parameters. The above theorems give the structure of the general solution of the linear non-homogeneous difference equation (7). We will now consider the following questions:

[1] how to construct linearly independent solutions to the homogeneous equation;

[2] how to find a particular solution to the non-homogeneous equation;

[3] how, using the general solution to the non-homogeneous equation, to find a unique solution to equation (7) satisfying additional conditions.

First of all we will study one possible way of constructing linearly independent solutions to the homogeneous equation. Since a particular solution to the m^{th} order linear equation is fully determined by given initial values at m points, for example at $i = i_0, i_0 + 1, \ldots, i_0 + m - 1$, then by lemma 3 the desired solution of equation (1) can be constructed in the following way. Let A be a non-singular matrix

$$A = \begin{Vmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mm} \end{Vmatrix}.$$

Construct m solutions to equation (1) $v_1(i), v_2(i), \ldots, v_m(i)$ using the initial values

$$v_l(i_0 + k - 1) = a_{lk}, \qquad l, k = 1, 2, \ldots, m. \tag{11}$$

Then $\Delta_{i_0}(v_1, \ldots, v_m) = \det A \neq 0$. Consequently, the problem of constructing the desired functions $v_1(i), \ldots, v_m(i)$ is solved.

Let us now consider the question of constructing a unique solution from the family of solutions to (10). From (10) it follows that we require precisely m conditions on the function $y(i)$ in order to determine the constants $c_1, c_2, \ldots, c_m$.

In the case of a Cauchy problem, i.e., when the initial conditions $y(i_0) = b_1, y(i_0 + 1) = b_2, \ldots, y(i_0 + m - 1) = b_m$ are given, it is easy to determine the constants $c_1, c_2, \ldots, c_m$. From (10) we obtain a system of linear algebraic equations in $c_1, c_2, \ldots, c_m$:

$$\begin{gathered} v_1(i_0)c_1 + v_2(i_0)c_2 + \ldots + v_m(i_0)c_m = b_1 - \bar{y}(i_0), \\ v_1(i_0 + 1)c_1 + v_2(i_0 + 1)c_2 + \ldots + v_m(i_0 + 1)c_m = b_2 - \bar{y}(i_0 + 1), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ v_1(i_0+m-1)c_1+v_2(i_0+m-1)c_2+\ldots+v_m(i_0+m-1)c_m = b_m-\bar{y}(i_0+m-1). \end{gathered} \tag{12}$$

Since the determinant $\Delta_{i_0}(v_1, \ldots, v_m)$ of this system is non-zero, this system has a unique solution $c_1, c_2, \ldots, c_m$, which fully determines the unique solution to the non-homogeneous equation (7).

In the case of a boundary-value problem, when the m auxiliary conditions for $y(i)$ are given at non-sequential points, we again obtain a system of linear algebraic equations in $c_1, c_2, \ldots, c_m$. But in this case, the solution of this system will exist only with additional assumptions about the coefficients of the difference equation.

Let us now consider the question of solving the equations (12). Since by (11) the matrix of the system (12) is A^T, then, choosing A as the identity matrix, we obtain the solution to the system (12) explicitly: $c_l = b_l - \bar{y}(i_0 + l - 1)$, $l = 1, 2, \ldots, m$. It is obvious that, from among all solutions to the non-homogeneous equation (7), it is possible to select the one for which $\bar{y}(i_0) = \bar{y}(i_0 + 1) = \ldots = \bar{y}(i_0 + m - 1) = 0$. Then we will have $c_l = b_l$, $l = 1, 2, \ldots, m$. Such a choice for the matrix A corresponds to the following initial values for $v_1(i), \ldots, v_m(i)$:

$$v_l(i_0 + l - 1) = 1,\ v_l(i_0 + k - 1) = 0,\ k = 1, 2, \ldots, m,\ k \neq l,\ l = 1, 2, \ldots, m.$$

We will now find particular solutions to the non-homogeneous equation, given m linearly independent solutions to the homogeneous equation. Let us consider finding a particular solution *with variable coefficients* in the general solution of the homogeneous equation.

Previously it was shown that the general solution of the homogeneous equation (1) has the form

$$\bar{\bar{y}}(i) = c_1 v_1(i) + \ldots + c_m v_m(i),$$

where $v_1(i), \ldots, v_m(i)$ are linearly independent solutions to equation (1), and $c_1, c_2, \ldots, c_m$ are arbitrary constants. We will now let $c_1, c_2, \ldots, c_m$ be functions of i and consider the problem of choosing them so that the function

$$\bar{y}(i) = c_1(i) v_1(i) + \ldots + c_m(i) v_m(i) \tag{13}$$

is a particular solution to the non-homogeneous equation (7). Notice that each function $c_l(i)$ is determined only up to a constant, since $v_l(i)$ is a solution of the homogeneous equation:

$$a_m(i) v_l(i + m) + \ldots + a_0(i) v_l(i) = 0, \qquad l = 1, 2, \ldots, m. \tag{14}$$

We introduce the following notation:

$$d_k(i) = \sum_{l=1}^{m} [c_l(i + k) - c_l(i)] v_l(i + k), \qquad k = 0, 1, \ldots, m.$$

Substituting (13) in (7), making the above substitution in the resulting

expression, and taking into account (14), we obtain

$$\begin{aligned} f(i) &= \sum_{k=0}^{m} a_k(i)\bar{y}(i+k) = \sum_{k=0}^{m} a_k(i) \sum_{l=1}^{m} c_l(i+k)v_l(i+k) \\ &= \sum_{k=0}^{m} a_k(i)d_k(i) + \sum_{k=0}^{m} a_k(i) \sum_{l=1}^{m} c_l(i)v_l(i+k) \\ &= \sum_{k=0}^{m} a_k(i)d_k(i) + \sum_{l=1}^{m} c_l(i) \left[\sum_{k=0}^{m} a_k(i)v_l(i+k) \right] \\ &= \sum_{k=0}^{m} a_k(i)d_k(i) = \sum_{k=1}^{m} a_k(i)d_k(i), \end{aligned}$$

since $d_0(i) \equiv 0$. The resulting relation is satisfied for all i if we set

$$d_k(i) = 0, \quad k = 1, 2, \ldots, m-1, \quad d_m(i) = f(i)/a_m(i). \tag{15}$$

Thus, the problem of constructing functions $c_1(i), c_2(i), \ldots, c_m(i)$ leads to the conditions (15), which must be satisfied identically for all i.

Let us transform the system of equations (15). We designate $b_l(i) = c_l(i+1) - c_l(i)$, $l = 1, 2, \ldots, m$. From the definition of $d_k(i)$, we obtain for $k = 1, 2, \ldots, m$:

$$\begin{aligned} d_k(i) - d_{k-1}(i+1) &= \sum_{l=1}^{m} [c_l(i+k) - c_l(i)]v_l(i+k) \\ &\quad - \sum_{l=1}^{m} [c_l(i+k) - c_l(i+1)]v_l(i+k) \\ &= \sum_{l=1}^{m} b_l(i)v_l(i+k). \end{aligned}$$

Substituting (15) and taking into account the equation $d_0(i) = 0$, we obtain a system of linear algebraic equations in $b_l(i)$ for fixed i:

$$\begin{aligned} &b_1(i)v_1(i+1) + b_2(i)v_2(i+1) + \ldots + b_m(i)v_m(i+1) = 0, \\ &b_1(i)v_1(i+2) + b_2(i)v_2(i+2) + \ldots + b_m(i)v_m(i+2) = 0, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &b_1(i)v_1(i+m) + b_2(i)v_2(i+m) + \ldots + b_m(i)v_m(i+m) = \frac{f(i)}{a_m(i)}. \end{aligned} \tag{16}$$

The determinant of the system (16) is equal to $\Delta_{i+1}(v_1, v_2, \ldots, v_m)$ and it is non-zero because of the linear independence of $v_1, v_2, \ldots, v_m$. Therefore the system (16) has the unique solution

$$b_l(i) = c_l(i+1) - c_l(i) = (-1)^{m+l} \frac{f(i)}{a_m(i)} \frac{\mathcal{D}_l(i)}{\mathcal{D}(i)}, l = 1, \ldots, m, \qquad (17)$$

where $\mathcal{D}(i) = \Delta_{i+1}(v_1, v_2, \ldots, v_m)$, and

$$\mathcal{D}_l(i) = \begin{vmatrix} v_1(i+1) & \ldots & v_{l-1}(i+1) & v_{l+1}(i+1) & \ldots & v_m(i+1) \\ v_1(i+2) & \ldots & v_{l-1}(i+2) & v_{l+1}(i+2) & \ldots & v_m(i+2) \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ v_1(i+m-1) & \ldots & v_{l-1}(i+m-1) & v_{l+1}(i+m-1) & \ldots & v_m(i+m-1) \end{vmatrix},$$

i.e., $\mathcal{D}_l(i)$ is obtained from the determinant $\mathcal{D}(i)$ by deleting the l^{th} column and the last row.

The equalities (17) are first-order difference equations in the functions $c_l(i)$, $l = 1, 2, \ldots, m$. Since $c_l(i)$ is only determined up to a constant, from (17) we find an explicit representation for $c_l(i)$:

$$c_l(i) = \sum_{j=i_0}^{i-1} (-1)^{m+l} \frac{f(j)}{a_m(j)} \frac{\mathcal{D}_l(j)}{\mathcal{D}(j)}, \quad l = 1, 2, \ldots, m.$$

Substituting this expression in (13) and changing the order of the summation in the resulting expression, we obtain the following formula for the particular solution $\bar{y}(i)$ of the non-homogeneous equation (7):

$$\begin{aligned} \bar{y}(i) &= \sum_{l=1}^{m} c_l(i) v_l(i) \\ &= \sum_{j=i_0}^{i-1} \left[f(j) \sum_{l=1}^{m} (-1)^{m+l} \mathcal{D}_l(j) v_l(i) \right] \Big/ \left(\mathcal{D}(j) a_m(j) \right) \\ &= \sum_{j=i_0}^{i-1} G(i,j) f(j), \end{aligned}$$

where

$$G(i,j) = \frac{1}{\mathcal{D}(j) a_m(j)} \sum_{k=1}^{m} (-1)^{m+k} \mathcal{D}_k(j) v_k(i). \qquad (18)$$

Notice that the sum in (18) is easy to compute

$$\sum_{k=1}^{m}(-1)^{m+k}\mathcal{D}_k(j)v_k(i) = \begin{vmatrix} v_1(j+1) & v_2(j+1) & \dots & v_m(j+1) \\ v_1(j+2) & v_2(j+2) & \dots & v_m(j+2) \\ \dots & \dots & \dots & \dots \\ v_1(j+m-1) & v_2(j+m-1) & \dots & v_m(j+m-1) \\ v_1(i) & v_2(i) & \dots & v_m(i) \end{vmatrix}.$$

This sum is equal to zero for $j = i-1, i-2, \dots, i-m+1$. Thus the particular solution to equation (7) has the following form

$$\bar{y}(i) = \sum_{j=i_0}^{i-m} \frac{\begin{vmatrix} v_1(j+1) & \dots & v_m(j+1) \\ \dots & \dots & \dots \\ v_1(j+m-1) & \dots & v_m(j+m-1) \end{vmatrix}}{\begin{vmatrix} v_1(j+1) & \dots & v_1(j+m) \\ \dots & \dots & \dots \\ v_m(j+1) & \dots & v_m(j+m) \end{vmatrix}} \frac{f(j)}{a_m(j)}, \tag{19}$$

where i_0 is arbitrary, and $\bar{y}(i) = 0$ for $i = i_0, i_0+1, \dots, i_0+m-1$.

For first-order equations ($m = 1$), the formula (19) takes the following form:

$$\bar{y}(i) = \sum_{j=i_0}^{i-1} \frac{v_1(i)}{v_1(j+1)} \cdot \frac{f(j)}{a_1(j)}, \quad \bar{y}(i_0) = 0. \tag{20}$$

1.2.4 Examples. We now consider several examples illustrating the application of the general theory. Suppose we must find the general solution of the first-order equation

$$y(i+1) - e^{2i}y(i) = 6i^2e^{i^2+i}. \tag{21}$$

We first find the solution of the homogeneous equation

$$y(i+1) - e^{2i}y(i) = 0. \tag{22}$$

From (22) we sequentially obtain

$$y(i+1) = e^{2i}y(i) = e^{2i}e^{2(i-1)}y(i-1) = \dots = e^{2\Sigma_{k=1}^{i}k}y(1) = e^{i(i+1)}y(1).$$

Setting $y(1) = 1$, we find the particular solution $v_1(i)$ of the homogeneous equation (22) in the form $v_1(i) = e^{i(i-1)}$. Consequently, the general solution of the homogeneous equation has the form $y(i) = ce^{i(i-1)}$, where c is an arbitrary constant.

We now construct a particular solution of the non-homogeneous equation (21) using (20). From (20) we obtain

$$\bar{y}(i) = \sum_{k=i_0}^{i-1} \frac{e^{i(i-1)}}{e^{k(k+1)}} \cdot \frac{6k^2 e^{k^2+k}}{1} = 6e^{i(i-1)} \sum_{k=i_0}^{i-1} k^2.$$

Since i_0 can be chosen arbitrarily, setting $i_0 = 1$ we get $\bar{y}(i) = i(i-1)(2i-1)e^{i(i-1)}$. Further, by theorem 3 the general solution to (21) can be written in the form

$$y(i) = \bar{y}(i) + \overline{\bar{y}}(i) = [c + i(i-1)(2i-1)]e^{i(i-1)}$$

where c is an arbitrary constant. The problem is solved.

Let us now find the general solution of the second-order equation

$$a_2(i)y(i+2) + a_1(i)y(i+1) + a_0(i)y(i) = f(i), \tag{23}$$

where $i = 0, 1, 2, \ldots,$

$$\begin{aligned} a_2(i) &= i^2 - i + 1, \\ a_0(i) &= a_2(i+1) = i^2 + i + 1, \\ a_1(i) &= -a_0(i) - a_2(i) = -2(i^2+1), \\ f(i) &= 2^i(i^2 - 3i + 1) = 2^i[2a_2(i) - a_0(i)]. \end{aligned} \tag{24}$$

Since the coefficients $a_2(i)$ and $a_0(i)$ are non-null, we can apply the general theory to find the general solution to (23).

First we construct linearly independent solutions to the homogeneous equation. Using (24), it can be written in the following form:

$$a_2(i)y(i+2) - [a_2(i) + a_2(i+1)]y(i+1) + a_2(i+1)y(i) = 0$$

or

$$a_2(i)[y(i+2) - y(i+1)] - a_2(i+1)[y(i+1) - y(i)] = 0. \tag{25}$$

The particular solutions $v_1(i)$ and $v_2(i)$ to the homogeneous equation (25) are determined by the following conditions: $v_1(0) = v_1(1) = 1$, $v_2(0) = 0$, $v_2(1) = 3$. Since the determinant satisfies

$$\Delta_0(v_1, v_2) = \begin{vmatrix} v_1(0) & v_1(1) \\ v_2(0) & v_2(1) \end{vmatrix} = 3 \neq 0,$$

by lemma 3 the functions $v_1(i)$ and $v_2(i)$ are linearly independent solutions to equation (25).

We now find $v_1(i)$ and $v_2(i)$ explicitly. From (25) it immediately follows that $v_1(i) \equiv 1$. Let us construct $v_2(i)$. From (25) we sequentially obtain

$$\begin{aligned} y(i+2) - y(i+1) &= \frac{a_2(i+1)}{a_2(i)}[y(i+1) - y(i)] \\ &= \frac{a_2(i+1)}{a_2(i-1)}[y(i) - y(i-1)] \\ &= \ldots = \frac{a_2(i+1)}{a_2(0)}[y(1) - y(0)]. \end{aligned}$$

Taking into account the initial values for $v_2(i)$, we obtain

$$v_2(i+1) - v_2(i) = 3a_2(i) = 3(i^2 - i + 1). \tag{26}$$

Summing the left- and right-hand sides of (26) for i between 0 and $k-1$, we get

$$v_2(k) = v_2(0) + 3\sum_{i=0}^{k-1}(i^2 - i + 1) = k(k^2 - 3k + 5).$$

Thus, the particular solutions of the homogeneous equation (25) are

$$v_1(k) \equiv 1, \qquad v_2(k) = k(k^2 - 3k + 5), \tag{27}$$

and the general solution (25) has the form

$$\overline{\overline{y}}(k) = c_1 + c_2 k(k^2 - 3k + 5).$$

We now construct a particular solution to the non-homogeneous equation (23). Substituting (24) and (27) into the formula (19), we obtain

$$\begin{aligned} \bar{y}(i) &= \sum_{k=0}^{i-2} \frac{v_2(i) - v_2(k+1)}{v_2(k+2) - v_2(k+1)} \cdot \frac{f(k)}{a_2(k)} \\ &= \sum_{k=0}^{i-2} \frac{v_2(i) - v_2(k+1)}{3a_2(k+1)a_2(k)}[2^{k+1}a_2(k) - 2^k a_2(k+1)]. \\ &= \frac{1}{3}\sum_{k=0}^{i-2}[v_2(i) - v_2(k+1)]\left[\frac{2^{k+1}}{a_2(k+1)} - \frac{2^k}{a_2(k)}\right]. \end{aligned} \tag{28}$$

Here (26) was used.

We now calculate the resulting expression. Denoting

$$v(k) = v_2(i) - v_2(k+1), \qquad u(k) = \frac{2^k}{a_2(k)},$$

we write (28) in the following form:

$$\bar{y}(i) = \frac{1}{3}\sum_{k=0}^{i-2}[u(k+1) - u(k)]v(k).$$

We will use now the formula for summation by parts (cf. (8) from Section 1) for a uniform grid with step $h = 1$. This gives

$$\bar{y}(i) = -\frac{1}{3}\sum_{k=0}^{i-1} u(k)[v(k) - v(k-1)]$$
$$+ \frac{1}{3}[u(i-1)v(i-1) - u(0)v(-1)].$$

Using (26), the condition $v_2(0) = 0$, and the definitions of the functions $v(k)$ and $u(k)$, we get

$$v(k) - v(k-1) = v_2(k) - v_2(k+1) = -3a_2(k),$$
$$v(i-1) = v_2(i) - v_2(i) = 0,$$
$$v(-1) = v_2(i) - v_2(0) = v_2(i),$$

and hence

$$\bar{y}(i) = \sum_{k=0}^{i-1} 2^k - \frac{1}{3}v_2(i) = 2^i - 1 - \frac{1}{3}i(i^2 - 3i + 5).$$

Consequently, the particular solution of (23) is found. By theorem 3, the general solution of the second-order non-homogeneous equation (23) has the form

$$\begin{aligned} y(i) &= \bar{y}(i) + \bar{\bar{y}}(i) \\ &= 2^i - 1 - \frac{1}{3}i(i^2 - 3i + 5) + c_1 + c_2 i(i^2 - 3i + 5) \\ &= \bar{c}_1 + 2^i + \bar{c}_2 i(i^2 - 3i + 5), \end{aligned}$$

where $\bar{c}_1 = c_1 - 1$, $\bar{c}_2 = c_2 - \frac{1}{3}$ are arbitrary constants. The problem is solved.

1.3 The solution of linear equations with constant coefficients

1.3.1 The characteristic equation. The simple-roots case. We now consider an important class of difference equations — linear equations with constant coefficients. For this class of equations, it is quite simple to find linearly-independent solutions to the corresponding homogeneous equations. This, as was shown above, leads to the problem of solving a non-homogeneous difference equation.

We wish to find linearly independent solutions of the m^{th} order homogeneous linear equation with constant coefficients

$$a_m y(i+m) + a_{m-1} y(i+m-1) + \ldots + a_0 y(i) = 0. \tag{1}$$

We will search for particular solutions of (1) in the form $v(i) = q^i$, where the number q remains to be chosen. Substituting $v(i)$ instead of $y(i)$ in (1), we obtain

$$q^i(a_m q^m + a_{m-1} q^{m-1} + \ldots + a_1 q + a_0) = 0.$$

Since we are searching for a non-zero solution to (1), we divide by q^i, and obtain the following equation for q:

$$a_m q^m + a_{m-1} q^{m-1} + \ldots + a_1 q + a_0 = 0. \tag{2}$$

Equation (2) is called the *characteristic equation* for (1). The roots $q_1, q_2, \ldots, q_m$ of equation (2) can be simple or multiple. We will consider each case separately.

Suppose the roots are simple. We will show that the functions

$$v_1(i) = q_1^i, \qquad v_2(i) = q_2^i, \ldots, \; v_m(i) = q_m^i \tag{3}$$

are linearly independent solutions to equation (1).

Actually, by lemma 3, it is sufficient to show that for some i the determinant $\Delta_i(v_1, v_2, \ldots, v_m)$ is non-zero. Setting $i = 0$ we find

$$\Delta_0(v_1, \ldots, v_m) = \begin{vmatrix} 1 & q_1 & q_1^2 & \ldots & q_1^{m-1} \\ 1 & q_2 & q_2^2 & \ldots & q_2^{m-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & q_m & q_m^2 & \ldots & q_m^{m-1} \end{vmatrix} = \begin{vmatrix} 1 & 1 & \ldots & 1 \\ q_1 & q_2 & \ldots & q_m \\ q_1^2 & q_2^2 & \ldots & q_m^2 \\ \ldots & \ldots & \ldots & \ldots \\ q_1^{m-1} & q_2^{m-1} & \ldots & q_m^{m-1} \end{vmatrix}$$

and consequently $\Delta_0(v_1, \ldots, v_m)$ is the Vandermonde determinant. It is non-zero because all q_k are distinct. Thus, the functions (3) are in fact linearly

independent solutions to (1), and therefore the general solution of the homogeneous equation (1) can be written in the form

$$y(i) = c_1 q_1^i + c_2 q_2^i + \ldots + c_m q_m^i, \tag{4}$$

where $c_1, c_2, \ldots, c_m$ are arbitrary constants.

If the roots $q_1, q_2, \ldots, q_m$ are real, then a real-valued solution $y(i)$ corresponds to a choice of the constants $c_1, c_2, \ldots, c_m$ as real numbers. We now consider how to obtain a real-valued solution when there are complex roots.

Let

$$q_n = \rho(\cos\varphi + i^* \sin\varphi), \qquad (i^* = \sqrt{-1})$$

be a complex root of the characteristic equation (2). Then

$$q_s = \rho(\cos\varphi - i^* \sin\varphi)$$

the conjugate of q_n, is also a root of equation (2). Consider the part of the general solution (4) formed by a linear combination of q_n^i and q_s^i:

$$y(i) = c_n q_n^i + c_s q_s^i = \rho^i[(c_n + c_s)\cos i\varphi + i^*(c_n - c_s)\sin i\varphi].$$

The function $y(i)$ will be real-valued if the constants c_n and c_s are complex conjugates. Setting

$$c_n = 0.5(\bar{c}_n - i^* \bar{c}_s), \qquad c_s = 0.5(\bar{c}_n + i^* \bar{c}_s),$$

where $\bar{c}_n$ and $\bar{c}_s$ are arbitrary real numbers, we obtain $y(i) = \rho^i(\bar{c}_n \cos i\varphi + \bar{c}_s \sin i\varphi)$.

1.3.2 The multiple-root case. Suppose now that the characteristic equation (2) has a root q_1 of multiplicity n_1, a root q_2 of multiplicity n_2, etc., i.e., $q_1, q_2, \ldots, q_s$ are roots of multiplicity $n_1, n_2, \ldots, n_s$ respectively, and $n_1 + n_2 + \ldots + n_s = m$. We will construct linearly independent solutions of the homogeneous equation (1). For this we require

Lemma 4. *If q_l is a root of the characteristic equation* (2) *having multiplicity n_l, then*

$$\sum_{k=0}^{m} a_k k^p q_l^k = 0, \qquad p = 0, 1, \ldots, n_l - 1. \tag{5}$$

Proof. Actually, since q_l is a root of equation (2) of multiplicity n_l, we have

$$\sum_{k=0}^{m} a_k q_l^k = 0, \tag{6}$$

$$\sum_{k=0}^{m} k(k-1)\dots(k-s+1)a_k q_l^k = 0, \quad s = 1,2,\dots,n_l - 1, \tag{7}$$

which was obtained from (2) by differentiating s times and then multiplying the result by q_l^s. We shall show that (5) is equivalent to (6), (7). Obviously, it is only necessary to prove the equivalence of (7) and (5) for $p \geq 1$.

Since $P_s(k) = k(k-1)\dots(k-s+1)$ is a polynomial of degree s in k, multiplying (5) by the corresponding coefficient of the polynomial $P_s(k)$ for $p = 1,2,\dots,s$ and summing the results gives us equation (7).

We will now show that (5) follows from $p = 1,2,\dots,n_l - 1$. We use the expansion for k^p:

$$k^p = \sum_{s=1}^{p} k(k-1)\dots(k-s+1)\alpha_s, \qquad 1 \leq p \leq k, \tag{8}$$

where $\alpha_s = \alpha_s(p)$ will be defined below. We multiply the s^{th} equation in (7) by α_s and sum for s betweeen 1 and p. Using (8) we obtain

$$\begin{aligned} 0 &= \sum_{s=1}^{p} \alpha_s \left(\sum_{k=0}^{m} k(k-1)\dots(k-s+1)a_k q_l^k \right) \\ &= \sum_{k=0}^{m} a_k q_l^k \left(\sum_{s=1}^{p} k(k-1)\dots(k-s+1)\alpha_s \right) = \sum_{k=0}^{m} a_k k^p q_l^k. \end{aligned}$$

It remains to justify the expansion (8). Notice that both sides of (8) are p^{th} degree polynomials in k. If we set $\alpha_p = 1$, then the coefficients of highest degree will be equal, and the coefficients of lower degree are zero. We find $\alpha_1, \alpha_2, \dots, \alpha_{p-1}$ by equating the values of the polynomials at $p-1$ points, for example, setting $k = 1,2,\dots,p-1$. For $k = 1$ this gives $\alpha_1 = 1$. For $k = n, 2 \leq n \leq p-1$ we have

$$\begin{aligned} n^p = \sum_{s=1}^{p} n(n-1)\dots(n-s+1)\alpha_s &= \sum_{s=1}^{n} n(n-1)\dots(n-s+1)\alpha_s \\ &= n!\alpha_n + n! \sum_{s=1}^{n-1} \frac{\alpha_s}{(n-s)!}. \end{aligned}$$

Hence it is possible to find α_n if $\alpha_1, \alpha_2, \ldots, \alpha_{n-1}$ are already determined. Thus, we obtain the following recurrence relation for the coefficients α_n:

$$\alpha_n = \frac{n^p}{n!} - \sum_{s=1}^{n-1} \frac{\alpha_s}{(n-s)!}, \quad n = 2, 3, \ldots, p-1, \quad \alpha_1 = 1.$$

The lemma is proved. □

Using lemma 4, we now find m particular solutions to the homogeneous equation (1). Since

$$(j+k)^n = \sum_{p=0}^{n} c_n^p k^p j^{n-p}, \quad c_n^p = \frac{n!}{p!(n-p)!},$$

multiplying (5) by $c_n^p j^{n-p} q_l^j$ and summing for p between zero and $n \leq n_l - 1$, we find that for any j,

$$\sum_{k=0}^{m} a_k (j+k)^n q_l^{k+j} = 0, \qquad n = 0, 1, \ldots, n_l - 1.$$

Using this, it is easy to show that the grid functions

$$v_{n_1+n_2+\ldots+n_{l-1}+n+1}(j) = j^n q_l^j, \quad 0 \leq n \leq n_l - 1, \quad l = 1, 2, \ldots, s, \qquad (9)$$

are particular solutions of the homogeneous equations (1), i.e., if q_l is a root of the characteristic equation of multiplicity n_l, then the functions

$$q_l^j, \; j q_l^j, \ldots, j^{n_l - 1} q_l^j, \quad l = 1, 2, \ldots, s$$

are solutions to (1).

It remains to prove that the functions $v_1(j), \ldots, v_m(j)$ defined in (9) are linearly independent solutions. For this, we compute the determinant $\Delta_0(v_1, \ldots, v_m)$, which in this case has the form

$$\Delta_0(v_1, \ldots, v_m) = \begin{vmatrix} 1 & q_1 & q_1^2 & \ldots & q_1^k & \ldots & q_1^{m-1} \\ 0 & q_1 & 2q_1^2 & \ldots & kq_1^k & \ldots & (m-1)q_1^{m-1} \\ 0 & q_1 & 2^2 q_1^2 & \ldots & k^2 q_1^k & \ldots & (m-1)^2 q_1^{m-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & q_2 & q_2^2 & \ldots & q_2^k & \ldots & q_2^{m-1} \\ 0 & q_2 & 2q_2^2 & \ldots & kq_2^k & \ldots & (m-1)q_2^{m-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & q_s & 2^n s^{-1} q_s^2 & \ldots & k^n s^{-1} q_s^k & \ldots & (m-1)^n s^{-1} q_s^{m-1} \end{vmatrix}.$$

It can be obtained directly from the Vandermonde determinant

$$W(x_1, x_2, \ldots, x_m) = \begin{vmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{m-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{m-1} & x_{m-1}^2 & \ldots & x_{m-1}^{m-1} \\ 1 & x_m & x_m^2 & \ldots & x_m^{m-1} \end{vmatrix} = \prod_{i=1}^{m-1} \prod_{j=i+1}^{m} (x_j - x_i)$$

in the following way. We take the first derivative of W along x_2, and multiply it by x_2. We denote the result by $W_2 = x_2 \frac{\partial W}{\partial x_2}$. We further compute

$$W_3 = x_3 \frac{\partial}{\partial x_3} \left(x_3 \frac{\partial W_2}{\partial x_3} \right), \quad W_4 = x_4 \frac{\partial}{\partial x_4} \left(x_4 \frac{\partial}{\partial x_4} \left(x_4 \frac{\partial W_3}{\partial x_4} \right) \right), \ldots$$

etc., until we obtain W_n. Then we compute

$$W_{n_1+2} = x_{n_1+2} \frac{\partial W_{n_1}}{\partial x_{n_1+2}},$$

and continue the process of differentiation, computing

$$W_{n_1+3} = x_{n_1+3} \frac{\partial}{\partial x_{n_1+3}} \left(x_{n_1+3} \frac{\partial W_{n_1+2}}{\partial x_{n_1+3}} \right),$$

until we obtain $W_{n_1+n_2}$, etc. In the end we obtain $W_m = W_m(x_1, x_2, \ldots, x_m)$. We now set

$$x_1 = x_2 = \ldots = x_{n_1} = q_1, \quad x_{n_1+1} = x_{n_1+2} = \ldots = x_{n_1+n_2} = q_2, \text{ etc.}$$

It is easy to verify that $\Delta_0(v_1, v_2, \ldots, v_m) = W_m$, and simple calculations give

$$W_m = \prod_{k=1}^{s} \prod_{m=1}^{n_k-1} m! q_k^m \prod_{i=1}^{s-1} \prod_{j=i+1}^{s} (q_j - q_i)^{n_i n_j}.$$

Hence it follows that $\Delta_0(v_1, \ldots, v_m)$ is non-zero, since $q_j \neq q_i$ for $j \neq i$, and therefore the functions $v_1(j), v_2(j), \ldots, v_m(j)$ constructed above are linearly independent solutions to the homogeneous equation (1). Here the general solution of (1) is written in the form

$$y(i) = \sum_{l=1}^{s} \sum_{n=0}^{n_l-1} c_n^{(l)} j^n q_l^j,$$

where $c_n^{(l)}$ are arbitrary constants.

1.3.3 Examples. We now consider the simplest examples of finding the general solution to homogeneous difference equations with constant coefficients.

[1] To find the general solution of the equation

$$y(i+2) - y(i+1) - 2y(i) = 0. \tag{10}$$

We form the characteristic equation $q^2 - q - 2 = 0$ and find its roots $q_1 = 2, q_2 = -1$. Since the roots are simple, the general solution of (10) has the form

$$y(i) = c_1 2^i + c_2(-1)^i.$$

[2] To find the general solution of the fourth-order equation

$$y(j+4) - 2y(j+3) + 3y(j+2) + 2y(j+1) - 4y(j) = 0. \tag{11}$$

The characteristic equation $q^4 - 2q^3 + 3q^2 + 2q - 4 = 0$ has two real roots $q_1 = 1$, $q_2 = -1$ and two complex-conjugate roots

$$q_3 = 2\left(\cos\frac{\pi}{3} + i\sin\frac{\pi}{3}\right) \text{ and } q_4 = 2\left(\cos\frac{\pi}{3} - i\sin\frac{\pi}{3}\right), \quad i = \sqrt{-1}.$$

Consequently, the general real-valued solution of equation (11) has the form

$$y(j) = c_1 + c_2(-1)^j + 2^j\left(c_3\cos\frac{\pi}{3}j + c_4\sin\frac{\pi}{3}j\right).$$

[3] To find the general solution of the fourth-order equation

$$y(j+4) - 7y(j+3) + 18y(j+2) - 20y(j+1) + 8y(j) = 0. \tag{12}$$

The characteristic equation

$$q^4 - 7q^3 + 18q^2 - 20q + 8 = (q-2)^3(q-1) = 0$$

has a root $q_1 = 2$ of multiplicity 3 and a root $q_2 = 1$ of multiplicity 1. Consequently, the general solution of (12) has the form

$$y(i) = c_1 + 2^j(c_2 + c_3 j + c_4 j^2),$$

and particular linearly independent solutions of (12) are the grid functions

$$v_1(j),\ v_2(j) = 2^j,\ v_3(j) = j2^j,\ v_4(j) = j^2 2^j.$$

[4] To find the general solution of the fourth-order equation

$$y(j+4)+8y(j+2)+16y(j)=0. \tag{13}$$

The characteristic equation $q^4+8q^2+16=(q^2+4)^2=0$ has the complex root

$$q_1=2\left(\cos\frac{\pi}{2}+i\sin\frac{\pi}{2}\right)$$

of multiplicity 2 and its conjugate root

$$q_2=2\left(\cos\frac{\pi}{2}-i\sin\frac{\pi}{2}\right)$$

also with multiplicity 2. Therefore the general real-valued solution of (13) has the form

$$y(j)=(c_1+c_2j)2^j\cos\frac{\pi}{2}j+(c_3+c_4j)2^j\sin\frac{\pi}{2}j.$$

We now consider two more examples. In one example we will find the solution of a Cauchy problem for a non-homogeneous equation; in the other, of a boundary-value problem for a fourth-order homogeneous equation.

[5] To find the solution of the following problem

$$y(i+1)-ay(i)=f(i),\quad i\geq 0,\quad y(0)=y_0, \tag{14}$$

where $a=$ constant. The characteristic equation $q-a=0$ has the single root $q_1=a$. Therefore the general solution of the homogeneous equations has the form $\overline{\overline{y}}(i)=ca^i$, $c=$ constant. A particular solution of the non-homogeneous equation (14) is found using the method of variation of parameters. Formula (20) of Section 2 gives the following particular solution to (14):

$$\bar{y}(i)=\sum_{k=0}^{i-1}a^{i-k-1}f(k)=\sum_{k=0}^{i-1}a^kf(i-k-1).$$

By theorem 3, the general solution of the non-homogeneous equation (14) has the form

$$y(i)=ca^i+\sum_{k=0}^{i-1}a^kf(i-k-1).$$

Setting $i=0$, we obtain $y_0=y(0)=c$ (the sum here vanishes). Thus, the solution of (14) is given by the formula

$$y(i)=y_0a^i+\sum_{k=0}^{i-1}a^kf(i-k-1),\quad i\geq 0.$$

[6] We now find the solution of the fourth-order equation

$$y(j+2)-y(j+1)+2y(j)-y(j-1)+y(j+2)=0, \quad 2 \le j \le N-2, \tag{15}$$

satisfying the following boundary conditions:

$$\begin{aligned} 2y(2)-y(1)+y(0)&=2,\\ y(3)-y(2)+y(1)-y(0)&=0,\\ y(N-3)-y(N-2)+y(N-1)-y(N)&=0,\\ 2y(N-2)-y(N-1)+y(N)&=0. \end{aligned} \tag{16}$$

The characteristic equation

$$q^4-q^3+2q^2-q+1=(q^2-q+1)(q^2+1)=0,$$

corresponding to (15) has the simple complex roots

$$\begin{aligned} q_1&=\cos\frac{\pi}{3}+i\sin\frac{\pi}{3},\\ q_2&=\cos\frac{\pi}{3}-i\sin\frac{\pi}{3},\\ q_3&=\cos\frac{\pi}{2}+i\sin\frac{\pi}{2},\\ q_4&=\cos\frac{\pi}{2}-i\sin\frac{\pi}{2}, \end{aligned} \qquad i=\sqrt{-1}.$$

consequently, the general real-valued solution of the homogeneous equation (15) has the form

$$y(j)=c_1\cos\frac{1}{3}\pi j+c_2\sin\frac{1}{3}\pi j+c_3\cos\frac{1}{2}\pi j+c_4\sin\frac{1}{2}\pi j. \tag{17}$$

We now isolate from the general solution of (17) the solution which satisfies the boundary conditions (16). For this we substitute (17) in (16) and obtain the following system for the constants c_1, c_2, c_3, and c_4:

$$\begin{aligned} \cos\frac{2\pi}{3}c_1+\sin\frac{2\pi}{3}c_2-c_3-c_4&=2,\\ c_1+0\cdot c_2+0\cdot c_3+0\cdot c_4&=0,\\ \cos\frac{N\pi}{3}c_1+\sin\frac{N\pi}{3}c_2+0\cdot c_3+0\cdot c_4&=0,\\ \cos\frac{(N-2)\pi}{3}c_1+\sin\frac{(N-2)\pi}{3}c_2&\\ -\left(\cos\frac{\pi N}{2}+\sin\frac{\pi N}{2}\right)c_3+\left(\cos\frac{\pi N}{2}-\sin\frac{\pi N}{2}\right)c_4&=0. \end{aligned}$$

The determinant of this system is equal to $-2\sin\frac{N\pi}{3}\cos\frac{N\pi}{2}$ and is non-zero if N is even and not divisible by 3.

In this case, using the fact that N is even, we obtain $c_1 = c_2 = 0$, $c_3 = c_4 = -1$. Thus, if N is not a multiple of 3, then the solution of the boundary-value problem (15), (16) exists and is given by the formula

$$y(i) = -\cos\frac{\pi j}{2} - \sin\frac{\pi j}{2}, \quad 0 \le j \le N.$$

If N is odd or a multiple of 3, then the solution of the boundary-value problem (15), (16) either does not exist or is not unique. This example illustrates the difference between boundary-value problems whose solutions do not always exist, and Cauchy problems possessing unique solutions.

1.4 Second-order equations with constant coefficients

1.4.1 The general solution of a homogeneous equation. The current section deals with various second-order equations with constant coefficients

$$a_2 y(j+2) + a_1 y(j+1) + a_0 y(j) = f(j), \quad a_0, a_2 \neq 0. \tag{1}$$

First of all, we will find the general solution of the corresponding homogeneous equation

$$a_2 y(j+2) + a_1 y(j+1) + a_0 y(j) = 0. \tag{2}$$

The characteristic equation $a_2 q^2 + a_1 q + a_0 = 0$ has the roots

$$q_1 = \frac{-a_1 + \sqrt{a_1^2 - 4a_0 a_2}}{2a_2}, \quad q_2 = \frac{-a_1 - \sqrt{a_1^2 - 4a_0 a_2}}{2a_2}.$$

According to the general theory of difference equations with constant coefficients found in Section 3, the functions $v_1(j) = q_1^j$, $v_2(j) = q_2^j$ are linearly independent solutions of equation (2) if $a_1^2 \neq 4a_0 a_2$, and $v_1(j) = q_1^j$, $v_2(j) = j q_1^j$ if $a_1^2 = 4a_0 a_2$. In the latter case, it will be convenient for us to use another set of linearly independent solutions

$$v_1(j) = \frac{q_2 q_1^j - q_1 q_2^j}{q_2 - q_1}, \quad v_2(j) = \frac{q_2^j - q_1^j}{q_2 - q_1}, \tag{3}$$

which take the following values for $j = 0$ and $j = 1$:

$$v_1(0) = 1, \quad v_1(1) = 0, \quad v_2(0) = 0, \quad v_2(1) = 1. \tag{4}$$

Clearly, it is only necessary to show that the functions (3) are solutions of the homogeneous equations if $a_1^2 = 4a_0a_2$. The linear independence of the functions (3) constructed above follows from the condition $\Delta_0(v_1, v_2) \neq 0$, where

$$\Delta_0(v_1, v_2) = \begin{vmatrix} v_1(0) & v_1(1) \\ v_2(0) & v_2(1) \end{vmatrix}.$$

If in (3) we take the limit as q_2 tends to q_1, then we obtain the functions $v_1(j) = -(j-1)q_1^j$, $v_2(j) = jq_1^{j-1}$, which in fact are solutions to the homogeneous equation (2). Notice that the functions $v_1(j)$ and $v_2(j)$ from (3) take real values even in the case when the roots q_1 and q_2 are complex. This allows us to avoid considering the complex-root case separately. Thus, the general solutions of the homogeneous equations (2) can be written in the form

$$\overline{\overline{y}}(j) = c_1v_1(j) + c_2v_2(j) = c_1\frac{q_2q_1^j - q_1q_2^j}{q_2 - q_1} + c_2\frac{q_2^j - q_1^j}{q_2 - q_1}, \tag{5}$$

where c_1 and c_2 are arbitrary constants. Notice that, by (4), we have $\overline{\overline{y}}(0) = c_1$, $\overline{\overline{y}}(1) = c_2$.

We now consider an example. It is necessary to find the general solutions of the homogeneous equation

$$y(j+2) - 2xy(j+1) + y(j) = 0, \tag{6}$$

where x is a real-valued parameter. In this case we have

$$q_1 = x + \sqrt{x^2 - 1}, \quad q_2 = \frac{1}{q_1}, \quad q_2 - q_1 = -2\sqrt{x^2 - 1}. \tag{7}$$

Substituting (7) in (5), we obtain the general solution to (6) for any x in the form

$$y(j) = -\frac{\left(x + \sqrt{x^2-1}\right)^{j-1} - \left(x + \sqrt{x^2-1}\right)^{-(j-1)}}{2\sqrt{x^2-1}}y(0) + \frac{\left(x + \sqrt{x^2-1}\right)^{j} - \left(x + \sqrt{x^2-1}\right)^{-j}}{2\sqrt{x^2-1}}y(1). \tag{8}$$

In particular, if $|x| \leq 1$, (8) can be written in the form

$$y(j) = -\frac{\sin(j-1)\arccos x}{\sin\arccos x}y(0) + \frac{\sin j\arccos x}{\sin\arccos x}y(1). \tag{9}$$

(In order to obtain (9), the identity $x = \cos(\arccos x)$ was used.)

This final result can be used to compute the integral in the problem posed in Section 1.1.4

$$I_k(\varphi) = \int_0^\pi \frac{\cos k\Psi - \cos k\varphi}{\cos \Psi - \cos \varphi} d\Psi, \quad k = 0, 1, \ldots .$$

It was shown that this problem leads to the solution of a Cauchy problem for the equation

$$I_{k+1} - 2\cos\varphi I_k + I_{k-1} = 0, \quad I_0 = 0, \; I_1 = \pi. \tag{10}$$

This equation is a particular case of (6) with $x = \cos\varphi$. Since $|x| \leq 1$, the general solution of (10) is given by (9), i.e.,

$$I_k = -\frac{\sin(k-1)\varphi}{\sin\varphi} I_0 + \frac{\sin k\varphi}{\sin\varphi} I_1.$$

Substituting the initial values for I_k, we obtain the solution of the problem

$$I_k(\varphi) = \pi \frac{\sin k\varphi}{\sin\varphi}.$$

As a second example, we consider the solution of the boundary-value problem

$$\begin{aligned} &y(j+1) - y(j) + y(j-1) = 0, \quad 1 \leq j \leq N-1, \\ &y(0) = 1, \quad y(N) = 0. \end{aligned} \tag{11}$$

The equation in problem (11) is also a particular case of (6), corresponding to $x = 1/2$. The formula (9) gives the following general solution to equation (11):

$$y(j) = \left(c_1 \sin \frac{(j-1)\pi}{3} + c_2 \sin \frac{j\pi}{3} \right) \Big/ \sin \frac{\pi}{3}.$$

The constants c_1 and c_2 are found from the boundary conditions for $y(j)$. If N is not a multiple of 3, then $c_1 = -1$, $c_2 = \sin \frac{1}{3}\pi(N-1) / \sin \frac{1}{3}\pi N$ and the solution of (11) has the form

$$y(j) = \sin \frac{1}{3}(N-j)\pi \Big/ \sin \frac{1}{3} N\pi, \quad 0 \leq j \leq N.$$

If N is a multiple of 3, then the solution of the boundary-value problem (11) does not exist.

1.4.2 The Chebyshev polynomials. We now return to equation (6). First of all we consider the following Cauchy problem:

$$\begin{aligned} &y(n+2) - 2xy(n+1) + y(n) = 0, \quad n \geq 0, \\ &y(0) = 1, \quad y(1) = x. \end{aligned} \tag{12}$$

Notice that from (12) it follows that

$$\begin{aligned} y(2) &= 2xy(1) - y(0) = 2x^2 - 1, \\ y(3) &= 2xy(2) - y(0) = 4x^3 - 3x, \end{aligned}$$

and in general $y(n)$ is polynomial of degree n in x. We denote this polynomial by $T_n(x)$. Substituting $T_n(x)$ for $y(n)$ in (12), we obtain a recurrence relation satisfied by these polynomials

$$\begin{aligned} &T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x), \quad n \geq 0 \\ &T_0(x) = 1, \quad T_1(x) = x, \quad -\infty < x < \infty. \end{aligned} \tag{13}$$

On the other hand, the general solution to (12) is given by (8) for any x. Substituting in (8) the initial values for $y(n)$, we have

$$T_n(x) = \frac{\left(x + \sqrt{x^2-1}\right)^n + \left(x + \sqrt{x^2-1}\right)^{-n}}{2}. \tag{14}$$

In particular, if $|x| \leq 1$, setting $x = \cos(\arccos x)$ gives us

$$T_n(x) = \cos(n \arccos x), \quad |x| \leq 1.$$

Thus, the solution of (12) has been found. The solution is the polynomial $T_n(x)$, which is defined for any x by (14) or by any formula

$$T_n(x) = \begin{cases} \cos(n \arccos x), & |x| \leq 1, \\ \frac{1}{2}\left[\left(x + \sqrt{x^2-1}\right)^n + \left(x + \sqrt{x^2-1}\right)^{-n}\right], & |x| \geq 1. \end{cases} \tag{15}$$

The polynomial $T_n(x)$ is called the *Chebyshev polynomial of the first kind of degree n.*

We now consider another Cauchy problem for equation (6)

$$\begin{aligned} &y(n+2) - 2xy(n+1) + y(n) = 0, \quad n \geq 0, \\ &y(0) = 1, \quad y(1) = 2x. \end{aligned} \tag{16}$$

It is obvious that here also $y(n)$ is a polynomial of degree n in x. Here we denote it by $U_n(x)$. The polynomial is an explicit form for $U_n(x)$. Substituting the initial values for $y(n)$ in (8), we have for any x:

$$\begin{aligned} U_n(x) &= \frac{2x\left(x+\sqrt{x^2-1}\right)^n - \left(x+\sqrt{x^2-1}\right)^{n-1}}{2\sqrt{x^2-1}} \\ &\quad + \frac{\left(x+\sqrt{x^2-1}\right)^{-(n-1)} - 2x\left(x+\sqrt{x^2-1}\right)^{-n}}{2\sqrt{x^2-1}} \\ &= \frac{\left(x+\sqrt{x^2-1}\right)^{n+1} - \left(x+\sqrt{x^2-1}\right)^{-(n+1)}}{2\sqrt{x^2-1}} \end{aligned} \tag{17}$$

In particular, if $|x| \leq 1$, then

$$U_n(x) = \frac{\sin(n+1)\arccos x}{\sin\arccos x}.$$

The polynomial $U_n(x)$ is called the *Chebyshev polynomial of the second kind of degree* n and is defined by the formulas

$$U_n(x) = \begin{cases} \dfrac{\sin(n+1)\arccos x}{\sin\arccos x}, \\ \qquad |x| \leq 1, \\ \dfrac{1}{2\sqrt{x^2-1}}\left[\left(x+\sqrt{x^2-1}\right)^{n+1} - \left(x+\sqrt{x^2-1}\right)^{-(n+1)}\right], \\ \qquad |x| \geq 1. \end{cases} \tag{18}$$

From (16) we obtain the following recurrence relation for the polynomials $U_n(x)$:

$$\begin{aligned} U_{n+2}(x) &= 2xU_{n+1}(x) - U_n(x), \quad n \geq 0, \\ U_0(x) &= 1, \quad U_1(x) = 2x. \end{aligned} \tag{19}$$

The formula (17) allows us to replace (8) by the following representation for the general solution of (6):

$$y(n) = -c_1U_{n-2}(x) + c_2U_{n-1}(x).$$

We now obtain still another representation for the general solution of (6). We shall show that the functions $v_1(n) = T_n(x)$ and $v_2(n) = U_{n-1}(x)$ are linearly independent solutions of the homogeneous equation (6). In fact,

it is only necessary to show their linear independence. Since the determinant

$$\Delta_0(v_1, v_2) = \begin{vmatrix} T_0(x) & T_1(x) \\ U_{-1}(x) & U_0(x) \end{vmatrix} = \begin{vmatrix} 1 & x \\ 0 & 1 \end{vmatrix} = 1$$

is non-zero, the assertion is true. Consequently, the general solution of (6) can be represented in the form

$$y(n) = c_1 T_n(x) + c_2 U_{n-1}(x), \tag{20}$$

where c_1 and c_2 are arbitrary constants, and the functions $T_n(x)$ and $U_n(x)$ are defined by (14) and (17) for any x.

In conclusion, we introduce several easily verified relations which show the connection between the Chebyshev polynomials $T_n(x)$ and $U_n(x)$ and also some properties of these polynomials. The formulas are as follows:

$$T_n(x) = T_{-n}(x), \quad U_{-n}(x) = -U_{n-2}(x), \quad n \geq 0, \tag{21}$$

$$T_{in}(x) = T_i(T_n(x)), \quad U_{in-1}(x) = U_{i-1}(T_n(x))U_{n-1}(x), \tag{22}$$

$$T_{2n}(x) = 2(T_n(x))^2 - 1, \tag{23}$$

$$T_{n-1}(x) - xT_n(x) = (1 - x^2)U_{n-1}(x), \tag{24}$$

$$U_{n-1}(x) - xU_n(x) = -T_{n+1}(x), \tag{25}$$

$$U_{n+i}(x) + U_{n-i}(x) = 2T_i(x)U_n(x). \tag{26}$$

By changing correspondingly the indices i and n, we obtain from (26)

$$U_{n+i-1}(x) + U_{n-i-1}(x) = 2T_i(x)U_{n-1}(x), \tag{27}$$

$$U_{n+i}(x) + U_{n-i-2}(x) = 2T_{i+1}(x)U_{n-1}(x). \tag{28}$$

Setting $i = n$ in (26)–(28), we have

$$2T_n(x)U_n(x) = U_{2n}(x) + 1, \tag{29}$$

$$2T_n(x)U_{n-1}(x) = U_{2n-1}(x), \tag{30}$$

$$2T_{n+1}(x)U_{n-1}(x) = U_{2n}(x) - 1. \tag{31}$$

Here we used equations (21) and $U_0(x) = 1$, $U_{-1}(x) = 0$. If we set $n = 0$ in (26), we obtain

$$2T_n(x) = U_n(x) - U_{n-2}(x). \tag{32}$$

1.4.3 The general solution of a non-homogeneous equation. We will now construct the general solution of the non-homogeneous equation (1).

$$a_2y(n+2)+a_1y(n+1)+a_0y(n)=f(n). \tag{33}$$

By theorem 3, the general solution to equation (33) is the sum $y(n)=\bar{y}(n)+\overline{\bar{y}}(n)$, where $\overline{\bar{y}}(n)$ is the general solution of the homogeneous equation (2), and $\bar{y}(n)$ is a particular solution of the non-homogeneous equation (33).

It was shown above that the functions

$$v_1(n)=\frac{q_2q_1^n-q_1q_2^n}{q_2-q_1},\quad v_2(n)=\frac{q_2^n-q_1^n}{q_2-q_1}, \tag{34}$$

are linearly independent solutions of equation (2), and that the solution $\overline{\bar{y}}(n)$ is defined by the formula (5):

$$\overline{\bar{y}}(n)=c_1v_1(n)+c_2v_2(n).$$

In order to find a particular solution $\bar{y}(n)$ to equation (33), we will use the method of variation of parameters, outlined in Section 1.2.3. Formula (1.2.19) gives the solution $\bar{y}(n)$ in the following form:

$$\bar{y}(n)=\sum_{k=n_0}^{n-2}\frac{\begin{vmatrix} v_1(k+1) & v_2(k+1)\\ v_1(n) & v_2(n)\end{vmatrix}}{\begin{vmatrix} v_1(k+1) & v_1(k+2)\\ v_2(k+1) & v_2(k+2)\end{vmatrix}}\cdot\frac{f(k)}{a_2}.$$

After some simple calculations we get

$$\bar{y}(n)=\sum_{k=n_0}^{n-2}\frac{q_2^{n-k-1}-q_1^{n-k-1}}{q_2-q_1}\cdot\frac{f(k)}{a_2},\quad n\neq n_0,n_0+1$$

and

$$\bar{y}(n_0)=\bar{y}(n_0+1)=0.$$

Consequently, the general solution of the non-homogeneous equation (33) has the form

$$y(n)=c_1\frac{q_2q_1^n-q_1q_2^n}{q_2-q_1}+c_2\frac{q_2^n-q_1^n}{q_2-q_1}+\sum_{k=n_0}^{n-2}\frac{q_2^{n-k-1}-q_1^{n-k-1}}{q_2-q_1}\cdot\frac{f(k)}{a_2}, \tag{35}$$

where c_1 and c_2 are arbitrary constants.

If we are solving a Cauchy problem, i.e., seeking the solution of (33) satisfying the conditions

$$y(n_0) = y_0, \quad y(n_0 + 1) = y_1, \tag{36}$$

then from (35) and (36) we obtain the following representation for the solution to this problem

$$\begin{aligned} y(n) = y_0 \frac{q_2 q_1^{n-n_0} - q_1 q_2^{n-n_0}}{q_2 - q_1} + y_1 \frac{q_2^{n-n_0} - q_1^{n-n_0}}{q_2 - q_1} \\ + \sum_{k=n_0}^{n-2} \frac{q_2^{n-k-1} - q_1^{n-k-1}}{q_2 - q_1} \cdot \frac{f(k)}{a_2}. \end{aligned} \tag{37}$$

We will now find the solution to the first boundary-value problem for a second-order difference equation with constant coefficients. It will be convenient to write the problem in the following form:

$$\begin{aligned} &a_2 y(n+1) + a_1 y(n) + a_0 y(n-1) = -f(n), \quad 1 \le n \le N-1, \\ &y(0) = \mu_1, \quad y(N) = \mu_2. \end{aligned} \tag{38}$$

This formulation is obtained from (33) by shifting the index n; therefore, using (35), we obtain the following formula for the general solution of (38):

$$y(n) = c_1 \frac{q_2 q_1^n - q_1 q_2^n}{q_2 - q_1} + c_2 \frac{q_2^n - q_1^n}{q_2 - q_1} - \sum_{k=1}^{n-1} \frac{q_2^{n-k} - q_1^{n-k}}{q_2 - q_1} \cdot \frac{f(k)}{a_2}. \tag{39}$$

We will determine the constants c_1 and c_2 from the condition that the solution (39) take on the values $y(0) = \mu_1$ and $y(N) = \mu_2$. Omitting the simple computations, we obtain the following formula for the solution of the boundary-value problem (38).

$$\begin{aligned} y(n) = &\frac{(q_1 q_2)^n (q_2^{N-n} - q_1^{N-n})}{q_2^N - q_1^N} \mu_1 + \frac{q_2^n - q_1^n}{q_2^N - q_1^N} \mu_2 \\ &+ \sum_{k=1}^{n-1} \frac{(q_1 q_2)^{n-k} (q_2^{N-n} - q_1^{N-n})(q_2^k - q_1^k)}{(q_2 - q_1)(q_2^N - q_1^N)} \cdot \frac{f(k)}{a^2} \\ &+ \sum_{k=n}^{N-1} \frac{(q_2^{N-k} - q_1^{N-k})(q_2^n - q_1^n)}{(q_2 - q_1)(q_2^N - q_1^N)} \cdot \frac{f(k)}{a_2}. \end{aligned} \tag{40}$$

Notice that the solution of the boundary-value problem (38) fails to exist only in the case when $q_1^N = q_2^N$ and $q_1 \neq q_2$.

We will now consider a particular case which uses the formula (40). Suppose we are required to solve the first boundary-value problem for the equation

$$\begin{aligned} &y(n+1) - 2xy(n) + y(n-1) = -f(n), \\ &1 \le n \le N-1, \quad y(0) = \mu_1, \quad y(N) = \mu_2. \end{aligned} \tag{41}$$

Earlier we found the roots q_1 and q_2 of the characteristic equation corresponding to (41)

$$q_1 = x + \sqrt{x^2 - 1}, \quad q_2 = x - \sqrt{x^2 - 1} = 1/q_1.$$

Substituting these values in (40) and taking into account formula (17) for the polynomial $U_n(x)$, we obtain the solution of problem (41) in the following form

$$\begin{aligned} y(n) = &\frac{U_{N-n-1}(x)}{U_{N-1}(x)} \left[\mu_1 + \sum_{k=1}^{n-1} U_{k-1}(x) f(k) \right] \\ &+ \frac{U_{n-1}(x)}{U_{N-1}(x)} \left[\mu_2 + \sum_{k=n}^{N-1} U_{N-k-1}(x) f(k) \right]. \end{aligned} \tag{42}$$

The solution exists and is given by the formula (42) if the following condition is satisfied:

$$x \neq \cos \frac{k\pi}{N}, \quad k = 1, 2, \ldots, N-1.$$

We now return to equation (38). If $a_0 a_2 > 0$, then the solution (40) of this problem can be written in a more compact form. To this end, we write the roots

$$q_1 = \frac{1}{2a_2} \left[-a_1 + \sqrt{a_1^2 - 4a_0 a_2} \right], \quad q_2 = \frac{1}{2a_2} \left[-a_1 - \sqrt{a_1^2 - 4a_0 a_2} \right]$$

of the characteristic equation corresponding to (38) in the following form:

$$q_1 = \rho \left(x + \sqrt{x^2 - 1} \right), \quad q_2 = \rho \left(x - \sqrt{x^2 - 1} \right), \tag{43}$$

where

$$\rho = \sqrt{\frac{a_0}{a_2}}, \quad x = -\frac{a_1}{2\sqrt{a_0 a_2}}. \tag{44}$$

We then substitute (43) into (40) and use (17). We obtain the solution of (38) for the case $a_0 a_2 > 0$ in the form

$$y(n) = \frac{U_{N-n-1}(x)}{U_{N-1}(x)}\rho^n \left[\mu_1 + \sum_{k=1}^{n-1} \frac{U_{k-1}(x)}{\rho^{k-1}} \cdot \frac{f(k)}{a_0}\right] + \frac{U_{n-1}(x)}{U_{N-1}(x)} \cdot \frac{1}{\rho^{N-n}} \left[\mu_2 + \sum_{k=n}^{N-1} \rho^{N-k-1} U_{n-k-1}(x) \frac{f(k)}{a_0}\right],$$

where ρ and x are defined in (44). The solution of (38) for the case $a_0 a_2 > 0$ exists if

$$a_1 + 2\sqrt{a_0 a_2} \cos \frac{k\pi}{N} \neq 0, \quad k = 1, 2, \ldots, N-1.$$

We now look at a first-order boundary-value problem for the three-point vector equation with constant coefficients

$$\begin{aligned} &Y_{n-1} - CY_n + Y_{n+1} = -F_n, \quad 1 \leq n \leq N-1 \\ &Y_0 = F_0, \quad Y_N = F_N, \end{aligned} \tag{45}$$

where Y_n and F_n are vectors, and C is a square matrix. It is easy to verify that the general solution of the non-homogeneous equation (45) has the form

$$Y_n = U_{n-2}\left(\frac{1}{2}C\right) C_1 + U_{n-1}\left(\frac{1}{2}C\right) C_2 - \sum_{k=1}^{n-1} U_{n-k-1}\left(\frac{1}{2}C\right) F_k,$$

where C_1 and C_2 are arbitrary vectors, and $U_n(X)$ is a matrix polynomial in the matrix X, defined by the recurrence relation (19).

If the matrix C is such that $U_{N-1}(\frac{1}{2}C)$ is nonsingular, then the solution of the boundary-value problem (45) is defined by a formula analogous to (42)

$$\begin{aligned} Y_n = {} & U_{N-1}^{-1}\left(\frac{1}{2}C\right) U_{N-n-1}\left(\frac{1}{2}C\right) \left[F_0 + \sum_{k=1}^{n-1} U_{k-1}\left(\frac{1}{2}C\right) F_k\right] \\ & + U_{N-1}^{-1}\left(\frac{1}{2}C\right) U_{n-1}\left(\frac{1}{2}C\right) \left[F_N + \sum_{k=n}^{N-1} U_{N-k-1}\left(\frac{1}{2}C\right) F_k\right]. \end{aligned} \tag{46}$$

Below it will be shown that a Dirichlet difference problem for Poisson's equation in a rectangle leads to problem (45).

In conclusion we remark that the existence condition for the solution of (45) can be formulated as follow: the solution exists and is defined by (46) if the numbers $\cos(k\pi/N)$, $k = 1, 2, \ldots, N-1$ are not eigenvalues of the matrix C.

1.5 Eigenvalue difference problems

1.5.1 A boundary-value problem of the first kind. In Chapter 4 we will look at the method of separation of variables, which is used to find the solutions of boundary-value grid problems for elliptic equations in a rectangle. In connection with this, it becomes necessary to represent the desired grid functions as an expansion in the eigenfunctions of the corresponding difference problem. In this section we will consider eigenvalue difference problems for the simplest second-order difference operator, defined on a uniform grid.

We now formulate the boundary-value problem of the first kind. Suppose that the uniform grid $\bar{\omega} = \{x_i = ih,\ i = 0, 1, \ldots, N, hN = l\}$ with step h has been introduced onto the interval $[0, l]$. We must find those values of the parameter λ (the eigenvalues) for which there exist non-trivial solutions $\mu(x_i)$ (eigenfunctions) to the following difference problem:

$$y_{\bar{x}x} + \lambda y = 0, \quad x \in \omega, \quad y(0) = y(l) = 0, \tag{1}$$

where

$$y_{\bar{x}x,i} = \frac{y(i+1) - 2y(i) + y(i-1)}{h^2}, \quad y(i) = y(x_i).$$

We now find the solution to (1). For this we write (1) in the form of a boundary-value problem for a second-order difference equation

$$\begin{aligned} &y(i+1) - 2\left(1 - \frac{h^2\lambda}{2}\right) y(i) + y(i-1) = 0, \quad 1 \le i \le N-1, \\ &y(0) = y(N) = 0. \end{aligned} \tag{2}$$

In Section 1.4.1 it was shown that the general solution of (2) has the form (see (1.4.20)) $y(i) = c_1 T_i(z) + c_2 U_{i-1}(z)$, where c_1 and c_2 are arbitrary constants, and where z here denotes

$$z = 1 - h^2\lambda/2. \tag{3}$$

The constants c_1 and c_2 are determined from the boundary conditions

$$y(0) = c_1 = 0, \quad y(N) = c_2 U_{N-1}(z) = 0. \tag{4}$$

Both here and later we use the formulas (1.4.15) and (1.4.18), which define the Chebyshev polynomials of the first and second kinds, and also the formulas (1.4.21)–(1.4.32).

Since we are seeking a non-trivial solution to (1), $c_2 \neq 0$, and from (4) we have the condition $U_{N-1}(z) = 0$, which determines the solution in the form $y_i = c_2 U_{i-1}(z)$.

Since the numbers $z_k = \cos\frac{k\pi}{N}$, $k = 1, 2, \ldots, N-1$ are the roots of the polynomial $U_{N-1}(z)$, from (3) we find the eigenvalues of (1) as

$$\lambda_k = \frac{4}{h^2}\sin^2\frac{k\pi}{2N} = \frac{4}{h^2}\sin^2\frac{k\pi h}{2l}, \quad k = 1, 2, \ldots, N-1. \tag{5}$$

Each eigenvalue λ_k corresponds to a non-null solution of (1)

$$y_k(i) = c_2 U_{i-1}(z_k) = \bar{c}_k \sin\frac{k\pi i}{N} = \bar{c}_k \sin\frac{k\pi x_i}{l}, \tag{6}$$

$$0 \le i \le N \quad \left(c_2 = \bar{c}_k \sin\frac{k\pi}{N}\right).$$

We define the scalar product of grid functions defined on the grid $\bar{\omega}$ in the following form

$$(u, v) = \sum_{i=1}^{N-1} u(i)v(i)h + 0.5h[u(0)v(0) + u(N)v(N)].$$

We now choose the constants c_k in (6) so that the functions $y_k(i)$ will have norm one, i.e., $(y_k, y_k) = 1$.

A simple computation gives $\bar{c}_k = \sqrt{2/l}$. Substituting this value for $\bar{c}_k$ in (6), we obtain the eigenfunctions $\mu_k(i)$ for (1)

$$\mu_k(i) = \sqrt{\frac{2}{l}}\sin\frac{k\pi i}{N} = \sqrt{\frac{2}{l}}\sin\frac{k\pi x_i}{l}, \tag{7}$$

$$i = 0, 1, \ldots, N, \quad k = 1, 2, \ldots, N-1.$$

Thus, the problem (1) is solved and the solution is given by (5) and (7).

We now enumerate the basic properties of the eigenfunctions and eigenvalues of the boundary-value problem of the first kind (1).

[1] The eigenfunctions are orthonormal:

$$(\mu_k, \mu_m) = \delta_{km}, \quad \delta_{km} = \begin{cases} 1, & k = m, \\ 0, & k \ne m. \end{cases}$$

[2] For any grid function $f(i)$ defined at the interior points of the grid $\bar{\omega}$, i.e., for $1 \le i \le N-1$, we have the expansion

$$f(i) = \frac{2}{N}\sum_{k=1}^{N-1} \varphi_k \sin\frac{k\pi i}{N}, \quad i = 1, 2, \ldots, N-1, \tag{8}$$

where

$$\varphi_k = \sum_{i=1}^{N-1} f(i) \sin \frac{k\pi i}{N}, \quad k = 1, 2, \ldots, N-1. \tag{9}$$

Let us clarify this assertion. Let $f(i)$ be an arbitrary grid function defined on ω (or defined on $\bar{\omega}$ and reducing to zero for $i = 0$ and $i = N$). Find its eigenfunction expansion

$$f(i) = \sum_{k=1}^{N-1} f_k \mu_k(i) = \sum_{k=1}^{N-1} \sqrt{\frac{2}{l}} f_k \sin \frac{k\pi i}{N}, \tag{10}$$

where f_k is the Fourier coefficient of the function $f(i)$. Computing the scalar product of (10) with $\mu_m(i)$ and using the orthonormality of the eigenfunctions, we obtain the Fourier coefficient

$$f_m = \sum_{k=1}^{N-1} f_k(\mu_k, \mu_m) = (f, \mu_m) = \sum_{i=1}^{N-1} \sqrt{\frac{2}{l}} f(i) \sin \frac{\pi m i}{N} h.$$

The connection of this formula with (8)–(9) is easy to establish if it is noted that $f_m = (\sqrt{2l}/N)\varphi_m$.

The expansion (8), (9) is convenient in that, for both the Fourier transform and the inverse Fourier transform of a function $f(i)$, it is only necessary to compute a single type of summation. An algorithm for efficiently computing such a summation will be considered in Chapter 4.

[3] The eigenvalues satisfy the inequalities

$$\frac{8}{l^2} \leq \frac{4}{h^2} \sin^2 \frac{\pi}{2N} = \lambda_1 \leq \lambda_k \leq \lambda_{N-1} = \frac{4}{h^2} \cos^2 \frac{\pi}{2N}, \; 1 \leq k \leq N-1.$$

1.5.2 A boundary-value problem of the second kind. We now consider a boundary-value problem

$$\begin{aligned} &y_{\bar{x}x} + \lambda y = 0, \quad x \in \omega \\ &\frac{2}{h} y_x + \lambda y = 0, \quad x = 0, \quad -\frac{2}{h} y_{\bar{x}} + \lambda y = 0, \; x = l. \end{aligned} \tag{11}$$

We now find the solution of (11). Writing out the difference derivatives at the points in (11), we obtain the problem

$$\begin{aligned} &y(i+1) - 2zy(i) + y(i-1) = 0, \quad 1 \leq i \leq N-1, \\ &y(1) - zy(0) = 0, \quad y(N-1) - zy(N) = 0, \end{aligned} \tag{12}$$

where $z = 1 - \lambda h^2/2$. From the general solution to equation (12) $y(i) = c_1 T_i(z) + c_2 U_{i-1}(z)$, we determine the solution satisfying the boundary conditions. Using (1.4.24), we get

$$y(1) - zy(0) = c_1 z + c - 2 - c_1 z = c_2 = 0, \quad c_2 = 0,$$

and also

$$y(N-1) - zy(N) = c_1(T_{N-1}(z) - zT_N(z)) = c_1(1 - z^2)U_{N-1}(z) = 0.$$

Since $c_1 \neq 0$, we obtain

$$z_k = \cos\frac{k\pi}{N}, \quad k = 0, 1, \ldots, N,$$

and consequently, the eigenvalues of (12) are

$$\lambda_k = \frac{4}{h^2}\sin^2\frac{k\pi}{2N} = \frac{4}{h^2}\sin^2\frac{k\pi h}{2l}, \quad k = 0, 1, \ldots, N. \tag{13}$$

Here, each λ_k corresponds to a non-null solution of (11)

$$y_k(i) = c_k T_i(z_k) = c_k \cos\frac{k\pi i}{N}, \quad 0 \leq i \leq N.$$

We choose the constants c_k from the condition $(y_k, y_k) = 1$, where the scalar product is defined above. Direct calculations show that

$$c_k = \sqrt{2/l}, \quad k = 1, 2, \ldots, N-1, \quad c_k = \sqrt{1/l}, \quad k = 0, N.$$

Thus, the normalized eigenfunctions for (11) are

$$\begin{aligned} \mu_k(i) &= \sqrt{\frac{2}{l}}\cos\frac{k\pi i}{N} = \sqrt{\frac{2}{l}}\cos\frac{k\pi x_i}{l}, \quad 1 \leq k \leq N-1, \\ \mu_k(i) &= \sqrt{\frac{1}{l}}\cos\frac{k\pi i}{N} = \sqrt{\frac{1}{l}}\cos\frac{k\pi x_i}{l}, \quad k = 0, N, \end{aligned} \tag{14}$$

defined on the grid $\bar{\omega}$. Notice that the eigenfunction corresponding to the zero eigenvalue $\lambda_0 = 0$ is the constant $\mu_0(i) = \sqrt{1/l}$.

