

INDEX

- Adams-Bashforth-Moulton method,
335–338, 367
- Adams-Bashforth predictor function,
335–336
- Adams-Moulton corrector function,
335–336
- Array, 14
destroying an array, 16
dynamic memory allocation, *see Memory allocation*
one-dimensional, 15
static memory allocation, *see Memory allocation*
three-dimensional, 21–22
two-dimensional, 17–21
- Applications,
accessing .Net from MFC, 92–94
Cholesky method, 154–156
creativity with keyboard graphics, 88–91
Crout method, 146–149
differentiation and integration,
displaying menu and file i/o, 80–87
drawing a polynomial, 110–115
drawing curves using MyParser, 115–125
eigenvalues and eigenvectors, 302–321
Gaussian elimination method, 134–136
Gauss-seidel method, 170–171
interpolation and approximation,
249–264
nonlinear equations, 208–224
ordinary differential equations, 358–378
partial differential equations, 411–436
scientific calculator, 100–106
simple multiplication calculator, 74–78
skeleton program, 61–68
systems of linear equations, 172–189
Thomas algorithm method, 160–161
- API or *Application programming interface*, 3
- Approximation, 227, 244
- Assembly language, 13
- ANSI, 13
- ASCII or *American Standard Code for Information Interchange*, 87
- Backward-difference rules,
first derivative, 269
second derivative, 269
- Bacteria population growth, 8
- Bisection method, 195–198
- C, 3, 13–14
- C functions,
atof(), 105, 122
fclose(), 79
fopen(), 79
fprintf(), 79
fscanf(), 79
rand(), 85
srand(), 85
time(), 85
- C++, 3, 13–14
- Cartesian coordinate system, 106
conversion to Windows, 107
- Central-difference rules,
first derivative, 270, 348, 352, 355, 384
second derivative, 270, 348, 355, 385,
398
- Chaotic theory, 43
- Characteristic polynomial, 150, 289
- Cholesky method, 131, 143, 150–156
- Class, 3
- Closed interval, 195
- Code2A, 17
- Code2B, 20
- Code2C, 27
- Code2D, 29
- Code2E, 33–34

442 INDEX

- Code2F, 40–43
- Code2G, 49–51
- Code3A, 61–68
 - class,
 - CCode3A, 62
 - files,
 - Code3A.cpp, 63
 - Code3A.h, 62
 - function,
 - OnPaint(), 68, 70
 - output, 66
- Code3B, 74–78
 - class,
 - CCode3B, 74
 - files,
 - Code3B.cpp, 77
 - Code3B.h, 75
 - function,
 - OnMyButton(), 76
 - output, 75
- Code3C, 80–87
 - class,
 - CCode3C, 81
 - files,
 - Code3C.cpp, 83
 - Code3C.h, 83
 - Code3C.rc, 81
 - functions,
 - OnClear(), 86
 - OnExit(), 87
 - OnFileOpen(), 85
 - OnFileSave(), 86
 - OnGenerate(), 85
 - output, 81
- Code3D, 88–91
 - class,
 - CCode3D, 88
 - files,
 - Code3D.cpp, 88
 - Code3D.h, 88
 - functions,
 - OnPaint(), 91
 - OnKeyDown(), 91
 - output, 88
- Code3E, 92–94
 - class,
 - CMain, 93
 - files,
 - Code3E.cpp, 94
 - Code3E.h, 93
- Code4A project, 100–106
 - class,
 - CCode4A, 101
 - files,
 - Code4A.cpp, 100
 - Code4A.h, 103
 - MyParser.obj, 100
 - function,
 - OnPaint(), 103–104
 - output, 101
- Code4B project, 110–115
 - class,
 - CCode4B, 111
 - files,
 - Code4B.cpp, 110
 - Code4B.h, 110
 - function,
 - OnPaint(), 111
 - output, 111, 112
- Code4C project, 115–125
 - class,
 - CCode4C, 118
 - files,
 - Code4C.cpp, 115
 - Code4C.h, 115
 - MyParser.obj, 115
 - functions,
 - DrawCurve(), 118, 120–121, 123
 - OnLButtonDown(), 121–122
 - OnPaint(), 121
 - output, 116
 - schematic drawing, 118
- Code5 project, 172–189
 - class,
 - CCode5, 173
 - files,
 - Code5.cpp, 173
 - Code5.h, 173, 176
 - Code5.rc, 173–174
 - functions,
 - OnCholesky(), 173, 184
 - OnCrout(), 173, 182
 - OnGauss(), 173, 181
 - OnGaussSeidel(), 187
 - OnReset(), 173, 188
 - OnThomas(), 173, 186
 - ReadInput(), 173, 179
 - ShowError(), 173
 - ShowResults(), 173–174

