

Index

- 3D surface diagrams, Excel 348
- abstract classes, concepts 286, 325–7, 371–3, 376–8
- Abstract Factory pattern, concepts 262–3, 272, 275–9
- AbstractProductFactory class 276–9
- Access 49
- accuracy requirements, software quality 79
- adaptor containers, background 21, 42–3, 45–6
- add-ins
 - automation add-ins 358–73
 - COM add-ins 357–8, 367–73
 - concepts 356–73
 - Excel 356–73
 - types 357–8, 367–73
 - XLA 357
 - XLL 357
- AddMenuItem function, COM add-ins 372–3
- AddPropertySet 350–5
- ADI *see* Alternating Direction Implicit
- aggregate classes, concepts 52–7, 374–82
- aggregation relationships
 - see also* Whole–Part . . .
 - concepts 282–8, 374–5
- Alexander, C. 261, 281, 296
- algorithms 3–4, 6, 20–35, 43–4, 79, 234–6, 376–8
 - see also* mutating . . . ; non-mutating . . .
 - Alternating Direction Implicit 234–6
 - concepts 3–4, 6, 20–35, 43–4, 79, 234–6, 376–8
 - examples 33–5, 43–4, 376–8
 - Standard Template Library 20–35, 43–4
 - useful algorithms 43–4
- Alternating Direction Implicit (ADI) 5–6, 94, 168, 225–44
- Asian options 242–3
- Black–Scholes models 228, 237–44
- boundary conditions 232–9
- C++ 234–6
 - concepts 5–6, 94, 168, 225–44
- critique 228, 230, 236, 241
- design issues 234–6
- heat equations 226–36
- historical background 227–8, 241
- multidimensional partial differential equations 225–44
 - solution 234–6
- American options 93–4, 102, 168, 171, 239, 343
 - see also* options
 - concepts 93–4, 168, 171, 326–7, 343
 - definition 343
- ANSI standards 64
- appendices 383–96
- application hierarchy, Bridge pattern 290–4
- application level, layering concepts 4, 65–6
- applications 1–4, 78–90, 212–24, 247–308
 - see also* software
 - categories 3–4, 78–90
 - conversion activities 247–52, 347–55
 - design patterns 1–2, 5, 21, 59–60, 64, 103–5, 112–16, 212, 217, 247–8, 259–308
 - flexibility levels 3, 103–5, 109, 111–12, 190, 218, 248–60, 261–3, 296, 298, 302
 - presentation activities 247–52, 297, 347–55, 375, 380–2
 - registration activities 247–9, 258–9, 347–8, 352–5, 368–9
- arithmetic averages, Asian options 237–9
- arrays
 - see also* vectors
 - associative arrays 63, 74–6, 356
 - boundary errors 80, 89, 97–9, 116
 - concepts 4, 9, 21, 24–7, 33–5, 63–90, 101–2, 114–16, 155–63, 218–19, 250–60, 396
 - creation 66–72, 250–60
 - dates 396
 - definition 66–8
 - examples 9, 25–7, 33–5, 66–72, 75–6, 101–2, 114–16, 218–24, 250–60, 396
 - Excel 252–8, 346–73
 - indexing preferences 66, 74, 114
 - libraries 63–77
 - max/min index values 67–76, 97–9, 142, 169–71, 367
 - printing functions 38–40, 69–70, 75–6, 250–60, 347–55, 380–2

- arrays (*continued*)
 size constraints 66
 standards 64
 work arrays 218–19
- ASCII files 212–13, 220, 223–4, 261–2, 315
- Asian options 225, 237–9, 242–3, 296
- assembly parts, Whole–Part patterns 287, 375–82
- asset-or-nothing options 189–90
- assignment operator (=) 15, 264, 386–7
- association relationships, concepts 283–6
- associative arrays/matrices, concepts 63, 74–6, 356
- associative containers, concepts 21, 27–30, 40–3
- asymmetric capped power call options 210–11, 221
- asymmetric plain vanilla power call options 210–11
- AT&T labs 383
- ATL wizard 358–73
- atomic functions, concepts 112–15, 155–63, 215–24
- AtomicDFunction class 112–15, 155–63, 215–24
- attribute lists and value types, Extensible Markup Language 318–20, 330–45
- attributes
 Document Object Model 335–45
 Extensible Markup Language 314–27, 330–45
 options 9, 29–30, 36, 47–60, 111, 266–70, 284, 339–40, 342, 344–5, 383–93
- automation add-ins, Excel 358–73
- averages 78, 85, 121–2, 237–9
- AWK language 74
- Babushka, I. 178
- backward sweep, *LU* decomposition 95–102, 162
- Balayage method 98
- banded matrices 36, 63, 66, 69, 93
- barrier options 13–14, 178–80, 195–6, 210–24, 327, 342–5
see also exotic ...
- base classes, concepts 286, 325–7, 371–3, 376–8
- Basic Linear Algebra Subprograms (BLAS), concepts 78
- basket options 225, 240–1, 326–7
- bear markets 13, 16
- bear spreads 374–8, 381
- behavioural patterns 5, 260, 296–307, 338–45, 396
see also extension ...; notification ...; variational ...
 concepts 5, 260, 296–307
 examples 299–307, 338–45
- Bermudan options 326–7
- Bernoulli distribution 124–6
see also binomial ...
- beta 95–7
- Bhansali, V. 225
- bidirectional iterators, background 22, 30–3
- binary associations, concepts 283–5
- binary function, function objects 23, 106–11
- binary options 179–80, 189–90, 327
- binary searches 27, 33–5
- binomial distribution 124–6
see also Bernoulli ...
- binomial method 1, 2–3, 36, 38, 124–6, 178, 182, 244, 383
- black boxes 64, 91, 175–6, 235, 247, 262–3, 289
- Black–Scholes models 2–5, 64, 78, 91–4, 102, 105, 131, 137–8, 143, 155, 158–61, 167, 175–244, 355
see also partial differential equations
 Alternating Direction Implicit 228, 237–44
 concepts 2–5, 64, 78, 91–4, 102, 105, 131, 137–8, 143, 158–61, 167, 175–202, 228, 235, 237–44, 355
 equations 186–7, 235
 exponentially fitted schemes 102, 137–8, 160, 179–80, 197, 199, 203–24, 241
 finite difference method 182–202, 237–44
 heat equations 190–1, 228
 terminal conditions 188–9, 192–202, 240
- BLAS *see* Basic Linear Algebra Subprograms
- block SOR, iterative methods 167
- block tridiagonal matrices, concepts 37–8, 93, 99–102, 164–7, 227–43
- bonds 1, 285
see also fixed-income applications
- booleans 14–15, 16, 51, 96, 106–8
- Borland 20
- bottom straddle, concepts 375–6
- boundary conditions 97–9, 155, 156–7, 159–60, 161–2, 176, 182–91, 195–9, 206, 226–36, 239–40
- boundary errors 80, 89, 97–9, 116
- boundary layer, concepts 197
- boundary value problems, concepts 4–5, 97–9, 155–63, 170–9, 182–91, 195–201, 212–24, 232–9, 355
- Box method *see* Crank–Nicolson scheme
- Box–Muller method 148–50
- Bridge pattern, concepts 290–4
- Bronson, R. 80–3
- Brownian motion 109, 151
- building blocks, concepts 3–4, 33, 36–46, 78
- bull markets 13
- bull spreads 282–3, 374–82
- Buschmann, F. 2, 247, 261, 281, 286–7, 306–7, 375
- butterfly spread 374–8, 381
- BVPSolver class 162–3
- C# 180, 279, 286, 296, 327, 335, 343, 347–8
- C++
see also individual topics
 advantages 2, 103–4, 180, 311, 343–4, 383
 complexity problems 297
 concepts 1–6, 9, 103–28, 180, 212–24, 261–79, 311, 334–45, 350, 383–93
 display example 247–60, 375

