

# Index

Note: Page numbers followed by *f* indicate figures, *t* indicate tables and *b* indicate boxes.

## A

**abs**, 14–15, 55–56, 106, 548–549, 583  
Absolute value, 548  
Abstract data types, 384  
Access, 413  
Action of loop, 153  
Addition, complex numbers, 546  
Advanced plotting techniques  
  animation, 435–436  
  applications, 452–457  
  **area** functions, 428–432  
  **bar** functions, 428–432  
  **barh** functions, 428–432  
  core graphics objects, 444–452  
  customizing plots, 428–438  
  file data plotting, 454–457  
  function plotting, 452–454  
  histograms, 432–434  
  log scales, 434–435  
  pie charts, 432–434  
  plot functions, 428–438  
  saving and printing plots, 457–458  
  **stem** functions, 428–432  
  3D plots, 439–444  
Algorithms, 80–81  
**all**, 64–65, 188, 583  
**angle**, 549, 583  
Animation, 435–436  
Anonymous functions, 359, 367–369  
**any**, 64, 188, 583  
API calls, 346  
App Designer, 481, 499–511  
  in Command Window, 500–505  
  components, 499–500

  examples, 508–511  
  simple label, 503*f*  
  slider and label, 509*f*  
  StartupFcn function, 509–510  
  UI figure functions, 505–508  
  **uicontrol** function, 505  
Appending to file, 102–103  
Application Programming Interface (API), 346, 348–349  
**area**, 428–432, 429*f*, 583  
Area plots, 428  
Argument(s), 14–15, 55–59, 108, 112–114, 359–365  
Arithmetic mean, 529, 532  
Array  
  division, 61–62  
  multiplication, 61–62  
  one-dimensional, 38  
  operations, 38, 60–61  
  operators, 62  
ASCII, 19, 26  
**asin**, 26, 583  
**asind**, 26, 583  
**asinh**, 26, 583  
Assignment operator, 6  
Assignment statement, 6–8  
Associativity, 13  
Attributes, 413, 418–419  
Audio file formats, 513  
Audio files. *See* Sound files  
**audioinfo**, 513  
**audioplayer**, 513, 583  
**audioread**, 513  
**audiorecorder**, 513, 583  
**audiowrite**, 513  
**axis**, 95, 97, 100, 428, 583  
**axis off**, 445–446  
**axis square**, 445–446

## B

Banded matrix, 554  
**bar**, 428–432, 429*f*, 436–437, 583  
**bar3**, 439–440, 583  
Bar chart, 99, 428, 431*f*  
  labels from file data, 456*f*  
  matrix data, 431*f*  
  subplot, 437*f*  
**barh**, 428–432, 436–437, 583  
**bar3h**, 439–440, 583  
BarWidth property, 428–430, 434–438  
Base, 384  
Base case, 373  
Base workspace, 115, 222–224  
Binary operator, 12  
Bins, 432  
**blanks**, 249, 583  
Block comment, 84  
Boolean expressions, 19  
Branching statements. *See* Selection statements  
Breakpoint alley, 231–232  
Breakpoints, 231–232, 236  
**bsxfun**, 377  
Bug, 228  
Button groups, 497–499  
  color choice for text, 499*f*  
  radio buttons, 499*f*

## C

Calculus operations  
  differentiation, 572–573  
  integration, 570–572  
  Symbolic Math Toolbox™, 573–575  
  trapezoidal rule, 570–572

