Index

: (statement separator), 28
label delimiter, 28
# (cast), 33, 236, 237, 238
file functions, 143–145
:: (scope qualifier), 58, 621
in C++, 58
& (concatenation), 241
+ (concatenation), 241
_ (continuation line character), 44
in implementing events, 121
in interface implementation, 84–85
and naming conventions. See Names, naming conventions
' See Comments
. (name qualifier). See Names, qualified names and UDTs, 244
= (assignment operator), 244
^ (power operator), 237
/ (division), 236, 237, 238
\ (integer division), 238

0.5 strong Itô-Taylor discretization. See Itô-Taylor discretization schemes, 0.5 strong
1.0 strong Itô-Taylor discretization. See Itô-Taylor discretization schemes, 1.0 strong
1.5 strong Itô-Taylor discretization. See Itô-Taylor discretization schemes, 1.5 strong
2.0 weak Itô-Taylor discretization. See Itô-Taylor discretization schemes, 2.0 weak
2-dimensional arrays. See Arrays

Accumulator account numeraire, 325, 424, 581
Accuracy. See Convergence
Acklam. See Index to library functions,

  normal_inverse()
ActiveX controls, buttons, 27, 571–572
Adapter pattern, 559, 560–561, 613
  See also: Design patterns
Aggregation, of objects, 613

American options. See also: Bermudan options
American put. See American put
American straddle, 450, 462, 475
  See also: Bermudan option valuation
American put, 451
  See also: Option valuation, American put
on a knock-in barrier option, 489
on a knock-out barrier option, 489
  See also: Index to implementations, American put
American straddle, 450, 462, 475
Antithetic variates, (Chapter 17), 283–290
average rate options (GBM), effectiveness with,
  288–289, 379–380, 381–382, 382–383, 384
and bias, 285
correlation, 284, 289
and the decorator pattern, 285, 304
with the Heston model, 290
and long-step Monte Carlo, 380, 382
and moment matching, 291
and slice-wise evolution, 285
and the uniform distribution, 284
with Wiener processes, 284
  See also: Variance reduction
API. See Application programming interface
Application methods
  Application.Max(), 30, 243, 252
  Application.NormSDist(), 17
  Application.NormSInv(), 22, 243, 246
  Application.Pi(), 38, 185
  Application.Run(), 83–84
Application object, 242
Application programming interface (API), 56, 81, 545
  GetTickCount(), 88
Arbitrage, 10
  and violation of put-call parity, 393
Array expansion idiom, 147, 166, 174
Arrays
  2-dimensional arrays, 268–270
    as separate 1-dimensional arrays, 268
Arrays (continued)
  striding, 268–270, 617
  performance, 273–274
  array utility functions, 536
    FindIndex(), 532
  passing ByRef, 107
  and DLLs, 545–547
  dynamic array, 266
  and file I/O, 146–148
  holistic Monte Carlo, 296
  and level 0 lattice implementation, 573
  and level 1 Crank-Nicolson implementation, 586
  performance, 266–267
  path-wise Monte Carlo, 157, 257
  performance
    as a container, 273–277
    higher dimensional arrays, 245, 296
    looping, 245, 275
    and pre-extraction, 272, 273
  re-sizing. See Array expansion idiom
    See also: Memory manager
  as return types, 107
  slice-wise Monte Carlo, 107–116, 255
  performance, 272–273
  static array, 261
  Variant array. See Variant arrays
  and wrapper objects, 263, 261–267, 270–271
Assignment
  See also: Copy assignment
  arrays, 245, 272
  casting, 244
  performance, 238, 241, 243, 244, 245
  references, 556
  UDTs, 244, 554
Association, of objects, 613
Associative container, 222, 262, 614
At the money (ATM) options. See also: Out of the
  money options; In the money options
  and antithetic variates, 288
  and Heston examples, 435
ATM. See At the money options
Average rate options
  See also: Option valuation, average rate options
  binary chop evolution, 13
  continuously averaged geometrically averaged
    explicit solution, European call, 18
  discretely reset arithmetically averaged, 117
    See also: Index to implementations, Arithmetic
      average rate option (GBM)
    See also: Index to implementations, Arithmetic
      average rate option (Heston)
  See also: Option valuation, average rate options
  benchmark values, 281
  and valuation
    in a book with control variates, 333, 377
    in a book with importance sampling, 352
    with control variates, 324–325, 338–342
    in Heston, 330
  discretely reset geometrically averaged, 117
    as a control variate, 317, 324–325
  explicit solutions, European call, 117, 523–526
  and LD sampling, 315
  library functions, 538
  See also: Index to implementations, Geometric
    average rate option
  fixed strike, 117
  state information, 107

