

Index

Note: **Boldface** numbers indicate illustrations or tables.

- % format character, 373
- %c format conversion specifier, 377
- %d format conversion specifier, 375, 377-378
- %f conversion characters, 42
- %s format conversion specifier, 377
- &, && AND operators, 96-97
- ' transpose operator (single quote), 28-29
- + addition command character, 17
- subtraction command character, 17
- .m file extension, 6
- == equivalence (equality) operator, 95, 276
- @ operator to create function handle, 346
- |, || OR operators, 97-98
- ~ NOT operator (tilde), 98
- ~= non-equivalence operator, 95
- ... continuation character (ellipses), 5, 75
- ! special command character (exclamation mark), 15
- & defining keyboard mnemonics (ampersand), 488
- () forming subscripts (parentheses), 46, 50-52, 71
- * multiplication command character, 17
- * skipping selected columns (asterisk), 400
- / division command character, 17
- : operator for shortcut expressions (colon), 28-29, 290-291
- ; operator for array construction (semi-colon), 25-27
- [] operator for array construction (square brackets), 25
- \n escape characters, 42
- ^ exponentiation command character, 17
- ^c interrupt program character, 15
- { } cell constructors (braces), 326

A

- abort Command, 15
- accelerator keys, 487
- Accelerator property, 487
- actual arguments, 201-202

- addpath Command, 17
- advantages of MATLAB, 2-3
- algorithm, 88, 212-213
 - design, 88
 - selection sort, 212-213, **213**
- alpha release, 90
- ans function, 38
- answer cell array, 480
- arguments, 201-202, 218-223
 - actual, 201-202
 - dummy, 201-202
 - input list, 200-201
 - optional, 218-223
 - output list, 200, 201
- Array Editor, **12**, 13
- arrays, 3, 21-25, 25-31, 31-34, 35-37, 45, 46-49,
315-357
 - cell, 323-336
 - classification of, 21-22
 - commands and functions for, **30**, **49**, **320**, **336**, **346**,
349, **354**
 - defined, 3, 21
 - element-by-element basis, 46
 - empty, 26
 - function handles, 345-351
 - initializing variables, 25-31
 - matrices, 21
 - multidimensional, 31-34, **32**
 - operations, 45, 46-49, **49**
 - row order, 25-26
 - size, 21
 - sparse, 315-323
 - structure, 336-345
 - subarrays, 35-73
 - two-dimensional, 31-32
 - vectors, 21, 22

- assignment statements, 25–27, 36–37, 325–326
 - allocating cell arrays, 325–326
 - subarrays, left-hand side in, 36–37
 - variables for, 25–27
 - at permissions string, 367
 - axes, 410–411, 428
 - lines, 411
 - patches, 411
 - position property, 428
 - textstrings, 411
 - axis Command, 117–120
 - axis Command/function, 117
- B**
- bar plots, 296, 298
 - beta release, 90
 - binary file, 383
 - binary I/O functions, 368–373, 383–388
 - example of reading and writing binary data, 370–372
 - fread function, 369–372
 - fwrite function, 368–369, 370–372
 - precision strings, 368–369, 370, 369, 370
 - size argument, 370
 - skip argument, 369
 - box_title, 480
 - braces as cell constructors, 326
 - branches, 85, 102–117
 - defined, 85, 102
 - examples using if constructs, 104–110, 112–113
 - if construct, 102–113
 - nested if constructs, 111–112
 - switch construct, 113–114
 - try/catch construct, 114–115
 - break statements, 167
 - breakpoints, 135–138, 137, 205
 - bugs, 71
 - button groups, 475, 476–479
 - defined, 475
 - SelectionChangeFcn property, 478
 - unibuttongroup function, 478–479, 477–478
 - ButtonDownFcn property, 497
- C**
- callback functions, GUI, 440, 456–457, 497–498
 - Callback property, 487
 - callback subfunction, structure of, 454
 - categorizing characters within a string, 276–278
 - isletter function, 276–278
 - isspace function, 276–278
 - isstrprop function, 276–278, 277
 - cell arrays, 323–336
 - allocating using assignment statements, 325–326
 - braces as cell constructors, 326
 - cell functions, 326, 328, 336
 - cell indexing, 325
 - celldisp function, 327
 - cellplot function, 327
 - cellstr function, 331
 - clear Command, 330
 - content indexing, 325
 - creating, 325–326
 - defined, 323
 - deleting cells, 330
 - extending, 327–330
 - nargout function, 335
 - plot function, 333–334
 - plotline function, 331–334, 334
 - pointer, 323, 324
 - preallocating, 326, 328–330
 - significance of, 332–336
 - strings, of, 331–332
 - types of, 323
 - using data, 330–331
 - varargin input argument, 332–335
 - varargout input argument, 334–335
 - viewing contents, 326–327
 - cell functions, 326, 328, 336
 - cell indexing, 325
 - celldisp function, 327
 - cellplot function, 327
 - cellstr function, 331
 - char variable, 24, 41, 272–274, 281
 - char(x) function, 273
 - character data, displaying, 377
 - checkbox GUI property, 465
 - Checkbox_callback function, 465, 465–466
 - checkboxes, 463–466, 465–466
 - Checked property, 487
 - child objects, 410
 - Children property, 424
 - clc Command, 15
 - clear Command, 11–12, 15, 330
 - clear argument, 392
 - clf Command, 15
 - clock function, 38
 - colon operator, 28–29
 - column major order, 33
 - column vector, 22
 - Command History Window, 6–7, 7
 - Command Window, 4–6, 6
 - command function duality, 118
 - comparing strings, 275–276
 - equality operator, 276
 - strcmp function, 275–276
 - strcmpi function, 275–276
 - strncamp function, 275
 - strncampi function, 275
 - compassplots, 296, 299
 - compiler, MATLAB, 3
 - complex data, 261–272
 - complex functions, 265–269
 - complex numbers, 260
 - complex variables, 264
 - plot function, 271–272
 - plotting, 269–272
 - polar coordinates, 262, 263
 - rectangular coordinates, 262–264, 263
 - relational operators, 264–265
 - complex numbers, 260

