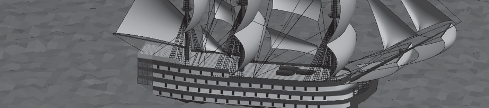
**Index**

# Symbols



./ (array division) 54, 74

.^ (array exponentiation) 54, 74

.\* (array multiplication) 51, 74

= (assignment operator) 21

\ (backslash) 126, 271

: (colon operator) 54, 56, 74

. (dot) 154, 160

. . . (ellipses) 35, 136, 149 [] (empty vector) 50

= = (equal to) 54, 74

> (greater than) 54, 74

> = (greater than or equal to) 54, 74

< (less than) 54, 74

< = (less than or equal to) 54, 74

2 (minus sign) 52

~= (not equal to) 54, 74

2-D plots/plotting 234–239

enhancement tools 237

parametric 238–239

simple plots 235–237

3-D plots/plotting 239–243

linear 239–241

parametric plots 241–242

& (element-wise AND) 53

| (element-wise OR) 53

/ (matrix division). *see* matrix division (/)

^ (matrix exponentiation) 271, 286

\* (matrix multiplication) 51, 268–270,

272–273, 286

% (percent sign) 33, 126

; (semicolon) 28, 62

&& (short-circuit AND) 53, 87

|| (short-circuit OR) 53, 87

~ (unary not) 54, 87, 100

\_ (underscore character) 22

A

A\* algorithm 411–413

code for 412

A/D (analog-to-digital) device 317 abstraction 18–19, 46, 106, 268

acosd() function 193

activation stack 186–187

actual parameters 109

adjacency matrix 399, 406

creation of 400

algorithms 19

A\* 411–413

Breadth-First Search 407–408

bubble sort 373–375, 383

complex, analyzing 370–371

Dijkstra’s 408–411

insertion sort 371–373, 382–383

measuring cost of 368–371

merge sort 377–379, 383

performance analysis of 380–382

Prim’s 404–406

quick sort 375–377, 383

radix sort 379–380, 383

for sorting data 371–380

all() function 87, 100

alpha() function 255, 261

ALU. *see* Arithmetic and Logic Unit (ALU)

American Standard Code for Information

Interchange (ASCII) 122, 129

ampersand (&) 53

analog-to-digital (A/D) device 317 AND

element-wise (&) 53

short-circuit (&&) 53, 54, 87

any() function 87, 100

API. *see* Application Programmer Interface (API)

Application Programmer Interface (API) 108

Arithmetic and Logic Unit (ALU) 5

arithmetic operations 213

with arrays 64

on character strings 125

with vectors 51–52

array division (./) 54

array exponentiation (.^) 54 array multiplication (.\*) 51 arrays 60–71

arithmetic operations with 64

cell (*see* cell arrays)

of character strings 131–132

concatenation 66–67

creating 62

elements of 61, 62–64

inserting data into 214

library functions with 65–66

linear 46

linearized 67–71

logical operations with 64–65

matrices *vs.,* 60

operations 64–71

properties of 61

reshaping 67

slicing of 67

structure 150–156 (*see also*

structure arrays)

transpose of 61

ASCII. *see* American Standard Code for Information Interchange (ASCII)

assignment operator ( = ) 21

.au files 317

auread() function 317, 334

auwrite() function 334 auxiliary (local) functions 111 axis() function 232, 260

**I–1**

**I–2** Index

# B

|  |  |  |
| --- | --- | --- |
|  | accessing 143–145 |  |
| Babbage, Charles 3  back dividing 271, 273, 286 | conversion to string  creating 142–143 | 394 |
| backslash (\) 126, 271 | extracting/sorting | 382 |
| backward difference | inserting data into 214  processing 145–146 | |

approximation 356

bar() function 239, 260

bar3() function 242, 261

barh() function 239, 260

barh3() function 243, 261 Basic Input/Output System

(BIOS) 7, 9

before() function 215

behavioral abstraction 268

BFS. *see* Breadth-First Search (BFS)

Big O algebra 368–371

O(1) (independent of N) 369

O(logN) (binary search)