We now formulate the properties of the eigenfunctions and eigenvalues of the boundary-value problem of the second kind (11).

[1] The eigenfunctions are orthonormal: $(\mu_k, \mu_m) = \delta_{km}$.

[2] For any grid function $f(i)$ defined on $\bar{\omega}$, we have the expansion

$$f(i) = \frac{2}{N} \sum_{k=0}^{N} \rho_k \varphi_k \cos \frac{k\pi i}{N}, \quad i = 0, 1, \ldots, N, \tag{15}$$

where

$$\varphi_k = \sum_{i=0}^{N} \rho_i f(i) \cos \frac{k\pi i}{N}, \quad k = 0, 1, \ldots, N, \tag{16}$$

$$\rho_i = \begin{cases} 1, & 1 \le i \le N-1, \\ 0.5, & i = 0, N. \end{cases} \tag{17}$$

The formulas (15) and (16) are modifications of the traditional expansion of $f(i)$ in the eigenfunctions $\mu_k(i)$

$$f(i) = \sum_{k=0}^{N} f_k \mu_k(i), \quad f_k = (f, \mu_k)$$

where the following substitutions have been made:

$$f_k = \begin{cases} \frac{\sqrt{2l}}{N} \varphi_k, & 1 \le k \le N-1, \\ \frac{1}{N} \sqrt{l} \varphi_k, & k = 0, N. \end{cases}$$

[3] The eigenvalues satisfy the inequalities

$$0 = \lambda_0 \le \lambda_k \le \lambda_N, \quad 0 \le k \le N.$$

1.5.3 A mixed boundary-value problem. We now consider an eigenvalue problem where on one side of the interval $[0, l]$ a first-kind boundary condition is given, and on the other — a second-kind condition; for example:

$$\begin{aligned} &y_{\bar{x}x} + \lambda y = 0, \quad x \in \omega, \\ &y(0) = 0, \quad -\frac{2}{h} y_{\bar{x}} + \lambda y = 0, \quad x = l. \end{aligned} \tag{18}$$

We will call such a problem a *mixed boundary-value problem.*

Let us find the solution of problem (18). The corresponding problem for a second-order difference equation has the form

$$\begin{gathered} y(i+1) - 2zy(i) + y(i-1) = 0, \quad 1 \le i \le N-1, \\ y(0) = 0, \quad y(N-1) - zy(N) = 0, \end{gathered}$$

where $z = 0.5\lambda h^2$. From the general solution to this equation

$$y(i) = c_1 T_i(z) + c_2 U_{i-1}(z)$$

we extract the solution satisfying the given boundary conditions. Using (1.4.25), we obtain

$$y(0) = c_1 = 0,$$
$$y(N-1) - zy(N) = c_2(U_{N-2}(z) - zU_{N-1}(z)) = -c_2 T_N(z) = 0.$$

Since $c_2 \neq 0$, we obtain $T_N(z_k) = 0$, where

$$z_k = \cos\frac{(2k-1)\pi}{2N}, \quad k = 1, 2, \ldots, N$$

and consequently, the eigenvalues for problem (18) are

$$\lambda_k = \frac{4}{h^2}\sin^2\frac{(2k-1)\pi}{4N} = \frac{4}{h^2}\sin^2\frac{(2k-1)\pi h}{4l}, \quad k = 1, 2, \ldots, N. \tag{19}$$

The normalized eigenfunctions for problem (18) corresponding to the eigenvalues λ_k are

$$\begin{aligned}\mu_k(i) &= \sqrt{\frac{2}{l}}\sin\frac{(2k-1)\pi i}{2N} \\ &= \sqrt{\frac{2}{l}}\sin\frac{(2k-1)\pi x_i}{2l}, \quad k = 1, 2, \ldots, N.\end{aligned} \tag{20}$$

We now formulate the properties of the eigenfunctions and eigenvalues of the mixed boundary-value problem (18).

[1] The eigenfunctions are orthonormal: $(\mu_k, \mu_m) = \delta_{km}$.

[2] For any grid function $f(i)$ defined on $\omega^+ = \{x_i = ih, 1 \leq i \leq N\}$ (or on $\bar{\omega}$, and reducing to zero for $i = 0$), we have the expansion

$$f(i) = \frac{2}{N}\sum_{k=1}^{N}\varphi_k \sin\frac{(2k-1)\pi i}{2N}, \quad i = 1, 2, \ldots, N, \tag{21}$$

where

$$\varphi_k = \sum_{i=1}^{N}\rho_i f(i)\sin\frac{(2k-1)\pi i}{2N}, \quad k = 1, 2, \ldots, N, \tag{22}$$

and where ρ_i is defined in (17).

[3] The eigenvalues satisfy the inequalities

$$\frac{8}{(2+\sqrt{2})l^2} \le \frac{4}{h^2}\sin^2\frac{\pi}{4N} = \lambda_1 \le \lambda_k \le \lambda_N = \frac{4}{h^2}\cos^2\frac{\pi}{4N}, \quad 1 \le k \le N.$$

If for equation (18) the first-kind boundary condition is given at the right end of the interval $[0, l]$, i.e.,

$$\begin{aligned} &y_{\bar{x}x} + \lambda y = 0, \quad x \in \omega \\ &\frac{2}{h}y_x + \lambda y = 0, \quad x = 0; \quad y(l) = 0, \end{aligned} \tag{23}$$

then the eigenvalues are defined by the formula (19), and the normalized eigenfunctions are

$$\mu_k(i) = \sqrt{\frac{2}{l}}\sin\frac{(2k-1)(N-i)\pi}{2N} = \sqrt{\frac{2}{l}}\sin\frac{(2k-1)\pi(l-x_i)}{2l},$$
$$k = 1, 2, \ldots, N.$$

We have the following proposition. *For any grid function $f(i)$ defined on $\omega^- = \{x_i = ih,\ i = 0, 1, \ldots, N-1, hN = l\}$ (or on $\bar{\omega}$ and reducing to zero for $i = N$), we have the expansion*

$$f(N-i) = \frac{2}{N}\sum_{k=1}^{N}\varphi_k \sin\frac{(2k-1)\pi i}{2N}, \quad i = 1, 2, \ldots, N, \tag{24}$$

where

$$\varphi_k = \sum_{i=1}^{N}\rho_{N-i} f(N-i)\sin\frac{(2k-1)\pi i}{2N}, \quad k = 1, 2, \ldots, N, \tag{25}$$

and where ρ_i is defined in (17).

Notice that the eigenfunctions constructed for (23) are also orthonormal:

$$(\mu_k, \mu_m) = \delta_{km}.$$

1.5.4 A periodic boundary-value problem. Suppose that, on the grid $\Omega = \{x_i = ih,\ i = 0, \pm 1, \pm 2, \ldots\}$ introduced on the line $-\infty < x < \infty$, we are seeking a non-trivial periodic solution with period N to the following eigenvalue problem:

$$\begin{aligned} &y_{\bar{x}x} + \lambda y = 0, \quad x \in \Omega, \\ &y(i+N) = y(i), \quad i = 0, \pm 1, \pm 2, \ldots, \quad h = l/N. \end{aligned} \tag{26}$$

Since the solution is periodic, it is sufficient to find it for $i = 0, 1, \ldots, N-1$. Writing out (26) at the points $i = 0, 1, \ldots, N-1$ and remembering that $y(-1) = y(N-1)$, $y(0) = y(N)$, we obtain the following problem:

$$\begin{aligned} &y(i+1) - 2zy(i) + y(i-1) = 0, \quad 0 \le i \le N-1, \\ &y(0) = y(N), \quad y(-1) = y(N-1), \end{aligned} \tag{27}$$

where $z = 0.5\lambda h^2$.

We shall now solve (27). Let us substitute the general solution

$$y(i) = c_1 T_i(z) + c_2 U_{i-1}(z)$$

into the boundary conditions. Taking into account the properties of the Chebyshev polynomials, we obtain the following system for determining the constants c_1 and c_2:

$$\begin{aligned} c_1(1 - T_N(z)) - c_2 U_{N-1}(z) &= 0, \\ c_1(T_{N-1}(z) - z) + c_2(1 + U_{N-2}(z)) &= 0. \end{aligned} \tag{28}$$

This system has a non-null solution if and only if its determinant is zero. We calculate it using (1.4.25), (1.4.29), and (1.4.31). We obtain

$$\begin{aligned} &(1 - T_N(z))(1 + U_{N-2}(z)) + (T_{N-1}(z) - z)U_{N-1}(z) \\ &\quad = 1 + U_{N-2}(z) - zU_{N-1}(z) - T_N(z) + T_{N-1}(z)U_{N-1}(z) - T_N(z)U_{N-2}(z) \\ &\quad = 2[1 - T_N(z)] = 0. \end{aligned}$$

From this it follows that, for $z = z_k$, where

$$z_k = \cos\frac{2k\pi}{N}, \quad k = 0, 1, \ldots, N-1, \tag{29}$$

the system (28) has a non-null solution. Thus, the eigenvalues of (26) are

$$\lambda_k = \frac{4}{h^2}\sin^2\frac{k\pi}{N} = \frac{4}{h^2}\sin^2\frac{k\pi h}{l}, \quad k = 0, 1, \ldots, N-1. \tag{30}$$

We now obtain the solution of (28). Since

$$\begin{aligned} &T_{N-1}(z_k) = z_k, \quad 0 \le k \le N-1, \\ &U_{N-2}(z_k) = \begin{cases} N-1, & k = 0, N/2, \\ -1, & k \ne 0, N/2, \end{cases} \\ &U_{N-1}(z_k) = \begin{cases} N, & k = 0, \\ -N, & k = N/2, \\ 0, & k \ne 0, N/2, \end{cases} \end{aligned}$$

we obtain, substituting (29) in (28), the following solution to (28):

[a] for $k = 0$ and $k = N/2$ we have $c_2 = 0$, $c_1 = c_1^{(k)} \neq 0$;

[b] for $k \neq 0$, $k \neq N/2$, $0 < k \leq N-1$, the constants $c_1 = c_1^{(k)}$, $c_2 = c_2^{(k)}$ are arbitrary, but not simultaneously zero. From this we obtain that the functions

$$\begin{aligned} y_k(i) &= c_1^{(k)} \cos \frac{2k\pi i}{N}, \quad k = 0, N/2, \\ y_k(i) &= c_1^{(k)} \cos \frac{2k\pi i}{N} + c_2^{(k)} \frac{2k\pi i}{N}, \quad 1 \leq k \leq N-1, \ k \neq 0, \frac{N}{2} \end{aligned} \tag{31}$$

are solutions of (27) corresponding to the eigenvalue λ_k. Notice that, in the case $k \neq 0, N/2$, the formulas (31) in fact determine two linearly independent functions

$$c_1^{(k)} \cos \frac{2k\pi i}{N} \quad \text{and} \quad c_2^{(k)} \sin \frac{2k\pi i}{N},$$

each of which is a solution to (27) corresponding to the eigenvalue λ_k.

We now construct the normalized eigenfunctions for (26). We remark that, for periodic grid functions, the scalar product introduced above can be written in the following form:

$$\begin{aligned} (u, v)_{\bar{\omega}} &= \sum_{i=1}^{N-1} u(i)v(i)h + 0.5h[u(0)v(0) + u(N)v(N)] \\ &= \sum_{i=0}^{N-1} u(i)v(i)h. \end{aligned}$$

We consider two cases. First suppose that N is even. From (31) we obtain that the eigenfunctions corresponding to λ_0 and $\lambda_{N/2}$ are

$$\mu_k(i) = \sqrt{\frac{1}{l}} \cos \frac{2k\pi i}{N}, \quad k = 0, \frac{N}{2}. \tag{32}$$

We further remark that it follows from (30) that

$$\lambda_{N-k} = \frac{4}{h^2} \sin^2 \frac{(N-k)\pi}{N} = \frac{4}{h^2} \sin^2 \frac{k\pi}{N} = \lambda_k,$$
$$k = 1, 2, \ldots, \frac{N}{2} - 1.$$

Choosing as eigenfunctions

$$\mu_k(i) = \sqrt{\frac{2}{l}} \cos \frac{2k\pi i}{N}, \quad 1 \le k \le \frac{N}{2} - 1$$

corresponding to the eigenvalue λ_k, and

$$\mu_{N-k}(i) = \sqrt{\frac{2}{l}} \sin \frac{2k\pi i}{N}, \quad 1 \le k \le \frac{N}{2} - 1,$$

corresponding to the eigenvalue $\lambda_{N-k} = \lambda_k$, in place of (32) we obtain a full system of eigenfunctions for the problem (26). Thus, the eigenvalues are λ_k, defined in (30), and the eigenfunctions of problem (26) are given by the formulas

$$\begin{aligned} \mu_k(i) &= \sqrt{\frac{1}{l}} \cos \frac{2k\pi i}{N}, \quad k = 0, \frac{N}{2}, \\ \mu_k(I) &= \sqrt{\frac{2}{l}} \cos \frac{2k\pi i}{N}, \quad 1 \le k \le \frac{N}{2} - 1, \\ \mu_k(i) &= \sqrt{\frac{2}{l}} \sin \frac{2(N-k)\pi i}{N}, \quad \frac{N}{2} + 1 \le k \le N - 1 \end{aligned} \tag{33}$$

for the case of N even.

We list here the basic properties of the eigenfunctions and eigenvalues of the periodic boundary-value problem (26).

[1] The eigenfunctions are orthonormal.

[2] Any periodic grid function $f(i)$ with period N, defined on the grid Ω, can be represented in the form

$$f(i) = \frac{2}{N} \sum_{k=0}^{N/2} \rho_k \varphi_k \cos \frac{2k\pi i}{N} + \frac{2}{N} \sum_{k=N/2+1}^{N-1} \varphi_k \sin \frac{2(N-k)\pi i}{N}, \tag{34}$$

where

$$\begin{aligned} \varphi_k &= \sum_{i=0}^{N-1} \rho_i f(i) \cos \frac{2k\pi i}{N}, \quad 0 \le k \le \frac{N}{2}, \\ \varphi_k &= \sum_{i=0}^{N-1} f(i) \sin \frac{2(N-k)\pi i}{N}, \quad \frac{N}{2} + 1 \le k \le N - 1, \end{aligned} \tag{35}$$

$$\rho_k = \begin{cases} 1, & k \ne 0, N/2, \\ 1/\sqrt{2}, & k = 0, N/2. \end{cases} \tag{36}$$

The formulas (34)–(36) follow from the expansion of the function $f(i)$ in the eigenfunctions $\mu_k(i)$:

$$f(i) = \sum_{k=0}^{N-1} f_k \mu_k(i), \quad f_k = (f, \mu_k)$$

with the substitution $f_k = \left(\sqrt{2l}/N\right) \varphi_k$.

[3] The eigenvalues satisfy the inequalities

$$0 = \lambda_0 \leq \lambda_k \leq \lambda_{N/2} = \frac{4}{h^2}, \quad 0 \leq k \leq N-1.$$

We now consider the case where N is odd. In this case, the eigenvalues of (26) are defined by the formulas (30), where $\lambda_0 = 0$, and $\lambda_{N-k} = \lambda_k$, $k = 1, 2, \ldots, (N-1)/2$.

The eigenfunctions corresponding to the eigenvalues λ_k are defined by the following formulas:

$$\begin{aligned} \mu_0(i) &= \sqrt{\frac{1}{l}}, && k = 0, \\ \mu_k(i) &= \sqrt{\frac{2}{l}} \cos \frac{2k\pi i}{N}, && 1 \leq k \leq \frac{N-1}{2}, \\ \mu_k(i) &= \sqrt{\frac{2}{l}} \sin \frac{2(N-k)\pi i}{N}, && \frac{N+1}{2} \leq k \leq N-1. \end{aligned} \tag{37}$$

The eigenfunctions (37) are orthonormal, and the eigenvalues λ_k satisfy the inequalities

$$0 = \lambda_0 < \lambda_k < \lambda_{\frac{N-1}{2}} = \frac{4}{h^2} \cos^2 \frac{\pi}{2N}, \quad 0 < k < N-1.$$

In addition, any periodic grid function $f(i)$ with period N (N odd), defined on the grid Ω, can be represented in the form

$$f(i) = \frac{2}{N} \sum_{k=0}^{(N-1)/2} \rho_k \varphi_k \cos \frac{2k\pi i}{N} + \frac{2}{N} \sum_{k=(N+1)/2}^{N-1} \varphi_k \sin \frac{2(N-k)\pi i}{N},$$

where

$$\varphi_k = \sum_{i=0}^{N-1} \rho_k f(i) \cos \frac{2k\pi i}{N}, \quad 0 \le k \le \frac{N-1}{2},$$

$$\varphi_k = \sum_{i=0}^{N-1} f(i) \sin \frac{2(N-k)\pi i}{N}, \quad \frac{N+1}{2} \le k \le N-1,$$

and where ρ_k is as defined above.

Chapter 2

The Elimination Method

In this chapter, we study several variants of a direct method for solving grid equations — the elimination method. The application of the method to the solution of both scalar and vector equations is considered.

In Section 1, the elimination method for scalar three-point equations is constructed and studied. Section 2 is devoted to several variants of the elimination method; flow, cyclic and non-monotonic elimination are considered here. In Section 3 we look at monotonic and non-monotonic elimination for five-point scalar equations. In Section 4, we construct block-elimination algorithms for two- and three-point vector equations, as well as the orthogonal elimination method for two-point equations.

2.1 The elimination method for three-point equations

2.1.1 The algorithm. In Chapter 1, methods were set forth for solving difference equations with constant coefficients. The current chapter is devoted to the construction of direct methods which solve boundary-value problems for three- and five-point difference equations with variable coefficients, and also three-point vector equations. Here we will study several variants of the elimination method, which is the Gaussian elimination method applied to a special system of linear algebraic equations, and which takes into account the band structure of the matrix of the system.

We will begin our study of the elimination method with the scalar-equation case. Suppose we must solve the following system of three-point equations

$$\begin{aligned} c_0 y_0 - b_0 y_1 &= f_0, && i = 0, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} &= f_i, && 1 \le i \le N-1, \\ -a_N y_{N-1} + c_N y_N &= f_N, && i = N, \end{aligned} \tag{1}$$

or, in vector form,

$$\mathcal{A}Y = F, \tag{2}$$

where $Y = (y_0, y_1, \ldots, y_N)^T$ is the vector of unknowns, $F = (f_0, f_1, \ldots, f_N)^T$ is the right-hand-side vector, and $\mathcal{A}$ is the square $(N+1) \times (N+1)$ matrix

$$\mathcal{A} = \begin{Vmatrix} c_0 & -b_0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 \\ -a_1 & c_1 & -b_1 & 0 & \ldots & 0 & 0 & 0 & 0 \\ 0 & -a_2 & c_2 & -b_2 & \ldots & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \ldots & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \ldots & -a_{N-2} & c_{N-2} & -b_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & -a_{N-1} & c_{N-1} & -b_{N-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & -a_N & c_N \end{Vmatrix}$$

with real or complex coefficients.

Systems of the form (1) arise from a three-point approximation to a boundary-value problem for second-order ordinary differential equations with constant and variable coefficients, and also when realizing difference schemes for equations with partial derivatives. In the latter case, we are usually required to solve, not a single problem (1), but a series of problems with different right-hand sides, where the number of problems in the series can be in the tens or hundreds and the number of unknowns in each problem is $N \approx 100$. Thus it is necessary to find efficient methods for solving problems of the form (1), where the number of operations is proportional to the number of unknowns. For the system (1), such a method is the *elimination method.*

The possibility of constructing an efficient method is restricted by the characteristics of the system (1). The matrix $\mathcal{A}$ corresponding to (1) belongs to the class of sparse matrices — of $(N+1)^2$ elements, no more than $3N+1$ are non-zero. Besides, it has a band structure (it is a tridiagonal matrix). Such a regular distribution of non-zero elements makes it possible to construct very simple computational formulas for solving the equation.

We now move on to construct the algorithm for solving the system (1). We first recall the sequence of operations for the Gaussian elimination method. At the first stage, the first equation is used to eliminate y_0 from all the other equations; then, the second equation is used to eliminate y_1 from equations $i = 2, 3, \ldots, N$ of the transformed system; and so forth. As a result, we obtain one equation in y_N. Here the forward path of the algorithm terminates. On the reverse path, we find y_i for $i = N-1, N-2, \ldots, 0$ from the transformed right-hand side and the already computed $y_{i+1}, y_{i+2}, \ldots, y_N$.

Following the idea of Gauss' method, we carry out the elimination of the unknowns in (1). We introduce the notation $\alpha_1 = b_0/c_0$, $\beta_1 = f_0/c_0$, and

write (1) in the following form

$$\begin{aligned} y_0 - \alpha_1 y_1 &= \beta_1, && i = 0, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} &= f_i, && 1 \le i \le N-1, \\ -a_N y_{N-1} + c_N y_N &= f_N, && i = N. \end{aligned} \tag{1'}$$

Take the first two equations of the system (1')

$$y_0 - \alpha_1 y_1 = \beta_1, \quad -a_1 y_0 + c_1 y_1 - b_1 y_2 = f_1.$$

Multiply the first equation by a_1 and add it to the second equation. We get $(c_1 - a_1\alpha_1)y_1 - b_1 y_2 = f_1 + a_1\beta_1$ or, after dividing by $c_1 - a_1\alpha_1$

$$y_1 - \alpha_2 y_2 = \beta_2, \quad \alpha_2 = \frac{b_1}{c_1 - \alpha_1 a_1}, \quad \beta_2 = \frac{f_1 + a_1 b_1}{c_1 - \alpha_1 a_1}.$$

All the remaining equations of the system (1') do not contain y_0, therefore this stage of the elimination process is completed. As a result we obtain a new "reduced" system

$$\begin{aligned} y_1 - \alpha_2 y_2 &= \beta_2, && i = 1, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} &= f_i, && 2 \le i \le N-1, \\ -a_N y_{N-1} + c_N y_N &= f_N, && i = N, \end{aligned} \tag{3}$$

which does not contain the unknown y_0 and which has a structure analogous to (1'). When this system has been solved, the unknown y_0 is found from the formula $y_0 = \alpha_1 y_1 + \beta_1$. We can apply the above described elimination procedure to the system (3). At the second stage, the unknown y_1 is eliminated, at the third y_2, and so forth. At the end of the l^{th} stage we obtain a system for the unknowns $y_l, y_{l+1}, \ldots, y_N$

$$\begin{aligned} y_l - \alpha_{l+1} y_{l+1} &= \beta_{l+1}, && i = l, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} &= f_i, && l+1 \le i \le N-1, \\ -a_N y_{N-1} + c_N y_N &= f_N, && i = N, \end{aligned} \tag{4}$$

and formulas for finding y_i for $i \le l-1$

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = l-1, l-2, \ldots, 0. \tag{5}$$

The coefficients α_i and β_i, clearly, are found from the formulas

$$\alpha_{i+1} = \frac{b_i}{c_i - a_i\alpha_i}; \ \beta_{i+1} = \frac{f_i + a_i\beta_i}{c_i - a_i\alpha_i}; \ i = 1, 2, \ldots; \ \alpha_1 = \frac{b_0}{c_0}, \ \beta_1 = \frac{f_0}{c_0}.$$

Substituting $l = N-1$ in (4), we obtain a system for y_N and y_{N-1}

$$y_{N-1} - \alpha_N y_N = \beta_N, \quad -a_N y_{N-1} + c_N y_N = f_N,$$

from which we find $y_N = \beta_{N+1}, y_{N-1} = \alpha_N y_N + \beta_N$.

Combining these equations with (5) ($l = N - 1$), we obtain the final formulas for finding the unknowns

$$\begin{aligned} &y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = N-1, N-2, \ldots, 0, \\ &y_N = \beta_{N+1}, \end{aligned} \tag{6}$$

where α_i and β_i are found from the recurrence formulas

$$\begin{aligned} &\alpha_{i+1} = \frac{b_i}{c_i - a_i \alpha_i}, \quad i = 1, 2, \ldots, N-1, \quad \alpha_1 = \frac{b_0}{c_0}, \\ &\beta_{i+1} = \frac{f_i + a_i \beta_i}{c_i - a_i \alpha_i}, \quad i = 1, 2, \ldots, N, \qquad \beta_1 = \frac{f_0}{c_0}. \end{aligned} \tag{8}$$

Thus, the formulas (6)–(8) describe Gauss' method which, when applied to the system (1), is given a special name — the *elimination method.* The coefficients α_i and β_i are called the *elimination coefficients*, formulas (7), (8) describe the *forward elimination path*, and (6) the *backward path.* Since the values y_i are found sequentially in reverse order, the formulas (6)–(8) are sometimes called the *right-elimination* formulas.

An elementary count of the arithmetic operations in (6)–(8) shows that realizing the elimination method using these formulas requires $3N$ multiplications, $2N + 1$ divisions and $3N$ additions and subtractions. If there is no difference between arithmetic operations, the total number of operations required for the elimination method is $Q = 8N + 1$. Of this total, $3N - 2$ operations are used for computing α_i, and $5N + 3$ operations for computing β_i and y_i.

Notice that the coefficients α_i do not depend on the right-hand side of the system (1), but are determined solely by the coefficients a_i, b_i, c_i of the difference equations. Therefore, if we must solve a series of problems (1) with different right-hand sides, but with the same matrix $\mathcal{A}$, then the elimination coefficients α_i are only computed for the first problem of the series. Thus, solving the first problem in the series costs $Q = 8N + 1$ operations, but solving each of the remaining problems only costs $5N + 3$ operations.

In conclusion we indicate the order of the computations for the formulas of the elimination method. Beginning with α_1 and β_1, we calculate and store α_i and β_i using (7) and (8). Then the solutions y_i is found using (6).

2.1.2 Two-sided elimination. Above, right-elimination formulas for solving the system (1) were obtained. The formulas for left elimination are derived analogously:

$$\xi_i = \frac{a_i}{c_i - b_i\xi_{i+1}}, \quad i = N-1, N-2, \ldots, 1, \quad \xi_N = \frac{a_N}{c_N}, \tag{9}$$

$$\eta_i = \frac{f_i + b_i\eta_{i+1}}{c_i - b_i\xi_{i+1}}, \quad i = N-1, N-2, \ldots, 0, \quad \eta_N = \frac{f_N}{c_N}, \tag{10}$$

$$y_{i+1} = \xi_{i+1}y_i + \eta_{i+1}, \quad i = 0, 1, \ldots, N-1, \qquad y_0 = \eta_0. \tag{11}$$

Here the values of y_i are found sequentially in order of increasing i (the left-hand ordering).

Sometimes it is convenient to combine left and right elimination, obtaining the socalled *two-sided-elimination method.* It is most appropriate to apply this method when it is only necessary to find one unknown, for example $y_m (0 \le m \le N)$ or a group of sequential unknowns. We now obtain the formulas for the two-sided-elimination method. Suppose $1 \le m \le N$ and use formulas (7), (10) to find $\alpha_1, \alpha_2, \ldots, \alpha_m, \beta_1, \beta_2, \ldots, \beta_m$ and $\xi_N, \xi_{N-1}, \ldots, \xi_m$, $\eta_N, \eta_{N-1}, \ldots, \eta_m$. We write out formulas (6), (11) for the reverse path of right and left elimination for $i = m-1$. We get the system

$$y_{m-1} = \alpha_m y_m + \beta_m, \quad y_m = \xi_m y_{m-1} + \eta_m,$$

from which we find y_m:

$$y_m = \frac{\eta_m + \xi_m\beta_m}{1 - \xi_m\alpha_m}.$$

Using y_m, we sequentially find $y_{m-1}, y_{m-1}, \ldots, y_0$ from (6) for $i = m-1, m-2, \ldots, 0$, and we compute $y_{m+1}, y_{m+2}, \ldots, y_N$ from (11) for $i = m, m+1, \ldots, N$.

Thus, the formulas for the two-sided elimination method have the form:

$$\begin{aligned}
\alpha_{i+1} &= \frac{b_i}{c_i - a_i\alpha_i}, \quad i = 1, 2, \ldots, m-1, & \alpha_1 &= \frac{b_0}{c_0}, \\
\beta_{i+1} &= \frac{f_i + a_i\beta_i}{c_i - a_i\alpha_i}, \quad i = 1, 2, \ldots, m-1, & \beta_1 &= \frac{f_0}{c_0}, \\
\xi_i &= \frac{a_i}{c_i - b_i\xi_{i+1}}, \quad i = N-1, N-2, \ldots, m, & \xi_N &= \frac{a_N}{c_N}, \\
\eta_i &= \frac{f_i + b_i\eta_{i+1}}{c_i - b_i\xi_{i+1}}, \quad i = N-1, N-2, \ldots, m, & \eta_N &= \frac{f_N}{c_N}
\end{aligned} \tag{12}$$

for computing the elimination coefficients and

$$\begin{aligned} y_i &= \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = m-1, m-2, \ldots, 0, \\ y_{i+1} &= \xi_{i+1} y_i + \eta_{i+1}, \quad i = m, m+1, \ldots, N-1, \\ y_m &= \frac{\eta_m + \xi_m \beta_m}{1 - \xi_m \alpha_m} \end{aligned} \tag{13}$$

for determining the solution.

It is obvious that the number of operations needed to find the solution of problem (1) using two-sided-elimination is just the same as for either left or right elimination, i.e., $Q \approx 8N$. Notice that for the special case of constant coefficients $a_i = b_i = 1$, $c_i = c$, for $i = 1, 2, \ldots, N-1$ and $b_0 = a_N = 0$, the number of operations can be reduced in the following way if N is an odd number. Suppose $N = 2M - 1$. Substitute $m = M$ in the formulas (12), (13) for the two-sided-elimination method. Then $\xi_{N-i+1} = \alpha_i$, $i = 1, 2, \ldots, M$. Consequently, the elimination coefficient ξ_i need not be found, and the formulas for the two-sided-elimination method will have the form

$$\begin{aligned} \alpha_{i+1} &= \frac{1}{c - \alpha_i}, && i = 1, 2, \ldots, M-1, && \alpha_1 = 0, \\ \beta_{i+1} &= (f_i + \beta_i)\alpha_{i+1}, && i = 1, 2, \ldots, M-1, && \beta_1 = \frac{f_0}{c_0}, \\ \eta_i &= (f_i + \eta_{i+1})\alpha_{N-i+1}, && i = N-1, N-2, \ldots, M, && \eta_N = \frac{f_N}{c_N}, \\ y_i &= \alpha_{i+1} y_{i+1} + \beta_{i+1}, && i = M-1, M-2, \ldots, 0, && \\ y_{i+1} &= \alpha_{N-i} y_i + \eta_{i+1}, && i = M, M+1, \ldots, N-1, && \end{aligned}$$

where $y_M = (\eta_M + \alpha_M \beta_M)/(1 - \alpha_M^2)$.

2.1.3 Justification of the elimination method. Above we obtained formulas for the elimination method without any assumptions about the coefficients of the system (1). Here we consider what requirements these coefficients must satisfy, in order that the method can be applied and the solution obtained with sufficient accuracy.

Let us clarify the situation. Since the computational formulas (6)–(8) of the elimination method contain division operations, it is necessary that the denominator $c_i - a_i \alpha_i$ in (7), (8) be non-zero. We will say that the algorithm for the right-elimination method is *correct* if $c_i - a_i \alpha_i \neq 0$ for $i = 1, 2, \ldots, N$. Later the solution y_i is found from the recurrence formula (6). This formula can induce accumulation of the rounding errors from the results of the arithmetic operations. In fact, suppose the elimination coefficients α_i and β_i are found exactly, and that the computation of y_N results in an error ϵ_N,

i.e., $\tilde{y}_N = y_N + \epsilon_N$. Since the solution $\tilde{y}_i$ is found using the formulas (6) $\tilde{y}_i = \alpha_{i+1}\tilde{y}_{i+1} + \beta_{i+1}$, $i = N-1, N-2, \ldots, 0$, the error $\epsilon_i = \tilde{y}_i - y_i$ will obviously satisfy the homogeneous equation $\epsilon_i = \alpha_{i+1}\epsilon_{i+1}$, $i = N-1, N-1, \ldots, 0$. From this it follows that, if all the α_i are greater than one in modulus, then it is possible to produce a large increase in the error ϵ_0, and if N is sufficiently large, the computed solution $\tilde{y}_i$ will be significantly different from the desired solution y_i.

Being unable to present a more detailed discussion of the questions of computational stability and the mechanism whereby instability arises, we formulate the requirements usually presented for the elimination method. We will require that the elimination coefficients α_i not exceed one in modulus. This is sufficient to guarantee that the error ϵ_i will not grow in the above-considered model of the situation. If the condition $|\alpha_i| \leq 1$ is satisfied, then we shall say that the right-elimination algorithm is *stable*.

We now clarify the correctness and stability conditions for the algorithm (6)–(8). The following lemma contains sufficient conditions for the correctness and stability of the right-elimination algorithm.

Lemma 1. *Suppose the coefficients of the system* (1) *are real and satisfy the conditions*

$$|b_0| \geq 0,\ |a_N| \geq 0,\ |c_0| > 0,\ |c_N| > 0,\ |a_i| > 0,\ |b_i| > 0, \quad i = 1, 2, \ldots, N-1,$$

$$|c_i| \geq |a_i| + |b_i|, \quad i = 1, 2, \ldots, N-1, \tag{14}$$

$$|c_0| \geq |b_0|, \quad |c_N| \geq |a_N|, \tag{15}$$

where at least one of the inequalities in (14) *or* (15) *is strict, i.e., the matrix* $\mathcal{A}$ *is diagonally-dominant. Then for the algorithm* (6)–(8) *of the elimination method we have*

$$c_i - a_i\alpha_i \neq 0, \quad |\alpha_i| \leq 1, \quad i = 1, 2, \ldots, N-1,$$

guaranteeing the correctness and stability of the method.

Proof. The proof of the lemma is carried out by induction. From the conditions of the lemma and (7) it follows that

$$0 \leq |\alpha_i| = \frac{|b_0|}{|c_0|} \leq 1. \tag{16}$$

We will show that from the inequalities $|\alpha_i| \leq 1$ $(i \leq N-1)$ and the conditions of the lemma it follows that

$$c_i - a_i\alpha_i \neq 0, \quad |\alpha_{i+1}| \leq 1, \quad i \leq N-1. \tag{17}$$

Then, using (16) we obtain that $|\alpha_i| \leq 1$ for $i = 1, 2, \ldots, N$ and $c_i - a_i\alpha_i \neq 0$ for $i = 1, 2, \ldots, N-1$. To complete the proof of the lemma, it remains to show that $c_N - a_N\alpha_N \neq 0$. Thus, we first establish (17). Suppose $|\alpha_i| \leq 1, i \leq N-1$. Then from (14)

$$|c_i - a_i\alpha_i| \geq |c_i| - |a_i||\alpha_i| \geq |b_i| + |a_i|(1 - |\alpha_i|) \geq |b_i| > 0, \qquad (18)$$

and consequently $c_i - a_i\alpha_i \neq 0$. Further, from (7) and (18) we obtain

$$|\alpha_{i+1}| = \frac{|b_i|}{|c_i - a_i\alpha_i|} \leq \frac{|b_i|}{|b_i|} = 1,$$

which is what we were required to prove.

It remains to prove that $c_N - a_N\alpha_N \neq 0$. For this we use the assumption that at least one of the inequalities (14) or (15) is strict. There are several possible cases. If $|c_N| > |a_N|$, then from $|a_N| \leq 1$ it follows that $c_N - a_N\alpha_N \neq 0$. If the inequality (14) is strict for some $i_0, 1 \leq i_0 \leq N-1$, then from (18) we obtain that $|c_{i_0} - a_{i_0}\alpha_{i_0}| > |b_{i_0}|$, and consequently we have that $|a_{i_0+1}| < 1$. By induction it is easy to establish that $|\alpha_i| < 1$ for $i \geq i_0 + 1$. Consequently, in this case we have that $|\alpha_N| < 1$, and therefore $c_N - a_N\alpha_N \neq 0$. If $|c_0| > |b_0|$, then $|\alpha_i| < 1$, starting with $i = 1$. Therefore we again obtain $|\alpha_N| < 1$ and $c_N - a_N\alpha_N \neq 0$. The lemma is proved. □

Remark 1. The correctness and stability conditions for the algorithm (6)–(8) formulated in lemma 1 are only sufficient conditions. These conditions can be weakened, allowing certain of the coefficients a_i and b_i to be zero. So, for example, if for some $1 \leq m \leq N-1$ it occurs that $a_m = 0$, then the system (1) splits into two systems:

$$\begin{aligned} c_m y_m - b_m y_{m+1} &= f_m, && i = m, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} &= f_i, && m+1 \leq i \leq N-1, \\ -a_N y_{N-1} + c_N y_N &= f_N, && i = N, \end{aligned}$$

for the unknowns $y_m, y_{m+1}, \ldots, y_N$ and

$$\begin{aligned} c_0 y_0 - b_0 y_1 &= f_0, && i = 0, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} &= f_i, && 1 \leq i \leq m-2, \\ -a_{m-1} y_{m-2} + c_{m-1} y_{m-1} &= f_{m-1} + b_{m-1} y_m \end{aligned}$$

for the unknowns $y_0, y_1, \ldots, y_{m-1}$. The algorithm (6)–(8) can be applied to each of these systems if they fulfill the conditions of lemma 1. But in this case the formulas (6)–(8) can be used to find the solution of the whole split system (1), and the algorithm will be correct and stable.

Remark 2. The conditions of lemma 1 guarantee the correctness and stability of the left- and two-sided elimination algorithms. The conditions can also be used for the case of a system (1) with complex coefficients a_i, b_i, and c_i.

We now show that, if the conditions of lemma 1 are satisfied, then the system (1) has a unique solution for any right-hand side. In fact, taking into account relation (7), it is possible to show, using direct multiplication of matrices, that the matrix $\mathcal{A}$ of the system (1) can be represented as the product of two triangular matrices L and U.

$$\mathcal{A} = LU,$$

where

$$L = \begin{Vmatrix} c_0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ -a_1 & \Delta_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & -a_2 & \Delta_2 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_3 & \Delta_3 & \dots & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & \Delta_{N-3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & -a_{N-2} & \Delta_{N-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & -a_{N-1} & \Delta_{N-1} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & -a_N & \Delta_N \end{Vmatrix}$$

$$U = \begin{Vmatrix} 1 & -\alpha_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -\alpha_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & -\alpha_3 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & 1 & -\alpha_{N-1} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & -\alpha_N \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{Vmatrix}$$

and $\Delta_i = c_i - a_i\alpha_i$, $i = 1, 2, \dots, N$. Since

$$\det \mathcal{A} = \det L \cdot \det U = c_0 \prod_{i=1}^{N} \Delta_i,$$

and by lemma 1, $c_0 \neq 0$ and $\Delta_i \neq 0$ for $i = 1, 2, \dots, N$, then $\det \mathcal{A} \neq 0$. Therefore the system (1) has a unique solution when the conditions of lemma 1 are satisfied, and this solution can be found by the elimination method (6)–(8).

2.1.4 Sample applications of the elimination method. We now consider several examples applying the elimination method described above.

Example 1. A boundary-value problem of the first kind. Suppose we are required to solve the following problem:

$$\begin{gathered}(k(x)u'(x))' - q(x)u(x) = -f(x), \quad 0 < x < l, \\ u(0) = \mu_1, \quad u(l) = \mu_2, \quad k(x) \geq c_1 > 0, \quad q(x) \geq 0.\end{gathered} \tag{19}$$

On the interval $0 \leq x \leq l$ we construct an arbitrary non-uniform grid $\bar{\omega} = \{x_i \in [0,l],\ i = 0,1,\ldots,N,\ x_0 = 0,\ x_N = l\}$ with steps $h_i = x_i - x_{i-1}$, $i = 1,2,\ldots,N$ and replace (19) by the following difference problem:

$$\begin{gathered}(ay_{\bar{x}})_{\hat{x},i} - d_i y_i = -\varphi_i, \quad 1 \leq i \leq N-1, \\ y_0 = \mu_1, \quad y_N = \mu_2,\end{gathered} \tag{20}$$

where $d_i = q(x_i), \varphi_i = f(x_i)$, and for a_i we use the simplest approximation to the coefficient $k(x)$: $a_i = k(x_i - 0.5h_i)$. Writing out the difference derivative in (20) at a point

$$(ay_{\bar{x}})_{\hat{x},i} = \frac{1}{\hbar_i}\left(a_{i+1}\frac{y_{i+1} - y_i}{h_{i+1}} - a_i\frac{y_i - y_{i-1}}{h_i}\right),$$

where $\hbar_i = 0.5(h_i + h_{i+1})$ is the average step at the point x_i, we obtain the problem (20) written in the form of a system

$$\begin{aligned} C_0 y_0 - B_0 y_1 &= f_0, && i = 0, \\ -A_i y_{i-1} + C_i y_i - B_i y_{i+1} &= f_i, && 1 \leq i \leq N-1, \\ -A_N y_{N-1} + C_N y_N &= f_N, && i = N. \end{aligned} \tag{1''}$$

Here

$$\begin{gathered}B_0 = A_N = 0, \quad C_0 = C_N = 1, \quad f_0 = \mu_1, \quad f_N = \mu_2, \quad f_i = \varphi_i, \\ A_i = \frac{a_i}{\hbar_i h_i}, \qquad B_i = \frac{a_{i+1}}{\hbar_i h_{i+1}}, \quad C_i = A_i + B_i + d_i, 1 \leq i \leq N-1.\end{gathered} \tag{21}$$

By construction, the coefficients a_i and d_i of the difference scheme (20) satisfy the following conditions: $a_i \geq c_1 > 0$, $d_i \geq 0$. Therefore from (21) it follows that for (1'') the conditions of lemma 1 are satisfied, and this problem can be solved by the elimination method.

Example 2. A boundary-value problem of the third kind. We now consider the case of boundary conditions of the third kind:

$$\begin{aligned}(k(x)u'(x))' - q(x)u(x) &= -f(x), \quad 0 < x < l,\\ k(0)u'(0) &= \kappa_1 u(0) - \mu_1,\\ -k(l)u'(l) &= \kappa_2 u(l) - \mu_2.\end{aligned} \tag{22}$$

We will assume that the following conditions are satisfied: $k(x) \geq c_1 > 0$, $q(x) \geq 0$, $\kappa_1 \geq 0$, $\kappa_2 \geq 0$, where if $q(x) \equiv 0$, then $\kappa_1^2 + \kappa_2^2 \neq 0$.

On the non-uniform grid introduced above, the problem (22) is approximated by the following difference scheme:

$$\begin{aligned}(ay_{\bar{x}})_{\hat{x},i} - d_i y_i &= -\varphi_i, \quad 1 \leq i \leq N-1,\\ \frac{2}{h_1} a_1 y_{x,0} &= \left(d_0 + \frac{2}{h_1}\kappa_1\right) y_0 - \left(\varphi_0 + \frac{2}{h_1}\mu_1\right), \quad i = 0,\\ -\frac{2}{h_N} a_N y_{\bar{x},N} &= \left(d_N + \frac{2}{h_N}\kappa_2\right) y_N - \left(\varphi_N + \frac{2}{h_N}\mu_2\right), \quad i = N,\end{aligned} \tag{23}$$

where the coefficients a_i, d_i and φ_i are chosen as indicated in Example 1. Writing out the second difference derivative $(ay_{\bar{x}})_{\hat{x}}$ at a point, and also the first derivatives

$$y_{x,i} = \frac{y_{i+1} - y_i}{h_{i+1}}, \quad y_{\bar{x},i} = \frac{y_i - y_{i-1}}{h_i},$$

we bring (23) into the form of $(1'')$ where

$$B_0 = \frac{2a_1}{h_1^2}, \; A_N = \frac{2a_N}{h_N^2}, \; C_0 = B_0 + d_0 + \frac{2}{h_1}\kappa_1,$$

$$C_N = A_N + d_N + \frac{2}{h_N}\kappa_2, \; f_0 = \varphi_0 + \frac{2}{h_1}\mu_1, \; f_N = \varphi_N + \frac{2}{h_N}\mu_2,$$

$$A_i = \frac{a_i}{h_i h_i}, \; B_i = \frac{a_{i+1}}{h_i h_{i+1}}, \; C_i = A_i + B_i + d_i, \; f_i = \varphi_i, \; 1 \leq i \leq N-1.$$

It is easy to check that, in this case also, the conditions of lemma 1 are satisfied.

Example 3. Difference schemes for a thermal-flow equation. We now consider a boundary-value problem of the first kind for a thermal-flow equation:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < l, \quad t > 0,\\ u(0,t) &= \mu_1(t), \quad u(l,t) = \mu_2(t),\\ u(x,0) &= u_0(x).\end{aligned} \tag{24}$$

On the plane (x,t) we introduce the grid $\bar{\omega} = \{(x_i,t_n),\ x_i = ih,\ i = 0,1,\ldots,N,\ h = l/N,\ t_n = n\tau,\ n = 0,1,\ldots\}$ with step h in space and τ in time. We approximate (24) by the difference scheme

$$\begin{aligned} &y_{t,i} = \sigma y^{n+1}_{\bar{x}x,i} + (1-\sigma)y^n_{\bar{x}x,i}, \quad 1 \le i \le N-1, \\ &y^n_0 = \mu_1(t_n), \quad y^n_N = \mu_2(t_n), \quad y^0_i = u_0(x_i), \quad n = 0,1,\ldots, \end{aligned} \tag{25}$$

where σ is a real parameter, $y^n_i = y(x_i,t_n)$,

$$y_{\bar{x}x,i} = \frac{1}{h^2}(y_{i+1} - 2y_i + y_{i-1}), \quad y_{t,i} = \frac{1}{\tau}(y^{n+1}_i - y^n_i). \tag{26}$$

It is known (see, for example, [9]), that the scheme (25) has order of approximation $O(\tau + h^2)$ for any σ, $O(\tau^2 + h^2)$ for $\sigma = 0.5$, and $O(\tau^2 + h^4)$ for $\sigma = 1/2 - h^2/(12\tau)$. The stability condition for the scheme (25) with respect to the initial data has the form

$$\sigma \ge 1/2 - h^2/(4\tau). \tag{27}$$

We turn now to the method for solving the equations (25) for y^{n+1}_i. Assuming that y^n_i is already known, we write (25) in the following form:

$$\begin{aligned} &\frac{1}{\sigma\tau}y^{n+1}_i - y^{n+1}_{\bar{x}x,i} = \varphi^n_i, \quad 1 \le i \le N-1, \\ &y^{n+1}_0 = \mu_1(t_{n+1}), \quad y^{n+1}_N = \mu_2(t_{n+1}), \end{aligned}$$

where

$$\varphi^n_i = \frac{1}{\sigma\tau}y^n_i + \left(\frac{1}{\sigma} - 1\right)y^n_{\bar{x}x,i}$$

if $\sigma \ne 0$. Using (26), we bring this scheme to the form of $(1'')$, where

$$\begin{aligned} &B_0 = A_N = 0, \quad C_0 = C_N = 1, \qquad f_0 = \mu_1(t_{n+1}),\ f_N = \mu_2(t_{n+1}), \\ &A_i = B_i = \frac{1}{h^2}, \quad C_i = A_i + B_i + \frac{1}{\sigma\tau}, \quad f_i = \varphi^n_i, \quad 1 \le i \le N-1. \end{aligned}$$

We now find the conditions under which the constructed system $(1'')$ can be solved using the elimination method. From lemma 1 it follows that the condition $|2/h^2 + 1/(\sigma\tau)| \ge 2/h^2$ must be satisfied. Solving this inequality, we find a sufficient condition $\sigma \ge -h^2/(4\tau)$ for applying elimination. Comparing this inequality with (27), we obtain that, if the stability condition (27) is satisfied for the scheme (25), then the elimination method can be used to find the solution at the new level.

Example 4. The non-stationary Schroedinger equation.

We now consider the non-stationary Schroedinger equation

$$\begin{aligned} i\frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2}, & 0 < x < l,\ t > 0, \\ u(0,t) &= u(l,t) = 0, & u(0,x) = u_0(x),\ i = \sqrt{-1}. \end{aligned}$$

For this equation, as for the thermal-flow equation (24), it is possible to construct a two-level scheme with weights

$$\begin{aligned} iy_{t,k} &= \sigma y^{n+1}_{\bar{x}x,k} + (1-\sigma) y^n_{\bar{x}x,k}, \quad 1 \le k \le N-1, \\ y^n_0 &= y^n_N = 0, \quad y^0_k = u_0(x_k), \end{aligned} \tag{28}$$

where the parameter $\sigma = \sigma_0 + i\sigma_1$ can take on complex values. The scheme (28) has approximation error $O(\tau + h^2)$ for any σ, $O(\tau^2 + h^2)$ for $\sigma = 0.5$, and $O(\tau^2 + h^4)$ for $\sigma = 1/2 - h^2 i/(12\tau)$. The stability condition with respect to the initial data has the form

$$\sigma = \operatorname{Re}\sigma \ge 0.5. \tag{29}$$

The scheme (28) is usually transformed to the system (1″), and the conditions of lemma 1 assume the following form: $|2/h^2 + i/(\sigma\tau)| \ge 2/h^2$. Solving this inequality, we obtain that the elimination method will be correct for finding the solution of the scheme (28) at the new level if the condition $\sigma_1 = \operatorname{Im}\sigma \ge -h^2/(4\tau)$ is satisfied.

Thus, for this example, the applicability condition for the elimination method does not coincide with the stability condition for the same difference scheme with respect to the initial data.

2.2 Variants of the elimination method

2.2.1 The flow variant of the elimination method. We will now look at a variant of the elimination method which is used to solve difference problems with widely varying coefficients. Examples of such problems are thermal-flow hydrodynamics and magnito-hydrodynamics problems, where the coefficient of thermal-flow or conductivity depends on the thermodynamic parameters of the medium. In the case of thermal problems, it is possible to have adiabatic cells, where the thermal-flow is infinitely large. In magnetic problems this corresponds to ideally conductive or non-conductive cells.

Often in such problems it is necessary to find, in addition to the solution, the heat flow (the thermal problem). Using the usual elimination formulas to solve the second-order difference equations produced by the difference schemes for these problems leads to a considerable loss of accuracy. The preceding investigation of numerical differentiation for flow problems leads to unsatisfactory results. Getting around this deficiency leads us to the so called *flow variant of the elimination method.* The formulas for this variant of elimination can be obtained by transforming the formulas for regular elimination.

Thus, we consider a boundary-value difference problem

$$\begin{aligned} -a_i y_{i-1} + c_i y_i - a_{i+1} y_{i+1} &= f_i, \qquad 1 \le i \le N-1, \\ y_0 - \kappa_1 y_1 &= \mu_1, \quad y_N - \kappa_2 y_{N-1} = \mu_2, \end{aligned} \tag{1}$$

where

$$c_i = a_i + a_{i+1} + d_i, \quad 0 < a_i < \infty, \tag{2}$$

$$d_i > 0, \quad i = 1, 2, \ldots, N-1, \quad |\kappa_1| \le 1, \quad |\kappa_2| \le 1. \tag{3}$$

The formulas for right elimination (see (2.1.6)–(2.1.8)) for problem (1) have the form (taking into account (2))

$$y_i = \bar{\alpha}_{i+1} y_{i+1} + \bar{\beta}_{i+1}, \quad i = N-1, N-2, \ldots, 0, \quad y_N = \frac{\mu_2 + \kappa_2 \bar{\beta}_N}{1 - \kappa_2 \bar{\alpha}_N} \tag{4}$$

$$\bar{\alpha}_{i+1} = \frac{a_{i+1}}{a_{i+1} + d_i + a_i(1 - \bar{\alpha}_i)}, \quad i = 1, 2, \ldots, N-1, \ \bar{\alpha}_1 = \kappa_1, \tag{5}$$

$$\bar{\beta}_{i+1} = (f_i + a_i \bar{\beta}_i) \frac{\bar{\alpha}_{i+1}}{a_{i+1}}, \quad i = 1, 2, \ldots, N-1, \ \bar{\beta}_1 = \mu_1. \tag{6}$$

We now introduce a new unknown grid function (the flow) with the formula

$$w_i = -a_i(y_i - y_{i-1}), \quad i = 1, 2, \ldots, N, \tag{7}$$

and rewrite (1) in the form

$$\begin{aligned} w_{i+1} - w_i + d_i y_i &= f_i, && 1 \le i \le N-1, \\ y_0 - \kappa_1 y_1 &= \mu_1, && i = 0, \\ -\kappa_2 w_N + a_N(1 - \kappa_2) y_N &= a_N \mu_2, && i = N. \end{aligned} \tag{8}$$

From (7) we find

$$y_i = y_{i+1} + \frac{1}{a_{i+1}} w_{i+1}, \quad i = 0, 1, \ldots, N-1,$$

and we substitute this expression in (4). As a result we find a relation connecting y_{i+1} with w_{i+1}:

$$w_{i+1} + a_{i+1}(1 - \bar{\alpha}_{i+1})y_{i+1} = a_{i+1}\bar{\beta}_{i+1}, \quad i = 0, 1, \ldots, N-1.$$

Introducing the notation

$$\alpha_i = a_i(1 - \bar{\alpha}_i), \quad \beta_i = \alpha_i\bar{\beta}_i, \quad i = 1, 2, \ldots, N,$$

we rewrite this relation in the form

$$w_i + \alpha_i y_i = \beta_i, \quad i = 1, 2, \ldots, N. \tag{9}$$

Notice that (8), (9) form an algebraic system containing $2N+1$ equations in the $2N+1$ variables $y_0, y_1, \ldots, y_N$ and $w_1, w_2, \ldots, w_N$. The structure of this system is such that it splits into two independent systems in the variables $y_0, y_1, \ldots, y_N$ and $w_1, w_2, \ldots, w_N$. Let us construct this system

From (9) we express y_i as $y_i = (\beta_i - w_i)/\alpha_i$, $i = 1, 2, \ldots, N$ and substitute it in (8) for $i = 1, 2, \ldots, N$. As a result we obtain the equations

$$\begin{aligned} w_i &= \frac{\alpha_i}{\alpha_i + d_i} w_{i+1} + \frac{d_i\beta_i - \alpha_i f_i}{\alpha_i + d_i}, \quad i = N-1, N-2\ldots., 1, \\ w_N &= \frac{a_N[(1-\kappa_2)\beta_N - \alpha_N\mu_2]}{(1-\kappa_2)a_N + \alpha_N\kappa_2}, \end{aligned} \tag{10}$$

which we can solve sequentially to find all the w_i.

We will now obtain the equations for y_i. For this we use (9) to express w_i as $w_i = -\alpha_i y_i + \beta_i$, $i = 1, 2, \ldots, N$ and substitute in (8) for $i = 1, 2, \ldots, N$. As a result we obtain the equations

$$\begin{aligned} y_i &= \frac{\alpha_{i+1}}{\alpha_i + d_i} y_{i+1} + \frac{f_i - \beta_{i+1} + \beta_i}{\alpha_i + d_i}, \quad i = N-1, N-2, \ldots, 1, \\ y_0 &= \kappa_1 y_1 + \mu_1, \\ y_N &= \frac{\kappa_2\beta_N + a_N\mu_2}{(1-\kappa_2)a_N + \alpha_N\kappa_2} \end{aligned} \tag{11}$$

for sequentially computing y_i.

We shall now give the recurrence formulas for determining α_i and β_i. Using (5) and (6) we find

$$\alpha_{i+1} = a_{i+1}(1 - \bar{\alpha}_{i+1}) = \frac{a_{i+1}[a_i(1 - \bar{\alpha}_i) + d_i]}{a_{i+1} + d_i + a_i(1 - \bar{\alpha}_i)} = \frac{a_{i+1}(\alpha_i + d_i)}{a_{i+1} + \alpha_i + d_i}, \quad i = 1, 2, \ldots, N-1, \quad \alpha_1 = a_1(1 - \kappa_1), \tag{12}$$

$$\beta_{i+1} = a_{i+1}\bar{\beta}_{i+1} = \frac{a_{i+1}(f_i + \beta_i)}{a_{i+1} + \alpha_i + d_i}, \quad i = 1, 2, \ldots, N-1, \quad \beta_1 = a_1\mu_1. \tag{13}$$

From the conditions (2), (3) and formula (12) it follows that $\alpha_i \geq 0$. Since the coefficient $\alpha_i/(\alpha_i + d_i)$ in (10) is not greater than one, the algorithm for computing w_i is guaranteed to be stable. Further, since it follows from the conditions $\alpha_i \geq 0$ and $d_i > 0$ that $a_{i+1} < a_{i+1} + \alpha_i + d_i$, we have from (12) the inequality $\alpha_{i+1} < \alpha_i + d_i$. Therefore, the coefficient $\alpha_{i+1}/(\alpha_i + d_i)$ in (11) is always less than one, and this guarantess that the computation of y_i is stable. Notice that the denominator in the expressions for w_N and y_N is always greater than zero.

Thus, the algorithm for flow elimination is described by (10)–(13). Notice that the indicated recurrence relations for α_i and β_i, and also the expressions for y_N and w_N are appropriate if $a_{i+1} < 1$. If $a_{i+1} \geq 1$, then it is recommended that the following formulas be used, which were obtained from (10)–(13) by dividing the numerator and denominator by a_{i+1}:

$$\alpha_{i+1} = \frac{\alpha_i + d_i}{1 + (\alpha_i + d_i)/a_{i+1}}, \quad \beta_{i+1} = \frac{f_i + \beta_i}{1 + (\alpha_i + d_i)/a_{i+1}},$$

$$y_N = \frac{\kappa_2\beta_N/a_N + \mu_2}{1 - \kappa_2 + \kappa_2\alpha_N/a_N}, \quad w_N = \frac{(1 - \kappa_2)\beta_N - \alpha_N\mu_2}{1 - \kappa_2 + \kappa_2\alpha_N/a_N}.$$

Let us now compute the number of arithmetic operations which are required to realize (10)–(13). With a reasonable organization of the computation, where expressions which occur in several expressions are computed only once, and where the general multiplier for several terms is extracted, the number of operations required for (10)–(13) is $Q = 21N + 1$. This is approximately twice as many operations as were required by the usual elimination method to find y_i for problem (1), but in addition the flow w_i has been found from formula (7).

2.2.2 The cyclic elimination method. Let us now consider the following system

$$-a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \quad i = 0, \pm 1, \pm 2, \ldots, \tag{14}$$

the coefficients and right-hand side of which are periodic with period N:

$$a_i = a_{i+N}, \quad b_i = b_{i+N}, \quad c_i = c_{i+N}, \quad f_i = f_{i+N}. \tag{15}$$

Systems of the type (14), (15) arise, for example, from three-point difference schemes designed to find periodic solutions of second-order ordinary differential equations, and also when approximating the solutions of equations with partial derivatives in cylindrical and spherical coordinates.

A solution of the system (14) satisfying the conditions (15) will, if it exists, also be periodic with period N, i.e.,

$$y_i = y_{i+N}. \tag{16}$$

Therefore it is sufficient to find the solution at, for example, $i = 0, 1, \ldots, N-1$. In this case, the problem (14)–(16) can be written as:

$$\begin{aligned} &-a_0 y_{N-1} + c_0 y_0 - b_0 y_1 = f_0, \quad i = 0 \\ &-a_i y_{i-1} + c_i y_i - c_i y_{i+1} = f_i, \quad 1 \le i \le N-1, \end{aligned} \tag{17}$$

$$y_N = y_0. \tag{18}$$

We appended the condition (18) to the system (17) so that the equations for $i = N-1$ would not include y_N, it having been replaced by y_0. This allows us to retain a unique form for the equations (17) for $i = 1, 2, \ldots, N-1$.

If we introduce the vector of unknowns $Y = (y_0, y_1, \ldots, y_{N-1})^T$ and the right-hand side $F = (f_o, f_1, \ldots, f_{N-1})^T$, then (17), (18) can be written in the vector form $\mathcal{A}Y = F$ where

$$\mathcal{A} = \begin{Vmatrix} c_0 & -b_0 & 0 & 0 & \ldots & 0 & 0 & -a_0 \\ -a_1 & c_1 & -b_1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & -a_2 & c_2 & -b_2 & \ldots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \ldots & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \ldots & c_{N-3} & -b_{N-3} & 0 \\ 0 & 0 & 0 & 0 & \ldots & -a_{N-2} & c_{N-2} & -b_{N-2} \\ -b_{N-1} & 0 & 0 & 0 & \ldots & 0 & -a_{N-1} & c_{N-1} \end{Vmatrix}$$

is the matrix of the system (17), (18). The presence of non-zero coefficients a_0 and b_{N-1} in (17) does not allow us to solve this system using the elimination method described in Section 1. To find the solution of the system (17), (18) we construct a variant of the elimination method called the *cyclic elimination method.*

The solution of the problem (17), (18) will be found in the form of a linear combination of the grid functions u_i and v_i

$$y_i = u_i + y_0 v_i, \quad 0 \le i \le N, \tag{19}$$

where u_i is the solution of the non-homogeneous three-point boundary-value problem

$$\begin{aligned} -a_i u_{i-1} + c_i u_i - b_i u_{i+1} &= f_i, \quad 1 \le i \le N-1, \\ u_0 = 0, \quad u_N &= 0 \end{aligned} \tag{20}$$

with homogeneous boundary conditions, and v_i is the solution of the homogeneous three-point boundary-value problem

$$\begin{aligned} -a_i v_{i-1} + c_i v_i - b_i v_{i+1} &= 0, \quad 1 \le i \le N-1, \\ v_0 = 1, \quad v_N &= 1 \end{aligned} \tag{21}$$

with non-homogeneous boundary conditions.

We now find under what conditions y_i from (19) is the desired solution. Multiplying (21) by y_0, adding it to (20), and taking into account (19), we find that the equations in (17) can be satisfied for $i = 1, 2, \ldots, N-1$. From the boundary conditions for u_i and v_i it follows that (18) will be satisfied. Thus, if y_i satisfied the remaining unused equation at $i = 0$ in (17), the problem would be solved. Substituting (19) in this equation, we obtain

$$-a_0 u_{N-1} - a_0 y_0 v_{N-1} + c_0 y_0 - b_0 u_1 - b_0 y_0 v_1 = f_0. \tag{22}$$

Thus, if we choose y_0 from the formula

$$y_0 = \frac{f_0 - a_0 u_{N-1} + b_0 u_1}{c_0 - a_0 v_{N-1} - b_0 v_1}, \tag{23}$$

then (22) will be satisfied, and consequently the solution of the problem (17), (18) can be found from (19).

We are left with solving (20) and (21). They are particular cases of the three-point systems of equations solved in Section 1 using the elimination method. For (20) and (21), the elimination formulas have the following form:

$$\begin{aligned} u_i &= \alpha_{i+1} u_{i+1} + \beta_{i+1}, \quad i = N-1, N-2, \ldots, 1, \quad u_N = 0, \\ v_i &= \alpha_{i+1} v_{i+1} + \alpha_{i+1}, \quad i = N-1, N-2, \ldots, 1, \quad v_N = 1, \end{aligned} \tag{24}$$

where the elimination coefficients α_i, β_i and γ_i are found from the following formulas

$$\alpha_{i+1} = \frac{b_i}{c_1 - a_i \alpha_i}, \quad i = 1, 2, \ldots, N, \quad \alpha_1 = 0, \tag{25}$$

$$\beta_{i+1} = \frac{f_i + a_i\beta_i}{c_i - a_i\alpha_i}, \quad i = 1,2,\ldots,N, \quad \beta_1 = 0, \tag{26}$$

$$\gamma_{i+1} = \frac{a_i\gamma_i}{c_i - a_i\alpha_i}, \quad i = 1,2,\ldots,N, \quad \gamma_1 = 1, \tag{27}$$

Let us transform (23). From (24) we obtain $u_{N-1} = \alpha_N u_N + \beta_N = \beta_N$, $v_{N-1} = \gamma_N + \alpha_N$. We substitute these expressions in (23) and take into account (15), (25)–(27):

$$y_0 = \frac{f_N + a_N\beta_N + \beta_N u_1}{c_N - a_N\alpha_N - a_N\gamma_N - b_N v_1} = \frac{\beta_{N+1} + \alpha_{N+1}u_1}{1 - \gamma_{N+1} - \alpha_{N+1}v_1}.$$

We have constructed an algorithm for solving problem (17), (18), called the method of cyclic elimination:

$$\begin{gathered}
\alpha_2 = b_1/c_1, \quad \beta_2 = f_1/c_1, \quad \gamma_2 = a_1/c_1, \\
\alpha_{i+1} = \frac{b_i}{c_i - a_i\alpha_i}, \ \beta_{i+1} = \frac{f_i + a_i\beta_i}{c_i - a_i\alpha_i}, \ \gamma_{i+1} = \frac{a_i\gamma_i}{c_i - a_i\alpha_i}, \ i = 2,3,\ldots,N; \\
u_{N-1} = \beta_N, \quad v_{N-1} = \alpha_N + \gamma_N, \\
u_i = \alpha_{i+1}u_{i+1} + \beta_{i+1}, \ v_i = \alpha_{i+1}v_{i+1} + \gamma_{i+1}, \ i = N-2, N-3, \ldots, 1; \\
y_0 = \frac{\beta_{N+1} + \alpha_{N+1}u_1}{1 - \gamma_{N+1} - \alpha_{N+1}v_1}, \quad y_i = u_i + y_0 v_i, \quad i = 1,2,\ldots,N-1.
\end{gathered} \tag{28}$$

An elementary computation indicates that the algorithm requires $6(N-1)$ multiplications, $5N-3$ additions and subtractions, and $3N+1$ divisions. If no distinction is made among arithmetic operations, the total number is $Q = 14N - 8$.

We now investigate the applicability of the algorithm (28). We have

Lemma 2. *Suppose the coefficients of the system* (14), (15) *satisfy the conditions*

$$|\alpha_i| > 0, \ |b_i| > 0, \ |c_i| \geq |a_i| + |b_i|, \quad i = 1,2,\ldots,N, \tag{29}$$

and there exists $1 \leq i_0 \leq N$ *such that* $|c_{i_0}| > |a_{i_0}| + |b_{i_0}|$. *Then*

$$\begin{gathered}
c_i - a_i\alpha_i \neq 0, \quad |\alpha_i| \leq 1, \quad |\alpha_i| + |\gamma_i| \leq 1, \quad i = 2,3,\ldots,N, \\
1 - \gamma_{N+1} - \alpha_{N+1}v_1 \neq 0.
\end{gathered}$$

Proof. In fact, since α_i, β_i and γ_i are elimination coefficients for the right-elimination method applied to the problems (20) and (21), and by (29), the conditions of lemma 1 are satisfied, from lemma 1 it follows that

$$\begin{aligned} &c_i - a_i\alpha_i \neq 0, \quad |\alpha_i| \leq 1, \quad i = 2, 3, \ldots, N, \\ &|c_i - a_i\alpha_i| \geq |c_i| - |a_i||\alpha_i| \geq |b_i| > 0. \end{aligned} \tag{30}$$

Further, on the basis of the conditions from lemma 2, $|a_1| + |b_1| \leq |c_1|$ and, consequently $|\alpha_2| + |\gamma_2| \leq 1$. By induction, we obtain from this the inequalities

$$|\alpha_i| + |\gamma_i| \leq 1, \quad i = 2, 3, \ldots, N, \tag{31}$$

since

$$\begin{aligned} |\alpha_{i+1}| + |\gamma_{i+1}| = \frac{|b_i| + |a_i||\gamma_i|}{|c_i - a_i\alpha_i|} &\leq \frac{|a_i| + |b_i| - |a_i|(1 - |\gamma_i|)}{|c_i| - |a_i||\alpha_i|} \\ &\leq \frac{|a_i| + |b_i| - |a_i||\alpha_i|}{|c_i| - |a_i||\alpha_i|} \leq 1 \end{aligned}$$

and thus we have (30). Notice that $|c_i| > |a_i| + |b_i|$ for $i = i_0$ and consequently $|\alpha_{i_0} + 1| + |\gamma_{i_0+1}| < 1$. Since $1 \leq i_0 \leq N$, $|\alpha_{N+1}| + |\gamma_{N+1}| < 1$.

It remains for us to show that $1 - \gamma_{N+1} - \alpha_{N+1}v_1 \neq 0$. On the basis of (28) and (31) we obtain

$$|v_{N-1}| \leq |\alpha_N| + |\gamma_N| \leq 1,$$

and further, by induction, we prove that $|v_i| \leq 1$, $1 \leq i \leq N - 1$, since by (31)

$$|v_i| \leq |\alpha_{i+1}||v_{i+1}| + |\gamma_{i+1}| \leq |\alpha_{i+1}| + |\gamma_{i+1}| \leq 1.$$

In particular, $|v_1| \leq 1$. Hence, taking into account $|\alpha_{N+1}| + |\gamma_{N+1}| < 1$, we deduce that

$$|1 - \gamma_{N+1} - \alpha_{N+1}v_1| \geq 1 - |\gamma_{N+1}| - |\alpha_{N+1}||v_1| \geq 1 - |\alpha_{N+1}| - |\gamma_{N+1}| > 0.$$

Lemma 2 is fully proved. □

In conclusion, notice that the elimination coefficient β_i, and consequently u_i and y_i, depend on the right-hand side f_i. The elimination coefficients α_i and γ_i and also v_i do not depend on f_i, and in a series of problems with difference right-hand sides they would be computed and saved. This allows the second and all subsequent problems to be solved in $Q = 9N-4$ operations.