- output, 172
 - schematic drawing, 173
- Code5A, 134–136
- Code5B, 146–149
- Code5C, 154–156
- Code5D, 160–161
- Code5E, 170–171
- Code6 project, 208–224
 - class,
 - CCode6, 209
 - files,
 - Code6.cpp, 209
 - Code6.h, 209
 - MyParser.obj, 209
 - functions,
 - BisectionFPP(), 210, 213, 218
 - DrawCurve(), 210, 213, 223
 - Newton(), 210, 213, 220
 - OnButton(), 210, 213, 217
 - OnLButtonDown(), 210, 213, 216
 - OnPaint(), 210, 213, 215
 - Secant(), 210, 213, 221
 - ShowTable(), 210, 213, 222
 - output, 209
 - schematic drawing, 210
- Code7 project, 249–264
 - class,
 - CCode7, 250
 - files,
 - Code7.cpp, 249
 - Code7.h, 249
 - functions,
 - CSpline(), 251, 253, 257, 260
 - DDifference(), 251, 253, 256
 - DrawCurve(), 251, 253
 - Lagrange(), 251, 253, 255
 - LSquare(), 251, 253, 260
 - OnLButtonDown(), 251, 253
 - OnPaint(), 253–254
 - ShowTable(), 251, 253
 - SolveSLE(), 251
 - output, 251
 - schematic drawing, 251
 - Code8 project, 279–285
 - class,
 - CCode8, 280
 - files,
 - Code8.cpp, 280
 - Code8.h, 280
 - MyParser.obj, 279
 - functions,
 - Differentiation(), 280–282
 - DrawCurve(), 280–281
 - Integration(), 280, 282–285
 - OnLButtonDown(), 280–281
 - OnButton(), 280–281
 - OnPaint(), 280
 - ShowTable(), 280–281
 - output, 279
 - schematic drawing, 281
- Code9 project, 302–321
 - class,
 - CCode9, 305
 - files,
 - Code9.h, 305
 - Code9.cpp, 305
 - functions,
 - OnButton(), 304–30308
 - PowerMtd(), 304, 306, 309–313
 - PowerTable(), 304, 306, 319–320
 - QRMtd(), 304, 306, 313–316
 - QRTable(), 304, 306, 320–321
 - OnPaint(), 306
 - output, 303
 - schematic drawing, 304
- Code10 project, 358–378
 - class,
 - CCode10, 359
 - files,
 - Code10.h, 359
 - Code10.cpp, 359
 - MyParser.obj, 359
 - functions,
 - DrawCurve(), 360
 - ODE1AB(), 360, 367
 - ODE1RK2(), 360, 364
 - ODE1RK4(), 360, 365
 - ODE1System(), 360, 370
 - ODE1Taylor(), 360, 363
 - ODE2toODE1System(), 360, 371
 - ODE2FD1(), 360, 373
 - ODE2FD2(), 360, 376
 - OnLButtonDown(), 360
 - OnButton(), 360
 - SolveSLE(), 360, 378
 - ShowTable(), 360
 - Output, 359
 - schematic drawing, 360
- Code11 project, 411–436
 - class,