- Document Object Model 334–45
- Excel interfaces 224, 247–60, 279, 296, 305, 346–73, 380–2, 393
- expertise development 5–6
- Extensible Markup Language 302–5, 311, 328–30, 334–48
- generic programming 9–19, 36, 112
- interface concepts 224, 247–60, 262–3, 311, 334–73
- iterative solutions 168–71, 219–20
- main features 9, 383–93
- overview 1–6, 180, 383
- self-test 383–4
- walk-through example 212–24
- C 64, 103
- CAD *see* Computer Aided Design
- calculate function 113
- call options 51–9, 210–11, 221, 290, 313, 342–5, 374–82
- callable bonds 224
- candlestick charts 13–14
- canonical distribution classes 262
- caps 324–7
- CASE tools 290
- cash flows 3, 36, 241, 272, 284
- CD, source code 1, 6, 52, 54, 83, 121, 153, 162–3, 170, 190, 202, 211, 219, 224, 236, 298–9, 302, 307, 342, 383, 394
- cdf *see* cumulative distribution function
- cell data, Excel 252–60, 346–73, 380–2
- centred divided differences
 - see also* Crank–Nicolson ...
 - concepts 157–9, 162, 179, 190–1, 192–204, 208, 222–4, 226–7, 234–9, 241–4, 355
- Ceponkus, A. 321–2
- CFL *see* Courant–Friedrichs–Lewy condition
- char pointers 19
- CharacterData interface, Document Object Model 335
- characteristic direction, concepts 239
- charts, Excel 250–8, 346–55, 380–2
- chi-squared distribution 119
 - see also* Gamma ...
- child elements, Extensible Markup Language 313–27, 331–45
- Choice specifier, Extensible Markup Language 316–17, 330–1
- chooser options
 - see also* exotic ...
 - concepts 57
- Circle class 299, 333–4
- classes 1–2, 3–4, 9–60, 63–77, 103–16, 117–28, 212–24, 247–95, 383–93
 - see also* function ...; template ...
 - Abstract Factory pattern 262–3, 272, 275–9
 - aggregate classes 52–7, 374–82
 - assembly parts 287, 375–82
 - Bridge pattern 290–4
 - concepts 1–6, 9–19, 63–77, 104, 111–16, 261–80, 383–93
 - creational patterns 5, 260, 261–80, 350–5
 - date class 394–6
 - Factory pattern 262–3, 272–5
 - foundation classes 3–4, 6, 63
 - object links 9, 103–16, 261–80, 383–93
 - option trading strategies 374–82
 - overview 1–2, 3–4, 261–3
 - Singleton pattern 251–2, 262–70
 - statistical distributions 117–28
 - structural patterns 5, 260, 281–95
 - time class 36, 395
- client code, concepts 89–90, 261–2, 290–4, 386–93
- CMyExcelFunctions class 360–73
- COC *see* cost of carry
- code
 - see also* software
 - background 1–11, 19–20, 313–19, 328–45
 - CD 1, 6, 52, 83, 121, 153, 162–3, 170, 190, 202, 211, 219, 224, 236, 298–9, 302, 307, 342, 383, 394
 - client code 89–90, 261–2, 290–4, 386–93
 - interpretation/compilation contrasts 357, 392
 - optimisation issues 6, 87, 213–14, 296
 - syntax 19–20, 313–19, 328–45
- code files, template classes 11–16
- cognitive distance, concepts 235
- cognitive overload 36
- collection members, Whole–Part patterns 287
- COM *see* Component Object Model
- combinations, concepts 374–8
- Command pattern, concepts 106, 215, 298
- CommandBarButton object 369–73
- CommandBarButtonEvents COM class 371–3
- comment lines, Extensible Markup Language 314–15
- comparison functions 34, 45, 87, 108
- compatibility problems, heat equations 187–8, 198
- compilation/interpretation contrasts, code 357, 392
- complex/simple data types, XML Schema 330–4
- complexity problems 290, 297
- Component Object Model (COM) 335–8, 346–55, 356–73
- components, Financial products Markup Language 324–5
- Composite pattern, concepts 288–9, 374–5
- composite vectors, concepts 101
- composition techniques, concepts 57–9
- composition type, aggregation relationships 287–8, 374–5
- compound options, concepts 283
- computational finance, concepts 1, 175–81
- Computer Aided Design (CAD) 271–2
- COMUtils class 366–7, 368–73
- concrete classes, definition 286
- conditionally stable explicit finite difference schemes, concepts 186–7, 227, 229
- conjugate gradient method, iterative solutions 93, 164, 167
- const type, examples 25–33, 38–9, 51–9, 67, 72–5, 85–7, 113, 118, 123–6, 140–1, 214–17, 220–4, 250, 256, 292, 299–300, 339–40, 382

- constructors 11–15, 24, 50–4, 66–9, 118–19, 123, 169, 215–17, 261–4, 270–5, 303, 342, 349, 385–93
- container contents, Whole–Part patterns 287
- containers
see also adaptor ...; associative ...; sequence ...
 concepts 20–35, 36–46, 66, 81–4, 338–9, 395–6
 recursive classes 36–7
 Standard Template Library 20–35, 43–6
 useful algorithms 43–4
- contingent premium options 189–90
- continuous functions, mapping types 110, 140–3
- continuous probability distributions, concepts 63, 117–28, 146–7, 277–9
- continuous sample space, concepts 145, 177–8, 205–8, 238–9
- continuous-state processes, random processes 150–1
- continuous-time processes, random processes 150–1
- convection-dominated problems, concepts 203–11, 228, 243
- convection–diffusion equations
see also Black–Scholes models
 concepts 4, 5, 102, 160, 180, 182–3, 195–9, 203–11, 217–18, 225–44
- convergence issues 136–8, 166–70, 207–9
- conversion activities, applications 247–52, 347–55
- convertible bonds 225, 239
- Cooney, M. 203
- copy concepts 15, 21–3, 41–3, 50–1, 68–9, 270–2
- core processes, activities 1, 80–4, 351–4
- cost of carry (COC) 52–7, 111, 268–9, 339–40, 385–93
- CoUninitialise function 337, 338
- Courant, R. 180, 210
- Courant–Friedrichs–Lewy condition (CFL) 210
- covariance 88
- Crank–Nicolson scheme
 concepts 2–5, 102, 134–40, 179–80, 185–6, 189, 191–9, 203, 208, 211, 226–7, 243
 critique 195–9, 211
 failings 2, 4, 102, 138, 189, 195, 203, 208, 211
 output examples 211
 spurious oscillation problems 102, 136, 138, 160–1, 189, 195–201, 203
 theta scheme 185
- CreateChart function 250–8, 349–50, 380–2
- CreateElement function 336, 338
- CreateInstance function 336, 337–8
- creational patterns
see also Abstract Factory ...; Prototype ...; Factory ...; Singleton ...
 concepts 5, 260, 261–80, 340, 350, 396
- cross-derivative terms, approximations 231–2, 237–44
- Crout’s method 93
- Cryer, C.W. 94, 164, 167
- cumulative distribution function (cdf), concepts 17–18, 117–24, 146–7, 247–60, 351–5
- currency
 binary associations 284
 FX resettable cross-currency swaps 324
- curse of dimensionality, concepts 225–6
- curve-fitting operations 78
- Dahlquist, G. 136, 166
- Data Definition Language (DDL) 315
- data structures 3, 9, 30, 36–46, 63–77, 210–11, 282–95, 328–9, 343, 394–6
see also arrays; matrices; vectors
- databases 49, 53, 59, 261–2, 297, 355
- DatSIM 162, 343, 355, 383–4, 388, 394–6
- date class, concepts 394–6
- date manipulations 3
- date-and-time types, XML Schema 328–9
- Dautray, R. 209
- DDL *see* Data Definition Language
- debug facilitation, software quality 3, 11, 82–3
- decision support systems 3
- decision tables 76
- Decorator pattern 76
- default virtual functions
see also polymorphism
 concepts 117–19, 128, 272
- delegation mechanism, concepts 5, 21, 57–9, 112
- delta 2, 5, 52–7, 111, 138, 160–1, 180, 189, 198, 201, 203, 210–24, 268–9, 305–6, 333, 378–82
 concepts 378–82
 hedging 378–9
- Deming, W.E. 282
- Demming, R. 356–73
- deployment issues 311–82
- derivatives
see also futures; options; swaps
 approximations 160–1, 195–7, 231–4, 237–44
 concepts 283–4, 324–7, 342–5, 374–82
- derived classes, concepts 286
- design patterns 1–2, 5, 21, 59–60, 64, 103–5, 112–16, 212, 217, 247–8, 259–308, 396
see also behavioural ...; creational ...; structural ...
 concepts 1–2, 5, 21, 59–60, 106, 212, 215, 217, 247–8, 259–308
 historical background 261–3
- destructors 13–15, 51–4, 169, 263–70, 303, 385–93
- DFunction class 215–17
- diagonal matrices, concepts 37–8
- diagonally-dominant matrices, concepts 101–2
- diagrammatic output, Excel 253–4, 348–55, 380–2
- dialog boxes 343, 347–8, 364
- difference operations, sets 40–3