2.2.3 The elimination method for complicated systems. We continue to construct variants of the elimination method in order to solve systems of difference equations with non-tridiagonal matrices. In Section 2.2.2, cyclic elimination was used to solve a system whose matrix only had two non-zero elements off the main diagonals. We shall now consider a more general case.

Suppose we must solve the following system of equations:

$$\begin{aligned} c_0 y_0 - \sum_{j=1}^{N-1} d_j y_j - \psi_0 y_N &= f_0, \quad i = 0, \\ -\varphi_i y_0 - a_i y_{i-1} + c_i y_i - b_i y_{i-1} - \psi_i y_N &= f_i, \quad 1 \le i \le N-1, \\ -\varphi_N y_0 - \sum_{j=0}^{N-1} g_j y_j + c_N y_N &= f_N, \quad i = N. \end{aligned} \tag{32}$$

A system of the form (32) arises when approximating second-order ordinary differential equations in the case where the associated boundary conditions satisfy auxiliary conditions of integral type, as well as in a series of other cases. In particular, all of the systems of difference equations considered above can be written in such a form. For example, if in (32) we set

$$\begin{aligned} &d_1 = b_0,\ d_{N-1} = a_0,\ d_i = 0, && 2 \le i \le N-2, \\ &\varphi_i = \psi_i = g_i = 0, && 1 \le i \le N-1, \\ &\psi_0 = 0,\ \varphi_N = c_N = 1, && f_N = 0, \end{aligned}$$

then we obtain the problem (17), (18).

If we introduce the vectors $Y = (y_0, y_1, \dots, y_N)^T$ and $F = (f_0, \dots, f_N)^T$, then (32) can be written in the vector form $\mathcal{A}Y = F$, where

$$\mathcal{A} = \left\| \begin{array}{c|ccccccc|c} c_0 & -d_1 & -d_2 & -d_3 & \cdots & -d_{N-3} & d_{N-2} & -d_{N-1} & -\psi_0 \\ \hline -\varphi_1 - a_1 & c_1 & -b_1 & 0 & \cdots & 0 & 0 & 0 & -\psi_1 \\ -\varphi_2 & -a_2 & c_2 & -b_2 & \cdots & 0 & 0 & 0 & -\psi_2 \\ -\varphi_3 & 0 & -a_3 & c_3 & \cdots & 0 & 0 & 0 & -\psi_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -\varphi_{N-3} & 0 & 0 & 0 & \cdots & c_{N-3} & -b_{N-3} & 0 & -\psi_{N-3} \\ -\varphi_{N-2} & 0 & 0 & 0 & \cdots & -a_{N-2} & c_{N-2} & -b_{N-2} & -\psi_{N-2} \\ -\varphi_{N-1} & 0 & 0 & 0 & \cdots & 0 & -a_{N-1} & c_{N-1} & -b_{N-1} - \psi_{N-1} \\ \hline -\varphi_N & -g_1 & -g_2 & -g_3 & \cdots & -g_{N-3} & -g_{N-2} & -g_{N-1} & c_N \end{array} \right\|$$

Clearly, the matrix $\mathcal{A}$ is obtained by bordering a tridiagonal matrix with columns and rows on all four sides. Notice that, by reordering the unknowns as $Y^* = (y_1, y_2, \dots, y_N, y_0)^T$, the system (32) becomes $\mathcal{A}^* Y^* = F^*$, where

the matrix $\mathcal{A}^*$ is obtained by bordering the same tridiagonal matrix with two columns on the right and two rows at the bottom. We now go on to construct a method for solving the problem (32). The solution of (32) will be found as a linear combination of three grid functions u_i, v_i, and w_i:

$$y_i = u_i + y_0 v_i + y_N w_i, \quad 0 \le i \le N, \tag{33}$$

where u_i, v_i, and w_i are solutions of the following three-point grid problems:

$$\left.\begin{aligned} -a_i u_{i-1} + c_i u_i - b_i u_{i+1} &= f_i, \quad 1 \le i \le N-1, \\ u_0 = 0, \quad u_N &= 0; \end{aligned}\right\} \tag{34}$$

$$\left.\begin{aligned} -a_i v_{i-1} + c_i v_i - b_i v_{i+1} &= \varphi_i, \quad 1 \le i \le N-1, \\ v_0 = 1, \quad v_N &= 0; \end{aligned}\right\} \tag{35}$$

$$\left.\begin{aligned} -a_i w_{i-1} + c_i w_i - b_i w_{i+1} &= \psi_i, \quad 1 \le i \le N-1, \\ w_0 = 0, \quad w_N &= 1; \end{aligned}\right\} \tag{36}$$

From (33)–(36) it is clear that, for $1 \le i \le N-1$, the equations of the system (32) are satisfied. The boundary conditions for u_i, v_i, and w_i guarantee that (33) is an identity for $i = 0$ and $i = N$. Thus, if the problems (34)–(36) are solved, and y_0 and y_N are known, the formula (33) will determine the solution to (32). We shall first find y_0 and y_N.

We will find the values of y_0 and y_N using the equations of (32) for $i = 0$ and $i = N$. Substituting y_i from (33) in these equations, we obtain a system of two equations for y_0 and y_N:

$$\left(c_0 - \sum_{j=1}^{N-1} d_j v_j\right) y_0 - \left(\psi_0 + \sum_{j=1}^{N-1} d_j w_j\right) y_N = f_0 + \sum_{j=1}^{N-1} d_j u_j,$$

$$-\left(\varphi_N + \sum_{j=1}^{N-1} g_j v_j\right) y_0 + \left(c_N - \sum_{j=1}^{N-1} g_j w_j\right) y_N = f_N + \sum_{j=1}^{N-1} g_j u_j.$$

If the determinant of this system

$$\begin{aligned} \Delta = &\left(c_0 - \sum_{j=1}^{N-1} d_j v_j\right)\left(c_N - \sum_{j=1}^{N-1} g_j w_j\right) \\ &- \left(\psi_0 + \sum_{j=1}^{N-1} d_j w_j\right)\left(\varphi_N + \sum_{j=1}^{N-1} g_j v_j\right) \end{aligned} \tag{37}$$

is non-zero, then we have that

$$y_0 = \frac{1}{\Delta}\left[\left(c_N - \sum_{j=1}^{N-1} g_j w_j\right)\left(f_0 + \sum_{j=1}^{N-1} d_j u_j\right) + \left(\psi_0 + \sum_{j=1}^{N-1} d_j w_j\right)\left(f_N + \sum_{j=1}^{N-1} g_j u_j\right)\right], \tag{38}$$

$$y_N = \frac{1}{\Delta}\left[\left(\varphi_N + \sum_{j=1}^{N-1} g_j v_j\right)\left(f_0 + \sum_{j=1}^{N-1} d_j u_j\right) + \left(c_0 - \sum_{j=1}^{N-1} d_j v_j\right)\left(f_N + \sum_{j=1}^{N-1} g_j u_j\right)\right], \tag{39}$$

Let us now look at a method for solving the auxiliary problems (34)–(36). Since here we are dealing with ordinary boundary-value problems for three-point equations, it is possible to use the elimination method described in Section 1. For (34)–(36), the formulas for the right-elimination algorithm take the following form:

$$\begin{aligned} u_i &= \alpha_{i+1}u_{i+1} + \beta_{i+1}, \quad i = N-1,\ldots,0, \quad u_N = 0, \\ v_i &= \alpha_{i+1}v_{i+1} + \gamma_{i+1}, \quad i = N-1,\ldots,0, \quad v_N = 0, \\ w_i &= \alpha_{i+1}w_{i+1} + \delta_{i+1}, \quad i = N-1,\ldots,0, \quad w_N = 1, \end{aligned} \tag{40}$$

where the elimination coefficients $\alpha_i, \beta_i, \gamma_i$, and δ_i are determined by the formulas

$$\begin{aligned} \alpha_{i+1} &= \frac{b_i}{c_i - a_i\alpha_i}, & \beta_{i+1} &= \frac{f_i + a_i\beta_i}{c_i - a_i\alpha_i}, \\ i &= 1,2,\ldots,N-1, & \alpha_1 &= 0, \quad \beta_1 = 0, \\ \gamma_{i+1} &= \frac{\varphi_i + a_i\gamma_i}{c_i - a_i\alpha_i}, & \delta_{i+1} &= \frac{\psi_i + a_i\delta_i}{c_i - a_i\alpha_i}, \\ i &= 1,2,\ldots,N-1, & \gamma_1 &= 1, \quad \delta_1 = 0. \end{aligned} \tag{41}$$

Thus, for problem (32), the elimination method is described by (33), (37)–(41).

Let us now consider the question of stability and correctness for the algorithm. By lemma 1, the conditions

$$|a_i| > 0, \quad |b_i| > 0, \quad |c_i| \geq |a_i| + |b_i|, \quad 1 \leq i \leq N-1 \tag{42}$$

are sufficient for the stability and correctness of the elimination method (40)–(41) for the auxiliary problems (34)–(36). It is possible to show that, if the

original system (32) has a unique solution, then the determinant Δ, defined by (37), is non-zero. In this case, the formulas (38) and (39) for computing y_0 and y_N will be correct. We now formulate this result in the form of a lemma.

Lemma 3. *If the system* (32) *has a unique solution and the conditions* (42) *are satisfied, then the algorithm* (33), (37)–(41) *of the elimination method for problem* (32) *is correct and stable.*

Notice that the formulation of simple and, at the same time, not overly restrictive sufficiency conditions for solving the system (32) is a complex problem. Here is an example of conditions which guarantee the correctness and stability of the above algorithm. Suppose that the matrix of the system (32) is diagonally dominant, i.e., that it satisfies the conditions

$$|c_i| \geq |a_i| + |b_i| + |\varphi_i| + |\psi_i|, \quad 1 \leq i \leq N-1, \tag{43}$$

$$|c_0| \geq |\psi_0| + \sum_{j=1}^{N-1} |d_i|, \quad |c_N| \geq |\varphi_N| + \sum_{j=1}^{N-1} |g_j|, \tag{44}$$

$$|a_i| > 0, \quad |b_i| > 0, \quad 1 \leq i \leq N-1, \quad |c_0| > 0, \quad |c_N| > 0,$$

where at least one of the inequalities in (43) or (44) is strict.

We will indicate the basic steps of the proof. It is first shown that $|\alpha_i| + |\gamma_i| + |\delta_i| \leq 1$, $1 \leq i \leq N$. It is further shown that $|v_i| + |w_i| \leq 1$ for $1 \leq i \leq N$, and, if for some i one of the inequalities in (43) is strict, $|v_i| + |w_i| < 1$ for all $1 \leq i \leq N$. We also have

$$\begin{aligned}
\left| c_0 - \sum_{j=1}^{N-1} d_j v_j \right| &\geq |c_0| - \sum_{j=1}^{N-1} |d_j||v_j| \\
&\geq |\psi_0| + \sum_{j=1}^{N-1} (1 - |v_j|)|d_j| \\
&\geq |\psi_0| + \sum_{j=1}^{N-1} |w_j||d_j| \\
&\geq \left| \psi_0 + \sum_{j=1}^{N-1} w_j d_j \right|
\end{aligned}$$

and analogously

$$\left| c_N - \sum_{j=1}^{N-1} g_j w_j \right| \geq \left| \varphi_N + \sum_{j=1}^{N-1} g_j v_j \right|,$$

where at least one of these inequalities is strict. From this it follows that the determinant Δ defined in (37) is non-zero. Stability and correctness of the elimination method for solving the auxiliary problems (34)–(36) then follows from (43).

As a sample problem which leads to (32), we consider the scheme with weights

$$\begin{aligned} y_{t,i} &= \sigma y^{n+1}_{\bar{x}x,i} + (1-\sigma) y^n_{\bar{x}x,i}, \quad 1 \le i \le N-1, \\ y^n_0 - y^n_k &= \mu_1(t_n), \\ y^n_N - y^n_k &= \mu_2(t_n), \\ y^0_i &= u_0(x_i), \quad n = 0,1,\ldots, \quad 1 \le k \le N-1, \end{aligned} \tag{45}$$

which approximates the thermal-flow equation with associated (non-local) boundary conditions

$$\begin{aligned} &\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < l, \quad t > 0, \\ &u(0,t) - u(\nu(t),t) = \mu_1(t), \\ &u(l,t) - u(\nu(t),t) = \mu_2(t), \quad u(x,0) = u_0(x), \end{aligned}$$

where the function $x = \nu(t)$ takes on values between 0 and l. Notice that, in the scheme (45), the curve $x = \nu(t)$ approximates the broken curve $x_k = \nu(t_n)$, so that the points (x_k, t_n) are nodes of the grid.

The difference scheme (45) is written in the form of the system (32) in the variables $y_i = y^{n+1}_i$ with the following values for the coefficients and right-hand side ($\sigma \neq 0$):

$$\begin{aligned} &c_0 = 1, \quad d_k = 1, \quad f_0 = \mu_1(t_{n+1}), \quad \psi_0 = 0, \quad d_j = 0, \quad j \neq k, \\ &c_N = 1, \quad q_k = 1, \quad f_N = \mu_2(t_{n+1}), \quad \varphi_N = 0, \quad g_j = 0, j \neq k, \\ &\varphi_i = \psi_i = 0, \quad a_i = b_i = 1/h^2, \quad c_i = a_i + b_i + 1/(\sigma\tau), \\ &f_i = \frac{1}{\sigma\tau} y^n_i + \left(\frac{1}{\sigma} - 1\right) y^n_{\bar{x}x,i}, \quad i = 1,2,\ldots,N-1. \end{aligned}$$

From this we obtain that the requirement $|2/h^2 + 1/(\sigma\tau)| > 2/h^2$ guarantees that the conditions (43), (44) are satisfied. Consequently, for $\sigma > -h^2/(4\tau)$ it is possible to use the variant of the elimination method described here to solve the equations of the scheme (45), and it will be stable and correct.

2.2.4 The non-monotonic elimination method. We now return to the elimination method constructed in Section 1 for solving three-point equations:

$$\begin{aligned} c_0 y_0 - b_0 y_1 &= f_0, \quad i=0, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} &= f_i, \quad i=1,2,\ldots,N-1, \\ -a_N y_{N-1} + c_N y_N &= f_N, \quad i=N, \end{aligned} \tag{46}$$

Remember that, for the right (left) elimination method, the unknowns y_i are found sequentially in reverse (forward) order. Thus, y_i is expressed only in terms of the later unknowns. This structure in the algorithm is the justification for calling the method *monotone* elimination.

The monotone sequence for determining the unknowns y_i on the reverse path of the method arises from using the natural order to eliminate the unknowns on the forward path. Thus, monotone elimination is Gaussian elimination without pivoting applied to a special system of linear algebraic equations (46) with a tridiagonal matrix. It is known that such a variant of Gaussian elimination is correct for systems of equations with diagonally-dominant matrices. For the system (46), this assertion was proved in lemma 1.

We now look at this in more detail. Remember that, in Section 2.1.1, at the l^{th} stage of the elimination process a "reduced" system

$$\begin{aligned} &(c_l - a_l\alpha_l)y_l - b_l y_{l+1} f_l + a_l\beta_l, \quad i=l, \\ &-a_i y_{i-1} + c_i y_i - b_i y_{i+1} f_i, \quad l+1 \le i \le N-1, \\ &-a_N y_{N-1} + c_N y_N = f_N \end{aligned} \tag{47}$$

was obtained for the unknowns $y_l, y_{l+1}, \ldots, y_N$. Assuming that $c_l - a_l\alpha_l$ was non-zero, we transformed the first equation of the system (47) to the form

$$y_l = \alpha_{l+1} y_{l+1} + \beta_{l+1}, \quad \alpha_{l+1} = b_l/(c_l - a_l\alpha_l) \tag{48}$$

and used it to eliminate y_l from equation (47) for $i = l+1$. Lemma 1 guaranteed that, if the matrix $\mathcal{A}$ of the system (46) were diagonally dominant, then $|c_l - a_l\alpha_l| \ge |b_l|$. Consequently, in the first equation of the system (47), the coefficient of y_l would be larger in modulus than the coefficient of y_{l+1}. Therefore choosing a pivot element is unnecessary, forming (48) is feasible, and the stability condition $|\alpha_{i+1}| \le 1$ is automatically satisfied.

If we do not have diagonal dominance, then it is impossible to guarantee that $c_l - a_l\alpha_l$ is non-zero, or that $|\alpha_{l+1}| \le 1$. In this case, the monotone elimination algorithm can give rise to division by zero or extreme sensitivity to rounding errors; consequently, the algorithm must be modified. The construction of a correct elimination algorithm for the system (46) is based on the use of column pivoting in Gaussian elimination. In this algorithm, the monotone ordering of unknowns y_i can be disturbed, and therefore this method will be called *non-monotonic elimination.*

We now move on to a description of the non-monotonic elimination algorithm. Suppose that, at the l^{th} stage of Gaussian elimination with column pivoting applied to the system (46), the following "reduced" system is obtained:

$$Cy_{m_l} - b_l y_{l+1} = F, \qquad i = l, \tag{49}$$

$$-Ay_{m_l} + c_{l+1}y_{l+1} - b_{l+1}y_{l+2} = \Phi, \qquad i = l+1, \tag{50}$$

$$-a_{l+2}y_{l+1} + c_{l+2}y_{l+2} - b_{l+2}y_{l+3} = f_{l+2}, \qquad i = l+2, \tag{51}$$

$$-a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \qquad l+3 \le i \le N-1, \tag{52}$$

$$-a_N y_{N-1} + c_N y_N = f_N, \qquad i = N, \tag{53}$$

where $m_l \le 1$. (If $l = 0$ in (49)–(53), set $C = c_0$, $A = a_1$, $F = f_0$, $\Phi = f_1$, and $m_0 = 0$).

We now describe the $(l+1)$-st step of the elimination process. The column-pivoting strategy leads us to two cases:

a) If $|C| \ge |b_l|$, then (49) is transformed to the form

$$y_{m_l} - \alpha_{l+1}y_{l+1} = \beta_{l+1}, \quad \alpha_{l+1} = b_l/C, \quad \beta_{l+1} = F/C,$$

where $|\alpha_{l+1}| \le 1$, and the unknown with index m_l is found from the unknown with index $l+1$. Further, this equation is used to eliminate y_{m_l} from (50). This gives:

$$Cy_{m_{l+1}} - b_{l+1}y_{l+2} = F, \quad i = l+1, \tag{54}$$

where $m_{l+1} = l+1$, $C = c_{l+1} - A\alpha_{l+1}$, $F = \Phi + A\beta_{l+1}$. Equation (51) is not changed, since it does not contain y_{m_l}, but is rewritten in the form

$$-Ay_{m_{l+1}} + C_{l+2}y_{l+2} - b_{l+2}y_{l+3} = \Phi, \quad i = l+2, \tag{55}$$

where $A = a_{l+2}$, $\Phi = f_{l+2}$. Combining (54) and (55) with (52), (53), we obtain a new "reduced" system of the form (49)–(53), in which l has been replaced by $l+1$. The $(l+1)^{\text{th}}$ step is completed.

b) If $|C| < |b_l|$, then (49) is transformed to the form

$$y_{l+1} - \alpha_{l+1}y_{m_l} = \beta_{l+1}, \quad \alpha_{l+1} = C/b_l, \quad \beta_{l+1} = -F/b_l,$$

where again $|\alpha_{l+1}| \le 1$, but this time the unknown with index $l+1$ is computed from the unknown with index m_l. The resulting equation is used to eliminate y_{l+1} from (50) and (51). Here (50) will be transformed to the form (54), where $m_{l+1} = m_l$, $C = c_{l+1}\alpha_{l+1} - A$, $F = \Phi - c_{l+1}\beta_{l+1}$, and (51) is transformed to the form (55), where the quantities A and Φ

are redefined using $A = a_{l+2}\alpha_{l+2}$, $\Phi = f_{l+2} + a_{l+2}\beta_{l+1}$. Equations (52), (53) are not changed, since they do not contain y_{l+1}. Again we obtain a system of the form (49)–(53). It has different coefficients C and A than the system obtained in case a), and the right-hand sides F and Φ are computed by different formulas.

Thus, one step of the process differs in the choice of a pivot element. Notice that if the original system is not singular then, in equation (49), the coefficients C and b_l cannot both be zero. This guarantees the correctness of the formulas for the coefficients α_{l+1} and β_{l+1}. Since all the computed α_{l+1} are less than one in modulus, the computation of the unknowns y_i on the reverse path of the method will be stable with respect to the rounding error.

For this algorithm, the unknowns may be computed out of sequence. This requires us to store information about the ordering of the unknowns. This information can be stored in two integer sets θ and κ: $\theta = \{\theta_i, 1 \le i \le N\}$, $\kappa = \{\kappa_i, 1 \le i \le N\}$, and the unknowns are found from the formulas $y_m = \alpha_{i+1}y_n + \beta_{i+1}$, $m = \theta_{i+1}$, $n = \kappa_{i+1}$, $i = N-1, N-2, \ldots, 0$. The sets θ and κ are constructed on the forward path of the algorithm.

The full algorithm for the non-monotonic elimination method can be described as follows.

[1] Give initial values for C, A, F, and Φ: $C = c_0$, $A = a_1$, $F = f_0$, $\Phi = f_1$, and formally set $\kappa_0 = 0$.

[2] Sequentially for $i = 0, 1, \ldots, N-1$, depending on the situation, perform the operations described in paragraphs a) or b):

a. if $|C| \ge |b_i|$, then

$$\alpha_{i+1} = b_i/C, \quad \beta_{i+1} = F/C, \quad C = c_{i+1} - A\alpha_{i+1}, \quad F = \Phi + A\beta_{i+1},$$
$$\theta_{i+1} = \kappa_i, \quad \kappa_{i+1} = i+1, \quad A = a_{i+2}, \quad \Phi = f_{i+2};$$

b. if $|C| < |b_i|$, then

$$\alpha_{i+1} = C/b_i, \quad \beta_{i+1} = -F/b_i, \quad C = c_{i+1}\alpha_{i+1} - A,$$
$$F = \Phi - c_{i+1}\beta_{i+1}, \quad \theta_{i+1} = i+1, \quad \kappa_{i+1} = \kappa_i,$$
$$A = a_{i+2}\alpha_{i+1}, \quad \Phi = f_{i+2} + a_{i+2}\beta_{i+1}.$$

Remark. For $i = N-1$ it is not necessary to redefine A and Φ in steps (a) and (b).

[3] First compute the unknown y_n, where $n = k_n$ using the formula $y_n = F/C$, and then sequentially for $i = N-1, N-2, \ldots, 0$ compute the remaining unknowns $y_m = \alpha_{i+1}y_n + \beta_{i+1}$, $m = \theta_{i+1}$, $n = \kappa_{i+1}$.

Notice that the algorithm presented here reduces to the usual right-elimination algorithm if the conditions of lemma 1 are satisfied.

An elementary count of the number of arithmetic operations for the non-monotonic elimination algorithm shows that, if step [b] is always used, $Q = 12N$ operations are required. This is 1.5 times more than for monotonic elimination.

Let us now consider a sample application of the non-monotonic elimination method. Suppose we must solve the following difference problem

$$\begin{aligned} -y_{i-1} + y_i - y_{i+1} &= 0, \quad 1 \le i \le N-1, \\ y_0 = 1, \quad y_N &= 0. \end{aligned} \tag{56}$$

The problem (56) is a particular case of the system (46), in which $f_0 = 1$, $b_0 = a_N = 0$, $c_0 = c_N = 1$, $f_N = 0$, $c_i = a_i = b_i = 1$, $f_i = 0$, $1 \le i \le N-1$. If N is not divisible by 3, then the solution of problem (56) exists and has the form (see Section 1.4.1)

$$y_i = \sin\frac{(N-i)\pi}{3} \Big/ \sin\frac{N\pi}{3}, \quad 0 \le i \le N. \tag{57}$$

The left and right elimination algorithms are not correct for (56), since the computation of α_3 (for right elimination) and ξ_{N-2} (for left elimination) leads to division by the zero values $c_2 - a_2\alpha_2$ and $c_{N-2} - b_{N-2}\xi_{N-1}$. The non-monotonic elimination algorithm allows us to accurately obtain the solution (57). To illustrate, we list the values of the coefficients α_i, β_i, and also θ_i and κ_i for $N = 11$ (Table 1).

Table 1

i	0	1	2	3	4	5	6	7	8	9	10	11
α_i		0	1	0	−1	1	0	−1	1	0	−1	1
β_i		1	1	−1	−1	−1	1	1	1	−1	−1	−1
θ_i		0	1	3	2	4	6	5	7	9	8	10
κ_i		1	2	2	4	5	5	7	8	8	10	11
y_i	1	1	0	−1	−1	0	1	1	0	−1	−1	0

2.3 The elimination method for five-point equations

2.3.1 The monotone elimination algorithm. Above we looked at several variants of the elimination method and used them to solve three-point difference equations. As was noted earlier, such difference equations arise when approximating second-order ordinary differential equations.

There are two ways of solving boundary-value problems for higher-order equations. The first possibility is to transform the problem to a system of first-order differential equations and then construct the corresponding difference scheme. In this case we obtain a boundary-value problem for a two-point vector equation. We looked at methods for solving such difference problems in Section 4.

The second possibility is to directly approximate the original differential problem. In this case we obtain multi-point difference equations. Most commonly, we encounter five-point equations of the following form:

$$c_0y_0 - d_0y_1 + e_oy_2 = f_0, \qquad i = 0, \tag{1}$$

$$-b_1y_0 + c_1y_1 - d_1y_2 + e_1y_3 = f_1, \qquad i = 1, \tag{2}$$

$$a_iy_{i-2} - b_iy_{i-1} + c_iy_i - d_iy_{i+1} + e_iy_{i+2} = f_i, \qquad 2 \le i \le N-2, \tag{3}$$

$$a_{N-1}y_{N-3} - b_{N-1}y_{N-2} + c_{N-1}y_{N-1} - d_{N-1}y_N = f_{N-1}, \qquad i = N-1, \tag{4}$$

$$a_Ny_{N-2} - b_Ny_{N-1} + c_Ny_N = f_N, \qquad i = N. \tag{5}$$

Such systems arise when approximating boundary-value problems for fourth-order ordinary differential equations, and also when realizing difference schemes for equations with partial derivatives. The matrix $\mathcal{A}$ of the system (1)–(5) is a square pentadiagonal matrix of size $(N+1) \times (N+1)$ and has no more than $5N-1$ non-zero elements.

To solve the system (1)–(5), we use the method of Gaussian elimination. Taking into account the structure of the system (1)–(5), it is easily seen that the reverse path of Gauss' method must have the form

$$y_i = \alpha_{i+1}y_{i+1} - \beta_{i+1}y_{i+2} + \gamma_{i+1}, \quad 0 \le i \le N-2, \tag{6}$$

$$y_{N-1} = \alpha_Ny_N + \gamma_N, \qquad i = N-1. \tag{7}$$

To realize (6), (7) it is necessary to give y_N, and also to determine the coefficients α_i, β_i, γ_i.

We will first derive the formulas for α_i, β_i, and γ_i. Using (6), we express y_{i-1} and y_{i-2} in terms of y_i and y_{i+1}. We obtain

$$y_{i-1} = \alpha_iy_i - \beta_iy_{i+1} + \gamma_i, \quad 1 \le i \le N-1, \tag{8}$$

$$y_{i-2} = (\alpha_i\alpha_{i-1} - \beta_{i-1})y_i - \beta_i\alpha_{i-1}y_{i+1} + \alpha_{i-1}\gamma_i + \gamma_{i-1} \tag{9}$$

for $2 \le i \le N-1$.

Substituting (8) and (9) in (3), we obtain

$$[c_i - a_i\beta_{i-1} + \alpha_i(a_i\alpha_{i-1} - b_i)]y_i = [d_i + \beta_i(a_i\alpha_{i-1} - b_i)]y_{i+1} - e_iy_{i+2} + [f_i - a_i\gamma_{i-1} - \gamma_i(a_i\alpha_{i-1} - b_i)], \quad 2 \le i \le N-2.$$

Comparing this expression with (6), we set that if

$$\begin{aligned} \alpha_{i+1} &= \frac{1}{\Delta_i}[d_i + \beta_i(a_i\alpha_{i-1} - b_i)], \quad \beta_{i+1} = \frac{e_i}{\Delta_i}, \\ \gamma_{i+1} &= \frac{1}{\Delta_i}[f_i - a_i\gamma_{i-1} - \gamma_i(a_i\alpha_{i-1} - b_i)], \end{aligned} \tag{10}$$

where $\Delta_i = c_i - a_i\beta_{i-1} + \alpha_i(a_i\alpha_{i-1} - b_i)$, then the equations (1)–(5) will be satisfied for $2 \le i \le N-2$.

The recurrence relation (10) connects α_{i+1}, β_{i+1}, and γ_{i+1} with α_i, α_{i-1}, β_i, β_{i-1}, γ_i, and γ_{i-1}. Therefore, if α_i, β_i, and γ_i are given for $i = 1, 2$, then (10) can be used to sequentially find α_i, β_i, and γ_i for $3 \le i \le N-1$.

We now find α_i, β_i, and γ_i for $i = 1, 2$. From (1) and (6) for $i = 0$ we obtain directly

$$\alpha_1 = d_0/c_0, \quad \beta_1 = e_0/c_0, \quad \gamma_1 = f_0/c_0. \tag{11}$$

Further, substituting (8) in (2) with $i = 1$, we obtain

$$(c_1 - b_1\alpha_1)y_1 = (d_1 - b_1\beta_1)y_2 - e_1y_3 + f_1 + b_1\gamma_1.$$

Consequently, (2) will be satisfied if we set

$$\alpha_2 = \frac{d_1 - b_1\beta_1}{c_1 - b_1\alpha_1}, \quad \beta_2 = \frac{e_1}{c_1 - b_1\alpha_1}, \quad \gamma_2 = \frac{f_1 + b_1\gamma_1}{c_1 - b_1\alpha_1}. \tag{12}$$

Thus, using (10)–(12), it is possible to find α_i, β_i, and γ_i for $1 \le i \le N-1$. It remains to determine α_N, γ_N and y_N in (7).

To do this, we will use equations (4) and (5). Substituting (8) and (9) in (4) with $i = N-1$ and comparing the resulting expression with (7), we find that α_N and γ_N are determined by (10) for $i = N-1$. We now find y_N. For this we substitute (6) for $i = N-2$ and (7) in (5). We obtain

$$[c_N - a_N\beta_{N-1} + \alpha_N(a_N\alpha_{N-1} - b_N)]y_N = f_N - a_N\gamma_{N-1} - \gamma_N(a_N\alpha_{N-1} - b_N)$$

or

$$y_N = \gamma_{N+1}$$

where γ_{N+1} is defined by (10) for $i = N$.

Gathering together the formulas derived above, we write the right-elimination algorithm for the system (1)–(5) in the following form:

1) using the formulas

$$\alpha_{i+1} = \frac{1}{\Delta_i}[d_i + \beta_i(a_i\alpha_{i-1} - b_i)], \quad i = 2,3,\ldots,N-1, \tag{13}$$

$$\alpha_1 = \frac{d_0}{c_0}, \quad \alpha_2 = \frac{1}{\Delta_1}(d_1 - \beta_1 b_1),$$

$$\gamma_{i+1} = \frac{1}{\Delta_i}[f_i - a_i\gamma_{i-1} - \gamma_i(a_i\alpha_{i-1} - b_i)], \quad i = 2,3,\ldots,N, \tag{14}$$

$$\gamma_1 = \frac{f_0}{c_0}, \quad \gamma_2 = \frac{1}{\Delta_1}(f_1 + b_1\gamma_1),$$

$$\beta_{i+1} = e_i/\Delta_i, \quad i = 1,2,\ldots,N-2, \quad \beta_1 = e_0/c_0, \tag{15}$$

where

$$\Delta_i = c_i - a_i\beta_{i+1} + \alpha_i(a_i\alpha_{i-1} - b_i), \quad 2 \le i \le N, \quad \Delta_1 = c_1 - b_1\alpha_1, \tag{16}$$

find the elimination coefficients α_i, β_i, and γ_i;

2) the unknowns y_i are found sequentially from the formulas

$$\begin{aligned} y_i &= \alpha_{i+1}y_{i+1} - \beta_{i+1}y_{i+2} + \gamma_{i+1}, \quad i = N-2, N-3, \ldots, 0. \\ y_{N-1} &= \alpha_N y_N + \gamma_N, \quad y_N = \gamma_{N+1}. \end{aligned} \tag{17}$$

This algorithm will be called the *monotone elimination* algorithm.

Remark. It is not difficult to construct the left and two-sided elimination algorithms for the system (1)–(5).

We now compute the number of arithmetic operations used by the algorithm (13)–(17). To realize (13)–(17) we require: $8N - 5$ additions and subtractions, $8N - 5$ multiplications, and $3N$ divisions. If no distinctions is made between operations, the total number of operations for the algorithm is $Q = 19N - 10$.

2.3.2 Justification of the method. The elimination algorithm (13)–(17) constructed above will be called *correct* if, for any $2 \le i \le N$

$$\Delta_i = c_i - a_i\beta_{i-1} + \alpha_i(a_i\alpha_{i-1} - b_i) \neq 0, \quad \Delta_1 = c_1 - \alpha_1 b_1 \neq 0.$$

The following lemma gives sufficient conditions for the correctness of the algorithm (13)–(17).

Lemma 4. *Suppose the coefficients of the system* (1)–(5) *satisfy the conditions*

$$|a_i| > 0,\ 2 \le i \le N, \qquad |b_i| > 0,\quad 1 \le i \le N,$$
$$|d_i| > 0,\ 0 \le i \le N-1, \quad |e_i| > 0,\quad 0 \le i \le N-2,$$

and the conditions

$$\begin{aligned} &|c_0| \ge |d_0| + |e_0|, && |c_1| \ge |b_1| + |d_1| + |e_1|, \\ &|c_N| \ge |a_N| + |b_N|, && |c_{N-1}| \ge |a_{N-1}| + |b_{N-1}| + |d_{N-1}|, \\ &|c_i| \ge |a_i| + |b_i| + |d_i| + |e_i|, && 2 \le i \le N-2, \end{aligned} \tag{18}$$

where at least one of the inequalities (18) *is strict. Then the algorithm* (13)–(17) *is correct and, in addition,*

$$|\alpha_i| + |\beta_i| \le 1, \quad 1 \le i \le N-1, \quad |\alpha_N| \le 1.$$

Proof. Using the conditions of the lemma, from (13) and (15) we obtain

$$|\alpha_1| + |\beta_1| = \frac{|d_0| + |e_0|}{|c_0|} \le 1.$$

Further, using the inequality $1 - |\alpha_1| \ge |\beta_1|$, we find that

$$\begin{aligned} |c_1 - b_1\alpha_1| &\ge |c_1| - |b_1||\alpha_1| \ge |b_1|(1 - |\alpha_1|) + |d_1| + |e_1| \\ &\ge |b_1||\beta_1| + |d_1| + |e_1| \ge |d_1 - b_1\beta_1| + |e_1| > 0. \end{aligned}$$

From this and (13)–(15) it follows that

$$|\alpha_2| + |\beta_2| = \frac{|d_1 - \beta_1 b_1| + |d_1|}{|c_1 - b_1\alpha_1|} \le 1.$$

The rest of the proof proceeds by induction. Suppose that

$$|\alpha_{i-1}| + |\beta_{i-1}| \le 1, \quad |\alpha_i| + |\beta_i| \le 1.$$

We will show that this implies that

$$\Delta_i = c_i - a_i\beta_{i-1} + \alpha_i(a_i\alpha_{i-1} - b_i) \ne 0, \quad |\alpha_{i+1}| + |\beta_{i+1}| \le 1.$$

In fact, from (18) and (19) we obtain

$$\begin{aligned}
|\Delta_i| &\geq |c_i| - |a_i||\beta_{i-1}| - |\alpha_i||\alpha_{i-1}||a_i| - |\alpha_i||b_i| \\
&\geq |a_i|(1 - |\beta_{i-1}|) + |b_i|(1 - |\alpha_i|) - |\alpha_i||\alpha_{i-1}||a_i| + |d_i| + |e_i| \\
&\geq |a_i||\alpha_{i-1}| + |b_i||\beta_i| - |\alpha_i||\alpha_{i-1}||a_i| + |d_i| + |e_i| \\
&\geq |a_i||\alpha_{i-1}|(1 - |\alpha_i|) + |d_i - b_i\beta_i| + |e_i| \\
&\geq |a_i||\alpha_{i-1}||\beta_i| + |d_i - b_i\beta_i| + |e_i| \\
&\geq |d_i + \beta_i(a_i\alpha_{i-1} - b_i)| + |e_i| > 0, \quad i \leq N-2.
\end{aligned} \tag{20}$$

From this and (13), (15) we find

$$|\alpha_{i+1}| + |\beta_{i+1}| = \frac{|d_i + \beta_i(a_i\alpha_{i-1} - b_i)| + |e_i|}{|\Delta_i|} \leq 1, \quad i \leq N-2.$$

Further, for $i = N-1$ we have in place of (20) the estimate

$$|\Delta_{N-1}| \geq |a_{N-1}||\alpha_{N-2}||\beta_{N-1}| + |b_{N-1}||\beta_{N-1}| + |d_{N-1}| > 0.$$

In addition, from this we get

$$|\Delta_{N-1}| \geq |d_{N-1} + \beta_{N-1}(a_{N-1}\alpha_{N-2} - b_{N-1})|,$$

and consequently,

$$|\alpha_N| = \frac{1}{|\Delta_{N-1}|}|d_{N-1} + \beta_{N-1}(a_{N-1}\alpha_{N-2} - b_{N-1})| \leq 1.$$

It remains to prove that $\Delta_N \neq 0$. We have

$$\begin{aligned}
|\Delta_N| &\geq |c_N| - |a_N||\beta_{N-1}| - |\alpha_N||\alpha_{N-1}||a_N| - |\alpha_N||b_N| \\
&= |c_N| - |a_N| - |b_N| + |a_N|(1 - |\beta_{N-1}|) + |b_N|(1 - |\alpha_N|) - |\alpha_N||\alpha_{N-1}||a_N| \\
&\geq |c_N| - |a_N| - |b_N| + (1 - |\alpha_N|)(1 - |\beta_{N-1}|)|a_N| + |b_N|(1 - |\alpha_N|).
\end{aligned}$$

By the assumptions of the lemma, it is easy to show that at least one of the inequalities $|c_N| \geq |a_N| + |b_N|$, $|\alpha_N| \leq 1$ is strict. Hence it follows that $\Delta_N \neq 0$. The lemma is proved. □

Remark. From the estimates $|\alpha_i| + |\beta_i| \leq 1$ indicated in lemma 4 it follows that formula (17) will not cause growth in the error of the computed y_N.

2.3.3 A variant of non-monotonic elimination. We introduce here the algorithm for the elimination method which would be obtained if the system (1)–(5) were solved using Gaussian elimination with column pivoting. Such an algorithm will be correct if the matrix $\mathcal{A}$ of the system (1)–(5) is non-singular. Since the method of constructing the algorithm is analogous to the development in Section 2.4, we will limit ourselves here to the final form of the algorithm.

1) Initially set: $C = c_0$, $D = d_0$, $B = b_1$, $Q = c_1$, $S = a_2$, $T = b_2$, $R = 0$, $A = a_3$, $F = f_0$, $\Phi = f_1$, $G = f_2$, $H = f_3$, and set $\kappa_0 = 0$, $\eta_0 = 1$.

2) Sequentially for $i = 0, 1, \ldots, N-2$, depending on the situation, perform the operations described in steps (a), (b), or (c):

[a] if $|C| \geq |D|$ and $|C| \geq |e_i|$, then

$$\begin{array}{lll} \alpha_{i+1} = D/C & \beta_{i+1} = e_i/c, & \gamma_{i+1} = F/C, \\ C = Q - B\alpha_{i+1}, & D = d_{i+1} - B\beta_{i+1}, & F = \Phi + B\gamma_{i+1} \\ B = T - S\alpha_{i+1}, & Q = c_{i+2} - S\beta_{i+1}, & \Phi = G - S\gamma_{i+1} \\ S = A - R\alpha_{i+1}, & T = b_{i+3} - R\beta_{i+1}, & G = H + R\gamma_{i+1}, \end{array} \tag{21}$$

$$\left.\begin{array}{lll} R = 0, & A = a_{i+1}, & H = f_{i+4} \\ \theta_{i+1} = \kappa_i, & \kappa_{i+1} = \eta_i, & \eta_{i+1} = i+2; \end{array}\right\} \tag{22}$$

[b] if $|D| > |C|$ and $|D| \geq |e_i|$, then

$$\begin{array}{lll} \alpha_{i+1} = C/D, & \beta_{i+1} = -e_i/D, & \gamma_{i+1} = -F/D, \\ C = Q\alpha_{i+1} - B, & D = Q\beta_{i+1} + d_{i+1}, & F = \Phi - Q\gamma_{i+1}, \\ B = T\alpha_{i+1} - S, & Q = T\beta_{i+1} + c_{i+2}, & \Phi = T\gamma_{i+1} + G, \\ S = A\alpha_{i+1} - R, & T = A\beta_{i+1} + b_{i+3}, & G = H - A\gamma_{i+1}, \end{array} \tag{23}$$

$$\left.\begin{array}{lll} R = 0, & A = a_{i+4}, & H = f_{i+4} \\ \theta_{i+1} = \eta_i, & \kappa_{i+1} = \kappa_i, & \eta_{i+1} = i+2; \end{array}\right\} \tag{24}$$

[c] if $|e_i| > C$ and $|e_i| > D$, then

$$\begin{array}{lll} \alpha_{i+1} = D/e_i, & \beta_{i+1} = C/e_i, & \gamma_{i+1} = F/e_i, \\ C = Q - d_{i+1}\alpha_{i+1}, & D = B - d_{i+1}\beta_{i+1}, & F = \Phi + d_{i+1}\gamma_{i+1}, \\ B = T - c_{i+2}\alpha_{i+1}, & Q = S - c_{i+2}\beta_{i+1}, & \Phi = G - c_{i+2}\gamma_{i+1}, \\ S = A - b_{i+3}\alpha_{i+1}, & T = R - b_{i+3}\beta_{i+1}, & G = H + b_{i+3}\gamma_{i+1}, \end{array} \tag{25}$$

$$\left.\begin{array}{lll} R = -a_{i+4}\alpha_{i+1}, & A = -a_{i+4}\beta_{i+1}, & H = f_{i+4} - a_{i+4}\gamma_{i+1}, \\ \theta_{i+1} = i+2, & \kappa_{i+1} = \eta_i, & \eta_{i+1} = \kappa_i. \end{array}\right\} \tag{26}$$

Remark. For $i \geq N-3$ it is not necessary to carry out (22), (24), or (26), and for $i = N-2$, (21), (23), and (25) can be omitted.

3) If $|C| \geq |D|$, then

$$\alpha_N = D/C, \quad \gamma_N = F/C, \quad \gamma_{N+1} = (\Phi + B\gamma_N)/(Q - B\alpha_N),$$
$$\theta_N = \kappa_{N-1}, \quad \kappa_N = \eta_{N-1}.$$

If $|D| > |C|$, then

$$\alpha_N = C/D, \quad \gamma_N = -F/D, \quad \gamma_{N+1} = (\Phi - Q\gamma_N)/(Q\alpha_N - B),$$
$$\theta_N = \eta_{N-1}, \quad \kappa_N = \kappa_{N-1}.$$

4) Compute the unknowns

$$y_n = \gamma_{N+1}, \quad y_m = \alpha_N y_n + \gamma_N, \quad m = \theta_N, \quad n = \kappa_N,$$

and then sequentially for $i = N-2, N-3, \ldots, 0$ determine the remaining unknowns

$$y_m = \alpha_{i+1} y_n - \beta_{i+1} y_\kappa + \gamma_{i+1}, \quad m = \theta_{i+1}, \quad n = \kappa_{i+1}, \quad \kappa = \eta_{i+1}.$$

Let us now consider an application of the non-monotonic elimination method. In Section 1.3.3, we solved the following boundary-value problem:

$$\begin{aligned}
y_0 - y_1 + 2y_2 &= 0, && i = 0,\\
-y_0 + y_1 - y_2 + y_3 &= 0, && i = 1,\\
y_{i-2} - y_{i-1} + 2y_i - y_{i+1} + y_{i+2} &= 0, && 2 \leq i \leq N-2,\\
y_{N-3} - y_{N-2} + y_{N-1} - y_N &= 0, && i = N-1,\\
2y_{N-2} - y_{N-1} + y_N &= 0, && i = N.
\end{aligned} \tag{27}$$

If N is even and not divisible by 3, then the system (27) has the unique solution

$$y_i = -\cos\frac{i\pi}{2} - \sin\frac{i\pi}{2}, \quad 0 \leq i \leq N. \tag{28}$$

It is not difficult to verify that the monotone elimination algorithm is not correct for (27) since the computation of the elimination coefficients α_2, β_2, and γ_2 leads to division by zero. The non-monotonic elimination algorithm allows us to accurately obtain the solution (28). To illustrate this algorithm, we include here (Table 2), which gives the values of the elimination coefficients α_i, β_i, and γ_i, and also θ_i, κ_i, and η_i for $N = 10$.

Table 2

i	0	1	2	3	4	5	6	7	8	9	10	11
α_i		1/2	1/2	−1/2	0	0	0	−1/3	−1/3	0	1	
β_i		1/2	1/2	−1/2	1	−1	1	−2/3	−2/3	1		
γ_i		1	1	−1	−2	−2	2	−4/3	−2/3	0	−2	1
θ_i		2	3	4	0	5	6	7	9	8	1	
κ_i		1	0	1	1	1	1	1	8	1	10	
η_i		0	1	0	5	6	7	8	1	10		
y_i	−1	−1	1	1	−1	−1	1	1	−1	−1	1	

From the table it is clear that the unknowns y_i are determined in the following order: $y_{10}, y_1, y_8, y_9, y_7, y_6, y_5, y_0, y_4, y_3, y_2$, i.e., in non-monotonic order.

2.4 The block-elimination method

2.4.1 Systems of vector equations. It was remarked above that one method for solving boundary-value problems for high-order ordinary differential equations is to transform the problem to a system of first-order equations and then to approximate this system by a difference scheme. As a result, we obtain a *two-point vector system* of equations with first-order boundary conditions. In the general case, this can be written in the following form:

$$\begin{aligned} P_{i+1}V_{i+1} - Q_iV_i = F_{i+1}, \quad & 0 \le i \le N-1, \\ P_0V_0 = F_0, \quad & Q_NV_N = F_{N+1}, \end{aligned} \tag{1}$$

where V_i is a vector of unknowns of dimension M, P_{i+1} and Q_i, for $0 \le i \le N-1$ are square $M \times M$ matrices, P_0 and Q_N are rectangular matrices of size $M_1 \times M$ and $M_2 \times M$, respectively. The vector F_{i+1} is of dimension M for $0 \le i \le N-1$, and F_0 and F_{N+1} are of dimension M_1 and M_2, respectively.

Notice that one way to solve the indicated differential equations is to directly approximate these equations by difference schemes. This way we obtain a system of multi-point scalar equations. Methods for solving three- and

five-point scalar equations were studied in Sections 2.1–2.3. If we approximate a high-order system of ordinary differential equations, then a system of multi-point vector equations arises. However, as for scalar equations, vector systems of multi-point equations can also lead to systems of the form (1). Any method for solving (1) will correspond to some method for solving the original multi-point system. To clarify the idea of the transformation consider as an example the system of five-point equations examined in Section 2.3 (see (2.3.1)–(2.3.5)). If we denote

$$\begin{aligned} Y_i &= (y_{i+1}, y_i, y_{i-1}, y_{i-2})^T, && 2 \le i \le N-1, \\ F_{i+1} &= (f_i, 0, 0, 0)^T, && 2 \le i \le N-2, \\ F_2 &= (f_0, f_1)^T, \quad F_N = (f_{N-1}, f_N)^T, \end{aligned}$$

then, using the relations connecting Y_{i+1} and Y_i, the system from Section 2.3 can be written in the form

$$\begin{aligned} P_{i+1}Y_{i+1} - Q_iY_i &= F_{i+1}, \quad 2 \le i \le N-2, \\ P_2Y_2 = F_2, \quad Q_{N-1}Y_{N-1} &= F_N, \end{aligned} \tag{2}$$

where

$$P_{i+1} = \begin{Vmatrix} e_i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Vmatrix}, \quad Q_i = \begin{Vmatrix} d_i & -c_i & b_i & -a_i \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{Vmatrix}, \quad 2 \le i \le N-2,$$

$$P_2 = \begin{Vmatrix} 0 & e_0 & -d_0 & c_0 \\ e_1 & -d_1 & c_1 & -b_1 \end{Vmatrix}, \quad Q_{N-1} = \begin{Vmatrix} -d_{N-1} & c_{N-1} & -b_{N-1} & a_{N-1} \\ c_N & -b_N & a_N & 0 \end{Vmatrix}.$$

In this case, $M_1 = M_2 = 2$, $M = 4$.

Ignoring the fact that multi-point vector equations can be reduced to the form (1), and limiting ourselves by constructing a method which only solves the system (1), we will consider separately the class of *three-point vector equations*. In addition, in Section 2.5.3 we will transform (1) to a system of three-point vector equations and obtain a method for solving (1) as a variant of a method for solving three-point equations.

Before describing three-point equations in general form, we will look at an example. We will show how a difference problem for the simplest elliptic equation leads to a system of three-point equation of special form.

Suppose the rectangular grid $\bar{\omega} = \{x_{ij} = (ih_1, jh_2) \in \bar{G},\ 0 \le i \le M,\ 0 \le j \le N,\ l_1 = Mh_1,\ l_2 = Nh_2\}$ with boundary γ, is introduced into the

rectangle $\bar{G} = \{0 \le x_\alpha \le l_\alpha,\ \alpha = 1,2,\}$, and that we must solve a *Dirichlet difference problem for Poisson's equation*

$$\begin{aligned} y_{\bar{x}_1 x_1} + y_{\bar{x}_2 x_2} &= -\varphi(x), \quad x \in \omega, \\ y(x) &= g(x), \qquad x \in \gamma, \end{aligned} \tag{3}$$

where

$$y_{\bar{x}_1 x_1} = \frac{1}{h_1^2}[y(i+1,j) - 2y(i,j) + y(i-1,j)]$$
$$y_{\bar{x}_2 x_2} = \frac{1}{h_2^2}[y(i,j+1) - 2y(i,j) + y(i,j-1)], \quad y(i,j) = y(x_{ij}).$$

We shall transform the scheme (3). For this we multiply (3) by $(-h_2^2)$ and write out the difference derivative $y_{\bar{x}_2 x_2}$ at a point. For $1 \le j \le N-1$ we obtain:
for $2 \le i \le M-2$

$$-y(i,j-1) + [2y(i,j) - h_2^2 y_{\bar{x}_1 x_1}(i,j)] - y(i,j+1) = h_2^2 \varphi(i,j);$$

for $i = 1$

$$-y(i,j-1) + \left[2y(i,j) - \frac{h_2^2}{h_1^2}(y(i+1,j) - 2y(i,j))\right] - y(i,j+1) = h_2^2 \bar{\varphi}(i,j);$$

for $i = M-1$

$$-y(i,j-1) + \left[2y(i,j) - \frac{h_2^2}{h_1^2}(y(i-1,j) - 2y(i,j))\right] - y(i,j+1) = h_2^2 \bar{\varphi}(i,j);$$

where

$$\bar{\varphi}(1,j) = \varphi(1,j) + \frac{1}{h_1^2} g(0,j),$$
$$\bar{\varphi}(M-1,j) = \varphi(M-1,j) + \frac{1}{h_1^2} g(M,j).$$

Besides, for $j = 0, N$ we have

$$y(i,0) = g(i,0), \quad y(i,N) = g(i,N), \quad 1 \le i \le M-1.$$

We now denote by Y_j the vector of dimension $M-1$, whose components are the values of the grid function $y(i,j)$ at the interior nodes in the j^{th} row of the grid $\bar{\omega}$:

$$Y_j = (y(1,j), y(2,j), \ldots, y(M-i,j))^T, \quad 0 \le j \le N,$$

and by F_j the vector of dimension $M-1$

$$F_j = (h_2^2\bar{\varphi}(1,j), h_2^2\varphi(2,j), \ldots, h_2^2\varphi(M-2,j), h_2^2\bar{\varphi}(M-1,j))^T,$$
$$1 \le j \le N-1,$$
$$F_j = (g(1,j), g(2,j), \ldots, g(M-1,j))^T, \quad j = 0, N.$$

We also define the square $(M-1)\times(M-1)$ matrix C in the following fashion:

$$CV = (\Lambda v(1), \Lambda v(2), \ldots, \Lambda v(M-1))^T,$$
$$V = (v(1), v(2), \ldots, v(M-1))^T,$$

where the difference operator Λ is

$$\Lambda v(i) = 2v(i) - h_2^2 v_{\bar{x}_1 x_1}(i), \quad 1 \le i \le M-1,$$
$$v(0) = v(M) = 0.$$

It is easy to see that C is a tridiagonal matrix of the form

$$C = \begin{Vmatrix} 2(1+\alpha) & -\alpha & 0 & \ldots & 0 & 0 & 0 \\ -\alpha & 2(1+\alpha) & -\alpha & \ldots & 0 & 0 & 0 \\ 0 & -\alpha & 2(1+\alpha) & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 2(1+\alpha) & -\alpha & 0 \\ 0 & 0 & 0 & \ldots & -\alpha & 2(1+\alpha) & -\alpha \\ 0 & 0 & 0 & \ldots & 0 & -\alpha & 2(1+\alpha) \end{Vmatrix}, \tag{4}$$

where $\alpha = h_2^2/h_1^2$, and that C is diagonally dominant, since $|1+\alpha| > |\alpha|$, $\alpha > 0$, and hence non-singular.

Using the above notation, it is possible to write the above equations in the form of a system of three-point vector equations:

$$-Y_{j-1} + CY_j - Y_{j+1} = F_j, \quad 1 \le j \le N-1,$$
$$Y_0 = F_0, \quad Y_N = F_N. \tag{5}$$

This is the desired three-point system of special form with constant coefficients.

The problem (5) is a special case of the following general problem: find the vector $Y_j (0 \le j \le N)$ which satisfies the following system:

$$\begin{aligned} C_0 Y_0 - B_0 Y_1 &= F_0, && j = 0, \\ -A_j Y_{j-1} + C_j Y_j - B_j Y_{j+1} &= F_j, && 1 \le j \le N-1, \\ -A_N Y_{N-1} + C_N Y_N &= F_N, && j = N, \end{aligned} \tag{6}$$

where Y_j and F_j are vectors of dimension M_j, C_j is a square $M_j \times M_j$ matrix, and A_j and B_j are rectangular matrices of size $M_j \times M_{j-1}$ and $M_j \times M_{j+1}$ respectively.

Systems of the form (6) are obtained from difference schemes for second-order elliptic equations with variable coefficients in arbitrary regions of any number of dimensions. In the two-dimensional case, as in the example analyzed above, the vector Y_j is the vector of unknowns in the j^{th} row of the grid $\bar{\omega}$, and in the three-dimensional case, it is the vector of unknowns in the j^{th} layer of the grid $\bar{\omega}$. In the latter case, C_j is a block-tridiagonal matrix with tridiagonal matrices on the main diagonal.

To solve the system (6), we will look at the *block elimination method*, which is analogous to the elimination method for three-point scalar equations.

2.4.2 Elimination for three-point vector equations. We now construct a method for solving a system of three-point vector equations (6). This system is related to a system of scalar three-point equations, methods for which we studied in Section 2.1. Thus, we will seek the solution of (6) in the form

$$Y_j = \alpha_{j+1} Y_{j+1} + \beta_{j+1}, \quad j = N-1, N-2, \ldots, 0, \tag{7}$$

where α_{j+1} is an as yet undefined matrix of size $M_j \times M_{j+1}$, and β_{j+1} is a vector of dimension M_j. From the formula (7) and the equations for the system (6) for $1 \le i \le N-1$, recurrences relations are found for the matrices α_j and the vectors β_j (as in the case of regular elimination). From (7) and (6) for $j = 0, N$, the initial values for α_1, β_1 and Y_N are found, allowing us to begin using the recurrence relations. Here are the final formulas for the method, which will be called the *block elimination method*:

$$\alpha_{j+1} = (C_j - A_j \alpha_j)^{-1} B_j, \qquad j = 1, 2, \ldots, N-1, \qquad \alpha_1 = C_0^{-1} B_0, \tag{8}$$

$$B_{j+1} = (C_j - A_j \alpha_j)^{-1} (F_j + A_j \beta_j), \; j = 1, 2, \ldots, N, \qquad \beta_1 = C_0^{-1} F_0, \tag{9}$$

$$Y_j = \alpha_{j+1} Y_{j+1} + \beta_{j+1}, \qquad j = N-1, N-2, \ldots, 0, \; Y_N = \beta_{N+1}. \tag{10}$$

We will say that the algorithm (8)–(10) is *correct* if the matrices C_0 and $C_j - A_j \alpha_j$ are non-singular for $1 \le j \le N$. Before defining stability for the algorithm (8)–(10), let us recall some results from linear algebra.

Let A be an arbitrary rectangular $m \times n$ matrix.

Let $\| x \|_n$ be a norm for the vector x in the n-dimensional space H_n. Then the norm of A is defined by the equation

$$\| A \| = \sup_{x \neq 0} \frac{\| Ax \|_m}{\| x \|_n}.$$

Obviously, the norm of A depends both on the matrix A, and on the vector norms introduced in H_n and H_m. For the case of the Euclidean norm in H_n and H_m ($\| x \|_n^2 = \sum_{i=i}^n x_i^2$), we have $\| A \| = \sqrt{\rho}$, where ρ is the largest eigenvalue in modulus of the matrix A^*A.

From the definition of the norm, it clearly follows that $\|Ax\|_m \leq \|A\| \|x\|_n$.

Further, suppose we are given matrices A and B of dimensions $m \times n$ and $n \times k$ respectively. Introducing vector norms in H_m, H_k and H_n, and defining with their aid the norms of the matrices A, B, and AB, we obtain the inequality $\| AB \| \leq \| A \| \| B \|$.

We will say that the algorithm is stable if $\| \alpha_j \| \leq 1$ for $1 \leq j \leq N$ (it is assumed, in the finite-dimensional spaces H_M to which the vectors Y_j belong, that a single type of norm has been introduced, for example Euclidean).

Lemma 5. *If the matrices C_j are non-singular for $0 \leq j \leq N$, and A_j and B_j are non-null matrices for $1 \leq j \leq N-1$, and the conditions*

$$\| C_0^{-1} B_0 \| \leq 1, \ \| C_N^{-1} A_N \| \leq 1, \ \| C_j^{-1} A_j \| + \| C_j^{-1} B_j \| \leq 1, \ 1 \leq j \leq N-1,$$

are satisfied where at least one of the inequalities is strict, then the algorithm (8)–(10) *of the block elimination method is stable and correct.*

Proof. We will give only the basic step, leaving it to the reader to complete the proof of the lemma. The proof uses the well-known fact: if the square matrix S satisfies $\| S \| \leq q \leq 1$, then the matrix $E - S$ is invertible, and $\| (E-S)^{-1} \| \leq 1/(1-q)$.

Let us assume now that $\| \alpha_j \| \leq 1$. From this and from the conditions of the lemma, we have

$$\| C_j^{-1} A_j \alpha_j \| \leq \| C_j^{-1} A_j \| \leq 1 - \| C_j^{-1} B_j \| < 1.$$

Since $C_j^{-1} A_j \alpha_j$ is a square matrix, the matrices $E - C_j^{-1} A_j \alpha_j$ and $C_j - A_j \alpha_j$ are invertible, and $\| (E - C_j^{-1} A_j \alpha_j)^{-1} \| \leq 1/\| C_j^{-1} B_j \|$. From this and (8), we at once obtain

$$\begin{aligned} \| \alpha_{j+1} \| &\leq \| (E - C_j^{-1} \Delta_j \alpha_j)^{-1} C_j^{-1} B_j \| \\ &\leq \| (E - C_j^{-1} A_j \alpha_j)^{-1} \| \cdot \| C_j^{-1} B_j \| \leq 1. \end{aligned}$$

The proof is completed by induction. □

We now apply lemma 5 to the system of the three-point vector equations (5) obtained from the Dirichlet difference problem for Poisson's equation in a rectangle. The system (5) is a special case of (6), where $C_j = C$, $B_j = A_j = E$, $1 \le j \le N-1$, $C_0 = C_N = E$, $B_0 = A_N = 0$, and the square matrix C is given in (4). The conditions of lemma 5 for this example take the form $\| C^{-1} \| \le 0.5$. For the case of the Euclidean norm we have, using the symmetry of C,

$$\| C^{-1} \| = \max_k |\lambda_k(C^{-1})| = \frac{1}{\min_k |\lambda_k(C)|},$$

where $\lambda_k(C)$ is an eigenvalue of the matrix C. From the definition of C we obtain that $\lambda_k(C)$ is an eigenvalue of the operator Λ defined above

$$\Lambda v(i) - \lambda_k v(i) = (2 - \lambda_k)v(i) - h_2^2 v_{\bar{x}_1 x_1}(i) = 0,$$
$$v(0) = v(M) = 0, \quad 1 \le i \le M-1.$$

If we substitute $\lambda_k = 2 + h_2^2 \mu_k$, this problem reduces to the eigenvalue difference problem considered in Section 1.5.1 for the simple difference operator: $v_{\bar{x}_1 x_1} + \mu_k v = 0$, $1 \le i \le M-1$, $v(0) = v(M) = 0$. Since this problem has the solution

$$\mu_k = \frac{4}{h_1^2} \sin^2 \frac{k\pi h_1}{2l_1} > 0, \quad k = 1, 2, \ldots, M-1,$$

$\lambda_k = \lambda_k(C) = 2 + h_2^2 \mu_k > 2$. Consequently, the condition $\| C^{-1} \| \le 0.5$ is satisfied. The algorithm (8)–(10) applied to the system (5) is correct and stable.

We consider now the question of the storage requirements and operation counts for the algorithm (8)–(10), assuming for simplicity that in the system (6) all the matrices are square and of size $M \times M$, and all the vectors Y_j and F_j have dimension M. In this case, the elimination coefficients α_j will be square matrices of size $M \times M$, and the vectors β_j will have dimension M.

To realize (8)–(10) it is necessary to store all the matrices α_j for $1 \le j \le N$, all the vectors β_j for $1 \le j \le N+1$, and the matrix $(C_N - A_N \alpha_N)^{-1}$ used to compute β_{N+1}. The vectors β_j can be stored in the positions reserved for the vectors of unknowns Y_{j-1}. To store all the matrices α_j and the matrix $(C_N - A_N \alpha_N)^{-1}$ it is necessary to retain $M^2(N+1)$ elements, since in the general case the matrices α_j are full and non-symmetric. The algorithm also requires M times as many auxiliary storage locations as there are unknowns in the problem, that is $M(N+1)$.

In the general case, the matrices $C_j - A_j\alpha_j$ are full for every j. Therefore, inverting them requires $O(M^3)$ arithmetic operations. Further, multiplying $(C_j - A_j\alpha_j)^{-1}$ by the matrix B_j requires not more than $O(M^3)$ operations. Therefore, computing α_{j+1} from α_j using (8) requires $O(M^3)$ arithmetic operations. To compute all the α_j and the matrix $(C_N - A_N\alpha_N)^{-1}$ requires $O(M^3N)$ operations.

If the matrix A_j is full, then computing β_{j+1} given β_j and $(C_j - A_j\alpha_j)^{-1}$ requires: $2M^2$ multiplications and $2M^2 - M$ additions. If A_j is diagonal, then the requirements are reduced to $M^2 + M$ multiplications and $(2M^2 - M)$ additions. Consequently, to compute β_j for $2 \le j \le N+1$ requires in the general case $2M^2N$ multiplications and $(2M^2 - M)N$ additions. Adding in the operations required to compute β_1 given C_0^{-1} (M^2 multiplications and $M^2 - M$ additions), we finally obtain that $M^2(2N+1)$ multiplications and $M^2(2N+1) - M(N+1)$ additions are used.

To find all the Y_j for $0 \le j \le N-1$ given Y_N requires M^2N multiplications and M^2N additions. Thus, to compute β_j and Y_j requires $M^2(3N+1)$ multiplications and $M^2(3N+1) - M(N+1)$ additions. If no distinction is made between these operations, this constitutes $Q \approx 6M^2N$ operations. This is the number of arithmetic operations necessary to find the solution to a new problem in a series. To solve the original problem (6), where it is necessary to compute the elimination matrices α_j, requires $Q = O(M^3N + M^2N)$ operations.

Suppose the series consists of n problems of the form (6). Then we must perform $Q_n = O(M^3N) + 6nM^2N$ operations. Here the number of unknowns in the series is equal to $nM(N+1)$. From this it follows that finding one unknown requires $q \approx O(M^2/n) + 6M$ arithmetic operations. Thus, as n increases, the number of operations per unknown decreases, but it is always greater than $6M$. This distinguishes the block elimination method from the scalar elimination method, where the number of operations per unknown is a finite quantity which does not depend on the number of unknowns.

2.4.3 Elimination for two-point vector equations. We now consider a method for solving two-point vector equations

$$\begin{aligned} P_{i+1}V_{i+1} - Q_iV_i = F_{i+1}, \quad 0 \le i \le N-1, \\ P_0V_0 = F_0, \quad Q_NV_N = F_{N+1}, \end{aligned} \tag{11}$$

where V_i is a vector of dimension M, P_{i+1} and Q_i, $0 \le i \le N-1$, are square $M \times M$ matrices, P_0 and Q_N are rectangular matrices of size $M_1 \times M$ and $M_2 \times M$ respectively, where $M_1 + M_2 = M$. The vector F_{i+1}, $0 \le i \le N-1$, has dimension M, and F_0 and F_{N+1} are of size M_1 and M_2 respectively.

We first transform the system (11) to the form (6). To do this, we transform the matrices in (11) to the following form:

$$\begin{aligned} P_0 &= \left\| P_0^{11} | - P_0^{12} \right\|, \qquad Q_N = \left\| -Q_N^{21} | Q_N^{22} \right\|, \\ P_{i+1} &= \left\| \frac{P_{i+1}^{11} | - P_{i+1}^{12}}{P_{i+1}^{21} | - P_{i+1}^{22}} \right\|, \quad Q_i = \left\| \frac{Q_i^{11} | - Q_i^{12}}{Q_i^{21} | - Q_i^{22}} \right\|, \end{aligned} \tag{12}$$

where P_i^{kl} and Q_i^{kl}, $0 \le i \le N$, are matrices of size $M_k \times M_l$, $k, l = 1, 2$. Corresponding to the transformation (12), we set

$$\begin{aligned} &V_i = \begin{pmatrix} v_i^1 \\ v_i^2 \end{pmatrix}, \quad 0 \le i \le N, \quad F_{i+1} = \begin{pmatrix} f_{i+1}^1 \\ f_{i+1}^2 \end{pmatrix}, \quad 0 \le i \le N-1, \\ &F_0 = f_0^1, \quad F_{N+1} = f_{N+1}, \end{aligned} \tag{13}$$

where v_i^k and f_i^k are vectors of dimension M_k, $k = 1, 2$. Using (12) and (13), we write the system (11) in the following form:

$$P_0^{11} v_0^1 - P_0^{12} v_0^2 = f_0^1,$$

$$\left.\begin{aligned} -Q_i^{11} v_i^1 + Q_i^{12} v_i^2 + P_{i+1}^{11} v_{i+1}^1 - P_{i+1}^{12} v_{i+1}^2 &= f_{i+1}^1, \\ -Q_i^{21} v_i^1 + Q_i^{22} v_i^2 + P_{i+1}^{21} v_{i+1}^1 - P_{i+1}^{22} v_{i+1}^2 &= f_{i+1}^2, \end{aligned}\right\} 0 \le i \le N-1, \tag{14}$$

$$-Q_N^{21} v_N^1 + Q_N^{22} v_N^2 = f_{N+1}^2.$$

We now introduce a new vector of unknowns, setting

$$Y_0 = v_0^1, \quad Y_{N+1} = v_N^2, \quad Y_{i+1} = \begin{pmatrix} v_i^2 \\ v_{i+1}^1 \end{pmatrix}, \quad 0 \le i \le N-1,$$

and the matrices

$$C_0 = P_0^{11}, \quad B_0 = \| P_0^{12} | 0^{11} \|, \quad C_{N+1} = Q_N^{22}, \quad A_{N+1} = \| 0^{22} | Q_N^{21} \|,$$

$$A_1 = \left\| \frac{Q_0^{11}}{Q_0^{21}} \right\|, \quad B_N = \left\| \frac{P_N^{12}}{P_N^{22}} \right\|, \quad A_{i+1} = \left\| \frac{0^{12} | Q_i^{11}}{0^{22} | Q_i^{21}} \right\|, \quad 1 \le i \le N-1,$$

$$B_{i+1} = \left\| \frac{P_{i+1}^{12} | 0^{11}}{P_{i+1}^{22} | 0^{21}} \right\|, \ 0 \le i \le N-2, \quad C_{i+1} = \left\| \frac{Q_i^{12} | P_{i+1}^{11}}{Q_i^{22} | P_{i+1}^{21}} \right\|, \quad 0 \le i \le N-1,$$

where 0^{kl} is the zero matrix of size $M_k \times M_l$, $k, l = 1, 2$.

With these substitutions, the system (14) will have the form

$$\begin{aligned} C_0Y_0 - B_0Y_1 &= F_0, && i=0,\\ -A_iY_{i-1} + C_iY_i - B_iY_{i+1} &= F_i, && 1 \le i \le N,\\ -A_{N+1}Y_N + C_{N+1}Y_{N+1} &= F_{N+1}, && i = N+1. \end{aligned} \tag{15}$$

Thus, the system of two-point vector equations (11) has been changed into a system of three-point vector equations of the form (15), which can be solved using the block elimination method constructed in Section 2.5.2. For (15), the block elimination method has the following from:

$$\alpha_{i+1} = (C_i - A_i\alpha_i)^{-1}B_i, \qquad i = 1,2,\dots,N, \qquad \alpha_1 = C_0^{-1}B_0, \tag{16}$$

$$\beta_{i+1} = (C_i - A_i\alpha_i)^{-1}(F_i + A_i\beta_i), \quad i = 1,2,\dots,N+1, \qquad \beta_1 = C_0^{-1}B_0, \tag{17}$$

$$Y_i = \alpha_{i+1}Y_{i+1} + \beta_{i+1}, \qquad i = N, N-1,\dots,0, \quad Y_{N+1} = \beta_{N+2}, \tag{18}$$

where the matrices α_1 and α_N have dimensions $M_1 \times M$ and $M \times M_2$ respectively, and α_i is a square $M \times M$ matrix for $2 \le i \le N$. For $2 \le i \le N+1$, the vectors β_i have dimension M, and β_1 and β_{N+2} have dimension M_1 and M_2.