369–370

O(2N) (exponential growth) 370

O(N) (linear with N) 369 O(N2) (proportional to N2) 370

binary files 168

binary search (O(logN)) 369–370 BIOS. *see* Basic Input/Output System

(BIOS)

bits 6

black box view, functions 106–107 bodies of rotation 250–255

continuous functions,

rotating 251–253

discrete functions, rotating

253–255

boolean value 50, 82, 84, 86–87, 100 Breadth-First Search (BFS) 407–408 break points 36

break statement 94, 96, 97, 100

bubble sort 373–375, 383

building (operation) 213, 216

# C

C (programming language) 12 cache memory 7

CAD. *see* computer-aided design (CAD)

case keyword 88, 89, 100

casting 122–123

catch keyword 192, 193, 194, 207

CAToString() function 394

ceil() function 55, 74

cell arrays 142–146

using 145 central difference

approximation 357

Central Processing Unit (CPU) 5, 7, 186

char() function 123, 125, 132, 136

character generators 121

character mapping 122

character strings 121–135

|  |  |  |
| --- | --- | --- |
| arithmetic operation on | 125 | of character strings 124 |
| arrays of 131–132 |  | of sounds 318–320 |
| casting 122–123  comparison of 129–131  concatenation of 124 |  | of vectors 55  conditional execution 82–83 |

conversion from numbers to 125–127

conversion to numbers 127–129 and delimiter 123

example using 132–135 format control strings 126 logical operation on 125 mapping 122

MATLAB implementation 123–125

as numerical values 122 operations 129–131

slicing of 124

and token 123

class() function 145, 160

classes 24

clc command 26, 33

clear command 33

clf command 232, 260

close all command 232, 260

code blocks 81, 82

coef() function 347–348 colon operator (:) 56

color mapped images 294 color masking 296–301

colormap() function 232, 260

Colossus 3–4

column vector 61

Command History window 26–27 Command window 25–26, 92, 107,

112, 122

comments 33

compile-time errors 12

compilers 11–12

compound surfaces, assembling 256 computer

hardware (*see* hardware,

computer)

internal details 6

internal organization of 5

memory (*see* memory, computer)

software (*see* software, computer)

computer-aided design (CAD) 10 computer architectures, history

of 3–5

computer languages 2–3, 10–11 concatenation

of arrays 66–67

continue statement 96, 100 continuous function, rotating

251–253

contour() function 255, 261

Control Unit 5

CPU. *see* Central Processing Unit (CPU)

cross() function 60, 74

csvread() function 170, 173, 177, 179

csvwrite() function 170, 179 cubic spline interpolation 343–344 cumsum() function 353, 354, 361

cumtrapz() function 354, 361 Current Directory window 30–31,

34–35, 107, 108

curve fitting 345–351

example of 349–351

linear regression 345–347

polynomial regression 347–349

cycles, graphs 396

cylinder, construction of 248–249

cylinder() function 107–109,

255, 261

# D

D/A (digital-to-analog) device 317 data abstraction 19, 46, 268

data bus 6

data collection. *see also* problem- solving

building 216

filtering 217–218

Index **I–3**

inserting data into 213–215 mapping 216–217

searching 219–220

sorting 220

summarizing 218–219

traversing 215–216

data typing 22–24

deal() function 143–145, 160

|  |  |  |
| --- | --- | --- |
| debugging 36 |  | endless recursion 188 |
| del2() function | 248 | engineering applications |