- differential equations 1–2, 4–6, 111, 131–71, 180, 201, 301–2
 - see also* ordinary ...; partial ...; stochastic ...
- digital options *see* binary ...
- direct solutions, numeric linear algebra 91, 93–4, 164, 168, 177
- Dirichlet boundary conditions 97–9, 156–7, 159, 161–2, 176, 182–6, 195–9, 226–36, 239–40, 243
- discontinuous payoff functions 2, 182, 187–9, 197–8
- discount curves 36
- discrete functions, mapping types 110, 140–3
- discrete maximum principle, concepts 206–8
- discrete probability distributions, concepts 63, 117–28, 146–7, 277–9
- discrete sample space, concepts 145, 177–8, 205–8, 238–9
- discrete-state processes, random processes 150–1
- discrete-time processes, random processes 150–1
- discretisation uses, finite difference method 133–7, 140–4, 167–8, 177–9, 182, 190–3, 196, 205–8, 212–26, 242–3
- disjoint sets 40–3
- dispersion measures *see* measures of dispersion
- displays
 - see also* printing ...
 - C++ example 247–60, 375, 380–2
 - design patterns 247–8, 259–60
 - Excel 247–60, 346–55, 375, 380–2
- divided differences, concepts 133–4, 157–9, 162, 179, 190–1, 192–204, 208, 222–4, 226–7, 234–9, 241–4, 355
- dividends 220–4, 344–5
- DllRegisterServer function 369
- DLLs 3, 336, 337, 357, 369
- DllUnRegisterServer function 369
- document interface, Document Object Model 335–45
- Document Object Model (DOM), concepts 303–5, 328, 334–45
- Document Type Definition (DTD)
 - see also* Extensible Markup Language
 - advantages 315, 320, 334
 - concepts 311, 315–27, 328, 331, 334, 345
 - entities 319
 - examples 315–20, 325–7, 331
 - limitations 320, 328, 334, 345
 - multiplicity option 317, 331–2
 - syntax 315–19, 328, 331
 - uses 315, 320, 328
 - validation issues 319–20
 - XML schema 334
- documentation needs, software 2, 64, 265–6, 272–6, 279, 285, 297, 338–9, 356
- documents, Extensible Markup Language 312–13, 314–15, 328–45
- DOM *see* Document Object Model
- domain concepts 109–11, 145–7, 237–9
- Doolan, E.P. 204
- double barrier call options 210–11
- double dispatch mechanism 341
- double sweep method, Godunov 97–9, 162, 192, 212–24, 229–30, 234–6, 242, 290
- doubles, examples 9–11, 26–33, 44, 50–9, 67–76, 86–9, 97, 104, 110–16, 120–2, 126, 141–50, 214–60, 268–70, 274–5, 292–4, 304–5, 329–30, 339, 344–5, 351–4, 361–7, 376–82, 386–93
- DoubleSweep class 212–24
- Douglas, J. 228–9
- down-and-in put options 343–5
- down-and-out put options 343–5
- drand48 function 148
- drift 180, 208–9, 220–4
- DTD *see* Document Type Definition
- Duffy, D. 1, 6, 102, 133, 137–8, 155, 160, 180, 197, 199–201, 204–6, 209, 218, 239, 241, 243, 248, 258, 281, 285, 292, 297, 356–73
- Duffy, I. 168
- dynamic associative arrays
 - see also* maps
 - concepts 29–30
- efficiency characteristics, software quality 3, 49, 65, 79–80, 213–14, 219–20, 296, 356, 391–2
- eigenvalue analysis 78, 83–4, 102
- element types, Extensible Markup Language 315–17, 330–1
- elementary events, concepts 145–7
- elements
 - Document Object Model 335–45
 - Extensible Markup Language 312–27, 328–45
 - XML Schema 329–34
- elliptic PDEs 177
- embedded data structures 30, 67, 113
- Emel'yanov, K.V. 161
- encapsulation concepts 47–8, 103–5, 112–14, 117, 215–18, 297, 301–5, 356, 383–6, 394–6
- entities
 - Extensible Markup Language 319, 328–45
 - function objects 106, 112–16
- EntityThing class 338–45
- EntityVisitor class 302–3, 340–5
- equalities 50–1, 78, 84, 87, 102, 108, 164, 196–7, 207–8
- equity derivatives 324–7
 - see also* derivatives
- ergodic processes, random processes 151
- Euclidean norm 81–3, 97
- Euler difference schemes, concepts 134–7, 139, 142–4, 152–4, 179–80, 182–91, 192–4, 257–8
- European options 9–10, 29–30, 51–60, 189, 198, 266–70, 274–5, 313, 339–45, 377–8, 384–93
 - see also* options
 - attributes 9, 29–30, 51–60, 266–70
 - definition 343
 - examples 9–10, 29–30, 51–60, 266–70, 274–5, 313, 339–45, 377–8, 384–93

- European options (*continued*)
- executive options 57–8
 - Extensible Markup Language 313, 326–7, 339–45
 - Financial products Markup Language 326–7
 - maps 29–30
 - payoff functions 189, 198, 343, 377–82
 - template classes 9–10, 266–70
- events, concepts 145–7, 370–3
- Excel 5, 49, 59, 65, 74–6, 162, 224, 247–60, 270, 279, 296–7, 305, 346–73, 380–2, 393
- add-ins 356–73
 - advanced topics 356–82
 - application life-cycle 247–52, 348–9
 - automation add-ins 358–73
 - basic data 247–9, 347–55
 - C++ interfaces 224, 247–60, 279, 296, 305, 346–73, 380–2, 393
 - cell data 252–60, 346–73, 380–2
 - charts 250–8, 346–55, 380–2
 - concepts 247–60, 279, 296–7, 305, 346–73, 380–2
 - converted data 247–52, 347–55
 - core processes 351–5
 - data sources 247–52, 347–55
 - design issues 355–73
 - diagrammatic output 253–4, 348–55, 380–2
 - display examples 247–60, 346–55, 375, 380–2
 - documentation shortfalls 356
 - event sinks 370–3
 - examples 247–60, 349–73, 380–2
 - Excel data sources 354–5
 - Extensible Markup Language 347–8, 352, 373
 - financial engineering 346–73
 - finite difference method 252–3, 255–8, 279, 382
 - future trends 373
 - initial value problems 255–8
 - matrices 252–8, 346–55
 - presentation activities 247–52, 347–55, 375, 380–2
 - property sets 258–60, 350–5
 - ranges 346–7, 364–7
 - registration activities 247–9, 347–8, 352–5, 368–9
 - Singleton pattern 251–2, 269–70
 - startup processes 348–9
 - technical details 348–55
 - toggle options 349
 - Tools/Add-ins menu 357–8, 361, 369, 372
 - UML model 247–8, 346–7
 - vectors 247–60, 348–55, 366–7, 380–2
 - workbooks 346–57, 373
 - worksheets 346–55, 358, 362–7
 - XP 358, 364
- ExcelDriver class 250–60, 269–70, 348–55, 380–2
- ExcelRangeToVector function 367
- exception handling 79–80, 84, 88–90
- executive options
- see also* exotic ...
 - concepts 57–9, 274–5
- exercise date, option definitions 343
- exercise price, definition 343
- exotic options
- see also* barrier ...; chooser ...; executive ...;
 - forward start ...
 - concepts 13–14, 47, 57–60, 189–91, 210–11, 343–5, 378–9
 - definitions 343
 - examples 57–9, 187–90, 210–24, 272, 343–5
 - exponentially fitted schemes 210–24, 242–3, 378
 - payoff functions 189–91, 210–11, 343–5
 - types 57, 178
- expected values, concepts 117–28
- experiments, concepts 145–7
- expiry date, option definitions 343
- explicit Euler difference scheme, concepts 134–7, 139, 142–4, 152–4, 179–80, 182–91, 226–7, 257–8
- explicit finite difference schemes
- Black–Scholes models 182–91, 242–3, 257–8
 - concepts 4–5, 134–7, 139, 142–4, 152–4, 179, 182–91, 226–36, 242–3, 257–8
 - implementation 190
 - theta scheme 185
- explicit time marching, exponentially fitted schemes 209–10
- explicit upwind schemes, concepts 210, 242–3
- exponential distribution 119, 123–4
- see also* Gamma ...
- exponentially fitted schemes
- advantages 208
 - Black–Scholes models 102, 137–8, 160, 179–80, 197, 199, 203–24, 241–3
 - concepts 160, 179–80, 203–24, 241–3, 378
 - exotic options 210–24, 242–3, 378
 - explicit time marching 209–10
 - parabolic problems 205–8, 212–24
 - two-point boundary value problems 160, 179–80, 203–11
 - walk-through example 212–24
 - zero volatilities 208–9
- Extensible Markup Language (XML)
- see also* Document Type Definition
 - advanced topics 328–45
 - advantages 311–12
 - attributes 314–27, 330–45
 - background 5, 59, 65, 224, 261–2, 279, 296, 302–5, 311–45
 - barrier options 342–5
 - C++ 302–5, 311, 328–30, 334–48
 - comment lines 314–15
 - concepts 311–45
 - Document Object Model 303–5, 328, 334–45
 - documents 312–13, 314–15, 328–45
 - elements 312–27, 328–45
 - examples 313–27, 329–45

- Excel 347–8, 352, 373
- files 312–13
- financial engineering 313, 324–7, 333–4, 338–45
- Financial products Markup Language 5, 320, 324–7, 334, 343
- historical background 312
- Internet 312, 320–7
- limitations 320
- namespaces 321–4, 329–30
- options 313, 324–7, 338–45
- overview 5, 296, 302–5, 311–15
- parser 335–8
- programming aspects 328–45
- property sets 338–45, 350
- quotes 314–15
- readers 303–5, 338–45
- royalties 311
- Schema 328–34, 373
- structure 312–15
- syntax 313–19, 328–45
- validation issues 319–20, 331–4
- Visitor pattern 302–5, 328, 338–45
- writers 303–5, 338–45
- Extensible Stylesheet Language Transformation (XSLT), concepts 320–3
- extension behavioural patterns
 - see also* Visitor ...
 - concepts 298, 301–5, 328, 338–45
- extension potential, concepts 248
- extrapolated Euler scheme 180
- extrapolation concepts 78, 139–40, 153–4, 158, 180, 201
- extremum operations, vectors 78