B-splines, 464, 465–466, 479–482
  See also: Basis functions
Backwards evolution. See Evolution, direction,
  backwards
Bad practice, 58, 233, 243, 517
  See also: Evil
Barrier options
  See also: Option valuation, barrier options
  barrier bond options, 398
  and bias, 102, 255, 396–397
  double barrier options, 95, 102, 255–258
  flexible barrier option, 488
  general formulation, 450–452
  mimicking barrier options. See Mimicking
    barrier options
  and plain Monte Carlo, 102
  state, 97, 107, 113
  vanilla barrier options
    knock-in, 7, 97, 98, 113, 258, 316
    knock-out, 16, 95–97, 98, 113, 290, 316, 398
  See also: Index to implementations,
    Barrier options
Base class, 613
  in C++, 86, 558
  VBA version, 193
Basis functions, in LSLS MC, 463–470
  See also: Continuation values; LSLS method
B-splines, 464, 465–466, 479–482
Chebyshev polynomials, 1st and 2nd kind, 464, 475
  comparison between, 466–468
Gegenbauer polynomials, 464
Laguerre polynomials, 464–465, 479–483
Legendre polynomials, 464–465, 466, 479–482
natural basis functions, 464, 465, 479–482
  and OLS, 468
power functions, 464–465, 466–468, 479–482
  scaled basis functions, 465
  and SVD, 468
Bermudan-2 put, 528–529
Bermudan-early control variate, 493
Bermudan option valuation, by Monte Carlo
  bundling algorithms, 457
  continuation values. See Continuation values
early exercise boundary. See Early exercise boundary
error
foresight error, 392, 452, 459
sub-optimal exercise error, 449, 454–455, 471, 474, 487
LSLS method. See LSLS method
stochastic mesh method, 457
Bermudan options
See also: Bermudan option valuation
Bermudan butterfly, 462
Bermudan compound option, 209
Bermudan-early put, 493
as a control variate, 497–498, 505
Bermudan put option. See Bermudan put option
Bermudan straddle, 462
Bermudan swaption, 449, 450, 465
Bermudan put control variates, 492
Bermudan put option
See also: Option valuation, Bermudan puts
Bermudan-2 put, 528–529
convergence to American put, 452–453
See also: Compound option
See also: Index to implementations, Bermudan puts
Bermudan-T1 control variate, 493
Bermudan terminal control variate, 493
Beta distribution, 11, 233, 536
Bias, in Monte Carlo. See also: Discretization, convergence
and antithetic variates, 285
and barrier option valuation, 7, 102
bias reduction methods. See Bias reduction methods
in bond option valuation, CIR, 390–392
in the drift, 391, 393
and predictor-corrector methods, 411–412
and early exercise
in American put valuation, 452, 453, 455, 503
high-bias estimator, 457
See also: Bias, foresight bias
low bias estimator, 457, 487
in LSLS method, 473–474
and basis functions, 468, 471, 479–482, 487, 509
in exponents, 424, 426–427
foresight bias, 392, 452, 459
in Heston, 436
and LD sampling, 314–315
measure of, 474
sources of bias, 389–390
Bias reduction methods
for barrier options, 396–397
martingale correction, 393
martingale simulation, 393
moment matching, 394
Bidirectional association, of objects, 613
Binary chop evolution. See Evolution, direction, binary chop
Binary files, 137
Binding
early bound, 139
late bound, 139, 244, 262, 263
Binomial lattice, 210, 567
Bisection method, root finding, 603
in exercises, 38, 79, 104
Bisection(), 604
Bit-arithmetic, 310
Bi-variate normal distribution, 529
Black-Scholes framework, 3, 7, 25, 324
See also: Black-Scholes model
Black-Scholes model
See also: Black-Scholes framework
See also: Black-Scholes PDE
See also: Geometric Brownian motion
See also: Implied volatility
See also: Simulation, geometric Brownian motion
Black-Scholes call control variate, 338–339
Black-Scholes formula, 25
explicit solutions
European call, 25
discretely reset geometrically averaged average rate option, 523–525