delimited text files 169, 172–173

delimiter 123, 168

dequeue() function 390, 391 derivative, of function 356 design templates 83–84

for functions in MATLAB, 107

for if statement 84

for loop 91

for switch statement 88

for while loop 94–95

diag() function 62, 74

diagonal array 61

diff() function 357, 361 difference engine, Babbage 3 differentiation 356–357

digital-to-analog (D/A) device 317 Dijkstra’s algorithm 408–411

code for 410

directional edges 396

discrete functions, rotating 253–255 disp() function 58, 74, 129, 136 division

matrix 271, 273–274

dlmread() function 170, 172–173, 179

dlmwrite() function 170, 173, 179

documentation section 107

dot (.) notation 154, 160

dot operator 64

double() function 123, 136

drivers 7

# E

edges, graphs 396

Editor window 32–33, 36

element-wise AND (&) 53, 54

element-wise OR (|) 53, 54 elements

arrays 61, 62–64

vectors 47

ellipses (. . .) 35, 136, 149

ellipsoid() function 255, 261

else keyword 83, 100

elseif keyword 84, 100 empty vector ([]) 50 encapsulation 106, 111–112

end keyword 193

end statement 83, 84, 89, 92, 100

ceramic composition 283–285

detecting edges 306–309

electrical circuit analysis 285–286

encryption 132–135

forces and moments 58–60

geographic data,

visualizing 256–259

geopolitical data,

processing 221–226

graphs 415

liquid levels, computation of

97–99

music synthesizer 332–334

physical structure,

assembling 156–160

robot arm motion 202–206

soil volume, computation of 71–73

solid object measurement 113–114

sorting 384–386

spacecraft launch 36–39

spreadsheet data 177–179

synthesizer notes, shaping 359–360

enqueue() function 390, 391, 393 equal to ( = = ) 54, 74

error() function 193, 207

Excel spreadsheets 170–172

exceptions 190–194

generic implementation for

191–192

historical approach 191

MATLAB implementation

193–194

execution errors 12

exponential growth (O(2N)) 370 extrapolation 344–345

eye() function 270, 286

# F

fact() function 195

false values 50, 74, 83, 84, 86, 100 Fast Fourier Transform (FFT)

324–328

implementation 326–327

overview 325–326

simple spectral analysis using 327–328

fclose() function 174, 180

FFT. *see* Fast Fourier Transform (FFT)

fft() function 327, 334

fgetl() function 174, 180

fgets() function 174, 180

fib() function 199

Fibonacci series 198–199

.field operator 152

fieldnames() function 147, 153, 154

figure() function 232, 260

Figure window 31–32 files

binary 168

delimited text 169

opening/closing 174

reading/writing 170

text (*see* text files)

fill() function 239, 260

filtering (operation) 213, 217–218

find() function 68, 74

fix() function 55, 74

floor() function 55, 74

flowcharts 83

folding (operation) 213

fopen() function 174, 180

for loop 90–94, 100 breaking out of 94 example of 92

indexing implementation using 93–94

MATLAB implementation 91–92

structure of 91

template 91

while loop *vs.,* 90

formal parameters 109 format control strings 126 forward difference

approximation 356

fprintf() function 92, 129, 130,

136, 176, 180

frame, stack 187

frequency, sound 322–324 function name section 107 functional programming 20 function(s)