- Callback functions, 419, 485, 487, 487*f*
  - Call-by-value method, 208
  - Cascading if-else statement, 134
  - case** expressions, 138–139
  - cast**, 25, 583
  - Casting, 23–26
  - Categorical arrays, 278, 301–302
    - categorical** function, 301
    - categories** function, 301
    - countcats** and **summary** function, 301–302
    - ordinal categorical arrays, 302
    - sorting, 311
  - categorical** function, 301, 583
  - categories**, 301, 583
  - ceil**, 26, 583
  - cell**, 279, 583
  - Cell array, 277–284, 360
    - celldisp**, 281–282
    - cell indexing, 280–282
    - cellplot**, 281–282
    - cellstr**, 283
    - content indexing, 280–281
    - creation, 278–279
    - sorting, 311
    - store strings, 282–284
      - iscellstr**, 284
      - strlength**, 283
    - strsplit** and **strjoin**, 283–284
  - celldisp**, 281–282, 583
  - Cell indexing, 280–282
  - cellplot**, 281–282, 583
  - cellstr**, 283, 583
  - char**, 11, 25, 245–246, 248–250, 265, 280, 583
  - Character, 18–19, 245–249, 255
  - Character encoding, 18
  - Character set, 18
  - Character vectors, 11, 23, 245–249
    - is** function, 264–265
    - operations on, 249–251
      - changing case, 256
      - comparison, 256–258
      - creating and concatenating, 251–255
      - find, replace and separate, 258–262
      - remove trailing/leading blanks, 255
  - checkcode**, 189, 228, 583
  - Child class, 384
  - chirp**, 511, 512*f*
  - circshift** function, 71
  - class**, 11, 368, 583
  - classdef**, 394
  - Class definition, 384, 394–398
    - access methods, 416–418
    - advantages, 420
    - constructor functions, 415–416
    - destructor functions, 419
    - method attributes, 418–419
  - Classes, 10, 383–384
  - clear**, 10, 227–228, 583
  - clearvars**, 9–10, 583
  - clf**, 98–99, 428, 583
  - clock**, 17, 584
  - Code Analyzer Reports, 189
  - Code cells, 235–237
  - coefs**, 540–541
  - collect**, 565, 584
  - Colon operator, 39–40
  - colorbar**, 441–443, 584
  - colorcube**, 513
  - colormap**, 468–474, 584
  - Column major order, 45
  - Column vector, 38, 42
  - Columnwise, 45
  - comet**, 584
  - comet3**, 439, 441, 584
  - Command History Window, 5, 8
  - Commands and functions, 116–117
  - Command Window, 4, 5*f*, 28–29, 204–205, 222–224, 226–227, 231
  - Comma-separated list, 281, 293
  - Comment blocks, 84–85
  - Comments, 83–84
  - Common logarithm, 28
  - Commutative operators, 22
  - Compiler, 81
  - complex**, 584
  - Complex conjugate, 548
  - Complex numbers
    - absolute value/magnitude, 548
    - addition, 546
    - complex conjugate, 548
    - complex equations as polynomials, 548
    - defined, 544
    - equality, 546
    - multiplication, 547–548
    - plotting, 549–550, 550*f*
    - polar form, 549
    - subtraction, 546
  - Complex plane, 549
  - Component library, 501*f*, 503
  - Components, 499–500
  - Computer program, 79
  - Concatenation, 40, 251–255
  - Conditional loops, 153
  - conj**, 584
  - connector**, 28–29, 584
  - Constants, 15–16
  - Constructor functions, 394, 415–416
  - contains**, 265, 584
  - Content indexing, 280–281
  - Continuation operator, 12
  - Control characters, 245–246
  - Conversion character, 88–89
  - Convert numbers to strings, 265–268
  - Copy constructor functions, 415–416
  - Core graphics objects, 444–452
  - cos** function, 100, 101*f*
  - count**, 259, 584
  - countcats** function, 301–302
  - Counted loops, 153
  - cross**, 70–71, 584
  - Cross product, 70–71
  - cummax**, 57–59, 529, 584
  - cummin**, 57–59, 529, 584
  - cumprod**, 57–58, 182, 188, 529–531, 584
  - cumsum**, 57–58, 182, 188, 529–531, 584
  - Current Folder Window, 5
  - Curve-fitting functions, 539–540
    - defined, 527
    - discrete/continuous properties, 539–540
    - extrapolation, 540–543
    - interpolation, 540–543
    - polynomials, 539
  - Curve-Fitting Toolbox™, 538
  - Customizing plots, 94–101
  - cylinder**, 584
- D**
- Database, 278, 295
  - datafun**, 528–529
  - Data sets
    - median**, 533–534
    - mode**, 533
  - Data transfer
    - dlmread**, 349
    - dlmwrite**, 349
    - load**, 327, 329, 331, 335–336, 338–339

- lower-level file I/O functions, 327, 331
  - alternate file input functions, 338–345
  - fscanf**, 338–339
  - opening and closing, 332–333
  - read from files, 334–336
  - textscan**, 338–341
  - write/append to files, 336–338
  - MAT-files, 328–329
  - setpref**, 348
  - weboptions**, 347
  - webread**, 346–348
  - with web sites, 345–349
  - webwrite**, 346–347
  - writing and reading spreadsheet files, 329–331
  - date**, 261–262, 584
  - datetick**, 458
  - dbcont**, 231, 584
  - dbquit**, 231, 584
  - dbstatus**, 236
  - dbstep**, 231, 584
  - dbstop**, 231, 584
  - deblank**, 255, 584
  - Debugging techniques
    - checkcode**, 228
    - Editor/Debugger, 230–232
    - errors types, 228–229
    - function stubs, 232–233
    - tracing, 229–230
  - Default, 11
  - Default input device, 80
  - Default output device, 81
  - deg2rad**, 28, 584
  - delete**, 419
  - Delete elements, 53
  - Delimiter, 259–262
  - demo**, 4, 584
  - Derived class, 384
  - Destructor functions, 419
  - det**, 561, 584
  - diag**, 552, 584
  - Diagonal matrix, 551–552
  - diff**, 59, 66–67, 573, 584
  - Differentiation, 572–573
  - Dimensions of vectors and matrices, 47–52
  - disp**, 88–93, 127–129, 132, 286, 327, 398–399, 545, 584
  - dlmread**, 349
  - dlmwrite**, 349
  - doc**, 4, 584
  - Documentation, 83–84
  - dot**, 70, 584
  - Dot operator, 285–288
  - dot product**, 70
  - double**, 10–11, 23–25, 62–63, 71, 106, 177, 265, 384, 401, 404, 468, 477, 545, 566, 569, 584
  - Double precision, 10
  - Dynamic field name, 289, 308
- E**
- echo**, 230, 584
  - Echo-printing, 156–157
  - Editor, 82*f*, 230–232, 344
  - Efficient code, writing, 188–189
  - Elements, 38
  - elfun**, 26–27
  - Ellipsis, 11–12
  - else** clause, 130–132
  - elseif** clause, 134–138
  - Empty array, 85–86
  - Empty vectors, 52–54
  - end**, 49, 202, 394, 584
  - End of the file, 334, 338, 343–344
  - Endpoint, 346
  - endsWith**, 265, 584
  - Equality
    - for complex numbers, 546
    - operator for, 19
  - erase**, 255–256, 585
  - errorbar**, 458
  - Error-checking, 131, 176–179
  - error** function, 132, 585
  - Error message, 87
  - Errors, types, 228–229
  - eval**, 585
  - Event-driven programming, 419–420, 485
  - Events, 419–420, 485
  - Executable file, 81
  - Execute a program, 81
  - Execution-time errors, 228–229
  - exit**, 4, 585
  - exp**, 28, 215–216, 585
  - expand**, 565, 585
  - Extending a vector, 174
  - External file, 80
  - Extrapolation, 540–543
  - eye**, 554, 585
  - ezplot**, 585
- F**
- FaceAlpha property, 443–444, 443*f*
  - FaceColor property, 450–451
  - Faces, 450
  - FaceVertexCData, 450–451
  - factor**, 565, 585
  - factorial**, 169, 182, 369, 374–375, 585
  - false**, 21–22, 64, 585
  - fclose**, 333–334, 337–340, 585
  - feof**, 334, 585
  - feval**, 372, 585
  - fgetl**, 334–339, 585
  - fgets**, 334, 585
  - fieldnames**, 288, 585
  - Fields, 277, 284
  - Field width, 90
  - figure**, 98–99, 481, 585
  - Figure Window, 95
  - File identifier, 332
  - File input and output (file I/O), 101–106, 327–345
    - appending data to data file, 103
    - load** function, 327, 329, 331, 335–336, 338–339
    - lower-level file I/O functions, 327, 331
      - alternate file input functions, 338–345
      - fscanf**, 338–339
      - opening and closing, 332–333
      - read from files, 334–336
      - textscan**, 338–341
      - write/append to files, 336–338
    - reading from file, 103–106
    - save** function, 327–328, 331
    - writing data to file, 102–103
  - File types, 327
  - find**, 64–67, 188, 585
  - fix**, 26, 106, 585
  - flip**, 50, 585
  - fliplr**, 50, 585
  - flipud**, 50, 106, 585
  - Floating point, 10
  - floor**, 26, 585
  - fopen**, 332–333, 335, 337–340, 585
  - for loop, 153–161, 375, 436
    - general form, 154
    - input in, 156–157
    - iterator variable, 156
    - preallocating vectors, 158–159
    - subplot, 159–161