- Façade pattern, concepts 289–90
- factorials 126
- Factory pattern, concepts 262–3, 272–9
- Farrell, P. 102, 158, 196, 206
- fault-tolerance properties 79–80
- FDM *see* finite difference method
- FEM *see* finite element method
- FIFO *see* First In First Out
- files, Extensible Markup Language 312–13
- filled diamonds, composition relationships 287–8
- FILO *see* First In Last Out
- financial derivatives *see* derivatives
- financial engineering 1–6, 35–46, 64–5, 108–9, 164, 175–81, 237–44, 313, 324–7, 333–73, 383
 - Abstract Factory pattern 279
 - aggregation relationships 283, 374–5
 - behavioural patterns 296–7, 299–307
 - Bridge pattern 290
 - Composite pattern 289
 - design patterns 259–308
 - Excel 346–73
 - Extensible Markup Language 313, 324–7, 333–4, 338–45
 - Factory pattern 274–5
 - function objects 108–9
 - overview 1–2, 35, 36–46, 64–5, 108–9, 164, 175–81, 383
 - Property pattern 6, 47–60
 - Prototype pattern 271–2
 - Singleton pattern 269–70
 - Standard Template Library 35, 36–46, 108–9
 - two-factor models 237–44
- financial instruments
 - see also* derivatives
 - overview 1–6, 35, 46
- financial models, activities 1
- Financial products Markup Language (FpML) 5, 320, 324–7, 334, 343, 350
 - concepts 5, 320, 324–7, 334, 343, 350
 - XML Schemas 334
- find concepts, non-mutating algorithms 21–3
- finite difference method (FDM) 1–3, 4–5, 38, 64, 78, 91, 93, 102, 116, 131–71, 175–244, 255–8, 279, 290
 - see also* explicit ...; implicit ...
 - Abstract Factory pattern 279
 - accuracy improvements 139–40
 - background 1–3, 4–5, 38, 64, 78, 91, 93, 102, 131–54, 155–63, 175–244, 255–8, 279, 290
 - Black–Scholes models 182–202, 237–44
 - C++ steps 140–4, 162–3, 180, 212–24, 252–3, 255–8
 - common schemes 134–5, 152–4, 179
 - concepts 1–3, 4–5, 38, 64, 78, 91, 93, 102, 131–54, 155–63, 175–81, 182–202, 244, 279, 290
 - convection-dominated problems 203–11, 228, 243
 - critique 244
 - discretisation uses 133–7, 140–4, 167–8, 177–9, 182, 190–3, 196, 205–8, 212–26, 242–3
 - Excel 252–3, 255–8, 279, 382
 - extrapolation techniques 139–40, 153–4, 158, 180, 201
 - initial conditions 177–8, 187–90, 193, 198, 255–8
 - mesh points 133–4, 137, 158, 162, 177–9, 183–6, 193–201, 203–24, 226–36
 - multidimensional diffusion equations 225–44
 - one-factor problems 2, 4–5, 38, 91, 94, 152–4, 179–224, 244
 - overview 1–3, 4–5, 38, 91, 131–44, 244
 - predictor–corrector method 138–9, 142–3, 153, 191, 232, 257–8
 - reading material 180
 - special schemes 137–8, 160, 179–80, 203–24
 - stability issues 136–7, 198–9, 203–11, 227, 229, 237, 244
 - stochastic differential equations 145, 151–4, 239
 - theoretical issues 136–7, 175–81
 - two-factor problems 2, 4–5, 38, 91, 94, 105, 179–80, 195, 225–44

- finite difference method (FDM) (*continued*)
 two-point boundary value problems 155–63,
 170–1, 182–91, 203–11
 walk-through example 212–24
- finite element method (FEM), concepts 92–3,
 176, 177–8, 206–7, 244, 290
- finite volume approximation methods, concepts
 178, 241
- First In First Out (FIFO) 45
- First In Last Out (FILO) 45
- first-class objects, concepts 103, 105, 112
- first-order systems, block tridiagonal matrices
 99–100
- FIX 5
- Fix, G. 191, 240
- fixed-income applications 241
see also bonds; interest-rate products
- flexibility levels, applications 3, 103–5, 109,
 111–12, 190, 218, 248–60, 261–3, 296,
 298, 302
- flexible attributes, options 36
- floats, examples 10–11
- Fokker–Planck equation 195
- Fortran 66, 78, 126
- forward iterators 21–2, 30–3
- forward start options
see also exotic ...
 concepts 57, 326–7
- forward sweep, *LU* decomposition 95–102
- foundation classes, concepts 3–4, 6, 63
- Fourier, J.B.J. 1, 2–3, 180
- Fourier transform methods 1, 2–3, 180
- fractional step methods *see* splitting methods
- Frobenius norm 82–4
- full matrices, concepts 37–8, 39–40, 63, 66,
 69, 92–3, 290
- full signature, concepts 117–18
- fully implicit scheme
see also implicit Euler ...
 concepts 192–4, 226–7, 242–3
- function classes, creation 111–16, 215–24,
 247–52, 272
- function objects (functors)
see also binary ...; generator predicate; unary
 ...
 concepts 21, 23, 103, 106–11, 298
 examples 106–11
- function pointers, concepts 103–6, 109, 140–1,
 217, 270, 275–9, 348–55
- function templates
 concepts 9, 11–12, 16–19, 78–90, 106–16
 examples 16–19, 78–90, 111–16
- function types, concepts 109–11, 214–15
- functionality characteristics, software quality 3,
 49, 296, 340–1, 391–2
- functions 9, 11–12, 16–19, 78–90, 103–16,
 212–24, 249–60, 364–7
 arrays 114–16
 C++ modelling 103–16, 212–24, 249–60
 mappings 109–11, 140–3, 175–6
 options pricing 111, 212–24, 272,
 383–93
 random variables 145–54
 relations 109–11
 types 109–11, 214–15
- functors *see* function objects
- futures 283–5
see also derivatives
- fuzzy sets 109
- FX resetable cross-currency swaps 324
- gamma 52–7, 95–7, 111, 119, 160, 180, 189,
 198, 211–24, 261–3, 268–9
- Gamma distribution, concepts 119, 122–4, 265
- Gamma, E. 2, 5, 47, 49, 60, 76, 103, 106, 109,
 116, 215, 218, 247, 261–3, 272–4, 281,
 288–90, 297–8, 302, 314, 328, 339, 341,
 348, 376, 393
- Gauss, C.F. 180
- Gaussian distribution
 concepts 17–18, 119, 121–2, 147, 151,
 238–9, 247–60, 269, 351–5
 examples 121–2, 247–60, 269, 351–5
- Gaussian elimination, full matrices 93
- Gauss–Seidel method, iterative solutions 93,
 164, 166, 168–9
- GDI 297, 393
- generalisation relationships (gen/spec), concepts
 286
- generalised Wiener process, concepts 152
- generator predicate, function objects 21, 23,
 106–11
- generic design patterns 1–2
- generic programming, concepts 9–19, 20, 36,
 112
- geometric averages, Asian options 237–9
- geometric Brownian motion, concepts 152
- geometric distribution
see also Pascal ...
 concepts 124, 126–7
- geometric-mean function, concepts 85
- get functions 48–9, 66, 73, 113
- GetPropertySet 350–5
- ghost points, boundary conditions 155, 159–60,
 197–9
- ‘glorified pointers’ 21–2
- GNU compiler 20
- gnuplot package 224
- Godunov’s double sweep method, concepts
 97–9, 162, 192, 212–24, 229–30, 234–6,
 242, 290
- GOF 5, 262, 281, 297, 302, 307, 339
- gold-plating 162
- Graphical User Interfaces (GUI) 289, 297, 352
- Greeks, overview 5
- Group specifier, Extensible Markup Language
 316–17, 330–1
- Gudgin, M. 328
- GUI *see* Graphical User Interfaces
- half-products, concepts 3–4
- harmonic-mean function, concepts 85
- Haug, E. 42, 49, 57, 121, 210–11, 249, 274,
 283–4, 343, 385, 391–2
- header files 11–16, 112–13, 169, 384, 386–8
- heap, concepts 262–3

- heat equations
see also one-dimensional ... ;
 three-dimensional ... ; two-dimensional ...
 concepts 4–6, 179, 182–91, 226–36
 examples 183–7, 226–36
 historical background 182–3
- hedging applications 203, 305–7, 333, 374–82
- high-flexibility level, applications 3
- historical volatility, concepts 88
- Hochstadt, H. 132
- holistic views 175
- Hoodbhoy, F. 321–2
- Hsu, H. 147, 150
- HTML 320–3
- Hull, J. 2, 88, 122, 342, 374–5, 377–8, 381–2
- hyper-geometric distribution, concepts 124
- hyperbolic PDEs 177, 188, 197, 204, 210–11, 243
- IBM 20
- IBVPs *see* initial boundary value problems
- identifiers, Extensible Markup Language
 318–19, 330–2
- IDispatch interface, automation add-ins 361, 368–9
- IDispEventImpl base class 371–3
- IDTExtensibility2 interface, COM add-ins
 367–70
- Il'in, A.M. 204
- implementation hierarchy, Bridge pattern
 290–4
- implicit Euler difference scheme, concepts
 134–40, 153–4, 179–80, 182–91, 192–4, 226–7, 257–8
- implicit finite difference schemes
see also Crank–Nicolson ... ; fully ... ; Keller box ...
 concepts 4–5, 134–40, 153–4, 179, 182–91, 192–202, 206, 209, 226–36, 242, 257–8
 types 192–202, 209, 226–7
- implicit upwind schemes, concepts 209, 242–3
- IMSL 64
- IMyExcelFunctions interface 364–5, 368
- in barrier options, concepts 343–5
- independent increments, random processes 151, 153
- independent processes, random processes
 150–1
- indexing operator 25–7, 29, 66, 74, 81–4, 114–16, 169–71
- inequalities 87, 102, 108, 164, 168, 171, 196–7, 207–8
- information hiding principle, concepts 82, 117, 175–6, 220, 334–5
- inheritors 5, 60, 103–5, 117, 124, 262, 271, 286
- inhomogeneous forcing term, concepts 132
- initial boundary value problems (IBVPs),
 concepts 177–8, 183–6, 195–201, 205–8, 212–24, 228, 355
- initial value problems (IVPs) 131–44, 151–5, 176, 179–80, 187–90, 195–201, 205–8, 212–24, 228, 255–8, 382
- C++ steps 140–4, 212–24, 255–8, 382
 concepts 131–44, 255–8
 definition 132
 examples 132–44, 212–24, 255–8, 382
 Excel 255–8
 non-linear scalar problems 138–9, 143, 153
 predictor–corrector methods 138–9, 142–3, 153, 257–8
 scalar linear problems 132–8, 140–4, 151–4, 255–8, 382
- inner product, concepts 43–4, 66, 72, 81–4
- input iterators 21–2
- input mechanisms, applications 247–52
- instantiated template classes
see also template classes
 concepts 9–13, 15–16, 19, 26–7, 114, 117, 212–24, 248, 263–5, 277–9
- instrument life cycle, concepts 1, 175–81
- integers, concepts 9–10
- integral calculus 2, 4, 100
- integration constants, two-point boundary value problems 156–7
- integration factors 132
- integration operations, function templates 17, 111
- interest rates 3, 131, 152, 220–4, 239–40, 268–9, 284, 339–45, 385–93
 caps 324–7
 convertible bonds 239
 curves 3
 floors 324–7
 swaps 284–5, 324–7
 swaptions 324–7
 term structure 152, 241
- interest-rate modelling problems 131, 231–2
- interest-rate products 1, 152, 224, 241
- interfaces
 Abstract Factory pattern 262–3, 275–9
 concepts 50–1, 105, 118–24, 224, 247–60, 262–5, 279–88, 296, 302–5, 311, 328–30, 334–73
 Excel 224, 247–60, 279, 296–7, 305, 346–73, 380–2, 393
 Extensible Markup Language 5, 59, 65, 224, 261–2, 279, 296, 302–5, 311, 328–30, 334–48
 Factory pattern 262–3, 272–5
 Whole–Part patterns 287–8
- Internet 312, 320–7
- interoperability requirements
 Excel 259, 350–73
 libraries 65
 SimplePropertySet class 259, 350–5
- interpolation operations, function templates 17, 111
- interpretation/compilation contrasts, code 357, 392
- intersection operations, sets 40–3
- inverse, matrices 83–4, 92, 102, 134, 169, 179, 207
- iostream library 38–9, 56, 247–52, 256–8, 260, 348, 354

