

Index

- adaptive Runge–Kutta method, 275–283
- algebra. *See* linear algebraic equations systems;
matrix algebra
- appendices, 409–412
- arithmetic operators, in Python, 7
- arrays
 - accessing/changing, 21
 - copying, 23
 - creating, 19–21
 - functions, 22–23
 - operations on, 21–22
- augmented assignment operators, 7
- augmented coefficient matrix, 28
- backward finite difference approximations,
183
- banded matrix, 56–66
- bisect, 145–146
- bisection method, for equation root, 145–148
- brent, 150–151
- Brent’s method, 148–153
- Bulirsch–Stoer algorithm, 286–290
- Bulirsch–Stoer method, 278–279, 283
 - algorithm, 286–290
 - midpoint method, 283–284
 - Richardson extrapolation, 284–286
- bulStoer, 287
- byte code, 1
- choleski(a), 47–48
- Choleski’s decomposition, 45–52
- cmath module, 18–19
- coefficient matrices, symmetric/banded, 56–66
 - symmetric, 59–60
 - symmetric/pentadiagonal, 61–66
 - tridiagonal, 57–59
- comparison operators, in Python, 8
- composite Simpson’s 1/3 rule, 204
- composite trapezoidal rule, 200–201
- conditionals, in Python, 9
- conjGrad, 86, 90
- conjugate gradient methods, 86, 88–96, 389–400
 - conjugate directions, 390–391
 - Fletcher–Reeves method, 396–400
 - Powell’s method, 391–396
- continuation character, 6
- cubicSpline, 117–119
- cubic splines, 115–121, 195
- curve fitting. *See* interpolation/curve fitting
- cyclic tridiagonal equation, 94
- deflation of polynomials, 172–173
- diagonal dominance, 68
- Doolittle’s decomposition, 42–45
- dot, 3
- eigenvals3, 371–372
- eigenvalue problems. *See* symmetric matrix
eigenvalue problems
- elementary operations, linear algebra, 31
- embedded integration formula, 275
- equivalent linear equation, 31
- error
 - control in Python, 15
 - in finite difference approximations, 186–187
- Euhler’s method, stability of, 272
- Euler–Maclaurin summation formula, 201
- evalPoly, 172
- evaluation of polynomials, 170–172
- exponential functions, fitting, 130–131
- false position method, roots of equations, 179
- finite difference approximations, 181–187
 - errors in, 186–187

420

Index

- finite difference approximations (*cont.*)
 - first central difference approximations, 182–183
 - first noncentral, 183–184
 - second noncentral, 185
- finite elements, 233
- first central difference approximations, 182–183
- first noncentral finite difference approximations, 183–184
- fletcherReeves, 397–398
- Fletcher–Reeves method, 396–400
- forward finite difference approximations, 183
- fourth-order differential equation, 316–320
- fourth-order Runge–Kutta method, 257–265
- functions, in Python, 15–16
- gaussElimin, 37–38
- Gauss elimination method, 33–41
 - algorithm for, 35–38
 - back substitution phase, 37
 - elimination phase, 35–36
 - back substitution phase, 35
 - elimination phase, 34–35
 - multiple sets of equations, 38–41
- Gauss elimination with scaled row pivoting, 68–76
- Gaussian integration, 215–230
 - abscissas/weights for Guaussian quadratures, 221–230
 - Gauss–Chebyshev quadrature, 222
 - Gauss–Hermite quadrature, 223
 - Gauss–Laguerre quadrature, 222–223
 - Gauss–Legendre quadrature, 221–222
 - Gauss quadrature with logarithmic singularity, 224
 - determination of nodal abscissas/weights, 219–221
 - formulas for, 215
 - orthogonal polynomials, 217–219
- Gauss–Jordan elimination, 32
- Gauss–Legendre quadrature over quadrilateral element, 233–240
- gaussNodes, 224–225
- gaussPivot, 71–72
- gaussQuad, 225–226
- gaussQuad2, 235–237
- gaussSeidel, 87–88
- Gauss–Seidel method, 85–88
- gerschgorin, 368–369
- Gerschgorin’s theorem, 368–369
- golden section search, 383–389
- goldSearch, 385–386
- Higher-order equations, shooting method, 301–307
- householder, 362–363
 - householder reduction to tridiagonal form, 357–365
 - accumulated transformation matrix, 361–365
 - householder matrix, 358–359
 - householder reduction of symmetric matrix, 359–361
- Idle (code editor), 3
- ill-conditioning, 29–30
- incremental search method, roots of equations, 143–145
- indirect methods. *See* iterative methods
- initial value problems
 - adaptive Runge–Kutta method, 275–283
 - Bulirsch–Stoer method, 278–279, 283
 - algorithm, 286–290
 - midpoint method, 283–284
 - Richardson extrapolation, 284–286
 - introduction, 248–249
 - multistep methods, 294
 - problem set, 266–271, 290–294
 - Runge–Kutta methods, 255–265
 - fourth-order, 257–265
 - second-order, 255–257
 - stability/stiffness, 271–275
 - stability of Euhler’s method, 272
 - stiffness, 272–275
 - Taylor series method, 249–255
- Input/output
 - printing, 14–15
 - reading, 13
- integration order, 234
- interpolation, derivatives by, 190–195
 - cubic spline interpolant, 195
 - polynomial interpolant, 190–191
- interpolation/curve fitting
 - interpolation with cubic spline, 115–121
- introduction, 103
- least-squares fit, 124–137
 - fitting a straight line, 126
 - fitting linear forms, 126–127
 - polynomial fit, 127–129
 - weighting of data, 129–137
 - fitting exponential functions, 130–131
 - weighted linear regression, 130
- polynomial interpolation, 104–115
 - Lagrange’s method, 104–105
 - limits of, 110–115