for loop (*Continued*)  
   sums and products, finding,  
     157–158  
**format**, 11–12, 585  
 Format specifier, 88  
 Formatting, 88  
**fplot**, 371–372, 372*b*, 567–568, 567*f*,  
   585  
**fprintf**, 88–94, 117, 132, 163, 251,  
   253, 286, 327, 336–339, 341,  
   545, 585  
**fread**, 348  
**frewind**, 348  
**fscanf**, 338–339, 585  
**fseek**, 348  
**func2str**, 370–371, 585  
 Function arguments, 14–15, 55–59  
**function call**, 107  
 Function functions, 359, 369–372  
 Function handles, 359, 367–373  
 Functions, 383  
   anonymous functions, 367–369  
   calling, 14–15, 109–111  
   commands and, 116–117  
   definitions, 107–109, 376  
   function handles, 367–373  
   with local variables, 114–115  
   nested functions, 366–367  
   passing structures to, 287–288  
   recursive functions, 373–377  
   stubs, 232–233  
   subfunctions, 213–215  
   user-defined  
     categories, 201–209  
     does not return any values,  
       206–207  
     function definition, 202–203,  
       206  
     passing arguments, 208–209  
     returns more than one value,  
       202–206  
     that returns values, 207–208  
   variable number of arguments,  
     359–365  
     input arguments, 360–362  
     output arguments, 362–365  
**fwrite**, 348  
**fzero**, 372–373, 585

## G

**gallery** function, 71  
 Gauss-Jordan method, 562  
**gca**, 585

**gcf**, 585  
**ge**, 407  
 General (inductive) case, 373  
**geomean**, 532, 534  
 Geometric mean, 531  
**get**, 585  
**GetAccess**, 413  
**getaudiodata**, 513–514, 585  
**getframe**, 586  
**ginput**, 458, 586  
 Global stream, 17  
 Global variables, 224  
**gong**, 511  
**graph2d**, 95  
**graph3d**, 95  
 Graphical User Interface  
   Development Environment  
     (GUIDE), 481  
 Graphical user interfaces (GUIs), 420  
   callback functions, 491*f*  
   editable text box, 487*f*  
   multiple callback functions,  
     490–491  
   objects, 384, 393–420  
     button groups, 497–499  
     push buttons, 484–494  
     sliders, 484–494  
     text boxes, 484–494  
     units property, 484  
   SelectedObject property, 497  
   static text box, 483*f*  
   Visible property, 482  
 Graphics  
   core objects, 444–452  
   objects, 384–393  
   and plot properties, 385–393  
   primitives, 444–445  
 Graphics objects, 444–452  
**grid**, 98–99, 428, 439, 586  
**groot**, 385, 482  
**gt**, 407  
**gtext**, 447, 458, 586  
**guihandles**, 514  
 GUIs. *See* Graphical user interfaces  
   (GUIs)

## H

**handel**, 384, 511  
 Handle classes, 384, 403–413  
**harmmean**, 532, 534  
 Harmonic mean, 531  
**help**, 4, 14, 26, 43, 97–99, 111, 117,  
   204, 214–215, 586