We will now transform the formulas (16)–(18). Taking into account the structure of the matrices B_i, we find that the matrices α_i have the form

$$\alpha_1 = \| \alpha_1^{12} | 0^{11} \|, \quad \alpha_{N+1} = \left\| \frac{\alpha_{N+1}^{22}}{\alpha_{N+1}^{12}} \right\|, \quad \alpha_i = \left\| \frac{\alpha_i^{22} | 0^{21}}{\alpha_i^{12} | 0^{11}} \right\|, \quad 2 \le i \le N. \tag{19}$$

Substituting (19) in (16) and using the definition of the matrices A_i, B_i, and C_i, we obtain the formulas for computing α_i^{12} and α_i^{22}

$$\left\| \frac{\alpha_{i+1}^{22}}{\alpha_{i+1}^{12}} \right\| = \left\| \frac{Q_{i-1}^{12} - Q_{i-1}^{11}\alpha_i^{12}|P_i^{11}}{Q_{i-1}^{22} - Q_{i-1}^{21}\alpha_i^{12}|P_i^{21}} \right\|^{-1} \left\| \frac{P_i^{12}}{P_i^{22}} \right\|, \quad 1 \le i \le N, \tag{20}$$

where $\alpha_1^{12} = (P_0^{11})^{-1}P_0^{12}$. Further, writing the vector β_i in the form

$$\beta_1 = \beta_1^1, \quad \beta_{N+2} = \beta_{N+2}^2, \quad \beta = \begin{pmatrix} \beta_i^2 \\ \beta_i^1 \end{pmatrix}, \quad 2 \le i \le N+1 \tag{21}$$

and substituting this expression in (17), we obtain

$$\begin{pmatrix} \beta_{i+1}^2 \\ \beta_{i+1}^1 \end{pmatrix} = \left\| \frac{Q_{i-1}^{12} - Q_{i-1}^{11}\alpha_i^{12}|P_i^{11}}{Q_{i-1}^{22} - Q_{i-1}^{21}\alpha_i^{12}|P_i^{21}} \right\|^{-1} \begin{pmatrix} f_i^1 + Q_{i-1}^{11}\beta_i^1 \\ f_i^2 + Q_{i-1}^{21}\beta_i^1 \end{pmatrix}, \quad 1 \le i \le N, \tag{22}$$

$$\beta_{N+2}^2 = \| Q_N^{22} - Q_N^{21}\alpha_{N+1}^{12} \|^{-1} (f_{N+1}^2 + Q_N^{21}\beta_{n+1}^1), \tag{23}$$

where $\beta_1^1 = \| P_0^{11} \|^{-1} f_0^1$.

We now substitute (19) and (21) in (18) and use the definition of Y_i. As a result we obtain the following formulas for computing the components of the vector of unknowns:

$$\begin{aligned} v_{i-1}^2 &= \alpha_{i+1}^{22} v_i^2 + \beta_{i+1}^2, \quad i = N, N-1, \ldots, 1, \quad v_N = \beta_{N+2}^2, \\ v_i^1 &= \alpha_{i+1}^{12} v_i^2 + \beta_{i+1}^1, \quad i = N, N-1, \ldots, 0. \end{aligned} \tag{24}$$

Thus, the *block elimination algorithm* for systems of two-point vector equations (11) is described by the formulas (20), (22)–(24).

Since these formulas are derived from the elimination algorithm for solving (15), to which our original two-point vector equations were transformed, the sufficient conditions for correctness and stability for the resulting algorithm are formulated in lemma 5, where it is necessary to change N to $N+1$, and where C_i, A_i, and B_i are defined above.

Using the two-way elimination algorithm for the system (15), it is possible to construct a corresponding algorithm for the original system of two-point vector equations (11).

2.4.4 Orthogonal elimination for two-point vector equations. We shall consider yet another method for solving the system of two-point equations (11), known as the *orthogonal elimination method.* This method involves inverting the matrices P_i for $1 \leq i \leq N$ and orthogonalizing the auxiliary rectangular matrices.

We will find the solution of the system (11) in the following form

$$V_i = B_i \beta_i + Y_i, \quad 0 \leq i \leq N, \tag{25}$$

where, for any i, B_i is a rectangular $M \times M_2$ matrix, and β_i and Y_i are vectors of size M_2 and M respectively.

Defining B_0 and Y_0 from the conditions $P_0 B_0 = 0^{12}$, $P_0 Y_0 = F_0$, where 0^{12} is the zero matrix of size $M_1 \times M_2$, we obtain that V_0 satisfies the condition $P_0 V_0 = F_0$. We shall now find the recurrence formulas for sequentially constructing, starting with B_0 and Y_0, the matrices B_i and Y_i.

We substitute (25) in (11). If P_{i+1} is non-singular, then we have that

$$B_{i+1}\beta_{i+1} + Y_{i+1} - P_{i+1}^{-1} Q_i B_i \beta_i = P_{i+1}^{-1}(F_{i+1} + Q_i Y_i), \quad 0 \leq i \leq N-1,$$

or

$$B_{i+1}\beta_{i+1} + Y_{i+1} - A_{i+1}\beta_i = X_{i+1}, \quad 0 \leq i \leq N-1, \tag{26}$$

where $A_{i+1} = P_{i+1}^{-1} Q_i B_i$, $X_{i+1} = P_{i+1}^{-1}(F_{i+1} + Q_i Y_i)$. The matrix A_{i+1} has size $M \times M_2$, and the vector X_{i+1} is of size M.

We determine B_{i+1} and Y_{i+1} in the following way

$$A_{i+1} = B_{i+1}\Omega_{i+1}, \quad Y_{i+1} = X_{i+1} - B_{i+1}\varphi_{i+1}, \tag{27}$$

where Ω_{i+1} and φ_{i+1} are an as yet undetermined square $M_2 \times M_2$ matrix and an M_2-vector. Substituting (27) in (26), we obtain the relation $B_{i+1}(\beta_{i+1} - \Omega_{i+1}\beta_{i+1}) = B_{i+1}\varphi_{i+1}$, which becomes an identity if we set

$$\Omega_{i+1}\beta_i = \beta_{i+1} - \varphi_{i+1}, \quad 0 \le i \le N-1. \tag{28}$$

Thus, given non-singular matrices Ω_i and vectors φ_i for $1 \le i \le N$, the formula (27) can be used to find, starting with B_0 and Y_0, all the necessary matrices B_i and vectors Y_i for $1 \le i \le N$.

It remains to define the vectors β_i. From (11) and (25) for $i = N$ we obtain the two relations $V_N = B_N\beta_N + Y_N$, $Q_N V_N = F_{N+1}$ with the known B_N and Y_N. Hence for β_N we have the equation $Q_N B_N \beta_N = F_{N+1} - Q_N Y_N$. This relation can be written in the form (28)

$$\Omega_{N+1}\beta_N = \beta_{N+1} - \varphi_{N+1}, \tag{29}$$

where $\beta_{N+1} = F_{N+1}$, $\varphi_{N+1} = Q_N Y_N$, $\Omega_{N+1} = Q_N B_N$.

If the matrix Ω_{N+1} is not singular, we can find all the β_i, $0 \le i \le N$, sequentially starting from β_{N+1} using the formulas (28), (29). The solution of the system (11) can be found using (25).

Since there is an arbitrariness in the choice of the matrices Ω_i and the vectors φ_i, the formulas derived above describe more a principle for constructing methods for solving (11), rather than a concrete algorithm. The choice of specific Ω_i and φ_i gives rise to several methods for the system (11). As before, we shall call such methods elimination methods, where on the forward path we compute B_i and Y_i, and on the reverse path — β_i and the solution V_i.

We shall examine now one possible choice for Ω_i and φ_i. Since the formulas (27) and (28) require the inverse of the matrix Ω_{i+1}, it must be easy to invert.

In the orthogonal elimination method, the matrix Ω_{i+1} and the vector φ_{i+1} are generated by the requirements: 1) the matrix B_{i+1} is constructed by orthonormalizing the columns of the matrix A_{i+1}; 2) the vector Y_{i+1} must be orthogonal to the columns of the matrix B_{i+1}.

As a consequence of these requirements we have

$$B^*_{i+1}B_{i+1} = E^{22}, \quad B^*_{i+1}Y_{i+1} = 0, \tag{29'}$$

where B^*_{i+1} is the conjugate of the matrix B_{i+1}, and E^{22} is the identity matrix of size $M_2 \times M_2$.

We first find an expression for φ_{i+1}. From (27) and (29′) we obtain $0 = B^*_{i+1}Y_{i+1} = B^*_{i+1}X_{i+1} - B^*_{i+1}B_{i+1}\varphi_{i+1} = B^*_{i+1}X_{i+1} - \varphi_{i+1}$. Thus, the vector φ_{i+1} is determined: $\varphi_{i+1} = B^*_{i+1}X_{i+1}$.

We now construct the matrices Ω_{i+1} and B_{i+1}. There exist several methods for orthonormalizing the columns of the matrix A_{i+1}. We shall consider the Gram-Schmidt method.

Suppose that the matrix A_{i+1} has rank M_2. We denote by a_k and b_k the k^{th} columns of the matrices A_{i+1} and B_{i+1} respectively, and by $(\ ,\ )$ the vector inner-product. As b_1 we take the normed column a_1

$$b_1 = a_1/\omega_{11}, \quad \omega_{11} = \sqrt{(a_1, a_1)}. \tag{30}$$

We will find the column b_k in the form

$$b_k = \frac{1}{\omega_{kk}}\left(a_k - \sum_{n=1}^{k-1}\omega_{nk}b_n\right), \quad 2 \le k \le M_2, \tag{31}$$

where the coefficients ω_{nk} are found from the orthogonality conditions for b_k with $b_1, b_2, \ldots, b_{k-1}$, and ω_{kk} is found from the condition on the norm of b_k:

$$\omega_{nk} = (b_n, a_k), \quad n = 1, 2, \ldots, k-1, \quad \omega_{kk} = \sqrt{(a_k, a_k) - \sum_{n=1}^{k}\omega^2_{nk}}. \tag{32}$$

Because of the assumption about the rank of the matrix A_{i+1}, the columns a_k are linearly independent for $1 \le k \le M_2$, and the ortonormalization process can be carried out without any problems.

From (30)–(32) it follows that the matrices A_{i+1} and B_{i+1} are connected by the relation $A_{i+1} = B_{i+1}\Omega_{i+1}$, where Ω_{i+1} is the square upper-triangular matrix of size $M_2 \times M_2$ with elements ω_{nk} for $1 \le n \le M_2$, $n \le k \le M_2$, defined in (30) and (32), and $\omega_{nk} = 0$ for $k < n$.

Thus, the formulas (30)–(32) determine the matrices B_{i+1} and Ω_{i+1}. A simple computation shows that the matrices B_{i+1} and Ω_{i+1} can be constructed using: $MM_2^2 + 0.5(M_2^2 - M_2)$ multiplications, $MM_2^2 - M_2$ additions and subtractions, MM_2 divisions, and M_2 square roots. All the indicated operations must be carried out N times on the forward path of the elimination algorithm. This requires $O(MNM_2^2)$ arithmetic operations and NM_2 square roots.

All that remains for us to show is how to find the matrix B_0 and the vector Y_0. We will assume that the matrices P_{i+1} and Q_i are not singular for $0 \le i \le N-1$. In addition, assume that the matrix P_0^{11} is non-singular and that the matrix Q_N has rank M_2.

Let us construct B_0 and Y_0. Let

$$A_0 = \left\| \frac{(P_0^{11})^{-1} P_0^{12}}{E^{22}} \right\|, \quad X_0 = \begin{pmatrix} (P_0^{11})^{-1} F_0 \\ 0 \end{pmatrix}$$

be a rectangular matrix of size $M \times M_2$ and a vector of dimension M. Since the dimension of the square identity matrix E^{22} is $M_2 \times M_2$, the rank of A_0 is equal to M_2. The matrix B_0 is constructed from A_0 using the orthonormalization process (30)–(32), Y_0 is obtained from the formula $Y_0 = X_0 - B_0\varphi_0$, and the orthogonality condition for the matrix B_0 gives $\varphi_0 = B_0^* X_0$. Since

$$B_0 = A_0\Omega_0^{-1}, \quad P_0 A_0 = \| P_0^{11} | - P_0^{12} \| \left\| \frac{(P_0^{11})^{-1} P_0^{12}}{E^{22}} \right\| = \| 0^{12} \|,$$

$P_0 B_0 = 0^{12}$. Further, we have

$$P_0 Y_0 = P_0 X_0 - P_0 B_0 \varphi_0 = P_0 X_0 = F_0.$$

Thus, the constructed B_0 and Y_0 satisfy the required relations: $P_0 B_0 = 0^{12}$ and $P_0 Y_0 = F_0$.

Notice that, because of the non-singularity of P_{i+1} and Q_i, the rank of the matrix A_{i+1} is the same as the rank of B_i. In addition, because of the non-singularity of Ω_0, the rank of B_0 is the same as the rank of A_0 and is equal to M_2. Therefore, the orthonormalization process (30)–(32) will proceed without complications. Further, since the ranks of the matrices Q_N and B_N are equal to M_2, the square matrix $\Omega_{N+1} = Q_N B_N$ will be non-singular, which allows us to find the vector β_N.

Thus, the algorithm for orthogonal elimination has the following form:

[1] $B_i\Omega_i = A_i$, $i = 0, 1, 2, \ldots, N$,

$$A_i = P_i^{-1} Q_{i-1} B_{i-1}, \quad 1 \le i \le N, \quad A_0 = \left\| \frac{(P_0^{11})^{-1} P_0^{12}}{E^{22}} \right\|. \tag{33}$$

The matrices B_i and Ω_i for $0 \le i \le N$ are computed from (30)–(32) and stored. We set $\Omega_{N+1} = Q_N B_N$.

[2] $Y_i = X_i - B_i\varphi_i$, $\varphi_i = B_i^* X_i$, $i = 0, 1, \ldots, N$,

$$X_i = P_i^{-1}(F_i + Q_{i-1} Y_{i-1}); \quad 1 \le i \le N, \quad X_0 = \begin{pmatrix} (P_0^{11})^{-1} F_0 \\ 0 \end{pmatrix}. \tag{34}$$

We compute and store the vectors Y_i and φ_i for $0 \le i \le N$. We set $\varphi_{N+1} = Q_N Y_N$.

[3] $\Omega_{i+1}\beta_i = \beta_{i+1} - \varphi_{i+1}$, $i = N, N-1, \ldots, 0$, $\beta_{N+1} = F_{N+1}$,

$$V_i = B_i\beta_i + Y_i, \quad 0 \le i \le N. \tag{35}$$

Remark. Since the matrices Ω_i are upper-triangular $M_2 \times M_2$ matrices for $1 \le i \le N$, computing β_i from β_{i+1} and φ_{i+1} requires $O(M_2^2)$ operations.

To illustrate this algorithm, we consider an example. Suppose we must solve the following three-point difference problem:

$$\begin{aligned} -y_{i-1} + y_i - y_{i+1} &= 0, \quad 1 \le i \le N-1, \\ y_0 = 1, \quad y_N &= 0. \end{aligned} \tag{36}$$

This problem was examined earlier in Section 2.4, where the non-monotonic three-point elimination method was used to find its solution for N not divisible by 3, namely,

$$y_i = \frac{\sin\dfrac{(N-i)\pi}{3}}{\sin\dfrac{N\pi}{3}}, \quad 0 \le i \le N.$$

We shall transform the system (36) to a system of two-point vector equations of the form (11) by setting

$$V_i = \begin{pmatrix} y_i \\ y_{i+1} \end{pmatrix}, \quad 0 \le i \le N-1.$$

It is not difficult to see that (36) is equivalent to the following system

$$\begin{aligned} V_{i+1} - QV_i &= 0, \quad 0 \le i \le N-2, \\ P_0V_0 &= 1, \quad Q_{N-1}V_{N-1} = 0, \end{aligned} \tag{37}$$

where $P_0 = \| \, 1|0 \, \|$, $Q_{N-1} = \| \, 0|1 \, \|$, $Q = \left\| \begin{smallmatrix} 0 & 1 \\ -1 & 1 \end{smallmatrix} \right\|$. The system (37) is a special case of (11) with $M_1 = M_2 = 1$, $M = 2$.

To solve (37) we use the orthogonal elimination algorithm (33)–(35). For this example, the matrices B_i have dimension 2×1, Ω_i is of dimension 1×1, the vectors Y_i are of size 2, and the vectors β_i and φ_i are of size 1.

In Table 3 are shown the matrices B_i and Ω_i and also the vectors Y_i, φ_i and β_i for $N = 11$. Applying the orthogonal elimination method allows us to obtain an accurate solution y_i to the problem (36).

Table 3

i	0	1	2	3	4	5	6	7	8	9	10	11
Ω_i	1	$\sqrt{2}$	$\frac{1}{\sqrt{2}}$	1	$\sqrt{2}$	$\frac{1}{\sqrt{2}}$	1	$\sqrt{2}$	$\frac{1}{\sqrt{2}}$	1	$\sqrt{2}$	$-\frac{1}{\sqrt{2}}$
φ_i	0	$-\frac{1}{\sqrt{2}}$	$-\frac{1}{2}$	1	$-\frac{1}{\sqrt{2}}$	$-\frac{1}{2}$	1	$-\frac{1}{\sqrt{2}}$	$-\frac{1}{2}$	1	$-\frac{1}{\sqrt{2}}$	$\frac{1}{2}$
β_i	1	$\frac{1}{\sqrt{2}}$	0	1	$\frac{1}{\sqrt{2}}$	0	1	$\frac{1}{\sqrt{2}}$	0	1	$\frac{1}{\sqrt{2}}$	0
B_i	$\binom{0}{1}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$	$\binom{1}{0}$	$\binom{0}{-1}$	$\begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$	$\binom{-1}{0}$	$\binom{0}{1}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$	$\binom{1}{0}$	$\binom{0}{-1}$	$\begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$	
Y_i	$\binom{1}{0}$	$\begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$	$\binom{0}{-1}$	$\binom{-1}{0}$	$\begin{pmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$	$\binom{0}{1}$	$\binom{1}{0}$	$\begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$	$\binom{0}{-1}$	$\binom{-1}{0}$	$\begin{pmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$	
y_i	1	1	0	−1	−1	0	1	1	0	−1	−1	0

2.4.5 Elimination for three-point equations with constant coefficients. We now turn again to the block-elimination method for three-point equations and consider a special case of such equations, namely:

$$\begin{aligned} -Y_{j-1} + CY_j - Y_{j+1} &= F_j, \quad 1 \le j \le N-1, \\ Y_0 &= F_0, \quad Y_N = F_N, \end{aligned} \tag{38}$$

where C is a square $M \times M$ matrix, and Y_j and F_j are unknown and given vectors of size M.

In Section 4.1 it was shown that systems of three-point equations of the form (38) arise from Dirichlet difference problems for Poisson's equation on a rectangular grid defined in a rectangle, and where the matrix C is symmetric and tridiagonal. Further, in Section 2.4.2 it was shown that the block elimination method, which for (38) has the form

$$\alpha_{j+1} = (C - \alpha_j)^{-1}, \qquad j = 1, 2, \ldots, N-1, \qquad \alpha_1 = 0, \tag{39}$$

$$\beta_{j+1} = \alpha_{j+1}(F_j + \beta_j), \qquad j = 1, 2, \ldots, N-1, \qquad \beta_1 = F_0, \tag{40}$$

$$Y_i = \alpha_{j+1} Y_{j+1} + \beta_{j+1}, \quad j = N-1, N-2, \ldots, 1, \quad Y_N = F_N, \tag{41}$$

is correct and stable. It was also shown there that the eigenvalues of the matrix C are greater than 2:

$$\lambda_k = \lambda_k(C) = 2 + 4\frac{h_2^2}{h_1^2} \sin^2 \frac{k\pi h_1}{2l_1} > 2. \tag{42}$$

Let us recall that, for the case of general three-point vector equations, the block elimination algorithm requires $O(M^3N)$ arithmetic operations for the computation of the matrices a_j, and $O(M^2N)$ operations for the computation of the elimination vectors β_j and the solution Y_j. To store the full and, generally speaking, non-symmetric matrices α_j requires that we retain the $M^2(N+1)$ elements of these matrices. Are these quantities reduced if the block elimination method is used to solve special three-point vector systems (38) with constant coefficients?

For the above example, all the matrices α_j will be symmetric due to the symmetry of the matrix C, but although C is tridiagonal, all the matrices α_j, $j \geq 2$, will be full. Consequently, it is possible, taking into account the symmetry of the matrices α_j, only to decrease the volume of the intermediate storage required, but not by more than a factor of two. The estimate for the number of arithmetic operations is not changed.

We now construct a modification of the algorithm (39)–(41) which does not require auxiliary storage for saving intermediate information, and which can be realized in $O(MN^2)$ operations if we are solving the problem (38) with a tridiagonal matrix C.

First of all we find an explicit form for the elimination matrices α_j for any j. For this we express α_j in terms of the matrix C using (39). Noting that

$$\alpha_1 = 0, \quad \alpha_2 = C^{-1}, \quad \alpha_3 = (C^2 - E)^{-1}C, \tag{43}$$

we find the solution of the non-linear difference equation (39) in the form

$$\alpha_j = P_{j-1}^{-1}(C)P_{j-2}(C), \quad j \geq 2, \tag{44}$$

where $P_j(C)$ is a polynomial in C of degree j. We rewrite (39) in the form

$$\alpha_{j+1}(C - \alpha_j) = E, \quad j \geq 2,$$

and substitute here (44). We obtain the recurrence relation

$$P_j(C) = CP_{j-1}(C) - P_{j-2}(C), \quad j \geq 2,$$

or upon shifting the index by 1 and using (43)

$$\begin{aligned} P_{j+1}(C) &= CP_j(C) - P_{j-1}(C), \quad j \geq 1, \\ P_0(C) &= E, \quad P_1(C) = C. \end{aligned} \tag{45}$$

Thus, the formulas (45) fully determine the polynomial $P_j(C)$ for any $j \geq 0$.

We now find the solution of (45). The corresponding algebraic polynomial satisfies the conditions

$$P_{j+1}(t) = tP_j(t) - P_{j-1}(t), \quad j \geq 1,$$
$$P_0(t) = 1, \quad P_1(t) = t.$$

which form a Cauchy problem for a three-point difference equation with constant coefficients. In Section 1.4.2, the solution $P_j(t) = U_j(t/2)$ was found, where $U_j(x)$ is the Chebyshev polynomial of the second kind of degree j

$$U_j(x) = \begin{cases} \dfrac{\sin((j+1)\arccos x)}{\sin \arccos x}, & |x| \leq 1 \\ \dfrac{\sinh((j+1)\operatorname{arccosh} x)}{\sinh \operatorname{arccosh} x}, & |x| \geq 1. \end{cases}$$

Thus, an explicit expression for the elimination matrices α_j has been found:

$$\alpha_j = U_{j-1}^{-1}\left(\frac{C}{2}\right) U_{j-2}\left(\frac{C}{2}\right), \quad j \geq 2, \quad \alpha_1 = 0. \tag{46}$$

This frees us from having to compute the elimination matrices α_j by the formula (39), which formed the bulk of the computational work in the algorithm (39)–(41). In addition, the matrices α_j need not be remembered.

We now look at the formulas (40) and (41). They involve multiplying the matrices α_{j+1} by the vectors $F_j + \beta_j$ and Y_{j+1}. We will now show that it is possible, without computing α_j by the formula (46), to determine the product of the matrix α_j and a vector. For this we require lemma 6, which we give without proof.

Lemma 6. *Suppose that the polynomial $f_n(x)$ of degree n has simple roots. The ratio of the polynomial $g_m(x)$ of degree m to the polynomial $f_n(x)$, where $n > m$ and there are no common roots, can be represented in the form of a sum of n elementary fractions*

$$\frac{g_m(x)}{f_n(x)} = \sum_{l=1}^{n} \frac{a_l}{x - x_l}, \quad a_l = \frac{g_m(x_l)}{f_n'(x_l)},$$

where x_l is a root of $f_n(x)$ and $f_n'(x)$ is the derivative of the polynomial $f_n(x)$.

Using lemma 6, we find the decomposition in simple fractions of the ratio $\varphi(x) = U_{j-2}(x)/U_{j-1}(x)$, $j \geq 2$. Since the roots of $U_{j-1}(x)$ are

$$x_k = \cos\frac{k\pi}{j}, \quad j = 1, 2, \ldots, j-1,$$

and

$$U_{j-2}(x_k) = (-1)^{k-1}, \quad \frac{d}{dx}[U_{j-1}(x_k)] = \frac{j(-1)^{k-1}}{\sin^2\frac{k\pi}{j}},$$

by lemma 6 we have the following decomposition for $\varphi(x)$:

$$\varphi(x) = \frac{U_{j-2}(x)}{U_{j-1}(x)} = \sum_{k=1}^{j-1} \frac{\sin^2\frac{k\pi}{j}}{j}\left(x - \cos\frac{k\pi}{j}\right)^{-1}. \tag{47}$$

From (46) and (47) there follows yet another representation for the matrices α_j, which we shall also use

$$\alpha_j = \sum_{k=1}^{j-1} a_{kj}\left(C - 2\cos\frac{k\pi}{j}E\right)^{-1}, \quad a_{kj} = \frac{2\sin^2\frac{k\pi}{j}}{j}, \quad j \geq 2. \tag{48}$$

Using (48), the product of the matrix α_j and the vector Y can be effected by the following algorithm: for $k = 1, 2, \ldots, j-1$ solve the equation

$$\left(C - 2\cos\frac{k\pi}{j}E\right)V_k = a_{kj}Y, \tag{49}$$

where α_{kj} is defined in (48), and the result $\alpha_j Y$ is obtained by summing the vectors V_k

$$\alpha_j Y = \sum_{k=1}^{j-1} V_k. \tag{50}$$

We remark that by (42) the matrix $C - 2\cos\frac{k\pi}{j}E$ is non-singular and also tridiagonal whenever C is. In this case, each of the equations (49) is solved in $O(M)$ arithmetic operations using the scalar three-point elimination method described in Section 2.1. Consequently, to solve all the problems (49) and also to compute the sum (50) requires $O(Mj)$ operations. Since in (40) and (41) the product of the matrix α_j and a vector is computed for $j = 2, 3, \ldots, N$, the modified block elimination method (40), (41) and (49), (50) requires $O(MN^2)$ arithmetic operations.

Thus, the *modified block elimination method* constructed above allows us to solve a Dirichlet difference problem for Poisson's equation in a rectangle using $O(MN^2)$ arithmetic operations. The reduction in the number of operations in comparison with the original algorithm (39)–(41) is achieved by taking into account the specifics of the problem being solved.

In the next two chapters we will look at other direct methods for solving the indicated problem and its related difference problems which will require even fewer operations than the method constructed here.

Chapter 3

The Cyclic Reduction Method

In this chapter we study a method for solving special grid elliptic equations — the cyclic reduction method. This direct method allows us to find the solution to a Dirichlet problem for Poisson's equation in a rectangle using $O(N^2 \log_2 N)$ arithmetic operations, where N is the number of grid nodes in any direction.

In Section 1 we state the boundary-value difference problems which can be solved using the cyclic reduction method. In Section 2 the algorithm is described for the case of a boundary-value problem of the first kind, and in Section 3 sample applications of the method are given. In Section 4 we give a generalization of the method for the case of general boundary conditions.

3.1 Boundary-value problems for three-point vector equations

3.1.1 Statement of the boundary-value problems. In Chapter 2 we constructed the scalar and block elimination methods to solve three-point scalar and vector equations. The block elimination method requires $O(M^3 N)$ arithmetic operations for equations with variable coefficients, where N is the number of equations, and M is the dimension of the vector of unknowns (the number of unknowns in the problem is equal to MN). For special classes of vector equations corresponding, for example, to a Dirichlet problem for Poisson's equation in a rectangle, a modification of the block elimination algorithm was presented. This algorithm allows the number of operations to be reduced to $O(MN^2)$.

This chapter is devoted to the further study of direct methods for solving special vector equations obtained from difference schemes for the simplest elliptic equations. We will construct the *cyclic reduction method*, which enables

us to solve the basic boundary-value problems in $O(MN \log_2 N)$ arithmetic operations. If we ignore the weak logarithmic dependence on N, the number of operations for this method is proportional to the number of unknowns MN. The creation of this method is an essential step in the development of both direct and iterative methods for solving grid equations.

We will formulate boundary-value problems for three-point vector equations which can be solved using the cyclic reduction method. We will consider the following problems:

(1) *A boundary-value problem of the first kind.*
We must find the solution of the equation

$$-Y_{j-1} + CY_j - Y_{j+1} = F_j, \qquad 1 \le j \le N-1, \tag{1}$$

which takes on the following values for $j = 0$ and $j = N$

$$Y_0 = F_0, \qquad Y_N = F_N. \tag{2}$$

Here Y_j is the j^{th} vector of unknowns, F_j is the given right-hand side, and C is a given square matrix.

(2) *Boundary-value problems of the second and third kinds.*
We seek the solution to equation (1) which satisfies the following boundary conditions for $j = 0$ and $j = N$:

$$\begin{aligned} (C + 2\alpha E)Y_0 - 2Y_1 &= F_0, \qquad j = 0, \\ -2Y_{N-1} + (C + 2\beta E)Y_N &= F_N, \qquad j = N, \end{aligned} \tag{3}$$

where $\alpha \ge 0$, $\beta \ge 0$. If $\alpha = \beta = 0$, the formulas (3) give boundary conditions of the second kind. We will also consider mixed boundary conditions, for example, a boundary condition of the first kind for $j = 0$, and a condition of the second or third kind for $j = N$.

(3) *A periodic boundary-value problem.*
We must find the solution of the equation $-Y_{j-1} + CY_j - Y_{j+1} = F_j$ which is periodic, $Y_{N+j} = Y_j$. It is assumed that the right-hand side F_j is also periodic, $F_{N+j} = F_j$. This problem can be formulated in the following equivalent form: find the solution of

$$\begin{aligned} -Y_{j-1} + CY_j - Y_{j+1} &= F_j, \qquad 1 \le j \le N-1, \\ -Y_{N-1} + CY_0 - Y_1 &= F_0, \qquad Y_N = Y_0. \end{aligned} \tag{4}$$

This sort of equation arises from difference schemes for elliptic equations in curvilinear orthogonal coordinate systems: i.e., in cylindrical, polar, and spherical systems.

In addition to the basic vector equation (1) containing one matrix C, we will also consider a boundary-value problem of the first kind for the more general equation

$$\begin{aligned} BY_{j-1} + AY_j - BY_{j+1} &= F_j, \qquad 1 \le j \le N-1, \\ Y_0 &= F_0, \qquad Y_N = F_N \end{aligned} \tag{5}$$

with square matrices A and B. A similar form of problem arises when solving a high-accuracy Dirichlet difference problem for Poisson's equation in a rectangle.

We now formulate requirements on the matrices C, A, and B which guarantee the applicability of the cyclic reduction method for the problems (1)–(5). For problems (1)–(4) we will assume that $(CY, Y) \ge 2(Y, Y)$ for any vector Y, and for problem (5) that $(AY, Y) \ge 2(BY, Y) > 0$. Here the usual vector inner-product is used.

3.1.2 A boundary-value problem of the first kind. We begin our study of the cyclic reduction method with a description of grid boundary-value problems for elliptic equations which can be written in the form of the special vector equations (1)–(5). Suppose that we have introduced the grid $\bar{\omega} = \{x_{ij} = (ih_1, jh_2) \in \bar{G},\ 0 \le i \le M,\ 0 \le j \le N,\ h_1 = l_1/M,\ h_2 = l_2/N\}$ with boundary γ in the rectangle $\bar{G} = \{0 \le x_\alpha \le l_\alpha,\ \alpha = 1, 2\}$, and that we wish to solve a Dirichlet difference problem for Poisson's equation

$$\begin{aligned} y_{\bar{x}_1 x_1} + y_{\bar{x}_2 x_2} &= -\varphi(x) \qquad x \in \omega, \\ y(x) &= g(x), \qquad x \in \gamma. \end{aligned} \tag{6}$$

In Section 2.4 it was shown that problem (6) can be written in the form (1), (2) where Y_j is the vector of dimension $M-1$ whose components are the values of the grid function $y(i, j) = y(x_{ij})$ at the inner nodes of the j^{th} row of the grid $\bar{\omega}$:

$$Y_j = (y(1, j), y(2, j), \ldots, y(M-1, j))^T, \qquad 0 \le j \le N.$$

C is a square matrix of dimension $(M-1) \times (M-1)$ which corresponds to the difference operator Λ, where

$$\begin{aligned} \Lambda y &= 2y - h_2^2 y_{\bar{x}_1 x_1}, \qquad h_1 \le x_1 \le l_1 - h_1, \\ y &= 0, \qquad x_1 = 0, l_1. \end{aligned} \tag{7}$$

The right-hand side F_j is the vector of dimension $M-1$ defined as follows:

(1) for $j = 1, 2, \ldots, N-1$

$$F_j = \left(h_2^2\bar{\varphi}(1,j), h_2^2\varphi(2,j), \ldots, h_2^2\varphi(M-2,j), h_2^2\bar{\varphi}(M-1,j)\right)^T, \quad (8)$$

where

$$\bar{\varphi}(1,j) = \varphi(1,j) + \frac{1}{h_1^2}g(0,j),$$

$$\bar{\varphi}(M-1,j) = \varphi(M-1,j) + \frac{1}{h_1^2}g(M,j);$$

(2) for $j = 0, N$

$$F_j = (g(1,j), g(2,j), \ldots, g(M-1,j))^T. \quad (9)$$

From (7) it follows that the matrix C is a symmetric tridiagonal matrix in this example.

We now consider a more complex difference problem which can also be written in the form of the equations (1), (2). Suppose that it is necessary to find the solution on the grid $\bar{\omega}$ of the Poisson difference equation

$$y_{\bar{x}_1 x_1} + y_{\bar{x}_2 x_2} = -\varphi(x), \qquad x \in \omega, \quad (10)$$

satisfying third- or second-kind boundary conditions on the sides $x_1 = 0$ and $x_1 = l_1$

$$\frac{2}{h_1}y_{x_1} + y_{\bar{x}_2 x_2} = \frac{2}{h_1}\kappa_{-1}y - \bar{\varphi}, \qquad x_1 = 0, \quad (11)$$

$$-\frac{2}{h_1}y_{\bar{x}_1} + y_{\bar{x}_2 x_2} = \frac{2}{h_1}\kappa_{+1}y - \bar{\varphi}, \qquad x_1 = l_1, \quad (12)$$

$$h_2 \le x_2 \le l_2 - h_2$$

and first-kind boundary conditions on the sides $x_2 = 0$, $x_2 = l_2$: $y(x) = g(x)$, $x_2 = 0$, l_2, $0 \le x_1 \le l_1$. In order to be able to write this problem in the form (1), (2) with a matrix C which does not depend on j, it is necessary to assume that $\kappa_{\pm 1} = \text{constant}$.

We now bring this problem into the form (1), (2). To do this we multiply (10)–(12) by $(-h_2^2)$ and write out the difference derivative $y_{\bar{x}_2 x_2}$ at a point for $j = 1, 2, \ldots, N-1$. We obtain the following equations:

(1) for $i = 0$

$$-y(0,j-1) + 2\left[\left(1 + \frac{h_2^2}{h_1}\kappa_{-1}\right)y(0,j) - \frac{h_2^2}{h_1}y_{x_1}(0,j)\right]$$
$$- y(0,j+1) = h_2^2\bar{\varphi}(0,j);$$

(2) for $i = 1, 2, \ldots, M-1$

$$-y(i,j-1) + [2y(i,j) - h_2^2 y_{\bar{x}_1 x_1}(i,j)] - y(i,j+1) = h_2^2 \varphi(i,j);$$

(3) for $i = M$

$$-y(M,j-1) + 2\left[\left(1 + \frac{h_2^2}{h_1}\kappa_{+1}\right) y(M,j) + \frac{h_2^2}{h_1} y_{\bar{x}_1}(M,j)\right] - y(M,j+1) = h_2^2 \bar{\varphi}(M,j).$$

We denote

$$\begin{aligned} Y_j &= (y(0,j), y(1,j), \ldots, y(M,j))^T, \quad 0 \le j \le N, \\ F_j &= \left(h_2^2\bar{\varphi}(0,j), h_2^2\varphi(1,j), \ldots, h_2^2\varphi(M-1,j), h_2^2\bar{\varphi}(M,j)\right)^T, \\ F_j &= (g(0,j), g(1,j), \ldots, g(M,j))^T, \quad j = 0, N. \end{aligned} \tag{13}$$

With this notation, the resulting equations are written in the form (1), (2), where the square matrix C of dimension $(M+1) \times (M+1)$ corresponds to the difference operator Λ:

$$\Lambda y = \begin{cases} 2\left(1 + \dfrac{h_2^2}{h_1}\kappa_{-1}\right) y - \dfrac{2h_2^2}{h_1} y_{x_1}, & x_1 = 0, \\ 2y - h_2^2 y_{\bar{x}_1 x_1}, & h_1 \le x_1 \le l_1 - h_1, \\ 2\left(1 + \dfrac{h_2^2}{h_1}\kappa_{+1}\right) y + \dfrac{2h_2^2}{h_1} y_{\bar{x}_1}, & x_1 = l_1. \end{cases} \tag{14}$$

Here again we have come across a case where C is a tridiagonal matrix. Posing boundary conditions of the third kind (11), (12) on the sides $x_1 = 0, l_1$ in place of boundary conditions of the first kind only leads to a different definition of the operator Λ — in place of (7) we have (14). The form of the equations (1) and the boundary conditions (2) is not changed. If boundary conditions of the first kind $y(x) = g(x)$ is given in place of (11) for $x_1 = 0$, and as before the condition (12) is given for $x_1 = l_1$, then the resulting difference problem also reduces to (1), (2). In this case

$$\begin{aligned} Y_j &= (y(1,j), y(2,j), \ldots, y(M,j))^T, && 0 \le j \le N, \\ F_j &= \left(h_2^2\bar{\varphi}(1,j), h_2^2\varphi(2,j), \ldots, h_2^2\varphi(M-1,j), h_2^2\bar{\varphi}(M,j)\right)^T, && 1 \le j \le N-1, \end{aligned}$$

where

$$\bar{\varphi}(1,j) = \varphi(1,j) + \frac{1}{h_1^2} g(0,j),$$

$\bar{\varphi}(M,j)$ is the change in the corresponding point of the right-hand side $\bar{\varphi}$ in (12), and the square matrix C corresponds to the difference operator Λ, where

$$\Lambda y = \begin{cases} 2y - h_2^2 y_{\bar{x}_1 x_1}, & h_1 \le x_1 \le l_1 - h_1, \\ 2\left(1 + \dfrac{h_2^2}{h_1}\kappa_{+1}\right) y + \dfrac{2h_2^2}{h_1} y_{\bar{x}_1}, & x_1 = l_1. \end{cases} \tag{15}$$

and $y = 0$ for $x_1 = 0$.

If a boundary condition of the first kind is given for $x_1 = l_1$, and the boundary condition of the third kind (11) is given for $x_1 = 0$, then in (1), (2)

$$Y_j = (y(0,j), y(2,j), \ldots, y(M-1,j))^T, \quad 0 \le j \le N,$$

$$F_j = \left(h_2^2\bar{\varphi}(0,j), h_2^2\varphi(1,j), \ldots, h_2^2\varphi(M-2,j), h_2^2\bar{\varphi}(M-1,j)\right)^T, \quad 1 \le j \le N-1,$$

where

$$\bar{\varphi}(M-1,j) = \varphi(M-1,j) + \frac{1}{h_1^2} g(M,j)$$

and the matrix C corresponds to the difference operator Λ, where

$$\Lambda y = \begin{cases} 2\left(1 + \dfrac{h_2^2}{h_1}\kappa_{-1}\right) y - \dfrac{2h_2^2}{h_1} y_{x_1}, & x_1 = 0, \\ 2y - h_2^2 y_{\bar{x}_1 x_1}, & h_1 \le x_1 \le l_1 - h_1, \end{cases} \tag{16}$$

and $y = 0$ for $x_1 = l - 1$.

Thus, we have shown that, if a boundary condition of the first kind is given in the direction x_2, and any combination of first-, second-, or third-kind boundary conditions is given in the direction x_1, then the difference schemes for Poisson's equation in a rectangle can be written in the form of a boundary-value problem of the first kind for the three-point vector equations (1), (2). The matrix C is defined with the aid of the difference operator Λ which, depending on the type of boundary condition on the sides $x_1 = 0$ and $x_1 = l_1$, is given by the formulas (7), (14)–(16).

3.1.3 Other boundary-value problems for difference equations. The type of boundary conditions for equation (1) fully determines the type of boundary conditions for the difference equation (10) on the sides of the rectangle $x_2 = 0$ and $x_2 = l_2$. We have looked at the case where boundary conditions of the first kind were given on these sides.

We will look now at other boundary-value problems for equation (10) which lead to the vector equations (1), (3). Suppose that we are required to find the solution of a *boundary value problem of the third kind* for Poisson's difference equation on the rectangular grid $\bar{\omega}$ defined above. The difference scheme has the following form:

$$y_{\bar{x}_1 x_1} + y_{\bar{x}_2 x_2} = \varphi(x), \qquad x \in \omega, \tag{17}$$

$$\frac{2}{h_1} y_{x_1} + y_{\bar{x}_2 x_2} = \frac{2}{h_1} \kappa_{-1} y - \bar{\varphi}, \quad x_1 = 0, \tag{18}$$

$$-\frac{2}{h_1} y_{\bar{x}_1} + y_{\bar{x}_2 x_2} = \frac{2}{h_1} \kappa_{+1} y - \bar{\varphi}, \quad x_1 = l_1, \ h_2 \le x_2 \le l_2 - h_2,$$

$$y_{\bar{x}_1 x_1} + \frac{2}{h_2} y_{x_2} = \frac{2}{h_2} \kappa_{-2} y - \bar{\varphi}, \quad x_2 = 0, \tag{19}$$

$$y_{\bar{x}_1 x_1} - \frac{2}{h_2} y_{\bar{x}_2} = \frac{2}{h_2} \kappa_{+2} y - \bar{\varphi}, \quad x_2 = l_2, \ h_1 \le x_1 \le l_1 - h_1. \tag{20}$$

At the corners of the grid, the approximation has the special form:

$$\frac{2}{h_1} y_{x_1} + \frac{2}{h_2} y_{x_2} = \left(\frac{2}{h_1} \kappa_{-1} + \frac{2}{h_2} \kappa_{-2} \right) y - \bar{\varphi}, \quad x_1 = 0, \quad x_2 = 0, \tag{21}$$

$$-\frac{2}{h_1} y_{\bar{x}_1} + \frac{2}{h_2} y_{x_2} = \left(\frac{2}{h_1} \kappa_{+1} + \frac{2}{h_2} \kappa_{-1} \right) y - \bar{\varphi}, \quad x_1 = l_1, \quad x_2 = 0, \tag{22}$$

$$\frac{2}{h_1} y_{x_1} - \frac{2}{h_2} y_{\bar{x}_2} = \left(\frac{2}{h_1} \kappa_{-1} + \frac{2}{h_2} \kappa_{+2} \right) y - \bar{\varphi}, \quad x_1 = 0, \quad x_2 = l_2, \tag{23}$$

$$-\frac{2}{h_1} y_{\bar{x}_1} - \frac{2}{h_2} y_{\bar{x}_2} = \left(\frac{2}{h_1} \kappa_{+1} + \frac{2}{h_2} \kappa_{+2} \right) y - \bar{\varphi}, \quad x_1 = l_1, \quad x_2 = l_2, \tag{24}$$

Here it is assumed that $\kappa_{\pm\alpha} = \text{constant}$, $\alpha = 1, 2$.

We will show that problem (17)–(24) reduces to (1), (3). In fact, denoting by Y_j the vector of dimension $M + 1$

$$Y_j = (y(0, j), y(1, j), \ldots, y(M, j))^T, \qquad 0 \le j \le N$$

and defining the right-hand side F_j for $j = 1, 2, \ldots, N - 1$ by the formulas (13), we obtain from (17) and (18), as in the previous section, equation (1) with a matrix C which corresponds to Λ from (14). It remains to show that the conditions (19)–(24) can be written in the form of the boundary conditions (3).

We multiply (19), (21) and (22) by $(-h_2^2)$ and write out the difference derivative y_{x_2} at a point. We obtain:

(1) for $i = 0$

$$2\left[\left(1+\frac{h_2^2}{h_1}\kappa_{-1}\right)y(0,0)-\frac{h_2^2}{h_1}y_{x_1}(0,0)\right]$$
$$+2h_2\kappa_{-2}y(0,0)-2y(0,1)=h_2^2\bar{\varphi}(0,0),$$

(2) for $i = 1,2,\ldots,M-1$

$$[2y(i,0)-h_2^2y_{\bar{x}_1x_1}(i,0)]+2h_2\kappa_{-2}y(i,0)-2y(i,1)=h_2^2\bar{\varphi}(i,0),$$

(3) for $i = M$

$$2\left[\left(1+\frac{h_2^2}{h_1}\kappa_{+1}\right)y(M,0)+\frac{h_2^2}{h_1}y_{\bar{x}_1}(M,0)\right]$$
$$+2h_2\kappa_{-2}y(M,0)-2y(M,1)=h_2^2\bar{\varphi}(M,0).$$

If we denote $\alpha = h_2\kappa_{-2}$, then these equations can be written in vector form

$$(C+2\alpha E)Y_0-2Y_1=F_0, \tag{25}$$

where $F_0=\left(h_2^2\bar{\varphi}(0,0),h_2^2\bar{\varphi}(1,0),\ldots,h_2^2\bar{\varphi}(M,0)\right)^T$.

Analogously, we obtain from (20), (23), and (24) the equation

$$-2Y_{N-1}+(C+2\beta E)Y_N=F_N,$$

where we have denoted $\beta = h_2\kappa_{+2}$ and $F_N = \left(h_2^2\bar{\varphi}(0,N),\ h_2^2\bar{\varphi}(1,N),\ldots,\right.$ $\left.h_2^2\bar{\varphi}(M,N)\right)^T$. Thus, the difference scheme (17)–(24) has been reduced to problem (1), (3).

We look now at the case where some combination of boundary conditions is given on the sides of the rectangle $\bar{G}$. As was remarked above, the problem differs from (18) in the boundary conditions on the sides $x_1 = 0$ and $x_1 = l_1$, but this only has an effect on the definition of the matrix C. If a boundary condition of the first kind is given for $x_2 = 0$, i.e., in place of (19), (21) and (22) we have $y(x) = g(x)$, $x_2 = 0$, then the condition (25) must be changed to the condition $Y_0 = F_0$, where $F_0 = (g(0,0),\ldots,g(M,0))^T$. In this case, the three-point vector boundary-value problem has the form

$$\begin{aligned} -Y_{j-1}+CY_j-Y_{j+1}&=F_j, \qquad 1\le j\le N-1,\\ Y_0&=F_0,\\ -2Y_{N-1}+(C+2\beta E)Y_N&=F_N. \end{aligned} \tag{26}$$

We also obtain an analogous system in the case when a boundary condition of the first kind is given on the side $x_2 = l_2$, and a boundary condition of the

third kind is given on the side $x_2 = 0$. In this case the vector boundary-value problem has the form

$$\begin{aligned} -Y_{j-1} + CY_j - Y_{j+1} &= F_j, \quad 1 \le j \le N-1, \\ (C + 2\alpha E)Y_0 - 2Y_1 &= F_0, \quad Y_N = F_N. \end{aligned} \tag{27}$$

We looked at examples of boundary-value problems for Poisson's difference equation in a rectangle and showed that they correspond to the vector boundary-value problems (1), (2) or (1), (3), or (26), (27) with a corresponding tridiagonal matrix C.

Difference schemes for more complex elliptic equations in both Cartesian and curvilinear orthogonal coordinate systems also lead to such vector boundary-value problems. We will give some examples. In a Cartesian system, the basic boundary-value problems for an elliptic equation are

$$\frac{\partial}{\partial x_1}\left(k_1(x_1)\frac{\partial u}{\partial x_1}\right) + k_2(x_1)\frac{\partial^2 u}{\partial x_2^2} - q(x_1)u = -f(x), \quad x \in G,$$

where the coefficients depend only on one variable. In this case, we can introduce into the rectangle $\bar{G}$ the rectangular grid $\bar{\omega}$ with uniform step h_2 in the direction x_2 and arbitrary non-uniform steps in the direction x_1.

In cylindrical coordinate systems, these examples are boundary-value problems for Poisson's equation in a finite circular cylinder or tube in the presence of axial symmetry:

$$\begin{gathered} \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{\partial^2 u}{\partial z^2} = -f(r,z) \\ 0 \le r_0 < r < R, \quad 0 < z < l. \end{gathered}$$

In this case, an arbitrary non-uniform grid can be introduced in the direction r, but a grid with constant step h_2 is introduced in the direction z.

If we must find the solution of Poisson's equation on the surface of a cylinder, i.e.,

$$\frac{1}{R^2}\frac{\partial^2 u}{\partial \varphi^2} + \frac{\partial^2 u}{\partial z^2} = -f(\varphi, z), \quad 0 \le \varphi \le 2\pi, \quad 0 < z < l,$$

then the corresponding difference problem reduces to the periodic vector boundary-value problem (4), where it is possible to have an arbitrary non-uniform grid in the direction z.

In polar coordinates, admissible examples are difference schemes for Poisson's equation in a circle, a ring, and a circular or ring sector

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 u}{\partial \varphi^2}=-f(r,\varphi),\quad (r,\varphi)\in G.$$

For the circle and the ring, the difference scheme leads to the periodic problem (4), and for the sectors — to the problems (1), (2) or (1), (3). Here it is possible to introduce a non-uniform grid in the direction r.

The difference scheme for Poisson's equation on the surface of a sphere of radius R:

$$\frac{1}{R^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial u}{\partial\theta}\right)+\frac{1}{R^2\sin^2\theta}\frac{\partial^2 u}{\partial\varphi^2}=-f(\varphi,\theta)$$

also leads to the periodic boundary-value problem (4).

3.1.4 A high-accuracy Dirichlet difference problem. We look now at an example of a difference scheme which leads to (5), a more general vector equation than (1). On the rectangular grid $\bar{\omega}=\{x_{ij}=(ih_1,jh_2)\in G,\ 0\le i\le M,\ 0\le j\le N,\ h_1M=l_1,\ h_2N=l_2\}$, we write the Dirichlet difference problem for the high-accuracy Poisson's equation

$$\begin{aligned} y_{\bar{x}_1x_1}+y_{\bar{x}_2x_2}+\frac{h_1^2+h_2^2}{12}y_{\bar{x}_1x_1\bar{x}_2x_2}&=-\varphi(x), && x\in\omega,\\ y(x)&=g(x), && x\in\gamma. \end{aligned} \tag{28}$$

The solution of the difference scheme (28) with a corresponding choice of right-hand side $\varphi(x)$ converges at rate $O(h_1^4+h_2^4)$ to a sufficiently smooth solution of the differential problem if $h_1\ne h_2$, and at rate $O(h^6)$ if $h_1=h_2=h$.

We shall reduce (28) to a boundary-value problem for a vector three-point equation

$$\begin{aligned} -BY_{j-1}+AY_j-BY_{j+1}&=F_j, && 1\le j\le N-1,\\ Y_0=F_0,\qquad Y_N&=F_N. \end{aligned} \tag{29}$$

To do this, it is necessary to multiply (28) by $(-h_2^2)$, write out the difference derivative

$$\left(y+\frac{h_1^2+h_2^2}{12}y_{\bar{x}_1x_1}\right)_{\bar{x}_2x_2}$$

at a point, and use the notation

$$Y_j = (y(1,j), y(2,j), \ldots, y(M-1,j))^T,$$
$$F_j = \left(h_2^2\bar{\varphi}(1,j), h_2^2\varphi(2,j), \ldots, h_2^2\varphi(M-2,j), h_2^2\bar{\varphi}(M-1,j)\right)^T,$$
$$1 \le j \le N-1,$$

where

$$\bar{\varphi}(1,j) = \varphi(1,j) + \frac{1}{h_1^2}\left(g(0,j) + \frac{h_1^2+h_2^2}{12} g_{\bar{x}_2 x_2}(0,j)\right),$$
$$\bar{\varphi}(M-1,j) = \varphi(M-1,j) + \frac{1}{h_1^2}\left(g(M,j) + \frac{h_1^2+h_2^2}{12} g_{\bar{x}_2 x_2}(M,j)\right)$$

and

$$F_j = (g(1,j), g(2,j), \ldots, g(M-1,j))^T, \qquad j = 0, N.$$

In this case, the matrices B and A correspond to the difference operators Λ_1 and Λ, where

$$\Lambda_1 y = y + \frac{h_1^2+h_2^2}{12} y_{\bar{x}_1 x_1}, \qquad h_1 \le x_1 \le l_1 - h_1,$$
$$\Lambda y = 2y - \frac{5h_2^2 - h_1^2}{6} y_{\bar{x}_1 x_1}, \qquad h_1 \le x_1 \le l_1 - h_1,$$

and $y = 0$ for $x_1 = 0$ and $x_1 = l_1$. These matrices are tridiagonal and, as is easily verified, they commute.

The boundary-value problem (29) can be reduced to problem (1), (2). To do this, each of the equations (29) must be multiplied on the left by B^{-1}, if the inverse of matrix B exists. We now find a sufficient condition for the existence of B^{-1}. Cleraly, an inverse to the matrix B exists if the system of linear algebraic equations

$$BY = F \tag{30}$$

has a unique solution for each right-hand side F.

Using the definition of the matrix B, (30) can be written in the form of a difference scheme

$$\Lambda_1 y = y + \frac{h_1^2+h_2^2}{12} y_{\bar{x}_1 x_1} + f, \quad h_1 \le x_1 \le l_1 - h_1, \qquad y(0) = y(l_1) = 0. \tag{31}$$

In Section 2.1 it was shown that, if the scheme (31) satisfies the sufficient conditions for the stability of the elimination method, then the solution of equation (31) exists and is unique for any right-hand side f, and this solution can be found by the elimination method. Writing out the difference derivative $y_{\bar{x}_1 x_1}$ at a point, we write (31) in the form of the scalar three-point equations

$$\begin{aligned} -A_i y_{i-1} + C_i y_i - B_i y_{i+1} &= F_i, \qquad 1 \le i \le M-1, \\ y_0 = 0, \qquad y_M &= 0, \end{aligned} \tag{32}$$

where

$$A_i = B_i = \frac{h_1^2 + h_2^2}{12h_1^2}, \qquad C_i = \frac{h_1^2 + h_2^2}{6h_1^2} - 1.$$

Recall that for (31) the sufficient conditions for the stability of the elimination method have the form $|C_i| \ge |A_i| + |B_i|$, $i = 1, 2, \ldots, M-1$. From these conditions we find that the matrix B has an inverse if the steps for the grid $\bar{\omega}$ satisfy the relation $h_2 \le \sqrt{2}h_1$. If this condition is satisfied, problem (29) can be reduced to problem (1), (2) with $C = B^{-1}A$.

3.2 The cyclic reduction method for a boundary-value problem of the first kind

3.2.1 The odd-even elimination process. We move on now to a description of the cyclic reduction method. We begin with a boundary-value problem of the first kind for three-point vector equations

$$\begin{aligned} -Y_{j-1} + CY_j - Y_{j+1} &= F_j, \qquad 1 \le j \le N-1, \\ Y_0 = F_0, \qquad Y_N &= F_N. \end{aligned} \tag{1}$$

The idea behind the cyclic reduction method for solving the problem (1) consists of sequentially eliminating from equation (1) the unknowns Y_j first with odd indices j, then with indices j which are multiples of 2, then 4, and so forth. Each step of the elimination process reduces the number of unknowns, and if N is a power of 2, i.e. $N = 2^n$, then at the end of the elimination process there remains one equation, from which it is possible to find $Y_{N/2}$. The reverse path of the method involves sequentially finding the unknowns Y_j first with indices j divisible by $N/4$, then $N/8$, $N/16$, and so forth.

Clearly, the cyclic reduction method is a modification of the Gaussian elimination method applied to the problem (1), in which the elimination of the unknowns is carried out in a special order. Recall that, unlike in this method, the elimination of the unknowns is carried out in the natural order in the block elimination method.

Thus, suppose $N = 2^n$, $n > 0$. For convenience we introduce the following notation: $C^{(0)} = C$, $F_j^{(0)} = F_j$, $j = 1, 2, \ldots, N-1$, and we write (1) in the form

$$\begin{aligned} &-Y_{j-1} + C^{(0)}Y_j - Y_{j+1} = F_j^{(0)}, \qquad 1 \le j \le N-1,\ N = 2^n, \\ &\qquad Y_0 = F_0, \qquad Y_N = F_N. \end{aligned} \tag{1'}$$

We look now at the first step of the elimination process. At this step, we eliminate the unknowns Y_j with odd indices j from the even-numbered equations of the system (1'). To do this, we write out three successive equations from (1'):

$$\begin{aligned} &-Y_{j-2} + C^{(0)}Y_{j-1} - Y_j = F_{j-1}^{(0)}, \\ &-Y_{j-1} + C^{(0)}Y_j - Y_{j+1} = F_j^{(0)}, \\ &-Y_j + C^{(0)}Y_{j+1} - Y_{j+2} = F_{j+1}^{(0)}, \qquad j = 2, 4, 6, \ldots, N-2. \end{aligned}$$

We multiply the second equation on the left by $C^{(0)}$ and add together all three of the resulting equations. We then have

$$\begin{aligned} &-Y_{j-2} + C^{(1)}Y_j - Y_{j+2} = F_j^{(1)}, \qquad j = 2, 4, 6, \ldots, N-2, \\ &\qquad Y_0 = F_0, \qquad Y_N = F_N, \end{aligned} \tag{2}$$

where

$$\begin{aligned} &C^{(1)} = [C^{(0)}]^2 - 2E, \\ &F_j^{(1)} = F_{j-1}^{(0)} + C^{(0)}F_j^{(0)} + F_{j+1}^{(0)}, \qquad j = 2, 4, 6, \ldots, N-2. \end{aligned}$$

The system (2) only contains the unknowns Y_j with even indices j, the number of unknowns in (2) is equal to $N/2 - 1$, and if this system has been solved, then the unknowns Y_j with odd indices can be found using (1') from the equations

$$C^{(0)}Y_j = F_j^{(0)} + Y_{j-1} + Y_{j+1}, \qquad j = 1, 3, 5, \ldots, N-1 \tag{3}$$

where the right-hand side is now known.

Thus, the original problem (1') is equivalent to the system (2) and the equations (3), where the structure of the system (2) is analogous to the original system.

At the second step of the elimination process, we eliminate the unknowns with indices j divisible by 2 but not by 4 from the equations of the "reduced" system (2) whose index is divisible by 4. By analogy with the first step, we take three equations from the system (2):

$$\begin{aligned} &-Y_{j-4} + C^{(1)}Y_{j-2} - Y_j = F_{j-2}^{(1)}, \\ &-Y_{j-2} + C^{(1)}Y_j - Y_{j+2} = F_j^{(1)}, \\ &-Y_j + C^{(1)}Y_{j+2} - Y_{j+4} = F_{j+2}^{(1)}, \qquad j = 4, 8, 12, \ldots, N-4, \end{aligned}$$

we multiply the second equation on the left by $C^{(1)}$, and add all three equations together. As a result we obtain a system of $N/4-1$ equations containing the unknowns Y_j with indices divisible by 4:

$$\begin{aligned} &-Y_{j-4} + C^{(2)}Y_j - Y_{j+4} = F_j^{(2)}, \qquad j = 4, 8, 12, \ldots, N-4, \\ &\qquad Y_0 = F_0, \qquad Y_N = F_N; \end{aligned}$$

the equations $C^{(1)}Y_j = F_j^{(1)} + Y_{j-2} + Y_{j+2}$, $j = 2, 6, 10, \ldots, N-2$ are used to find the unknowns with indices divisible by 2 but not by 4, and the equations (3) are used to find the unknowns with odd indices. Here the matrix $C^{(2)}$ and the right-hand sides $F_j^{(2)}$ are defined by the formulas

$$\begin{aligned} &C^{(2)} = [C^{(1)}]^2 - 2E, \\ &F_j^{(2)} = F_{j-2}^{(1)} + C^{(1)}F_j^{(1)} + F_{j+2}^{(1)}, \qquad j = 4, 8, 12, \ldots, N-4. \end{aligned}$$

This process of elimination can be continued. At the end of the l-th step we obtain a reduced system for the unknowns with indices divisible by 2^l:

$$\begin{aligned} &-Y_{j-2^l} + C^{(l)}Y_j - Y_{j+2^l} = F_j^{(l)}, \qquad j = 2^l, 2\cdot 2^l, 3\cdot 2^l, \ldots, N-2^l, \\ &\qquad Y_0 = F_0, \qquad Y_N = F_N, \end{aligned} \tag{4}$$

and a group of equations

$$\begin{aligned} &C^{(k-1)}Y_j = F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ &j = 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N-2^{k-1}, \end{aligned} \tag{5}$$

which we solve sequentially for $k = l, l-1, \ldots, 1$ to find the remaining unknowns. The matrices $C^{(k)}$ and the right-hand sides $F_j^{(k)}$ are found using the

recurrence formulas

$$\begin{aligned} C^{(k)} &= \left[C^{(k-1)}\right]^2 - 2E, \\ F_j^{(k)} &= F_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} F_j^{(k-1)} + F_{j+2^{k-1}}^{(k-1)}, \\ j &= 2^k, 2\cdot 2^k, 3\cdot 2^k, \ldots, N-2^k, \end{aligned} \tag{6}$$

for $k = 1, 2, \ldots$.

From (4) it follows that after the $(n-1)$-st elimination step ($l = n-1$) there remains one equation for $Y_{2^{n-1}} = Y_{N/2}$:

$$C^{(n-1)} Y_j = F_j^{(n-1)} + Y_{j-2^{n-1}} + Y_{j+2^{n-1}} = F_j^{(n-1)} + Y_0 + Y_N, \quad j = 2^{n-1},$$
$$Y_0 = F_0, \qquad Y_N = F_N$$

with a known right-hand side. Joining this equation with (5), we discover that all the unknowns can be found sequentially from the equations

$$\begin{aligned} &C^{(k-1)} Y_j = F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \quad Y_0 = F_0, \quad Y_N = F_N, \\ &j = 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N-2^{k-1}, \quad k = n, n-1, \ldots, 1. \end{aligned} \tag{7}$$

Thus, the formulas (6) and (7) fully describe the cyclic reduction method. The right-hand sides are transformed using the formulas (6), and the solution of the original problem (1) is found from equations (7).

We call this method the cyclic (complete) reduction method since here we sequentially reduce the number of equations in the system to the point where there remains only one equation for $Y_{N/2}$. In the method of incomplete reduction which will be looked at in Chapter 4, only a partial reduction in the order of the system is achieved and the "reduced" system is solved by a special method.

3.2.2 Transformation of the right-hand side and inversion of the matrices. Computing the right-hand side $F_j^{(k)}$ using the recurrence formulas (6) can lead to accumulation of rounding error if the norm of the matrix $C^{(k-1)}$ is greater than one. In addition, the matrices $C^{(k)}$ are, generally speaking, full matrices even when the original matrix $C^{(0)} = C$ is tridiagonal. This essential fact leads to an increase in the volume of computational work when $F_j^{(k)}$ is computed using the formulas (6). For the examples considered in Section 3.1, the norms of the matrices are considerably greater than one, and so the algorithm for the method will be computationally unstable.

In order to get around this difficulty, we will not compute the vectors $F_j^{(k)}$; instead we will compute the vectors $p_j^{(k)}$, which are related to the $F_j^{(k)}$ by the following relations:

$$F_j^{(k)} \equiv \prod_{l=0}^{k-1} C^{(l)} p_j^{(k)} 2^k, \tag{8}$$

where we formally set

$$\prod_{l=0}^{-1} C^{(l)} = E,$$

since $p_j^{(0)} \equiv F_j^{(0)} \equiv F_j$.

We now find recurrence relations for the $p_j^{(k)}$. To do this, we substitute (8) in (6). Taking into account that $C^{(l)}$ is a non-singular matrix for any l, from (6) we obtain

$$2 \prod_{l=0}^{k-1} C^{(l)} p_j^{(k)} = \prod_{l=0}^{k-2} C^{(l)} \left[p_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} p_j^{(k-1)} + p_{j+2^{k-1}}^{(k-1)} \right]$$

or

$$2C^{(k-1)} p_j^{(k)} = p_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} p_j^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}. \tag{9}$$

Denoting $s_j^{(k-1)} = 2p_j^{(k)} - p_j^{(k-1)}$, we obtain from (9) that $p_j^{(k)}$ can be found sequentially from the following formulas:

$$\begin{aligned} &C^{(k-1)} s_j^{(k-1)} = p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, \quad p_j^{(k)} = 0.5 \left(p_j^{(k-1)} + s_j^{(k-1)} \right), \\ &j = 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \ldots, N - 2^k, \quad k = 1, 2, \ldots, n-1, \quad p_j^{(0)} \equiv F_j. \end{aligned} \tag{10}$$

The recurrence relations (10) involve the addition of vectors, the multiplication of a vector by a scalar, and the inversion of the matrices $C^{(k-1)}$.

It remains now to eliminate $F_j^{(k)}$ from the equations (7). Substituting (8) in (7) we obtain

$$\begin{aligned} &C^{(k-1)} Y_j = 2^{k-1} \prod_{l=0}^{k-2} C^{(l)} p_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ &Y_0 = F_0, \quad Y_N = F_N, \\ &j = 2^{k-1}, 3 \cdot 2^{k-1}, \ldots, N - 2^{k-1}, \quad k = n, n-1, \ldots, 1. \end{aligned} \tag{11}$$

Here also it is necessary to invert the matrices $C^{(k-1)}$ but, in addition, the operation of multiplication of a matrix by a vector appears in the right-hand side of (11). In the algorithm examined below, a method of inverting the matrix $C^{(k-1)}$ is used which allows us to avoid the undesirable operation of matrix-vector multiplication, and the realization of (11) reduces to the inversion of matrices and the addition of vectors.

We look now at the question of inverting the matrices $C^{(k-1)}$ defined by the recurrence relations (6)

$$C^{(k)} = \left[C^{(k-1)}\right]^2 - 2E, \qquad k = 1, 2, \ldots, \quad C^{(0)} = C. \tag{12}$$

From (12) it follows that $C^{(k)}$ is a monic polynomial of degree 2^k in the matrix C. This polynomial is a Chebyshev polynomial and can be expressed in the following way:

$$C^{(k)} = 2T_{2^k}\left(\frac{1}{2}C\right), \qquad k = 0, 1, \ldots, \tag{13}$$

where $T_n(x)$ is the Chebyshev polynomial of first kind of degree n (see Section 1.4.2):

$$T_n(x) = \begin{cases} \cos(n \arccos x), & |x| \le 1, \\ \frac{1}{2}[(x + \sqrt{x^2 - 1})^n + (x + \sqrt{x^2 - 1})^{-n}], & |x| \ge 1. \end{cases}$$

In fact, using the properties of the polynomial $T_n(x)$

$$T_{2n}(x) = 2[T_n(x)]^2 - 1, \quad T_1(x) = x,$$

(13) follows directly from (12).

Further, using the relation

$$\prod_{l=0}^{k-2} 2T_{2^l}(x) = U_{2^{k-1}-1}(x),$$

connecting the Chebyshev polynomials of the first kind with the polynomials of the second kind $U_n(x)$, where

$$U_n(x) = \begin{cases} \dfrac{\sin((n+1)\arccos x)}{\sin(\arccos x)}, & |x| \le 1, \\ \dfrac{1}{2\sqrt{x^2 - 1}}[(x + \sqrt{x^2 - 1})^{n+1} - (x + \sqrt{x^2 - 1})^{-(n+1)}], & |x| \ge 1, \end{cases}$$

it is easy to compute the product of the polynomials $C^{(l)}$

$$\prod_{l=0}^{k-2} C^{(l)} = U_{2^{k-1}-1}\left(\frac{1}{2}C\right). \tag{14}$$

Thus, an explicit expression has been obtained for $C^{(k)}$ and $\prod_{l=0}^{k-1} C^{(l)}$.

In the following, we require lemma 6 (see Section 2.4.1). According to lemma 6, any ratio $g_m(x)/f_n(x)$ of polynomials without common roots where $n > m$ and where $f_n(x)$ has simple roots can be expanded in elementary fractions in the following fashion:

$$\frac{g_m(x)}{f_n(x)} = \sum_{l=1}^{n} \frac{a_l}{x - x_l}, \qquad a_l = \frac{g_m(x_l)}{f_n'(x_l)},$$

where x_l are the roots of the polynomial $f_n(x)$.

We shall use lemma 6 to expand the ratios $1/T_n(x)$ and $U_{n-1}(x)/T_n(x)$ in elementary fractions. The roots of the polynomial $T_n(x)$ are known:

$$x_l = \cos\frac{(2l-1)}{2n}\pi, \qquad l = 1, 2, \ldots, n, \tag{15}$$

and at these points the polynomials $U_{n-1}(x)$ take on non-zero values

$$U_{n-1}(x_l) = \frac{\sin(n \arccos x_l)}{\sin(\arccos x_l)} = \frac{(-1)^{l+1}}{\sin\frac{(2l-1)}{2n}\pi}, \qquad l = 1, 2, \ldots, n.$$

Therefore, using the relation $T_n'(x) = nU_{n-1}(x)$, we obtain from lemma 6 the following expansions:

$$\frac{1}{T_n(x)} = \sum_{l=1}^{n} \frac{(-1)^{l+1} \sin\frac{(2l-1)\pi}{2n}}{n(x - x_l)}, \tag{16}$$

$$\frac{U_{n-1}(x)}{T_n(x)} = \sum_{l=1}^{n} \frac{1}{n(x - x_l)}, \tag{17}$$

where x_l is defined in (15). The necessary expansion has been found.