library implementation procedures, 538–540
quadratic payoff option, 526–528
2-reset Bermudan option, 528–529
Monte Carlo applications, benchmarking, 254–255
parameter validation, 40
Black-Scholes PDE, 581
implementation, 585–590, 598–601
numerical assessment, 593–595, 601–602
Block comments, 26, 571
Blow up, discretizations, 391, 420
Bodge, 613
and overflow, 420
and rectification functions, 407, 416
CIR, 410, 417, 420
Heston, 432, 446
and weak design, 155
Bond options
barrier bond options, 398
in CIR, 389–391
and early exercise, 460–461
in Fong and Vasicek, 325–327
in Vasicek, 326, 327–328
Book of options, 153–176
benefits of scale, 174
and control variates, 333, 377
and importance sampling, 352, 368
and knock-out options, 16, 258
Boolean
and IIF(), 239
in masks, 335, 494
Boolean (continued)
as state, 97, 107, 111–113, 256
and the Static initialization idiom, 47, 247
to toggle functionality, 361
Borel measure, 4
Bracketing interval, 603
Branching probabilities. See Lattice methods, branching probabilities
Bridge density, 292
bridge density of a Wiener process, 293
of the maximum from a bridge, 397
sampling from, 292
Bridge distribution, GBM, sampling from the maximum, 397
Broadie and Kaya, 432, 446
Brownian bridge, 118, 293
and evolution, 294–297
and the LSLS method, 449, 482
and stratified sampling, 292–293
See also: Bridge density
Bullet, bite, 16
Bundling algorithms, Bermudan option valuation, 457
Buttons, 19–20, 25–26, 34
ActiveX controls, 27, 571–572
and invocation chain, 101
See also: Calling procedure
ByRef, 555
and DLLs, 245–247
and error codes, 550
passing, examples
arrays, 107, 285, 376, 479, 494, 536
Doubles, 45
 Enums, 181
UDTs, 146
ByVal, 555
and DLLs, 545–546
performance, 242–243
C++, 39, 40, 47, 58
call(), 216
 comparison with VBA, xv, 36, 55, 556, 557, 565
delete, 58
DevC++, 36, 241, 628
DLLs, 545–547, 628
functors, 71, 614
inheritance, 86, 558
initialization, 28, 554
meta-class data, 559
objects in, 55, 56, 58, 60, 549, 555, 556
operator(), 71, 614
operator overloading, 556, 557, 615
performance, xv, 36, 235, 241
polymorphic factory, 179, 183, 216
polymorphism in, 558
std::cout(), 138
std::exp(), 241
streams, 137, 138
swap(), 217
this, 66
Calibration, models, 10, 12, 18, 610
exact, 392
moment matching, 394–396
Call, 621
Call-back, 559, 562
call-back procedure, 100
See Registration and call-back
Calling procedure, 19, 119, 605
See also: Invoker; main()
Casts. See also: Type conversion
between interfaces, 195–196
and file input, 148
and validation, 185, 201
Catch and re-throw idiom, 44, 76, 531
Catching errors. See Error handling, trapping errors
Cells(). See also: Front-end
clearing contents, 45, 72, 130, 585
and range names, 29
validating input from, 28, 533
Central moments
of log-normal distribution, 418
of normal distribution, 23
of uniform distribution, 22
CEV (Constant elasticity of variance) model, 102–103, 114
Channels. See also: Streams, I/O
and the Environment object, 172
error channel, 19, 21
and the factory, 198, 212, 226
file number for, 144
I/O, 19, 78, 122, 137, 174
costs, 174–176
input channel, 20, 78
monitor channel, 130
output channel, 78, 127, 130
sequential, 137
Chebyshev polynomials, 1st and 2nd kind, 464, 475
See also: Basis functions
Chirayukool, Pokpong, xvii, 447
CIR. See Cox, Ingersoll and Ross model
Class modules
and events, 119–121
and factory registration, 188, 197
and fixed length Strings, 263
and interfaces, 84–86
and object definition, 56, 59–63, 556
Private scope, 58
and re-use, 57
and state, 40
structure of, 59
and the VBComponents object, 180–183
See also: Index to implementations, Utility code modules
Class_Initialize(), 59–60, 120, 557, 613
and encapsulation, 555