- components, 440, 441–442, 459–475
 - concatenating strings, 274
 - strcat function, 274
 - strvcat function, 274
 - conditional breakpoint, 137–138
 - constructs, 91
 - containers, 440, 441, 475–479
 - content indexing, 325
 - context menu, 483, 488–489
 - continue statements, 168
 - contour plots, 305–306, 308
 - conversion characters, 42
 - CreateFcn property, 497
 - CTRL+key combinations, 487
 - CurrentCharacter property, 497
- D**
- data dictionary, 23
 - data files, 43–45
 - load Command, 43–45
 - save Command, 43
 - data hiding, 200
 - Data, 91–102, 200, 223–231, 261–295
 - complex, 261–272
 - hiding, 200
 - integer, 293–295
 - logical, 91–102
 - MATLAB types, 262
 - multidimensional arrays, 290–292
 - sharing, 223–231
 - single, 292–293, 295
 - string functions, 272–290
 - DataName function, 420
 - DataValue function, 420
 - date function, 38
 - deblank function, 273, 280
 - debugging, 71–73, 134–140
 - breakpoints, 135–138, 137, 205
 - bugs, 71
 - conditional breakpoint, 137–138
 - defined, 71
 - logical error, 72
 - M-Lint, 138–139
 - programs in MATLAB, 134–140, 136
 - run-time error, 71
 - symbolic debugger, 73
 - syntax error, 71
 - decimal data, displaying, 375–376
 - decomposition process, 88
 - dec2hex function, 282
 - default_answer cell array, 480
 - default format, 40
 - default properties, 432–434
 - default string, 433–434
 - Default values, 432–433, 433
 - DeleteFcn property, 497
 - delim function, 279
 - demo Command, 14
 - desktop, 4, 5
 - device-independent plotting, MATLAB, 2–3
 - dialog boxes, 479–483, 479
 - answer cell array, 480
 - box_title, 480
 - default_answer cell array, 480
 - directoryname, 482
 - error and warning, 479–480
 - error_string message, 480
 - filename, 481
 - filter_spec parameter, 481
 - input, 480–481, 481
 - line_no parameter, 480
 - modal string, 480
 - modal, 479
 - non-modal string, 480
 - non-modal, 479
 - ok button, 480
 - pathname, 481
 - prompt cell array, 480
 - start_path parameter, 482
 - title parameter, 480, 481, 482
 - types of, 479
 - uigetdir, 482, 483
 - uigetfile, 481–482, 482
 - uisetcolor, 483
 - uisetfile, 481
 - uisetfont, 483
 - warning_string message, 480
 - diary Command, 15
 - directoryname, 482
 - disadvantages of MATLAB, 3
 - disp function, 41
 - docking, Windows to desktop, 10–11
 - double data type, 317
 - double function, 273
 - double variable, 24, 292–293, 294, 295
 - dummy arguments, 201–202
 - dynamic field names, 343–344
- E**
- ease of use, MATLAB, 2
 - edit boxes, 460–462, 461–462
 - Edit Window, 7–8, 9
 - edit GUI property, 460
 - element-by-element performance operations, 46
 - ellipsis, 5
 - empty array, 26
 - end function, 35
 - end statement, 111
 - enhancements, GUI, 497–502
 - ButtonDownFcn property, 497
 - CreateFcn property, 497
 - creating a histogram GUI, 498
 - CurrentCharacter property, 497
 - DeleteFcn property, 497
 - KeyPressFcn property, 497
 - WindowButtonDownFcn callback function, 497
 - WindowButtonMotionFcn callback function, 497
 - WindowButtonUpFcn callback function, 497
 - eps function, 38
 - equivalence (equality) operator, 95, 276

- error dialog box, 479–480, **480**
- error function, 218–219
- error_string message, **480**
- escape characters, 42, **376**
- eval function, 237, 282
- events, GUI, **439**
- exist function, 389–391, **389**
- explode function, 300
- expression statement, 25
- ezplot function, 300–301