We obtain now an expression for the matrices $\left[C^{(k-1)}\right]^{-1}$ and

$$\left[C^{(k-1)}\right]^{-1} \prod_{l=0}^{k-2} C^{(l)}$$

in terms of the matrix C. Taking into account the expansions for the algebraic polynomials (16), (17), from (13) and (14) we obtain

$$\left[C^{(k-1)}\right]^{-1} = \sum_{l=1}^{2^{k-1}} \alpha_{l,k-1} \left(C - 2\cos\frac{(2l-1)\pi}{2^k}E\right)^{-1},$$

$$\left[C^{(k-1)}\right]^{-1} \prod_{l=0}^{k-2} C^{(l)} = \frac{1}{2^{k-1}} \sum_{l=1}^{2^{k-1}} \left(C - 2\cos\frac{(2l-1)\pi}{2^k}E\right)^{-1}.$$

These relations allow us to write in the following form both the formulas (10):

$$\begin{aligned} s_j^{(k-1)} &= \sum_{l=1}^{2^{k-1}} \alpha_{l,k-1} C_{l,k-1}^{-1} \left(p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}\right), \\ p_j^{(k)} &= 0.5\left(p_j^{(k-1)} + s_j^{(k-1)}\right), \\ p_j^{(0)} &\equiv F_j, \\ j &= 2^k, 2\cdot 2^k, 3\cdot 2^k, \ldots, N-2^k, \quad k = 1,2,\ldots,n-1, \end{aligned} \tag{18}$$

and the formulas (11):

$$\begin{aligned} Y_j &= \sum_{l=1}^{2^{k-1}} C_{l,k-1}^{-1} \left[p_j^{(k-1)} + \alpha_{l,k-1}(Y_{j-2^{k-1}} + Y_{j+2^{k-1}})\right], \\ Y_0 &= F_0, \quad Y_N = F_N, \\ j &= 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N-2^{k-1}, \\ k &= n, n-1, \ldots, 1, \end{aligned} \tag{19}$$

where we have denoted

$$C_{l,k-1} = C - 2\cos\frac{(2l-1)\pi}{2^k}E, \quad \alpha_{l,k-1} = \frac{(-1)^{l+1}}{2^{k-1}}\sin\frac{(2l-1)\pi}{2^k}. \tag{20}$$

Thus, the resulting formulas (18), (19) describe the cyclic reduction method for solving the problem (1). These formulas contain only the operations of addition of vectors, multiplication of a vector by a scalar, and inversion of matrices.

Notice that if C is a tridiagonal matrix, then any matrix $C_{l,k-1}$ will also be tridiagonal. The problem of inverting such matrices was solved in Chapter 2. Further, if the matrix C satisfies the condition $(CY, Y) \geq 2(Y, Y)$, then it follows from (20) that the matrices $C_{l,k}$ will be positive definite and consequently will have bounded inverses. Then from the expansion of $\left[C^{(k-1)}\right]^{-1}$ we obtain that the matrices $C^{(k-1)}$ are non-singular for any $k \geq 1$. Recall that this assumption was used to obtain the formulas (10).

3.2.3 The algorithm for the method. The formulas (18), (19) obtained above serve as a basis for the first algorithm of the method. We shall look first at which intermediate quantities must be computed at which stage and then remembered for subsequent use.

An analysis of the formulas (19) shows that for fixed k, the vectors $p_j^{(k-1)}$ with indices $j = 2^{k-1}, 3 \cdot 2^{k-1}, \ldots, N - 2^{k-1}$ are used to compute Y_j. Any vector $p_j^{(l)}$ with the same index j but with index l less than $k-1$ is auxiliary and is only stored temporarily. Therefore the vectors $p_j^{(k)}$ defined at the k-th stage by (18) can be overwritten on the vectors $p_j^{(k)}$; it is also possible to overwrite the unknowns Y_j computed using (19). The method does not require any auxiliary computer storage — all the vectors $p_j^{(k)}$ can be stored in place and then overwritten by the Y_j.

We shall illustrate the organization of the computations in this algorithm with an example. Suppose $N = 16$ $(n = 4)$. In figure 1 we indicate the sequence of computation and the storing of the vectors $p_j^{(k)}$. A shaded square denotes that for this value of the index k, the vector $p_j^{(k)}$ with corresponding index j is stored for later use. Correspondingly, an unshaded square denotes that $p_j^{(k)}$ is auxiliary and is stored only temporarily. The arrows indicate which vectors $p_j^{(k-1)}$ are used to compute $p_j^{(k)}$.

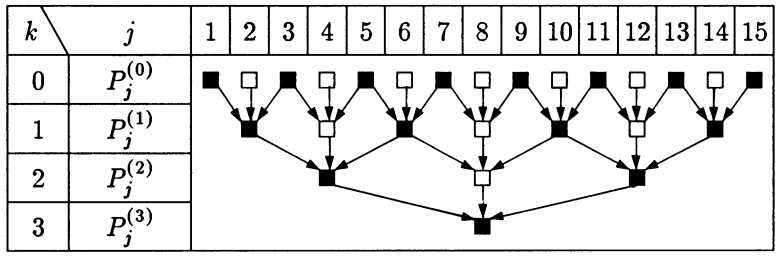

Figure 1.

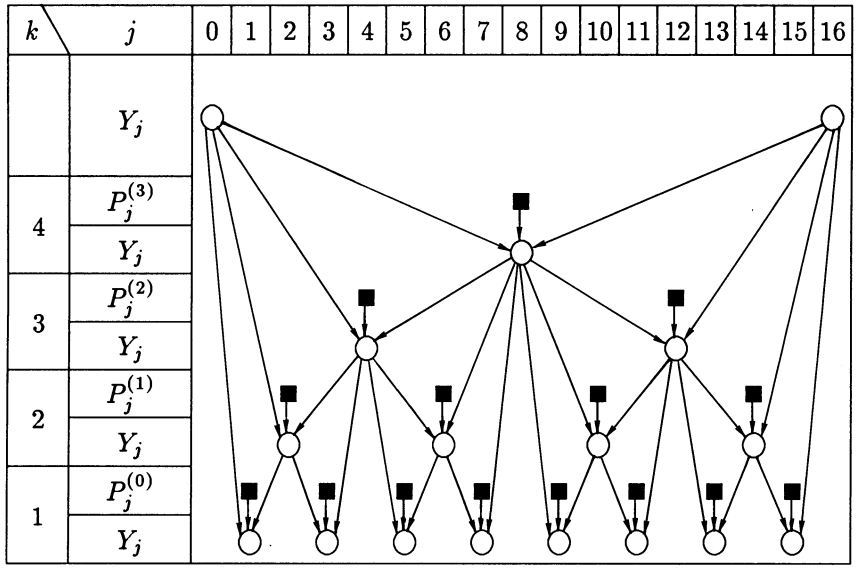

Figure 2.

At the end of the forward path of the method, we will have stored the following vectors $p_j^{(k)}$:

$$p_1^{(0)}, p_2^{(1)}, p_3^{(0)}, p_4^{(2)}, p_5^{(0)}, p_6^{(1)}, p_7^{(0)}, p_8^{(3)}, p_9^{(0)}, p_{10}^{(1)}, p_{11}^{(0)}, p_{12}^{(2)}, p_{13}^{(0)}, p_{14}^{(1)}, p_{15}^{(0)}.$$

They are used to compute Y_j on the reverse path of the method.

In figure 2 we indicate the computation sequence for the unknowns Y_j (symbolically denoted by o). The arrows indicate which Y_j's were found at the preceding step and which $p_j^{(k-1)}$ (symbolically denoted by ■) were used to compute Y_j for a given k.

We move on now to a description of the algorithm for the cyclic reduction method. Using (18), the forward path of the algorithm is realized as follows:

1) Initially set $p_j^{(0)} = F_j$, $j = 1, 2, \ldots, N-1$.
2) For each fixed $k = 1, 2, \ldots, n-1$ and for fixed $j = 2^k, 2 \cdot 2^k, \ldots, N - 2^k$ initially compute and store the vectors

$$\varphi = p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}. \tag{21}$$

Then for $l = 1, 2, \ldots, 2^{k-1}$ solve the equations

$$C_{l,k-1} v_l = \alpha_{l,k-1} \varphi. \tag{22}$$

Find $p_j^{(k)}$ by gradually accumulating and overwriting results in the place of $p_j^{(k-1)}$

$$p_j^{(k)} = 0.5\left(p_j^{(k-1)} + v_1 + v_2 + \cdots + v_{2^{k-1}}\right). \tag{23}$$

Using (19), the reverse path of the method is realized as follows:

1) Initially give values for Y_0 and Y_N: $Y_0 = F_0$, $Y_N = F_N$.
2) For each fixed $k = n, n-1, \ldots, 1$ for fixed $j = 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N - 2^{k-1}$ compute and store the vectors

$$\varphi = Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \qquad \psi = p_j^{(k-1)}. \tag{24}$$

Then for $l = 1, 2, \ldots, 2^{k-1}$ solve the equations

$$C_{l,k-1}v_l = \psi + \alpha_{l,k-1}\varphi. \tag{25}$$

Find the vector of unknowns Y_j by gradually accumulating and overwriting results in the place of $p_j^{(k-1)}$

$$Y_j = v_1 + v_2 + \cdots + v_{2^{k-1}}. \tag{26}$$

We now calculate the number of arithmetic operations required to realize this algorithm. Suppose that the dimension of the vector of unknowns Y_j is M, and let $\mathring{q}$ denote the number of operations required to solve an equation of the form (22) or (25) for a given right-hand side. We will assume that the quantities $\alpha_{l,k}$ have already been found.

We first calculate the number of operations Q_1 for the forward path. For fixed k and j, the computation of the vector φ using the formulas (21) requires $M + \mathring{q}$ operations. Therefore finding all the v_l requires $2^{k-1}(M + \mathring{q})$ operations. The computation of $p_j^{(k)}$ using formula (23) is accomplished at a cost of $2^{k-1}M + M$ operations. Thus, to compute $p_j^{(k)}$ for one k and j requires $M + 2^{k-1}(2M + \mathring{q})$ operations.

Further, for each fixed k it is necessary to compute $N/2^k - 1$ different $p_j^{(k)}$. Consequently, the total number of operations Q_1 required to realize the forward path is equal to

$$\begin{aligned} Q_1 &= \sum_{k=1}^{n-1} \left[M + (2M + \mathring{q})2^{k-1}\right]\left(\frac{N}{2^k} - 1\right) \\ &= (M + 0.5\mathring{q})Nn - (M + \mathring{q})N - M(n-1) + \mathring{q}. \end{aligned} \tag{27}$$

We calculate now the number of operations Q_2 required on the reverse path. For fixed k and j, the computations in the formulas (24) require M operations, to find all the v_l in (25) requires $(2M + \mathring{q})2^{k-1}$ operations, and to compute Y_j from (26) requires $(2^{k-1} - 1)M$ operations. Since the number of different values of j for each fixed k is equal to $N/2^k$, Q_2 is equal to

$$\begin{aligned} Q_2 &= \sum_{k=1}^{n} \left[M + (2M + \mathring{q})2^{k-1} + \left(2^{k-1} - 1\right) M\right] \frac{N}{2^k} \\ &= (1.5M + 0.5\mathring{q})Nn. \end{aligned} \tag{28}$$

Adding (27) and (28) and taking into account that $n = \log_2 N$, we obtain the following estimate for the number of operations for the cyclic reduction method realized using the above algorithm

$$Q = Q_1 + Q_2 = (2.5M + \mathring{q})N \log_2 N - (M + \mathring{q})N - M(n-1) + \mathring{q}. \tag{29}$$

From (29) it follows that, if $q = O(M)$, then $Q = O(MN \log_2 N)$.

3.2.4 The second algorithm of the method. The principle merit of the above algorithm is its minimal storage requirements — it does not require auxiliary memory for the storage of auxiliary information. The cost of this property is an increase in the volume of computational work due to the repeated computation of intermediate quantities. We look now at another algorithm for the method which is characterized by a smaller volume of computational work, but which requires auxiliary storage compared with to the total number of unknowns in the problem.

To construct the second algorithm, we turn to the formulas (6), (7) describing the cyclic reduction method

$$\begin{aligned} C^{(k)} &= \left[C^{(k-1)}\right]^2 - 2E, \\ F_j^{(k)} &= F_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} F_j^{(k-1)} + F_{j+2^{k-1}}^{(k-1)}, \qquad (6') \\ j &= 2^k, 2\cdot 2^k, 3\cdot 2^k, \ldots, N - 2^k, \quad k = 1, 2, \ldots, n-1, \\ C^{(k-1)} Y_j &= F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ Y_0 &= F_0, \quad Y_N = F_N, \qquad (7') \\ j &= 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N - 2^{k-1}, \quad k = n, n-1, \ldots, 1. \end{aligned}$$

Here, as in the first algorithm, the vectors $F_j^{(k)}$ are not directly computed, but instead we define the vectors $p_j^{(k)}$ and $q_j^{(k)}$ which are related to $F_j^{(k)}$ by

the following relations:

$$\begin{aligned} F_j^{(k)} &= C^{(k)} p_j^{(k)} + q_j^{(k)}, \\ j &= 2^k, 2\cdot 2^k, 3\cdot 2^k, \ldots, N-2^k, \quad k = 0,1,\ldots,n-1. \end{aligned} \tag{30}$$

We now find recurrence relations for computing the vectors $p_j^{(k)}$ and $q_j^{(k)}$. Since we have introduced two vectors in place of the one vector $F_j^{(k)}$, there is some arbitrariness in the definition of $p_j^{(k)}$ and $q_j^{(k)}$. We will choose $p_j^{(0)}$ and $q_j^{(0)}$ so that they satisfy the initial condition $F_j^{(0)} \equiv F_j$. To do this we set

$$p_j^{(0)} = 0, \quad q_j^{(0)} = F_j, \qquad j = 1,2,\ldots,N-1. \tag{31}$$

Furhter, substituting (30) in (6′), we obtain

$$\begin{aligned} C^{(k)} p_j^{(k)} + q_j^{(k)} = C^{(k-1)} \left[q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} p_j^{(k-1)} + p_{j+2^{k-1}}^{(k-1)} \right] \\ + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, \quad j = 2^k, 2\cdot 2^k, \ldots, N-2^k, \quad k = 1,2,\ldots,n-1. \end{aligned}$$

Taking

$$q_j^{(k)} = 2p_j^{(k)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)} \tag{32}$$

and taking into account that $C^{(k)} + 2E = [C^{(k-1)}]^2$, we find that

$$C^{(k-1)} p_j^{(k)} = q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} p_j^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}. \tag{33}$$

Here we again assume that $C^{(l)}$ is a non-singular matrix for any l.

Setting $s_j^{(k-1)} = p_j^{(k)} - p_j^{(k-1)}$, we obtain from (31)–(33) the following recurrence relations for computing the vectors $p_j^{(k)}$ and $q_j^{(k)}$:

$$\begin{aligned} C^{(k-1)} s_j^{(k-1)} &= q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, \\ p_j^{(k)} &= p_j^{(k-1)} + s_j^{(k-1)}, \\ q_j^{(k)} &= 2p_j^{(k)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, \\ q_j^{(0)} &\equiv F_j, \quad p_j^{(0)} \equiv 0, \\ j &= 2^k, 2\cdot 2^k, 3\cdot 2^k, \ldots, N-2^k, \\ k &= 1,2,\ldots,n-1. \end{aligned} \tag{34}$$

It remains to eliminate $F_j^{(k-1)}$ from the formulas (7'). Substituting (30) in (7') and setting $t_j^{(k-1)} = Y_j - p_j^{(k-1)}$, we obtain the following formulas for computing Y_j:

$$\begin{aligned} C^{(k-1)} t_j^{(k-1)} &= q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ Y_j &= p_j^{(k-1)} + t_j^{(k-1)}, \\ Y_0 &= F_0, \quad Y_N = F_N, \\ j &= 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N - 2^{k-1}, \\ k &= n, n-1, \ldots, 1. \end{aligned} \tag{35}$$

Thus, we have obtained the formulas (34), (35) which form the basis for the second algorithm for the cyclic reduction method. These formulas contain the operations of addition of vectors and inversion of the matrices $C^{(k-1)}$.

We now consider the question of inverting the matrices $C^{(k-1)}$. As was shown above, the matrix $C^{(k)}$ is a polynomial of degree 2^k in the matrix C and is defined by the formula (13) for the Chebyshev polynomial of the first kind $T_n(x)$:

$$C^{(k)} = 2T_{2^k}\left(\frac{1}{2}C\right),$$

where the coefficient of highest degree is equal to one. Since the roots of the polynomial $T_n(x)$ are known (see (15)), $C^{(k)}$ can be represented in the following factored form:

$$C^{(k)} = \prod_{l=1}^{2^k}\left(C - 2\cos\frac{(2l-1)\pi}{2^{k+1}}E\right), \quad k = 0, 1, \ldots.$$

Using the notation (20), the matrix $C^{(k-1)}$ can be written in the following form:

$$C^{(k-1)} = \prod_{l=1}^{2^{k-1}} C_{l,k-1}, \quad C_{l,k-1} = C - 2\cos\frac{(2l-1)\pi}{2k}E. \tag{36}$$

The factorization (36) allows us to solve easily equations of the form $C^{(k-1)}v = \varphi$ with a given right-hand side φ. The following algorithm solves this problem by sequentially inverting the factors in (36):

$$v_0 = \varphi, \quad C_{l,k-1}v_l = v_{l-1}, \quad l = 1, 2, \ldots, 2^{k-1},$$

where $v = v_{2^{k-1}}$. We will use this algorithm to invert the matrices $C^{(k-1)}$.

We now describe the second algorithm for the cyclic reduction method. The forward path of the method is realized using (34) in the following fashion:

1) Initially define $q_j^{(0)}$: $q_j^{(0)} = F_j$, $j = 1, 2, \ldots, N-1$.

2) (The first step for $k = 1$ is carried out separately using the formulas but taking into account the initial conditions $p_j^{(0)} \equiv 0$.) Solve the equations for $p_j^{(1)}$ and compute $q_j^{(1)}$:

$$\begin{aligned} Cp_j^{(1)} &= q_j^{(0)}, \\ q_j^{(1)} &= 2p_j^{(1)} + q_{j-1}^{(0)} + q_{j+1}^{(0)}, \qquad j = 2, 4, 6, \ldots, N-2. \end{aligned} \tag{37}$$

3) For each fixed $k = 2, 3, \ldots, n-1$ compute and store the vectors

$$v_j^{(0)} = q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, \quad j = 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \ldots, N - 2^k. \tag{38}$$

Then for fixed $l = 1, 2, 3, \ldots, 2^{k-1}$ for each $j = 2^k$, $2 \cdot 2^k$, $3 \cdot 2^k, \ldots$, $N - 2^k$ solve the equations

$$C_{l,k-1} v_j^{(l)} = v_j^{(l-1)} \tag{39}$$

with the same matrix but with different right-hand sides. As a result, the vectors $v_j^{(2^{k-1})}$ have been found (in the formulas (34) these vectors correspond to $s_j^{(k-1)}$). The vectors $p_j^{(k)}$ and $q_j^{(k)}$ are computed using the formulas

$$\begin{aligned} p_j^{(k)} &= p_j^{(k-1)} + v_j^{(2^{k-1})}, \\ q_j^{(k)} &= 2p_j^{(k)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, \\ & j = 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \ldots, N - 2^k. \end{aligned} \tag{40}$$

The reverse path of the method is realized according to (35):

1) Initially give values for Y_0 and Y_N: $Y_0 = F_0$, $Y_N = F_N$.

2) For each fixed $k = n, n-1, \ldots, 2$ compute and store the vectors

$$\begin{aligned} v_j^{(0)} &= q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ & j = 2^{k-1}, 3 \cdot 2^{k-1}, 5 \cdot 2^{k-1}, \ldots, N - 2^{k-1}. \end{aligned} \tag{41}$$

Then for fixed $l = 1, 2, \ldots, 2^{k-1}$ for each $j = 2^{k-1}, 3 \cdot 2^{k-1}, 5 \cdot 2^{k-1}, \ldots, N - 2^{k-1}$ solve the equations

$$C_{l,k-1} v_j^{(l)} = v_j^{(l-1)}. \tag{42}$$

As a result, the vectors $v_j^{(2^{k-1})}$ have been found (in (35) they correspond to the vectors $t_j^{(k-1)}$). Further, compute Y_j from the formula

$$Y_j = p_j^{(k-1)} + v_j^{(2^{k-1})}, \quad j = 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N - 2^{k-1}. \tag{43}$$

3) The final step of the reverse path for $k = 1$ is accomplished by solving the equation

$$CY_j = q_j^{(0)} + Y_{j-1} + Y_{j+1}, \qquad j = 1, 3, 5, \ldots, N-1. \tag{44}$$

Remark on the algorithm. All the new vectors $p_j^{(k)}$ determined using the formulas (37) and (40) are overwritten on the $p_j^{(k-1)}$. All the vectors $v_j^{(l)}$ in the formulas (38), (39), (41), (42), the new vectors $q_j^{(k)}$ defined by the formulas (37), (40), and also the solution Y_j from (43) and (44) are overwritten on the $q_j^{(k-1)}$. Consequently, this algorithm requires 1.5 times as much computer storage as the number of unknowns in the problem.

The reduction in the computational work in this algorithm is achieved by solving a series of problems (39) and (42) for different j with identical matrices $C_{l,k-1}$ (the full computation is only required to solve the first equation in the series; solving each of the subsequent problems requires significantly fewer arithmetic operations). We now count the number of operations for the second algorithm, denoting as before by $\mathring{q}$ the number of operations required to solve an equation of the form (39) or (42) for a given right-hand side, and by $\bar{q}$ the number of operations to solve this equation with a different right-hand side ($\bar{q} < \mathring{q}$).

The number of operations required to realize the forward path is equal to

$$\begin{aligned} Q_1 &= \sum_{k=1}^{n-1} \left\{ 6M\left(\frac{N}{2^k} - 1\right) + \left[\mathring{q} + \bar{q}\left(\frac{N}{2^k} - 2\right)\right] 2^{k-1} \right\} - 3M\left(\frac{N}{2} - 1\right) \\ &= 0.5\bar{q}Nn + (0.5\mathring{q} - 1.5\bar{q} + 4.5M)N - 6Mn - (\mathring{q} - 2\bar{q} + 3M), \end{aligned}$$

and for the reverse path

$$\begin{aligned} Q_2 &= \sum_{k=1}^{n} \left\{ 3M\frac{N}{2^k} + \left[\mathring{q} + \left(\frac{N}{2^k} - 1\right)\bar{q}\right] 2^{k-1} \right\} - \frac{MN}{2} \\ &= 0.5\bar{q}Nn + (\mathring{q} - \bar{q} + 2.5M)N - \mathring{q} + \bar{q} - 3M. \end{aligned}$$

The total number of operations for the second algorithm is equal to

$$Q = Q_1 + Q_2 = \bar{q} N \log_2 N + (1.5\mathring{q} - 2.5\bar{q} + 7M)N - 6Mn - 2\mathring{q} + 3\bar{q} - 6M. \quad (45)$$

From the estimate (45) it follows that, if $\mathring{q} = O(M)$, then $\bar{q} = O(M)$ and $Q = O(MN \log_2 N)$. Here the coefficient of the principle term $MN \log_2 N$ is less than in the estimate (29), since $\bar{q} < \mathring{q}$.

We now quickly examine one peculiarity of the second algorithm. In the first algorithm the matrices $C^{(k-1)}$ are inverted by inverting the factors $C_{l,k-1}$ and sequentially summing the results, but in the second algorithm the factors are sequentially inverted and the result is obtained after inverting the last factor. From the point of view of the actual computation where rounding errors have an effect, the order in which the factors $C_{l,k-1}$ are inverted is significant in the second algorithm. We will come across an analogous situation in Chapter 6 when we study the Chebyshev iterative method.

It is possible to recommend the following order for inverting the matrices $C_{l,k-1}$. The matrix $C^{(k-1)}$ is placed in correspondence with the vector $\theta_{2^{k-1}}$ of dimension 2^{k-1} whose components are the integers 1 through 2^{k-1}. Suppose

$$\theta_{2^{k-1}} = \{\theta_{2^{k-1}}(1), \theta_{2^{k-1}}(2), \ldots, \theta_{2^{k-1}}(2^{k-1})\},$$

i.e. the l-th element of the vector $\theta_{2^{k-1}}$ is denoted by $\theta_{2^{k-1}}(l)$. The number $\theta_{2^{k-1}}(l)$ determines the order for inverting the matrices $C_{l,k-1}$.

The vector $\theta_{2^{k-1}}$ is constructed recursively. Let $\theta_2 = \{2, 1\}$. Then the process of doubling the dimension of the vector is described by the following formulas:

$$\begin{aligned}
\theta_{2m} = \{&\theta_{2m}(4i-3) = \theta_m(2i-1),\\
&\theta_{2m}(4i-2) = \theta_m(2i-1) + m,\\
&\theta_{2m}(4i-1) = \theta_m(2i) + m,\\
&\theta_{2m}(4i) = \theta_m(2i),\\
&i = 1, 2, \ldots, m/2\},\\
&m = 2, 4, 8, \ldots.
\end{aligned}$$

For example: $\theta_{16} = \{2, 10, 14, 6, 8, 16, 12, 4, 3, 11, 15, 7, 5, 13, 9, 1\}$ and consequently the matrix $C_{6,16}$ will be inverted sixteenth and the matrix $C_{12,16}$ seventh.

3.3 Sample applications of the method

3.3.1 A Dirichlet difference problem for Poisson's equation in a rectangle. We will now use the cyclic reduction method constructed above to find the solution to a Dirichlet difference problem for Poisson's equation in a rectangle. As was shown earlier, the difference problem

$$\begin{aligned} y_{\bar{x}_1 x_1} + y_{\bar{x}_2 x_2} &= -\varphi(x), \qquad x \in \omega, \\ y(x) &= g(x), \qquad x \in \gamma, \end{aligned}$$

defined on the rectangular grid $\bar{\omega} = \{x_{ij} = (ih_1, jh_2),\ 0 \le i \le M,\ 0 \le j \le N,\ h_1 M = l_1,\ h_2 N = l_2\}$ can be described in the form of a boundary-value problem of the first kind for the three-point vector equations

$$\begin{aligned} -Y_{j-1} + CY_j - Y_{j+1} &= F_j, \qquad 1 \le j \le N-1, \\ Y_0 = F_0, \quad Y_N &= F_N. \end{aligned} \tag{1}$$

Here

$$Y_j = (y(1,j), y(2,j), \ldots, y(M-1,j)), \qquad 0 \le j \le N,$$

is the vector of unknowns, the components of which are the values of the grid function $y(i,j)$ in the j-th row of the grid

$$\begin{gathered} F_j = \left(h_2^2 \bar{\varphi}(1,j), h_2^2 \varphi(2,j), \ldots, h_2^2 \varphi(M-2,j), h_2^2 \bar{\varphi}(M-1,j)\right), \\ 1 \le j \le N-1, \\ F_j = (g(1,j), g(2,j), \ldots, g(M-1,j)), \qquad j = 0, N, \end{gathered}$$

where

$$\begin{aligned} \bar{\varphi}(1,j) &= \varphi(1,j) + \frac{1}{h_1^2} g(0,j), \\ \bar{\varphi}(M-1,j) &= \varphi(M-1,j) + \frac{1}{h_1^2} g(M,j). \end{aligned}$$

The square matrix C corresponds to the difference operator Λ where

$$\begin{aligned} \Lambda y &= 2y - h_2^2 y_{\bar{x}_1 x_1}, \qquad h_1 \le x_1 \le l_1 - h_1, \\ Y &= 0, \qquad x_1 = 0, l_1, \end{aligned}$$

so that

$$CY_j = (\Lambda y(1,j), \Lambda y(2,j), \ldots, \Lambda y(M-1,j)).$$

The problem (1) can be solved by either of the two cyclic reduction algorithms developed above. The basic step in these algorithms is the solution of equations of the form

$$C_{l,k-1}V = F, \qquad C_{l,k-1} = C - 2\cos\frac{(2l-1)\pi}{2^k}E \tag{2}$$

with a given right-hand side F. Here V is the vector of unknowns, $V = (v(1), v(2), \ldots, v(M-1))$, of dimension $M-1$ (for simplicity, the index of V and F is dropped).

Recall that the number of operations used to solve (1) using the first algorithm is determined by the number of operations $\mathring{q}$ required to solve equation (2) (see (29), Section 3.2.3); for the second algorithm, it is determined by the number of auxiliary operations $\bar{q}$ required to solve equation (2), but with a different right-hand side (see (45), Section 3.2.4).

For this example, we derive a method for solving equation (2) and estimate $\mathring{q}$ and $\bar{q}$. From the definition of the matrix C it follows that solving equation (2) is equivalent to solving the following difference problem:

$$\begin{aligned} &2\left(1 - \cos\frac{(2l-1)\pi}{2^k}\right)v - h_2^2 v_{\bar{x}_1 x_1} = f(i), \qquad 1 \le i \le M-1, \\ &v(0) = v(M) = 0, \end{aligned} \tag{3}$$

where $f(i) = f_i$ is the i-th component of the vector F. Writing out the difference derivative $v_{\bar{x}_1 x_1}$ at a point, we write (3) in the form of the usual three-point difference equation for scalar unknowns $v(i) = v_i$:

$$\begin{aligned} &-v_{i-1} + av_i - v_{i+1} = bf_i, \qquad 1 \le i \le M-1, \\ &v_0 = v_M = 0, \end{aligned} \tag{4}$$

where

$$a = 2\left[1 + b\left(1 - \cos\frac{(2l-1)\pi}{2^k}\right)\right], \qquad b = \frac{h_1^2}{h_2^2}.$$

The problem (4) is a special case of the three-point boundary-value problems which were solved in Chapter 2. It was shown that the elimination method was an effective method for solving problems of the form (4). We now state the computational formulas of the elimination method for the problem (4):

$$\begin{aligned} &\alpha_{i+1} = 1/(a - \alpha_i), && i = 1, 2, \ldots, M-1, && \alpha_1 = 0, \\ &\beta_{i+1} = (bf_i + \beta_i)\alpha_{i+1}, && i = 1, 2, \ldots, M-1, && \beta_1 = 0, \\ &v_i = \alpha_{i+1}v_{i+1} + \beta_{i+1}, && i = M-1, M-2, \ldots, 1, && v_M = 0. \end{aligned}$$

From these formulas it follows that (4), and in turn (2), can be solved in $\mathring{q} = 7(M-1)$ operations if a and b are given. In order to solve (2) with a different right-hand side F it is not necessary to recompute the elimination coefficients α_i, and thus the auxiliary number of operations $\bar{q}$ is equal to $\bar{q} = 5(M-1)$. These operations are expended to compute β_i and to find the solution v_i. Notice that the elimination method for (4) will be stable since the sufficient conditions for stability with respect to rounding errors are satisfied, i.e. $a \geq 2$.

Substituting $\mathring{q}$ in the estimate (29), Section 3.2.3 for the number of operations for the first algorithm, we obtain, retaining the principal terms, that $Q^{(1)} \approx 9.5MN\log_2 N - 8MN$. For the second algorithm, we obtain from the estimate (45), Section 3.2.4 the following estimate for the number of operations: $Q^{(2)} \approx 5MN\log_2 + 5MN$. Thus, for each of the algorithms considered, the number of operations for the cyclic reduction method applied to a Dirichlet difference problem for Poisson's equations in a rectangle is $O(MN\log_2 N)$, and the second algorithm requires fewer arithmetic operations. For example, if $M = N = 64$, we obtain $Q^{(1)} \approx 1.4Q^{(2)}$ and if $M = N = 128$, $Q^{(1)} \approx 1.46Q^{(2)}$.

We will not state the computational formulas of the algorithm for this difference problem since at the vector level they are similar to those described in Section 3.2.

In Section 3.1.2, various difference boundary-value problems were stated which reduced to the problem (1). They differ from this Dirichlet boundary-value problem on the sides of the rectangle, $x_1 = 0$ and $x_1 = l_1$, and produce a different matrix C. So for the problem (10)–(12), Section 3.1.2 with second- or third-kind boundary conditions for $x_1 = 0, l_1$, the equation (2) is equivalent to the difference problem

$$
\begin{aligned}
&2\left(1 - \cos\frac{(2l-1)\pi}{2^k}\right)v - h_2^2 v_{\bar{x}_1 x_1} = f, \qquad 1 \leq i \leq M-1,\\
&2\left(1 + \frac{h_2^2}{h_1}\kappa_{-1} - \cos\frac{(2l-1)\pi}{2^k}\right)v - \frac{2h_2^2}{h_1}v_{x_1} = f, \qquad i = 0,\\
&2\left(1 + \frac{h_2^2}{h_1}\kappa_{+1} - \cos\frac{(2l-1)\pi}{2^k}\right)v + \frac{2h_2^2}{h_1}v_{\bar{x}_1} = f, \qquad i = M.
\end{aligned}
$$

This problem in the usual three-point form is

$$
\begin{aligned}
&-v_{i-1} + av_i - v_{i+1} = bf_i, \qquad 1 \leq i \leq M-1,\\
&v_0 = \bar{\kappa}_1 v_1 + \mu_1,\\
&v_M = \bar{\kappa}_2 v_{M-1} + \mu_2,
\end{aligned}
\tag{5}
$$

where

$$\bar{\kappa}_1 = \frac{2}{a + 2h_1\kappa_{-1}}, \quad \bar{\kappa}_2 = \frac{2}{a + 2h_1\kappa_{+1}}, \quad \mu_1 = \frac{bf_0}{a + 2h_1\kappa_{-1}}, \quad \mu_2 = \frac{bf_M}{a + 2h_1\kappa_{+1}},$$

and a and b are defined above.

Since $a > 2$ and $\kappa_{\pm 1} \geq 0$, $0 < \bar{\kappa}_1 < 1$ and $0 < \bar{\kappa}_2 < 1$, and the elimination method for solving (5) will also be stable, and the cyclic reduction algorithm will in this case require $O(MN \log_2 N)$ arithmetic operations.

3.3.2 A high-accuracy Dirichlet difference problem. In Section 3.1.4 we transformed a high-accuracy Dirichlet problem for Poisson's equation

$$y_{\bar{x}_1 x_1} + y_{\bar{x}_2 x_2} + \frac{h_1^2 + h_2^2}{12} y_{\bar{x}_1 x_1 \bar{x}_2 x_2} = -\varphi(x), \qquad x \in \omega,$$
$$y(x) = g(x), \qquad x \in \gamma,$$

to a boundary-value problem of the first kind for the unreduced three-point vector equation

$$\begin{aligned} &-BY_{j-1} + AY_j - BY_{j+1} = F_j, \qquad 1 \leq j \leq N-1, \\ &Y_0 = F_0, \qquad Y_N = F_N. \end{aligned} \tag{6}$$

The square matrices B and A of dimension $(M-1) \times (M-1)$ correspond to the difference operators Λ_1 and Λ, where

$$\Lambda_1 y = y + \frac{h_1^2 + h_2^2}{12} y_{\bar{x}_1 x_1}, \qquad h_1 \leq x_1 \leq l_1 - h_1,$$
$$\Lambda y = 2y - \frac{5h_2^2 - h_1^2}{6} y_{\bar{x}_1 x_1}, \qquad h_1 \leq x_1 \leq l_1 - h_1$$

and $y = 0$ for $x_1 = 0$ and $x_1 = l_1$.

It was shown that, if the condition $h_2 \leq \sqrt{2} h_1$ is satisfied, then (6) can be reduced to the standard form

$$\begin{aligned} &-Y_{j-1} + CY_j - Y_{j+1} = \Phi_j, \qquad 1 \leq j \leq N-1, \\ &Y_0 = \Phi_0, \qquad Y_N = \Phi_N, \end{aligned} \tag{7}$$

where $C = B^{-1}A$, $\Phi_j = B^{-1}F_j$, $1 \le j \le N-1$ and $\Phi_j = F_j$ for $j = 0, N$. In addition, it was remarked that the matrices A and B commute.

To solve (7), we will use the first algorithm of the method. Since the matrix $C_{l,k-1}$ can be written in the form

$$C_{l,k-1} = C - 2\cos\frac{(2l-1)\pi}{2^k}E = B^{-1}\left(A - 2\cos\frac{(2l-1)\pi}{2^k}B\right),$$

the formulas (18), (19), Section 3.2 which define the first algorithm take the following form:

$$s_j^{(k-1)} = \sum_{l=1}^{2^{k-1}} \alpha_{l,k-1}\left(A - 2\cos\frac{(2l-1)\pi}{2^k}B\right)^{-1} B\left(p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}\right),$$

$$p_j^{(k)} = 0.5\left(p_j^{(k-1)} + s_j^{(k-1)}\right),$$

$$j = 2^k, 2\cdot 2^k, \ldots, N-2^k, \quad k = 1,2,\ldots,n-1,$$

$$Bp_j^{(0)} \equiv F_j,$$

$$Y_j = \sum_{l=1}^{2^{k-1}}\left(A - 2\cos\frac{(2l-1)\pi}{2^k}B\right)^{-1} B\left[p_j^{(k-1)} + \alpha_{l,k-1}(Y_{j-2^{k-1}} + Y_{j+2^{k-1}})\right],$$

$$Y_0 = F_0,\ Y_N = F_N,\ j = 2^{k-1}, 3\cdot 2^{k-1}, \ldots, N-2^{k-1},$$

$$k = n, n-1, \ldots, 1.$$

In order to avoid inverting the matrices to find $p_j^{(0)}$ and multiplying $p_j^{(k-1)}$ by the matrix B to compute Y_j, we set $\bar{p}_j^{(k)} = Bp_j^{(k)}$, $\bar{s}_j^{(k)} = Bs_j^{(k)}$. Then using the commutativity of the matrices A and B, and consequently of the matrices $(A - 2\cos\frac{(2l-1)\pi}{2^k}B)^{-1}$ and B, the formulas written above take the form (where the overline on $\bar{p}_j^{(k)}$ and $\bar{s}_j^{(k)}$ has been dropped):

$$s_j^{(k-1)} = \sum_{l=1}^{2^{k-1}} \alpha_{l,k-1}\left(A - 2\cos\frac{(2l-1)\pi}{2^k}B\right)^{-1} B\left(p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}\right),$$

$$p_j^{(k)} = 0.5\left(p_j^{(k-1)} + s_j^{(k-1)}\right),$$

$$j = 2^k, 2\cdot 2^k, \ldots, N-2^k, k = 1,2,\ldots,n-1,$$

$$p_j^{(0)} \equiv F_j,$$

$$Y_j = \sum_{l=1}^{2^{k-1}} \left(A - 2\cos\frac{(2l-1)\pi}{2^k} B\right)^{-1} \left[p_j^{(k-1)} + \alpha_{l,k-1} B(Y_{j-2^{k-1}} + Y_{j+2^{k-1}})\right],$$

$$Y_0 = F_0, \quad Y_N = F_N,$$

$$j = 2^{k-1}, 3\cdot 2^{k-1}, \ldots, N - 2^{k-1}, \; k = n, n-1, \ldots, 1.$$

The resulting formulas give rise to the following changes in the first algorithm: formula (21) Section 3.2 changes to

$$\varphi = B\left(p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}\right),$$

and in place of equation (22), we solve the equations

$$\left(A - 2\cos\frac{(2l-1)\pi}{2^k} B\right) v_l = \alpha_{l,k-1}\varphi$$

with the computed φ. Analogously, (24) is changed to

$$\varphi = B(Y_{j-2^{k-1}} + Y_{j+2^{k-1}}), \quad \psi = p_j^{(k-1)}$$

and in place of (25), we solve the equation

$$\left(A - 2\cos\frac{(2l-1)\pi}{2^k} B\right) v_l = \psi + \alpha_{l,k-1}\varphi.$$

Consequently, for this problem the basic step of the algorithm is the solution of equations of the form

$$\left(A - 2\cos\frac{(2l-1)\pi}{2^k} B\right) V = F \tag{8}$$

with a given right-hand side F. Using the definition of the matrices A and B with the aid of the difference operations Λ and Λ_1, we obtain that (8) is equivalent to finding the solution of the following difference problem

$$2\left(1 - \cos\frac{(2l-1)\pi}{2^k}\right) v - \left(\frac{5h_2^2 - h_1^2}{6} + \frac{h_1^2 + h_2^2}{6}\cos\frac{(2l-1)\pi}{2^k}\right) v_{\bar{x}_1 x_1} = f,$$

$$1 \le i \le M-1, \quad v_0 = v_M = 0. \tag{9}$$

Writing out this equation at a point, we obtain a boundary-value problem of the first kind for the scalar three-point equation

$$\begin{gathered} -v_{i-1} + av_i - v_{i+1} = bf_i, \qquad 1 \le i \le M-1, \\ v_0 = v_M = 0, \end{gathered} \tag{10}$$

where

$$a = 2\left[1 + b\left(1 - \cos\frac{(2l-1)\pi}{2^k}\right)\right],$$

$$b = \frac{6h_1^2}{5h_2^2 - h_1^2 + (h_1^2 + h_2^2)\cos\frac{(2l-1)\pi}{2^k}}$$

The difference problem (10) can be solved by the elimination method, which will be numerically stable if the condition $|a| \ge 2$ is satisfied. We will show that for any h_1 and h_2 this condition is satisfied. In fact, if h_1 and h_2 are such that

$$\frac{h_2^2}{h_1^2} \ge \frac{1 - \cos\dfrac{(2l-1)\pi}{2^k}}{5 + \cos\dfrac{(2l-1)\pi}{2^k}}, \tag{11}$$

then $0 < b \le \infty$ and consequently, $a > 2$. Notice that if equality holds in (11) the coefficient for $v_{\bar{x}_1 x_1}$ in (9) reduces to zero, and v can be found explicitly from (9).

If (11) is not satisfied, then

$$b < -6 \Big/ \left(1 - \cos\frac{(2l-1)\pi}{2^k}\right),$$

and consequently, $a < -10$. The result is proved.

Thus, a high-accuracy Dirichlet difference problem can be solved by the cyclic reduction method in $O(MN\log_2 N)$ arithmetic operations.

3.4 The cyclic reduction method for other boundary-value problems

3.4.1 A boundary-value problem of the second kind. Above we studied the use of the cyclic reduction method for a boundary-value problem of the first kind for three-point vector equations. We will begin studying the use of the method for more complex boundary conditions by considering a *boundary-value problem of the second kind.* Suppose we must find the solution of the

following problem:

$$\begin{aligned} CY_0 - 2Y_1 &= F_0, && j = 0, \\ -Y_{j-1} + CY_j - Y_{j+1} &= F_j, && 1 \le j \le N-1, \\ -2Y_{N-1} + CY_N &= F_N, && j = N, \end{aligned} \tag{1}$$

where $N = 2^n$, $n > 0$.

The process of sequentially eliminating the unknowns in (1) is realized in the same way as in the case of first-kind boundary conditions. Namely, for even j we will have the equations

$$-Y_{j-2} + C^{(1)}Y_j - Y_{j-2} = F_j^{(1)}, \qquad j = 2,4,6,\ldots,N-2, \tag{2}$$

and for odd j, the equations

$$C^{(0)}Y_j = F_j^{(0)} + Y_{j-1} + Y_{j+1}, \qquad j = 1,3,5,\ldots,N-1, \tag{3}$$

where, as before, we denote

$$\begin{aligned} &F_j^{(1)} = F_{j-1}^{(0)} + C^{(0)}F_j^{(0)} + F_{j+1}^{(0)}, \quad C^{(1)} = [C^{(0)}]^2 - 2E, \\ &C^{(0)} = C, \qquad F_j^{(0)} \equiv F_j. \end{aligned}$$

Only the equations of the system (1) corresponding to $j = 0$ and $j = N$ remain untransformed. We now eliminate from these equations the unknowns Y_j for odd j. For this we use the two neighboring equations. We write out the equations for $j = 0$ and $j = 1$:

$$C^{(0)}Y_0 - 2Y_1 = F_0^{(0)}, \qquad -Y_0 + C^{(0)}Y_1 - Y_2 = F_1^{(0)}.$$

Multiply the first equation on the left by $C^{(0)}$, and the second by 2, and add the two resulting equations to find

$$C^{(1)}Y_0 - 2Y_2 = F_0^{(1)}, \tag{4}$$

where $F_0^{(1)} = C^{(0)}F_0^{(0)} + 2F_1^{(0)}$. Analogously we obtain the equation

$$-2Y_{N-2} + C^{(1)}Y_N = F_N^{(1)}, \tag{5}$$

where $F_N^{(1)} = 2F_{N-1}^{(0)} + C^{(0)}F_N^{(0)}$.

Combining (2), (4), and (5), we obtain a "reduced" full system of equations for the unknowns with even index j, having a structure analogous to (1):

$$\begin{aligned} C^{(1)}Y_0 - 2Y_2 &= F_0^{(1)}, && j = 0, \\ -Y_{j-2} + C^{(1)}Y_j - Y_{j+2} &= F_j^{(1)}, && j = 2,4,6,\ldots,N-2, \\ -2Y_{N-2} + C^{(1)}Y_N &= F_N^{(1)}, && j = N, \end{aligned}$$

and a group of equations (3) for the unknowns with odd index j.

Continuing the elimination process further, after the n-th elimination step we obtain a system for Y_0 and Y_N:

$$C^{(n)}Y_0 - 2Y_N = F_0^{(n)}, \qquad -2Y_0 + C^{(n)}Y_N = F_N^{(n)} \tag{6}$$

and equations for determining the remaining unknowns:

$$\begin{gathered} C^{(k-1)}Y_j = F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ j = 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N-2^{k-1}, \quad k = n, n-1, \ldots, 1, \end{gathered} \tag{7}$$

where $F_i^{(k)}$ and $C^{(k)}$ are defined recursively for $k = 1,2,\ldots,n$:

$$\begin{aligned} F_0^{(k)} &= C^{(k-1)}F_0^{(k-1)} + 2F_{2^{k-1}}^{(k-1)}, \\ F_j^{(k)} &= F_{j-2^{k-1}}^{(k-1)} + C^{(k-1)}F_j^{(k-1)} + F_{j+2^{k-1}}^{(k-1)}, \\ & \qquad j = 2^k, 2\cdot 2^k, 3\cdot 2^k, \ldots, N-2^k, \\ F_N^{(k)} &= 2F_{N-2^{k-1}}^{(k-1)} + C^{(k-1)}F_N^{(k-1)}, \\ C^{(k)} &= [C^{(k-1)}]^2 - 2E. \end{aligned} \tag{8}$$

Thus, it is necessary to solve the system (6) and then sequentially find all the remaining unknowns from (7).

Here, as in the second algorithm for cyclic reduction applied to a boundary-value problem of the first kind, in place of the vectors $F_j^{(k)}$ we will define the vectors $p_j^{(k)}$ and $q_j^{(k)}$ connected with $F_i^{(k)}$ by the relations

$$\begin{gathered} F_j^{(k)} = C^{(k)}p_j^{(k)} + q_j^{(k)}, \\ j = 0, 2^k, 2\cdot 2^k, 3\cdot 2^k, \ldots, N-2^k, N, \quad k = 0,1,\ldots,n. \end{gathered} \tag{9}$$

From (8) we find, as before, that $p_j^{(k)}$ and $q_j^{(k)}$ can be found for $j \neq 0, N$ from the formulas

$$\begin{aligned}
&C^{(k-1)} s_j^{(k-1)} = q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, \\
&p_j^{(k)} = p_j^{(k-1)} + s_j^{(k-1)}, \\
&q_j^{(k)} = 2p_j^{(k)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, \\
&j = 2^k, 2 \cdot 2^k, \ldots, N - 2^k, \quad k = 1, 2, \ldots, n-1, \\
&q_j^{(0)} \equiv F_j, \quad p_j^{(0)} \equiv 0.
\end{aligned} \tag{10}$$

We will now find the formulas for $p_j^{(k)}$ and $q_j^{(k)}$ for $j = 0, N$. Substituting (9) with $j = 0$ in (8) for $F_0^{(k)}$, we obtain

$$C^{(k)} p_0^{(k)} + q_0^{(k)} = C^{(k-1)} \left[q_0^{(k-1)} + 2p_{2^{k-1}}^{(k-1)} + C^{(k-1)} p_0^{(k-1)} \right] + 2q_{2^{k-1}}^{(k-1)}.$$

Choosing $q_0^{(k)} = 2p_0^{(k)} + 2q_{2^{k-1}}^{(k-1)}$ and taking into account (12), Section 3.2.1, we find an equation for $p_0^{(k)}$

$$C^{(k-1)} p_0^{(k)} = C^{(k-1)} p_0^{(k-1)} + q_0^{(k-1)} + 2p_{2^{k-1}}^{(k-1)}.$$

Thus, the vectors $p_0^{(k)}$ and $q_0^{(k)}$ can be found from the following recurrence relations:

$$\begin{aligned}
&C^{(k-1)} s_0^{(k-1)} = q_0^{(k-1)} + 2p_{2^{k-1}}^{(k-1)}, \\
&p_0^{(k)} = p_0^{(k-1)} + s_0^{(k-1)}, \\
&q_0^{(k)} = 2p_0^{(k)} + 2q_{2^{k-1}}^{(k-1)}, \quad k = 1, 2, \ldots, n, \\
&q_0^{(0)} = F_0, \quad p_0^{(0)} = 0.
\end{aligned} \tag{11}$$

The formulas for $p_N^{(k)}$ and $q_N^{(k)}$ are obtained analogously:

$$\begin{aligned}
&C^{(k-1)} s_N^{(k-1)} = q_N^{(k-1)} + 2p_{N-2^{k-1}}^{(k-1)}, \\
&p_N^{(k)} = p_N^{(k-1)} + s_N^{(k-1)}, \\
&q_N^{(k)} = 2p_N^{(k)} + 2q_{N-2^{k-1}}^{(k-1)}, \quad k = 1, 2, \ldots, n, \\
&q_N^{(0)} = F_N, \quad p_N^{(0)} = 0.
\end{aligned} \tag{12}$$

Thus, the formulas (10)–(12) enable us to fully determine all the necessary vectors $p_j^{(k)}$ and $q_j^{(k)}$. We must still eliminate $F_j^{(k)}$ from (6) and (7). Substituting (9) in (7), we obtain the following formulas for computing Y_j:

$$\begin{aligned} C^{(k-1)}t_j^{(k-1)} &= q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ Y_j &= p_j^{(k-1)} + t_j^{(k-1)}, \\ j &= 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N-2^{k-1}, \\ k &= n, n-1, \ldots, 1. \end{aligned} \tag{13}$$

It remains to find Y_0 and Y_N from (6). But it was noted earlier that, from (11) and (12) for $k = n$, it follows that

$$q_0^{(n)} = 2p_0^{(n)} + 2q_{2^{n-1}}^{(n-1)}, \quad q_N^{(n)} = 2p_N^{(n)} + 2q_{2^{n-1}}^{(n-1)},$$

i.e.

$$q_0^{(n)} - q_N^{(n)} = 2\left(p_0^{(n)} - p_N^{(n)}\right). \tag{14}$$

Further, from (9) and (14) we obtain that

$$F_0^{(n)} - F_N^{(n)} = C^{(n)}\left(p_0^{(n)} - p_N^{(n)}\right) + q_0^{(n)} - q_N^{(n)} = \left(C^{(n)} + 2E\right)\left(p_0^{(n)} - p_N^{(n)}\right).$$

Taking into account formula (12), Section 3.2.1, we finally have

$$F_0^{(n)} - F_N^{(n)} = \left[C^{(n-1)}\right]^2\left(p_0^{(n)} - p_N^{(n)}\right). \tag{15}$$

We shall take advantage of this relation to find Y_0 and Y_N from (6). Subtracting the second equation in (6) from the first, and taking into account (15) and (12), Section 3.2.1, we obtain that

$$\begin{aligned} \left(C^{(n)} + 2E\right)(Y_0 - Y_N) &= \left[C^{(n-1)}\right]^2 (Y_0 - Y_N) = F_0^{(n)} - F_N^{(n)} \\ &= \left[C^{(n-1)}\right]^2\left(p_0^{(n)} - p_N^{(n)}\right). \end{aligned}$$

Considering that $C^{(n-1)}$ is a non-singular matrix, from this we find

$$Y_0 = Y_N + p_0^{(n)} - p_N^{(n)}. \tag{16}$$

Substituting this formula for Y_0 in the second equation of the system (9), we obtain an equation for finding Y_N:

$$B^{(n)}Y_N = F_N^{(n)} + 2\left(p_0^{(n)} - p_N^{(n)}\right) = B^{(n)}p_N^{(n)} + q_N^{(n)} + 2p_0^{(n)},$$

where $B^{(n)} = C^{(n)} - 2E$. Consequently, if we denote $t^{(n)} = Y_N - p_N^{(n)}$, Y_N can be found by solving the equation

$$B^{(n)}t^{(n)} = q_N^{(n)} + 2p_0^{(n)}, \qquad \left(Y_N = p_N^{(n)} + t^{(n)}\right). \tag{17}$$

From (16) we obtain that Y_0 can be found from the formula

$$Y_0 = p_0^{(n)} + t^{(n)}, \tag{18}$$

where $t^{(n)}$ is defined above.

Thus, the formulas (10)–(13), (17), and (18) define the cyclic reduction method for solving a boundary-value problem of the second kind for the three-point vector equations (1).

Remark. If Y_0 is given, i.e. in place of problem (1) we solve the problem

$$\begin{aligned} -Y_{j-1} + CY_j - Y_{j+1} &= F_j, \qquad 1 \le j \le N-1, \\ -2Y_{N-1} + CY_N &= F_N, \qquad j = N,\ Y_0 = F_0, \end{aligned}$$

then the vectors $p_0^{(k)}$ and $q_0^{(k)}$ are not needed, and it follows from (6) and (9) that Y_N can be found by solving the equation

$$C^{(n)}t_N^{(n)} = q_N^{(n)} + 2Y_0, \qquad (Y_N = p_N^{(n)} + t_N^{(n)}).$$

Analogously, if Y_N is given, then the vectors $p_N^{(k)}$ and $q_N^{(k)}$ are not needed, and Y_0 is determined from the equations $C^{(n)}t_0^{(n)} = q_0^{(n)} + 2Y_N$, $Y_0 = p_0^{(n)} + t_0^{(n)}$.

To complete the description of the reduction method, it is necessary to indicate how to invert the matrices $C^{(k)}$ and $B^{(n)} = C^{(n)} - 2E$. To invert the matrices $C^{(k-1)}$, the factorization obtained above (see (36) Section 3.2) is used

$$C^{(k-1)} = \prod_{l=1}^{2^{k-1}} C_{l,k-1}, \quad C_{l,k-1} = C - 2\cos\frac{(2l-1)\pi}{2^k}E. \tag{19}$$

Notice that, since the condition $(CY, Y) \ge 2(Y, Y)$ is satisfied, all the matrices $C_{l,k-1}$ are non-singular, and consequently the matrix $C^{(k-1)}$ is non-singular. We now look more closely at the question of inverting the matrix $B^{(n)}$.

From the definition of $B^{(n)}$ and the relation (12), Section 3.2.1 we obtain

$$\begin{aligned} B^{(n)} &= C^{(n)} - 2E = [C^{(n-1)}]^2 - 4E = (C^{(n-1)} + 2E)(C^{(n-1)} - 2E) \\ &= [C^{(n-2)}]^2[C^{(n-1)} - 2E] = \cdots = [C^{(n-2)}C^{(n-3)} \ldots C^{(0)}]^2(C^{(1)} - 2E) \\ &= [C^{(n-2)}C^{(n-3)} \ldots C^{(0)}]^2(C^{(0)} - 2E)(C^{(0)} + 2E) \\ &= \left[\prod_{k=1}^{n-1} C^{(k-1)}\right]^2 (C - 2E)(C + 2E). \end{aligned}$$

Substituting here (19), we find the following representation for the matrix:

$$B^{(n)} = \left[\prod_{k=1}^{n-1}\prod_{l=1}^{2^{k-1}} C_{l,k-1}\right]^2 (C - 2E)(C + 2E). \tag{20}$$

Thus, the matrix $B^{(n)}$ is factored and can be inverted by sequentially inverting the factors.

Remark 1. It is possible to obtain a more compact form for (20):

$$B^{(n)} = \prod_{l=1}^{2^n}\left(C - 2\cos\frac{l\pi}{2^{n-1}}E\right)$$

Remark 2. From (20) it follows that the matrix $B^{(n)}$ will be non-singular if $(CY, Y) > 2(Y, Y)$. If there exists a vector $Y^* \neq 0$ for which $CY^* = 2Y^*$, then $B^{(n)}$ is singular and direct application of the reduction method is impossible. This is a consequence of the singularity of the matrix in (1) for this case. In fact, in this case the homogeneous system (1) has the non-null solution $Y_j = Y^*$, and therefore the system (1) is not soluble for every right-hand side. If for a given right-hand side a solution exists, then it is not unique, and it is only determined up to a term in Y^*. One of the possible solutions is chosen at the step where the matrix $B^{(n)}$ is inverted. This situation occurs when solving a Neumann problem for Poisson's equation in a rectangle. This question will be looked at in more detail in Chapter 12 in connection with the solution of singular grid equations.

3.4.2 A periodic problem. Periodic three-point vector problems arise when difference methods are used to solve elliptic equations in curvilinear orthogonal coordinate systems — cylindrical, polar, and spherical systems. In Section 3.1.3, examples of differential problems were introduced whose difference

schemes led to the following problem: find the solution of the equations

$$\begin{aligned} -Y_{j-1} + CY_j - Y_{j+1} &= F_j, \qquad 1 \le j \le N-1, \\ -Y_{N-1} + CY_0 - Y_1 &= F_0, \qquad j = 0,\ Y_N = Y_0. \end{aligned} \tag{21}$$

The problem (21) can be solved using the cyclic reduction method also. We now look at the first step of the elimination process. As before, we eliminate the unknowns Y_j with odd indices from the even numbered equations of the system (21) using the two neighboring equations. We obtain

$$-Y_{j-2} + C^{(1)}Y_j - Y_{j+2} = F_j^{(1)}, \qquad j = 2,4,6,\ldots,N-2. \tag{22}$$

It remains to eliminate Y_1 and Y_{N-1} from equation (21) for $j = 0$. For this we write out the following three equations from the system (21):

$$\begin{aligned} -Y_0 + CY_1 - Y_2 &= F_1, && j = 1, \\ -Y_{N-1} + CY_0 - Y_1 &= F_0, && j = 0, \\ -Y_{N-2} + CY_{N-1} - Y_N &= F_{N-1}, && j = N-1, \end{aligned}$$

multiply the second equation on the left by C, add all three equations, and remember that $Y_N = Y_0$. As a result we obtain the equation

$$-Y_{N-2} + C^{(1)}Y_0 - Y_2 = F_0^{(1)}, \qquad Y_N = Y_0, \tag{23}$$

where

$$F_0^{(1)} = F_1^{(0)} + C^{(0)}F_0^{(0)} + F_{N-1}^{(0)}, \quad C^{(0)} = C, \quad F_j^{(0)} \equiv F_j.$$

Putting together (22) and (23), we obtain a full system for the unknowns Y_j with even indices, having a structure analogous to (21). The unknowns Y_j with odd indices are found from the usual equations

$$C^{(0)}Y_j = F_j^{(0)} + Y_{j-1} + Y_{j+1}, \qquad j = 1,3,5,\ldots,N-1.$$

The elimination process can be carried further. After the l-th step of the process, we obtain a system for the unknowns Y_j with indices divisible by 2^l:

$$\begin{aligned} -Y_{j-2^l} + C^{(l)}Y_j - Y_{j+2^l} &= F_j^{(l)}, \quad j = 2^l, 2\cdot 2^l, 3\cdot 2^l, \ldots, N-2^l, \\ -Y_{N-2^l} + C^{(l)}Y_0 - Y_{2^l} &= F_0^{(l)}, \quad j = 0, \quad Y_N = Y_0, \end{aligned}$$

and a group of equations

$$\begin{aligned} C^{(k-1)}Y_j &= F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ j &= 2^{k-1}, 3\cdot 2^{k-1}, 5\cdot 2^{k-1}, \ldots, N-2^{k-1}, \quad k = l, l-1, \ldots, 1 \end{aligned} \tag{24}$$

for subsequently finding the remaining unknowns. The right-hand sides $F_j^{(k)}$ are determined recursively for $k = 1, 2, \ldots, n-1$:

$$\begin{aligned} F_j^{(k)} &= F_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} F_j^{(k-1)} + F_{j+2^{k-1}}^{(k-1)}, \\ j &= 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \ldots, N - 2^k, \\ F_0^{(k)} &= F_{2^{k-1}}^{(k-1)} + C^{(k-1)} F_0^{(k-1)} + F_{N-2^{k-1}}^{(k-1)}, \\ F_j^{(0)} &\equiv F_j. \end{aligned} \tag{25}$$

At the end of the $(n-1)$-th step of the elimination process we obtain a system in Y_0 and $Y_{2^{n-1}}$ ($Y_N = Y_0$):

$$\begin{aligned} C^{(n-1)} Y_0 - 2Y_{2^{n-1}} &= F_0^{(n-1)}, \\ -2Y_0 + C^{(n-1)} Y_{2^{n-1}} &= F_{2^{n-1}}^{(n-1)}. \end{aligned} \tag{26}$$

Having solved this system, we find Y_0, $Y_{2^{n-1}}$ and $Y_N = Y_0$, and the remaining unknowns can be found using (24) as the solution of the equations

$$\begin{gathered} C^{(k-1)} Y_j = F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ j = 2^{k-1}, 3 \cdot 2^{k-1}, 5 \cdot 2^{k-1}, \ldots, N - 2^{k-1}, \quad k = n-1, n-2, \ldots, 1. \end{gathered}$$

Before solving (26), we find the recurrence relations for the vectors $p_j^{(k)}$ and $q_j^{(k)}$, which are related to $F_j^{(k)}$ by the following equation:

$$F_j^{(k)} = C^{(k)} p_j^{(k)} + q_j^{(k)}, \quad j = 0, 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \ldots, N - 2^k.$$

Using the recurrence formulas (25) for $F_j^{(k)}$, we obtain

$$\begin{aligned} C^{(k-1)} s_j^{(k-1)} &= q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, \\ p_j^{(k)} &= p_j^{(k-1)} + s_j^{(k-1)}, \\ q_j^{(k)} &= 2p_j^{(k)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, \\ j &= 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \ldots, N - 2^k, \quad k = 1, 2, \ldots, n-1, \\ q_j^{(0)} &\equiv F_j, \quad p_j^{(0)} \equiv 0, \quad j = 1, 2, \ldots, N-1, \end{aligned} \tag{27}$$

from which we find $p_j^{(k)}$ and $q_j^{(k)}$ for $j \neq 0$, and also

$$\begin{aligned} C^{(k-1)} s_0^{(k-1)} &= q_0^{(k-1)} + p_{2^{k-1}}^{(k-1)} + p_{N-2^{k-1}}^{(k-1)}, \\ p_0^{(k)} &= p_0^{(k-1)} + s_0^{(k-1)}, \\ q_0^{(k)} &= 2p_0^{(k)} + q_{2^{k-1}}^{(k-1)} + q_{N-2^{k-1}}^{(k-1)}, \quad k = 1, 2, \ldots, n-1, \\ q_0^{(0)} &= F_0, \quad p_0^{(0)} = 0 \end{aligned} \tag{28}$$

from which we find $p_0^{(k)}$ and $q_0^{(k)}$.

We turn now to the solution of (26). From (27) and (28) for $k = n-1$ we obtain the relations

$$\begin{aligned} q_{2^{n-1}}^{(n-1)} &= 2p_{2^{n-1}}^{(n-1)} + q_{2^{n-2}}^{(n-2)} + q_{3 \cdot 2^{n-2}}^{(n-2)}, \\ q_0^{(n-1)} &= 2p_0^{(n-1)} + q_{2^{n-2}}^{(n-1)} + q_{3 \cdot 2^{n-2}}^{(n-2)}, \end{aligned}$$

from which we find

$$q_0^{(n-1)} - q_{2^{n-1}}^{(n-1)} = 2\left(p_0^{(n-1)} - p_{2^{n-1}}^{(n-1)}\right). \tag{29}$$

We now subtract the second equation in (26) from the first. Using (29) and (12) from Section 3.2.1 we obtain

$$\begin{aligned} &(C^{(n-1)} + 2E)(Y_0 - Y_{2^{n-1}}) = [C^{(n-2)}]^2 (Y_0 - Y_{2^{n-1}}) = F_0^{(n-1)} - F_{2^{n-1}}^{(n-1)} \\ &= C^{(n-1)}(p_0^{(n-1)} - p_{2^{n-1}}^{(n-1)}) + q_0^{(n-1)} - q_{2^{n-1}}^{(n-1)} = [C^{(n-2)}]^2 (p_0^{(n-1)} - p_{2^{n-1}}^{(n-1)}). \end{aligned}$$

Assuming that $C^{(n-2)}$ is a non-singular, we conclude that

$$Y_{2^{n-1}} = Y_0 - p_0^{(n-1)} + p_{2^{n-1}}^{(n-1)}. \tag{30}$$

Substituting (30) in the first equation in (26), we obtain

$$\begin{aligned} (C^{(n-1)} - 2E)Y_0 &= F_0^{(n-1)} - 2(p_0^{(n-1)} - p_{2^{n-1}}^{(n-1)}) \\ &= (C^{(n-1)} - 2E)p_0^{(n-1)} + q_0^{(n-1)} + 2p_{2^{n-1}}^{(n-1)}. \end{aligned}$$

Consequently, Y_0 can be found from the formulas

$$\begin{aligned} B^{(n-1)} t^{(n-1)} &= q_0^{(n-1)} + 2p_{2^{n-1}}^{(n-1)}, \quad B^{(n-1)} = C^{(n-1)} - 2E, \\ Y_0 &= p_0^{(n-1)} + t^{(n-1)}, \end{aligned} \tag{31}$$

and using (30) $Y_{2^{n-1}}$ can then be found from the relation

$$Y_{2^{n-1}} = p_{2^{n-1}}^{(n-1)} + t^{(n-1)}. \tag{32}$$

The remaining unknowns are found sequentially from the formulas

$$\begin{aligned} &Y_N = Y_0, \\ &C^{(k-1)} t_j^{(k-1)} = q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ &Y_j = p_j^{(k-1)} + t_j^{(k-1)}, \\ &j = 2^{k-1}, 3 \cdot 2^{k-1}, 5 \cdot 2^{k-1}, \ldots, N - 2^{k-1}, \\ &k = n-1, n-2, \ldots, 1. \end{aligned} \tag{33}$$

Thus, the formulas (27), (28), (31)–(33) describe the cyclic reduction method for solving the periodic problem (21). The matrices $C^{(k-1)}$ and $B^{(n-1)}$ are inverted using the factorizations (19), (20), where in (20) it is necessary to change n to $n-1$.

We now estimate the number of arithmetic operations Q required to realize the cyclic reduction method for a periodic problem. As before, we denote by $\mathring{q}$ the number of operations required to solve the equation $C_{l,k-1}V = F$, and by $\bar{q}$ the number of auxiliary operations needed to solve the same system but with a different right-hand side F. The estimate is given by the formula

$$Q = \bar{q} N \log_2 N + (1.5\mathring{q} - 2\bar{q} + 7M)N - 2\mathring{q} + 2\bar{q} - 14M.$$

A comparison of this estimate with the estimate (45) Section 3.2 obtained in the case of a first-kind boundary-value probelm indicates that the cost of solving a periodic problem is practically the same as the cost of solving a boundary-value problem of the first kind.

3.4.3 A boundary-value problem of the third kind

3.4.3.1 *The elimination process.* We consider now the cyclic reduction method for solving a boundary-value problem of the third kind for the three-point vector equations

$$\begin{aligned} (C + 2\alpha E)Y_0 - 2Y_1 &= F_0, && j = 0, \\ -Y_{j-1} + CY_j - Y_{j+1} &= F_j, && 1 \le j \le N-1, \\ -2Y_{N-1} + (C + 2\beta E)Y_N &= F_N, && j = N. \end{aligned} \tag{34}$$

Assuming that $\alpha \ge 0$, $\beta \ge 0$, $\alpha^2 + \beta^2 \ne 0$, we introduce the following notation

$$C^{(0)} = C, \quad C_1^{(0)} = C + 2\alpha E, \quad C_2^{(0)} = C + 2\beta E, \quad F_j^{(0)} \equiv F_j,$$

using which we write (34) in the form

$$\begin{aligned} C_1^{(0)}Y_0 - 2Y_1 &= F_0^{(0)}, && j = 0, \\ -Y_{j-1} + C^{(0)}Y_j - Y_{j+1} &= F_j^{(0)}, && 1 \le j \le N-1, \\ -2Y_{N-1} + C_2^{(0)}Y_N &= F_N^{(0)}, && j = N. \end{aligned} \tag{34'}$$

Suppose $N = 2^n$. The elimination process for (34') is carried out in the same way as for the system (1), which corresponds to the case $C_1^{(0)} = C_2^{(0)} = C^{(0)}$ ($\alpha = \beta = 0$).