- ISO 9126 quality characteristics 3, 80, 180, 391–2
- ISO standards 3, 64, 80, 180, 312, 391–2
- iterative solutions, concepts 91, 93–4, 164–71, 177, 219–20, 290
- Iterator pattern, concepts 297–301
- iterators, concepts 20–35, 42–3, 54, 66, 297–301
- Ito's lemma 241
- IVPs *see* initial value problems
- Jäckel, P. 33
- Jackson, M. 214
- Jacobi method, iterative solutions 93, 164–5, 169
- Java 180, 279, 286, 296, 311, 327, 335
- JavaScript 335
- Journal of Object Oriented Programming (JOOP)* 287
- jump rates, executive options 57–9, 274–5
- jump-diffusion processes 125, 198
- 'junk' functions 48
- Keller Box scheme, concepts 91, 161, 180, 199–201, 378
- keys, containers 21, 27–30
- knockin options 327, 343–5
- knockout options 327, 343–5
- Knuth algorithms 23
- kurtosis functions 78, 86–7
- Laplace, P.S. 180
- lattice methods 1, 38, 182
see also binomial method
- layering concepts, software 3–4, 65–6
- LCPs *see* Linear Complementarity Problems
- learnability requirements, software quality 80
- Levin, A. 241
- libraries 1–10, 38–9, 56, 64–77, 79–90, 117, 128, 247–52, 256–8, 311, 335–8, 346–55, 356–73
see also Standard Template Library
- iostream library 38–9, 56, 247–52, 256–8, 348, 354
- layering concepts 3–4, 65–6
- overview 1–2
- line diagrams, Excel 253–4, 348, 380–2
- Linear Complementarity Problems (LCPs)
see also Projected SOR method
- concepts 164, 168
- linear congruential generators
see also random number ...
- concepts 148
- linear systems of equations 63, 73, 91–102, 164–71, 178, 192–202, 218–24, 227, 234–6
see also numeric linear algebra
- linear TPBVPs, concepts 156–7
- linearity boundary conditions, concepts 155, 157, 159–60, 176, 195–9, 232–4
- Linux 20, 65, 148, 224
- Lions, J.L. 209
- lists, concepts 3–4, 9, 21, 24–7, 37, 52–7, 348–55
- locally one-dimensional methods *see* splitting methods
- lognormal distribution, concepts 119, 122, 238
- long positions, definition 343
- low-flexibility level, applications 3
- lower triangular matrices
see also LU decomposition
- concepts 37–8, 91, 93–102, 165
- LU decomposition, concepts 63, 91, 93–102, 158, 162, 164–5, 168, 192, 229, 234–6, 242, 290
- M*-matrices, concepts 102
- maintainability characteristics, software quality 3, 11, 49, 65, 87, 219–20, 297, 392, 393
- many-to-many relationships, concepts 283–4
- Maple 1, 64
- mappings, functions 109–11, 140–3, 175–6
- maps, background 21, 27, 29–30
- Marchuk, G.I. 140
- Markov chains 134, 150–1
- mathematical algorithms, C++ code 79, 99, 103–16
- MathML 320
- Matlab 1
- matrices 3–4, 6, 37–40, 46, 63–102, 158, 162, 164, 252–8, 290, 346–55
see also LU decomposition; numeric linear algebra
- associative matrices 63, 74–6, 356
- background 3–4, 6, 37–40, 46, 63–102, 252–8, 290
- concepts 17, 27, 37–40, 46, 63–90, 91–102, 178, 218–24, 252–8, 290
- creation 66–72
- definition 68–9
- descriptive detail 68–9
- details 66–72
- diagonally-dominant matrices 101–2
- direct solutions 91, 93–4, 164, 168, 177
- examples 37–40, 66–76, 80, 82–4, 94–102, 254–8, 346–55
- Excel 252–8, 346–73
- internal structure 37–40
- inverse 83–4, 92, 102, 134, 169, 179, 207
- iterative solutions 91, 93–4, 164–71, 177, 219–20, 290
- libraries 63–77
- M*-matrices, concepts 102
- max/min index values 68–76, 97–9, 142, 169–71, 367
- memory issues 92–3
- multiplication 73–4
- non-zero elements 37–8, 92–3
- norms 78, 80, 82–4, 97
- output mechanisms 38–40, 247–52
- positive-definite matrices 101–2
- printing functions 38–40, 69–72, 75–6, 78, 247–60, 347–55, 380–2
- properties 78–90
- solution conditions 101–2

- sparse matrices 36–8, 65, 66, 92–4, 164–8, 290
- specialised operations 78–90
- standards 64
- structure types 37–8, 65, 91–102, 164–8, 290
- vectors 73–4, 83–4, 92–3
- zero elements 37–8, 92–3
- matrix algebra 2, 4, 33, 37–8, 66, 78, 91–102, 164–71, 178, 192–202, 218–24, 227
- matrix inequalities, American options 102, 168, 171
- matrix solvers 92
- maturity, option definitions 343
- max, non-mutating algorithms 21–3
- mean, concepts 78, 85, 121–2, 239
- mean-deviation function, concepts 86
- measures of central tendency, concepts 85–6
- measures of dispersion, concepts 78, 86–7
- mechanisms, concepts 3–4, 10–11, 17, 38–40, 247–52
- Mediator pattern, concepts 297–8, 305–7
- medium-flexibility level, applications 3
- memory issues, matrices 92–3
- merge, mutating algorithms 21–3, 33–5
- mesh points, finite difference method 133–4, 137, 158, 162, 177–9, 183–6, 193–201, 203–24, 226–36
- meshless approximation methods, concepts 178–9
- message-passing metaphor, concepts 390–1
- meta-class aspects, template classes 11, 19
- method of lines (MOL), concepts 144, 179, 190–1
- Meyer, P.L. 117, 123
- MFC graphics library 311
- Microsoft 5, 20, 49, 59, 65, 74, 76, 148–50, 162, 296–7, 303, 335–8, 346, 357, 364, 367
- MidpointRule class 291–4
- Milstein scheme, concepts 153–4
- min, non-mutating algorithms 21–3
- min function template 16–17
- Mirani, R. 210
- MISCFUN function 126
- Model-View-Controller pattern (MVC), concepts 306–7
- modelling, overview 1–6, 103–16, 188–9, 201
- modifiers, definition 385
- MOL *see* method of lines
- moments, skewness and kurtosis functions, concepts 78, 86–7
- Monte Carlo simulation methods 1, 2–3, 33, 211, 290, 296, 383
- Morton, K. 176–8, 180, 187, 195, 196, 199, 203, 206
- multi-asset options 225
- multi-factor problems, finite difference method 2, 4–5, 38, 91, 137–8, 143, 167–8, 225, 237–44, 355
- multi-step method, finite difference method 134
- multidimensional interest rate models 225
- multidimensional PDEs 225–44
- multimaps, associative containers 21, 27–30
- multiplicity option, Extensible Markup Language 317, 331–4
- multisets, associative containers 21, 27–30
- mutating algorithms, background 21–3
- MVC *see* Model-View-Controller pattern
- MyAtExcelAddin project 358–71
- MyConfigurator class 276–9
- MyExcelFunctions class 359–67
- MyPI function, automation add-ins 361–3
- name-and-string types, XML Schema 329
- namespaces, Extensible Markup Language 321–4, 329–30
- negative inverse, matrices 102
- nested elements, Extensible Markup Language 312–27, 331, 339
- nested objects
 - see also* Composite pattern
 - concepts 288–9
- nested vectors, concepts 101, 223–4
- nesting classes, concepts 36–7, 114
- Neumann boundary conditions, concepts 156–7, 159, 176, 197–9, 206, 232–4, 240
- Newton’s methods 138–9, 180
- Newton–Cotes technique 290
- nodes, Document Object Model 334–45
- noise, stochastic differential equations 145
- non-dimensional form, heat equations 183
- non-linear scalar problems, initial value problems 138–9, 143, 153
- non-linearities 2, 138–9, 143, 153, 155–7, 178, 201
- non-mutating algorithms (non-modifying algorithms), background 21–3
- non-zero elements, matrices 37–8, 92–3
- normal distribution *see* Gaussian ...
- normal processes, random processes 151
- norms, concepts 78, 80–4, 97
- notification behavioural patterns, concepts 298, 305–7
- numeric linear algebra
 - block tridiagonal matrices 37–8, 93, 99–102, 164–7, 227–43
 - concepts 63, 78, 83–4, 91–102, 164–71, 177, 219–20, 290
 - direct solutions 91, 93–4, 164, 168, 177
 - Godunov’s double sweep method 97–9, 162, 192, 212–24, 229–30, 234–6, 242, 290
 - LU decomposition 63, 91, 93–102, 158, 162, 164–5, 192, 219–20, 229, 234–6, 242
- numeric types, XML Schema 328–9
- numerical methods, background 1, 36–7, 43–4, 63–77, 78–90, 131–44, 177–81, 192–202, 218–24, 227–8, 237–44, 290–4, 301–2, 343–4
- NumericMatrix class, concepts 63, 69, 72–4, 82–3, 155–63, 168–70, 234–6, 348
- NumIntegrator class 291–4