- F**
- factory string, 433–434
- false function, 91–92
- fclose function, 368
- feof function, 389, 392
- error function, 389, 392, 393
- feval function, 237, 347
- fgetl function, 382
- fgets function, 383
- fieldnames function, **339**
- fields, 336, 337–345, **337, 341**
 - adding fields, 339–340
 - dynamic field names, 343–344
 - fieldnames function, 339
 - getfield function, 342
 - removing fields, 340
 - setfield function, 342
 - size function, 344
- figure function, 121, 410, 429, 432–433, 440, **457–458**
- figure toolbar, 132–133, **133**
- Figure Window, 8, 10
- figures, 121, 132–133, 410–411, 426–428, 429–431, **431–432, 454, 456**
 - annotating and saving plots, 132–133
 - application data, adding to, 454, 456
 - axes objects, 428
 - axes, 410–411, 428
 - creating multiple Windows, 121
 - default properties, 432–434
 - object positions, 426–428
 - positioning objects within, 429–431
 - printing-related properties, 431–432, **432**
 - types of, 410
 - uicontrol objects, 428
 - Units property, 427–428
 - vector values, 426–427
- file id (fid), 364, 373, 379, 392–393, 399
- filename function, 359, 481
- files, 363–365, 365–368, 368–369, 369–372, 383–388, **388–401, 402, 403**
 - at permissions string, 367
 - binary, 383
 - comparison of formatted and unformatted, **384–388, 384**
 - fclose function, 368
 - file id (fid), 364, 373, 379
 - fopen function, 365–367, **366, 367**
 - formatted, 383
 - fread function, 369–372
 - fwrite function, 368–369, 370–372
 - input/output statements, 364
 - opening and closing, 365–368
 - positioning, 388–401, **402, 403**
 - processing, introduction to, 363–365
 - status functions, 388–401, **402, 403**
 - unformatted, 383
 - wt permissions string, 367
- filter_spec parameter, **481**
- Findobj function, 422–423
- findobj function, 456–457
- findstr function, 278
- floating-point data, displaying, 376
- fopen function, 365–367, **366, 367**
- for loop, 153–170, 341–342
 - break statements, 167
 - continue statements, 168
 - details of operation, 161–163
 - indentation, 161
 - just-in-time (JIT) compiler, 163–167
 - loop index, 154, 161
 - nesting, 168–170
 - preallocating arrays, 161–162
 - structure arrays, 341–342
 - vectorizing arrays, 162–163
- format, 40
- format Command, 40–41, **41**
- format conversion specifiers, 374–377
 - character data, 377
 - decimal (%d), 375
 - decimal data, 375–376
 - escape characters, 376
 - floating-point data, 376
 - format flags, 375
 - format strings, use of, 377–378
 - fprintf function, 373–374, **374, 375, 376, 377, 377–378**
 - structure of, 374
 - understanding, 374–377
- format flags, 375
- format string, 373–374, **374, 399**
- format strings, use of, 377–378
- formatted file, 383
- formatted I/O functions, 373–383, 383–388
 - comparing to binary I/O functions, 383–388
 - fgetl function, 382
 - fgets function, 383
 - format conversion specifiers, 374–377
 - format flags, 375
 - format string, 373–374, **374**
 - format strings, use of, 377–378
 - fprintf function, 373–374, **374, 375, 376, 377–378**
 - fscanf function, 380–382, **381**
 - generating a table of information, 379–380
 - line character array, 382–383
 - sprintf function, 379
- fplot function, 300–301
- fprintf function, 41–43, 42, 281, 282, 359, **373–374, 374, 375, 376, 377–378**
- fread function, 369–372

- frewind function, 392
 - fscanf function, 380–382, 381
 - fseek function, 392
 - ftell function, 389, 392
 - function functions, 236–240, 237
 - function handles, 345–353
 - @ operator, 346
 - creating and using, 346–349
 - defined, 345
 - feval function, 347
 - func2str function, 347–348
 - functions for manipulation, 349
 - nested functions, 350–351
 - parentheses, 347
 - persistent statement, 350
 - plotfunc function, 348
 - significance of, 349–350
 - str2func function, 346
 - functions, 53–54, 199–259, 345–353, 359–407, 421
 - array inputs, 53–54
 - binary I/O, 368–373, 383–388
 - built-in, 53–54
 - common, 55, 75–77
 - comparison of binary and formatted I/O, 383–388
 - defined, 53
 - file id (fid), 364, 373, 379
 - file opening and closing, 365–368
 - file processing, introduction to, 363–365
 - formatted I/O, 373–383, 383–388
 - function function, 236–240
 - function handles, 345–353
 - input/output (I/O), 359–407
 - input/output statements, 364
 - load Command, 361–363, 363
 - manipulating user-defined data, 421
 - MATLAB, 201–207
 - optional results, 53
 - order of evaluation, 245–246
 - preserving data between calls, 231–236
 - save Command, 361–363, 362
 - scope, 240
 - textread function, 361
 - user-defined, 199–259
 - func2str function, 347–348
 - fwrite function, 368–369, 370–372
 - fzero function, 236
- G**
- Gca function, 422
 - gcbf (get callback figure) function, 456, 469
 - gcb0 (get callback object) function, 456
 - gcf function, 422
 - gco function, 422, 424
 - get function, 412–413, 422, 459
 - getappdata function, 420–421
 - getfield function, 342
 - global memory, 223–231
 - global statement, 223
 - graphical image, exporting a plot, 57–59
 - graphical user interface (GUI), 3, 439–507
 - axes, 440, 442
 - callbacks, 440, 497–498
 - components, 440, 441–442, 459–475
 - containers, 440, 441, 475–479
 - creating and displaying, 440–457
 - defined, 3, 439
 - dialog boxes, 479–483, 479
 - enhancements, 497–502
 - error dialog box, 480
 - figure properties, 457–458
 - graphical controls, 440, 441
 - guide function, 443–444, 443, 447, 448–449
 - how it works, 439–440
 - interface components, 459–475
 - menus, 440, 442, 438–494
 - object properties, 457–459
 - properties, 457–459, 477–478, 484, 485
 - pseudocode (pcode), 495–496
 - static elements, 440, 442
 - tips for creating, 494–502
 - tool tips, 494–495
 - toolbars, 440, 442, 496–497
 - uicontrol properties, 458–459
 - unibuttongroup properties, 477–478
 - unipanel properties, 477–478
 - graphics objects, MATLAB, 409
 - Greek and mathematical symbols, 125
 - grid Command, 56
 - guide function, 443–444, 443, 447, 448–449
 - guide window, 445
- H**
- H1 comment line, 202
 - handle graphics, 409–437
 - axes objects, 428
 - changing object properties, 411–418
 - default properties, 423–434
 - default string, 433–434
 - Default values, 432–433, 433
 - defined, 409
 - factory string, 433–434
 - figure objects, 426–428
 - figures, 410–411, 426–428
 - functions for manipulating user-defined data, 421
 - graphic objects, hierarchy of, 410
 - graphics object properties, 434
 - line object, 409
 - listing possible property values, 418–420
 - MATLAB system, 409–411
 - object handles, 411, 422–423
 - position and units, 426–431
 - position property, 426–431
 - positioning objects within a figure, 429–431
 - printer positions, 431–432
 - printing-related figure properties, 432
 - property name, 411
 - remove string, 433–434
 - root object, 410, 432
 - selecting objects with mouse, 423–426
 - set function, 418–420, 422, 432–433
 - text objects, 428
 - uicontrol objects, 428