421

Index

- Neville's method, 108–110
- Newton's method, 105–108
- problem set, 121–124, 137–140
- rational function interpolation, 140–141
- interval halving method. *See* bisection method
- `inversePower`, 345–346
- `inversePower3`, 373–374
- iterative methods, 85–96
 - conjugate gradient method, 86, 88–96
 - Gauss–Seidel method, 85–88
- `jacobi`, 331
- Jacobian matrix, 235
- Jacobi method, 326–342
 - Jacobi diagonalization, 328–333
 - Jacobi rotation, 327–328
 - similarity transformation/diagonalization, 326–327
 - transformation to standard form, 333–342
- Jenkins–Traub algorithm, 179–180
- knots of spline, 115
- Kronecker delta, 104
- Lagrange's method, 104–105
- Laguerre's method, 173–178
- `lamRange`, 369–370
- LAPACK (Linear Algebra PACKage), 27
- least-squares fit, 124–137
 - fitting linear forms, 126–127
 - fitting straight line, 126
 - polynomial fit, 127–129
 - weighting data, 129–137
 - fitting exponential functions, 130–131
 - weighted linear regression, 130
- linear algebraic equations systems. *See also* matrix algebra
 - back substitution, 32
 - direct methods overview, 31–33
 - elementary operations, 31
 - equivalent equations, 31
 - forward substitution, 32
 - Gauss elimination method, 33–41
 - algorithm for, 35–38
 - back substitution phase, 37
 - elimination phase, 35–36
 - back substitution phase, 35
 - elimination phase, 34–35
 - multiple sets of equations, 38–41
 - ill-conditioning, 29–30
 - introduction, 27–33
 - iterative methods, 85–96
 - conjugate gradient method, 86, 88–96
 - Gauss–Seidel method, 85–88
 - linear systems, 30
 - LU decomposition methods, 41–52
 - Choleski's decomposition, 45–52
 - Doolittle's decomposition, 42–45
 - matrix inversion, 82–84
 - methods of solution, 30–31
 - notation in, 27–28
 - pivoting, 66–76
 - diagonal dominance, 68
 - Gauss elimination with scaled row pivoting, 68–76
 - when to pivot, 74–76
 - problem set, 53–56, 77, 86, 96–101
 - QR decomposition, 101
 - singular value decomposition, 101
 - symmetric/banded coefficient matrices, 56–66
 - symmetric coefficient, 59–60
 - symmetric/pentadiagonal coefficient, 61–66
 - tridiagonal coefficient, 57–59
 - uniqueness of solution, 28–29
- linear forms, fitting, 126–127
- linear systems, 30
- `linInterp`, 297
- lists, 5–7
- loops, 10–11
- LR algorithm, 380
- `LUdecomp`, 44
- `LUdecomp3`, 59
- `LUdecomp5`, 66
- LU decomposition methods, 41–52
 - Choleski's decomposition, 45–52
 - Doolittle's decomposition, 42–45
- `LUpivot`, 72–73
- mathematical functions, 12–13
- `math` module, 17–18
- MATLAB, 2–3
- matrix algebra, 412–417
 - addition, 413
 - determinant, 414–415
 - example, 416–417
 - inverse, 414
 - multiplication, 413–414
 - positive definiteness, 415
 - transpose, 413
 - useful theorems, 415
- matrix inversion, 82–84
- methods of feasible directions, 408