acosd() 193

all() 87, 100

alpha() 255

any() 87, 100

auread() 317, 334

**I–4** Index

|  |  |  |
| --- | --- | --- |
| function(s) (*continued*)  axis() 232, 260  bar() 239, 260  bar3() 242  barh() 239, 260  barh3() 243  before() 215  black box view of 106–107 | hold off() 233, 260  hold on() 233, 261  ifft() 327, 334  image() 295, 310  imread() 295, 310  imshow() 295, 310  imwrite() 295, 310  input() 89, 90, 96, 112, 127–128, | polyval() 348, 361  rand() 47, 62  randn() 47  read() 173  readStruct() 179  reshape() 67, 279, 286  rmfield() 147, 153, 155  rot90() 310 |
| CAToString() 394 | 136, 192 | round() 55 |
| ceil() 55 | instances 187 | save() 180 |
| char() 123, 125, 132, 136 | integral of 351 | semilogx() 237, 261 |
| class() 145, 160 | interp1() 341, 344, 361 | semilogy() 237, 261 |
| coef() 347–348 | interp2() 343, 361 | setfield() 155, 161 |
| colormap() 232, 260 | interp3() 343, 361 | shading() 233, 261 |
| contour() 255 | int2str() 125, 136 | size() 48, 58, 61, 146 |
| cross() 60, 74 | inv() 271, 286 | sort() 155, 161, 382 |
| csvread() 170, 173, 177, 179 | isa() 146, 161 | sound() 318, 335 |
| csvwrite() 170, 179 | is\_before() 391–392, 393 | sphere() 255 |
| cumsum() 353, 354, 361 | iscell() 146, 161 | spline() 344, 361 |
| cumtrapz() 354, 361 | ischar() 125, 136, 146, 161 | sprintf() 126, 129, 130, |
| cylinder() 107–109, 255 | isempty() 390 | 136, 348 |
| deal() 143–145, 160 | isfield() 155, 161 | sscanf() 127, 129, 136 |
| defined 106, 107–108 | islogical() 146, 161 | strcmp() 130, 131, 136 |
| del2() 248 | isnumeric() 146, 161 | strcmpi() 131, 136 |
| dequeue() 390, 391 | isPal() 197 | str2num() 127, 128, 129, 136 |
| derivative of 356 | isspace() 125, 136 | strtok() 129, 180 |
| diag() 62 | isstruct() 146, 161 | struct() 149, 150, 161 |
| diff() 357, 361 | it() 372, 376 | subplot() 233–234, 261 |
| disp() 58, 129, 136 | largest() 145 | sum() 55, 66, 155 |
| dlmread() 170, 172–173, 179 | lasterror() 193, 207 | surf() 243, 244 |
| dlmwrite() 170, 173, 179 | legend() 233, 261 | surfc() 247, 255 |
| double() 123, 136 | length() 48, 58, 61 | surfz() 255 |
| ellipsoid() 255 | lightangle() 248 | text() 233, 261 |
| enqueue() 390, 391, 393 | linspace() 47, 74, 310 | textscan() 175, 180 |
| error() 193, 207 | load() 180 | title() 233, 261 |
| eye() 270, 286 | loglog() 237, 261 | toString() 394 |
| fact() 195 | magic() 62, 75 | tril() 310 |
| fclose() 174 | MATLAB implementation (*see* | uint8/16() 123, 136, 310 |
| fft() 327, 334 | functions, in MATLAB) | view() 247, 261 |
| fgetl() 174, 180 | max() 66, 92 | waterfall() 255 |
| fgets() 174, 180 | mean() 55, 66 | wavread() 317, 335 |
| fib() 199 | mesh() 243 | wavwrite() 335 |
| fieldnames() 147, 153, 154 | meshc() 255 | xlabel() 233, 261 |
| figure() 232, 260 | meshgrid() 243, 245, 255 | xlsread() 170, 177, 180 |
| fill() 239, 260 | meshz() 255 | xlswrite() 170, 172, 180 |
| find() 68 | min() 66 | ylabel() 233, 261 |
| fix() 55 | nargin() 110 | zeros() 47, 62 |
| floor() 55 | nargout() 110 | zeros of 199–202 |
| fopen() 174, 180 | num2str() 125, 136 | zlabel() 233, 261 |
| fprintf() 92, 129, 130, 136, | ones() 47, 62, 305 | functions, in MATLAB, 46, 107–114 |
| 176, 180 | peek() 390 | auxiliary (local) 111 |
| getfield() 155 | pie() 239, 261 | calling 109 |
| gplot() 405 | pie3() 243 | defined 107–108 |
| grAdjacency() 399 | plot() 232, 235, 239, 261 | encapsulation in 111–112 |
| grid off() 232, 260 | plot3() 239 | and global variables 112–113 |
| grid on() 232, 260 | plotyy() 237 | returning multiple results |
| gt() 373, 376 | polar() 239, 261 | from 110–111 |
| hist() 239, 260 | polyfit() 347–348, 361, 384, 385 | storing/using 109 |

Index **I–5**

structures 148–150

|  |  |  |
| --- | --- | --- |
| template of 107 | | I/O. *see* Input/Output (I/O) |
| G  Gaussian Elimination | 271 | identity matrix 270  if statements 83–88, 100, 131  example 85 |

generations, of computer language 10–11

getfield() function 155

global keyword 112, 207

Global Scope 112

global variables 112–113, 115

gplot() function 405

grAdjacency() function 399

graphs 396–404

A\* algorithm 411–413

Breadth-First Search 407–408

building 398–401

creating 31, 32

cycles 396

defined 389

Dijkstra’s algorithm 408–411

examples 396–397, 415

minimum spanning trees of (*see*

minimum spanning trees (MSTs))

nodes 389

paths on 396, 406–414

processing 397–398

I

in logical expressions 86–87 MATLAB implementation 84–86

script with 86

short-circuit evaluation 87–88

template for 84

ifft() function 327, 334

image() function 295, 310

images 291–309

color mapped 294

color masking with 296–301

displaying 295

format of 294–295

gray scale 293

kaleidoscope, creation of 301–303

nature of 292

operation on 295–306

reading 295

resolution of 292

stretching/shrinking 295–296

on surface 303–306

true color 293

types 293–295

writing 295

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| searching 403–404  traversal 401–403 |  | imread() function | 295, 310 | L |
| weighted 396, 398 |  | imshow() function | 295, 310 | largest() function 145 |
| gray scale images 293 |  | imwrite() function | 295, 310 | lasterror() function 193, 207 |
| greater than (>) 54, 74 |  | in-line coding 195 |  | least squares technique 346 |
| greater than or equal to (> = ) | 54, 74 | inner dimensions | 269 | legalist approach 195 |
| grid off() function 232, 260 | | input() function | 89, 90, 96, 100, | legend() function 233, 261 |