handle graphics, *continued*
 Units property, 427–428
 user-defined data, 420–421

handle, 409, 411
 handles structure, 463, 465
 Help Browser, 13–14, 13
 help Command, 14
 helpdesk Command, 13
 helpwin Command, 13
 hierarchy, 50–53, 50, 410
 graphic objects, 410
 operations, 50–53, 50
 histogram, 228, 230, 302–303, 302, 498–502, 502
 hnd1 function, 420–421
 hold Command, 120
 host function, 243
 h_fig function, 422
 h_obj function, 422

I

i function, 38
 identity matrices, 29
 if construct, 102–113
 else clause, 103
 elseif clause, 103
 end statement, 111
 nested, 111
 if/else construct, 173–174
 index, loop variable, 154, 161
 Inf function, 38
 Inf statement, 370, 380
 input argument list, 200, 201
 input dialog box, 480–481, 481
 input function, 29–30
 input/output (I/O) functions, 359–407
 binary, 368–373, 383–388
 comparison of binary and formatted, 383–388
 fclose function, 368
 file id (fid), 364
 file opening and closing, 365–368
 file positioning and status functions, 388–401, 402, 403
 file processing, introduction to, 363–365
 filename function, 359
 fopen function, 365–367, 366, 367
 format conversion specifiers, 374–377
 formatted, 373–383, 383–388
 fprintf function, 373–374, 374
 fread function, 369–372
 fwrite function, 368–369, 370–372
 input/output statements, 364
 load Command, 361–363, 363
 precision strings, 368–369, 370, 369, 370
 save Command, 361–363, 362
 size argument, 370
 skip argument, 369
 textread function, 359–361
 inputname function, 218–219
 int2str function, 41, 281
 integer data types, 293–295

double function, 294
 limitations of, 295
 single function, 294
 interface components, GUI, 459–475
 checkbox property, 465
 checkboxes, 463–466, 465–466
 Checkbox_callback function, 465, 465–466
 edit boxes, 460–462, 461–462
 edit property, 460
 gcbf function, 469
 handles structure, 463, 465
 list boxes, 466, 469, 470–471
 listbox property, 469
 max property, 460, 470
 min property, 460, 465, 470
 open string, 469
 popup menus, 466, 468
 pushbutton property, 463
 pushbuttons, 462–463
 radiobutton property, 465
 radiobuttons, 463–466, 467–468
 SelectionMode property, 469
 slider property, 469
 sliders, 469–471, 471
 static text fields, 460
 String property, 460, 462
 temperature conversion program, 472–475
 text property, 460
 TextBox field, 463, 465
 toggle buttons, 463, 464
 togglebutton' property, 463
 ToggleButton_Callback function, 463
 uicontrol function, 460, 462–463, 465, 469
 Value property, 463, 465, 466, 469–470
 ischar function, 273
 isletter function, 276–278
 isspace function, 276–278
 issparse function, 318
 isstrprop function, 276–278, 277

J

j function, 38
 just-in-time (JIT) compiler, 163–167

K

keyboard mnemonics, 487–488
 KeypressFcn property, 497

L

Label property, 487
 Layout Editor, 443, 460, 469, 475
 legend Command, 61
 legend function, 60–63
 legends, 60–63
 lexicographic order, 285
 line character array, 382–383
 line color, 60–63, 60
 line object, 409
 line plots, 303–305, 304
 line style, 60–63, 60