- object-oriented methods 1–2, 36, 47–8, 64, 103, 112, 117, 144, 208, 212–24, 255–308, 334–5, 383–6, 390
 - see also* encapsulation ...; information hiding ...; inheritors ...; polymorphism ...; run-time switching
- concepts 1–2, 47–8, 103, 112, 117, 144, 208, 261–3, 279–80, 286, 289, 334–5, 383–6, 390
- incorrect methods 286
- overview 1–2, 47–8, 117, 261–3, 279–80, 383–6
- objects 5, 9–10, 103–17, 255–308, 346–56, 383–93
 - see also* design patterns
 - behavioural patterns 5, 260, 296–307
 - classes 9, 103–16, 261–80, 383–93
 - concepts 5, 9–10, 103–17, 255–308, 346–55
 - creational patterns 5, 260, 261–80, 350
 - structural patterns 5, 260, 281–95
 - template classes 9, 103–16, 263–80
- Observer pattern, concepts 298, 305–7
- ODEs *see* ordinary differential equations
- oil industry 3
- on-demand loading, COM add-ins 358
- OnConnection event, Com add-ins 357–8, 369–71
- OnDisconnection event, Com add-ins 357–8, 369–73
- one-dimensional data structures
 - see also* arrays; vectors
 - concepts 63
- one-dimensional heat equations, concepts 4–5, 179, 182–91, 226
- one-factor Black–Scholes models 2–3, 4–5, 38, 78, 91, 94, 137–8, 143, 175–81, 186–91, 192–202, 244, 355
- one-factor problems, finite difference method 2, 4–5, 38, 91, 94, 152–4, 179–224, 244
- one-step method, finite difference method 134–7
- one-to-many relationships, concepts 284, 298, 306–7
- OpenGL 49, 297, 393
- operability requirements, software quality 65, 80
- operations research 78
- operators, types 15–17, 27–8, 32, 50–1, 67–8, 78
- optimisation issues 6, 87, 213–14, 296
- options 1–9, 29–36, 47–60, 111, 168, 171, 189–201, 210–24, 239–43, 266–70, 282–5, 313, 324–7, 338–45, 374–93
 - see also* American ...; call ...; European ...; put ...
 - attributes 9, 29–30, 36, 47–60, 111, 266–70, 284, 339–40, 342, 344–5, 383–93
 - background reading 2
 - combinations 374–8
 - concepts 55–6, 282–7, 342–5, 374–82
 - definitions 342–3
 - delta 2, 5, 52–7, 111, 138, 160–1, 189, 198, 201, 210–24, 268–9, 305–6, 378–82
 - Extensible Markup Language 313, 324–7, 338–45
 - Financial products Markup Language 324–7, 343, 350
 - payoff functions 2, 182–5, 187–90, 210–11, 272, 343–5, 375–82
 - pricing examples 51–9, 111, 138, 175–81, 182, 210–24, 272, 290, 296, 305–6, 311, 383–93
 - Property pattern 47–60, 338–9, 350
 - sensitivities 47, 52–7, 111, 211–24, 268–9, 290, 383–93
 - trading strategies 374–82
 - two underlying assets 189, 239–40
 - vector-valued functions 111
 - walk-through example 212–24
- options on options
 - see also* exotic ...
 - concepts 57
- OptionStrategy class 376–8
- Oracle 49, 297
- ordinary differential equations (ODEs) 1–2, 4, 6, 13, 105, 131–71, 177–8, 190–1, 205–8, 355
 - method of lines 144, 179, 190–1
 - overview 1–2, 4, 6
 - stochastic differential equations 145, 152–4
- oscillation issues 102, 136–8, 158, 160–1, 188–9, 191, 195–201, 203–5, 208, 237, 244
- OTC *see* over-the-counter derivatives
- out barrier options, concepts 343–5
- outcomes, experiments 145–7
- output iterators 21–2
- output mechanisms 38–40, 247–52
 - see also* displays; printing ...
- over-engineered systems, software 64, 161–2
- over-the-counter derivatives (OTC) 324
- overloaded operator 49–51, 69–72, 75, 251, 388
- overview 1–6
- OWL graphics library 311
- parabolic PDEs, concepts 177–81, 182–91, 195–9, 205–11, 212–44
- ParabolicFDM class 212–24
- ParabolicPDE class 212–24
- parameter entities, Extensible Markup Language 319
- parameter sets, random processes 150–1
- parametrised factory methods 274–5
- parent elements, Extensible Markup Language 313–27, 331–45
- parent–child relationships, concepts 282–3, 313–27, 331–45
- parser, Extensible Markup Language 335–8
- partial differential equations (PDEs) 1–2, 4, 13, 78, 91, 93, 100–1, 159, 167–8, 175–81, 187–202, 212–44, 279
 - see also* Black–Scholes models; heat ...
 - Alternating Direction Implicit 225–44

- block tridiagonal matrices 100, 227–343
- concepts 1–5, 175–91, 214–41, 279
- function classes 217–18
- historical background 176, 180, 182–3
- numerical approximations 177–9, 237–44
- robust schemes 180, 203–5
- splitting methods 225–44
- types 176–7, 179, 182–3, 187–9
- partial integral differential equations (PIDEs), concepts 177
- partial summation functions, examples 43–4
- Pascal distribution
 - see also* geometric ...
 - concepts 124, 126–7
- pattern concepts, XML Stylesheet Language 322–4
- patterned matrices 36, 63, 66, 69, 92–3
- payoff, option definitions 343
- payoff functions
 - concepts 2, 182–5, 187–202, 210–11, 272, 343–5, 375–82
 - exotic options 187–91, 210–11, 343–5
 - options theory 2, 182–5, 188–90, 210–11, 272, 343–5, 374–82
 - spreads 374–82
- payoffGraph function 376–8
- PDEs *see* partial differential equations
- pdf *see* probability distribution function
- Peaceman, D. 227–9, 237, 239
- PIDEs *see* partial integral differential equations
- piecewise smooth boundary conditions, concepts 201
- plain options
 - see also* options
 - definitions 342–3
- pmf *see* probability mass function
- POC *see* proof-of-concept tests
- Point class 47–8, 299, 332–4, 336–8
- Poisson distribution, concepts 123–5, 127, 147
- Polar–Marsaglia method 148–50, 153
- policy concepts 15
- Polyline class 299, 333–4
- polymorphism paradigms 103–5, 112, 117–19, 128, 169, 217–19, 263, 270–1, 274–9
- pop functions, stacks 45–6
- portability characteristics, software quality 3, 49, 65, 219–20, 297, 341, 391–3
- portfolios
 - C++ 374–82
 - concepts 38, 88, 282, 305–6, 324–7, 374–82
 - hedging 203, 305–7, 333, 374–82
 - hierarchical examples 374–5
 - theory 88, 374–82
- PortfolioThing class 376–8
- POSA 281, 307
- positive-definite matrices, concepts 101–2, 167
- positivity results, concepts 132–3, 136
- power call options 210–11, 221
- predictor–corrector methods, concepts 138–9, 142–3, 153, 191, 232, 257–8
- presentation activities, applications 247–52, 297, 347–55, 375, 380–2
- pricing, background 1–6, 111, 175–81, 182, 210–24, 272, 290, 296, 305–6, 311, 383–93
- printing functions 38–40, 69–70, 75–6, 78, 247–60, 267, 347–55, 380–2
 - see also* displays
- priority queues, concepts 21, 45–6
- probability, concepts 17–18, 117–28, 145–54
- probability distribution function (pdf)
 - see also* continuous ...; discrete ...
 - concepts 17–18, 117–24, 147–54, 249–60, 277–9, 351–5
- probability mass function (pmf), concepts 146–7
- procedural functions 16
- ProductFactory class 272–5
- Projected SOR method (PSOR), iterative solutions 93–4, 101, 164, 168
- proof-of-concept tests (POC) 191
- property lists, concepts 52–7
- Property pattern 6, 47–60, 300–5, 314, 338–45, 350–5, 356
 - advantages 49–50
 - concepts 6, 47–60, 314, 338–9, 350
 - examples 50–60
 - interfaces 50–1
- property sets, concepts 47–60, 84, 86, 115, 258–60, 300–5, 338–45, 350–5
- Prototype pattern, concepts 262–3, 266–7, 270–2, 340, 350–1
- pseudo-random numbers 23, 147–8
- PSOR *see* Projected SOR method
- publishers 298, 306
- pure virtual functions
 - see also* polymorphism
 - concepts 117–19, 128, 263, 270–1, 303, 376–8
- push functions, stacks 45–6, 107
- put options 56–9, 290, 343–5, 375–8
- quadratic form 83–5, 101–2
- quadrature techniques 290
- quality issues, software 3, 11, 36, 49, 64–5, 79–80, 219–20, 261–308, 340–1, 391–3
- quantos 327
- ‘Quants’ 2
 - see also* financial engineering
- queues, concepts 21, 45–6
- quotes, Extensible Markup Language 314–15
- Rachford, H.H. 228–9
- rand function 33, 148
- Randall, C. 91, 160, 168, 177, 197, 227, 236, 237, 392
- random access iterators, background 22, 25–6, 30–3
- random interest rates, convertible bonds 239
- random number generators, background 21, 23, 33–5, 106–11, 147–54
- random processes, concepts 150–4, 239
- random variables, concepts 145–54, 239
- ranges
 - Excel 346–7, 364–7