interpreted code 13

int2str() function 125, 136

inv() function 271, 273, 274, 286

isa() function 146, 161

is\_before() function 391–392, 393

iscell() function 146, 161

ischar() function 125, 136, 146, 161

isempty() function 390

isfield() function 155, 161

islogical() function 146, 161

isnumeric() function 146, 161

isPal() function 197

isspace() function 125, 136

isstruct() function 146, 161

it() function 372, 376

iteration 90

# J

Joint Photographic Experts Group (JPEG) 294, 301

# K

kaleidoscope, creation of 301–303

grid on() function 232, 260

gt() function 373, 376

# H

hardware, computer 5–6 interaction with software 8

hardwiring 6

heap 8

help command 108, 115

helper functions 111

heterogeneous collections 142 high-level I/O functions 169–173

with delimited text files 172–173

with Excel spreadsheets 170–172

exploration 169–170

hist() function 239, 260

hold off() function 233, 260

hold on() function 233, 261

homogeneous collections 46

112, 127–128, 136, 192

Input/Output (I/O) 6, 168–179

devices 5, 7

high-level 169–173 (*see also* high-

level I/O functions)

lower-level 174–177 (*see also*

lower-level I/O functions)

and MATLAB workspace 168–169

inserting data, in collection 213–215 template for 215

insertion sort 371–373, 382–383 integral, of function 351 integration 351–355

interp1() function 341, 344, 361

interp2() function 343, 361

interp3() function 343, 361

interpolation 340–345

cubic spline 343–344

extrapolation 344–345

linear 340–343

length() function 48, 58, 61, 74

less than (<) 54, 74

less than or equal to (< = ) 54, 74 library functions

with arrays 65–66

with vectors 54, 55

lightangle() function 248, 261

linear arrays 46

linear equations, simultaneous 281–283

linear interpolation 340–343

linear matrices 47

linear regression 345–347

linearized array 67–71 line(s)

intersecting 282–283

rotating 275–276

linker 12

linspace() function 47, 74, 310

**I–6** Index

load() function 180

loader 12

Local Scope 112

logic errors 12, 23

logical expressions 86–87

logical indexing 50 logical operations

with arrays 64–65

on character strings 125

with vectors 52–54

logical value 50

loglog() function 237, 261 loop-and-a-half iteration style

96–97

lower-level I/O functions 174–177 opening/closing files 174

# M

magic() function 62, 75 mapping

character 122

operation 213, 216–217

mass memory 7 MATLAB

advantages 18

components of 18

and data manipulation 20–24

introduction to 13–14, 17–18

and problem-solving 14–15

programming concepts 14

starting/stopping 20–21

user interface 24–33 (*see also* user

interface)

matrix(-ces) 267–286

adjacency 399, 400, 406

arrays *vs.,* 60

examples using 283–286

identity 270

implementation 271–274

linear 47

operations on 268–274

rotating coordinates 274–281

sparse 399, 400

matrix division (/) 271, 273–274, 286 for solving simultaneous linear

equations 281–283

matrix exponentiation (^) 271, 286

matrix multiplication (\*) 51, 268–270,

272–273, 286

for 2-D rotation 274–278

for 3-D rotation 278–281

max() function 66, 75, 92

mean() function 55, 66, 75

mechanical memory 6–7

memory, computer 6–8

layout 8

Mercator projection 303

merge sort 377–379, 383

mesh() function 243, 261

meshc() function 255, 262

meshgrid() function 243, 245,

255, 262

meshz() function 255, 262

min() function 66, 75 minimum spanning trees

(MSTs) 404–406

minus, unary (2) 52 multiplication

array 51

matrix. *see* matrix multiplication (\*)

music synthesizer 332–334

musical sounds 321–324

about 321

changing frequency of 322–324

# N

NaN keyword 344, 361, 392

nargin() function 110, 115

nargout() function 110, 115

Newton’s method 202

nodes, graphs 389 not equal to (~=) 54 numbers

conversion, to strings 125–127

conversion from strings to 127–129

numerical indexing 49–50

numerical methods 339–360

analytical operations 357

curve fitting 345–351 (*see also*

curve fitting)