- linear regression, 175
 - lines, 411
 - LineStyle option, 409
 - LineWidth properties, 123, 124
 - line_no parameter, 480
 - list boxes, 466, 469, 470-471
 - listbox GUI property, 469
 - load Command, 43-45, 361-363, 363
 - load-ascii Command, 363
 - logarithmic scales, 63
 - logic operators, 96-99, 97
 - AND, 96-97
 - hierarchy of operations, 99
 - NOT, 98
 - OR, 97-98
 - using numerical data, 98-99
 - logical arrays, 170-175
 - if/else constructs, 173-175
 - masking operations with, 170-173
 - logical data type, 91-102
 - equivalence operator, 95
 - false function, 91-92
 - logic operators, 96-99, 96
 - logical functions, 100-101, 101
 - logical variable, creation, 92
 - non-equivalence operator, 95
 - real function, 92
 - relational operators, 92-94, 92, 94
 - roundoff errors, 95
 - true function, 91-92
 - truth tables, 96, 97
 - logical error, 72
 - logical functions, 100-101, 101
 - loglog function, 63
 - lookfor Command, 14
 - loop index, 154, 161
 - loops, 85, 147-198
 - defined, 85, 147
 - break statements, 167
 - continue statements, 168
 - examples of, 175-190
 - for loop, 153-170
 - just-in-time (JIT) compiler, 163-167
 - logical arrays, 170-175
 - loop index, 154, 161
 - nesting, 168-170
 - vectorization, 163, 170-175
 - while loop, 147-153
 - low-level graphics commands, use of, 415-418
 - lowercase function, 280
 - lowercase string conversion, 280
- M**
- M-files, 5, 15-17, 201-206, 451-454
 - callback subfunction structure, 454
 - calling with arguments, 453-454, 455
 - calling without arguments, 452-453
 - defined, 5
 - examination of, 451-452
 - manual modification of, 454, 456
 - MATLAB functions, 201-206
 - script files, 201
 - search path for, 15-17
 - M-Lint, 138-139
 - Marker properties, 123, 124
 - marker style, 60-63, 60
 - masking operations using logical arrays, 170-173
 - mat2str function, 282
 - MATLAB Editor, 9
 - matrices, 21
 - Matrix Laboratory (MATLAB), 1-20, 21-84, 262, 409-411
 - advantages of, 2-3
 - array operations, 45, 46-49
 - arrays, 21-25, 31-34
 - basics of, 21-84
 - built-in functions, 53-54
 - commands and functions for, 75-77, 140-141, 191, 237, 247-248, 284, 296, 306, 309-311, 320, 330, 336, 346, 349, 354, 405, 435, 504-505
 - data files, 43-45
 - data types, 262
 - debugging, 71-73
 - disadvantages of, 3
 - displaying output data, 40-43
 - environment, 3-17
 - handle graphics system, 409-411
 - hierarchy of graphic objects, 410
 - hierarchy of operations, 50-53, 73, 99
 - introduction to, 1-20
 - plotting, 54-71
 - scalar operations, 45-46
 - special symbols, 75
 - special values, 38-39
 - subarrays, 35-37
 - use as scratch pad, 17
 - variables, 21-25, 25-30
 - matrix operations, 46, 47-49, 49
 - max function, 53
 - max property, 460, 470
 - memory, 33, 223-231, 231-236
 - global memory, 223-231
 - persistent memory, 231-236
 - storing multidimensional arrays, 33
 - Menu Editor, 485, 487, 485-486
 - menus, 440, 442, 483-494
 - accelerator keys, 487
 - context, 483, 488-489, 490-491
 - creating, 487
 - default menu, suppressing, 485-486
 - defined, 483
 - guide Layout Editor, 485
 - keyboard mnemonics, 487-488
 - Label property, 488
 - Menu Editor, 485, 487, 485-486
 - MenuBar property, 486
 - propedit Property Editor, 485
 - set Command, 488-489
 - standard, 483

menus, *continued*

- uicontextmenu objects, 485, 488
- uicontextmenu properties, 485
- uimenu objects, 484–485
- uimenu properties, 484
- mesh plots, 305–306, 307
- min property, 460, 465, 470
- modal string, 480
- mouse, selecting objects with, 423–426
- multidimensional arrays, 31–34, 32, 34, 290–292
 - accessing with one dimension, 33
 - column major order, 33
 - displaying data, use for, 290–292
 - ndims function, 291
 - size function, 291
 - storing in memory, 33
 - two-dimensional, 31–32
- multiple plots, 58–59

N

- n statement, 359, 370, 380, 399
- Nan function, 38
- nargchk function, 218–219
- nargin function, 218
- nargout function, 218, 335
- ndims function, 291
- nested functions, 243–245, 243, 350–351
 - function handles, 350–351
 - persistent statement, 350
 - user-defined functions, 243–245, 243
- nested if constructs, 111–112
- nested loops, 168
- nesting loops, 168–170, 344–345
 - for loops, 168–170
 - structure arrays, 344–345
- [n m] statement, 370, 381
- non-equivalence operator, 95
- non-modal string, 480
- num2str function, 41, 281
- numeric-to-string conversions, 281–282
 - dec2hex function, 282
 - int2str function, 281
 - mat2str function, 282
 - num2str function, 281
 - sprintf function, 282

O

- object handles, 422–423, 423–426
 - finding, 422–423
 - Children property, 424
 - Findobj function, 422–423
 - Gca function, 422
 - gcf function, 422
 - gco function, 422
 - h_fig function, 422
 - h_obj function, 422
 - selecting with mouse, 423–426
 - selection region, 423
 - stacking order, 424
 - waitforbuttonpress function, 424