We write out the reduced system obtained at the end of the n-th step of the elimination process

$$C_1^{(n)}Y_0 - 2Y_N = F_0^{(n)}, \qquad -2Y_0 + C_2^{(n)}Y_N = F_N^{(n)}, \tag{6'}$$

and the group of equations

$$\begin{gathered} C^{(k-1)}Y_j = F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ j = 2^{k-1}, 3\cdot 2^{k-1}, \ldots, N - 2^{k-1}, \quad k = n, n-1, \ldots, 1 \end{gathered} \tag{35}$$

for sequentially finding the unknowns Y_j. Here the right-hand sides $F_j^{(k)}$ are determined using the recurrence relations

$$F_j^{(k)} = F_{j-2^{k-1}}^{(k-1)} + C^{(k-1)}F_j^{(k-1)} + F_{j+2^{k-1}}^{(k-1)}, \tag{36}$$

$$j = 2^k, 2\cdot 2^k, \ldots, N - 2^k, \qquad k = 1, 2, \ldots, n-1,$$

$$F_0^{(k)} = C^{(k-1)}F_0^{(k-1)} + 2F_{2^{k-1}}^{(k-1)}, \qquad k = 1, 2, \ldots, n, \tag{37}$$

$$F_N^{(k)} = 2F_{N-2^{k-1}}^{(k-1)} + C^{(k-1)}F_N^{(k-1)}, \qquad k = 1, 2, \ldots, n, \tag{38}$$

and the matrices $C_1^{(k)}$, $C_2^{(k)}$ and $C^{(k)}$ are found using

$$\begin{aligned} C^{(k)} &= [C^{(k-1)}]^2 - 2E, && k = 1, 2, \ldots, n-1,\ C^{(0)} = C, \\ C_1^{(k)} &= C^{(k-1)}C_1^{(k-1)} - 2E, && k = 1, 2, \ldots, n,\ C_1^{(0)} = C + 2\alpha E, \\ C_2^{(k)} &= C^{(k-1)}C_2^{(k-1)} - 2E, && k = 1, 2, \ldots, n,\ C_2^{(0)} = C + 2\beta E. \end{aligned} \tag{39}$$

From the system (6') we obtain equations for determining Y_0 and Y_N. From (39) it is possible to conclude that $C_1^{(k)}$, $C_2^{(k)}$, and $C^{(k)}$ are matrix polynomials of degree 2^k in the matrix C. Consequently they commute. Therefore from (6') we obtain the equations

$$\mathcal{D}^{(n+1)}Y_0 = F_0^{(n+1)}, \qquad C_2^{(n)}Y_N = F_N^{(n)} + 2Y_0 \tag{40}$$

and the equivalent equations

$$\mathcal{D}^{(n+1)}Y_N = F_N^{(n+1)}, \qquad C_1^{(n)}Y_0 = F_0^{(n)} + 2Y_N, \tag{40'}$$

where we have denoted

$$F_0^{(n+1)} = C_2^{(n)}F_0^{(n)} + 2F_N^{(n)}, \tag{41}$$

$$F_N^{(n+1)} = 2F_0^{(n)} + C_1^{(n)}F_N^{(n)}, \tag{42}$$

$$\mathcal{D}^{(n+1)} = C_1^{(n)}C_2^{(n)} - 4E = C_2^{(n)}C_1^{(n)} - 4E. \tag{43}$$

Thus it is possible to use the equations (40) or (40') to find Y_0 and Y_N. We will use (40).

Instead of the vectors $F_j^{(k)}$, we will determine the vectors $p_j^{(k)}$ and $q_j^{(k)}$, which are related to $F_j^{(k)}$ by the following equations:

$$F_0^{(k)} = C_1^{(k)}p_0^{(k)} + q_0^{(k)}, \tag{44}$$

$$F_N^{(k)} = C_2^{(k)}p_N^{(k)} + q_N^{(k)}, \qquad k = 0,1,\ldots,n, \tag{45}$$

$$F_0^{(n+1)} = \mathcal{D}^{(n+1)}p_0^{(n+1)} + q_0^{(n+1)}, \tag{46}$$

$$F_j^{(k)} = C^{(k)}p_j^{(k)} + q_j^{(k)}, \tag{47}$$

$$j = 2^k, 2\cdot 2^k, \ldots, N - 2^k, \quad k = 0,1,2,\ldots,n-1.$$

We now obtain recurrence formulas for $p_j^{(k)}$ and $q_j^{(k)}$. If $j \neq 0$, N then, assuming as before the non-singularity of the matrices $C^{(k-1)}$, we obtain from (36), (39), and (47) the following formulas

$$\begin{aligned} C^{(k-1)}s_j^{(k-1)} &= q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, \\ p_j^{(k)} &= p_j^{(k-1)} + s_j^{(k-1)}, \\ q_j^{(k)} &= 2p_j^{(k-1)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, \\ j &= 2^k, 2\cdot 2^k, \ldots, N-2^k, \quad k = 1,2,\ldots,n-1, \\ q_j^{(0)} &\equiv F_j, \qquad p_j^{(0)} \equiv 0. \end{aligned} \tag{48}$$

We now find formulas for $p_0^{(k)}$ and $q_0^{(k)}$ for $k = 0, 1, \ldots, n+1$. Substituting (44) and (47) in (37), and (44)–(46) in (41), we obtain for $k = 1, 2, \ldots, n$

$$\begin{aligned} &C_1^{(k)} p_0^{(k)} + q_0^{(k)} \\ &\quad = C^{(k-1)} \left(C_1^{(k-1)} p_0^{(k-1)} + q_0^{(k-1)} + 2p_{2^{k-1}}^{(k-1)} \right) + 2q_{2^{k-1}}^{(k-1)} \end{aligned} \tag{49}$$

and for $k = n + 1$

$$\mathcal{D}^{(n+1)} p_0^{(n+1)} + q_0^{(n+1)} = C_2^{(n)} \left(C_1^{(n)} p_0^{(n)} + q_0^{(n)} + 2p_N^{(n)} \right) + 2q_N^{(n)}. \tag{50}$$

We choose $q_0^{(k)}$ and $q_0^{(n+1)}$ from the formulas

$$\begin{aligned} q_0^{(k)} &= 2p_0^{(k)} + 2q_{2^{k-1}}^{(k-1)}, \qquad k = 1, 2, \ldots, n, \\ q_0^{(n+1)} &= 4p_0^{(n+1)} + 2q_N^{(n)} \end{aligned} \tag{51}$$

and use the following equations which arise from (39) and (43)

$$C_1^{(k)} + 2E = C^{(k-1)} C_1^{(k-1)}, \qquad \mathcal{D}^{(n+1)} + 4E = C_2^{(n)} C_1^{(n)}.$$

Then, if $C^{(k-1)}$ and $C_2^{(n)}$ are non-singular, (49) and (50) can be written in the form of a single equation

$$\begin{aligned} C_1^{(k-1)} p_0^{(k)} = C_1^{(k-1)} p_0^{(k-1)} + q_0^{(k-1)} + 2p_{2^{k-1}}^{(k-1)}, \\ k = 1, 2, \ldots, n+1. \end{aligned}$$

Combining these equations with (51), we obtain the final formulas for computing $p_0^{(k)}$ and $q_0^{(k)}$:

$$\begin{aligned} &C_1^{(k-1)} s_0^{(k-1)} = q_0^{(k-1)} + 2p_{2^{k-1}}^{(k-1)}, \\ &p_0^{(k)} = p_0^{(k-1)} + s_0^{(k-1)}, \qquad k = 1, 2, \ldots, n+1, \\ &q_0^{(k)} = 2p_0^{(k)} + 2q_{2^{k-1}}^{(k-1)}, \qquad k = 1, 2, \ldots, n, \\ &q_0^{(n+1)} = 4p_0^{(n+1)} + 2q_N^{(n)}, \\ &q_0^{(0)} = F_0, \qquad p_0^{(0)} = 0. \end{aligned} \tag{52}$$

Analogously, using (45), (47), and the recurrence relations (38) and (39), we obtain the formulas for computing $p_N^{(k)}$ and $q_N^{(k)}$:

$$\begin{aligned} C_2^{(k-1)} s_N^{(k-1)} &= q_N^{(k-1)} + 2p_{N-2^{k-1}}^{(k-1)}, \\ p_N^{(k)} &= p_N^{(k-1)} + s_N^{(k-1)}, \\ q_N^{(k)} &= 2p_N^{(k)} + 2q_{N-2^{k-1}}^{(k-1)}, \quad k = 1, 2, \ldots, n, \\ q_N^{(0)} &= F_N, \qquad p_N^{(0)} = 0. \end{aligned} \tag{53}$$

It remains to eliminate $F_j^{(k)}$ from (35) and (40). Substituting (47) in (35), and (45) and (46) in (40), we obtain the following formulas for finding Y_j:

$$\mathcal{D}^{(n+1)} s_0^{(n+1)} = q_0^{(n+1)}, \quad Y_0 = p_0^{(n+1)} + s_0^{(n+1)}, \tag{54}$$

$$C_2^{(n)} s_N^{(n)} = q_N^{(n)} + 2Y_0, \quad Y_N = p_N^{(n)} + s_N^{(n)}, \tag{55}$$

$$\begin{aligned} C^{(k-1)} s_j^{(k-1)} &= q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ Y_j &= p_j^{(k-1)} + s_j^{(k-1)}, \\ j &= 2^{k-1}, 3 \cdot 2^{k-1}, \ldots, N - 2^{k-1}, \quad k = n, n-1, \ldots, 1. \end{aligned} \tag{56}$$

Thus, the formulas (48), (52)–(56) describe the cyclic reduction method for the boundary-value problem of the third kind (34).

Remark 1. If equation (40′) is used to find Y_0 and Y_N, then, introducing in place of $p_0^{(n+1)}$ and $q_0^{(n+1)}$ the vectors $p_N^{(n+1)}$ and $q_N^{(n+1)}$ related to $F_N^{(n+1)}$ by the equation

$$F_N^{(n+1)} = \mathcal{D}^{(n+1)} p_N^{(n+1)} + q_N^{(n+1)},$$

we obtain from (38), (42), (44), and (47) the following formulas for finding $p_N^{(k)}$ and $q_N^{(k)}$:

$$\begin{aligned} C_2^{(k-1)} s_N^{(k-1)} &= q_N^{(k-1)} + 2p_{N-2^{k-1}}^{(k-1)}, \\ p_N^{(k)} &= p_N^{(k-1)} + s_N^{(k-1)}, \qquad k = 1, 2, \ldots, n+1, \\ q_N^{(k)} &= 2p_N^{(k)} + 2q_{N-2^{k-1}}^{(k-1)}, \qquad k = 1, 2, \ldots, n, \\ q_N^{(n+1)} &= 4p_N^{(n+1)} + 2q_0^{(n)}, \\ q_N^{(0)} &= F_N, \qquad p_N^{(0)} = 0. \end{aligned} \tag{53′}$$

The formulas (53′) replace (53). Since in this case it is not necessary to compute the vector $F_0^{(n+1)}$ or the vectors $p_0^{(n+1)}$ and $q_0^{(n+1)}$, (52) can be changed to:

$$\begin{aligned} C_1^{(k-1)} s_0^{(k-1)} &= q_0^{(k-1)} + 2p_{2^{k-1}}^{(k-1)}, \\ p_0^{(k)} &= p_0^{(k-1)} + s_0^{(k-1)}, \\ q_0^{(k)} &= 2p_0^{(k)} + 2q_{2^{k-1}}^{(k-1)}, \quad k = 1, 2, \ldots, n, \\ q_0^{(0)} &= F_0, \qquad p_0^{(0)} = 0. \end{aligned} \tag{52′}$$

From (35) and (40′) we obtain formulas for finding Y_0 and Y_N:

$$\mathcal{D}^{(n+1)} s_N^{(n+1)} = q_N^{(n+1)}, \qquad Y_N = p_N^{(n+1)} + s_N^{(n+1)}, \tag{55′}$$

$$C_1^{(n)} s_0^{(n)} = q_0^{(n)} + 2Y_N, \qquad Y_0 = p_0^{(n)} + s_0^{(n)}. \tag{54′}$$

The remaining unknowns are found using (56). Thus, the formulas (48), (52′)–(55′), and (56) can also be used to solve (34).

Remark 2. If Y_N is given, i.e. in place of (34) it is necessary to solve the boundary-value problem

$$\begin{aligned} (C + 2\alpha E)Y_0 - 2Y_1 &= F_0, && j = 0, \\ -Y_{j-1} + CY_j - Y_{j+1} &= F_j, && 1 \le j \le N-1, \\ Y_N &= F_N, && j = N, \end{aligned}$$

then the cyclic reduction method is described by (48), (52′), (54′), and (56). If we are given Y_0, i.e. we are solving the problem

$$\begin{aligned} -Y_{j-1} + CY_j - Y_{j+1} &= F_j, && 1 \le j \le N-1, \\ -2Y_{N-1} + (C + 2\beta E)Y_N &= F_N, && j = N,\ Y_0 = F_0, \end{aligned}$$

then the method is described by (48), (53), (55), and (56).

3.4.3.2 *Factoring the matrices.* From (39) and (43) it follows that $C_1^{(k)}$, $C_2^{(k)}$ and $C^{(k)}$ are matrix polynomials of degree 2^k, and $\mathcal{D}^{(n+1)}$ is a polynomial of degree 2^{n+1}, in the matrix C with the coefficient of highest degree equal to 1. Having in mind the necessity of inverting these matrices, we now factor them. We will obtain an explicit representation of these polynomials in terms of known polynomials, and study the roots of the indicated polynomials.

In Section 3.2.2 it was shown that $C^{(k)}$ can be expressed in terms of the Chebyshev polynomials of the first kind in the following way:

$$C^{(k)} = 2T_{2^k}\left(\frac{1}{2}C\right), \qquad k = 0, 1, \ldots. \tag{57}$$

Further, from (39) we find

$$\begin{aligned} C_1^{(k)} - C^{(k)} &= C^{(k-1)}\left[C_1^{(k-1)} - C^{(k-1)}\right] = \ldots \\ \ldots &= \prod_{l=0}^{k-1} C^{(l)}\left[C_1^{(0)} - C^{(0)}\right] = 2\alpha \prod_{l=0}^{k-1} C^{(l)}. \end{aligned} \tag{58}$$

Since

$$\prod_{l=0}^{k-1} C^{(l)} = \prod_{l=0}^{k-1} 2T_{2^l}\left(\frac{1}{2}C\right) = U_{2^k-1}\left(\frac{1}{2}C\right),$$

where $U_n(x)$ is the Chebyshev polynomial of the second kind, we obtain from (58) the following representation for $C_1^{(k)}$:

$$C_1^{(k)} = 2T_{2^k}\left(\frac{1}{2}C\right) + 2\alpha U_{2^k-1}\left(\frac{1}{2}C\right), \qquad k = 0, 1, \ldots. \tag{59}$$

Analogously we obtain a representation for $C_2^{(k)}$:

$$C_2^{(k)} = 2T_{2^k}\left(\frac{1}{2}C\right) + 2\beta U_{2^k-1}\left(\frac{1}{2}C\right), \qquad k = 0, 1, \ldots. \tag{60}$$

Further, substituting (59) and (60) in (43), we have

$$\begin{aligned} \mathcal{D}^{(n+1)} &= 4\left[T_{2^k}\left(\frac{1}{2}C\right)\right]^2 - 4E \\ &+ 4(\alpha+\beta)T_{2^k}\left(\frac{1}{2}C\right)U_{2^k-1}\left(\frac{1}{2}C\right) + 4\alpha\beta\left[U_{2^k-1}\left(\frac{1}{2}C\right)\right]^2. \end{aligned} \tag{61}$$

Since

$$1 - T_n(x) = U_{n-1}(x)(1 - x^2), \tag{62}$$

from (61) we obtain

$$\begin{aligned} \mathcal{D}^{(n+1)} = U_{2^n-1}\left(\frac{1}{2}C\right)\Big[&(C^2 + 4\alpha\beta E - 4E)U_{2^n-1}\left(\frac{1}{2}C\right) \\ &+ 4(\alpha+\beta)T_{2^n}\left(\frac{1}{2}C\right)\Big]. \end{aligned}$$

Thus, we have obtained a representation for $C^{(k)}$, $C_1^{(k)}$, $C_2^{(k)}$, and $\mathcal{D}^{(n+1)}$ in terms of known polynomials. Since the roots of the Chebyshev polynomials of the first and second kinds are known, from (57) and (62) we obtain

$$C^{(k)} = \prod_{l=1}^{2^k} \left(C - 2\cos\frac{(2l-1)\pi}{2^{k+1}} E \right),$$

$$\mathcal{D}^{(n+1)} = \prod_{l=1}^{2^n-1} \left(C - 2\cos\frac{l\pi}{2^n} E \right) \left[(C^2 + 4\alpha\beta E - 4E) U_{2^n-1}\left(\frac{1}{2}C\right) + 4(\alpha+\beta) T_{2^n}\left(\frac{1}{2}C\right) \right].$$

Therefore it follows from (59), (60) that it remains for us to find the roots of the polynomials

$$\begin{aligned} P_m(t) &= 2T_m\left(\frac{t}{2}\right) + 2\alpha U_{m-1}\left(\frac{t}{2}\right), \\ Q_m(t) &= 2T_m\left(\frac{t}{2}\right) + 2\beta U_{m-1}\left(\frac{t}{2}\right), \\ m &= 2^k, \quad k = 0, 1, \ldots, n-1, \end{aligned} \tag{63}$$

which correspond to the matrix polynomials $C_1^{(k)}$ and $C_2^{(k)}$, and the roots of the polynomial

$$R_{2^n+1}(t) = (t^2 + 4\alpha\beta - 4) U_{2^n-1}\left(\frac{t}{2}\right) + 4(\alpha+\beta) T_{2^n}\left(\frac{t}{2}\right), \tag{64}$$

which generates the polynomial $\mathcal{D}^{(n+1)}$.

This problem can be solved in two ways. The first method involves approximately finding the roots of the polynomials, the second involves transforming this problem to an eigenvalue problem for some tridiagonal matrix. We shall look in more detail at the second method.

We denote by $S_k(\lambda)$ the folowing k-th order determinant:

$$S_k(\lambda) = \begin{vmatrix} \lambda+2\alpha & 2 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 \\ 1 & \lambda & 1 & 0 & \ldots & 0 & 0 & 0 & 0 \\ 0 & 1 & \lambda & 1 & \ldots & 0 & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & \ldots & 1 & \lambda & 1 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 & \lambda & 1 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 1 & \lambda \end{vmatrix}, k \geq 2$$

and set $S_1(\lambda) = \lambda + 2\alpha$. From the definition and the structure of the matrix corresponding to $S_k(\lambda)$, we find a recurrence relation for $S_k(\lambda)$:

$$\begin{aligned} S_{k+1}(\lambda) &= \lambda S_k(\lambda) - S_{k-1}(\lambda), \qquad k \geq 2, \\ S_2(\lambda) &= \lambda S_1(\lambda) - 2, \quad S_1(\lambda) = \lambda + 2\alpha. \end{aligned} \tag{65}$$

Using the recurrence relation for the Chebyshev polynomials (see Section 1.4.2)

$$\begin{aligned} T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x), \quad T_1(x) = x, \quad T_0(x) = 1, \\ U_{n+1}(x) &= 2xU_n(x) - U_{n-1}(x), \quad U_1(x) = 2x, \quad U_0(x) = 1 \end{aligned}$$

and the relation (65), we obtain a representation for $S_m(\lambda)$ in terms of the Chebyshev polynomials:

$$S_m(\lambda) = 2T_m\left(\frac{\lambda}{2}\right) + 2\alpha U_{m-1}\left(\frac{\lambda}{2}\right), \quad m \geq 1.$$

Comparing this expression with (63) we find that the roots of the polynomial $P_m(t)$ coincide with the roots of the determinant $S_m(\lambda)$.

The problem of finding the roots of $S_m(\lambda)$ is equivalent to the problem of finding those values of the parameter λ for which the system of algebraic equations

$$\begin{aligned} y_{i-1} + \lambda y_i + y_{i+1} &= 0, \qquad 1 \leq i \leq m-1, \\ (\lambda + 2\alpha)y_0 + 2y_1 &= 0, \qquad i = 0, \\ y_m &= 0 \end{aligned} \tag{66}$$

has a non-null solution. We shall write (66) in another form. Using the notation for the second difference derivative

$$y_{\bar{x}x,i} = \frac{1}{h}(y_{x,i} - y_{\bar{x},i}) = \frac{1}{h^2}(y_{i+1} - 2y_i + y_{i-1}),$$

we rewrite (66) in the following form

$$\begin{aligned} &y_{\bar{x}x} + \mu y = 0, \qquad 1 \leq i \leq m-1, \\ &\frac{2}{h}y_x + \frac{2\alpha}{h^2}y + \mu y = 0, \quad i = 0, \; y_m = 0, \end{aligned} \tag{66'}$$

where λ and μ are connected by the relation $\lambda = \mu h^2 - 2$. Thus, to find the roots of the polynomial $C_1^{(k)}$ it is sufficient to solve the problem (66') for $m = 2^k$, $k = 0, 1, \ldots$.

By analogy with the above, it is possible to show that the roots of the polynomial $Q_m(t)$ are found by solving the problem

$$\begin{aligned} y_{\bar{x}x} + \mu y &= 0, \qquad 1 \le i \le m-1, \\ -\frac{2}{h} y_{\bar{x}} + \frac{2\beta}{h^2} y + \mu y &= 0, \qquad i = m,\ y_0 = 0, \end{aligned} \tag{67}$$

where the relation $\lambda = \mu h^2 - 2$ determines these roots.

To find the roots of the polynomial $R_{2^n+1}(t)$ defined in (64), it is necessary to solve the following eigenvalue problem:

$$\begin{aligned} y_{\bar{x}x} + \mu y &= 0, \qquad 1 \le i \le 2^n - 1, \\ \frac{2}{h} y_x + \frac{2\alpha}{h^2} y + \mu y &= 0, \qquad i = 0, \\ -\frac{2}{h} y_{\bar{x}} + \frac{2\beta}{h^2} y + \mu y &= 0, \qquad i = 2^n, \end{aligned} \tag{68}$$

and the roots are found from the equation $\lambda = \mu h^2 - 2$.

Notice that the problems (66)–(68) can be solved using the QR algorithm for the complete eigenproblem.

Chapter 4

The Separation of Variables Method

In this chapter we study variants of the method of separation of variables, which we use to solve the simplest elliptic grid equations in a rectangle. In Section 4.1 we present an algorithm for the fast Fourier transform of real and complex functions. In Section 4.2 we consider a classical variant of the method of separation of variables, using the Fourier transform algorithm. In Section 4.3 we construct a combined method, including incomplete reduction and separation of variables. We consider an application of this method to the solution of second and fourth order boundary value difference problems for Poisson's equation. In Section 4.4 we describe a stable variant of the staircase algorithm for solving systems with tridiagonal and block tridiagonal matrices, also using the Fourier transform algorithm.

4.1 The algorithm for the discrete Fourier transform

4.1.1 Statement of the problem. One of the methods of separating variables for finding the solution of multi-dimensional grid problems is the expansion of the solution in a finite Fourier sum using the eigenfunctions of the corresponding grid operators. The effectiveness of this method depends on how quickly the Fourier coefficients of the given grid function can be computed and how quickly the desired function can be regenerated from the Fourier coefficients.

If, for example, we have defined the function $f(i)$ and the orthonormal system of functions $\mu_k(i)$, $k = 0,1,\dots N$, on the grid $\bar{\omega} = \{x_i = ih, 0 \leq i \leq N, hN = l\}$, containing $N+1$ nodes, and the Fourier coefficients of the

function $f(i)$ are computed from the formulas

$$\varphi_k = \sum_{i=0}^{N} f(i)\mu_k(i)h, \qquad k = 0, 1, \ldots, N, \tag{1}$$

then computing all the coefficients φ_k requires $(N+1)(N+2)$ multiplications and $N(N+1)$ additions.

In the general case of an arbitrary system of functions $\{\mu_k(i)\}$ this is the minimal number of arithmetic operations required. In a series of special cases where the orthonormal system of functions has a special form, the number of arithmetic operations necessary to compute sums of the form (1) can be significantly reduced. We shall look at these cases and develop algorithms which allow us to compute all the Fourier coefficients and regenerate the function from the Fourier coefficients in $O(N \ln N)$ arithmetic operations.

We move on now to a description of some special cases.

Problem 1. *Expansion in sines.* Suppose that we have introduced on the interval $0 \le x \le l$ the uniform grid $\bar{\omega} = \{x_j = jh,\ 0 \le j \le N,\ hN = l\}$ with step h. We denote by $\omega = \{x_j = jh,\ 1 \le j \le N-1\}$ the set of interior nodes of the grid $\bar{\omega}$.

Suppose that the real-valued grid function $f(j)$ is defined on ω (or $f(j)$ is defined on $\bar{\omega}$, where $f(0) = f(N) = 0$).

In Section 1.5 it was shown that the function $f(j)$ can be represented in the form of an expansion

$$f(j) = \frac{2}{N} \sum_{k=1}^{N-1} \varphi_k \sin \frac{k\pi j}{N}, \qquad j = 1, 2, \ldots, N-1, \tag{2}$$

where the coefficients φ_k are determined by the formula

$$\varphi_k = \sum_{j=1}^{N-1} f(j) \sin \frac{k\pi j}{N}, \qquad k = 1, 2, \ldots, N-1. \tag{3}$$

Comparing (2) and (3) we find that the problems of computing the coefficients φ_k for the given function $f(j)$ and regenerating this function from $\{\varphi_k\}$ reduce to the computation of $N-1$ sums of the form

$$y_k = \sum_{j=1}^{N-1} a_j \sin \frac{k\pi j}{N}, \qquad k = 1, 2, \ldots, N-1. \tag{4}$$

The formula (4) describes a rule for transforming a grid function a_j, $1 \le j \le N-1$ defined on the grid ω into the grid function y_j, $1 \le j \le N-1$. The algebraic interpretation of (4) is as follows: if we denote by $a = (a_1, a_2, \dots, a_{N-1})$ the vector of dimension $N-1$, then (4) describes the transformation of the vector a which results when we move from the natural basis to the basis formed by the system of orthogonal vectors

$$z_k = \big(z_k(1), z_k(2), \dots, z_k(N-1)\big), \quad z_k(j) = \sin\frac{k\pi j}{N}.$$

Problem 2. *Expansion in shifted sines.* Suppose that the real-valued grid function $f(j)$ is defined on the set $\omega^+ = \{x_j = jh,\ 1 \le j \le N\}$ (or on $\bar{\omega}$, where $f(0) = 0$). In Section 1.5 it was shown that such a function $f(j)$ can be represented in the form

$$f(j) = \frac{2}{N}\sum_{k=1}^{N} \varphi_k \sin\frac{(2k-1)\pi j}{2N}, \qquad j = 1, 2, \dots, N, \tag{5}$$

where the coefficients φ_k are determined by the formula

$$\varphi_k = \sum_{j=1}^{N} \rho_j f(j) \sin\frac{(2k-1)\pi i}{2N}, \qquad k = 1, 2, \dots, N, \tag{6}$$

where

$$\rho_j = \begin{cases} 1, & j \ne 0, N, \\ 0.5, & j = 0, N. \end{cases} \tag{7}$$

If the function $f(j)$ is defined on the set $\omega^- = \{x_j = jh,\ 0 \le j \le N-1\}$ (or on $\bar{\omega}$, where $f(N) = 0$), then the expansion corresponding to (5) and (6) has the form

$$f(N-j) = \frac{2}{N}\sum_{k=1}^{N} \varphi_k \sin\frac{(2k-1)\pi j}{2N}, \qquad j = 1, 2, \dots, N, \tag{8}$$

$$\varphi_k = \sum_{j=1}^{N} \rho_{N-j} f(N-j) \sin\frac{(2k-1)\pi j}{2N}, \qquad k = 1, 2, \dots, N, \tag{9}$$

where the function ρ_j is defined in (7).

From (5), (6), (8), and (9) it follows that here the problem is to compute sums of the form

$$y_k = \sum_{j=1}^{N} a_j \sin \frac{(2k-1)\pi j}{2N}, \qquad k = 1, 2, \ldots, N, \tag{10}$$

$$y_j = \sum_{k=1}^{N} a_k \sin \frac{(2k-1)\pi j}{2N}, \qquad j = 1, 2, \ldots, N, \tag{10'}$$

Problem 3. *Expansion in cosines.* Suppose that the real-valued function $f(j)$ is defined on the grid $\bar{\omega}$. Then for the function $f(j)$ we have the expansion

$$f(j) = \frac{2}{N} \sum_{k=0}^{N} \rho_k \varphi_k \cos \frac{k\pi j}{N}, \qquad j = 0, 1, \ldots, N, \tag{11}$$

where

$$\varphi_k = \sum_{j=0}^{N} \rho_j f(j) \cos \frac{k\pi j}{N}, \qquad k = 0, 1, \ldots, N, \tag{12}$$

and ρ_j is defined in (7). From the formulas (11) and (12) comes the problem of computing sums of the form

$$y_k = \sum_{j=0}^{N} a_j \cos \frac{k\pi j}{N}, \qquad k = 0, 1, \ldots, N. \tag{13}$$

Problem 4. *Transformation of a real-valued periodic grid function.* Assume that on the axis $-\infty < x < \infty$ the uniform grid $\Omega = \{x_j = jh, j = 0, \pm 1, \pm 2, \ldots, Nh = l\}$ with step h has been defined. Suppose that the real-valued grid function

$$f(j) = f(j + N), \quad j = 0, \pm 1, \ldots,$$

periodic with period N, has been defined on Ω. In Section 1.5 it was shown that the function $f(j)$ could be represented for $0 \le j \le N - 1$ in the form (for even N)

$$f(j) = \frac{2}{N} \left[\sum_{k=0}^{N/2} \rho_k \varphi_k \cos \frac{2k\pi j}{N} + \sum_{k=1}^{N/2-1} \bar{\varphi}_k \sin \frac{2k\pi j}{N} \right], \quad j = 0, 1, \ldots, N-1, \tag{14}$$

where the coefficients φ_k and $\bar{\varphi}_k$ are defined by the formulas

$$\varphi_k = \sum_{j=0}^{N-1} f(j) \cos \frac{2k\pi j}{N}, \quad k = 0, 1, \ldots, \frac{N}{2}, \tag{15}$$

$$\bar{\varphi}_k = \sum_{j=1}^{N-1} f(j) \sin \frac{2k\pi j}{N}, \quad k = 1, 2, \ldots, \frac{N}{2} - 1, \tag{16}$$

and the function ρ_k is

$$\rho_k = \begin{cases} 1 & k \neq 0, N/2, \\ 0.5, & k = 0, N/2. \end{cases}$$

The formulas (14)–(16) lead us to the problem of computing sums of three forms:

$$y_k = \sum_{j=0}^{N/2} a_j \cos \frac{2k\pi j}{N} + \sum_{j=1}^{N/2-1} \bar{a}_j \sin \frac{2k\pi j}{N}, \quad k = 0, 1, \ldots, N-1, \tag{17}$$

$$\left.\begin{aligned} y_k &= \sum_{j=0}^{N-1} a_j \cos \frac{2k\pi j}{N}, \quad k = 0, 1, \ldots, N/2, \\ \bar{y}_k &= \sum_{j=1}^{N-1} a_j \sin \frac{2k\pi j}{N}, \quad k = 1, 2, \ldots, N/2 - 1, \end{aligned}\right\} \tag{18}$$

where the coefficients in the sums (18) are the same.

Problem 5. *Transformation of a complex periodic grid function.* Suppose that the grid function $f(j)$, periodic with period N, is defined on the grid Ω and takes on complex values. The function $f(j)$ can be represented for $0 \leq j \leq N-1$ in the form

$$f(j) = \frac{1}{N} \sum_{k=0}^{N-1} \varphi_k \exp\left(\frac{2k\pi j}{N} i\right), \quad j = 0, 1, \ldots, N-1, \quad i = \sqrt{-1}, \tag{19}$$

where the complex coefficients φ_k are defined by the formula

$$\varphi_k = \sum_{j=0}^{N-1} f(j) \exp\left(\frac{-2k\pi j}{N} i\right), \quad k = 0, 1, \ldots, N-1. \tag{20}$$

Notice that $\varphi_0 = \varphi_N$ and, in addition,

$$\varphi_{N-k} = \sum_{j=0}^{N-1} f(j) \exp\left(\frac{2k\pi j}{N} i\right), \quad k = 0, 1, \ldots, N-1.$$

Therefore, the computation of the coefficients φ_k and the regeneration of the function $f(j)$ leads to the computation of sums of the form

$$y_k = \sum_{j=0}^{N-1} a_j \exp\left(\frac{2k\pi j}{N} i\right), \quad k = 0, 1, \ldots, N-1 \tag{21}$$

with complex a_j.

Thus, it is necessary for us to construct algorithms which compute sums of the form (4), (10), (13), (17), (18), and (21), and which requires less than $O(N^2)$ arithmetic operations. It is easiest to construct the algorithm in the case when N is a power of 2: $N = 2^n$, and we will limit ourselves to this case.

4.1.2 Expansion in sines and shifted sines. We now consider in more detail an algorithm for computing the sums (4), assuming that $N = 2^n$. In this case (4) has the form

$$y_k = \sum_{j=1}^{2^n-1} a_j^{(0)} \sin\frac{k\pi j}{2^n}, \quad k = 1, 2, \ldots, 2^n - 1, \tag{22}$$

where we have introduced the notation $a_j^{(0)} = a_j$.

The idea of the method consists in first grouping together the terms in the sum (22) having a common multiplier, and then carrying out the multiplication. At the first stage of the algorithm, the terms with indices j and $2^n - j$ are grouped together for $j = 1, 2, \ldots, 2^{n-1} - 1$ using the relation

$$\sin\frac{k\pi(2^n - j)}{2^n} = (-1)^{k-1} \sin\frac{k\pi j}{2^n}. \tag{23}$$

We write out (22) in the form of three terms

$$y_k = \sum_{j=1}^{2^{n-1}-1} a_j^{(0)} \sin\frac{k\pi j}{2^n} + \sum_{j=2^{n-1}+1}^{2^n-1} a_j^{(0)} \sin\frac{k\pi j}{2^n} + a_{2^{n-1}}^{(0)} \sin\frac{k\pi}{2}$$

and make the change $j' = 2^n - j$ in the second sum. Taking into account (23) we obtain

$$y_k = \sum_{j=1}^{2^{n-1}-1} \left[a_j^{(0)} + (-1)^{k-1} a_{2^n-j}^{(0)}\right] \sin \frac{k\pi j}{2^n} + a_{2^{n-1}}^{(0)} \sin \frac{k\pi}{2}. \tag{24}$$

If we denote

$$\begin{aligned} a_j^{(1)} &= a_j^{(0)} - a_{2^n-j}^{(0)}, \\ a_{2^n-j}^{(1)} &= a_j^{(0)} + a_{2^n-j}^{(0)}, \quad j = 1, 2, \ldots, 2^{n-1} - 1, \\ a_{2^{n-1}}^{(1)} &= a_{2^{n-1}}^{(0)} \end{aligned}$$

then from (24) we have

$$y_{2k-1} = \sum_{j=1}^{2n-1} a_{2^n-j}^{(1)} \sin \frac{(2k-1)\pi j}{2^n}, \quad k = 1, 2, \ldots, 2^{n-1}, \tag{25}$$

$$y_{2k} = \sum_{j=1}^{2^{n-1}-1} a_j^{(1)} \sin \frac{k\pi j}{2^{n-1}}, \quad k = 1, 2, \ldots, 2^{n-1} - 1. \tag{26}$$

Thus as the end of the first stage we have two sums of the form (25) and (26), each of which contains about half as many terms as the original sum (22). Besides, the sums of the form (26) and the original sum have an analogous structure. Therefore the grouping process described above can be applied to (26).

At the second stage, as above, we partition the sums (26) into three terms and use (23), with n changed to $n-1$, to group the terms of the sum (26) with indices j and $2^{n-1} - 1$ for $j = 1, 2, \ldots, 2^{n-2} - 1$. As the end of the second stage, in place of (26) we obtain

$$y_{2(2k-1)} = \sum_{j=1}^{2^{n-2}} a_{2^{n-1}-j}^{(2)} \sin \frac{(2k-1)\pi j}{2^{n-1}}, \quad k = 1, 2, \ldots, 2^{n-2}, \tag{27}$$

$$y_{2^2 k} = \sum_{j=1}^{2^{n-2}-1} a_j^{(2)} \sin \frac{k\pi j}{2^{n-2}}, \quad k = 1, 2, \ldots, 2^{n-2} - 1, \tag{28}$$

where

$$\begin{aligned} a_j^{(2)} &= a_j^{(1)} - a_{2^{n-1}-j}^{(1)}, \\ a_{2^{n-1}-j}^{(2)} &= a_j^{(1)} + a_{2^{n-1}-j}^{(1)}, \quad j = 1, 2, \ldots, 2^{n-2} - 1, \\ a_{2^{n-2}}^{(2)} &= a_{2^{n-2}}^{(1)}. \end{aligned}$$

Thus, the original problem (22) is equivalent to computing the sums (25), (27), and (28). The formula (28) allows us to compute y_k for k divisible by 4, (27) for k divisible by 2 but not divisible by 4, and (25) is used to compute y_k for odd k.

Continuing the process of transforming summations, we obtain at the end of the p^{th} stage

$$\begin{aligned} y_{2^{s-1}(2k-1)} &= \sum_{j=1}^{2^{n-s}} a_{2^{n-s+1}-j}^{(s)} \sin \frac{(2k-1)\pi j}{2^{n-s+1}}, \\ &\qquad k = 1, 2, \ldots, 2^{n-s}, \quad s = 1, 2, \ldots, p, \\ y_{2^p k} &= \sum_{j=1}^{2^{n-p}-1} a_j^{(p)} \sin \frac{k\pi j}{2^{n-p}}, \\ &\qquad k = 1, 2, \ldots, 2^{n-p} - 1, \end{aligned} \tag{29}$$

where $p = 1, 2, \ldots, n-1$, and the coefficients $a_j^{(p)}$ are defined recursively

$$\begin{aligned} a_j^{(p)} &= a_j^{(p-1)} - a_{2^{n-p+1}-j}^{(p-1)}, \\ a_{2^{n-p+1}-j}^{(p)} &= a_j^{(p-1)} + a_{2^{n-p+1}-j}^{(p-1)}, \qquad j = 1, 2, \ldots, 2^{n-p} - 1, \\ a_{2^{n-p}}^{(p)} &= a_{2^{n-p}}^{(p-1)}, \qquad\qquad p = 1, 2, \ldots, n-1. \end{aligned} \tag{30}$$

Substituting $p = n - 1$ in (29) we find

$$\begin{aligned} y_{2^{n-1}} &= \sum_{j=1}^{1} a_j^{(n-1)} \sin \frac{\pi j}{2} = a_1^{(n-1)}, \\ y_{2^{s-1}(2k-1)} &= \sum_{j=1}^{2^{n-s}} a_{2^{n-s+1}-j}^{(s)} \sin \frac{(2k-1)\pi j}{2^{n-s+1}}, \quad k = 1, 2, \ldots, 2^{n-s} \end{aligned} \tag{31}$$

for $s = 1, 2, \ldots, n-1$.

Thus, the original problem (22) reduces to the computation of the $(n-1)^{\text{st}}$ group of sums (31). The required transformation of the coefficients $a_j^{(0)}$ is described by the formulas (30).

The second step of the algorithm consists in transforming the sums (31) which, after setting for each fixed s

$$\begin{aligned} z_k^{(0)}(1) &= y_{2^{s-1}(2k-1)}, & k &= 1,2,\ldots,2^{n-s}, \\ b_j^{(0)}(1) &= a^{(s)}_{2^{n-s+1}-j}, & j &= 1,2,\ldots,2^{n-s}, \\ l &= n-s, & s &= 1,2,\ldots,n-1, \end{aligned}$$

are written in the following form;

$$z_k^{(0)}(1) = \sum_{j=1}^{2^l} b_j^{(0)}(1) \sin \frac{(2k-1)\pi j}{2^{l+1}}, \quad k = 1,2,\ldots,2^l, \tag{32}$$

where $l = 1,2,\ldots,n-1$. Here the coefficients $b_j^{(0)}(1)$ and the functions $z_k^{(0)}(1)$ depend on the index l, but since here we are developing a method for computing the sum (32) for fixed l, the index is everywhere dropped.

We will now concentrate on the transformation of the sum (32). We represent it in the form of two terms, having divided the terms into those with even and odd indices j:

$$\begin{aligned} z_k^{(0)}(1) = &\sum_{j=1}^{2^{l-1}} b_{2j}^{(0)}(1) \sin \frac{(2k-1)\pi j}{2^l} \\ &+ \sum_{j=1}^{2^{l-1}} b_{2j-1}^{(0)}(1) \sin \frac{(2k-1)\pi(2j-1)}{2^{l+1}}. \end{aligned} \tag{33}$$

Using the equation

$$\sin\frac{(2k-1)(2j-2)\pi}{2^{l+1}} + \sin\frac{(2k-1)2j\pi}{2^{l+1}} = 2\cos\frac{(2k-1)\pi}{2^{l+1}} \sin\frac{(2k-1)(2j-1)\pi}{2^{l+1}},$$

we write out the second term in the form of two sums:

$$\begin{aligned} &\sum_{j=1}^{2^{l-1}} b_{2j-1}^{(0)}(1) \sin \frac{\pi(2k-1)(2j-1)}{2^{l+1}} \\ &= \frac{1}{2\cos\frac{(2k-1)\pi}{2^{l+1}}} \times \left[\sum_{j=1}^{2^{l-1}} b_{2j-1}^{(0)}(1) \sin \frac{(2k-1)\pi j}{2^l} \right. \\ &\qquad \left. + \sum_{j=1}^{2^{l-1}} b_{2j-1}^{(0)}(1) \sin \frac{(2k-1)\pi(j-1)}{2^l} \right] \end{aligned}$$

$$= \frac{1}{2\cos\frac{(2k-1)\pi}{2^{l+1}}} \left(b^{(0)}_{2^l-1}(1)\sin\frac{(2k-1)\pi}{2} + \sum_{j=1}^{2^{l-1}-1} \left(b^{(0)}_{2j+1}(1) + b^{(0)}_{2j-1}(1) \right) \sin\frac{(2k-1)\pi j}{2^l} \right). \quad (34)$$

Notice that the second sum in the square brackets was obtained by changing the index $j = j' + 1$.

We denote

$$\begin{aligned} b^{(1)}_j(1) &= b^{(0)}_{2j-1}(1) + b^{(0)}_{2j+1}(1), && j = 1, 2, \ldots, 2^{l-1} - 1, \\ b^{(1)}_{2^{l-1}}(1) &= b^{(0)}_{2^l-1}(1), \\ b^{(1)}_j(2) &= b^{(0)}_{2j}(1), && j = 1, 2, \ldots, 2^{l-1} \end{aligned}$$

and substitute (34) in (33). We obtain the expression

$$z^{(0)}_k(1) = \sum_{j=1}^{2^{l-1}} b^{(1)}_j(2)\sin\frac{(2k-1)\pi j}{2^l} + \frac{1}{2\cos\frac{(2k-1)\pi}{2^{l+1}}} \sum_{j=1}^{2^{l-1}} b^{(1)}_j(1)\sin\frac{(2k-1)\pi j}{2^l},$$

for $k = 1, 2, \ldots, 2^l$. Substituting $2^l - k + 1$ for k, we obtain

$$z^{(0)}_{2^l-k+1}(1) = -\sum_{j=1}^{2^{l-1}} b^{(1)}_j(2)\sin\frac{(2k-1)\pi j}{2^l} + \frac{1}{2\cos\frac{(2k-1)\pi}{2^{l+1}}} \sum_{j=1}^{2^{l-1}} b^{(1)}_j(1)\sin\frac{(2k-1)\pi j}{2^l}.$$

Consequently, if we denote

$$z^{(k)}_k(s) = \sum_{j=1}^{2^{l-1}} b^{(1)}_j(s)\sin\frac{(2k-1)\pi j}{2^l}, \quad k = 1, 2, \ldots, 2^{l-1}, \quad s = 1, 2,$$

then the original sum $z^{(0)}_k(1)$ can be computed from the formulas

$$\begin{aligned} z^{(0)}_k(1) &= z^{(1)}_k(2) + \frac{1}{2\cos\frac{(2k-1)\pi}{2^{l+1}}} z^{(1)}_k(1), \\ z^{(0)}_{2^l-k+1}(1) &= -z^{(1)}_k(2) + \frac{1}{2\cos\frac{(2k-1)\pi}{2^{l+1}}} z^{(1)}_k(1), \quad k = 1, 2, \ldots, 2^{l-1}. \end{aligned}$$

Thus, the first step gives rise to the sums $z_k^{(1)}(1)$ and $z_k^{(1)}(2)$, each of which contains half as many terms as the original sum $z_k^{(0)}(1)$, but which has the same structure as $z_k^{(0)}(1)$. Thus the transformation process described above for the original summation can be applied separately to the sums $z_k^{(1)}(1)$ and $z_k^{(1)}(2)$. As a result, we obtain the sums $z_k^{(2)}(s)$, $s = 1,2,3,4$, which also preserve the structure of the original sum. Continuing the transformation process, at the m^{th} stage we obtain the sums

$$z_k^{(m)}(s) = \sum_{j=1}^{2^{l-m}} b_j^{(m)}(s) \sin \frac{(2k-1)\pi j}{2^{l-m+1}}, \qquad k = 1,2,\dots,2^{l-m}, \quad s = 1,2,\dots,2^m \tag{35}$$

for each $m = 0,1,\dots,l$, where the coefficients $b_j^{(m)}(s)$ are defined recursively for $s = 1,2,\dots,2^{m-1}$ by the formulas

$$\begin{aligned} b_j^{(m)}(2s-1) &= b_{2j-1}^{(m-1)}(s) + b_{2j+1}^{(m-1)}(s), && j = 1,2,\dots,2^{l-m}, \\ & && m = 1,2,\dots,l-1, \\ b_{2^{l-m}}^{(m)}(2s-1) &= b_{2^{l-m+1}-1}^{(m-1)}(s) && m = 1,2,\dots,l, \\ b_j^{(m)}(2s) &= b_{2j}^{(m-1)}(s), && j = 1,2,\dots,2^{l-m}, \\ & && m = 1,2,\dots,l. \end{aligned} \tag{36}$$

Here the sums at the m^{th} stage are connected with the sums obtained at the $(m-1)^{\text{st}}$ stage by the following formulas:

$$\begin{aligned} z_k^{(m-1)}(s) &= z_k^{(m)}(2s) + \frac{1}{2\cos\frac{\pi(2k-1)}{2^{l-m+2}}} z_k^{(m)}(2s-1), \\ z_{2^{l-m+1}-k+1}^{(m-1)}(s) &= -z_k^{(m)}(2s) + \frac{1}{2\cos\frac{\pi(2k-1)}{2^{l-m+2}}} z_k^{(m)}(2s-1), \end{aligned} \tag{37}$$

$$k = 1,2,\dots,2^{l-m}, \quad s = 1,2,\dots,2^{m-1}, \quad m = 1,2,\dots,l.$$

Substituting $m = l$ in (35), we obtain

$$z_1^{(l)}(s) = b_1^{(l)}(s), \quad s = 1,2,\dots,2^l. \tag{38}$$

Thus, the sums $z_k^{(0)}(1)$ are computed as follows. Starting from the given coefficients $b_j^{(0)}$, $1 \le j \le 2^l$, the remaining coefficients $b_1^{(l)}(s)$, $1 \le s \le 2^l$, are computed from the formulas (36). By (38), they are used as initial data for the recurrence relations (37). Using (37) sequentially for $m = l, l-1, \dots, 1$, we obtain as a result $z_k^{(0)}(1)$ and consequently $y_{2^{s-1}(2k-1)}$.

Thus, the algorithm for computing the sums (22) is described by the formulas (30), (36), and (38).

Remark. In the recurrence relations (37) it is possible to avoid the division by $2\cos\frac{(2k-1)\pi}{2^{l-m+2}}$ by making the change

$$z_k^{(m)}(s) = \sin\frac{(2k-1)\pi}{2^{l-m+1}} w_k^{(m)}(s).$$

Here the formulas for computing $w_k^{(m)}(s)$ take the form

$$\begin{aligned} w_k^{(m-1)}(s) &= 2\cos\frac{\pi(2k-1)}{2^{l-m+2}} w_k^{(m)}(2s) + w_k^{(m)}(2s-1), \\ w_{2^{l-m+1}-k+1}^{(m-1)}(s) &= -2\cos\frac{\pi(2k-1)}{2^{l-m+2}} w_k^{(m)}(2s) + w_k^{(m)}(2s-1), \\ k = 1,2,\ldots,2^{l-m}, &\quad s = 1,2,\ldots,2^{m-1}, \quad m = l, l-1,\ldots,1, \end{aligned} \tag{39}$$

where $w_1^{(l)}(s) = b_1^{(l)}(s)$, $s = 1,2,\ldots,2^l$ and

$$z_k^{(0)}(1) = \sin\frac{(2k-1)\pi}{2^{l+1}} w_k^{(0)}(1), \quad k = 1,2,\ldots,2^l. \tag{40}$$

We now compute the number of arithmetic operations required to realize the algorithm (30), (36)–(38). We will assume that the values of the trigonometric functions have been previously computed.

An elementary computation shows:

[1] realizing (30) requires

$$Q_1 = \sum_{p=1}^{n-1} 2(2^{n-p} - 1) = 2\cdot 2^n - 2(n+1)$$

additions and subtractions;

[2] realizing (36) for fixed l requires

$$\bar{q}_l = \sum_{m=1}^{l-1} (2^{l-m} - 1)\cdot 2^{m-1} = (l-2)2^{l-1} + 1$$

additions, and realizing (37) requires

$$\bar{\bar{q}}_l = \sum_{m=1}^{l} 2\cdot 2^{l-m}\cdot 2^{m-1} = 2l\cdot 2^{l-1}$$

additions and

$$q_l^* = \sum_{m=1}^{l} 2^{l-m} \cdot 2^{m-1} = l \cdot 2^{l-1}$$

multiplications. In all, the formulas (36) and (37) require for fixed l

$$q_l = \bar{q}_l + \bar{\bar{q}}_l = (3l-2) \cdot 2^{l-1} + 1 \tag{42}$$

addtions and q_l^* multiplications. For all $l = 1, 2, \ldots, n-1$ the total is

$$Q_2 = \sum_{l=1}^{n-1} q_l = \sum_{l=1}^{n-1} \left[(3l-2) \cdot 2^{l-1} + 1\right] = \frac{3}{2} n 2^n - 4 \cdot 2^n + n + 4$$

additions and

$$Q_3 = \sum_{l=1}^{n-1} q_l^* = \sum_{l=1}^{n-1} l 2^{l-1} = \frac{n}{2} 2^n - 2^n + 1$$

multiplications.

Thus, the algorithm (30), (36)–(38) is characterized by the following estimates for the number of arithmetic operations: $Q_+ = Q_1 + Q_2 = (3n/2 - 2)2^n - n + 2$ additions and $Q_* = (n/2 - 1)2^n + 1$ multiplications. If no distinction is made between additions and multiplications, then the total number of operations is

$$Q = Q_1 + Q_2 + Q_3 = (2 \log_2 N - 3)N - \log_2 N + 3, \quad N = 2^n.$$

For comparison, we give here an estimate of the number of operations required to compute the sums (22) directly. We will have $(2^n - 1)^2$ multiplications and $(2^n - 2)(2^n - 1)$ additions, or in total $Q = (N-1)(2N-1)$. For example, if $N = 128$ ($n = 7$) we obtain $Q = 1404$ operations (of which 321 are multiplications) for the constructed algorithm, and $Q = 32,131$ operations (of which 15,873 are multiplications) for the direct algorithm.

Notice that using (39) and (40) in place of (37) and (38) results in the following estimates for the number of operations: $Q_+ = \left(\frac{3}{2}n - 2\right) 2^n - n + 2$ additions and $Q_* = \frac{n}{2} 2^n - 1$ multiplications, or in total $Q = (2 \log_2 N - 2)N - \log_2 N + 1$, $N = 2^n$, or slightly more than in the algorithm (30), (36)–(38).

Thus, the problem 1 stated above is solved. We now consider problem 2 involving the expansion in shifted sines. Assuming that $N = 2^n$, we write out the sum appearing in problem 2 in the following form

$$y_k = \sum_{j=1}^{2^n} a_j \sin \frac{(2k-1)\pi j}{2^{n+1}}, \quad k = 1, 2, \ldots, 2^n. \tag{43}$$

Comparing (43) with (32) we find that computing the sums (43) in shifted sines is the second step in the algorithm outlined above for computing the sums (22), if we substitute $l = n$ in (32). Consequently, if we denote

$$z_k^{(0)}(1) = y_k, \quad k = 1, 2, \ldots, 2^n,$$
$$b_j^{(0)}(1) = a_j, \quad j = 1, 2, \ldots, 2^n,$$

the formulas (36)–(38) for $l = n$ describe the algorithm for computing the sums (43). Substituting $l = n$ in the formulas (41) and (42), we obtain the following estimates for the new algorithm: $Q_+ = q_n = \left(\frac{3}{2}n - 1\right) 2^n + 1$ additions and $Q_* = q_n^* = \frac{n}{2} 2^n$ multiplications, or in total $Q = (2 \log_2 N - 1)N + 1$, $N = 2^n$. Thus, the sums (43) can be computed in about the same number of arithmetic operations as the sums (22).

Recall that the sums (43) are used to compute the Fourier coefficients for the grid function a_i defined for $i = 1, 2, \ldots, N$. To regenerate a function with given Fourier coefficients, it is necessary to compute the sums

$$y_j = \sum_{k=1}^{2^n} a_k \sin \frac{(2k-1)\pi j}{2^{n+1}}, \quad j = 1, 2, \ldots, 2^n. \tag{43'}$$

Using for $j \neq 2^n$ relation

$$\sin \frac{(2k-1)\pi j}{2^{n+1}} = \frac{1}{2 \cos \frac{\pi j}{2^{n+1}}} \left[\sin \frac{(k-1)\pi j}{2^n} + \sin \frac{k\pi j}{2^n} \right],$$

we obtain

$$y_j = \frac{1}{2 \cos \frac{\pi j}{2^{n+1}}} \left[\sum_{k=1}^{2^n} a_k \sin \frac{(k-1)\pi j}{2^n} + \sum_{k=1}^{2^n} a_k \sin \frac{k\pi j}{2^n} \right]$$
$$= \frac{1}{2 \cos \frac{\pi j}{2^{n+1}}} \sum_{k=1}^{2^n - 1} a_k^{(0)} \sin \frac{k\pi j}{2^n}, \quad j = 1, 2, \ldots, 2^{n-1},$$

where $a_k^{(0)}$ is computed from the formula $a_k^{(0)} = a_k + a_{k+1}$, $k = 1, 2, \ldots, 2^n - 1$. Comparing the resulting sum with (22), we find that the problem reduces to the already solved problem 1.

To compute y_{2^n} we obtain the formula

$$y_{2^n} = \sum_{k=1}^{2^n} a_k (-1)^{k-1} = \sum_{k=1}^{2^{n-1}} (a_{2k-1} - a_{2k}).$$

Here the sum is computed directly.

For this algorithm, we obtain the following estimate for the number of operations: $Q = 2N \log_2 N - \log_2 N$.

4.1.3 Expansion in cosines. We look now at an algorithm for solving problem 3, which consists of computing the sums (13) for $N = 2^n$. We have

$$y_k = \sum_{j=0}^{2^n} a_j^{(0)} \cos \frac{k\pi j}{2^n}, \quad k = 0, 1, \ldots, 2^n. \tag{44}$$

where we have introduced the notation $a_j^{(0)} = a_j$.

The principle for constructing the algorithm is exactly the same as for the expansion in sines, and consists of two steps. In the first step, we group together the terms of the sums with indices j and $2^n - j$ for $j = 0, 1, \ldots, 2^{n-1} - 1$, then with indices j and $2^{n-1} - j$, $j = 0, 1, \ldots, 2^{n-2} - 1$, and so forth.

At the end of the p^{th} stage we have

$$\begin{aligned} y_{2^{s-1}(2k-1)} &= \sum_{j=0}^{2^{n-s}-1} a_{2^{n-s+1}-j}^{(s)} \cos \frac{(2k-1)\pi j}{2^{n-s+1}}, \\ &\qquad k = 1, 2, \ldots, 2^{n-s}, \quad s = 1, 2, \ldots, p, \\ y_{2^p k} &= \sum_{j=0}^{2^{n-p}} a_j^{(p)} \cos \frac{k\pi j}{2^{n-p}}, \\ &\qquad k = 0, 1, \ldots, 2^{n-p}. \end{aligned} \tag{45}$$

The formulas are correct for $p = 1, 2, \ldots, n$. The coefficients $a_j^{(p)}$ are defined recursively

$$\begin{aligned} a_j^{(p)} &= a_j^{(p-1)} + a_{2^{n-p+1}-j}^{(p-1)}, \\ a_{2^{n-p+1}-j}^{(p)} &= a_j^{(p-1)} - a_{2^{n-p+1}-j}^{(p-1)}, \\ &\qquad j = 0, 1, \ldots, 2^{n-p} - 1, \\ a_{2^{n-p}}^{(p)} &= a_{2^{n-p}}^{(p-1)}, \\ &\qquad p = 1, 2, \ldots, n. \end{aligned} \tag{46}$$

Substituting $s = p = n$ in (45), we find

$$y_0 = a_0^{(n)} + a_1^{(n)}, \quad y_{2^n} = a_0^{(n)} - a_1^{(n)}, \quad y_{2^{n-1}} = a_2^{(n)}, \tag{47}$$

and the remaining y_k are found from the formulas

$$y_{2^{s-1}(2k-1)} = \sum_{j=0}^{2^{n-s}-1} a^{(s)}_{2^{n-s+1}-j} \cos\frac{(2k-1)\pi j}{2^{n-s+1}},$$
$$k = 1,2,\ldots,2^{n-s}, \quad s = 1,2,\ldots,n-1.$$

Substituting for each fixed s

$$z^{(0)}_k(1) = y_{2^{s-1}(2k-1)}, \quad k = 1,2,\ldots,2^{n-s},$$
$$b^{(0)}_j(1) = a^{(s)}_{2^{n-s+1}-j}, \quad j = 0,1,\ldots,2^{n-s}-1,$$
$$l = n-s, \quad s = 1,2,\ldots,n-1$$

we are led to compute the following sums:

$$z^{(0)}_k(1) = \sum_{j=0}^{2^l-1} b^{(0)}_j(1)\cos\frac{(2k-1)\pi j}{2^{l+1}}, \quad k = 1,2,\ldots,2^l, \tag{48}$$
$$l = 1,2,\ldots,n-1.$$

The second step of the algorithm consists of computing the sums (48). As before, sequentially separating the terms with even and odd indices j, we have the following recurrence relations:

$$z^{(m-1)}_k(s) = z^{(m)}_k(2s) + \frac{1}{2\cos\frac{(2k-1)\pi}{2^{l-m+2}}} z^{(m)}_k(2s-1),$$
$$z^{(m)}_{2^{l-m+1}-k+1}(s) = z^{(m)}_k(2s) + \frac{1}{2\cos\frac{(2k-1)\pi}{2^{l-m+2}}} z^{(m)}_k(2s-1), \tag{49}$$
$$k = 1,2,\ldots,2^{l-m}, \quad s = 1,2,\ldots,2^{m-1}, \quad m = 1,2,\ldots,l$$

for computing

$$z^{(m)}_k(s) = \sum_{j=0}^{2^{l-m}-1} b^{(m)}_j(s)\cos\frac{(2k-1)\pi j}{2^{l-m+1}}, \tag{50}$$
$$k = 1,2,\ldots,2^{l-m}, \quad s = 1,2,\ldots,2^m$$

for $m = 0, 1, \ldots, l$. The coefficients $b_j^{(m)}(s)$ are also defined recursively for $s = 1, 2, \ldots, 2^{m-1}$, starting with $b_j^{(0)}(1)$, using the formulas

$$\begin{aligned} b_j^{(m)}(2s-1) &= b_{2j-1}^{(m-1)}(s) + b_{2j+1}^{(m-1)}(s), \\ &\qquad j = 1, 2, \ldots, 2^{l-m} - 1, \quad m = 1, 2, \ldots, l-1, \\ b_0^{(m)}(2s-1) &= b_1^{(m-1)}(s), \quad m = 1, 2, \ldots, l, \\ b_j^{(m)}(2s) &= b_{2j}^{(m-1)}(s), \\ &\qquad j = 0, 1, \ldots, 2^{l-m} - 1, \quad m = 1, 2, \ldots, l. \end{aligned} \tag{51}$$

Substituting $m = l$ in (50), we find the initial conditions for the relations (49)

$$z_1^{(l)}(s) = b_0^{(l)}(s), \quad s = 1, 2, \ldots, 2^l. \tag{52}$$

Thus, the algorithm for computing the sums (44) is described by the formulas (46), (47), (49), (51), and (52).

An elementary count of the number of arithmetic operations for this algorithm shows: $Q_+ = (3/2n-2)2^n + n + 2$ additions and $Q_* = (n/2-1)2^n + 1$ multiplications, and in total

$$Q = Q_+ + Q_* = (2\log_2 N - 3)N + \log_2 N + 3, \quad N = 2^n.$$

Notice that, as in the preceding algorithm, here it is possible to substitute in (49)

$$z_k^{(m)}(s) = \sin \frac{(2k-1)\pi}{2^{l-m+1}} w_k^{(m)}(s);$$

and then from (52) it follows that $w_1^{(l)}(s) = b_0^{(l)}(s)$, $s = 1, 2, \ldots, 2^l$.

The recurrence relations for $w_k^{(m)}(s)$ have the form

$$\begin{aligned} w_k^{(m-1)}(s) &= 2\cos \frac{(2k-1)\pi}{2^{l-m+2}} w_k^{(m)}(2s) + w_k^{(m)}(2s-1), \\ w_{2^{l-m+1}-k+1}^{(m-1)}(s) &= 2\cos \frac{(2k-1)\pi}{2^{l-m+2}} w_k^{(m)}(s) - w_k^{(m)}(2s-1), \\ &\qquad k = 1, 2, \ldots, 2^{l-m}, \quad s = 1, 2, \ldots, 2^{m-1}, \quad m = 1, 2, \ldots, l. \end{aligned}$$

4.1.4 Transforming a real-valued periodic grid function. Problem 4 involving the transformation of a real-valued periodic grid function consists of regenerating a function using (17) from given Fourier coefficients a_j and $\bar{a}_j$ and of finding the coefficients for a given function using (18).

Suppose $N = 2^n$ and that the Fourier coefficients are given. Then it is necessary to compute the sums

$$y_k = \sum_{j=0}^{2^{n-1}} a_j^{(0)} \cos \frac{2k\pi j}{2^n} + \sum_{j=1}^{2^{n-1}+1} \bar{a}_j^{(0)} \sin \frac{2k\pi j}{2^n}, \quad k = 0, 1, \ldots, 2^n - 1. \quad (53)$$

We now construct the corresponding algorithm. To do this, we change the index in (53) from k to $2^n - k$. Taking into account the identities

$$\cos \frac{2(2^n - k)\pi j}{2^n} = \cos \frac{2k\pi j}{2^n}, \quad \sin \frac{2(2^n - k)\pi j}{2^n} = -\sin \frac{2k\pi j}{2^n},$$

we obtain that y_k can be computed using the formulas

$$\begin{aligned} y_k &= \bar{y}_k + \bar{\bar{y}}_k, \\ y_{2^n-k} &= \bar{y}_k - \bar{\bar{y}}_k, \quad k = 1, 2, \ldots, 2^{n-1} - 1, \\ y_0 &= \bar{y}_0, \quad y_{2^{n-1}} = \bar{y}_{2^{n-1}}, \end{aligned} \quad (54)$$

where

$$\bar{y}_k = \sum_{j=0}^{2^{n-1}} a_j^{(0)} \cos \frac{k\pi j}{2^{n-1}}, \quad k = 0, 1, \ldots, 2^{n-1}, \quad (55)$$

$$\bar{\bar{y}}_k = \sum_{j=1}^{2^{n-1}-1} \bar{a}_j^{(0)} \sin \frac{k\pi k}{2^{n-1}}, \quad k = 1, 2, \ldots, 2^{n-1} - 1. \quad (56)$$

Thus, computing the sums (53) leads to the computation of the sums (55) and (56) and to the sequential use of the formulas (54).

Comparing the formulas (55) and (56) with the formulas (44) and (22), we find that the sums (55) and (56) can be computed using the algorithms in Sections 4.1.2 and 4.1.3, after having changed n to $n - 1$.

We now calculate the number of operations required to compute the sums (53) by this method. From the estimates for the number of operations found for the algorithm in Section 4.1.2, we obtain that the sums (56) can be computed using $Q_+ = (3n/4 - 7/4)2^n - n + 3$ additions and $Q_* = (n/4 - 3/4)2^n + 1$ multiplications. The estimates for the algorithm in Section 4.1.3 give the following values for the sums (55): $Q_+ = (3n/4 - 7/4)2^n + n + 1$

additions and $Q_* = (n/4 - 3/4)2^n + 1$ multiplications. Adding in here the $Q_+ = 2^n - 2$ additions required to compute (54), we obtain for the algorithm $Q_+ = (3n/2-5/2)2^n+2$ additions and $Q_* = (n/2-3/2)2^n+2$ multiplications, and in total $Q = (2\log_2 N - 4)N + 4$, $N = 2^n$.

We turn now to the computation of the Fourier coefficients of a real-valued periodic grid function. The problem consists of finding the sums

$$y_k = \sum_{j=0}^{2^n-1} a_j^{(0)} \cos \frac{2k\pi j}{2^n}, \quad k = 0, 1, \ldots, 2^{n-1}, \tag{57}$$

$$\bar{y}_k = \sum_{j=1}^{2^n-1} a_j^{(0)} \sin \frac{2k\pi j}{2^n}, \quad k = 1, 2, \ldots, 2^{n-1} - 1, \tag{58}$$

where $a_j^{(0)}$ is a given function.

The algorithm for computing (57) and (58) is related to the algorithms in Section 4.1.2 and 4.1.3, but differs in several details. Here in the first step we initially group together the terms of the sums (57) and (58) with indices j and $2^{n-1} + j$ for $j = 0, 1, \ldots, 2^{n-1} - 1$, then the terms with indices j and $2^{n-2} + j$ for $j = 0, 1, \ldots, 2^{n-2} - 1$, and so forth. We will examine in more detail the process of sequentially grouping the terms for a sample sum (57). The transformation of the sums (58) is analogous.