422

Index

- methods of solution, 30–31
 - midpoint, 285–286
 - minimization along line, 383–389
 - bracketing, 383
 - golden section search, 383–389
 - modules, in Python, 17
 - multiple integrals, 233
 - Gauss–Legendre quadrature over quadrilateral element, 233–240
 - quadrature over triangular element, 240–245
 - multistep methods, for initial value problems, 294
 - `NameError`, 23
 - `Namespace`, 23
 - natural cubic spline, 115
 - Nelder–Mead method, 407
 - neville, 109–110
 - Neville’s method, 108–110
 - Newton–Cotes formulas, 199–207
 - composite trapezoidal rule, 200–201
 - recursive trapezoidal rule, 202–203
 - Simpson’s rules, 203–207
 - trapezoidal rule, 200
 - `newtonPoly`, 107–108
 - `newtonRaphson`, 155–156
 - `newtonRaphson2`, 160–161
 - Newton–Raphson method, 153–158
 - norm of matrix, 29
 - notation, 27–28
 - `Numarray` module, 3
 - `numarry` module, 19–23, 7
 - accessing/changing array, 21
 - array functions, 22–23
 - copying arrays, 23
 - creating an array, 19–21
 - operations on arrays, 21–22
 - numerical differentiation
 - derivatives by interpolation, 190–195
 - cubic spline interpolant, 195
 - polynomial interpolant, 190–191
 - finite difference approximations, 181–187
 - errors in, 186–187
 - first central difference approximations, 182–183
 - first noncentral, 183–184
 - second noncentral, 185
 - introduction, 181
 - problem set, 195–197
 - Richardson extrapolation, 187–190
 - numerical instability, 243, 262
 - numerical integration
 - Gaussian integration, 215–230
 - abscissas/weights for Gaussian quadratures, 221–230
 - Gauss–Chebyshev quadrature, 222
 - Gauss–Hermite quadrature, 223
 - Gauss–Laguerre quadrature, 222–223
 - Gauss–Legendre quadrature, 221–222
 - Gauss quadrature with logarithmic singularity, 224
 - determination of nodal abscissas/weights, 219–221
 - formulas for, 215
 - orthogonal polynomials, 217–219
 - introduction, 198–199, 217
 - multiple integrals, 233
 - Gauss–Legendre quadrature over quadrilateral element, 233–240
 - quadrature over triangular element, 240–245
 - Newton–Cotes formulas, 199–207
 - composite trapezoidal rule, 200–201
 - recursive trapezoidal rule, 202–203
 - Simpson’s rules, 203–207
 - trapezoidal rule, 200
 - problem set, 212–215, 230–233, 245–247
 - Romberg integration, 207–212
- operators
 - arithmetic, 7
 - comparison, 8
- optimization
 - conjugate gradient methods, 389–400
 - conjugate directions, 390, 391
 - Fletcher–Reeves method, 396–400
 - Powell’s method, 391–396
- introduction, 381–383
- minimization along line, 383–389
 - bracketing, 383
 - golden section search, 383–389
- Nelder–Mead method, 407
- problem set, 401–407
- simplex method, 407–408
- simulated annealing method, 407
- orthogonal polynomials, 217–219
- overrelaxation factor, 86, 96–101
- piecewise cubic curve, 116
- pivoting, 66–76
 - diagonal dominance, 68
 - Gauss elimination with scaled row pivoting, 68–76
 - when to pivot, 74–76