- object properties, 411–418, 457–459
 - after creation time, 412–415
 - at creation time, 411–412
 - changing, 411–418
 - figure properties, 457–458
 - get function, 412–413
 - Property Editor, 413, 415
 - set function, 412–413
 - uicontrol properties, 458–459
 - using low-level graphics commands, 415–418
 - value pairs, 411–412
- offset string, 393
- ok button, 480
- open string, 469
- operations, 45–53
 - array, 46–47, 49
 - hierarchy of, 50–53
 - matrix, 46, 47–49
 - scalar, 45–46
- optional arguments, 218–223
- order of function evaluation, 245–246
- origin string, 393
 - bof value, 393
 - cof value, 393
 - eof value, 393
- otherwise code block, 113
- output argument list, 200, 201
- output data, displaying, 40–43
 - changing default format, 40–41
 - conversion characters, 42
 - disp function, 41
 - escape characters, 42
 - format Command, 40–41, 41
 - fprintf function, 41–43, 42

P

- panels, 475–478
 - defined, 475
 - grouping related characteristics, 477
 - styles, 476
 - unipanel function, 475–476, 477–478
- parent objects, 410
- parentheses, use of, 46, 50–52, 71, 323, 347
- pass-by-value scheme, 207–217
- patches, 411
- path Command, 17
- Path Tool, 16
- path2rc Command, 17
- pathname, 481
- pcode Command, 495–496
- persistent memory, 231–236
- persistent statement, 231, 350, 447, 448
- pi function, 38
- pie function, 300
- pie plots, 296, 299
- platform independence, 2
- plot Command, 120, 123, 409
- plot function, 54, 271–272, 303, 333–334
- plotfunc function, 348
- plotline function, 331–334, 334

- plots, 295–303, 303–306
 - bar, 296, 298
 - compass, 296, 299
 - contour plots, 305–306, 308
 - explode function, 300
 - histogram, 302–303, 302
 - line, 303–305
 - mesh plots, 305–306, 307
 - pie function, 300
 - pie, 296, 299
 - plot function, 54, 303
 - stair, 296, 297
 - stem, 296, 297
 - surface plots, 305–306, 307
 - three-dimensional functions, 306
 - three-dimensional plots, 303–306
 - two-dimensional functions, 296
 - two-dimensional plots, 295–303
 - plotting, 54, 56–71, 117–134, 208–211, 269–272, 296, 300–301, 306
 - annotating and saving plots, 132–133
 - axis Command, 117–120
 - complex data, 269–272
 - creating multiple figures, 121
 - enhanced control of lines, 123
 - enhanced control of text strings, 123–125
 - explode function, 300
 - ezplot function, 300–301
 - figure function, 121
 - fplot function, 300–301
 - graphical image, 57–59
 - hold Command, 120
 - introduction to, 54
 - legends, 60–63, 62
 - limits, controlling x- and y-axis, 117–120
 - line color, 60–63
 - line style, 60–63
 - logarithmic scales, 63
 - loglog function, 63
 - marker style, 60–63
 - multiple plots on same axis, 120
 - multiple plots, 58–59
 - plot function, 63, 271–272
 - polar plots, 125
 - printing, 57
 - problem-solving examples, 64–71
 - rectangular-to-polar conversion, 208–211
 - semilogx function, 63
 - semilogy function, 63
 - simple xz plots, 56–57
 - stream modifiers, 123
 - subplots, 121–123, 122
 - three-dimensional functions, 306
 - two-dimensional functions, 296, 300–301
 - pointer, 323, 324
 - polar function, 125
 - polar plots, 125
 - popup menus, 466, 468
 - pos values in legend Command, 61
 - Position property, 426–431
 - axes objects, 428
 - figure objects, 426–428
 - positioning objects within a figure, 429–431
 - printer use, 431–432
 - text objects, 428
 - uicontrol objects, 428
 - preallocating arrays, 161–162, 362
 - precision strings, 368–369, 370, 369, 370
 - predefined functions, MATLAB, 2
 - preserving data between calls to a function, 231–236
 - primary function, 241
 - print Command, 57–58
 - print options, 58
 - printer positions, 431–432
 - printing a plot, 57
 - printing-related figure properties, 432
 - private functions, 242–243
 - problem-solving examples, 64–71
 - program design, 85–146
 - branches, 102–117
 - debugging programs, 134–139
 - logical data type, 91–102
 - plotting features, 117–134
 - pseudocode, 91
 - top-down techniques, 85–91
 - programs in MATLAB, 134–140
 - prompt cell array, 480
 - propedit Property Editor, GUI, 485
 - properties, graphic objects, 409
 - Property Editor, 413, 415, 485
 - Property Inspector, 444, 445, 459
 - property name, 411
 - Propertyname value pairs, 411–412
 - pseudocode, 88, 91, 495–496
 - pushbutton property, 463
 - pushbuttons, 462–463
- Q**
- quadratic equation, program to solve, 105–108
- R**
- radiobutton property, 465
 - radiobuttons, 463–466, 467–468
 - rand function, 230–231
 - randn function, 230–231
 - random number generator, 224–231
 - real function, 92
 - relational operators, 92–94, 92, 94
 - remainder function, 279
 - remove string, 433–434
 - repl function, 279
 - reset function, 232–233
 - reusable code, 200
 - rmpath Command, 17
 - root object, 410, 432
 - roundoff errors, 95
 - row order, 25–26
 - row vector, 22
 - run-time error, 71
 - running averages, 231–236