and initialization, 72
Class_Terminate(), 59–60, 120, 557, 614
Cleanliness
  clean recompile, 190, 518–521
code, 57
interface mechanism, 102
and re-factoring, 616
Client, 613
  client code, 35, 56, 62, 81, 557
  communication with, 33, 47, 91, 166, 169, 214, 244
  contract with, 613
code, 57
and re-factoring, 616
Client code, 35, 56, 62, 81, 557
clone(), 216
Clustering
  LD sampling, 307
  stratified sampling, 292, 297–298
CN. See Crank-Nicolson
Code modules. See Standard modules
CodeModule object, 180–183
Cody. See Index to library functions, Cody_erf()
Cohesion, 39, 613
Collection object, 262–263, 613
  and containers, 270, 275
  and FileSystemObject, 139–140
  performance, 274, 275–276
    looping with For-Each, 263, 275
    looping on keys, 275
  and states, 276
Comma-delimited files, 144
Comments
  block comments, 26, 571
  and CodeModuleProperty, 181–182, 183
  in-line comments, 622
toggling, 249, 256
  and tricky code, 248
Complex numbers,
  the complex number exercise, 79
  and Heston, 329
  library object, 540
Compiler, See VBA compiler
Compile-time errors, 40, 42, 194, 223, 244
Composited object, 72, 76, 613
Composition, 559, 613
  and the adapter pattern, 560
  composited object, 72, 76, 613
  and the decorator pattern, 124, 562, 614
  and the façade pattern, 560
  and the factory, 163, 127, 218
  and polymorphism, 153
See also: Design patterns
Compound option, 483–484, 529
See also: Option valuation, compound options
Concept code, 565, 613
Conforming object, 84–86, 557–558
  factory creation. See Instantiation
  in the level 4 plain Monte Carlo, 88–100
  and meta-class data, 558–559
See also: Meta-classes
nomenclature, 85
over-ride, 85
See also: Derived objects; Interfaces
Console, 164
Const
  and arithmetic operations, 236
  Const expression, 48, 179
  data members, 247
  honte, 245
  and initialization, 554
  and loop bounds, 245
  and magic numbers, 615
  performance, 236, 245, 247
  and the Static initialisation idiom, 48, 254, 257
  Const variables. See Const
  Constant elasticity of variance. See CEV
  Constant GBM IS density, 350–351
See also: Importance sampling
Constructors, 55, 60–61, 613
  and Class_Initialize(), 72, 557
  conversion constructor, 55, 554, 556, 614
  copy constructors, 55, 553, 614
  default constructor, 60, 179, 553, 554, 556
  and events, 120
  and the factory
    embryonic factory, 123, 130
    polymorphic factory, 198–200
    semi-polymorphic factory, 212, 214, 218
  and initializing, 310, 335
  inputting from, 69, 92
  limitations of in VBA, 60, 69
  and outputting, 72, 141
  and POD-like objects, 263
  RAI. See RAI idiom
  and reference counting, 61–63
  and registration, 183, 184
  and side-effects, 200
  and the singleton pattern 563
Containers. See also: VBA containers
  associative container, 222, 262, 614
  queue, 263
  set, 222, 263
Continuation lines, 44
Continuation values
basis functions, in LSLS MC. See Basis functions and control variates, compound option benchmark, 483
and control variates, 498–505
European put control variate, 498
implementation, 499
performance, 499–503
in LSLS MC, 458
notation, 450
slice of, 477
Continuously reset geometrically averaged average rate options. See Average rate options
Contract, option, 10
Contract, programming, 40, 91, 223, 613
Control statements
Do-Exit-Loop, performance, 240–241
Do-Until. See Do-Until
Do-While. See Do-While
End, 493
Exit, 42, 256, 495
For-Each. See For-Each
For-Next. See For-Next
GoTo. See GoTo
If-Then. See If-Then
If-Then-Else. See If-Then
On Error, 42–43, 50, 517
Resume, 42, 44, 50
Resume Next, 42, 517
Select-Case. See Select-Case
While-Wend, 240
Control variates, (Chapters 20 and 21), 317–332, 333–344
asset control variates, 320–321