differentiation 356–357

example using 359–360

implementation 357–358

integration 351–355

interpolation 340–345 (*see also*

interpolation)

numerical values 122

num2str() function 125, 136

# O

object code 12

object-oriented programming (OOP) 20

objects 24

ones() function 47, 62, 75, 305 OOP. *see* object-oriented

programming (OOP)

operating systems (OS) 7–8, 9 operation(s)

analytical 357

on arrays 64–71

character string 129–131

frequency domain 328–332

on graphs 397–398

on queues 390

summary of 212–220

on vectors 51–58

operators dot 64

.field 152

logical 53

precedence 54

OR

element-wise (|) 53, 54

short-circuit (||) 53, 54, 87

OS. *see* operating systems (OS) otherwise keyword 88, 89, 100

# P

page buffer 7

palindromes, determination 197–198

parabolic dish 245–247

paradigms, programming 20 parameters

cell arrays of 145

formal *vs.* actual 109

value 126

variable numbers of 109–110

parameters section 107 parametric plots

2-D 238–239

3-D 241–242

passing by reference 109 passing by value 109 paths, on graphs 396

A\* algorithm 411–413

Breadth-First Search 407–408

Dijkstra’s algorithm 408–411

searching 406–414

pause() function 323

peek() function 390

percent sign (%) 33, 126

pie() function 239, 261

pie3() function 243, 262

pixels 292

plaid surface 243

playback 316–317

plot() function 232, 235, 239, 261

plot3() function 239, 262

plots (plotting) 231–259

Index **I–7**

2-D 234–239 (*see also* 2-D plots/ plotting)

3-D 239–243 (*see also* 3-D plots/ plotting)

data, manipulation of 256 enhancement tools 237 figures as containers for 232 functions for enhancement

232–233

manually editing 234–235

subplots 233–234

surface plots 243–256 (*see also*

surface plots)

plotyy() function 237

polar() function 239, 261

polyfit() function 347–348, 361,

384, 385

polynomial regression 347–349

polyval() function 348, 361

Prim’s algorithm 404–406

priority queues 391–393

problem-solving 14–15, 211–226. *see also* data collection

assembling solution steps for 212 example 221–226

inserting into collection 213–215 larger problems 220–221

plan for 212

procedural abstraction 19, 106, 268

procedural programming 20

program bugs 12

programming 211

programming languages 10–11

overview of 18–20

# Q

queue(s) 390–396

implementation 390–391

nature of 390

operations on 390

overview 390

priority 391–393

testing 393–396

quick sort 375–377, 383

# R

radix sort 379–380, 383

RAM. *see* Random-Access Memory (RAM)

rand() function 47, 62, 75

randn() function 47, 62, 75 Random-Access Memory (RAM) 7 read() function 173

Read-Only Memory (ROM) 7

readStruct() function 179

recording, sound 316–317

recursion 185–206

activation stack 186–187

defined 187–188

endless 188

examples 197–202

implementation 188–190

reshape() function 67, 279, 286 resolution

of images 292

of recorded data 317

<return info section>, 107 RGB (red, green, and blue) 292

rmfield() function 147, 153, 155 ROM. *see* Read-Only Memory (ROM) rot90() function 310

rotations

2-D 275–278

3-D 278–281

round() function 55, 75

runtime errors 12

# S

save() function 180

scalar vectors 51

scale, playing a musical 322–323 scripts 33–39

creating 33–34

debugging 36

example using 36–39

punctuating 35

running 35

searching (operation) 213, 219–220

semicolon (;) 28, 62

semilogx() function 237, 261

semilogy() function 237, 261

setfield() function 155, 161

shading() function 233, 261

short-circuit AND (&&) 53, 54, 87

short-circuit evaluation 87–88

short-circuit OR (||) 53, 54, 87 shortening, of vector 50–51 shrinking images 295–296