S

- save Command, 43, 361–363, 362
 - save-ascii Command, 363
 - scalar operations, 45–46
 - arithmetic between, 46
 - assignment operator, 46
 - scalar value, assignment to subarray, 37
 - scope of a function, 240
 - script files, 6, 201
 - search path, 15–17
 - searching and replacing characters within a string, 278–279
 - delim function, 279
 - findstr function, 278
 - remainder function, 279
 - repl function, 279
 - srch function, 279
 - str function, 279
 - string function, 279
 - strmatch function, 278–279
 - strep function, 279
 - strtok function, 279
 - selection region, 423 token function, 279
 - selection sort, 212, 213
 - SelectionChangeFcn property, 478
 - SelectionType property, 469
 - semilogx function, 63
 - semilogy function, 63
 - Separator property, 487
 - set Command, 488–489
 - set function, 412–413, 418–420, 422, 432–433, 459
 - setappdata function, 420–421
 - setfield function, 342
 - shape of array values, 36
 - signed integers, 293
 - simple xz plots, 56–57
 - single data type, 292–293, 295
 - double variable, 292–293
 - limitations of, 295
 - single variable, 292–293, 294, 295
 - size argument, 370, 380–381
 - size function, 29, 291, 344
 - skip argument, 369
 - slider GUI property, 469
 - sliders, 469–471, 471
 - sorting data, 212–217
 - sparse arrays, 315–323
 - defined, 315–316
 - double data type, 317
 - generating sparse matrices, 319
 - issparse function, 318
 - sparse attribute, 317–323
 - sparse matrix functions, 318, 320
 - sparse matrix, 316
 - whos Command, 318
 - working with sparse matrices, 319–320
 - sparse attribute, 317–323
 - sparse function, 318, 320
 - sparse matrices, 316–323
 - defined, 316
 - double data type, 317
 - functions, 320
 - generating, 319
 - issparse function, 318
 - solving simultaneous equations with, 320–323
 - sparse attribute, 317–323
 - sparse function, 318, 320
 - speye function, 319
 - sprand function, 319
 - sprandn function, 319
 - whos Command, 318
 - working with, 319–320
- special symbols, 19, 75
 - special values, predefined, 38
 - speye function, 319
 - sprand function, 319
 - sprandn function, 319
 - sprintf function, 282, 379
 - sqrt function, 54
 - srch function, 279
 - stacking order, 424
 - stair plots, 296, 297
 - standard menu, 483
 - Start Button, 7, 8
 - start_path parameter, 482
 - static text fields, 460
 - status functions, file positioning, 388–401
 - clear argument, 392
 - exist function, 389–391, 389
 - feof function, 389, 392
 - error function, 389, 392, 393
 - frewind function, 392
 - fseek function, 392
 - ftell function, 389, 392
 - offset string, 393
 - origin string, 393
 - textscan function, 398–401
 - uiimport function, 401, 402, 403
 - stem plots, 296, 297
 - str function, 279
 - str2func function, 346
 - strcamp function, 275–276
 - strcampi function, 275–276
 - strcat function, 274
 - stream modifiers, 123
 - string Command, 11
 - string conversion functions, 273
 - char(x) function, 273
 - double function, 273
 - string function, 279, 446
 - string functions, 272–290, 331–332
 - categorizing characters, 276–278
 - cell arrays of, 331–332
 - char variable, 272–274
 - common MATLAB, 284
 - comparing for equality, 275–276
 - comparing individual characters, 276
 - comparing, 275–276
 - comparison function, example of, 285–288
 - concatenating, 274