- ranges (*continued*)
 mapping concepts 109–11, 140–3
 random variables 145–7
 Rayleigh quotient 83–4
 rbegin, reverse iterators 31–3
 readers
 Excel 349–55
 Extensible Markup Language 303–5,
 338–45
 real option theory 124
 real-time embedded systems, Property pattern
 49
 real-valued functions, concepts 63, 103,
 110–11, 115–16, 214–15, 290–4
 realisation concepts, random processes 150–1
 rebates 344–5
 recipe-type schemes 180
 rectangular distribution *see* uniform ...
 recursion 36–7, 126, 148, 284–5, 324–5
 recursive classes
 see also unary associations
 concepts 36–7, 126, 284–5
 refactoring opportunities 293–4
 references, Extensible Markup Language
 318–19, 330–1
 registration activities, applications 247–9,
 258–9, 347–8, 352–5, 368–9
 relational databases 53, 59, 261–2
 relations, functions 109–11
 relaxation parameters, iterative solutions 93,
 164, 166–8
 reliability characteristics, software quality 3, 11,
 36, 49, 65, 79–80, 269–70, 391–2
 remove 21–3, 54, 372
 RemoveMenuItem function, COM add-ins
 372–3
 rend, reverse iterators 31–3
 replace concepts 21–3, 69
 repositories, Singleton pattern 265
 resource efficiency, concepts 65, 79, 296
 response times 79
 reusability benefits, template classes 9, 17, 20,
 36, 47, 78, 82, 85, 112, 117, 140–1, 155,
 190, 234–6, 248–60, 393
 reverse iterators 21–2, 30–3
 Reynolds number, concepts 197
 rho 52–7, 111
 Riablenki, V.S. 230
 Richardson extrapolation, concepts 139–40,
 153–4, 201
 Richardson's Deferred Approach to the Limit
 211
 Richtmyer, R.D. 180, 187
 risk management, Financial products Markup
 Language 324–7
 Robin boundary conditions, concepts 156–7,
 159, 176, 195–9, 232–4
 robust schemes 180, 203–5
 Rogue Wave 20
 role concepts, Unified Modelling Language
 283–4, 298
*r*th-moment functions, concepts 86–7
 RTTI *see* run-time type identification
 rules-of-thumb 6
 Rumbaugh, J. 265, 285
 run-time errors 80, 88–90
 run-time switching 112, 117
 run-time type identification (RTTI) 59
 Runge–Kutta scheme 191
 Samarski, A.A. 205
 sample points, concepts 145–7
 sample space, concepts 145–7, 150–1, 177–8,
 205–8, 237–8
 scalar functions, concepts 63, 100–1, 103–5,
 110–11, 112–15, 144, 382
 scalar linear problems, initial value problems
 132–8, 140–4, 151–4, 255–8, 382
 'scalar' *LU* decomposition algorithm 100–1
 ScalarIVP class 140–4, 256–8, 382
 ScalarIVPSolver class 141–4, 257–8
 scale factors, concepts 122
 scale parameter, concepts 120
 scatter diagrams, Excel 348, 351–5
 Schema, Extensible Markup Language 328–34,
 373
 Schuss, Z. 152
 SDEs *see* stochastic differential equations
 search issues 21–3, 27, 33–5
 selectors, definition 385
 self-adjoint equations, concepts 199–200
 self-awareness issues 48, 59
 self-test, C++ 383–4
 semi-discretisation concepts 190–1, 205–8
 semi-norms, concepts 80–3
 sensitivities
 see also delta; gamma
 options 47, 52–7, 111, 211–24, 268–9, 290,
 383–93
 separation mechanisms, Bridge pattern 290–4
 sequence containers, background 21, 23–7,
 33–5, 37, 43–4
 Sequence specifier, Extensible Markup Language
 316–17, 330
 serialisation concepts 59
 server code, concepts 89–90
 sets
 concepts 9, 21, 27–30, 37, 40–3, 48–9,
 52–7, 66, 73, 111
 examples 27–30, 37, 40–3, 52–7
 parameter sets 150–1
 property sets 47–60, 84, 86, 115, 258–60,
 300–5, 338–45, 350–5
 Seydel, R. 1, 150
 SGML *see* Standard Generalised Markup
 Language
 Shaidurov, V.V. 140
 Shape class 299, 333–4
 ShapeComposite class 270–1, 333–4
 short positions, definition 343
 shout options 189
 simple data types, XML Schema 330–1
 SimplePropertySet class 47–60, 84, 86, 115,
 258–9, 266–70, 302–5, 338–45, 350–5
 single barrier call options 210–11
 Singleton pattern, concepts 251–2, 262–70

- singular perturbation problems, concepts 137–8, 160, 197
- skewness functions 78, 86–7
- Skonnard, A. 328
- slists, sequence containers 21, 24
- Smith, G.D. 198
- 'snapshots', options 55–6
- SNOBOL4 language 74
- software 1–4, 20, 64–6, 79–80, 175–81, 212–24, 259–308, 356, 391–3
see also applications; code
 black boxes 64, 91, 175–6, 235, 247, 262–3, 289
 characteristics levels 3–4, 11, 49, 65, 79–80, 87, 219–20, 296–7, 356, 391–3
 design patterns 1–2, 5, 21, 59–60, 64, 103–5, 112–16, 212, 217, 247–8, 259–308
 documentation needs 2, 64, 265–6, 272–6, 279, 285, 297, 338–9, 356
 generic software 9–19, 20, 36, 112
 ISO 9126 quality characteristics 3, 80, 180, 391–2
 layering concepts 3–4, 65–6
 optimisation issues 6, 87, 213–14, 296
 over-engineered systems 64, 161–2
 quality issues 3, 11, 36, 49, 64–5, 79–80, 219–20, 261–308, 340–1, 356, 391–3
 requirements 3–4, 64–5, 79–80, 213–14, 391–3
- SOR *see* Successive Overrelaxation
- sort, mutating algorithms 21–3, 33–5
- source code *see* code
- space issues, concepts 145–7, 150–1, 177–8, 205–8, 237–8
- sparse matrices, concepts 36–8, 65, 66, 92–4, 164–8, 290
- special elements, Extensible Markup Language 316–17
- specialisations 122–3, 265, 286, 301–2
see also generalisation
- spectral and pseudo-spectral approximation methods, concepts 178
- Spiegel, M. 87, 117
- spikes, energy prices 124
- splitting methods
 basics 230–2
 concepts 5–6, 94, 168, 225–44
 critique 228, 230, 236, 241
 historical background 227–8, 241
- spreads, concepts 282, 287–8, 374–82
- spreadsheets 5, 49, 59, 65, 74, 76, 162, 224
- spurious oscillation problems 102, 136–8, 160–1, 188–9, 191, 195–201, 203–5, 208, 237, 244
- SQL Server 49, 297
- square root 82–4, 152–4
- stability issues, finite difference method 136–7, 198–9, 203–11, 227, 229, 237, 244
- stacks, concepts 21, 45–6, 262–3
- standard difference schemes 158
- Standard Generalised Markup Language (SGML) 312
- Standard Template Library (STL)
see also algorithms; containers; function objects; iterators
 adaptor 42–3
 advantages 20–46, 65, 76
 components 20–35
 concepts 3–4, 5–6, 9–10, 19, 20–46, 53, 65, 76, 103, 106–11, 393, 396
 examples 20, 24–35, 38–46
 financial engineering 35, 36–46, 108–9
 portability benefits 20, 49, 65
 syntax 20
 uses 20–46, 76, 103, 106–11, 393
- standard-deviation function, concepts 86, 88, 121–2
- State pattern 297
- states, random processes 150–1
- static member functions 251–2, 264, 267, 366–7, 369–73
- stationary independent increments, random processes 151
- stationary processes, random processes 150–1
- statistics
see also continuous probability ...; discrete probability ...; probability ...
 concepts 4, 6, 17, 36, 78, 85–8, 117–28, 145–7
- StatisticsMechanisms* package 147
- Steele, J. 152
- step size, concepts 133–4, 183–5
- step structures 189–90
- stiff problems, concepts 137–8
- STL *see* Standard Template Library
- stochastic differential equations (SDEs) 1–6, 13, 33, 105, 131–71, 239, 355
 concepts 1–2, 4, 5–6, 139, 144, 145–54, 239
 definition 151–3
 finite difference method 145, 151–4, 239
 noise 145
 ordinary differential equations 145, 152–4
 predictor–corrector methods 139, 153, 257–8
 theoretical background 151–4
- stochastic processes *see* random ...
- stock indexes 283–4
- stock prices 151–2, 186, 212–24
- storage classes, arrays 66–8
- Stoyan, G. 205
- straddles 333, 375–8
- Strang, G. 191, 240
- strangles 333, 374–8
- straps 375
- strategies, trading strategies 374–82
- Strategy pattern, concepts 218, 297–8
- strike price, definition 343
- strings, examples 10, 19, 27–30, 41–3, 50, 115, 314–15, 329–30, 339, 344–5
- strips 374–8
- Stroustrup, B. 383
- structural patterns 5, 260, 281–95, 314, 396
see also Bridge ...; Composite ...; Façade ...; Whole–Part ...