Help browser, 90  
 Help command, 84  
**help topics**, 14  
 High-level languages, 81  
**hilb** function, 71  
**histogram**, 428, 432–434, 432*f*, 529,  
   586  
**hold**, 98–99, 428, 586  
**hold on**, 99, 101  
 Hybrid languages, 383

## I

**i**, 16, 586  
 Identifier names, 9  
 Identity matrix, 553–554  
 if-else statement, 130–132, 374–375  
 if statement, 125–130  
   nested for loops and, combining,  
     166–168  
**imag**, 586  
**image**, 470, 475, 479*f*, 586  
 Image files, 477–480  
 Image processing  
   alternate number types for image  
     matrices, 480  
   colormaps, 468–474  
   image files, 477–480  
   RGB matrices (*see* True color  
     matrices)  
   true color matrices, 468, 474–477  
**imagesc**, 514  
**im2double**, 586  
**imfinfo**, 513  
**imread**, 477, 586  
**imshow**, 478, 586  
**imwrite**, 477, 586  
 Index, 40  
 Index vector, 41, 312–315  
**ind2sub** function, 71  
**inf**, 16, 339, 586  
 Infinite loop, 169  
 Infinite recursion, 374  
 Inheritance, 384  
   handle classes, 403–413  
   subclasses, 401–403  
   value classes, 403–413  
 Inner dimensions, 68  
 Inner functions, 366  
 Inner loop, 161  
 Inner parentheses, 13  
 Inner product, 70  
**input**, 85–89, 93–94, 143–144, 249,  
   253, 586