Thus, we represent (57) in the following form:

$$y_k = \sum_{j=0}^{2^{n-1}-1} a_j^{(0)} \cos \frac{2k\pi j}{2^n} + \sum_{j=2^{n-1}}^{2^n-1} a_j^{(0)} \cos \frac{2k\pi j}{2^n}$$

and change the second summation, setting $j = 2^{n-1} + j'$. This gives

$$y_k = \sum_{j=0}^{2^{n-1}-1} \left[a_j^{(0)} + (-1)^k a_{2^{n-1}+j}^{(0)}\right] \cos \frac{2k\pi j}{2^n}, \quad k = 0, 1, \ldots, 2^{n-1}.$$

Denoting

$$\begin{aligned} a_j^{(1)} &= a_j^{(0)} + a_{2^{n-1}+j}^{(0)}, \\ a_{2^{n-1}+j}^{(1)} &= a_j^{(0)} - a_{2^{n-1}+j}, \quad j = 0, 1, \ldots, 2^{n-1} - 1, \end{aligned} \tag{59}$$

we obtain in place of (57) the following sums:

$$\begin{aligned} y_{2k-1} &= \sum_{j=0}^{2^{n-1}-1} a^{(1)}_{2^{n-1}+j} \cos\frac{(2k-1)\pi j}{2^{n-1}}, \quad k = 1,2,\ldots,2^{n-2}, \\ y_{2k} &= \sum_{j=0}^{2^{n-1}-1} a^{(1)}_{j} \cos\frac{2k\pi j}{2^{n-1}}, \quad k = 0,1,\ldots,2^{n-2}. \end{aligned} \tag{60}$$

Analogously in place of (58) we obtain the sums

$$\begin{aligned} \bar{y}_{2k-1} &= \sum_{j=1}^{2^{n-1}-1} a^{(1)}_{2^{n-1}+j} \sin\frac{(2k-1)\pi j}{2^{n-1}}, \quad k = 1,2,\ldots,2^{n-2}, \\ \bar{y}_{2k} &= \sum_{j=1}^{2^{n-1}-1} a^{(1)}_{j} \sin\frac{2k\pi j}{2^{n-1}}, \quad k = 1,2,\ldots,2^{n-2}-1, \end{aligned} \tag{61}$$

where $a^{(1)}_j$ is defined in (59). With this the first step is completed. For the second step, a means of transforming the sums (60) and (61) is described. As a result of the p^{th} step we have

$$\begin{aligned} y_{2^{s-1}(2k-1)} &= \sum_{j=0}^{2^{n-s}-1} a^{(s)}_{2^{n-s}+j} \cos\frac{(2k-1)\pi j}{2^{n-s}}, \\ &\quad k = 1,2,\ldots,2^{n-s-1}, \quad s = 1,2,\ldots,p, \\ y_{2^p k} &= \sum_{j=0}^{2^{n-p}-1} a^{(p)}_{j} \cos\frac{2k\pi j}{2^{n-p}}, \\ &\quad k = 0,1,\ldots,2^{n-p-1}. \end{aligned} \tag{62}$$

where $p = 1,2,\ldots,n-1$ and

$$\begin{aligned} \bar{y}_{2^{s-1}(2k-1)} &= \sum_{j=1}^{2^{n-s}-1} a^{(s)}_{2^{n-s}+j} \sin\frac{(2k-1)\pi j}{2^{n-s}}, \\ &\quad k = 1,2,\ldots,2^{n-s-1}, \quad s = 1,2,\ldots,p, \\ \bar{y}_{2^p k} &= \sum_{j=1}^{2^{n-p}-1} a^{(p)}_{j} \sin\frac{2k\pi j}{2^{n-p}}, \\ &\quad k = 1,2,\ldots,2^{n-p-1}-1, \end{aligned} \tag{63}$$

where $p = 1, 2, \ldots, n-2$. The coefficients $a_j^{(p)}$ are found recursively from the formulas

$$\begin{aligned} a_j^{(p)} &= a_j^{(p-1)} + a_{2^{n-p}+j}^{(p-1)}, \quad j = 0, 1, \ldots, 2^{n-p} - 1, \\ a_{2^{n-p}+j}^{(p)} &= a_j^{(p-1)} - a_{2^{n-p}+j}^{(p-1)}, \quad p = 1, 2, \ldots, n. \end{aligned} \tag{64}$$

Setting $p = n-1$ and $s = p = n-1$ in (62), we obtain

$$\begin{aligned} y_0 &= a_0^{(n-1)} + a_1^{(n-1)} = a_0^{(n)}, \\ y_{2^{n-1}} &= a_0^{(n-1)} - a_1^{(n-1)} = a_1^{(n)}, \\ y_{2^{n-2}} &= a_2^{(n-1)}, \end{aligned} \tag{65}$$

and from (63) for $p = n-2$ we find

$$\bar{y}_{2^{n-2}} = a_1^{(n-2)} - a_3^{(n-2)} = a_3^{(n-1)}. \tag{66}$$

The remaining y_k and $\bar{y}_k$ are found from the formulas

$$y_{2^{s-1}(2k-1)} = \sum_{j=0}^{2^{n-s}-1} a_{2^{n-s}+j}^{(s)} \cos \frac{(2k-1)\pi j}{2^{n-s}},$$

$$\bar{y}_{2^{s-1}(2k-1)} = \sum_{j=1}^{2^{n-s}-1} a_{2^{n-s}+j}^{(s)} \sin \frac{(2k-1)\pi j}{2^{n-s}},$$

$$k = 1, 2, \ldots, 2^{n-s-1}, \quad s = 1, 2, \ldots, n-2.$$

Here we define for fixed s:

$$z_k^{(0)}(1) = y_{2^{s-1}(2k-1)}, \quad \bar{z}_k^{(0)}(1) = \bar{y}_{2^{s-1}(2k-1)},$$

$$k = 1, 2, \ldots, 2^{n-s-1}, \quad b_j^{(0)}(1) = a_{2^{n-s}+j}^{(s)}, \quad j = 0, 1, \ldots, 2^{n-s} - 1,$$

$$l = n - s, \quad s = 1, 2, \ldots, n-2.$$

This leads us to the computation of the sums

$$\begin{aligned} z_k^{(0)}(1) &= \sum_{j=0}^{2^l-1} b_j^{(0)}(1) \cos \frac{(2k-1)\pi j}{2^l}, \\ \bar{z}_k^{(0)}(1) &= \sum_{j=1}^{2^l-1} b_j^{(0)}(1) \sin \frac{(2k-1)\pi j}{2^l}, \\ k &= 1, 2, \ldots, 2^{l-1}, \quad l = 2, 3, \ldots, n-1. \end{aligned} \tag{67}$$

At the second stage of the algorithm, the sums (67) are computed. Here, as in the algorithm in Section 4.1.2, these sums are transformed by separating the terms with even and odd indices j and using the identities

$$\sin\frac{(2k-1)(2j-2)\pi}{2^{l-m+1}}+\sin\frac{(2k-1)2j\pi}{2^{l-m+1}}=2\cos\frac{(2k-1)\pi}{2^{l-m+1}}\sin\frac{(2k-1)(2j-1)\pi}{2^{l-m+1}},$$

$$\cos\frac{(2k-1)(2j-2)\pi}{2^{l-m+1}}+\cos\frac{(2k-1)2j\pi}{2^{l-m+1}}=2\cos\frac{(2k-1)\pi}{2^{l-m+1}}\cos\frac{(2k-1)(2j-1)\pi}{2^{l-m+1}},$$

for $m=1,2,\ldots$. This gives the following recurrence formulas:

$$\begin{aligned}
z_k^{(m-1)}(s) &= z_k^{(m)}(2s)+\frac{1}{2\cos(2k-1)\pi/2^{l-m+1}}z_k^{(m)}(2s-1),\\
z_{2^{-m}-k+1}^{(m-1)}(s) &= z_k^{(m)}(2s)-\frac{1}{2\cos(2k-1)\pi/2^{l-m+1}}z_k^{(m)}(2s-1),\\
\bar{z}_k^{(m-1)}(s) &= \bar{z}_k^{(m)}(2s)+\frac{1}{2\cos(2k-1)\pi/2^{l-m+1}}\bar{z}_k^{(m)}(2s-1),\\
\bar{z}_{2^{l-m}-k+1}^{(m-1)}(s) &= -\bar{z}_k^{(m)}(2s)+\frac{1}{2\cos(2k-1)\pi/2^{l-m+1}}\bar{z}_k^{(m)}(2s-1),
\end{aligned}\tag{68}$$

$$\begin{aligned}
k &= 1,2,\ldots,2^{l-m-1},\\
s &= 1,2,\ldots,2^{m-1},\\
m &= 1,2,\ldots,l-1
\end{aligned}$$

for sequentially computing the sums

$$\begin{aligned}
z_k^{(m)}(s) &= \sum_{j=0}^{2^{l-m}-1} b_j^{(m)}(s)\cos\frac{(2k-1)\pi j}{2^{l-m}},\\
\bar{z}_k^{(m)}(s) &= \sum_{j=1}^{2^{l-m}-1} b_j^{(m)}(s)\sin\frac{(2k-1)\pi j}{2^{l-m}},
\end{aligned}\tag{69}$$

$$k=1,2,\ldots,2^{l-m-1},\quad s=1,2,\ldots,2^m$$

for $m=0,1,\ldots,l-1$.

The coefficients $b_j^{(m)}(s)$ are also defined recursively for $s = 1,2,\ldots,$ 2^{m-1}, starting with $b_j^{(0)}(1)$, using the formulas

$$\begin{aligned} b_j^{(m)}(2s-1) &= b_{2j-1}^{(m-1)}(s) + b_{2j+1}^{(m-1)}(s), \quad j = 1,2,\ldots,2^{l-m}-1, \\ b_0^{(m)}(2s-1) &= b_1^{(m-1)}(s) - b_{2^{l-m+1}-1}^{(m-1)}(s), \\ b_j^{(m)}(2s) &= b_{2j}^{(m-1)}(s), \quad j = 0,1,\ldots,2^{l-m}-1, \\ & s = 1,2,\ldots,2^{m-1}, \quad m = 1,2,\ldots,l-1. \end{aligned} \tag{70}$$

Setting $m = l-1$ in (69), we obtain the initial values for the relations (68).

$$z_1^{(l-1)}(s) = b_0^{(l-1)}(s), \quad \bar{z}_1^{(l-1)}(s) = b_1^{(l-1)}(s), \quad s = 1,2,\ldots,2^{l-1}. \tag{71}$$

Thus, the algorithm for simultaneously computing the sums (57) and (58) is described by the formulas (64)–(66), (68), (70), and (71). Notice that, as in the algorithms in Sections 4.1.2 and 4.1.3, here in the relations (68) it is possible to change

$$\begin{aligned} z_k^{(m)}(s) &= \sin\frac{(2k-1)\pi}{2^{l-m}} w_k^{(m)}(s), \\ \bar{z}_k^{(m)}(s) &= \sin\frac{(2k-1)\pi}{2^{l-m}} \bar{w}_k^{(m)}(s), \end{aligned}$$

which allows us to avoid dividing by $2\cos(2k-1)\pi/2^{l-m+1}$.

An elementary count of the number of arithmetic operations for this algorithm gives: $Q_+ = 3n/2 \cdot 2^n - 1$ additions and $Q_* = (n/2 - 3/2)2^n + 2$ multiplications, and in total $Q = (2\log_2 N - 3/2)N + 1$, $N = 2^n$.

Thus, computing the Fourier coefficients and regenerating a real-valued periodic grid function using this algorithm requires $O(N \ln N)$ arithmetic operations.

4.1.5 Transforming a complex-valued periodic grid function. We look now at problem 5 involving the computation of the Fourier coefficients and the regeneration of a complex periodic grid function. In Section 4.1.1 it was shown that this problem leads to the computation of the sums (21), which in the case $N = 2^n$ have the form

$$y_k = \sum_{j=0}^{2^n-1} a_j^{(0)} \exp\left(\frac{2k\pi j}{2^n}\right), \quad k = 0,1,\ldots,2^n-1, \tag{72}$$

where $a_j^{(0)}$ is a complex number.

The algorithm for computing the sums (72) is constructed in the same way as the algorithm for computing the Fourier coefficients of a real-valued periodic grid function. At the first stage, we group together the terms of the sums (72) with indices j and $2^{n-1}+j$ for $j=0,1,\dots,2^{n-1}-1$, then the terms with indices j and $2^{n-2}+j$ for $j=0,1,\dots,2^{n-2}-1$ and so forth. Taking into account the identity $e^{\pi k i}=(-1)^k$, we obtain at the end of the p^{th} step the following sums:

$$\begin{aligned} y_{2^{s-1}(2k-1)} &= \sum_{j=0}^{2^{n-s}-1} a_{2^{n-s}+j} \exp\left(\frac{(2k-1)\pi j}{2^{n-s}} i\right), \\ & k=1,2,\dots,2^{n-s}, \quad s=1,2,\dots,p, \\ y_{2^p k} &= \sum_{j=0}^{2^{n-p}-1} a_j^{(p)} \exp\left(\frac{2k\pi j}{2^{n-s}} i\right), \\ & k=0,1,\dots,2^{n-p}-1, \end{aligned} \tag{73}$$

where the coefficients $a_j^{(p)}$ are found from the recurrence relations (64).

Setting $s=p=n$ in (73), we have

$$y_0 = a_0^{(n)}, \quad y_{2^{n-1}} = a_1^{(n)}, \tag{74}$$

and the remaining y_k are found from the formulas

$$\begin{aligned} y_{2^{s-1}(2k-1)} &= \sum_{j=0}^{2^{n-s}-1} a_{2^{n-s}+j}^{(s)} \exp\left(\frac{(2k-1)\pi j}{2^{n-s}} i\right), \\ & k=1,2,\dots,2^{n-s}, \quad s=1,2,\dots,n-1. \end{aligned}$$

For fixed j we make the substitutions

$$\begin{aligned} z_k^{(0)}(1) &= y_{2^{s-1}(2k-1)}, & k &= 1,2,\dots,2^{n-s}, \\ b_j^{(0)}(1) &= a_{2^{n-s}+j}^{(s)}, & j &= 0,1,\dots,2^{n-s}-1, \\ l &= n-s, & s &= 1,2,\dots,n-1, \end{aligned}$$

which lead us to the computation of the sums

$$z_k^{(0)}(1) = \sum_{j=0}^{2^l-1} b_j^{(0)}(1) \exp\left(\frac{(2k-1)\pi j}{2^l} i\right), \quad k=1,2,\dots,2^l \tag{75}$$

for $l=1,2,\dots,n-1$.

The second step of the algorithm, involving the computation of the sums (75), is constructed, as before, by separating the terms with even and odd indices j using the identities

$$\exp\left(\frac{(2k-1)(2j-2)\pi}{2^{l-m+1}}i\right)+\exp\left(\frac{(2k-1)2j\pi}{2^{l-m+1}}i\right)$$
$$=2\cos\frac{(2k-1)\pi}{2^{l-m+1}}\exp\left(\frac{(2k-1)(2j-1)\pi}{2^{l-m+1}}i\right).$$

We obtain the recurrence relations

$$\begin{aligned}
z_k^{(m-1)}(s)&=z_k^{(m)}(2s)+\frac{1}{2\cos\frac{(2k-1)\pi}{2^{l-m+1}}}z_k^{(m)}(2s-1),\\
z_{2^{l-m}+k}^{(m-1)}(s)&=z_k^{(m)}(2s)-\frac{1}{2\cos\frac{(2k-1)\pi}{2^{l-m+1}}}z_k^{(m)}(2s-1),\\
&k=1,2,\dots,2^{l-m},\quad s=1,2,\dots,2^{m-1},\\
&m=1,2,\dots,l-1
\end{aligned}\tag{76}$$

for computing the sums

$$\begin{aligned}
z_k^{(m)}(s)&=\sum_{j=0}^{2^{l-m}-1}b_j^{(m)}(s)\exp\left(\frac{(2k-1)\pi j}{2^{l-m}}i\right),\\
&k=1,2,\dots,2^{l-m},\quad s=1,2,\dots,2^m
\end{aligned}\tag{77}$$

for $m=0,1,\dots,l-1$. The coefficients $b_j^{(m)}$ are computed from the recurrence formulas (70). It remains to indicate the initial values for (76). Setting $m=l-1$ in (77), we obtain

$$\begin{aligned}
z_1^{(l-1)}(s)&=b_0^{(l-1)}(s)+ib_1^{(l-1)}(s),\\
z_2^{(l-1)}(s)&=b_0^{(l-1)}(s)-ib_1^{(l-1)}(s),\quad s=1,2,\dots,2^{l-1}.
\end{aligned}\tag{78}$$

Thus, the algorithm for computing the sums (72) is described by the formulas (64), (70), (74), (76), and (78). Notice that this algorithm does not contain (except for the simple formula (78)) any multiplications involving complex numbers. Therefore, it is easy to separate the real and imaginary parts of the computed quantities in the formulas. This is useful when implementing the algorithm on a computer without complex arithmetic. Further, in (76) it is possible to make the following useful change

$$z_k^{(m)}(s)=\sin\frac{(2k-1)\pi}{2^{l-m}}w_k^{(m)}(s).$$

We now compute the number of arithmetic operations for this algorithm. We obtain $Q_+ = (3n/2 - 1/2)2^n$ complex additions and $Q_* = (n/2 - 3/2)2^n$ products of a complex number times a real number. If we express these values in terms of the number of operations on real numbers, we obtain $Q_+ = (3n-1)2^n$ real additions and $Q_* = (n-3)2^n$ real multiplications, and in total $Q = (4\log_2 N - 4)N$, $N = 2^n$ real operations. This estimate is twice as large as the estimate obtained in Section 4.1.4 for the case of a real-valued periodic grid function, which is natural considering that the complex case involves twice as many real numbers.

With this we conclude our look at algorithms for the fast Fourier transform, and move on to using them to solve elliptic grid equations.

4.2 The solution of difference problems by the Fourier method

4.2.1 Eigenvalue difference problems for the Laplace operator in a rectangle. In Section 1.5, we looked at eigenvalue boundary-value problems for the second difference derivative operator defined on a uniform grid on an interval. In the two-dimensional case, the analogs of these problems are eigenvalue problems for the Laplace operator on a uniform rectangular grid in a rectangle. We shall use the method of separation of variables for finding the *eigenvalues* λ_k *and the eigenfunctions* $\mu_k(i,j)$ *of the Laplace difference operator*

$$\Lambda = \Lambda_1 + \Lambda_2, \quad \Lambda_\alpha y = y_{\bar{x}_\alpha x_\alpha}, \quad \alpha = 1,2.$$

Suppose that, in the rectangle $\bar{G} = \{0 \le x_\alpha \le l_\alpha, \alpha = 1,2\}$, we have defined the uniform rectangular grid $\bar{\omega}$ with steps h_1 and h_2: $\bar{\omega} = \{x_{ij} = (ih_1, jh_2) \in \bar{G},\ 0 \le i \le N_1,\ 0 \le j \le N_2,\ h_\alpha N_\alpha = l_\alpha,\ \alpha = 1,2\}$. As usual, we denote by ω the interior, and by γ the boundary, of the grid $\bar{\omega}$.

The simplest eigenvalue problem for the Laplace operator in the case of Dirichlet boundary conditions is: find those values of the parameter λ for which there exists a non-trivial solution $y(x)$ to the following problem:

$$\begin{aligned} \Lambda y(x) + \lambda y(x) &= 0, \quad x \in \omega, \\ y(x) &= 0, \quad x \in \gamma. \end{aligned} \tag{1}$$

We will seek the eigenfunctions $\mu_k(i,j)$ of (1) corresponding to the eigenvalue λ_k in the form

$$\mu_k(i,j) = \mu_{k_1}^{(1)}(i)\mu_{k_2}^{(2)}(j), \quad k = (k_1, k_2). \tag{2}$$

We substitute the function $\mu_k(i,j)$ in place of $y(x_{ij}) = y(i,j)$ in (1). Since

$$\Lambda_1 y(i,j) = \frac{1}{h_1^2}[y(i+1,j) - 2y(i,j) + y(i-1,j)],$$

the operator Λ_1 only acts on a grid function which depends on the argument i. Analogously, the operator Λ_2 acts on a function which depends on the argument j. Therefore, after substituting (2) in (1), we have

$$\mu_{k_2}^{(2)}(j)\Lambda_1\mu_{k_1}^{(1)}(i) + \mu_{k_1}^{(1)}(i)\Lambda_2\mu_{k_2}^{(2)}(j) + \lambda_k\mu_{k_1}^{(1)}(i)\mu_{k_2}^{(2)}(j) = 0 \tag{3}$$

for $1 \le i \le N_1 - 1$, $1 \le j \le N_2 - 1$, and also

$$\mu_{k_1}^{(1)}(0) = \mu_{k_1}^{(1)}(N_1) = 0, \quad \mu_{k_2}^{(2)}(0) = \mu_{k_2}^{(2)}(N_2) = 0. \tag{4}$$

From (3) we find that

$$\frac{\Lambda_1\mu_{k_1}^{(1)}(i)}{\mu_{k_1}^{(1)}(i)} = -\frac{\Lambda_2\mu_{k_2}^{(2)}(j)}{\mu_{k_2}^{(2)}(j)} - \lambda_k. \tag{5}$$

Since the left-hand side does not depend on j, the right-hand side does not depend on j either. On the other hand, since the right-hand side does not depend on i, nor does the left. As a result, the left- and right-hand sides are constants. We set

$$\frac{\Lambda_1\mu_{k_1}^{(1)}(i)}{\mu_{k_1}^{(1)}(i)} = -\lambda_{k_1}^{(1)}, \quad \frac{\Lambda_2\mu_{k_2}^{(2)}(j)}{\mu_{k_2}^{(2)}(j)} = -\lambda_{k_2}^{(2)}, \quad \lambda_k = \lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)} \tag{6}$$

and add in the boundary conditions (4). As a result we obtain one-dimensional grid eigenvalue problems

$$\begin{gathered}\Lambda_1\mu_{k_1}^{(1)} + \lambda_{k_1}^{(1)}\mu_{k_1}^{(1)} = 0, \quad 1 \le i \le N_1 - 1, \\ \mu_{k_1}^{(1)}(0) = \mu_{k_1}^{(1)}(N_1) = 0\end{gathered} \tag{7}$$

and

$$\begin{gathered}\Lambda_2\mu_{k_2}^{(2)} + \lambda_{k_2}^{(2)}\mu_{k_2}^{(2)} = 0, \quad 1 \le j \le N_2 - 1, \\ \mu_{k_2}^{(2)}(0) = \mu_{k_2}^{(2)}(N_2) = 0\end{gathered} \tag{8}$$

The solutions of the problems (7) and (8) were found earlier in Section 1.5:

$$\begin{aligned}\lambda_{k_\alpha}^{(\alpha)} &= \frac{4}{h_\alpha^2}\sin^2\frac{k_\alpha\pi}{2N_\alpha} = \frac{4}{h_\alpha^2}\sin^2\frac{k_\alpha\pi h_\alpha}{2l_\alpha}, && k_\alpha = 1,2,\ldots,N_\alpha - 1, \\ \mu_{k_1}^{(1)}(i) &= \sqrt{\frac{2}{l_1}}\sin\frac{k_1\pi i}{N_1}, && k_1 = 1,2,\ldots,N_1 - 1, \\ \mu_{k_2}^{(2)}(j) &= \sqrt{\frac{2}{l_2}}\sin\frac{k_2\pi j}{N_2}, && k_2 = 1,2,\ldots,N_2 - 1.\end{aligned}$$

Thus, the *eigenfunctions and eigenvalues for the Laplace difference operator* Λ in the case of Dirichlet boundary conditions have been found

$$\mu_k(i,j) = \mu^{(1)}_{k_1}(i)\mu^{(2)}_{k_2}(j) = \frac{2}{\sqrt{l_1 l_2}} \sin\frac{k_1\pi i}{N_1} \sin\frac{k_2\pi j}{N_2},$$
$$0 \le i \le N_1, \quad 0 \le j \le N_2, \tag{9}$$
$$\lambda_k = \lambda^{(1)}_{k_1} + \lambda^{(2)}_{k_2} = \sum_{\alpha=1}^{2} \frac{4}{h_\alpha^2} \sin^2\frac{k_\alpha \pi h_\alpha}{2l_\alpha},$$

where $k_\alpha = 1, 2, \ldots, N_\alpha - 1$, $\alpha = 1, 2$.

We will now examine the basic properties of the eigenfunctions and eigenvalues (9). We define the inner product of grid functions defined on the grid $\bar{\omega}$ in the following way

$$(u,v) = \sum_{x\in\bar{\omega}} u(x)v(x)\hbar_1(x_1)\hbar_2(x_2),$$
$$\hbar_\alpha(x_\alpha) = \begin{cases} 0.5h_\alpha & x_\alpha = 0, l_\alpha \\ h_\alpha, & h_\alpha \le x_\alpha \le l_\alpha - h_\alpha. \end{cases}$$

If we denote

$$(u,v)_{\bar{\omega}_\alpha} = \sum_{x_\alpha\in\bar{\omega}_\alpha} u(x)v(x)\hbar_\alpha(x_\alpha), \quad \alpha = 1,2, \tag{10}$$

where

$$\bar{\omega}_1 = \{x_1(i) = ih_1, \quad 0 \le i \le N_1\},$$
$$\bar{\omega}_2 = \{x_2(j) = jh_2, \quad 0 \le j \le N_2\}$$

then it is clear that $\bar{\omega} = \bar{\omega}_1 \times \bar{\omega}_2$, and $x_{ij} = (x_1(i), x_2(j))$, and besides,

$$(u,v) = ((u,v)_{\bar{\omega}_1}, 1)_{\bar{\omega}_2} = ((u,v)_{\bar{\omega}_2}, 1)_{\bar{\omega}_1}. \tag{11}$$

Recall that in Section 1.5 it was remarked that the grid functions $\mu^{(1)}_{k_1}(i)$ and $\mu^{(2)}_{k_2}(j)$ are orthonormal with respect to the inner product (10), i.e.

$$\left(\mu^{(\alpha)}_{k_\alpha}, \mu^{(\alpha)}_{m_\alpha}\right)_{\bar{\omega}_\alpha} = \delta_{k_\alpha,m_\alpha} = \begin{cases} 1, & k_\alpha = m_\alpha, \\ 0, & k_\alpha \ne m_\alpha. \end{cases}$$

Therefore from this and from (11) it follows that the system of eigenfunctions $\mu_k(i,j)$ defined by the formulas (9) is orthonormal:

$$(\mu_k, \mu_m) = \delta_{k,m} = \begin{cases} 1, & k = m, \\ 0, & k \ne m, \ k = (k_1,k_2), \ m = (m_1,m_2). \end{cases}$$

Since the number of eigenfunctions $\mu_k(i,j) = \mu_{k_1 k_2}(i,j)$ is equal to $(N_1 - 1)(N_2 - 1)$ and this coincides with the number of interior nodes of the grid $\bar{\omega}$, any grid function $f(i,j)$ defined on ω (or defined on $\bar{\omega}$ and reducing to zero on γ), can be represented in the following form:

$$f(i,j) = \sum_{k_1=1}^{N_1-1} \sum_{k_2=1}^{N_2-1} f_{k_1 k_2} \mu_{k_1}^{(1)} \mu_{k_2}^{(2)}(j), \qquad 1 \le i \le N_1 - 1, \quad 1 \le j \le N_2 - 1, \tag{12}$$

where the Fourier coefficients $f_{k_1 k_2}$ are defined in the following fashion:

$$f_k = f_{k_1 k_2} = (f, \mu_k) = \sum_{i=1}^{N_1-1} \sum_{j=1}^{N_2-1} f(i,j) \mu_{k_1}^{(1)}(i) \mu_{k_2}^{(2)}(j) h_1 h_2, \qquad k_1 = 1,2,\ldots,N_1 - 1, \quad k_2 = 1,2,\ldots,N_2 - 1. \tag{13}$$

For the eigenvalues λ_k we have the estimates

$$\lambda_{\min} = \lambda_1^{(1)} + \lambda_2^{(2)} \le \lambda_k = \lambda_{k_1} + \lambda_{k_2} \le \lambda_{N_1-1}^{(1)} + \lambda_{N_2-1}^{(2)} = \lambda_{\max},$$

where

$$\lambda_{\min} = \sum_{\alpha=1}^{2} \frac{4}{h_\alpha^2} \sin^2 \frac{\pi h_\alpha}{2 l_\alpha} \ge 8 \left(\frac{1}{l_1^2} + \frac{1}{l_2^2} \right) > 0,$$

$$\lambda_{\max} = \sum_{\alpha=1}^{2} \frac{4}{h_\alpha^2} \cos^2 \frac{\pi h_\alpha}{2 l_\alpha} < 4 \left(\frac{1}{h_1^2} + \frac{1}{h_2^2} \right).$$

We look now at an example of a more complex eigenvalue problem for the Laplace difference operator. Suppose that, as before, we have Dirichlet boundary conditions on the sides of the rectangle $x_1 = 0$ and $x_1 = l_1$, but that Neumann conditions are given for $x_2 = 0$ and $x_2 = l_2$, i.e., the following eigenvalue problem is given:

$$\Lambda y(x) + \lambda y(x) = 0, \quad x \in \omega_1 \times \bar{\omega}_2, \quad y(x) = 0, \quad x_1 = 0, \quad x_1 = l_1.$$

Here $\Lambda = \Lambda_1 + \Lambda_2$, the operator Λ_1 is defined above, and

$$\Lambda_2 y = \begin{cases} \dfrac{2}{h_2} y_{x_2}, & x_2 = 0, \\ y_{\bar{x}_2 x_2}, & h_2 \le x_2 \le l_2 - h_2, \\ -\dfrac{2}{h_2} y_{\bar{x}_2}, & x_2 = l_2. \end{cases} \tag{15}$$

Using the definition of the operators Λ_1 and Λ_2, the problem (14) can be written in the form:

$$\left.\begin{aligned} y_{\bar{x}_1 x_1} + y_{\bar{x}_2 x_2} + \lambda y = 0, \quad & x \in \omega, \\ y_{\bar{x}_1 x_1} + \frac{2}{h_2} y_{x_2} + \lambda y = 0, \quad & x_2 = 0, \\ y_{\bar{x}_1 x_1} - \frac{2}{h_2} y_{\bar{x}_2} + \lambda y = 0, \quad & x_2 = l_2, \end{aligned}\right\} h_1 \le x_1 \le l_1 - h_1,$$

$$y(0, x_2) = y(l_1, x_2) = 0, \quad 0 \le x_2 \le l_2.$$

The solution of (14) is found by the method of separation of variables. Substituting the grid functions $\mu_k(i,j)$ from (2) in place of y in (14), we obtain for $\mu^{(1)}_{k_1}(i)$ the problem (7), and for $\mu^{(2)}_{k_2}(j)$ we have the following boundary-value problem:

$$\Lambda_2 \mu^{(2)}_{k_2} + \lambda^{(2)}_{k_2} \mu^{(2)}_{k_2} = 0, \quad 0 \le j \le N_2$$

or by (15)

$$\begin{aligned} \left(\mu^{(2)}_{k_2}\right)_{\bar{x}_2 x_2} + \lambda^{(2)}_{k_2} \mu^{(2)}_{k_2} = 0, \quad & 1 \le j \le N_2 - 1, \\ \frac{2}{h_2} \left(\mu^{(2)}_{k_2}\right)_{x_2} + \lambda^{(2)}_{k_2} \mu^{(2)}_{k_2} = 0, \quad & j = 0, \\ -\frac{2}{h_2} \left(\mu^{(2)}_{k_2}\right)_{\bar{x}_2} + \lambda^{(2)}_{k_2} \mu^{(2)}_{k_2} = 0, \quad & j = N_2. \end{aligned} \tag{16}$$

The problem (16) was also solved earlier in Section 1.5. The solution has the form

$$\lambda^{(2)}_{k_2} = \frac{4}{h_2^2} \sin^2 \frac{k_2 \pi}{2N_2} = \frac{4}{h_2^2} \sin^2 \frac{k_2 \pi h_2}{2l_2}, \quad k_2 = 0, 1, \ldots, N_2,$$

$$\mu^{(2)}_{k_2}(j) = \begin{cases} \sqrt{\dfrac{2}{l_2}} \cos \dfrac{k_2 \pi j}{N_2}, & 1 \le k_2 \le N_2 - 1, \\ \sqrt{\dfrac{1}{l_2}} \cos \dfrac{k_2 \pi j}{N_2}, & k_2 = 0, N_2. \end{cases} \tag{17}$$

Thus, the solution of the problem (14), (15) has been found:

$$\begin{aligned} \mu_k(i,j) = \mu^{(1)}_{k_1}(i) \mu^{(2)}_{k_2}(j), \quad & 0 \le i \le N_1, \qquad 0 \le j \le N_2, \\ \lambda_k = \lambda^{(1)}_{k_1} + \lambda^{(2)}_{k_2}, \quad & 1 \le k_1 \le N_1 - 1, \quad 0 \le k_2 \le N_2, \end{aligned}$$

where $\lambda^{(1)}_{k_1}$ and $\mu^{(1)}_{k_1}(i)$ are defined above, and $\lambda^{(2)}_{k_2}$ and $\mu^{(2)}_{k_2}(j)$ are defined in (17).

Eigenvalue problems for the Laplace difference operator in a rectangle with other combinations of boundary conditions on the sides of the rectangle $\bar{G}$ are solved analogously. The method of separation of variables allows us to reduce them to one-dimensional problems which were solved in Section 1.5. The generalization to the multi-dimensional case is obvious. Recall that the analytic solution in the form of sines and cosines of the corresponding one-dimensional problems was obtained in Section 1.5, but only for boundary conditions of first and second order, for their combinations, and also for the case of periodic boundary-value problems. Therefore, if we are given boundary conditions of these types on the sides of a rectangle (or on the boundary of a rectangular parallelipiped in the three-dimensional case), then the eigenfunctions for the Laplace operator can be represented in the form of a product of sines and cosines.

4.2.2 Poisson's equation in a rectangle; expansion in a double series. We look now at the method of separation of variables applied to the solution of a *Dirichlet difference problem for Poisson's equation* on a uniform grid in a rectangle:

$$\begin{aligned} &\Lambda y = -\varphi(x), \quad x \in \omega, \quad y(x) = g(x), \quad x \in \gamma, \\ &\Lambda = \Lambda_1 + \Lambda_2, \quad \Lambda_\alpha y = y_{\bar{x}_\alpha x_\alpha}, \quad \alpha = 1, 2. \end{aligned} \tag{18}$$

We first transform the problem (18) to a problem with homogeneous boundary conditions by changing the right-hand side of the equations at the boundary nodes. The standard way of performing this transformation consists of transferring known quantities to the right-hand side of the equation at a boundary node. For example, if $x = (h_1, h_2) \in \omega$, then Poisson's equation at this point can be written in the following form:

$$\begin{aligned} &\frac{1}{h_1^2}\left[y(0, h_2) - 2y(h_1, h_2) + y(2h_1, h_2)\right] \\ &\quad + \frac{1}{h_2^2}\left[y(h_1, 0) - 2y(h_1, h_2) + y(h_1, 2h_2)\right] = -\varphi(h_1, h_2). \end{aligned}$$

Since $y(0, h_2) = g(0, h_2)$, $y(h_1, 0) = g(h_1, 0)$, by transferring these quantities from the left- to the right-hand side of the equation we obtain

$$\begin{aligned} &\frac{1}{h_1^2}\left[-2y(h_1, h_2) + y\,(2h_1, h_2)\right] + \frac{1}{h_2^2}\left[-2y(h_1, h_2) + y(h_1, 2h_2)\right] \\ &\qquad = -\left[\varphi(h_1, h_2) + \frac{1}{h_1^2} g(0, h_2) + \frac{1}{h_2^2} g(h_1, 0)\right]. \end{aligned}$$

By carrying out a similar transformation at each boundary point, we obtain difference equations which do not contain the values of $y(x)$ on γ in the left-hand side. The right-hand sides of the equations for the boundary nodes differ from the right-hand side $\varphi(x)$. If we denote by $f(x)$ the constructed right-hand side, then it is defined by the formulas

$$f(x) = \varphi(x) + \frac{1}{h_1^2}\varphi_1(x) + \frac{1}{h_2^2}\varphi_2(x), \quad x \in \omega, \tag{19}$$

where

$$\varphi_1(x) = \begin{cases} g(0, x_2), & x_1 = h_1, \\ 0, & 2h_1 \le x_1 \le l_1 - 2h_1, \\ g(l_1, x_2), & x_1 = l_2, \end{cases}$$

$$\varphi_2(x) = \begin{cases} g(x_1, 0), & x_2 = h_2, \\ 0, & 2h_2 \le x_2 \le l_2 - 2h_2, \\ g(x_1, l_2), & x_2 = l_2. \end{cases}$$

The left-hand side of the transformed equations differs from the Laplace difference operator at the boundary nodes. However, if we set $y(x) = u(x)$, $x \in \omega$, $u(x) = 0$, $x \in \gamma$, then the equations at all the nodes of the grid ω can be written identically:

$$\Lambda u = -f(x), \quad x \in \omega, \quad u(x) = 0, \quad x \in \gamma. \tag{20}$$

Since $u(x)$ coincides with $y(x)$ for $x \in \omega$, it suffices to find the solution of (20).

We now find the solution of the problem (20). Since the function $u(x)$ reduces to zero on γ, it is possible to represent it in the form of an expansion in the eigenfunctions $\mu_k(i,j)$ of the Laplace operator

$$u(i,j) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=1}^{N_2-1} u_{k_1 k_2} \mu_{k_1}^{(1)}(i) \mu_{k_2}^{(2)}(j), \tag{21}$$

which is valid for $0 \le i \le N_1$, $0 \le j \le N_2$. Further, the grid function $f(x)$ defined on ω also admits the representation

$$f(i,j) = \sum_{k_1=1}^{N_1-1} \sum_{k_2=1}^{N_2-1} f_{k_1 k_2} \mu_{k_1}^{(1)}(i) \mu_{k_2}^{(2)}(j), \tag{22}$$

for $1 \le i \le N_1 - 1$, $1 \le j \le N_2 - 1$, where the Fourier coefficients $f_{k_1 k_2}$ are defined in (13). Since $\mu_k(i,j) = \mu_{k_1}^{(1)}(i)\mu_{k_2}^{(2)}(j)$ is an eigenfunction of the

Laplace operator corresponding to the eigenvalue λ_k, i.e.,

$$\Lambda\mu_k + \lambda_k\mu_k = 0, \quad x \in \omega, \quad \lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)} = \lambda_k,$$

after substituting (21) and (22) in (20) we get

$$\Lambda u = -\sum_{k_1=1}^{N_1-1}\sum_{k_2=1}^{N_2-1}\left(\lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)}\right) u_{k_1k_2}\mu_{k_1}^{(1)}(i)\mu_{k_2}^{(2)}(j)$$

$$= -f(i,j) = -\sum_{k_1=1}^{N_1-1}\sum_{k_2=1}^{N_2-1} f_{k_1k_2}\mu_{k_1}^{(1)}(i)\mu_{k_2}^{(2)}(j),$$

$$1 \le i \le N_1 - 1, \quad 1 \le j \le N_2 - 1.$$

Using the orthonormality of the eigenfunctions $\mu_k(i,j)$, from this we obtain

$$u_{k_1k_2} = \frac{f_{k_1k_2}}{\lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)}}, \quad 1 \le k_1 \le N_1 - 1, \quad 1 \le k_2 \le N_2 - 1.$$

Substituting this expression in (21), we obtain the following representation for the solution to problem (20):

$$u(i,j) = \sum_{k_1=1}^{N_1-1}\sum_{k_2=1}^{N_2-1} \frac{f_{k_1k_2}}{\lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)}}\mu_{k_1}^{(1)}(i)\mu_{k_2}^{(2)}(j), \quad 0 \le i \le N_1, \quad 0 \le j \le N_2. \tag{23}$$

Thus, the formulas (13) and (23) give the solution to (20). We now analyze them from a computational point of view. In order to compute the solution $u(i,j)$ using the formulas (13) and (20), where $\mu_k(i,j) = \mu_{k_1}^{(1)}(i)\mu_{k_2}^{(2)}(j)$ and $\lambda_k = \lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)}$ are defined in (9), it is convenient to introduce three auxiliary quantities: $\varphi_{k_2}(i)$, $\varphi_{k_1k_2}$ and $u_{k_2}(i)$. Then the computations can be organized as follows:

$$\varphi_{k_2}(i) = \sum_{j=1}^{N_2-1} f(i,j)\sin\frac{k_2\pi j}{N_2}, \tag{24}$$

$$1 \le k_2 \le N_2 - 1, \quad 1 \le i \le N_1 - 1,$$

$$\varphi_{k_1k_2} = \sum_{i=1}^{N_1-1} \varphi_{k_2}(i)\sin\frac{k_1\pi i}{N_1}, \tag{25}$$

$$1 \le k_1 \le N_1 - 1, \quad 1 \le k_2 \le N_2 - 1,$$

$$u_{k_2}(i) = \sum_{k_1=1}^{N_1-1} \frac{\varphi_{k_1 k_2}}{\lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)}} \sin \frac{k_1 \pi i}{N_1}, \tag{26}$$

$$1 \le i \le N_1 - 1, \quad 1 \le k_2 \le N_2 - 1,$$

$$u(i,j) = \frac{4}{N_1 N_2} \sum_{k_2=1}^{N_2-1} u_{k_2}(i) \sin \frac{k_2 \pi j}{N_2}, \tag{27}$$

$$1 \le j \le N_2 - 1, \quad 1 \le i \le N_1 - 1.$$

We now calculate the number of arithmetic operations for the algorithm (24)–(27), assuming that the quantities $\left(\lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)}\right)^{-1}$ are given, and that the sums (24)–(27) are computed using the algorithm for the fast Fourier transform given in Section 4.1.2. In order to apply this algorithm, it is necessary that N_1 and N_2 are powers of 2: $N_1 = 2^n$, $N_2 = 2^m$.

Recall that sums of the form

$$y_k = \sum_{j=1}^{2^n-1} a_j \sin \frac{k\pi j}{2^n}, \quad k = 1, 2, \ldots, 2^n - 1,$$

are computed using $Q_+ = (3/2n - 2)2^n - n + 2$ additions and subtractions, and $Q_* = (n/2 - 1)2^n + 1$ multiplications, if the algorithm in Section 4.1.2 is used.

An elementary count gives the following totals for the number of arithmetic operations involved in computing the solution $u(i,j)$ using the formulas (24)–(27):

$$Q_+ = (N_1 N_2 - N_1 - N_2)[3 \log_2(N_1 N_2) - 8] + (N_1 + 2) \log_2 N_2 + (N_2 + 2) \log_2 N_1 - 8$$

additions and subtractions, and

$$Q_* = (N_1 N_2 - N_1 - N_2)[\log_2(N_1 N_2) - 2] + N_1 \log_2 N_2 + N_2 \log_2 N_1 - 2$$

multiplications. If there is no difference between arithmetic operations, and if $N_1 = N_2 = N = 2^n$, the total number of operations for the algorithm (24)–(27) is

$$Q = (N^2 - 1.5N)(8 \log_2 N - 10) + 5N + 4 \log_2 N - 10.$$

Thus, this method of solving problem (20) can be realized using $O(N^2 \log_2 N)$ arithmetic operations. This type of estimate for the number of operations was also obtained for the cyclic reduction method examined in Chapter 3. A comparison of these estimates shows that the method of separation of variables requires about 1.5 times as many operations as the cyclic reduction method.

Notice that it is also possible to construct an algorithm analogous to the one presented above for the case when a mixture of first- and second-order boundary conditions, or a set of periodic boundary conditions, is given on the sides of the rectangle, if the problem is not singular. All that is necessary is that the corresponding eigenfunctions and eigenvalues for the type of boundary conditions be substituted in (13) and (23), that the limits of the summations be changed, and that the corresponding fast Fourier transform algorithm from Section 4.1 be used to compute the summations which arise. The estimate for the number of operations will be of the same form as for the case of the Dirichlet problem considered above.

We described the simplest variant of the method of separation of variables. If it is necessary to solve a more general boundary-value difference problem, for example Poisson's equation in polar or cylindrical coordinate systems with boundary conditions which assume the separation of variables, then again it is possible to use the expansions (21) and (22). But in this case at least one of the eigenfunctions $\mu_{k_1}^{(1)}(i)$ or $\mu_{k_2}^{(2)}(j)$ will not be sines or cosines. This prevents us using the fast Fourier transform algorithm to compute the necessary sums. Therefore for these problems the number of arithmetic operations will be of the same order as in the case when the sums were computed directly without taking into account the form of the eigenfunctions $\mu_{k_1}^{(1)}(i)$ and $\mu_{k_2}^{(2)}(j)$, i.e., $O(N^3)$.

Consequently, it is necessary to modify the method so that the number of arithmetic operations will remain $O(N^2 \log_2 N)$ when one of the functions $\mu_{k_1}^{(1)}(i)$ or $\mu_{k_2}^{(2)}(j)$ is a sine or a cosine. It is clear that this problem can be solved by a modified method and, as is indicated below, with fewer arithmetic operations. This method — expansion in a single series — will be constructed in Section 4.2.3. From the computational point of view it differs from the method constructed here in that the two sums from (24)–(27) are not computed, but instead a series of boundary-value problems for three-point difference equations is solved.

4.2.3 Expansion in a single series. We turn now to the problem (20):

$$\begin{aligned} &\Lambda u = -f(x), \quad x \in \omega, \quad u(x) = 0, \quad x \in \gamma, \\ &\Lambda = \Lambda_1 + \Lambda_2, \quad \Lambda_\alpha u = u_{\bar{x}_\alpha x_\alpha}, \quad \alpha = 1, 2. \end{aligned} \tag{28}$$

We will consider the desired function $u(x_{ij}) = u(i,j)$ and the given function $f(i,j)$ as grid functions of the argument j for fixed i, $0 \le i \le N_1$. Since $u(i,j)$ reduces to zero for $j = 0$ and $j = N_2$, and $f(i,j)$ is given for $1 \le j \le N_2 - 1$, they can be represented in the form of summations in the eigenfunctions $\mu_{k_2}^{(2)}(j)$ of the difference operator Λ_2:

$$u(i,j) = \sum_{k_2=1}^{N_2-1} u_{k_2}(i)\mu_{k_2}^{(2)}(j), \quad 0 \le j \le N_2, \quad 0 \le i \le N_1, \tag{29}$$

$$f(i,j) = \sum_{k_2=1}^{N_2-1} f_{k_2}(i)\mu_{k_2}^{(2)}(j), \quad 0 \le j \le N_2 - 1, \quad 1 \le i \le N_1 - 1, \tag{30}$$

where

$$\mu_{k_2}^{(2)}(j) = \sqrt{\frac{2}{l_2}} \sin \frac{k_2 \pi j}{N_2}, \quad k_2 = 1, 2, \ldots, N_2 - 1. \tag{31}$$

We substitute the expressions (29) and (30) in (28) and take into account the equations

$$\begin{aligned} &\Lambda_2 \mu_{k_2}^{(2)} + \lambda_{k_2}^{(2)} \mu_{k_2}^{(2)} = 0, \quad 1 \le j \le N_2 - 1, \\ &\mu_{k_2}^{(2)}(0) = \mu_{k_2}^{(2)}(N_2) = 0. \end{aligned} \tag{32}$$

As a result we obtain

$$\sum_{k_2=1}^{N_2-1} \left[\Lambda_1 u_{k_2}(i) - \lambda_{k_2}^{(2)} u_{k_2}(i) + f_{k_2}(i)\right] \mu_{k_2}^{(2)}(j) = 0$$

for $1 \le i \le N_1 - 1$, $1 \le j \le N_2 - 1$, and also $u_{k_2}(0) = u_{k_2}(N_1) = 0$, $k_2 = 1, 2, \ldots, N_2 - 1$.

Hence, using the orthogonality of the system of eigenfunctions $\mu_{k_2}^{(2)}(j)$, we obtain a series of boundary-value problems for determining the functions $u_{k_2}^{(i)} = 1, 2, \ldots, N_2 - 1$:

$$\begin{aligned} &\Lambda_1 u_{k_2}(i) - \lambda_{k_2}^{(2)} u_{k_2}(i) = -f_{k_2}(i), \quad 1 \le i \le N_1 - 1, \\ &u_{k_2}(0) = u_{k_2}(N_1) = 0. \end{aligned} \tag{33}$$

The eigenvalues $\lambda_{k_2}^{(2)}$ for the problem (32) are known

$$\lambda_{k_2}^{(2)} = \frac{4}{h_2^2} \sin^2 \frac{k_2 \pi}{2N_2}, \quad k_2 = 1, 2, \ldots, N_2 - 1, \tag{34}$$

and the Fourier coefficients $f_{k_2}(i)$ for each $1 \le i \le N_1 - 1$ are computed using the formulas

$$f_{k_2}(i) = \left(f, \mu_{k_2}^{(2)}\right)_{\bar{\omega}_2} = \sum_{j=1}^{N_2-1} h_2 f(i,j)\mu_{k_2}^{(2)}(j), \quad 1 \le k_2 \le N_2 - 1. \tag{35}$$

Thus, the formulas (29), (31), and (33)–(35) fully describe a method for solving problem (20). The functions $f_k(i)$ are found for $1 \le i \le N_1 - 1$ using the formulas (35), then the problems (33) are solved for $1 \le k_2 \le N_2 - 1$ to determine the functions $u_{k_2}(i)$, and the desired solution $u(i,j)$ is computed using the formulas (29).

We look now at the algorithm which implements this method. In place of $u_{k_2}(i)$ and $f_{k_2}(i)$, it is convenient to introduce the auxiliary functions $v_{k_2}(i)$ and $\varphi_{k_2}(i)$ using the formulas

$$u_{k_2}(i) = \frac{\sqrt{2l_2}}{N_2} v_{k_2}(i), \quad f_{k_2}(i) = \frac{\sqrt{2l_2}}{N_2} \varphi_{k_2}(i). \tag{36}$$

We substitute (31) and (36) in (29), (33), and (35), take into account that $h_2 N_2 = l_2$, and write out the difference operator Λ_1 at a point. As a result we obtain

$$\varphi_{k_2}(i) = \sum_{j=1}^{N_2-1} f(i,j) \sin \frac{k_2 \pi j}{N_2}, \quad \begin{matrix} 1 \le k_2 \le N_2 - 1, \\ 1 \le i \le N_1 - 1, \end{matrix} \tag{37}$$

$$\left.\begin{matrix} -v_{k_2}(i-1) + \left(2 + h_1^2 \lambda_{k_2}^{(2)}\right) v_{k_2}(i) - v_{k_2}(i+1) = h_1^2 \varphi_{k_2}(i), \\ 1 \le i \le N_1 - 1, \quad v_{k_2}(0) = v_{k_2}(N_1) = 0, \quad 1 \le k_2 \le N_2 - 1, \end{matrix}\right\} \tag{38}$$

$$u(i,j) = \frac{2}{N_2} \sum_{k_2=1}^{N_2-1} v_{k_2}(i) \sin \frac{k_2 \pi j}{N_2}, \quad \begin{matrix} 1 \le j \le N_2 - 1, \\ 1 \le i \le N_1 - 1, \end{matrix} \tag{39}$$

where $\lambda_{k_2}^{(2)}$ is defined in (34).

It is clear that the sums (37) and (39) can be computed using the discrete fast Fourier transform algorithm described in Section 4.1.2. To solve the three-point boundary-value problems (38) it is appropriate to use the elimination algorithm constructed in Section 2.1. For the problem (38), the elimination

algorithm is described by the formulas

$$\begin{aligned}
&\alpha_{i+1} = \frac{1}{c_{k_2} - \alpha_i}, && 1 \le i \le N_1 - 1, && \alpha_1 = 0, \\
&\beta_{i+1} = \left[h_1^2 \varphi_{k_2}(i) + \beta_i\right] \alpha_{i+1}, && 1 \le i \le N_1 - 1, && \beta_1 = 0, \\
&v_{k_2}(i) = \alpha_{i+1} v_{k_2}(i+1) + \beta_{i+1}, && 1 \le i \le N_1 - 1, && v_{k_2}(N_1) = 0,
\end{aligned} \tag{40}$$

where $c_{k_2} = 2 + h_1^2 \lambda_{k_2}^{(2)}$ and $k_2 = 1, 2, \ldots, N_2 - 1$.

We now compare the formulas (37), (39), and (40) with the formulas (24)–(27) obtained earlier for the method involving the expansion in a double series. Here, instead of computing two sums (25) and (26), we solve a series of boundary-value problems (38) using the elimination method (40). Therefore, computing the sums (37) and (39) will require approximately half as many arithmetic operations as the algorithm (24)–(27). Clearly, the additional work to solve the problems (38) is $O(N_1 N_2)$ operations, and this does not affect the principle term in the estimate for the number of arithmetic operations for the algorithm (37), (39), (40). We give now precise estimates for the number of operations for this algorithm. We have (for $N_2 = 2^m$) $Q_{\pm} = [(3 \log_2 N_2 - 1)N_2 - 2 \log_2 N_2 + 1](N_1 - 1)$ additions and subtractions, $Q_* = [(\log_2 N_2 + 2)N_2 - 2](N_1 - 1)$ multiplications and $Q_/ = (N_1 - 1)(N_2 - 1)$ divisions; if $N_1 = N_2 = N = 2^n$, then the total number of operations is equal to

$$Q = (N^2 - 1.5)(4 \log_2 N + 2) - N + 2 \log_2 N + 2.$$

We considered an expansion in a single series for a Dirichlet difference problem for Poisson's equation. An essential fact is that the eigenfunctions for the difference operator Λ_2 allow us to use the fast Fourier transform algorithm to compute the corresponding sums. This will also be possible in the case when the boundary conditions of the first kind are replaced by conditions of the second kind, a mixture of first- and second-kind conditions, or even periodic conditions on the sides $x_2 = 0$ and $x_2 = l_2$ of the rectangle $\bar{G}$.

We look now at the following example of a boundary-value problem for Poisson's equation:

$$\begin{aligned}
&u_{\bar{x}_1 x_1} + u_{\bar{x}_2 x_2} = -\varphi(x), \quad x \in \omega, \\
&u(x) = 0, \quad x_1 = 0, l_1, \quad 0 \le x_2 \le l_2, \\
&u_{\bar{x}_1 x_1} + \frac{2}{h_2} u_{x_2} = -\varphi(x) - \frac{2}{h_2} g_{-2}(x), \quad x_2 = 0, \\
&u_{\bar{x}_1 x_1} - \frac{2}{h_2} u_{\bar{x}_2} = -\varphi(x) - \frac{2}{h_2} g_{+2}(x), \quad x_2 = l_2, \\
&\qquad h_1 \le x_1 \le l_2 - h_1.
\end{aligned} \tag{41}$$

The scheme (41) is a difference approximation for the problem

$$\begin{aligned}
&\frac{\partial^2 u}{\partial x_1^2}+\frac{\partial^2 u}{\partial x_2^2}=-\varphi(x), && x\in G,\\
&u(x)=0, && x_1=0,l_1,\quad 0\le x_2\le l_2,\\
&\frac{\partial u}{\partial x_2}=-g_{-2}(x), && x_2=0,\\
&-\frac{\partial u}{\partial x_2}=-g_{+2}(x), && x_2=l_2,\quad 0\le x_1\le l_1.
\end{aligned}$$

We write out the problem (41) in another form, introducing the notation:

$$\Lambda_2 u=\begin{cases}\dfrac{2}{h_2}u_{x_2}, & x_2=0,\\ u_{\bar{x}_2x_2}, & h_2\le x_2\le l_2-h_2,\\ -\dfrac{2}{h_2}u_{\bar{x}_2}, & x_2=l_2,\end{cases}$$

$$\varphi_2(x)=\begin{cases}\dfrac{2}{h_2}g_{-2}(x), & x_2=0,\\ 0, & h_2\le x_2\le l_2-h_2,\\ \dfrac{2}{h_2}g_{+2}(x), & x_2=l_2,\end{cases}$$

$$f(x)=\varphi(x)+\varphi_2(x),\qquad \Lambda_1 u=u_{\bar{x}_1x_1}$$

for $h_1\le x_1\le l_1-h_1$, $0\le x_2\le l_2$.

In the new notation, the problem (41) is written in the form

$$\begin{aligned}
&\Lambda u=(\Lambda_1+\Lambda_2)u=-f(x),\quad h_1\le x_1\le l_1-h_1,\quad 0\le x_2\le l_2,\\
&u(x)=0,\quad x_1=0,l_1,\quad 0\le x_2\le l_2.
\end{aligned}\tag{42}$$

Expanding $u(i,j)$ and $f(i,j)$ in summations involving the eigenfunctions of the operator Λ_2, we have

$$\begin{aligned}
u(i,j)&=\sum_{k_2=0}^{N_2}u_{k_2}(i)\mu_{k_2}^{(2)}(j),\quad 0\le j\le N_2,\quad 0\le i\le N_1,\\
f(i,j)&=\sum_{k_2=0}^{N_2}f_{k_2}(i)\mu_{k_2}^{(2)}(j),\quad 0\le j\le N_2,\quad 1\le i\le N_1-1,
\end{aligned}\tag{43}$$

where

$$\mu_{k_2}^{(2)}(j) = \begin{cases} \sqrt{\dfrac{1}{l_2}} \cos \dfrac{k_2 \pi j}{N_2}, & k_2 = 0, N_2, \\ \sqrt{\dfrac{2}{l_2}} \cos \dfrac{k_2 \pi j}{N_2}, & 1 \le k_2 \le N_2 - 1, \end{cases}$$

is the eigenfunction of the operator Λ_2 corresponding to the eigenvalue

$$\lambda_{k_2}^{(2)} = \frac{4}{h_2^2} \sin^2 \frac{k_2 \pi}{2N_2}, \quad k_2 = 0, 2, \ldots, N_2. \tag{44}$$

The Fourier coefficient $f_{k_2}(i)$ for each $1 \le i \le N_1 - 1$ is computed using the formulas

$$f_{k_2}(i) = \sum_{j=1}^{N_2 - 1} h_2 f(i,j) \mu_{k_2}^{(2)}(j) + 0.5 h_2 \left[f(i,0) \mu_{k_2}^{(2)}(0) + f(i, N_2) \mu_{k_2}^{(2)}(N_2) \right].$$

Substituting (43) in (42), we obtain the following analog to the formulas (37)–(39) for the problem (42) under consideration:

$$\varphi_{k_2}(i) = \sum_{j=0}^{N_2} \rho_j f(i,j) \cos \frac{k_2 \pi j}{N_2},$$

$$0 \le k_2 \le N_2, \quad 1 \le i \le N_1 - 1,$$

$$-v_{k_2}(i-1) + \left(2 + h_1^2 \lambda_{k_2}^{(2)}\right) v_{k_2}(i) - v_{k_2}(i+1) = h_1^2 \varphi_{k_2}(i),$$

$$1 \le i \le N_1 - 1, \quad v_{k_2}(0) = v_{k_2}(N_1) = 0, \quad 0 \le k_2 \le N_2,$$

$$u(i,j) = \frac{2}{N_2} \sum_{k_2=0}^{N_2} \rho_{k_2} v_{k_2}(i) \cos \frac{k_2 \pi j}{N_2},$$

$$0 \le j \le N_2, \quad 1 \le i \le N_1 - 1,$$

where $\lambda_{k_2}^{(2)}$ is defined in (44), and

$$\rho_j = \begin{cases} 0.5, & j = 0, N_2, \\ 1, & 1 \le j \le N_2 - 1. \end{cases}$$

We give here an estimate of the number of operations for this algorithm when $N_1 = N_2 = N = 2^n$: $Q_\pm = [(3 \log_2 N_2 - 1) N_2 + 2 \log_2 N_2 + 7](N_1 - 1)$ additions and subtractions, $Q_* = [(\log_2 N_2 + 2) N_2 + 10](N_1 - 1)$ multiplications, and $Q_/ = (N_2 + 1)(N_1 - 1)$ divisions, or in total

$$Q = \left(N^2 - \frac{N}{2}\right)(4 \log_2 N + 2) + 17N - 2 \log_2 N - 18.$$

Further since the eigenfunctions of the difference operator Λ_1 are not used for the expansion in a single series, and the only requirement on Λ_1 is that it be possible to separate variables, Λ_1 can be a more general operator than we considered here. If we limit ourselves to second-order elliptic equations, then the most general case corresponds to a difference approximation to the differential operator

$$L_1 u = \frac{1}{k_2(x_1)} \frac{\partial}{\partial x_1} \left(k_1(x_1) \frac{\partial u}{\partial x_1} \right) + r(x_1) \frac{\partial u}{\partial x_1} - q(x_1) u,$$

the coefficients of which depend only on x_1. The boundary conditions on the sides $x_1 = 0$ and $x_1 = l_2$ of the rectangle $\bar{G}$ can be any combination of first-, second- or third-kind boundary conditions (the coefficients in a boundary condition of the third kind must be constants). This allows us to solve boundary-value problems for Poisson's equations in cylindrical, spherical, and polar coordinate systems.

4.3 The method of incomplete reduction

4.3.1 Combining the Fourier and reduction methods. The method constructed in Section 4.2.3 involving an expansion in a single series allows us to compute only two Fourier sums at a cost of $O(N_1 N_2 \log_2 N_2)$ operations and then solve a series of three-point boundary-value problems at a cost of $O(N_1 N_2)$ operations. Clearly, further refinement of the separation of variables method is possible by diminishing the number of terms in the computed sums while still making it possible to use the fast Fourier transform.

We achieve this goal by combining the method involving an expansion in a single series with the reduction method studied in Chapter 3. We first construct this combined method for the simplest Dirichlet problem

$$\begin{gathered} \Lambda u = -f(x), \quad x \in \omega, \quad u(x) = 0, \quad x \in \gamma, \\ \Lambda = \Lambda_1 + \Lambda_2, \quad \Lambda_\alpha u = u_{\bar{x}_\alpha x_\alpha}, \quad \alpha = 1, 2 \end{gathered} \tag{1}$$

on the rectangular grid $\bar{\omega}$.

To simplify the description of the method, we switch from the point (scalar) notation of problem (1) to vector notation.

We introduce the vector of unknowns U_j as follows:

$$U_j = \big(u(1,j), u(2,j), \ldots, u(N_1 - 1, j)\big)^T, \quad 0 \le j \le N_2,$$

and define the right-hand side vector F_j by the formula

$$F_j = \big(h_2^2 f(1,j), h_2^2 f(2,j), \ldots, h_2^2 f(N_1 - 1, j)\big)^T, \quad 1 \le j \le N_2 - 1.$$

Then the difference problem (1) can be written (see Section 3.1) in the form of a system of vector equations

$$\begin{gathered} -U_{j-1} + CU_j - U_{j+1} = F_j, \quad 1 \le j \le N_2 - 1, \\ U_0 = U_{N_2} = 0, \end{gathered} \tag{2}$$

where the square tridiagonal matrix C is defined by

$$\begin{aligned} &CU_j = \big((2E - h_2^2\Lambda_1)u(1,j), \ldots, (2E - h_2^2\Lambda_1)u(N_1 - 1, j)\big)^T, \\ &\Lambda_1 u = u_{\bar{x}_1, x_1}, \quad u(0,j) = u(N_1, j) = 0. \end{aligned}$$

Assume that N_2 is a power of 2 : $N_2 = 2^m$. Recall that the first step in the elimination process for the cyclic reduction method consists (see Section 2.2) in extracting from (2) a "reduced" system for the unknowns U_j with even indices j

$$\begin{gathered} -U_{j-2} + C^{(1)}U_j - U_{j+2} = F_j^{(1)}, \quad j = 2,4,6,\ldots,N_2 - 2, \\ U_0 = U_{N_2} = 0, \end{gathered} \tag{3}$$

and the equations

$$CU_j = F_j + U_{j-1} + U_{j+1}, \quad j = 1,3,5,\ldots,N_2 - 1 \tag{4}$$

for determining the unknowns with odd indices j. Here we denote

$$F_j^{(1)} = F_{j-1} + CF_j + F_{j+1}, \quad j = 2,4,6,\ldots,N_2 - 2, \tag{5}$$

$$C^{(1)} = [C]^2 - 2E. \tag{6}$$

We shall look further at the system (3). We introduce the notation

$$\begin{aligned} v_j &= \big(v(1,j), v(2,j), \ldots, v(N_1 - 1, j)\big)^T, \\ \Phi_j &= \big(h_2^2\varphi(1,j), h_2^2\varphi(2,j), \ldots, h_2^2\varphi(N_1 - 1, j)\big)^T \end{aligned}$$

and set

$$\begin{gathered} v_j = U_{2j}, \quad 0 \le j \le N_2/2, \quad \Phi_j = F_{2j}^{(1)}, \quad 1 \le j \le N_2/2 - 1, \\ v(0,j) = v(N_1, j) = 0, \quad 0 \le j \le N_2/2. \end{gathered}$$

This notation allows us to write the system (3) in the form

$$\begin{gathered} -V_{j-1} + C^{(1)}V_j - V_{j+1} = \Phi_j, \quad j = 1,2,\ldots,M_2 - 1, \\ V_0 = V_{M_2} = 0, \end{gathered} \tag{7}$$

where $2M_2 = N_2$, and by (5)

$$\Phi_j = F_{2j-1} + CF_{2j} + F_{2j+1}, \quad j = 1,2,\ldots,M_2 - 1. \tag{8}$$

Notice now that the grid function $v(i,j)$ is defined for $0 \le i \le N_1$ and $0 \le j \le M_2$, and reduces to zero for $j = 0$ and $j = M_2$. The function $\varphi(i,j)$ is defined for $1 \le i \le N_1 - 1$ and $1 \le j \le M_2 - 1$. Therefore these functions can be represented in the form of a single Fourier series

$$\begin{aligned} v(i,j) &= \sum_{k_2=1}^{M_2-1} y_{k_2}(i)\mu_{k_2}^{(2)}(j), \\ &\qquad 0 \le i \le N_1, \quad 0 \le j \le M_2, \\ \varphi(i,j) &= \sum_{k_2=1}^{M_2-1} z_{k_2}(i)\mu_{k_2}^{(2)}(j), \\ &\qquad 1 \le i \le N_1 - 1, \quad 1 \le j \le M_2 - 1, \end{aligned} \tag{9}$$

where the functions

$$\mu_{k_2}^{(2)}(j) = \frac{2}{\sqrt{l_2}} \sin \frac{k_2 \pi j}{M_2}, \quad k_2 = 1,2,\ldots,M_2 - 1 \tag{10}$$

form an orthonormal system on the grid $\bar{\omega}$ in terms of the inner product

$$(u,v) = \sum_{j=1}^{M_2-1} u(j)v(j)h_2.$$

The Fourier coefficients $z_{k_2}(i)$ for the function $\varphi(i,j)$ are found from the formulas

$$\begin{aligned} z_{k_2}(i) = \left(\varphi, \mu_{k_2}^{(2)}\right) &= \sum_{j=1}^{M_2-1} h_2 \varphi(i,j)\mu_{k_2}^{(2)}(j), \\ 1 \le k_2 \le M_2 - 1, &\quad 1 \le i \le N_1 - 1. \end{aligned} \tag{11}$$

From (9) we obtain the following expansions for the vectors V_j and Φ_j:

$$\begin{aligned} V_j &= \sum_{k_2=1}^{M_2-1} Y_{k_2}\mu_{k_2}^{(2)}(j), \quad 0 \le j \le M_2, \\ \Phi_j &= \sum_{k_2=1}^{M_2-1} h_2^2 Z_{k_2}\mu_{k_2}^{(2)}(j), \quad 1 \le j \le M_2 - 1, \end{aligned} \tag{12}$$

where

$$Y_{k_2} = \left(y_{k_2}(1), y_{k_2}(2), \ldots, y_{k_2}(N_1 - 1)\right)^T,$$
$$Z_{k_2} = \left(z_{k_2}(1), z_{k_2}(2), \ldots, z_{k_2}(N_1 - 1)\right)^T.$$

We substitute (12) in (7) and take into account

$$\mu_{k_2}^{(2)}(j-1) + \mu_{k_2}^{(2)}(j+2) = 2\cos\frac{k_2\pi}{M_2}\mu_{k_2}^{(2)}(j), \quad 1 \le k_2 \le M_2 - 1.$$

We obtain

$$\sum_{k_2=1}^{M_2-1}\left(C^{(1)} - 2\cos\frac{k_2\pi}{M_2}E\right)Y_{k_2}\mu_{k_2}^{(2)}(j) = \sum_{k_2=1}^{M_2-1} h_2^2 Z_{k_2}\mu_{k_2}^{(2)}(j),$$

from which we obtain (using the orthonormality of the system (10))

$$\left(C^{(1)} - 2\cos\frac{k_2\pi}{M_2}E\right)Y_{k_2} = h_2^2 Z_{k_2}, \quad 1 \le k_2 \le M_2 - 1. \tag{13}$$

We use the relation (6) and obtain

$$\begin{aligned} C^{(1)} - 2\cos\frac{k_2\pi}{M_2}E &= [C]^2 - 2\left(1 + \cos\frac{k_2\pi}{M_2}\right)E \\ &= \left(C - 2\cos\frac{k_2\pi}{M_2}E\right)\left(C + 2\cos\frac{k_2\pi}{M_2}E\right). \end{aligned}$$

Since the matrix $C^{(1)} - 2\cos(k_2\pi/M_2)E$ is factored, we can use the following algorithm to solve the equation (13)

$$\begin{aligned} &\left(C - 2\cos\frac{k_2\pi}{2M_2}E\right)W_{k_2} = h_2^2 Z_{k_2}, \\ &\left(C + 2\cos\frac{k_2\pi}{2M_2}E\right)Y_{k_2} = W_{k_2}, \quad 1 \le k_2 \le M_2 - 1, \end{aligned} \tag{14}$$

where the auxiliary vector W_{k_2} has components $w_{k_2}(i)$:

$$W_{k_2} = \left(w_{k_2}(1), w_{k_2}(2), \ldots, w_{k_2}(N_1 - 1)\right)^T,$$
$$w_{k_2}(0) = w_{k_2}(N_1) = 0.$$

The required formulas have been found. Transforming (4), (8), and (14) from vector to scalar notation, and using the relation $u(i, 2j) = v(i, j)$ arising from

the definition of V_j, we obtain the following formulas for this method:

$$\begin{aligned} &\varphi(i,j) = f(i,2j-1) + 2f(i,2j) + f(i,2j+1) - h_2^2\Lambda_1 f(i,2j), \\ &1 \le j \le N_2/2 - 1, \quad 1 \le i \le N_1 - 1, \quad f(0,2j) = f(N_1,2j) = 0 \end{aligned} \tag{15}$$

for computing the function $\varphi(i,j)$; the equations

$$\begin{aligned} &2\left(1 - \cos\frac{k_2\pi}{2M_2}\right) w_{k_2}(i) - h_2^2\Lambda_1 w_{k_2}(i) = h_2^2 z_{k_2}(i), \\ &\qquad 1 \le i \le N_1 - 1, \\ &w_{k_2}(0) = w_{k_2}(N_1) = 0, \\ &2\left(1 + \cos\frac{k_2\pi}{2M_2}\right) y_{k_2}(i) - h_2^2\Lambda_1 y_{k_2}(i) = w_{k_2}(i), \\ &\qquad 1 \le i \le N_1 - 1, \\ &y_{k_2}(0) = y_{k_2}(N_1) = 0, \end{aligned} \tag{16}$$

for defining $y_{k_2}(i)$ for $k_2 = 1,2,\dots,M_2 - 1$; and the equations

$$\begin{aligned} &2u(i,2j-1) - h_2^2\Lambda_1 u(i,2j-1) = h_2^2 f(i,2j-1) + u(i,2j-2) + u(i,2j), \\ &1 \le i \le N_1 - 1, \quad u(0,2j-1) = u(N_1,2j-1) = 0 \end{aligned} \tag{17}$$

for finding the solution for $j = 1,2,\dots,M_2$. For the Fourier coefficients $z_{k_2}(i)$ we have the formula (11), and from (9) we obtain

$$u(i,2j) = \sum_{k_2=1}^{M_2-1} y_{k_2}(i)\mu_{k_2}^{(2)}(j), \quad 1 \le j \le M_2 - 1, \quad 1 \le i \le N_1 - 1. \tag{18}$$

Thus, the formulas (10, (11), (15)–(18) fully describe the method for solving the problem (1), a combination of the Fourier method involving the expansion in a single series and the reduction method.

We move on now to construct the algorithm for the method. In the formulas (9), (16), and (18) we set $y_{k_2}(i) = a\bar{y}_{k_2}(i)$, $w_{k_2}(i) = a\bar{w}_{k_2}(i)$, $z_{k_2}(i) = a\bar{z}_{k_2}(i)$, where $a = 2\sqrt{l_2}/N_2$, and in the resulting formulas we omit the bar. This change allows us to omit the normalizing multiplier $2/\sqrt{l_2}$ for the eigenfunctions $\mu_{k_2}^{(2)}(j)$ in the sums (11) and (18). Further, the problems (16) and (17) will be solved using the elimination method. It is easy to convince oneself that here the conditions for the correctness and stability of the usual elimination method are satisfied. Let us examine the specifics of the problems (17). Since the coefficients of (17) do not depend on j, the eliminations coefficients α_j need only be computed once when solving (17) for $j = 1$, and then used to solve the equations (17) for the remaining j.

We summarize here the computational formulas. First we compute

$$\begin{aligned}\varphi(i,j) =& f(i,2j-1)+f(i,2j+1)\\ &+2\left(1+\frac{h_2^2}{h_1^2}\right)f(i,2j)-\frac{h_2^2}{h_1^2}[f(i-1,2j)+f(i+1,2j)],\\ &1\le j\le M_2-1,\quad 1\le i\le N_1-1,\end{aligned}\tag{19}$$

where $f(0,2j)=f(N_1,2j)=0$. The values of $\varphi(i,j)$ can be overwritten on $f(i,2j)$. The sums

$$z_{k_2}(i)=\sum_{j=1}^{M_2-1}\varphi(i,j)\sin\frac{k_2\pi j}{M_2},\quad 1\le k_2\le M_2-1\tag{20}$$

for $1\le i\le N_1-1$ are computed using the fast Fourier transform, and $z_{k_2}(i)$ is overwritten on $\varphi(i,k_2)$. The elimination method

$$\begin{aligned}&\alpha_{i+1}=1/(c_{k_2}-\alpha_i),\quad \beta_{i+1}=\left[h_1^2 z_{k_2}(i)+\beta_i\right]\alpha_{i+1},\\ &\qquad i=1,2,\dots,N_1-1,\quad \alpha_1=\beta_1=0,\\ &w_{k_2}(i)=\alpha_{i+1}w_{k_2(i+1)}+\beta_{i+1},\\ &\qquad i=N_1-1,N_1-2,\dots,1,\\ &w_{k_2}(N_1)=0,\quad c_{k_2}=2+2\frac{h_1^2}{h_2^2}-2\frac{h_1^2}{h_2^2}\cos\frac{k_2\pi}{N_2}\end{aligned}\tag{21}$$

solves the first of the equations (16), and analogously the formulas

$$\begin{aligned}&\alpha_{i+1}=\frac{1}{c_{k_2}-\alpha_i},\quad \beta_{i+1}=\left[\frac{h_1^2}{h_2^2}w_{k_2}(i)+\beta_i\right]\alpha_{i+1},\\ &\qquad i=1,2,\dots,N_1-1,\quad \alpha_1=\beta_1=0,\\ &y_{k_2}(i)=\alpha_{i+1}y_{k_2}(i+1)+\beta_{i+1},\\ &\qquad i=N_1-1,N_2-1,\dots,1,\\ &y_{k_2}(N_1)=0,\quad c_{k_2}=2+2\frac{h_1^2}{h_2^2}+2\frac{h_1^2}{h_2^2}\cos\frac{k_2\pi}{N_2}\end{aligned}\tag{22}$$

solve the second of the equations (16). Here the computations proceed sequentially for $k_2=1,2,\dots,M_2-1$ and the results $w_{k_2}(i)$ and $y_{k_2}(i)$ are overwritten sequentially on $z_{k_2}(i)$.