423

Index

- polyFit, 128–129
- polynomial fit, 127–129
- polynomial interpolant, 190–191
- polynomial interpolation, 104–115
 - Lagrange's method, 104–105
 - limits of, 110–115
 - Neville's method, 108–110
 - Newton's method, 105–108
- polynomials, zeroes of, 170–178
 - deflation of polynomials, 172–173
 - evaluation of polynomials, 170–172
- Laguerre's method, 173–178
- polyRoots, 174–176
- powell, 393–394
- Powell's method, 391–396
- Prandtl stress function, 235
- printing input, 14–15
- printSoln, 251–252
- Python
 - arithmetic operators, 7
 - cmath module, 18–19
 - comparison operators, 8
 - conditionals, 9
 - error control, 15
 - functions, 15–16
 - general information, 1–4
 - obtaining Python, 3–4
 - overview, 1–3
 - lists, 5–7
 - loops, 10–11
 - mathematical functions, 12–13
 - math module, 17–18
 - modules, 17
 - numarray module, 19–23
 - accessing/changing array, 21
 - array functions, 22–23
 - copying arrays, 23
 - creating an array, 19–21
 - operations on arrays, 21–22
 - printing input, 14–15
 - reading input, 13
 - scoping of variables, 23–24
 - strings, 5
 - tuples, 5
 - type conversion, 11–12
 - variables, 4
 - writing/running programs, 24–26
- Python interpreter, 1, 1
- QR algorithm, 380
- quadrature. *See* numerical integration
- quadrature over triangular element, 240–245
- rational function interpolation, 140–141
- reading input, 13
- recursive trapezoidal rule, 202–203
- relaxation, 86
- relaxation factor, 86, 88–96
- Richardson extrapolation, 187–190, 284–286
- romberg, 209–210
- Romberg integration, 207–212
- rootsearch, 144
- roots of equations
 - Brent's method, 148–153
 - false position method, 179
 - incremental search method, 143–145
 - introduction, 142–143
 - Jenkins–Traub algorithm, 179–180
 - method of bisection, 145–148
 - Newton–Raphson method, 153–158
 - problem set, 164–169, 178–179
 - secant method, 179
 - systems of equations, 158–163
 - Newton–Raphson method, 159–163
 - zeroes of polynomials, 170–178
 - deflation of polynomials, 172–173
 - evaluation of polynomials, 170–172
 - Laguerre's method, 173–178
- row pivoting, 56, 67
- Runge–Kutta–Fehlberg formulas, 275–276
- Runge–Kutta methods, 255–265
 - fourth-order, 257–265
 - second-order, 255–257
- run_kut4, 258–259
- run_kut5, 278–279, 283
- scaled row pivoting, 68–76
- secent formula, 179
- second forward finite difference
 - approximations, 185
- second noncentral finite difference
 - approximations, 185
- second-order differential equation, 296–301, 311–316
- second-order Runge–Kutta method, 255–257
- shape functions, 234
- shooting method, 296–307
 - higher-order equations, 301–307
 - second-order differential equation, 296–301
- Shur's factorization, 380
- similarity transformation, 104
- Simpson's 3/8 rule, 204
- Simpson's rules, 203–207
- slicing operator, 3
- sortJacobi, 333
- sparsely populated matrix, 56

424

Index

- stability/stiffness, 271–275
 - stability of Euhler's method, 272
 - stiffness, 272–275
- stdForm, 335–336
- stiffness, 272–275
- straight line, fitting, 126
- strings, 5
- Strum sequence, 365–375
- sturmSeq, 366–367, 375
- swapCols, 70
- swapRows, 70
- symmetric/banded coefficient matrices, 56–66
 - symmetric coefficient, 59–60
 - symmetric/pentadiagonal coefficient, 61–66
 - tridiagonal coefficient, 57–59
- symmetric coefficient matrix, 59–60
- symmetric matrix eigenvalue problems
 - eigenvalues of symmetric tridiagonal matrices, 365–375
 - bracketing eigenvalues, 369–371
 - computation of eigenvalues, 371–372
 - computation of eigenvectors, 373–375
 - Gerschgorin's theorem, 368–369
 - Strum sequence, 365–375
 - householder reduction to tridiagonal form, 357–365
 - accumulated transformation matrix, 361–365
 - householder matrix, 358–359
 - householder reduction of symmetric matrix, 359–361
- introduction, 324–326
- inverse power/power methods, 342–351
 - eigenvalue shifting, 344
 - inverse power method, 342–344
 - power method, 345–351
- Jacobi method, 326–342
 - Jacobi diagonalization, 328–333
 - Jacobi rotation, 327–328
 - similarity transformation/diagonalization, 326–327
 - transformation to standard form, 333–342
- LR algorithm, 380
- problem set, 351–357, 375–380
- QR algorithm, 380
- Shur's factorization, 380
- symmetric/pentadiagonal coefficient matrix, 61–66
- synthetic division, 172–173
- systems of equations
 - Newton–Raphson method, 159–163
 - roots of equations, 158–163
- taylor, 250–251
- Taylor series, 249–255, 409–412
 - function of several variables, 410–412
 - function of single variable, 409–410
- transpose operator, 413
- trapezoid, 202–203
- trapezoidal rule, 200
- triangleQuad, 242
- tridiagonal coefficient matrix, 57–59
- tuples, 5
- two-point boundary value problems
 - finite difference method, 310–320
 - fourth-order differential equation, 316–320
 - second-order differential equation, 311–316
 - introduction, 295–296
 - problem set, 307–310, 320–323
 - shooting method, 296–307
 - higher-order equations, 301–307
 - second-order differential equation, 296–301
- type(a), 13
- type conversion, 11–12
- underrelaxation factor, 86, 90
- variables
 - Python, 4
 - scoping, 23–24
 - Taylor series, 409–410, 410–412
- weighted linear regression, 130
- writing/running programs, in Python, 24–26
- ZeroDivisionError, 15
- zeroes of polynomials, 170–178
 - deflation of polynomials, 172–173
 - evaluation of polynomials, 170–172
 - Laguerre's method, 173–178
- zero offset, 3