444 INDEX

- Code11 project (Contd.)
 - CCode11, 413
 - files,
 - Code11.h, 413
 - Code11.cpp, 413
 - MyParser.obj, 411
 - functions
 - Clear(), 415
 - Heat(), 414–415, 430–434
 - OnButton(), 414–415, 422
 - OnLButtonDown(), 414–415, 421
 - OnPaint(), 414–415, 418
 - Poisson(), 414–415, 425–430
 - SolveSLE(), 414–415
 - ShowTable(), 414–415
 - Wave(), 414–415, 434–436
 - output, 412
 - schematic drawing, 414
- Color management, 59
- Complexity, 16, 19
- Complex number, 43–51
 - addition, 44
 - arithmetic, 48–51
 - conjugate, 46
 - division, 46
 - inverse, 47
 - multiplication, 45
 - subtraction, 44
- Computational fluid dynamics, 8, 157
- Conformal mapping, 43
- Constructor, 67–68
- Crank-Nicolson's method, 398–405
- Crout method, 131, 143–149, 156
- Cubic spline, 239–244
- Curve, 106
 - drawing, 106
 - fitting, 227
 - parametric, 116
 - polynomial, 110–111
- Data passing, 35
- Derivative,
 - analytical, 267
 - numerical, *see numerical differentiation*
- Device context, 58
- Destructor, 68
- Diffusion equation, 382, 438–439
- Dirichlet boundary conditions, 345, 383, 387
- Display context, 58
- Displaying graphics, 58
- Displaying text, 68
- delete, 16–18, 21, 87, 115
- Domain, 106
- Doolittle method, 131, 143, 144, 149
- Edit box, *see Standard control*
- Eigenvalue, 288
 - most dominant, 288, 291
 - least dominant, 288, 291
- Eigenvector, 288
 - Transformation, 289
- Eigen-pair, 289
- Elimination method, 131
- Equation parser, 96–97
 - operand, 97
 - operator, 97
- Euler's method, 329–330
- Event, 67, 69, 103
 - ON_BN_CLICKED, 76, 103, 105, 178, 213, 306, 361, 416
 - ON_COMMAND, 70
 - ON_WM_KEYDOWN, 70
 - ON_WM_LBUTTONDOWN, 70, 120, 122, 213, 361, 416
 - ON_WM_PAINT, 67–68, 70, 103, 119–121, 213, 306, 361, 416
 - ON_WM_RBUTTONDOWN, 70
- False position method, 195, 198–201
- File, 78
 - binary file, 78
 - input/output, 78–80
 - pointer, 79
 - resource, 81
 - text file, 78
- Finite-difference formula, 348, 350, 387–388, 399, 407
 - molecule, 389, 399, 408
- Finite-difference method, 346–347, 373, 376, 391, 409
- Finite-element modeling, 9, 437
- Fixed-point iteration method, 195, 206–208
- Flood control modeling, 11
- Fortran, 13
- Forward-difference rules,
 - first derivative, 269
 - second derivative, 269

- Gauss-Jordan method, 131, 139–141
- Gauss-Seidel method, 131, 162, 165, 167–171
- Gaussian elimination, 131–136, 263
- Gaussian elimination with partial pivoting, 131, 136–139
- Gaussian quadrature method, 276–278, 285
- GDI or *Graphics Device Interface*, 57–59, 69
- Graphical user interface, 3

- Heat equation, 397–405, 430
- Helmholtz's equation, 382, 438–439
- Heun's method, 332–333
- High-performance computing, 13–14
- Householder transformation, 295–297, 316–317

- Intermediate-value theorem, 194
- Interpolation, 227
- ISO or *International Standards Organization*, 13
- Iterative methods, 131, 161–171

- Jacobi method, 131, 162–167

- Keyboard control, 87–91
 - VK_DOWN, 88
 - VK_LEFT, 88
 - VK_RIGHT, 88
 - VK_RETURN, 88
 - VK_SPACE, 88
 - VK_UP, 88
- Klein-Gordon's equation, 382, 406, 438

- Lagrange method, 228–231
- Lagrange operator, 230
- Lagrange polynomial, 230
- Laplace's equation, 382, 394–397
- Least-squares approximation, 244–249
- Legendre polynomials, 276
- Linear equation, 127
- Linear transformation, 108
- Lipschitz condition, 326
- List view box, *see Standard control*
- Lower triangular matrix, *see Matrix*
- LU factorization, 131, 142–161