- conversion, 273
 - ischar function, 273
 - lowercase conversion, 280
 - numeric-to-string conversions, 281–282
 - searching and replacing characters, 278–279
 - string-to-numeric conversions, 282–283
 - trimming whitespace, 280
 - two-dimensional character arrays, 273–274
 - uppercase conversion, 280
 - whos Command, 272
 - String GUI property, 460, 462
 - string-to-numeric conversions, 282–283
 - eval function, 282–283
 - sscanf function, 282–283
 - str2double function, 282–283
 - strmatch function, 278–279
 - strncamp function, 275
 - strncampi function, 275
 - strongly typed language, 25
 - strep function, 279
 - strtok function, 279
 - strtrim function, 280
 - struct function, 336, 339
 - structure arrays, 336–345, 346
 - adding fields, 339–340
 - building, 337–339
 - creating, 336–339
 - defined, 336
 - dynamic field names, 343–344
 - fieldnames function, 339
 - fields, 336, 337–345, 337, 341
 - for loop, 341–342
 - getfield function, 342
 - nesting, 344–345
 - removing fields, 340
 - setfield function, 342
 - size function, 344
 - struct function, 336, 339
 - structure functions, 346
 - structure, defined, 336
 - using data, 340–342
 - structured program, 91
 - sub-tasks, 199
 - subarrays, 35–37
 - assigning a scalar, 37
 - assignment statement, use of on left-hand side, 36–37
 - end function, 35
 - shape, 36
 - subfunctions, 241–242, 241
 - subplot Command, 121–123
 - subplots, 121–123, 122
 - surface plots, 305–306, 307
 - switch construct, 113–114
 - symbolic debugger, 73
 - syntax error, 71
- T**
- Tag property, 487
 - test execution, 71
 - text objects, 428
 - text GUI property, 460
 - TextBox field, 463, 465
 - textread function, 359–361, 400–401
 - filename function, 359
 - format string, 373–374, 374
 - n statement, 359
 - relation to textscan, 400–401
 - textscan function, 398–401
 - textstrings, 411
 - three-dimensional plots, 303–306, 307–308
 - contour plots, 305–306, 308
 - functions, 306
 - line plot, 303–305, 304
 - mesh plots, 305–306, 307
 - surface plots, 305–306, 307
 - title function, 56
 - title parameter, 480, 481, 482
 - toggle buttons, 463, 464
 - togglebutton GUI property, 463
 - ToggleButton_Callback function, 463
 - token function, 279
 - tool tips, GUI, 494–495
 - ToolBar property, 496–497
 - toolbars, 440, 442, 496–497
 - top-down design techniques, 85–91, 199
 - algorithm, 88
 - alpha release, 90
 - beta release, 90
 - defined, 86
 - limitations to, 199
 - process steps, 86–88, 87
 - pseudocode, 88
 - unit testing, 89
 - transpose operator, 28–29
 - trimming whitespace from strings, 280
 - deblank function, 280
 - true function, 91–92
 - strtrim function, 280
 - truth tables, 96, 97
 - try/catch construct, 114–115
 - two-dimensional arrays, 31–32, 32, 273–274
 - char variable, 273–274
 - character, 273–274
 - deblank function, 273
 - introduction to, 31–32, 32
 - two-dimensional plots, 295–303, 304
 - line plots, 303–305, 304
 - pie function, 300
 - pie plots, 296, 299
 - stair plots, 296, 297
 - stem plots, 296, 297
 - Type property, 422
- U**
- uibuttongroups object, 411
 - uicontextmenu function, 411, 440
 - uicontextmenu GUI objects, 485
 - uicontextmenu properties, 485
 - uicontrol function, 411, 428, 440, 457, 458–459, 460, 462–463, 465, 469

- uiicontrol GUI objects, 494, 497
 - uigetdir dialog box, 482, 483
 - uigetfile dialog box, 481–482, 482
 - uiimport function, 401, 402, 403
 - uimenu function, 411, 440
 - uimenu GUI objects, 484–485
 - uimenu properties, 484
 - uipanel object, 411
 - uisetcolor dialog box, 483
 - uisetfile dialog box, 481
 - uisetfont dialog box, 483
 - uitoolbar function, 411, 440, 497
 - undocking, Windows to desktop, 10–11
 - unformatted file, 383
 - unibuttongroup function, 478–479, 477–478
 - unipanel function, 475–476, 477–478
 - unit testing, 89, 199
 - Units property, 427–428
 - unsigned integers, 293
 - uppercase function, 280
 - uppercase string conversion, 280
 - user-defined data, 421–421
 - DataName function, 420
 - DataValue function, 420
 - functions for manipulating, 421
 - getappdata function, 420–421
 - hdl function, 420–421
 - setappdata function, 420–421
 - user-defined functions, 199–259
 - benefits of, 199–200
 - data hiding, 200
 - function functions, 236–240, 237
 - global memory, 223–231
 - H1 comment line, 202
 - MATLAB functions, 201–207
 - nested functions, 243–245, 243
 - optional arguments, 218–223
 - order of evaluation, 245–246
 - pass-by-value scheme, 207–217
 - preserving data between calls, 231–236
 - private functions, 242–243
 - random number generator, 224–231
 - rectangular-to-polar plot conversion, 208–211
 - selection sort, 212
 - sharing data, 223–231
 - sorting data, 212–213
 - subfunctions, 241–242, 241
 - variable passing, 207–217
- V**
- value pairs, 411–412
 - Value property, 463, 465, 466, 469–470
 - var statement, 25
 - varargin input argument, 332–335
 - varargout input argument, 334–335
 - variable passing, 207–217
 - variables, 22–25, 25–31
 - assignment statements, 25–27
 - built-in functions, 29
 - defined, 22
 - functions for initializing variables, 30
 - keyboard input, 29–30
 - row order, 25
 - shortcut expressions, 28–29
 - vector, 21, 22
 - vectorization, 163, 170–175
 - vectorizing arrays, 162–163
- W**
- waitforbuttonpress function, 424
 - warning dialog box, 479–480
 - warning function, 218–219
 - warning_string message, 480
 - weakly typed language, 25
 - which Command, 16–17
 - while loop, 147–153
 - whos Command, 11, 272, 318, 362
 - WindowButtonDownFcn callback function, 497
 - WindowButtonMotionFcn callback function, 497
 - WindowButtonUpFcn callback function, 497
 - Windows for MATLAB environment, 4–6, 6–7, 7–8, 9, 8, 10–11
 - Command, 4–6, 6
 - Command history, 6–7, 7
 - docking, 10–11
 - Edit, 7–8, 9
 - Figure, 8, 10
 - undocking, 10–11
 - workspace, 11–12
 - Workspace Browser, 12–13, 12
 - wt permissions string, 367
- X**
- xlabel function, 56
- Y**
- ylabel function, 56
- Z**
- zeros function, 29