Input arguments, 359–362  
Instances, 384

Instantiation, 384

**int**, 573, 586

**int8**, 10, 586

**int16**, 10, 586

**int32**, 10, 23–25, 177, 586

**int64**, 10, 586

Integers, error-checking for,  
177–179

Integration, 570–572

Interpolation, 540–543

Interpreter, 81–82

**intersect**, 586

**intmax**, 23, 586

**intmin**, 23, 586

**int2str**, 266, 586

**inv**, 556–557, 559, 563, 586

**isa**, 586

**isbanded**, 556, 586

**iscellstr**, 284, 586

**ischar**, 145, 264, 586

**iscolumn**, 145

**isdiag**, 556, 586

**isempty**, 143–144, 178, 250,  
586

**isequal**, 65, 142, 188, 586

**isfield**, 288–289, 586

**isfloat**, 145

**is function**, 142–145, 264–265

**isinteger function**, 145

**iskeyword**, 144–145, 586

**isletter**, 142, 188, 264, 586

**islogical function**, 145

**ismember**, 535, 586

**isnumeric function**, 145

**isreal**, 145, 545–546, 587

**isrow function**, 145

**issorted**, 587

**issortedrows**, 311, 587

**isspace**, 264, 587

**isstr function**, 145

**isstring**, 264, 587

**isStringScalar**, 264–265, 587

**isstrprop**, 265, 587

**isstruct**, 288, 587

**issymmetric**, 556, 587

**istril**, 556, 587

**istriu**, 556, 587

**isvarname function**, 145

**isvector function**, 145

Iterate, 39

Iterator variable, 154, 156

## J

**j**, 16, 587

JavaScript Object Notation (JSON),  
346–348

**jet**, 469, 471*f*, 587

**join**, 263–264, 587

**jsondecode**, 347–348

**jsonencode**, 347–348

## K

Keywords, 9

## L

**laughter**, 511

**le**, 407

Leading blanks, 90, 249

**legend**, 98–99, 101, 428, 587

**length**, 47–48, 72, 86, 107, 587

**limit**, 574, 587

**line**, 384–385, 445, 446*f*, 587

background color, 447*f*

modified edge color, 447*f*

text box, 447*f*

use of, 446*f*

Linear algebraic equations, 557–563

augmented matrix, 562–563

Reduced Row Echelon Form, 562

Symbolic Math Toolbox™, 551

2 x 2 systems, 559–561

Linear indexing, 45

Line types, 98

**linspace**, 39–40, 87–88, 587

**listdlg**, 140–142, 141*f*, 587

Listeners, 419–420

Live Editor, 233–234, 234–235*f*

Live script, 80, 233–237

**load**, 102–104, 106, 117, 327, 329,

331, 335–336, 338–339, 587

Locale setting, 26

Local functions, 115–116, 213–215

Local variables, 114–115, 222–223

**log**, 28, 587

**log2**, 587

**log10**, 587

Logarithm, 28

**logical**, 11, 19–20, 188, 587

Logical arrays, 64, 71

Logical built-in functions, 64–67

Logical errors, 228–229

Logical expressions, 19

Logical indexing, 63

Logical operators, 19, 21, 22*t*

**logical true/false**, 62–65, 67, 72,  
129–130, 143

**logical vectors**, 62–67

**loglog**, 434–435, 587

**log2**, **log10**, 28

Log scales, 434–435

**logspace**, 40, 587

**lookfor**, 4, 84, 587

Looping statements

for loop, 154–161

input, 156–157

iterator variable, 156

preallocating vectors, 158–159

subplot, 159–161

sums and products, 157–158

nested for loops, 161–168

timing, 189–191

vectorizing, 179–189

sums and products, 180–183

writing efficient code, 188–189

while loops, 168–179

counting, 174–175

input in, 172–174

multiple conditions, 170

reading from file, 170–172

Loop variable, 154

**lower**, 256, 587

Lower-level file I/O functions, 327,  
331

alternate file input functions,  
338–345

**fscanf**, 338–339

opening and closing, 332–333

**fclose**, 333–334

file identifier, 332

**fopen**, 332–333

permission string, 332

read from files, 334–336

**fgetl**, 334–336

**fgets**, 334

**fscanf**, 338–345

**textscan**, 338–345

**textscan**, 338–341

write/append to files, 336–338

Lower triangular matrix, 554–555

**lt**, 407

## M

Machine language, 81

Maclaurin series, 216

**magic function**, 71

Main function, 213–215, 224

Main program, 210, 216–217, 232

Markers, 98  
 MAT-files, 328–329  
   appending variables to, 329  
   reading from, 329  
   writing variables, 328–329  
 MathWorks, 28  
 MathWorks Cloud, 28–29  
 MATLAB Command Window, 4, 5*f*  
 MATLAB Drive, 29  
 MATLAB Mobile, 28–29  
 MATLAB Online™, 29  
 MATLAB programs, 4  
   algorithms, 80–81  
   code cells, 235–237  
   commands and functions, 116–117  
   debugging techniques  
     **checkcode**, 228  
     Editor/Debugger, 230–232  
     errors types, 228–229  
     function stubs, 232–233  
     tracing, 229–230  
   file input/output, 101–106  
     appending data to data file, 103  
     reading from file, 103–106  
     writing data to file, 102–103  
   input function, 85–88  
   live script, 233–237  
   local functions, 213–215  
   local functions in scripts,  
     115–116  
   menu-driven modular program,  
     215–222  
   modular programs, 210–212  
   output function, 88–93  
   scripts, 81–84  
     customizing plots, 94–101  
     documentation, 83–84  
     with input and output, 93–94  
   user-defined functions, 106–115  
     calling from script, 111–112  
     calling function, 109–111  
     categories, 201–209  
     does not return any values,  
       206–207  
     function definition, 202–203,  
       206  
     function definitions, 107–109  
     local variables, 114–115  
     passing arguments, 208–209  
     passing multiple arguments,  
       112–114  
     returns more than one value,  
       202–206

  scope of variables, 115  
   that returns values, 207–208  
   variable scope, 222–227  
     base workspace, 222  
     global variables, 224  
     local variables, 222–223  
     persistent variables, 225–227  
 Matrix  
   augmentation, 557  
   elements, 44–46  
   multiplication, 68–71  
   operations, 551–557, 562–563  
   properties, 551–557  
   square, 551–556  
   three-dimensional, 54–55  
   variables, 42–46  
**max**, 56–58, 188, 436, 528–529, 587  
**maxk**, 529, 587  
**mean**, 529–532, 587  
**median**, 533–534, 587  
**menu**, 140–141, 145, 587  
 Menu-driven modular program,  
   215–222  
**mesh**, 441, 588  
**meshc**, 458  
**meshgrid**, 71, 444*f*, 588  
**methods**, 384, 394–396, 588  
   access methods, 416–418  
   attributes, 418–419  
   constructor functions, 415–416  
   destructor functions, 419  
**min**, 56–57, 188, 436, 528–529, 588  
**mink**, 529, 588  
**mlock**, 237  
 Mnemonic, 9  
**mobiledev**, 28–29, 588  
**mod**, 26–27, 588  
**mode**, 102, 533, 588  
 Modular approach, 80  
 Modular programs, 210–212,  
   215–222  
**movegui**, 482, 588  
**movie**, 588  
 Multiple arguments, 112–114  
 Multiple components properties,  
   508  
 Multiplication, complex numbers,  
   547–548  
**munlock**, 237

**N**  
**namelengthmax**, 9, 588  
**NaN**, 16, 588

**nargin**, 360–361, 377, 394, 396–397,  
   400, 588  
**narginchk**, 377  
**nargout**, 364, 588  
**nargoutchk**, 377  
 Natural exponential base, 215–216  
 Natural logarithm, 28  
**ndims** function, 71  
 Nested for loops, 161–168  
 Nested functions, 359, 366–367  
 Nested if-else statements, 132–138  
 Nested parentheses, 13  
 Nested structure, 297–300  
**newline**, 246, 588  
 Newline character, 89, 246  
**nthroot**, 27–28, 588  
**numden**, 566–567, 588  
**numel**, 48, 72, 588  
 Numerical expressions, 11–18  
 Numerical functions, 26–28  
**num2str**, 266, 588

## O

Object code, 81  
 Object handle, 384  
 Object-oriented programming (OOP)  
   abstract data types, 384  
   built-in data types, 384  
   class definition, 384, 394–398  
   classes, 10, 383–384  
   constructor functions, 394,  
     415–416  
   graphics and plot properties,  
     385–393  
   inheritance, 384  
     handle classes, 403–413  
     subclasses, 401–403  
     value classes, 403–413  
   instances, 384  
   objects, 393–420  
   user-defined classes and objects  
     access methods, 416–418  
     advantages, 420  
     class definition, 384, 394–398  
     constructor functions, 415–416  
     destructor functions, 419  
     method attributes, 418–419  
     value classes, 384, 403–413  
 Objects, 384, 393–420  
**ode23**, 377  
**ode45**, 377  
**odeset**, 377  
 One-dimensional array, 38