- Managed extension, 4, 61, 92
- Mathematical modeling, 4
 - black-box model, 5
 - deterministic model, 5
 - heterogeneous model, 5
 - homogeneous model, 5
 - linear model, 5
 - model, 5
 - nonlinear model, 5
 - stochastic model, 5
 - white-box model, 5
- Matrix, 17–21
 - addition, 35–43
 - augmented, 30, 130
 - backward substitutions, 31, 132, 134, 139, 143, 145–146, 157
 - determinant, 28–29
 - diagonally-dominant, 162–163
 - forward substitutions, 143, 145, 157
 - Hermitian, 295
 - ill-conditioned, 136
 - inverse, 28, 30–33, 37–43
 - matrix algebra, 35–40
 - multiplication, 18–22, 36, 38–43
 - norm, 22, 23
 - orthogonal, 55, 295
 - pivot element, 24–26
 - partial pivoting, 136
 - positive-definite, 150–151
 - reduction, 22
 - row operations, 23–26, 31–33, 129, 132–133, 138, 139
 - similar, 295
 - singular, 28
 - square, 28, 30
 - subtraction, 35–43
 - symmetric, 150, 294
 - triangular, 24–25, 243, 259
 - upper, 25, 27, 28, 132
 - lower, 25
 - tridiagonal, 156–157, 349, 353–354
- Mean-value theorem, 194
- Memory, 14, 68
 - dynamic memory allocation, 14–21, 68
 - static memory allocation, 15
- Memory context, 58
- Menu, *see Standard control*
- Message handler, 70
- Method, 69
- MFC or *Microsoft Foundation Classes*, 3, 56–57

446 INDEX

- MFC Application project, 61
- MFC classes,
 - hierarchical chart, 57
 - CBitmap, 58
 - CBrush, 58
 - CButton, 71
 - CClientDC, 58
 - CDC, 57–58
 - CDialog, 173
 - CEdit, 72, 76, 117
 - CFileDialog, 80
 - CFile, 78
 - CFont, 58
 - CFrameWnd, 57, 66, 103, 111, 305
 - CListCtrl, 74
 - CMetaFileDC, 58
 - CPaintDC, 58, 68
 - CPen, 58
 - CPalette, 58
 - CPoint, 110
 - CRect, 74
 - CStatic, 73
 - CString, 72
 - CWindowDC, 58
- MFC files,
 - afxwin.h, 66
 - afxdlgs.h, 83
 - resource.h, 83
- MFC functions,
 - Arc(), 69
 - AfxMessageBox(), 93
 - BitBlt(), 69
 - Create(), 67, 71, 73, 74, 76
 - DoModal(), 80
 - Ellipse(), 69
 - FillRect(), 69
 - FillSolidRect(), 69
 - Format(), 73, 181
 - GetClientRect(), 86
 - GetDlgItemText(), 180
 - GetFileName(), 80, 85–86
 - GetPixel(), 69
 - GetWindowText(), 105
 - Invalidate(), 70
 - InvalidateRect(), 70, 121, 218
 - LineTo(), 69, 110
 - MoveTo(), 69
 - Polyline(), 69
 - PtInRect(), 123, 216, 421
 - Rectangle(), 69
 - RGB(), 59–60, 69
 - SelectObject(), 69
 - SetBkColor(), 69
 - SetDlgItemText(), 181
 - SetFocus(), 76
 - SetPixel(), 69, 110
 - SetTextColor(), 69
 - SetWindowText(), 73, 76
 - TextOut(), 69
- MFC skeleton project, 61
- Micro-electro mechanical system (MEMS), 10
- Microsoft C++, 3
- Microsoft DOS, 3
- Modal window, 80, 177
- Modified Euler-Cauchy method, 333
- Modified Loiuville's equation, 382, 438
- Modular program, 14
- Multistep method, 335
- MyParser, 96–100, 115, 126
 - character codes, 99
 - functions, 100
 - parse(), 98, 105, 123, 219–221, 282–283, 363–377, 427–436
- .Net
 - accessing .Net from MFC, 92–94
 - accessing MFC from .Net, 95
 - framework, 92
 - MFC compatibility, 92
- new, 16–18, 20, 21
- Neumann boundary conditions, 345, 383
- New project, 61
- Newton interpolation methods, 231–239
 - backward-difference, 236–237
 - backward-difference operator, 236
 - divided-difference, 232–233
 - forward-difference, 234–236
 - forward-difference operator, 234
 - Stirling's method, 237–239
- Newton-Raphson method, 195, 201–203
- Nonlinear equation, 127, 193
 - existence of solutions, 194
 - finding the roots, 193
- Numerical differentiation, 268
- Numerical integration, 271
- Numerical methods, 1, 7
- Numerical modeling, 3