auxiliary model control variates, 325–330
and Fong and Vasichek, 325–328
and Heston, 328–330
average rate options (GBM), effectiveness with
See Option valuation, average rate options, control variates
OTM options, 339, 342
bank of, 333, 335
Bermudan put. See LSLS method, using control variates
using rollback control variates, 499–505
using valuation control variates, 497
call control variate, 338
combined with importance sampling, 371–372
correlation. See Correlation, and control variates
delta control variates, 321–322
and Heston, 327, 330
with Heston. See Heston model, variance reduction, control variates
payoff matching control variates, 323–324
quadratic option, effectiveness with. See Option valuation, quadratic option, control variates
registration of, 335, 373, 377
tailored control variates, average rate options, 324–325, 371
See also: Variance reduction
Convergence
of discretization schemes. See Discretization, convergence
of numerical methods
Monte Carlo. See Option valuation
See also Discretization, convergence
See Lattice methods, convergence
See PDE methods, convergence
See Root finding algorithms, convergence
See Simulation
Convergence criteria, of discretization schemes
strong, 399–400
test function, 400, 413, 435
weak, 399–400
Conversion constructor, 55, 554, 556, 614
Convertible bond, 450, 461
Copula distributions, 5, 538
Copy assignment, 55, 553, 556
Copy constructor, 55, 553, 614
Correlation
and antithetic variates, 284, 289
and control variates, 317, 318
asset control variates, 320
auxiliary model control variates, 330
and the Bermudan put, 497–498
delta control variates, 342
in the Heston model, 436–438
payoff approximating control variates, 324
and discretization, 404
and the Heston process, 432–434
generating correlated normals, 279
and stratification, 441
Coupling, 104, 208, 356, 614
and Enums, 153
and the factory, 119, 133, 179
psychic, 228
Cox, Ingersoll and Ross (CIR) model, 390–392, 400–401
Feller condition, 408, 417–424, 429
and linear growth condition, 408
and Lipschitz condition, 402, 408, 410, 432
option valuation
and barrier options, 398
bond option (by Monte Carlo), 390–392, 460–461
pure discount bond (explicit solution), 430
simulation. See Simulation, Cox, Ingersoll and Ross and skewness, 429
strong solution to the SDE, absence of, 417
See also: Heston model
Crank-Nicolson. See PDE methods, Crank-Nicolson
Ctor. See Constructors
CV. See Control variates
Data. See also: Object oriented programming; Objects arrays. See Arrays assignment. See Assignment casts. See Casts Const variables. See Const containers. See Containers data section, 59  
See also: Declaration section declaration. See Declaration files. See Files Friend data, 31, 50, 55, 58, 614, 621 global variables. See Global variables handles. See Handles initialization. See Initialization input. See Input invariant, 615 lifetime. See Lifetime literals. See Literals local variables, 39, 47, 50 magic number, 615 memory manager, 174 names. See Names objects. See Objects output. See Output plain old data (POD). See Plain old data pre-extraction, 272, 273 Private data. See Private data Public data. See Public data references. See References scope. See Scope Static variables. See Static variables storage. See Storage type conversion. See Type conversion user defined type (UDT). See User defined types VBA types. See VBA primitive types visibility. See Visibility Data members, 614  
See also: Sentinel; Plain old data; Composited object accessing, 59  
See also: Properties; Data members, restricting access Const data members, 247 and the decorator pattern, 124, 562 and the factory, 127–130 and globals, 62 initialization, 60, 84. See also: Constructors and interfaces, 84–86, 557–558 and meta-class data, 558–559 and Properties, 60–61, 69, 72, 92, 614, 616 restricting access to, 157, 159 and serialization, 127 and state, 40, 553–554, 616 Data section, 59  