To compute the sums

$$u(i,2j)=\frac{4}{N_2}\sum_{k_2=1}^{M_2-1}y_{k_2}(i)\sin\frac{k_2\pi j}{M_2},\quad 1\le j\le M_2-1,\tag{23}$$

for $1 \le i \le N_1 - 1$ we again use the fast Fourier transform. The problems (17) are solved by the elimination method taking into account the specifics of these equations:

$$\begin{aligned}
&\alpha_{i+1} = 1/(c - \alpha_i), \\
&\qquad i = 1, 2, \ldots, N_1 - 1, \quad \alpha_1 = 0, \\
&\beta_{i+1} = \left[h_1^2 f(i, 2j-1) + \frac{h_1^2}{h_2^2}\big(u(i, 2j-2) + u(i, 2j)\big) + \beta_i\right] \alpha_{i+1}, \\
&\qquad i = 1, 2, \ldots, N_1 - 1, \quad \beta_1 = 0, \\
&u(i, 2j-1) = \alpha_{i+1} u(i+1, 2j-1) + \beta_{i+1}, \\
&\qquad i = N_1 - 1, N_1 - 2, \ldots, 1, \quad u(N_1, 2j-1) = 0, \\
&\qquad c = 2(1 + h_1^2/h_2^2)
\end{aligned} \tag{24}$$

for $1 \le j \le M_2$. The solution $u(i, j)$ is overwritten on $f(i, j)$, and consequently, the algorithm does not require auxiliary storage for intermediate information.

We now calculate the number of arithmetic operations for the algorithm (19)–(24). The computations in the formulas (19), (21), (22), and (24) require $Q_\pm = (6.5N_2 - 9)(N_1 - 1)$ additions and subtractions, $Q_* = (6N_2 - 8)(N_1 - 1)$ multiplications and $Q_/ = (N_2 - 1)(N_1 - 1)$ divisions. To compute the sums (20) and (23) we require

$$Q_\pm = \left[\left(\frac{3}{2}\log_2 N_2 - \frac{7}{2}\right) N_2 - 2\log_2 N_2 + 6\right](N_1 - 1)$$

additions and subtractions and

$$Q_* = \left[\left(\frac{1}{2}\log_2 N_2 - 1\right) N_2 + 1\right](N_1 - 1)$$

multiplications. If $N_1 = N_2 = N = 2^n$, then the algorithm (19)–(24) requires in total

$$Q = (N^2 - 2N)(2\log_2 N + 9) - 2N + 2\log_2 N + 11 \tag{25}$$

arithmetic operations.

For comparison, we give here the number of operations for the method involving the expansion in a single series (see Section 4.2.3):

$$Q = \left(N^2 - \frac{3}{2}N\right)(4\log_2 N + 2) - N + 2\log_2 N + 2, \tag{26}$$

for the method involving the expansion in a double series (see Section 4.2.2):

$$Q = \left(N^2 - \frac{3}{2}N\right)(8\log_2 N - 10) + 5N + 4\log_2 N - 10, \qquad (27)$$

and also the number of operations for the second cyclic reduction algorithm (see Section 3.2.4):

$$Q = \left(N^2 - \frac{11}{5}N\right)(5\log_2 N + 5) + N + 6\log_2 N + 5. \qquad (28)$$

If we compare the constants in the principal terms for the estimates (25)–(28), we find that the combined method requires one quarter as many operations as the method involving the expansion in a double series. This result is valid for large N. To obtain a real comparison between these methods for reasonable N, we give here a table of values of Q for these methods.

Table 4

Estimate / N	(25)	(26)	(27)	(28)
32	18,383	21,496	29,510	28,541
64	83,601	104,950	152,334	138,537
128	371,515	485,708	745,582	643,921

Thus, the combination of the Fourier and reduction methods allows us to reduce the number of operations in comparison with the original method involving the expansion in a single series. We can generalize this combined method by including in it l elimination steps from the reduction method before carrying out the expansion in a single series. Then the method for Section 4.2.3 can be treated as a special case of such a generalized method for $l = 0$, and the method constructed in this section corresponds to $l = 1$. The cyclic reduction method can be considered as this method with $l = \log_2 N_2$.

The data in Table 4 indicate that there is an optimal generalized method from the point of view of operation counts for some $1 \leq l \leq \log_2 N_2$. Analyzing the operation counts for the method involving l elimination steps shows that the optimal value is $l = 1$ or $l = 2$. Here, the insignificant improvement in the operation count for the method with $l = 2$ can be lost due to the increased complexity of the algorithm.

4.3.2 The solution of boundary-value problems for Poisson's equations in a rectangle. We look now at an application of the method constructed in Section 4.3.1 for finding the solution to boundary-value problems for Poisson's equation in a rectangle. Suppose that, in the region $\bar{G} = \{0 \le x_\alpha \le l_\alpha, \alpha = 1, 2\}$, it is necessary to find the solution to the equations

$$\frac{\partial^2 v}{\partial x_1^2} + \frac{\partial^2 v}{\partial x_2^2} = -\varphi(x), \quad x \in G, \tag{29}$$

satisfying the following boundary conditions on the boundary Γ of the rectangle $\bar{G}$:

$$\begin{aligned}
&\frac{\partial v}{\partial x_1} = \kappa_{-1} v - g_{-1}(x_2), && x_1 = 0, \\
&-\frac{\partial v}{\partial x_1} = \kappa_{+1} v - g_{+1}(x_2), && x_1 = l_1, \quad 0 \le x_2 \le l_2, \\
&\frac{\partial v}{\partial x_2} = -g_{-2}(x_1), && x_2 = 0, \\
&-\frac{\partial v}{\partial x_2} = -g_{+2}(x_1), && x_2 = l_2, \quad 0 \le x_1 \le l_1,
\end{aligned} \tag{30}$$

where $\kappa_{+1} \ge 0$, $\kappa_{-1} \ge 0$, $\kappa_{+1}^2 + \kappa_{-1}^2 > 0$.

We will assume that κ_{+1} and κ_{-1} are constants in the condition (30). Under this assumption, the variables in the problem (29), (30) are separated.

On the rectangular grid $\bar{\omega} = \{x_{ij} = (ih_1, jh_2) \in \bar{G}, 0 \le i \le N_1, 0 \le j \le N_2, h_\alpha N_\alpha = l_\alpha, \alpha = 1, 2\}$, the problem (29)–(30) corresponds to the difference scheme

$$\Lambda u = (\Lambda_1 + \Lambda_2) u = -f(x), \quad x \in \bar{\omega}, \tag{31}$$

where $f(x) = \varphi(x) + \varphi_1(x) + \varphi_2(x)$,

$$\Lambda_1 u = \begin{cases} \dfrac{2}{h_1}(u_{x_1} - \kappa_{-1} u), & x_1 = 0, \\ u_{\bar{x}_1 x_1}, & h_1 \le x_1 \le l_1 - h_1, \\ \dfrac{2}{h_1}(u_{\bar{x}_1} - \kappa_{+1} u), & x_1 = l_1; \end{cases}$$

$$\Lambda_2 u = \begin{cases} \dfrac{2}{h_2} u_{x_2}, & x_2 = 0, \\ u_{\bar{x}_2 x_2}, & h_2 \le x_2 \le l_2 - h_2, \\ -\dfrac{2}{h_2} u_{\bar{x}_2}, & x_2 = l_2; \end{cases}$$

and the functions $\varphi_\alpha(x)$ are defined by the relation

$$\varphi_\alpha(x) = \begin{cases} \dfrac{2}{h_\alpha} g_{-\alpha}(x_\beta), & x_\alpha = 0, \\ 0, & h_\alpha \le x_\alpha \le l_\alpha - h_\alpha, \quad \beta = 3 - \alpha, \quad \alpha = 1, 2, \\ \dfrac{2}{h_\alpha} g_{+\alpha}(x_\beta), & x_\alpha = l_\alpha. \end{cases}$$

In Chapter 3 it was shown that the scheme (31) has the following structure in vector form:

$$\begin{aligned} CU_0 - 2U_1 &= F_0, \\ -U_{j-1} + CU_j - U_{j+1} &= F_j, \quad 1 \le j \le N_2 - 1, \\ -2U_{N_2-1} + CU_{N_2} &= F_{N_2}, \end{aligned} \tag{32}$$

where

$$\begin{aligned} U_j &= \big(u(0,j), u(1,j), \ldots, u(N_1,j)\big)^T, \\ F_j &= \big(h_2^2 f(0,j), h_2^2 f(1,j), \ldots, h_2^2 f(N_1,j)\big)^T, \\ CU_j &= \big((2E - h_2^2\Lambda_1)u(0,j), \ldots, (2E - h_2^2\Lambda_1)u(N_1,j)\big)^T, \\ & \qquad 0 \le j \le N_2. \end{aligned}$$

The vector system (32) differs from the system (2) considered earlier in the boundary conditions and in the definition of the matrix C. Nevertheless, constructing the analog of the method in Section 4.3.1 for the problem (32) does not present any difficulty. Since the derivation of the basic formulas for this method differs only in details from the development in Section 4.3.2, we will limit ourselves to a summary of the principal intermediate and final formulas. For the cyclic reduction method, the necessary formulas are described in Section 3.4.

Thus, after one step of elimination, we will have the following problem for the vectors $V_j = U_{2j}$, $0 \le j \le M_2$, where $2M_2 = N_2$,

$$\begin{aligned} C^{(1)}V_0 - 2V_1 &= \Phi_0, \\ -V_{j-1} + C^{(1)}V_j - V_{j+1} &= \Phi_j, \quad 1 \le j \le M_2 - 1, \\ -2V_{M_2-1} + C^{(1)}V_{M_2} &= \Phi_{M_2}, \end{aligned} \tag{33}$$

and where the right-hand side $\Phi_j = F_{2j}^{(1)}$, $0 \le j \le M_2$ is defined by the formulas

$$\Phi_j = \begin{cases} CF_0 + 2F_1, & j = 0, \\ F_{2j-1} + CF_{2j} + F_{2j+1}, & 1 \le j \le M_2 - 1, \\ CF_{N_2} + 2F_{N_2-1}, & j = M_2. \end{cases}$$

For the vectors V_j and Φ_j we have the expansions

$$V_j = \sum_{k_2=0}^{M_2} Y_{k_2} \mu_{k_2}^{(2)}(j), \quad \Phi_j = \sum_{k_2=0}^{M_2} h_2^2 Z_{k_2} \mu_{k_2}^{(2)}(j), \quad 0 \le j \le M_2,$$

where

$$\mu_{k_2}^{(2)}(j) = \begin{cases} \dfrac{2}{\sqrt{l_2}} \cos \dfrac{k_2 \pi j}{M_2}, & 1 \le k_2 \le M_2 - 1, \\ \sqrt{\dfrac{1}{l_2}} \cos \dfrac{k_2 \pi j}{M_2}, & k_2 = 0, M_2. \end{cases}$$

By (33), the Fourier coefficients of the vectors V_j and Φ_j are linked by the relation

$$\left(C^{(1)} - 2 \cos \frac{k_2 \pi}{M_2} E \right) Y_{k_2} = h_2^2 Z_{k_2}, \quad 0 \le k_2 \le M_2,$$

and the components of the vector Z_{k_2} can be expressed in terms of the components of the vector Φ_j in the following way

$$Z_{k_2}(i) = \sum_{j=1}^{M_2-1} h_2 \varphi(i,j) \mu_{k_2}^{(2)}(j) + 0.5 h_2 \left[\varphi(i,0) \mu_{k_2}^{(2)}(0) + \varphi(i, M_2) \mu_{k_2}^{(2)}(M_2) \right], \quad 0 \le i \le N_1.$$

The unknowns U_j with odd indices j are determined, as before, from the equations (4).

In these formulas, it remains to convert to scalar notation and to the unnormalized eigenfunctions $\bar{\mu}_{k_2}^{(2)}(j) = \cos \frac{k_2 \pi j}{M_2}$.

As a result, we obtain the following formulas for solving problem (31): for each $0 \le i \le N_1$ we compute

$$\varphi(i,j) = \begin{cases} 2[f(i,0) + f(i,1)] - h_2^2 \Lambda_1 f(i,0), & j = 0, \\ f(i,2j-1) + f(i,2j+1) & \\ \quad + 2f(i,2j) - h_2^2 \Lambda_1 f(i,2j), & 1 \le j \le M_2 - 1, \\ 2[f(i,N_2) + f(i,N_2-1)] - h_2^2 \Lambda_1 f(i,N_2), & j = M_2, \end{cases}$$

and solve the equations

$$4\sin^2\frac{k_2\pi}{2N_2}w_{k_2}(i) - h_2^2\Lambda_1 w_{k_2}(i) = h_2^2 z_{k_2}(i), \quad 0 \le i \le N_1$$
$$4\cos^2\frac{k_2\pi}{2N_2}y_{k_2}(i) - h_2^2\Lambda_1 y_{k_2}(i) = w_{k_2}(i), \quad 0 \le i \le N_1$$

for $0 \le k_2 \le M_2$, where

$$z_{k_2}(i) = \sum_{j=0}^{M_2} \rho_j \varphi(i,j)\cos\frac{k_2\pi j}{M_2},$$
$$0 \le k_2 \le M_2, \quad 0 \le i \le N_1.$$

The solution $u(i,j)$ of problem (31) is determined from the formulas

$$u(i,2j) = \sum_{k_2=0}^{M_2} \rho_{k_2} y_{k_2}(i)\cos\frac{k_2\pi j}{M_2}, \quad 0 \le j \le M_2, \quad 0 \le i \le N_1$$

and from the equations

$$2u(i,2j-1) - h_2^2\Lambda_1 u(i,2j-1) = h_2^2 f(i,2j-1) + u(i,2j-2) + u(i,2j),$$
$$1 \le j \le M_2, \quad 0 \le i \le N_1.$$

Here we use the notation

$$\rho_j = \begin{cases} 1, & 1 \le j \le M_2 - 1, \\ 0.5, & j = 0, M_2, \quad M_2 = 0.5N_2, \end{cases}$$

and the operator Λ_1 is defined above. In order to find $w_{k_2}(i)$, $y_{k_2}(i)$, and $u(i,2j-1)$, we use here three-point equations with third-kind boundary conditions, which we solve by the elimination method.

Notice that these formulas are not affected if the grid is non-uniform in the direction x_1. Only the form of the operator Λ_1 is changed: it will be the difference analog of the second derivative and the third-kind boundary conditions on the non-uniform grid.

It should generally be noted that it is possible to construct variants of the separation of variables method using cyclic reduction and achieve an operation count of $O(N^2 \log_2 N)$ in all but one case. The exception occurs in the case when a third-kind boundary condition is only given on one side of the rectangle in the direction in which the unknowns are being eliminated.

4.3.3 A high-accuracy Dirichlet difference problem in a rectangle. We will look now at one more sample application of the separation of variables method. Suppose that, on the rectangular grid $\bar{\omega}$, we are required to solve a *high-accuracy Dirichlet difference problem for Poisson's equation*

$$\begin{aligned} \Lambda u = \left(\Lambda_1 + \Lambda_2 + \frac{h_1^2 + h_2^2}{12}\Lambda_1\Lambda_2\right) u = -f(x), \quad x \in \omega, \\ u(x) = 0, \quad x \in \gamma, \end{aligned} \tag{34}$$

where $\Lambda_\alpha u = u_{\bar{x}_\alpha x_\alpha}$, $\alpha = 1, 2$. For simplicity, homogeneous boundary conditions are given — a problem with non-homogeneous boundary conditions can be reduced to (34) by changing the right-hand side at the boundary nodes.

In Section 3.1.4, we obtained a vector version of problem (34) in the following form:

$$\begin{aligned} -BU_{j-1} + AU_j - BU_{j+1} = F_j, \quad 1 \le j \le N_2 - 1, \\ U_0 = U_{N_2} = 0, \end{aligned} \tag{35}$$

where

$$U_j = \left(u(1,j), u(2,j), \ldots, u(N_1 - 1, j)\right)^T, \quad 0 \le j \le N_2,$$

$$F_j = \left(h_2^2 f(1,j), h_2^2 f(2,j), \ldots, h_2^2 f(N_1 - 1, j)\right)^T, \quad 1 \le j \le N_2 - 1,$$

and where the matrices B and A are defined by the relations

$$BU_j = \left(\left(E + \frac{h_1^2 + h_2^2}{12}\Lambda_1\right)u(1,j), \ldots, \left(E + \frac{h_1^2 + h_2^2}{12}\Lambda_1\right)u(N_1 - 1, j)\right)^T,$$

$$AU_j = \left(\left(2E - \frac{5h_2^2 - h_1^2}{6}\Lambda_1\right)u(1,j), \ldots, \left(2E - \frac{5h_2^2 - h_1^2}{6}\Lambda_1\right)u(N_1 - 1, j)\right)^T.$$

The matrices A and B commute, i.e., $AB = BA$.

We now construct a combined method for separating the variables in (34). Initially we perform one elimination step from the reduction method for the system (35). We give here a description of this step which is independent of the presentation in Chapter 3. We write out three consecutive equations from the system (35) for $j = 2, 4, 6, \ldots, N_2 - 2$:

$$\begin{aligned} -BU_{j-2} + AU_{j-1} - BU_j = F_{j-1}, \\ -BU_{j-1} + AU_j - BU_{j+1} = F_j, \\ -BU_j + AU_{j+1} - BU_{j+2} = F_{j+1}, \end{aligned}$$

multiply the first and third equations on the left by B, the middle by A, and add them. Using the commutativity of A and B we obtain

$$-B^2U_{j-2} + (A^2 - 2B^2)U_j - B^2U_{j+2} = F_j^{(1)},$$
$$j = 2,4,6,\ldots,N_2 - 2,$$
$$U_0 = U_{N_2} = 0,$$

where $F_j^{(1)} = B(F_{j-1} + F_{j+1}) + AF_j$, $j = 2,4,6,\ldots,N_2 - 2$. Noticing as usual that $V_j = U_{2j}$, $0 \le j \le M_2$, $\Phi_j = F_{2j}^{(1)}$, $1 \le j \le M_2 - 1$, where $2M_2 = N_2$, we write this system in the form

$$-B^2V_{j-1} + (A^2 - 2B^2)V_j - B^2V_{j+1} = \Phi_j, \quad 1 \le j \le M_2 - 1,$$
$$V_0 = V_{M_2} = 0, \tag{36}$$

where

$$\Phi_j = B(F_{2j-1} + F_{2j+1}) + AF_{2j}, \quad 1 \le j \le M_2 - 1. \tag{37}$$

The remaining unknown vectors are found from the equations

$$AU_{2j-1} = F_{2j-1} + B(U_{2j-2} + U_{2j}), \quad 1 \le j \le M_2. \tag{38}$$

As before, the "reduced" system (36) will be solved by the Fourier method. We substitute the expansions (12) in (36), where $\mu_{k_2}^{(2)}(j)$ is defined in (10). As a result, we obtain the following equation for the Fourier coefficients Y_{k_2} and Z_{k_2} of the vectors V_j and Φ_j

$$\left(A^2 - 4\cos^2\frac{k_2\pi}{2M_2}B^2\right)Y_{k_2} = h_2^2 Z_{k_2}, \quad 1 \le k_2 \le M_2 - 1, \tag{39}$$

which is analogous to the equation (13), where the components of the vectors Z_{k_2} and Φ_j are connected by the formula (11). To solve equation (39), it is possible to use the algorithm

$$\left(A - \cos\frac{k_2\pi}{2M_2}B\right)W_{k_2} = h_2^2 Z_{k_2},$$
$$\left(A + \cos\frac{k_2\pi}{2M_2}B\right)Y_{k_2} = W_{k_2}, \quad 1 \le k_2 \le M_2 - 1. \tag{40}$$

Thus, the method for solving problem (34) in vector form is described by the formulas (37), (11), (40), (12), and (38). Converting to scalar notation and to the unnormalized eigenfunctions $\bar{\mu}_{k_2}^{(2)}(j) = \sin\frac{k_2\pi j}{M_2}$ using the change in notation from Section 4.3.1, we obtain the following formulas:

$$\varphi(i,j) = \left(E + \frac{h_1^2+h_2^2}{12}\Lambda_1\right)[f(i,2j-1) + f(i,2j+1) + f(i,2j)]$$
$$-h_2^2\Lambda_1 f(i,2j), \quad 1 \le j \le M_2 - 1, \quad 1 \le i \le N_1 - 1, \tag{41}$$
$$f(0,j) = 0, \quad 1 \le j \le N_1 - 1$$

for computing $\varphi(i,j)$; the equations

$$4\sin^2\frac{k_2\pi}{2N_2}w_{k_2}(i) - h_2^2\left(1 - \frac{4}{h_2^2}\sin^2\frac{k_2\pi}{2N_2}\cdot\frac{h_1^2+h_2^2}{12}\right)\Lambda_1 w_{k_2}(i) = h_2^2 z_{k_2}(i),$$
$$1 \le i \le N_1 - 1, \quad w_{k_2}(0) = w_{k_2}(N_1) = 0 \tag{42}$$

for computing $w_{k_2}(i)$; and

$$4\cos^2\frac{k_2\pi}{2N_2}y_{k_2}(i) - h_2^2\left(1 - \frac{4}{h_2^2}\cos^2\frac{k_2\pi}{2N_2}\cdot\frac{h_1^2+h_2^2}{12}\right)\Lambda_1 y_{k_2}(i) = w_{k_2}(i),$$
$$1 \le i \le N_1 - 1, \quad y_{k_2}(0) = y_{k_2}(N_2) = 0 \tag{43}$$

for computing $y_{k_2}(i)$, where $1 \le k_2 \le M_2 - 1$, and where

$$z_{k_2}(i) = \sum_{j=1}^{M_2-1}\varphi(i,j)\sin\frac{k_2\pi j}{M_2}, \quad 1 \le k_2 \le M_2 - 1, \quad 1 \le i \le N_1 - 1. \tag{44}$$

The solution $u(i,j)$ to (34) is determined from the formulas

$$u(i,2j) = \frac{4}{N_2}\sum_{k_2=1}^{M_2-1} y_{k_2}(i)\sin\frac{k_2\pi j}{M_2}, \quad 1 \le j \le M_2 - 1, \quad 1 \le i \le N_1 - 1, \tag{45}$$

and from the equations

$$2u(i,2j-1) - \frac{5h_2^2 - h_1^2}{6}\Lambda_1 u(i,2j-1)$$
$$= h_2^2 f(i,2j-1) + \left(E + \frac{h_1^2+h_2^2}{12}\Lambda_1\right)[u(i,2j-2) + u(i,2j)], \tag{46}$$
$$1 \le i \le N_1 - 1,$$
$$u(0,2j-1) = u(N_1,2j-1) = 0, \quad 1 \le j \le M_2.$$

It remains for us to show that the three-point equations (42), (43), and (46) are soluble. Then either the usual elimination method or the non-monotonic elimination method can be used to solve them.

It is sufficient to show that the eigenvalues of the difference operator

$$\mathcal{R} = \lambda_{k_2}^{(2)} E - \left(1 - \frac{h_1^2 + h_2^2}{12}\lambda_{k_2}^{(2)}\right)\Lambda_1, \quad \lambda_{k_2}^{(2)} = \frac{4}{h_2^2}\sin^2\frac{k_2\pi}{2N_2}$$

are non-zero for $1 \le k_2 \le N_2 - 1$. In fact, for $1 \le k_2 \le N_2/2 - 1$, the operator $h_2^2\mathcal{R}$ is the same as the operator for problem (42), and for $k_2 = N_2/2$, it is the same as the operator for problem (46). If $N_2/2 + 1 \le k_2 \le N_2 - 1$, then the operator $h_2^2\mathcal{R}$ has the form

$$h_2^2\mathcal{R} = 4\sin^2\frac{k_2\pi}{2N_2} - h_2^2\left(1 - \frac{h_1^2 + h_2^2}{12}\frac{4}{h_2^2}\sin^2\frac{k_2\pi}{2N_2}\right)\Lambda_1.$$

The change $k_2 = N_2 - k_2'$ gives

$$h_2^2\mathcal{R} = 4\cos^2\frac{k_2'\pi}{2N_2} - h_2^2\left(1 - \frac{h_1^2 + h_2^2}{12}\frac{4}{h_2^2}\cos^2\frac{k_2'\pi}{2N_2}\right)\Lambda_1,$$

where $1 \le k_2' \le N_2 - 1$, i.e., in this case the operator $h_2^2\mathcal{R}$ is the same as the operator in (43).

We now find the eigenvalues of the operator $\mathcal{R}$ for a fixed value of k_2. Since the eigenvalues of the operator Λ_1 for the case of boundary conditions of the first kind are (see Section 1.5)

$$\lambda_{k_1}^{(1)} = \frac{4}{h_1^2}\sin^2\frac{k_1\pi}{2N_1}, \quad k_1 = 1, 2, \ldots, N_1 - 1,$$

the eigenvalues λ of the operator $\mathcal{R}$ are

$$\lambda_{k_1k_2} = \lambda_{k_1}^{(1)}\lambda_{k_2}^{(2)} - \frac{h_1^2 + h_2^2}{12}\lambda_{k_1}^{(1)}\lambda_{k_2}^{(2)}, \quad 1 \le k_1 \le N_1 - 1, \quad 1 \le k_2 \le N_2 - 1.$$

Since we have the following estimates for the eigenvalues $\lambda_{k_1}^{(1)}$ and $\lambda_{k_2}^{(2)}$:

$$0 < \lambda_{k_\alpha}^{(\alpha)} < \frac{4}{h_\alpha^2}, \quad \alpha = 1, 2,$$

it is easy to show that for any k_1 and k_2

$$\lambda_{k_1k_2} = \lambda_{k_1}^{(1)}\left(1 - \frac{h_2^2}{12}\lambda_{k_2}^{(2)}\right) + \lambda_{k_2}^{(2)}\left(1 - \frac{h_1^2}{12}\lambda_{k_1}^{(1)}\right) > \frac{2}{3}\left(\lambda_{k_1}^{(1)} + \lambda_{k_2}^{(2)}\right) > 0,$$

which is what we were required to prove.

For problem (42), it is easily seen that the sufficient condition for applicability of the usual elimination method has the form

$$1 + \frac{2h_1^2 - h_2^2}{3h_2^2} \sin^2 \frac{k_2 \pi}{2N_2} \geq 0 \tag{47}$$

and clearly this is satisfied for any k_2. For problem (43), the analogous condition has the form

$$1 + \frac{2h_1^2 - h_2^2}{3h_2^2} \cos^2 \frac{k_2 \pi}{2N_2} \geq 0$$

and this is satisfied for all k_2. For problem (46), the corresponding condition is (47) with $k_2 = 0.5N_2$. Consequently, problems (42), (43), and (46) can be solved by the usual elimination method.

4.4 The staircase algorithm and the reduction method for solving tridiagonal systems of equations

4.4.1 The staircase algorithm for the case of tridiagonal matrices with scalar elements. We write a tridiagonal system in the form of a three-point boundary-value difference problem:

$$-y_{i-1} + Cy_i - y_{i+1} = F_i,\ 1 \leq i \leq N-1,\ y_0 = 0,\ y_N = 0, \tag{1}$$

where C is a scalar, and we assume that $N = 2k + 1$. If the second-order difference equation (1) is written as a recurrence relation

$$y_{i+1} = Cy_i - y_{i-1} - F_i,\ i \geq 1,\ y_0 = 0, \tag{2}$$

then it is not difficult to notice that all the unknowns y_i can be written recursively using (2), given the value of y_1. Thus any y_i can be expressed linearly in terms of y_0 and y_1. This allows us to write

$$y_{i+1} = \alpha_i y_1 - \beta_{i-1} y_0 - p_i \tag{3}$$

for any $i \geq 1$, with as yet undetermined coefficients α_i, β_i, p_i. If we set

$$\alpha_0 = 1,\ \beta_{-1} = 0,\ p_0 = 0, \tag{4}$$

then (3) will also be valid for $i = 0$. Thus, the solution of (1) will be found in the form (3) for any $i \geq 0$.

Writing (1) in the form of a recurrence relation

$$y_{i-1} = Cy_i - y_{i+1} - F_i,\ i \leq N-1,\ y_N = 0 \tag{5}$$

and proceeding analogously, we obtain the solution of (1) in the form

$$y_{i-1} = \xi_{N-i}y_{N-1} - \eta_{N-i-1}y_N - q_{N-i}, \tag{6}$$

for any $i \leq N$, if we set

$$\xi_0 = 1,\ \eta_{-1} = 0,\ q_0 = 0. \tag{7}$$

We note that if y_{N-1} is found, then all the y_i's can be computed recursively using (5).

We now find y_1 and y_{N-1}. To do this we determine the coefficients α_i, β_i, ξ_i, η_i, p_i, q_i. Comparing (2) and (3) for $i = 1$, and (5) and (6) for $i = N-1$, we obtain

$$\alpha_1 = \xi_1 = C,\ \beta_0 = \eta_0 = 1,\ p_1 = F_1,\ q_1 = F_{N-1}. \tag{8}$$

We find now the recurrence formulas for determining the desired coefficients. We extract from (3) the expressions for y_i and y_{i-1}:

$$y_i = \alpha_{i-1}y_1 - \beta_{i-2}y_0 - p_{i-1},\ y_{i-1} = \alpha_{i-2}y_1 - \beta_{i-3}y_0 - p_{i-2}$$

and substitute in (1). We obtain

$$-(\alpha_{i-2} - C\alpha_{i-1} + \alpha_i)y_1 + (\beta_{i-3} - C\beta_{i-2} + \beta_{i-1})y_0 + p_{i-2} - Cp_{i-1} + p_i = F_i, i \geq 2.$$

For these equations to be identities for all i, it is sufficient to set

$$p_i = Cp_{i-1} - p_{i-2} + F_i, \tag{9}$$

$$\alpha_i = C\alpha_{i-1} - \alpha_{i-2},\ \beta_{i-1} = C\beta_{i-2} - \beta_{i-3}, \tag{10}$$

for $i \geq 2$.

Analogously, using (6) and (1), we obtain for $i \leq N-2$ the recurrence relations

$$q_{N-i} = Cq_{N-i-1} - q_{N-i-2} + F_i,$$

$$\xi_{N-i} = C\xi_{N-i-1} - \xi_{N-i-2},\ \eta_{N-i-1} = C\eta_{N-i-2} - \eta_{N-i-3}.$$

Changing $N-i$ to i, we obtain the formulas

$$q_i = Cq_{i-1} - q_{i-2} + F_{N-i}, \tag{11}$$

$$\xi_i = C\xi_{i-1} - \xi_{i-2},\ \eta_{i-1} = C\eta_{i-2} - \eta_{i-3}, \tag{12}$$

for $i \geq 2$.

Thus (4), (7)–(12) define the desired coefficients. Comparing (10) and (12) with (4), (7), (8), we obtain that $\beta_i = \eta_i = \xi_i = \alpha_i$ for $i \geq 0$. Thus, (3) and (6) take the form

$$y_{i+1} = \alpha_i y_1 - \alpha_{i-1} y_0 - p_i, \qquad i \geq 0, \tag{13}$$
$$y_{i-1} = \alpha_{N-i} y_{N-1} - \alpha_{N-i-1} y_N - q_{N-i}, \qquad i \leq N \tag{14}$$

where

$$p_i = Cp_{i-1} - p_{i-2} + F_i, \quad i \geq 2, \quad p_0 = 0, \quad p_1 = F_1, \tag{15}$$
$$q_i = Cq_{i-1} - q_{i-2} + F_{N-i}, \quad i \geq 2, \quad q_0 = 0, \quad q_1 = F_{N-1}, \tag{16}$$
$$\alpha_i = C\alpha_{i-1} - \alpha_{i-2}, \quad i \geq 2, \quad \alpha_0 = 1, \quad \alpha_1 = C. \tag{17}$$

We now find y_1 and y_{N-1}. For this we set $i = k$ in (13), and $i = k+2$ in (14). Since $N = 2k+1$ we obtain

$$y_{k+1} = \alpha_k y_1 - \alpha_{k-1} y_0 - p_k, \; y_{k+1} = \alpha_{k-1} y_{N-1} - \alpha_{k-2} y_N - q_{k-1}.$$

Subtracting the first equation from the second, we obtain an equation relating y_1 and y_{N-1}:

$$\alpha_{k-1} y_{N-1} - \alpha_k y_1 + \alpha_{k-1} y_0 - \alpha_{k-2} y_N = q_{k-1} - p_k. \tag{18}$$

We obtain another equation for y_1 and y_{N-1} by setting $i = k-1$ in (13) and $i = k+1$ in (14) and subtracting the second equation from the first,

$$-\alpha_k y_{N-1} + \alpha_{k-1} y_1 - \alpha_{k-2} y_0 + \alpha_{k-1} y_N = p_{k-1} - q_k. \tag{19}$$

Taking into account that $y_0 = y_N = 0$, we add and subtract (18) and (19). We obtain the equivalent system

$$(\alpha_k - \alpha_{k-1})(y_{N-1} + y_1) = v_0 = p_k - q_{k-1} - p_{k-1} + q_k, \tag{20}$$
$$(\alpha_k + \alpha_{k-1})(y_{N-1} - y_1) = w_0 = q_{k-1} - p_k - p_{k-1} + q_k,$$

which we solve to find y_1 and y_{N-1}:

$$y_1 = 0.5(v_k - w_k), \; y_{N-1} = 0.5(v_k + w_k), \tag{21}$$

where

$$v_k = (\alpha_k - \alpha_{k-1})^{-1} v_0, \; w_k = (\alpha_k + \alpha_{k-1})^{-1} w_0. \tag{22}$$

Thus, this algorithm for solving (1) consists of using (15)–(17) to compute the coefficients p_{k-1}, p_k, q_k, q_{k-1}, α_{k-1}, and α_k; using (22), (21) for the values v_k, w_k and y_1, y_{N-1}; (2) for the unknowns y_j for $j = 2, 3, \ldots, k$; and (5) for $j = N-2, N-3, \ldots, k+1$ with data y_0, y_N and computed y_1, y_{N-1}. It is easy to calculate that it requires $7N - 9$ arithmetic operations. The resulting method for solving (1) is called the staircase algorithm.

We now determine when this is a valid algorithm. If $\alpha_k^2 \neq \alpha_{k-1}^2$, then from (22) it follows that (1) is solvable for any right-hand side, and in this case the formulas (22) do not involve division by zero. We note that in view of the definition in (17), α_k is an algebraic polynomial of degree k in C, $\alpha_k = U_k(\frac{C}{2})$, where $U_k(x)$ is the Chebyshev polynomial of the second kind of degree k:

$$U_k(x) = \begin{cases} \dfrac{\sin(k+1)\arccos x}{\sin \arccos x}, & |x| \leq 1, \\ \dfrac{(x+\sqrt{x^2-1})^{k+1} - (x-\sqrt{x^2-1})^{k+1}}{2\sqrt{x^2-1}}, & |x| \geq 1. \end{cases}$$

From this we easily find that $\alpha_k^2 - \alpha_{k-1}^2 = \alpha_{2k}$. Therefore if $\frac{C}{2}$ is not a root of the polynomial $U_{2k}(x)$, i.e. $C \neq 2\cos\frac{m\pi}{N}$, m an integer, then the staircase algorithm is correct.

We turn our attention to the fact that the staircase algorithm can be numerically unstable. In fact, if $|C| > 2$, then the algorithm is characterized by error growth that is exponential in N, since among the roots of the characteristic equation $q^2 - Cq + 1 = 0$ corresponding to the difference equations (2), (5), (15)–(17) is one whose modulus is greater than one.

4.4.2 The staircase algorithm for the case of a block-tridiagonal matrix. We consider the problem (1) assuming that y_i and F_i are vectors of dimension M, and C is a square matrix of size $M \times M$. We limit ourselves to the case when C is a tridiagonal matrix. In subsection 2, §1 of Chapter III it was shown that the Dirichlet difference problem for Poisson's equation on a rectangle with a uniform grid in each direction, introduced into a rectangle, can be written as a system of three-point vector equations (1). In this case the components of the vector of unknowns are the values of the desired grid function corresponding to the i-th row of the grid, the matrix C is tridiagonal, and its order M is equal to the number of inner nodes of a row of the grid.

The staircase algorithm described above can also be used in this case. The only difficulty that arises when solving three-point vector equations with this algorithm is the finding of v_k and w_k in (22). In this case α_k is a matrix polynomial in the matrix C. Using the explicit expression for α_k and taking

into account that α_k is a monic polynomial, it is possible to use the following expressions

$$\begin{aligned}\alpha_k - \alpha_{k-1} &= \prod_{l=1}^{k}\left(C - 2\cos\frac{(2l-1)\pi}{2k+1}E\right),\\ \alpha_k + \alpha_{k-1} &= \prod_{l=1}^{k}\left(C - 2\cos\frac{2l\pi}{2k+1}E\right).\end{aligned} \tag{23}$$

Using (22), (23) it is possible to construct the following algorithm to find v_k and w_k, with v_0 and w_0 given by (20):

$$\begin{gathered}\left[C - 2\cos\frac{(2l-1)\pi}{2k+1}E\right]v_l = v_{l-1},\\ \left[C - 2\cos\frac{2l\pi}{2k+1}E\right]w_l = w_{l-1},\\ l = 0,1,\ldots,k.\end{gathered} \tag{24}$$

Since each of the systems (24) has a tridiagonal matrix (there are $2k$ such systems), and can be solved, for example, by elimination in $O(M)$ arithmetic operations, finding v_k and w_k requires $O(MN)$ operations. To compute the vectors p_{k-1}, q_{k-1}, p_k and q_k using (15), (16), also requires $O(MN)$ operations. Obviously, the number of operations required to find the vectors y_i, $2 \le i \le N-2$ using (2), (5) is the same. Thus, to solve (1) with the special block-tridiagonal matrices, the staircase algorithm requires the same number of arithmetic operations as unknowns.

From (24) it follows that if the numbers $2\cos\frac{l\pi}{N}$, $1 \le l \le N-1$, are not equal to the eigenvalues of the matrix C, then (1) is soluble for any right-hand side and the staircase algorithm is correct. If among the eigenvalues of the matrix is a value greater than 2 in modulus, then as in the scalar case the algorithm displays error growth exponential in N. Such error growth is connected with the fact that the Cauchy problem (15)–(17), (2), (5) for difference equations of second order is solved on an interval whose length $k = 0.5(N-1)$ grows linearly with N.

4.4.3 Stability of the staircase algorithm. We now construct a variant of the staircase algorithm, numerically stable in the sense that error growth is like a power in N. We will consider problem (1), assuming initially that y_i, F_i, and C are scalars, $|C| > 2$.

We write N in the form $N = 2kL + 1$ where k and L are integers and decompose (1) into L subsystems containing $2k$ equations

$$-y_{2lk+j-1} + Cy_{2lk+j} - y_{2lk+j+1} = F_{2lk+j},\ 1 \le j \le 2k, \tag{25}$$

for $l = 0, 1, \ldots, L-1$. We denote

$$w_{2l} = y_{2lk},\ w_{2l+1} = y_{2lk+1},\ l = 0, 1, \ldots, L. \tag{26}$$

For fixed l, (25) can be written in the form of a first-order boundary-value problem for the three-point equation

$$-u_{j-1}^{(l)} + Cu_j^{(l)} - u_{j+1}^{(l)} = \varphi_j^{(l)},\ 1 \le j \le 2k,\ u_0^{(l)} = w_{2l},\ u_{2k+1}^{(l)} = w_{2l+3}, \tag{27}$$

where we denote

$$u_j^{(l)} = y_{2lk+j},\ \varphi_j^{(l)} = F_{2lk+j},\ 0 \le j \le 2k+1. \tag{28}$$

We note that by (26), (28)

$$u_1^{(l)} = w_{2l+1},\ u_{2k}^{(l)} = w_{2l+2},\ 0 \le l \le L-1. \tag{29}$$

Consequently, if the values w_m, $0 \le m \le 2L+1$ are known, then $u_j^{(l)}$, $2 \le j \le 2k-1$, can be computed using the recurrence formulas

$$\begin{aligned} &u_{j+1}^{(l)} = Cu_j^{(l)} - u_{j-1}^{(l)} - \varphi_j^{(l)}, \quad 1 \le j \le k-1, \quad u_0^{(l)} = w_{2l}, \quad u_1^{(l)} = w_{2l+1}, \\ &u_{j-1}^{(l)} = Cu_j^{(l)} - u_{j+1}^{(l)} - \varphi_j^{(l)}, \quad 2k \ge j \ge k+2, \quad u_{2k+1}^{(l)} = w_{2l+3}, \quad u_{2k}^{(l)} = w_{2l+2}. \end{aligned}$$

Substituting here the notation (28), we obtain recurrence formulas for computing the desired unknowns ($l = 0, 1, \ldots, L-1$)

$$\begin{aligned} y_{j+1} &= Cy_j - y_{j-1} - F_j, \quad && 2lk+1 \le j \le (2l+1)k-1, \\ y_{2lk} &= w_{2l}, && y_{2lk+1} = w_{2l+1}; \\ y_{j-1} &= Cy_j - y_{j+1} - F_j, && 2(l+1)k \ge j \ge (2l+1)k+2, \\ y_{2(l+1)k+1} &= w_{2l+3}, && y_{2(l+1)k} = w_{2l+2}. \end{aligned} \tag{30}$$

Thus, we must find w_m, $1 \le m \le 2L$. Analogs of (18), (19) are valid for the system (27) for fixed l, which due to (29) have the following form ($l = 0, 1, \ldots, L-1$):

$$\begin{aligned} \alpha_{k-1}w_{2l} - \alpha_k w_{2l+1} + \alpha_{k-1}w_{2l+2} - \alpha_{k-2}w_{2l+3} &= q_{k-1}^{(l)} - p_k^{(l)} = g_{2l+1}, \\ -\alpha_{k-2}w_{2l} + \alpha_{k-1}w_{2l+1} - \alpha_k w_{2l+2} + \alpha_{k-1}w_{2l+3} &= p_{k-1}^{(l)} - q_k^{(l)} = g_{2l+2}, \end{aligned} \tag{31}$$

where α_k, α_{k-1} are determined from (17), and $p_k^{(l)}$, $p_{k-1}^{(l)}$, $q_k^{(l)}$, $q_{k-1}^{(l)}$ are computed, taking into account (28), according to the following formulas, analogous to (15), (16) ($l = 0, 1, \ldots, L-1$):

$$\begin{aligned} p_j^{(l)} &= Cp_{j-1}^{(l)} - p_{j-2}^{(l)} + F_{2lk+j}, \qquad 2 \le j \le k, \quad p_0^{(l)} = 0, \quad p_1^{(l)} = F_{2lk+1}, \\ q_j^{(l)} &= Cq_{j-1}^{(l)} - q_{j-2}^{(l)} + F_{2(l+1)k+1-j}, \quad 2 \le j \le k, \quad q_0^{(l)} = 0, \quad q_1^{(l)} = F_{2(l+1)k}. \end{aligned} \tag{32}$$

Since $w_0 = w_{2L+1} = 0$, (31) is a system of $2L$ equations in $2L$ unknowns. To solve this system we multiply each pair of equations (31) on the left by the matrix

$$(\alpha_{k-2}^2 - \alpha_{k-1}^2)^{-1} \begin{pmatrix} \alpha_{k-1} & \alpha_{k-2} \\ \alpha_{k-2} & \alpha_{k-1} \end{pmatrix},$$

and take into account that $\alpha_{k-1}^2 - \alpha_k \alpha_{k-2} = 1$, $k \ge 2$. As a result we obtain a system of $2L$ equations with a tridiagonal matrix

$$\begin{aligned} -w_{2l} + aw_{2l+1} - bw_{2l+2} &= \varphi_{2l+1}, \\ -bw_{2l+1} + aw_{2l+2} - w_{2l+3} &= \varphi_{2l+2}, \\ l = 0, 1, \ldots, L-1, \ w_0 = w_{2L+1} &= 0, \end{aligned} \tag{33}$$

where

$$\begin{aligned} a &= \alpha_{k-1}(\alpha_k - \alpha_{k-2})b, \qquad b = (\alpha_{k-1}^2 - \alpha_{k-2}^2)^{-1}, \\ \varphi_{2l+1} &= \varphi_{2l+1}^+ + \varphi_{2l+2}^-, \qquad \varphi_{2l+2} = \varphi_{2l+1}^+ - \varphi_{2l+2}^-, \\ \varphi_{2l+1}^+ &= \frac{g_{2l+2} + g_{2l+1}}{2(\alpha_{k-2} - \alpha_{k-1})}, \qquad \varphi_{2l+1}^- = \frac{g_{2l+2} - g_{2l+1}}{2(\alpha_{k-2} + \alpha_{k-1})}. \end{aligned} \tag{34}$$

We show that if $|C| > 2$, the matrix of the system (33) is diagonally dominant, i.e., it satisfies the inequality $|a| > 1 + |b|$. In fact, from the properties of the Chebyshev polynomials $T_m(x)$ and $U_m(x)$ of the first and second kinds

$$\begin{gathered} U_m^2(x) - U_{m-1}^2(x) = U_{2m}(x), \ U_m(x) > 0, \ T_m(x) \ge x, \ x > 1, \\ U_{m-1}(x)[U_m(x) - U_{m-2}(x)] = U_{2m-1}(x), \\ U_m(-x) = (-1)^m U_m(x), \ U_m(x) = T_m(x) + xU_{m-1}(x), \end{gathered}$$

it follows that

$$\begin{aligned} a &= \alpha_{2k-1}/\alpha_{2k-2}, \\ b &= 1/\alpha_{2k-2} > 0, \\ \frac{|a|}{1+b} &= \frac{|\alpha_{2k-1}|}{1+\alpha_{2k-2}} \\ &= \frac{U_{2k-1}(x)}{1+U_{2k-2}(x)} \\ &= \frac{T_{2k-1}(x)+xU_{2k-2}(x)}{1+U_{2k-2}(x)} \\ &\geq x > 1, \ x = |C|/2. \end{aligned}$$

The required inequality is proved. Therefore to solve (33) in this case it is possible to use the monotonic elimination method described in Section 1 of Chapter II.

Thus, this variant of the staircase algorithm consists of the computation for each l, $0 \leq l \leq L-1$, of the quantities $p_j^{(l)}$ and $q_j^{(l)}$ according to the recurrence formulas (32), solving the system (33), (34) with a tridiagonal matrix, and finding of the desired unknowns y_j for each l, $0 \leq l \leq L-1$ using the recurrence formulas (30). Since in the case $|C| > 2$ the growth of the error depends exponentially on the length of the interval on which are solved the Cauchy problems (30), (32) for the second-order difference equations, then choosing $k = O(\ln N)$ it is possible to guarantee the power character of the error growth as a function of the number of unknowns N.

We consider now the case when y_i and F_i are vectors of dimension M, and C is a symmetric matrix whose eigenvalues λ_m satisfy the condition $|\lambda_m| > 2$, $1 \leq m \leq M$. We denote by $v_m = (v_m^{(1)}, v_m^{(2)}, \ldots, v_m^{(M)})^T$ the eigenvector of the matrix C corresponding to the eigenvalue λ_m, and by $V = [v_1, v_2, \ldots, v_M]$ the orthogonal matrix that reduces C to diagonal form: $V^T C V = \Lambda = \operatorname{diag}\{\lambda_m\}_{m=1}^M$, $V^T V = E$. For this vector case the basic problem is the solution of the system of four-point vector equations (31).

To solve this sytem, we multiply each equation in (31) on the left by the matrix V^T and take into account that

$$V^T \alpha_k V = V^T U_k \left(\frac{C}{2}\right) V = U_k \left(\frac{\Lambda}{2}\right) = \hat{\alpha}_k = \operatorname{diag}\left\{\mu_k^{(-)}\right\}_{m=1}^M,$$

and use the following notation

$$w_n = V\hat{w}_n, \ \hat{g}_n = V^T g_n, \ \mu_k^{(-)} = U_k\left(\frac{\lambda_m}{2}\right).$$

As a result we obtain the system ($l = 0, 1, \ldots, L-1$)

$$\begin{aligned} \hat{\alpha}_{k-1}\hat{w}_{2l} - \hat{\alpha}_k\hat{w}_{2l+1} + \hat{\alpha}_{k-1}\hat{w}_{2l+2} - \hat{\alpha}_{k-2}\hat{w}_{2l+3} &= \hat{g}_{2l+1}, \\ -\hat{\alpha}_{k-2}\hat{w}_{2l} + \hat{\alpha}_{k-1}\hat{w}_{2l+1} - \hat{\alpha}_k\hat{w}_{2l+2} + \hat{\alpha}_{k-1}\hat{w}_{2l+3} &= \hat{g}_{2l+2}, \end{aligned} \tag{35}$$

linking the vectors $\hat{w}_n = (\hat{w}_n^{(1)}, \hat{w}_n^{(2)}, \ldots, \hat{w}_n^{(M)})^T$, $0 \le n \le 2L+1$. Since $\hat{\alpha}_k$ are diagonal matrices, (35) decouples into M subsystems of $2L$ equations each for the components of the vectors $\hat{w}_n$ ($m = 1, 2, \ldots, M$):

$$\begin{aligned} \mu_{k-1}^{(-)}\hat{w}_{2l}^{(-)} - \mu_k^{(-)}\hat{w}_{2l+1}^{(-)} + \mu_{k-1}^{(-)}\hat{w}_{2l+2}^{(-)} - \mu_{k-2}^{(-)}\hat{w}_{2l+3}^{(-)} &= \hat{g}_{2l+1}^{(-)}, \\ -\mu_{k-2}^{(-)}\hat{w}_{2l}^{(-)} + \mu_{k-1}^{(-)}\hat{w}_{2l+1}^{(-)} - \mu_k^{(-)}\hat{w}_{2l+2}^{(-)} + \mu_{k-2}^{(-)}\hat{w}_{2l+3}^{(-)} &= \hat{g}_{2l+2}^{(-)}, \\ l = 0, 1, \ldots, L-1. \end{aligned} \tag{36}$$

Applying the transformation described for the scalar case to (36), we obtain M systems with tridiagonal matrices ($m = 1, 2, \ldots, M$)

$$\begin{aligned} -\hat{w}_{2l}^{(-)} + a^{(-)}\hat{w}_{2l+1}^{(-)} - b^{(-)}\hat{w}_{2l+2}^{(-)} &= \varphi_{2l+1}^{(-)}, \\ -b^{(-)}\hat{w}_{2l+1}^{(-)} + a^{(-)}\hat{w}_{2l+2}^{(-)} - \hat{w}_{2l+3}^{(-)} &= \varphi_{2l+2}^{(-)}, \\ l = 0, 1, \ldots, L-1,\ \hat{w}_0^{(-)} &= \hat{w}_{2L+1}^{(-)} = 0, \end{aligned} \tag{37}$$

where

$$\begin{aligned} a^{(-)} &= \mu_{k-1}^{(-)}[\mu_k^{(-)} - \mu_{k-2}^{(-)}]b^{(-)}, & b^{(-)} &= [(\mu_k^{(-)})^2 - (\mu_{k-2}^{(-)})^2]^{-1}, \\ \varphi_{2l+1}^{(-)} &= \varphi_{2l+1}^{+} + \varphi_{2l+2}^{-}, & \varphi_{2l+2} &= \varphi_{2l+1}^{+} - \varphi_{2l+2}^{-}, \\ \varphi_{2l+1}^{+} &= \frac{\hat{g}_{2l+2}^{(-)} + \hat{g}_{2l+1}^{(-)}}{2[\mu_{k-2}^{(-)} - \mu_{k-1}^{(-)}]}, & \varphi_{2l+2}^{-} &= \frac{\hat{g}_{2l+2}^{(-)} - \hat{g}_{2l+1}^{(-)}}{2[\mu_{k-2}^{(-)} + \mu_{k-1}^{(-)}]}. \end{aligned}$$

The system (37) is diagonally dominant if $|\lambda_m| > 2$, $1 \le m \le M$, therefore it is possible to use the monotonic elimination method to solve it.

Thus, for the system of three-point vector equations (1) with the aforementioned assumptions on the matrix C, this variant of the staircase algorithm consists of the computation for each l, $0 \le l \le L-1$, of the vectors $p_j^{(l)}$, $q_j^{(l)}$, $j = k, k-1$, via (32), computation of the vectors $\hat{g}_{2l+1}$ and $\hat{g}_{2l+2}$ using the formulas ($l = 0, 1, \ldots, L-1$)

$$\hat{g}_{2l+1}^{(-)} = \sum_{j=1}^{M} v_m^{(j)} g_{2l+1}^{(j)},\ \hat{g}_{2l+2}^{(-)} = \sum_{j=1}^{M} v_m^{(j)} g_{2l+2}^{(j)},\ 1 \le m \le M, \tag{38}$$

the solution of the systems (37) for each fixed m, $1 \le m \le M$, computation of the vectors w_n using the formulas ($n = 1, 2, \dots, 2L$)

$$w_n^{(j)} = \sum_{m=1}^{M} \hat{w}_n^{(-)} v_m^{(j)}, \quad q \le j \le M, \tag{39}$$

and finding the desired vectors y_j using (30) for each l, $0 \le l \le L-1$.

In the special case when (1) corresponds to a Dirichlet difference problem for Poisson's equation on a grid with MN internal nodes, we have

$$v_m^{(j)} = \alpha \sin \frac{m\pi j}{M+1}, \ 1 \le j \le M, \ \lambda_m = 2 + \frac{4h_2^2}{h_1^2} \sin^2 \frac{m\pi}{2(M+1)}, \ 1 \le m \le M,$$

where α is the normalized multiplier, and h_1 and h_2 are the steps of the grid. Therefore the sums in (38), (39) can be computed using the fast Fourier transform, which is described in Section 1 of Chapter IV for the case $M = 2^n - 1$. Then the computation of all the necessary sums requires $O(LM \log M)$ operations, where $L = \frac{N-1}{2k}$. Since the computations in (30), (32) and the solution of all the systems (37) requires $O(MN)$ operations, the overall number of operations is $O(MN + \frac{MN}{k} \log M)$. If M and N are of the same order, then choosing $k = O(\log M)$ we obtain that the number of operations for the staircase algorithm for this example is proportional to the number of unknowns, and moreover the rate of error growth is guaranteed.

4.4.4 The reduction method for three-point scalar equations. In a series of cases of solving systems of linear algebraic equations with tridiagonal matrices, the accuracy of the computed solution is of great significance. Analysis of the formulas for the elimination method applied to such systems shows that the formulas for computing the elimination coefficients can be a source of errors. Below we will consider the reduction method for solving such systems, which is free from such a deficiency.

Thus, suppose we need to solve a three-point difference problem

$$-a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \ 1 \le i \le N-1, \ y_0 = 0, \ y_N = 0, \tag{40}$$

where $c_i = a_i + b_i + d_i$, $a_i > 0$, $b_i > 0$, $d_i \ge 0$. We assume that $N = 2^n$. The idea of the reduction method consists in the sequential elimination from (40) of unknowns with odd numbers, then with numbers divisible by 2, and so forth.

We write three successive equations of the system (40) with numbers $i-1$, i, $i+1$, where i is an even number

$$-a_{i-1}y_{i-2} + (a_{i-1} + b_{i-1} + d_{i-1})y_{i-1} - b_{i-1}y_i = f_{i-1}, \tag{41}$$

$$-a_i y_{i-1} + (a_i + b_i + d_i)y_i - b_i y_{i+1} = f_i, \tag{42}$$

$$-a_{i+1}y_i + (a_{i+1} + b_{i+1} + d_{i+1})y_{i+1} - b_{i+1}y_{i+2} = f_{i+1}. \tag{43}$$

Multiplying (41) by $\alpha_i^{(1)} = a_i(a_{i-1} + b_{i-1} + d_{i-1})^{-1}$, (43) by $\beta_i^{(1)} = b_i(a_{i+1} + b_{i+1} + d_{i+1})^{-1}$ and adding the resulting equations to (42), we find

$$-a_i^{(1)}y_{i-2} + (a_i^{(1)} + b_i^{(1)} + d_i^{(1)})y_i - b_i^{(1)}y_{i+2} = f_i^{(1)},$$

$$i = 2, 4, 6, \ldots, N-2, \tag{44}$$

$$y_0 = 0,\ y_N = 0,$$

where

$$a_i^{(1)} = \alpha_i^{(1)}a_{i-1},\ b_i^{(1)} = \beta_i^{(1)}b_{i+1}$$

$$d_i^{(1)} = \alpha_i^{(1)}d_{i-1} + d_i + \beta_i^{(1)}d_{i+1},\ f_i^{(1)} = \alpha_i^{(1)}f_{i-1} + f_i + \beta_i^{(1)}f_{i+1}.$$

If the unknowns with even numbers are found (they satisfy the system (44)), then the remaining unknowns are determined using

$$y_i = (f_i + a_i y_{i-1} + b_i y_{i+1})/(a_i + b_i + d_i),\ i = 1, 3, 5, \ldots, N-1.$$

This elimination process can obviously be applied to (44), from which at the second step will be eliminated the unknowns with numbers divisible by 2, but not by 4. After the l-th step of the elimination process we obtain the system

$$-a_i^{(l)}y_{i-2^l} + (a_i^{(l)} + b_i^{(l)} + d_i^{(l)})y_i - b_i^{(l)}y_{i+2^l} = f_i^{(l)},$$

$$i = 2^l, 2\cdot 2^l, 3\cdot 2^l, \ldots, N-2^l, \tag{45}$$

$$y_0 = 0,\ y_N = 0,$$

where

$$
\begin{aligned}
a_i^{(l)} &= \alpha_i^{(l)} a_{i-2^{l-1}}^{(l-1)}, \\
b_i^{(l)} &= \beta_i^{(l)} b_{i+2^{l-1}}^{(l-1)}, \\
d_i^{(l)} &= \alpha_i^{(l)} d_{i-2^{l-1}}^{(l-1)} + d_i^{(l-1)} + \beta_i^{(l)} d_{i+2^{l-1}}^{(l-1)}, \\
f_i^{(l)} &= \alpha_i^{(l)} f_{i-2^{l-1}}^{(l-1)} + f_i^{(l-1)} + \beta_i^{(l)} f_{i+2^{l-1}}^{(l-1)}, \\
\alpha_i^{(l)} &= a_i^{(l-1)} \left[a_{i-2^{l-1}}^{(l-1)} + b_{i-2^{l-1}}^{(l-1)} + d_{i-2^{l-1}}^{(l-1)} \right]^{-1}, \\
\beta_i^{(l)} &= b_i^{(l-1)} \left[a_{i+2^{l-1}}^{(l-1)} + b_{i+2^{l-1}}^{(l-1)} + d_{i+2^{l-1}}^{(l-1)} \right]^{-1}, \\
& i = 2^l, 2 \cdot 2^l, 3 \cdot 2^l, \ldots, N - 2^l, \; l \geq 1.
\end{aligned}
\tag{46}
$$

Here we used the notation $a_i^{(0)} \equiv a_i$, $b_i^{(0)} \equiv b_i$, $d_i^{(0)} \equiv d_i$, $f_i^{(0)} \equiv f_i$.

The elimination process is completed at the $(n-1)$-st step, when (45) will consist of one equation involving the unknown $y_{N/2} = y_{2^{n-1}}$. From this equation we find

$$
y_{2^{n-1}} = \frac{f_{2^{n-1}}^{(n-1)} + a_{2^{n-1}}^{(n-1)} y_0 - b_{2^{n-1}}^{(n-1)} y_N}{a_{2^{n-1}}^{(n-1)} + b_{2^{n-1}}^{(n-1)} + d_{2^{n-1}}^{(n-1)}}, \; y_0 = y_N = 0. \tag{47}
$$

The remaining unknowns are determined using

$$
\begin{aligned}
y_i &= (f_i^{(l)} + a_i^{(l)} y_{i-2^l} + b_i^{(l)} y_{i+2^l}) \Big/ (a_i^{(l)} + b_i^{(l)} + d_i^{(l)}), \\
& i = 2^l, 3 \cdot 2^l, 5 \cdot 2^l, \ldots, N - 2^l,
\end{aligned}
\tag{48}
$$

where $l = n-2, n-3, \ldots, 0$, $y_0 = y_N = 0$. We note that (48) incorporates (47) for $l = n-1$.

Thus, in the forward path of the reduction method we compute $a_i^{(l)}$, $b_i^{(l)}$, $d_i^{(l)}$, $f_i^{(l)}$ for $l = 1, 2, \ldots, n-1$ using (46), and on the reverse path we find the desired solution using (48) for $l = n-1, n-2, \ldots, 0$. Note that the method does not require auxiliary memory, since the quantities $a_i^{(l)}$, $b_i^{(l)}$, $d_i^{(l)}$, $f_i^{(l)}$ can be overwritten on $a_{i-2^{l-1}}^{(l-1)}$, $b_{i+2^{l-1}}^{(l-1)}$, $d_i^{(l-1)}$, $f_i^{(l-1)}$. The method requires $12N$ additions, $8N$ multiplications and $3N$ divisions.

The method can obviously be generalized to the case of arbitrary N, and other types of boundary conditions.

Index

alternate triangle method (ATM) I-xxix–xxxv; II-190, 225–267, 284, 301, 427, 431
  block form II-241
  general theory II-225–241
  modified ATM I-xxx; II-243–250, 253
  parameters II-228–233
alternating directions iterative method (ADI) I-xxxi; II-269–301, 303, 321–325, 341–343, 461–465, 479–482
  commutative case II-272
  iterative scheme II-269–271
  non-commutative case II-269–301
  parameters II-271–273, 276–280
Andreev, A.B. II-417, 496
axial symmetry II-449

Buzbee, B.L. II-497–498

Cauchy problem I-13–16
characteristic equation I-30–34
Chebyshev method I-xxix–xxxiv; II-65, 69–86, 107–108, 112–115, 117–124, 131, 136–143, 145, 214, 284, 301, 303, 309–314, 333–339, 405, 421, 428, 431–432, 487
  choice of operator II-72–75
  iteration count II-70
  optimality II-71–72
  parameters II-69–71, 82–86
  stability II-75–82
Chebyshev polynomial I-xxvii–xxviii, 41–43, 48, 133–134, 167–169, 230, 233; II-36, 130, 439, 489–494
Chebyshev semi-iterative method II-125, 129–133, 136–143
  algorithm II-132–133
  choice of operator II-132
  parameters II-129–131
conjugate correction method II-175
conjugate direction methods II-164–180
conjugate error method II-176, 313, 315
conjugate gradient method II-175, 368, 497
conjugate residual method II-175
Concus, P. II-497–498
cyclic reduction I-xxv, 117–170, 236–238; II-343–346, 459–461, 478–479, 497

difference derivatives I-4–8
  central I-4
  general order I-5
  left I-4
  product I-6

right I-4
summation by parts I-6
difference identities II-25–27
difference scheme I-2
operator form II-21–54
domain augmentation II-433, 442–445
domain decomposition II-433–442
algorithm II-438–442

elimination method I-xxv, 61–116, 146–147, 207–208, 211, 226–227; II-281–283, 441, 477, 497
block I-61, 97–116
column pivoting I-95
cyclic I-xxv, 61, 77–80
flow I-xxv, 61, 73–76
incomplete See incomplete reduction
monotone I-xxv, 86, 92, 234–235
non-monotone I-xxv, 61, 86–89, 95–97, 226
odd-even See cyclic reduction
orthogonal I-61, 107–111
two-sided I-65–66
elliptic integral II-276–277
error problem II-66–67

Faddeev, D.K. II-496
Faddeeva, V.N. II-496
FFT See Fourier transform (fast)
fixed point theorems II-19–21
Forsythe, G. II-495
Fourier transform (fast) I-xxv, 50, 171–211, 213–218, 224; II-431, 497
algorithm I-171–196
complex grid function I-175–176, 193–196
cosine expansion I-174, 185–187
real grid function I-174–175, 188–193
shifted sine expansion I-173–174, 176–185
sine expansion I-172–173, 176–185
full relaxation II-200

Gateaux derivative II-6, 354–355
Gauss-Seidel method I-xxviii–xxix; II-189–199, 209, 220–223
block version II-194–196
convergence II-196–199, 220–223
Gaussian elimination
See elimination method
Gelfond, A.O. II-495
generalized solution II-19
Golub, G.H. II-497–498
gradient descent methods II-367
Green's formulas II-26–27, 31–35, 39, 243, 256, 286, 349, 405, 407, 426, 474, 477
grid I-1–4
non-uniform I-2
uniform I-2
grid functions I-1–4
space of II-21–24
Gulin, A.V. I-xxviii; II-496

Hamming, R.W. II-497–498

incomplete reduction I-xxv, 131, 171, 211–227; II-459–461, 478–479
indefinite equations II-303–325
invariant subspace II-7
iterative methods II-1–63
canonical form II-1
classification II-60–63
convergence II-1, 58–60
explicit II-57–58
general II-56–58
implicit II-57–58
nonlinear II-57
theory II-1–63
three-level II-125–143
convergence II-125–129, 138–143
stability II-136–143
stationary II-133–136
two-level II-65–124

Jacobi elliptic function II-277

Karamzin, Y.N. II-496
Karchevskij, M.M. II-495
Kucherov, A.B. II-243

Lamé equations II-428
Laplace difference operator I-xxi, xxiii, xxv, xxviii, xxix, xxxiii, 196–203; II-164, 209, 406, 425
  eigenvalue problem I-196–201
lattice See grid
linear independence I-18–20; II-2
linear spaces II-1–4
  Banach space II-3
  complex II-2
  Hilbert space II-4
  normed II-3, 6
  real II-2
Lipschitz condition II-5

Marchuk, G.I. II-495
minimal correction method II-160–161
minimal error method II-161, 313
minimal residual method II-158–159, 330–333

Newton-Kantorovich method II-358–362
Nielsen, C.W. II-497–498
Nikolaev, E.S. II-82, 243
nonlinear equations II-351–387
norm II-3
normal solution II-18

O'Leary, D.P. II-497–498
Oganesyan, L.A. II-495
operators II-1, 5–54
  abelian See operators, commutative
  adjoint II-8
  bounded II-5
  bounds II-11, 28–42
  commutative II-6
  continuous II-5
  domain II-5
  eigenelement II-16–17
  eigenvalue II-16–17
  energy II-11
  energy equivalence II-11
  energy space II-12
  finite-dimensional II-15–17
  function of II-14–15
  inverse II-6
  linear II-1, 5
  monotonic II-11, 352
  non-linear II-1
  non-negative II-10
  normal II-9
  numerical radius II-10
  positive definite II-10
  potential II-365
  range II-5
  self-adjoint II-9
  skew-symmetric II-9
  solubility II-18–21
  spectral radius II-7, 16
  spectrum II-16
  strictly monotonic II-11
  strongly monotonic II-11, 352
Ortega, J.M. II-495

Paige, C.C. II-497–498
Poisson's equation I-xxv, 47, 117, 119–120, 123, 125–126, 145–148, 201–205, 208–211, 219–227, 230, 236; II-105–115, 162–164, 207–212, 225, 233–241, 264–267, 280–284, 290, 339–346, 401–409, 427, 431, 433–438
  in a ring II-453–454, 470–473
  in a ring sector II-454, 483–485
  on a cylinder I-125, 211; II-447–469, 487
  polar coordinates I-126, 211; II-452–453, 470–487
  spherical coordinates I-126, 211

QR algorithm I-170

Rayleigh quotient II-16
regularizer principle II-389–393
resolving operator II-67

Rheinbolt, W.C. II-495
Richardson method
  See Chebyshev method
Rivkind, V.Y. II-495
Rukhovets, L.A. II-495

Samarskii, A.A. I-xxviii; II-82, 417, 496
Saunders, M.A. II-497–498
Seidel method
  See Gauss-Seidel method
separation of variables I-xxv, 171–238; II-343–346, 402–406, 408, 431, 459–461, 478–479
simple iteration method II-65, 86–105, 109, 145, 284, 303, 317–321, 351–354, 377–378
  non-self-adjoint case II-90–105
  parameters II-86–88
  resolving operator II-98–105
  transformation operator II-88–98
singular equations II-303, 326–350
SOR
  See successive over-relaxation method
staircase algorithm I-171, 227–238
  block tridiagonal matrices I-230–231
  stability I-231–236
  tridiagonal matrices I-227–230
steady state method II-55–56
steepest descent method II-156–158, 162–164
successive over-relaxation method (SOR) I-xxviii-xxix; II-189, 199–214, 220–223, 284, 301
  convergence II-200, 220–223
  parameter II-200–204
  spectral radius estimate II-204–207

transformation operator II-67
trapezoid rule II-22
triangular methods II-189, 215–223
  convergence rate II-217–218
  parameter II-219–220

under relaxation II-200

Vainikko, G.M. II-495
Varga, R.S. II-497–498
variation of parameters I-21–26
variational iterative methods II-145–187, 303, 313–315
  three-level II-145, 164–180
    optimality II-176–180
    parameters II-164–174
  two-level II-145–164
    acceleration II-181–187
    asymptotic behavior II-153–156
    convergence II-149–151
    optimality II-151–153
    parameters II-145–149

Wasow, W.R. II-495
weak nonlinearity II-385
Wilkinson, J.H. II-497–498

Young, D.M. II-496–498