- Object, 3, 112
- Object-oriented approach, 14
- Object-oriented programming, 3
- Open interval, 195
- Ordinary differential equations, 324
 - Dirichlet, *see Dirichlet boundary conditions*
 - decay equation, 325
 - damped harmonic oscillator equation, 325
 - degree, 325
 - explicit form, 325
 - first order, 325, 370
 - initial value, 326, 339
 - initial value problem, 326, 328, 332, 334, 337, 339–340
 - system, 338
 - implicit form, 325
 - Neumann, *see Neumann boundary conditions*
 - order, 325
 - pendulum equation, 325
 - second order, 341
 - boundary, 345
 - boundary conditions, 345
 - boundary value problem, 346, 350, 373
 - differentiated boundary conditions, 351–352, 354
 - Dirichlet, *see Dirichlet boundary conditions*
 - initial conditions, 342
 - initial value problem, 342, 344
 - linear, 346
 - Neumann, *see Neumann boundary conditions*
 - reduction to first order system, 343, 371
 - Van der Pol equation, 325
- Parametric curve, *see Curve*
- Partial differential equations, 381
 - boundary value problem, 383
 - boundary conditions, 383, 394, 407
 - degree, 381
 - Dirichlet, *see Dirichlet boundary conditions*
 - elliptic, 382
 - hyperbolic, 382
 - initial conditions, 383, 395, 406
 - linear, 382
 - Neumann, *see Neumann boundary conditions*
 - order, 381
 - parabolic, 382
 - second order, 382
 - Pascal, 13
 - Pixel, 14, 68
 - Poisson's equation, 385–394, 425
 - Potential equation, 385
 - Printed-circuit board design, 9
 - Printer context, 58
 - Polynomial, *see Curve*
 - Power method, 291–292, 310–313
 - Pulldown menu, *see Standard control*
 - Push button, *see Standard control*
 - QR method, 294–302, 313–319
 - QR factorization, 297–302
 - Resources, 71, 81, 101–102
 - Row operations, *see Matrix*
 - RK2, *see Runge-Kutta of order 2 method*
 - RK4, *see Runge-Kutta of order 4 method*
 - Runge-Kutta of order 2 method, 330–333, 364
 - Runge-Kutta of order 4 method, 333–335, 339–341, 365
 - Scalar, 15
 - Secant method, 195, 203–205
 - Shifted Power method, 292–294
 - Simpson's method, 273–275
 - Simpson's 3/8 method, 275
 - Simulation, 6
 - macroscopic, 6
 - mesoscopic, 6
 - microscopic, 6
 - Single-row routing, 10
 - Singular, 106
 - Sink, 10
 - Smartdust, 10
 - SMART project, 11
 - Software development kit (SDK), 3
 - Spline, 239
 - cubic, *see Cubic spline*
 - linear, 239
 - quadratic, 239

448 INDEX

- Standard control , 71
 - edit box, 71, 101–102
 - list view box, 73–74
 - menu, 78
 - pulldown menu, 78
 - push button, 71, 101–102, 209
 - static box, 73, 102
- Static box, *see Standard control*
- Structured program, 14
- Symbolic computing, 267
- System of first order ordinary differential equation, 338–341
- System of linear equations, 127, 290, 349–350, 357, 375, 390, 395, 402, 405, 428
 - consistent solutions, 128–130
 - existence of solutions, 128
 - homogeneous, 128–129
 - inconsistent solutions, 128–130
 - matrix form, 128
 - nonhomogeneous, 128
 - unique solutions, 129
- Taylor series, 268–269, 327
- Taylor series method, 327–330, 362
- Thomas algorithm, 131, 143, 156–161, 243, 259
- Trapezium method, 271–273
- Triangular matrix, *see Matrix*
- Upper triangular matrix, *see Matrix*
- Vector, 15
 - dot product 16, 17
 - magnitude or length, 15
 - norm, 22–23
- Vibrating membrane equation, 382, 438
- Visual C++.Net, 4
- Visualization, 6
- Wave equation, 382, 406–411, 434
- Win32 project, 61–62
- Window, 56
- Windows, 56, 68
 - child, 56
 - coordinate system, 106–110
 - overlapped, 56
 - pop-up, 56, 80
- Wireless sensor networks, 10