**ones**, 44, 54, 60, 64, 588

Open interval, 16

Operators

binary, 12

dot, 285–288

logical, 19, 21, 22t

with numerical expressions,  
12–14

precedence rules, 13–14, 22–23,  
22t, 67t

scientific/exponential notation, 13

short-circuit, 21

unary, 12

Ordinal categorical arrays, 302,  
311–312

Ordinary differential equation  
(ODE), 377

Ordinary method, 397

Outer dimensions, 68

Outer functions, 366

Outer loop, 161

**outer product**, 70–71

Output arguments, 359–360,  
362–365

Output, scripts with, 93–94

Output statements, 88–93

Overloading functions, 398–400

## P

Parent, 384

**parula**, 469–470, 470f, 473–474, 588

**pascal** function, 71

**patch**, 449–452, 451–452f, 588

**pause** function, 191

Permission string, 332

Persistent variables, 225–227

**pi**, 15–16, 588

**pie**, 433–434, 588

**pie3**, 439, 588

Pie charts, 428, 432–434, 433–434f,  
438–439

exploding, 438f  
labels from file data, 456f

**pink**, 588

Pixels, 467–468

Place holder, 88–89

**play**, 588

**plot**, 95–98, 550–551, 588

**plot3**, 588. *See also* Three-  
dimensional plots

Plot symbols, 98

Plotting, 549–550. *See also* Advanced  
plotting techniques

animation, 435–436

applications, 452–457

**colorbar** functions, 441–443

colormaps, 473–474

customizing, 97–98

file data plotting, 454–457

GUI Figure Window, 495f

properties, 436

saving and printing plots,  
457–458

3D pie chart, 441f

3D plots, 427, 439–444, 439f

**plotyy**, 458

**plus**, 15, 250, 588

Polar form, 549

**polarplot**, 549, 588

**polyder**, 572–573, 589

**polyfit**, 540–541, 589

**polyint**, 572, 589

Polynomials, 539, 548

**poly2sym**, 564–565, 589

**polytemp**, 541

**polyval**, 539, 541–542, 548–549,  
572–573, 589

Preallocating vectors, 158–159

Precedence, 13

**pretty**, 567, 589

Primitive objects, 384–385

**print**, 589

Printing vectors and matrices, 91–93

Private access, 413, 418

Procedural programming language,  
383

**prod**, 56–57, 182, 188, 529–531, 589

**profile**, 589

**profile off**, 191

**profile on**, 191

**profile viewer**, 191

Programming concepts, 530b

Programs. *See* MATLAB programs

Prompting, 4, 80

**properties**, 384, 394–396, 413–415,  
589

Protected access, 413, 418

Pseudorandom, 16

Public access, 413, 418

Publishing code, 235–237

Push buttons, 484–494

## Q

**quad**, 571–572, 589

Query parameters, 346, 348–349

**quit**, 4, 589

## R

**rad2deg**, 28, 589

**rand**, 16–17, 43–44, 54, 589

**randi**, 43, 589

**randn**, 17, 589

Random numbers

global stream, 17

pseudorandom, 16

**rand** function, 16–17

**randi** function, 17–18

**randn** function, 17

**rng** function, 16–17

seed, 16

**range** function, 23–26, 71

Reading from file. *See also* File input  
and output (file I/O)

MATLAB programs, 103–106

while loops, 170–172

**real**, 589

Record, 278

**recordblocking**, 513, 589

**rectangle**, 448–449f, 589

Recursive functions

base case, 373

defined, 359, 373

general (inductive) case, 373

infinite recursion, 374

Reduced Row Echelon Form, 562

Relational expressions, 19–23

logical operators, 19, 21, 22t

operator precedence rules, 22–23,  
22t

relational operators, 19

with vectors and matrices, 62–64

Relational operators, 19

**rem**, 26–27, 589

**repelem**, 52, 589

**repmat**, 51, 589

Reserved words, 9

**reshape**, 49–50, 589

RESTful, 346–347

Return value(s), 14–15

**rgb2gray**, 589

RGB matrices, 468, 474, 480

**rmfield**, 286–287, 589

**rng**, 16–17, 589

**roots**, 385, 548, 589

**rot90**, 50–51, 589

**round**, 26–27, 589

Roundoff errors, 23

Row vector, 38–41

**rref**, 562–563, 589

Run a program, 81

Running sum, 57, 157  
Runtime errors, 228–229

## S

Sampling frequency, 511  
Saturation arithmetic, 24  
**save**, 102–104, 106, 327–328, 331, 589  
Scalar and array operations, 59–62  
Scalar multiplication, 60  
Scalars, 21, 38  
Scientific/exponential notation, 13  
Scope of variables, 115  
Script files, 4, 81–83  
Scripts, 79–84  
    customizing plots, 94–101  
    documentation, 83–84  
    with input and output, 93–94  
    local functions in, 115–116  
    user-defined function from, 111–112  
Seed, 16  
SelectedObject property, 497  
Selection sort, 305–306, 305*b*  
Selection statements, 125  
    if-else statement, 130–132  
    if statement, 125–130  
    is functions, 142–145  
    nested if-else statements, 132–138  
    switch statement, 138–142  
**semilogx**, 434–435, 589  
**semilogy**, 434–435, 435*f*, 589  
**sendmail**, 348  
Sentinel, 170–171  
**set**, 590  
**SetAccess**, 413  
**setdiff**, 590  
Set operations  
    **intersect** function, 535–538  
    **ismember** function, 537–538  
    **issorted** function, 535  
    **setdiff** function, 535–536  
    **setxor** function, 535–536  
    **union** function, 535–536  
    **unique** function, 535–536  
**setpref**, 348  
**setxor**, 590  
**shftdim** function, 71  
Short-circuit operators, 21  
Side effects, 207  
**sign**, 26–27, 56, 66–67, 184, 590  
Simplification functions, 565–566  
**simplify**, 590