- structural patterns (*continued*)
 - concepts 5, 260, 281–95, 314, 396
 - examples 281–95
- structured products, Financial products Markup Language 324–7
- stylesheets, Extensible Stylesheet Language Transformation 320–3
- subclasses, concepts 286
- subjects 306–7
- subscribers 298, 306
- subset operations, sets 40–3
- Successive Overrelaxation (SOR), iterative solutions 93, 164, 166–8
- suitability requirements, libraries 64
- sum-of-reciprocals function, concepts 85, 88–9
- sum-of-the-squares function 82, 85
- summation examples 43–4
- sums 43–4, 66, 72, 78, 85, 88–9
- Sun, Y. 239
- superclasses, concepts 286
- supershare options 189–90
- Sutherland, I. 272
- swap
 - function 16–17, 27–8, 41–3
 - mutating algorithms 22–3
- swaps 284–5, 324–7
- swaptions 324
- symmetric-difference operations, sets 40–3
- syntax 19–20, 313–19, 328–45

- tags, Extensible Markup Language 312, 320–4, 344
- TanhRule class 291–4
- Tavella, D. 91, 160, 168, 177, 197, 227, 236, 237, 392
- Taylor's expansion 134
- template classes
 - activation 11–16
 - code files 11–16
 - concepts 4–6, 9–35, 47–60, 103, 111–16, 117–28, 140–4, 263–70, 300–4, 338–45, 366–7, 393
 - definition 11–15
 - European options 9–10, 266–70
 - meta-class aspects 11, 19
 - object links 9, 103–16, 263–80
 - reusability benefits 9, 17, 20, 36, 47, 78, 82, 85, 112, 117, 140–1, 155, 190, 234–6, 248–60, 393
 - shorthand notation 114–15
- templates 4, 5–6, 9–60, 103–5, 117, 124, 262, 322–4
 - see also* function templates; template classes
 - inheritors 5, 60, 103–5, 117, 124, 262
 - types 11–15
 - XML Stylesheet Language 322–4
- temporal data types 3, 36, 210–11, 328–9, 343, 394–6
- term structure of interest rates 152, 241
- terminal conditions, Black–Scholes models 188–9, 192–202, 240
- testability characteristics, software quality 3, 11, 82–3

- text interface, Document Object Model 335
- theta 52–7, 111, 185
- Thomas, J.W. 159, 187, 199, 233–6, 243
- three-dimensional heat equations, concepts 228–9, 231–2, 235
- three-dimensional matrices, concepts 38, 228
- time class 36, 395
- time efficiency, concepts 65, 296
- time-dependent volatilities 210–11
- Tools for Computational Finance* (Seydel) 1
- Tools/Add-ins menu, Excel 357–8, 361, 369, 372
- Tools/Customise menu, Excel 357–8
- top-down approaches 235–6
- Topper, J. 210–11, 222, 239–41
- TPBVPs *see* two-point boundary value problems
- trades, concepts 325–7
- trading strategies, C++ 374–82
- traditional centred-difference schemes, two-point boundary value problems 157–9, 162, 179–80, 203–4, 208, 241–3
- trapezoidal difference scheme, concepts 134–5, 139, 142–3
- tree data structures 36
- triangular matrices, concepts 37–8, 91
- tridiagonal matrices, concepts 36, 37–8, 66, 93–102, 155–63, 164, 184–5, 219, 227–43
- trinomial method, concepts 36, 178, 182, 186–87, 383
- try, throw and catch blocks, exception handling 89–90
- two underlying assets, options 189, 239–40
- two-dimensional data structures
 - see also* matrices
 - concepts 63
- two-dimensional heat equations
 - concepts 4–5, 179, 226–36
 - examples 226–36
- two-factor Black–Scholes models 2–3, 4–5, 38, 78, 91, 94, 105, 137–8, 143, 175–81, 195, 228, 237–44, 355
- two-factor problems, finite difference method 2, 4–5, 38, 91, 94, 105, 179–80, 195, 225–44
- two-point boundary value problems (TPBVPs)
 - approximations 158–63, 195–7
 - Black–Scholes models 158–9, 182–91
 - concepts 4, 5, 97–9, 155–63, 170–1, 176–80, 182–91, 203–11
 - derivatives approximations 160–1, 195–7
 - description 155–7
 - design issues 161–3
 - exponentially fitted schemes 160, 179–80, 203–11
 - extrapolation 158
 - finite difference method 155–63, 182–91, 203–22
 - traditional centred-difference schemes 157–9, 162, 179–80, 203–4, 208, 241–3
- TwoVarDFFunction class 112–14, 215–24

- UI *see* User Interface
- UML *see* Unified Modelling Language

- unary associations, concepts 284–5
- unary function, function objects 23, 106–11
- understandability requirements, software quality 80, 87
- Unified Modelling Language (UML) 2, 52, 57, 64–9, 162–3, 175–6, 212–13, 220, 247–8, 265–6, 272–4, 281–97, 302, 321, 325–6, 332–3, 338, 346–7, 375
- uniform distribution, concepts 119–21, 127, 147, 261–3
- Uniform Resource Locators (URLs) 321–2, 329
- uniformly convergent difference schemes, concepts 138, 204–5, 207–8
- uniformly stable schemes, concepts 187, 204–5
- union operations, sets 40–3
- univariate functions *see* scalar functions
- up-and-in call options 343–5
- up-and-in put options 343–5
- up-and-out call options 343–5
- up-and-out put options 343–5
- upper triangular matrices *see also* LU decomposition concepts 93–102, 165
- upwind schemes, concepts 209–10, 239, 242–3
- URLs *see* Uniform Resource Locators
- usability characteristics, software quality 3, 11, 36, 49, 64, 80, 87, 296–7, 356, 392, 393
- User Interface (UI) 311, 343
- user-defined complex/simple data types, XML Schema 330–4
- utility functions, printing 38–40, 69–70, 75–6, 78
- vanilla options *see also* options definitions 342–3
- Varga, R.S. 102, 165–6, 207
- variance 86, 88, 117–28, 239 function concepts 86, 88, 117–28 of return 88
- VARIANT, automation add-ins 364–7
- variational behavioural patterns, concepts 297–301
- Vasicek model 152
- VBA *see* Visual Basic for Applications
- VBScript 335
- vector functions, concepts 103, 110–11, 115–16, 143–4
- vector space, concepts 72–3, 109–11
- vector-valued functions, concepts 103, 110–11, 116
- vectors *see also* arrays boundary errors 80, 89, 97–9, 116 concepts 3–6, 9, 21, 24–7, 33–40, 43–4, 46, 63–6, 72–4, 78–102, 103, 110–11, 131, 155–63, 212–24, 247–60, 347–55, 366–7, 376–82 definition 9, 63, 72 examples 9, 25–7, 33–5, 37, 43–4, 72–4, 78, 80–4, 94–102, 212–24, 247–60, 348–55, 366–7, 376–82 Excel 247–60, 348–55, 366–7, 380–2 indexing operator 25–7, 74, 81–4, 114–16, 169–71 inequalities 87, 102, 108, 164, 168, 171, 207–8 matrices 73–4, 83–4, 92–3 nested vectors 101, 223–4 norms 78, 80–4, 97 printing functions 38–40, 72, 78, 247–60, 347–55, 380–2 size flexibility 66 useful algorithms 43–4
- vega 52–7, 111
- virtual constructors *see also* Factory pattern concepts 272–5
- virtual functions *see also* polymorphism concepts 117–19, 128, 169, 217–19, 263, 270–1, 274–9, 292–3, 303, 351, 376–8
- Visitor pattern, concepts 47, 294, 297, 298, 301–5, 328, 338–45
- Visual Basic 64, 272, 327, 335, 356–8
- Visual Basic for Applications (VBA) 356–8
- Visual C++ 367
- Visual Studio 357, 358, 359
- void types 14–17, 31–9, 51, 58–9, 69, 75–6, 96–7, 104–5, 141–4, 169–70, 217–19, 248–52, 256, 265, 267, 292, 300–4, 336–7, 341–2, 350–4, 370, 373
- volatility *see also* historical ... concepts 2, 88, 111, 177, 180, 196–7, 208–11, 220–4, 268–9, 284, 385–93
- von Neumann, J. 156–7, 159, 176, 180, 197–9, 206, 227
- VRML 297
- W3C *see* World Wide Web Consortium
- weakly stable difference scheme, concepts 197, 208–9, 239
- Weibull distribution 119
- weighted-average-mean function, concepts 85
- white boxes 289
- Whole–Part patterns *see also* aggregation relationships concepts 282–3, 286–8, 375
- wide-sense stationary processes, random processes 150–1
- Wiener process, concepts 151–3, 290
- Wilmott, P. 2, 91, 93, 157, 164, 168, 182, 191, 378
- Windows 20, 65, 148–50, 335
- Word 297
- work arrays, concepts 218–19
- workbooks, Excel 346–57, 373
- worksheets, Excel 346–55, 358, 362–7
- workspace containers, efficiency enhancers 79
- World Wide Web Consortium (W3C) 311–12, 328
- writers Excel 349–55 Extensible Markup Language 303–5, 338–45

- XLA, add-ins 357
- XLL, add-ins 357
- XML *see* Extensible Markup Language
- XML data types 329
- XML Stylesheet Language (XSL), concepts 320–4
- XMLReader class 303–5, 340–5
- XMLWriter class 303–5, 340–5
- XSL *see* XML Stylesheet Language
- XSLT *see* Extensible Stylesheet Language Transformation
- Yanenko, N.N. 228, 231–2
- zero
 - divides 80, 89
 - matrix elements 37–8, 92–3
 - volatilities 197, 208–9