See also: Declaration section Date, 56, 145, 148 Debug menu, 518 Declaration. See also: Dim and constants, 245 and constructors, 59  
See also: Conforming object Design flaw, 38, 45, 50, 153, 591 Design patterns adapter pattern, 560–561, 613 composition. See Composition
Design patterns (continued)
decorator pattern. See Decorator pattern
façade pattern. See Façade pattern
factory pattern. See Factory pattern
gang of four, 559
polymorphic factory. See Factory pattern
prototype pattern, 223
registration. See Registration
registration and call-back. See Registration and call-back
singleton pattern. See Singleton pattern
strategy pattern. See Strategy pattern
template pattern, 562–563, 617
wrapper objects. See Wrapper objects
See also: Invocation chain
Destructor, 614
automatic execution, 60, 61
Class_Terminate(), 59–60, 120, 557
and End, 495
and files, 141
and object definition, 59
and RAII. See RAII idiom
and reference counting, 61–63
side-effects, 200
DevC++ 36, 241, 628
Developer tab, 180, 517, 571
Dictionary object, 262–263, 614
and containers, 270, 275
looping through with keys, 262
and OnceOnly, 222
and parameter values, 194, 203
performance, 275–276
and registration, 184, 188, 200–203
Digital option, importance sampling example, 346–348
Dim, 615
and declaring object references, 57
and object instantiation, Dim-New, 57–58, 68, 72, 76
See also: ReDim
Disambiguation, 44, 58, 181
Discount factors, simulation of. See Simulation, discount factors
Discretely reset arithmetically averaged average rate options. See Average rate options
Discretely reset geometrically averaged average rate options. See Average rate options
Discretization. See also: Monte Carlo; Simulation blowing up, 391, 420
convergence
convergence criteria
strong, 399–400
test function, 400, 413, 435
weak, 399–400
empirical rates, 413–416
and the CIR process, 420–424
and discount factors, 428–429
and GBM, 413
and the Heston process, 435–436
and Vasicek, 414
theoretical rates
0.5 strong, 402
1.0 strong, 402–403
1.0 weak, 402
1.5 strong, 404–406
2.0 weak, 410–411
and transforming the SDE, 407–408
exact simulation. See Simulation, exact simulation
and negative values
with the CIR process, 407–408
in the Euler scheme, 407
and rectification functions, 327, 407, 420
schemes
Euler scheme. See Euler discretization scheme
Itô-Taylor discretizations. See Itô-Taylor discretization schemes
log-Euler, with CIR. See Log-Euler discretization schemes
log-normal approximation. See Log-normal approximation discretization schemes
Milstein scheme. See Milstein discretization scheme
moment freezing. See Moment freezing discretization schemes
moment matching approximations
See Log-normal approximation
See Moment freezing
See Normal approximation
normal approximation, CIR, 418
predictor-corrector method. See Predictor-Corrector method
of a transformed SDE, 407–408
Discretization error. See Discretization, convergence
Discretization schemes. See Discretization, schemes
Dispatch, 90, 95, 614
double dispatch, 157
and the façade pattern, 113
and the factory, 188, 198, 218, 222, 223
Distributions
See also: Importance sampling
See also: Moments
See also: Simulation
beta distribution, 11, 233, 536
and bias. See Bias
bi-variate normal distribution, 529
bridge density. See Bridge density of CIR process, 391, 417
closure under convolution, 10
copula distributions, 5, 538
library functions, 533, 536, 538
log-normal density. See Log-normal density
normal density, 293, 536
normal distribution, 22, 25, 251, 536
Index