Simpson's method, 571–572  
Simulink toolboxes, 593  
**sin**, 14, 26, 100, 101*f*, 106, 131, 370–372, 370*f*, 372*f*, 435–436, 590  
**sind**, 26, 131, 590  
**single**, 10, 590  
**sinh**, 26, 590  
**size**, 47–48, 136, 590  
Sliders, 484–494  
    App Designer, 505–511  
    GUI, 494–496  
**solve**, 568, 590  
**sort**, 306, 309–311, 590  
Sorting, 278, 304–312  
    ascending/descending, 304, 306  
    categorical arrays, 311–312  
    **issortedrows** function, 311  
    programming concept, 305*b*  
    selection sort, 305–306, 305*b*  
    **sortrows** function, 310–311  
    strings, 309–312  
    tables, 310–311  
    vector of structures, 306–309  
**sortrows**, 310–311, 590  
**sound**, 590  
Sound files, 511–514  
Sound wave, 511  
**sphere**, 590  
    functions, 441, 474*f*  
    mesh plot, 442*f*  
    surf plot, FaceAlpha modified, 442–443*f*  
**spiral**, 440–441, 590  
**splat**, 511  
Spreadsheet files, 329–331  
**sprintf**, 251–255, 253*f*, 428, 590  
**sqrt**, 27–28, 532–533, 590  
Square matrix, 43, 551–556  
Standard deviation, 532–533  
Standard error, 332–333  
Standard input, 332–333  
Standard output, 332–333  
**startsWith**, 265, 590  
Statistical functions  
    geometric mean, 531  
    harmonic mean, 531  
    **mean**, 529–532  
    **median**, 533–534  
    **mode**, 533  
    standard deviation, 532–533  
    variance, 532–533

Statistics, 528  
**std**, 532–533, 590  
**stem**, 428–432, 429*f*, 590  
**stem3**, 439, 590  
Step value, 39  
**strcat**, 251, 590  
**strcmp**, 256–257, 590  
**strcmpi**, 590  
**str2double**, 266–268, 336, 590  
**strfind**, 258, 590  
**str2func**, 371, 452–454, 590  
**string**, 247, 250–251, 590  
String arrays, 245–249, 262–263  
Strings, 11, 18–19  
    convert numbers to strings, 265–268  
    **is** function, 264–265  
    operations on, 250–251  
        changing case, 256  
        comparison, 256–258  
        creating and concatenating, 251–255  
        find, replace and separate, 258–262  
        remove trailing/leading blanks, 255  
        sorting, 309–312  
**strings**, 250, 262, 267–268, 590  
String scalar, 246–247  
**strip**, 255, 590  
**strjoin**, 263, 283, 591  
**strlen**, 247, 250, 262, 283, 591  
**strncmp**, 257, 591  
**strncmpi**, 257, 591  
**str2num**, 266–268, 590  
**strrep**, 259, 261, 591  
**strsplit**, 263, 283–284, 591  
**strtok**, 259, 261–262, 336, 375, 591  
**strtrim**, 255, 591  
**struct**, 284–285, 591  
Structures, 277, 284–300  
    **disp**, 286  
    dot operator, 285–288  
    **fieldnames**, 288  
    fields, 284  
    **fprintf** function, 286  
    index vectors, 312–315  
    **isfield**, 288  
    **isstruct**, 288  
    nested structure, 297–300  
    **rmfield**, 286–287  
    sorting vector of structures, 306–309



**struct** function, 284–285  
 vector of structures, 289–297  
 Subclasses, 384, 401–403  
**sub2ind** function, 71  
**subplot**, 159–161, 428–430,  
 434–437, 511, 542–543, 591  
 file types, 453f  
**subs**, 565–566, 591  
 Subscript, 40  
 Subscripted indexing, 44  
 Substring, 249  
 Subtraction, complex numbers, 546  
**sum**, 56–58, 63, 71, 188, 293,  
 296–297, 529–531, 591  
**summary**, 301–302, 591  
 Sums and products  
 finding, 157–158  
 vectorizing, 180–183  
 Superclass, 384, 401  
**surf**, 441, 591  
**surfc**, 458  
 Switch statement, 138–142  
**sym**, 563–564, 591  
 Symbolic mathematics, 551  
 defined, 528  
 expressions, 563–565, 567  
 simplification functions, 565–566  
 solving equations, 568–569  
 variables, 563–565  
**symmetric**, 555–556  
**sym2poly**, 564–565, 591  
**syms**, 564, 591  
 Syntax errors, 228

## T

Tab completion, 15  
**table** function, 302–303, 591  
 Tables, 278, 302–304  
 Temporary variable, 128  
 Text. *See also* Character; Character  
 vectors; Strings  
 convert text to numbers, 265–268  
 operations on, 249–263  
**text**, 447, 591  
**text box**, 446–447, 484–494  
**textscan**, 338–341, 456–457, 591  
 Three-dimensional bar chart, 440f  
 Three-dimensional matrix, 54–55,  
 480  
 Three-dimensional pie chart, 441f  
 Three-dimensional plots, 427,  
 439–444, 439f  
 Three-dimensional space, 451–452