Simpson’s rule 351, 353 simultaneous linear equations,

solving 281–283

size() function 48, 58, 61, 75, 146 slicing

of arrays 67

of character strings 124

of sounds 318–320

of vectors 56–58

software, computer 8–10

categories of 8

interaction with hardware 8

tools (*see* software tools)

software tools 9–10

solid-state memory 6–7

sort() function 155, 161, 382

sorting 213, 220, 367–386

algorithm for 371–380

applications 382–383

bubble 373–375, 383

example using 384–386

insertion 371–373, 382–383

and measuring algorithm

cost 368–371

merge 377–379, 383

quick 375–377, 383

radix 379–380, 383

sound() function 318, 319,

323, 335

sound(s) 315–334

example using 332–334

Fast Fourier Transform 325–328

(*see also* Fast Fourier Transform

(FFT))

frequency domain

operations 328–332

intensity 316

musical 321–324

physics of 316

recording/playback 316–317

slicing/concatenating 318–320

source code 12

spacecraft launch, example 36–39 sparse matrix 399, 400

sphere, construction of 249–250

sphere() function 255, 262

spline() function 344, 361

spreadsheets 170–172

sprintf() function 126, 129, 130,

136, 348

square array 61

sscanf() function 127, 129, 136

stack 8, 186–187

strcmp() function 130, 131, 136

strcmpi() function 131, 136

stretching images 295–296 strings

cell arrays conversion to 394

character 121–135 (*see also*

character strings)

**I–8** Index

strings (*continued*) conversion from numbers

to 125–127

|  |  |  |
| --- | --- | --- |
| str2num() function | 127, 128, 129, 136 | true color images 293 |
| strong typing 24  strtok() function | 129, 180 | true values 50, 82, 84, 86, 100  try keyword 192, 193, 194, 207 |
| struct() function | 149, 150, 161 | tune, playing 323–324 |

structure arrays 150–156

constructing 150–152

elements, acessing 152–154

inserting data into 214

manipulation 154–156

structure(s) 146–150

constructing/accessing 147–148

functions 148–150

manipulation 154–156

subplot() function 233–234, 261

sum() function 55, 66, 75, 155, 346

surf() function 243, 244, 262, 303 surface, images on 303–306 surface plots 243–256

3-D parametric surfaces 248–250

bodies of rotation 250–255 (*see*

*also* bodies of rotation)

compound surfaces, assembly

of 256

cube 243–245

functions to create 243

manipulation of 247–248

parabolic dish 245–247

surfc() function 247, 255, 262

surfz() function 255

switch statement 88–90, 100

MATLAB implementation 89–90

template for 88

synthesizer notes, shaping 359–360

# T

technology, advancement in 2 text files 33

delimited 169, 172–173

reading 174–176

writing 176–177

text() function 233, 261

textscan() function 175, 180

title() function 233, 261

token 123

toString() function 394

trapezoidal rule 351, 352–353

traversing (operation) 213, 215–216

graphs 401–403

tril() function 310

type, data 24

typographical errors 13, 23

# U

uint8/16() function 123, 136,

293, 310

unary minus (2) 52

unary not (~) 54, 87, 100 underscore character (\_) 22 untyped languages 22

user interface 24–33 Command History window

26–27

Command window 25–26, 92,

107, 112, 122

Current Directory window

30–31, 34–35, 107

Editor window 32–33, 36

Figure window 31–32

Variable Editor window 31

Workspace window 27–30, 48

utilities, operating systems 9

# V

value parameters 126

value(s) 24

assigning, to variables 21–22

boolean/logical 50

parameters 126

Variable Editor window 31 variable scoping 112 variable(s)

assigning values to 21–22

global 112–113

names 21–22

vector(s) 46–60

arithmetic operations with 51–52

concatenation of 55

creating 47–48

elements 47

extracting/sorting 382

indexing of 48–50

inserting data into 213

library functions with 54, 55 logical operations with 52–54 operating on 51–58

scalar 51

shortening 50–51

size of 48

slicing 56–58

vectors of indices 56

view() function 247, 261

virtual memory 7

von Neumann architecture 4–5

# W

waterfall() function 255, 262

.wav files 317, 318

wavread() function 317, 335

wavwrite() function 335

weak typing 24

weighted graph 396, 398

while loop 94–97, 100

breaking 97

example 95

loop-and-a-half iteration

style 96–97

for loop *vs.,* 90

MATLAB implementation 95

structure of 94

template for 94–95

who command 30

whos command 30

workspace, saving 168–169

Workspace window 27–30, 48

wrapper function 185, 195–197

template for 196

# X

xlabel() function 233, 261

xlsread() function 170, 177, 180

xlswrite() function 170, 172, 180

# Y

ylabel() function 233, 261

# Z

zeros() function 47, 62

zlabel() function 233, 261