Throwing an error, 132  
**tic/toc**, 189–190, 373, 591  
**timeit**, 373, 591  
 Timing functions, 189–191, 373  
**title**, 95, 428, 591  
**toggle**, 99  
 Token, 259–260  
 Toolstrip, 6, 82f  
 Top-down design approach, 80  
**trace**, 552, 591  
 Tracing, 229–230  
 Trailing blanks, 249  
 Trailing zeros, 90  
**train**, 511, 512f  
 Transpose, 42  
 Trapezoidal rule, 570–572  
**trapz**, 591  
 Tridiagonal matrix, 554  
 Trigonometric functions, 26  
**tril**, 555, 591  
**trimmean**, 532  
**triu**, 555, 591  
**true**, 21–22, 64, 399–400, 591  
 True color matrices, 468, 474–477  
 Truth table, 22, 22t  
**try/catch** functions, 145  
**type**, 82–83, 102–103, 335, 337,  
 591  
 Type casting, 23–26  
 Type ranges, 23–26

## U

**uibuttongroup**, 481, 497, 514, 591  
**uicontrol**, 482, 505, 591  
**uifigure**, 505  
 UI Figure Functions, 505–508  
 UI Figure Window, 501–502,  
 505–506f  
**uilabel**, 505  
**uint8**, 10, 468, 592  
**uint16**, 10, 591  
**uint32**, 10, 591  
**uint64**, 10, 591  
**uipanel**, 481, 592  
**uislider**, 505–506  
**uitable**, 514  
 Unary operator, 12  
**union**, 592  
**unique**, 592  
 Unsigned integer types, 10  
 Unwinds, 45  
**upper**, 256, 592  
 Upper triangular matrix, 555

User, 6, 85  
 User-defined classes and objects  
 access methods, 416–418  
 advantages, 420  
 class definition, 384, 394–398  
 constructor functions, 415–416  
 destructor functions, 419  
 inheritance and handle class,  
 401–413  
 method attributes, 418–419  
 overloading functions and  
 operators, 398–400  
 property definition blocks,  
 413–415  
 User-defined function, 106–115, 367.  
*See also* Functions  
 calling from script, 111–112  
 calling function, 109–111  
 categories, 201–209  
 does not return any values,  
 206–207  
 function definition, 107–109,  
 202–203, 206  
 local variables, 114–115  
 passing arguments, 208–209  
 passing multiple arguments,  
 112–114  
 returns more than one value,  
 202–206  
 output arguments, 203, 205  
 scope of variables, 115  
 that returns values, 207–208  
 User-defined **handle** class, 404–405

## V

Value classes, 384, 403–413  
**var**, 532, 592  
**varargin**, 361–362, 377, 396–397,  
 592  
**varargout**, 362–363, 365, 592  
 Variable number of arguments,  
 359–365  
 input arguments, 360–362  
 output arguments, 362–365  
 Variables  
 assignment statement, 6–11  
 decrementing, 9  
 identifier names, 9–10  
 incrementing, 8  
 initializing, 8  
 MAT-files, 328–329  
 types, 10–11, 23–26  
**char**, 11

Variables (*Continued*)

- classes, 10
- integer, 10
- logical**, 11
- single** and **double**, 10
- string**, 11
- unsigned, 10
- Variable scope, 222–227
  - base workspace, 222
  - global variables, 224
  - local variables, 222–223
  - persistent variables, 225–227
- Variance, 532–533
- Vectorized code, 154, 179–189
- Vectorizing, 179–189
  - selection statements, 184–188
  - sums and products, 180–183
  - writing efficient code, 188–189
- Vectors and matrices, 38–55
  - column vectors, 42
  - dimensions, 47–52
  - elements, referring and modifying, 40–41
  - empty vectors, 52–54
  - extending, 174

- as function arguments, 55–59
- logical built-in functions, 64–67
- loops with, 179–189
- matrix multiplication, 68–71
- matrix variables, 42–46
- preallocating, 158–159
- printing, 91–93
- relational expressions with, 62–64
- row vectors, 38–41
- scalar and array operations on, 59–62
- of structures, 306–309, 314–315
- three-dimensional matrices, 54–55
- of variables, 47
- Vertices, 450
- Visible property, 482

**W**

- weboptions**, 347
- webread**, 346–348
- webwrite**, 346–347
- While loops, 153–154, 168–179
  - counting, 174–175
  - input in, 172–174
  - multiple conditions, 170

- reading from file, 170–172
- White space characters, 245–246
- who**, 9, 328, 592
- whos**, 9, 592
- Workspace Window, 5–8, 10–11, 24, 29, 545
- Writing to file, 102–103

**X**

- xlabel**, 95, 428, 592
- xlim**, 458
- xlsread**, 329–331, 592
- xlswrite**, 329–330, 592
- xor**, 21, 592
- xticklabels**, 454–455, 455*f*, 592

**Y**

- ylabel**, 95, 428, 592
- ylim**, 458

**Z**

- Zero crossings, 66–67
- zeros**, 44, 48, 54, 60, 64, 158, 592
- zlabel**, 439, 592
- zlim**, 458