Ornstein-Uhlenbeck process.
See Ornstein-Uhlenbeck process

Division by zero, 475

DLL. See Dynamic linked library

Do-Exit-Loop, performance, 240–241
Do-Until
and multiple input, 147, 167, 168
performance, 240–241
Do-While
performance, 240–241
in the PSOR solver, 600

Double, 621
and ^, 237
and arithmetic operations, 236–241
casting, 244
casting to Double, 33
casting to Long, 238
and comparison tests, 239
and DLLs, 545
and explicit typing, 243
input, 43, 200, 201, 203, 533
as loop counter, 245
and my_max(), 30
and POD objects, 263
and state, 265–268, 273–277
validation, 41, 533

Double barrier options, 95, 102, 255–258
Double dispatch, 157
Dtor. See Destructor
Dummy variables, 28

Dynamic linked library (DLL), 545–547
and the Err object, 40
“Bad DLL” error message, 519, 521
implementation, 628

Early bound, 139

Early exercise boundary
See also: Barrier options, mimicking barrier option benchmark, from lattice method, 452–453
construction of, in the LSLS method, 463
using CV-rollback, 498–503
plain LSLS, 470–471
in two-pass method, 505–509
convergence to, in the LSLS method
with CV-rollback, single pass, 499–505
with CV-rollback, two pass, 505–509
plain LSLS, 470–471

EEB. See Early exercise boundary

Effective dimension, and path dependency, 107

Efficiency. See Efficiency gain
See also: Performance; Variance reduction

Efficiency gain, 282

Elegance
and the decorator pattern, 285
Error handling (continued)
   by the factory, 183, 185–188, 200–201
   of files, 149
   of keys in a Collection, 263
run-time errors, 40, 85, 194–195, 217, 244
throwing errors, 39, 40–45, 50
trapping errors
   cleaning up, 42, 44
   in main(), 42, 49, 68, 76
   and validation, 222, 533
See also: Validation
Error handling idiom, 42, 50
Error messages
   error handling, 151, 166
   Excel, 263, 519–521
Error trapping. See Error handling
Euler discretization scheme, 14, 37, 402
   in 1-dimension, 406–407
   and bias, 328, 407
   with CEV, 103
   with CIR, 327, 390, 406
     blowing up, 391
     order of convergence, 420–424
     with transformed SDE, 407–408
   convergence, theoretical order, 402
   and delta control variates, 328
   with exotic processes, 416
   floored, 327, 329
   with Fong and Vasicek, 327
   with GBM, benchmark, 37, 406
   order of convergence, 413–414
   with Heston, 329, 432
   order of convergence, 435–436
   negative values, 407
   and predictor-corrector schemes, 411–412
   Euler predictor, 412, 413, 425
   transforming the SDE, 407–408
   with Vasicek, benchmark, 406
   order of convergence, 414
European call options, 7
   Black-Scholes, 25
   in the Heston model, 328–329, 439–442
   general European style options, 3–7, 386
   in Merton jump-diffusion, 23
See also: Exercise streams, implied volatility
See also: Implied volatility
See also: Index to implementations, European call
See also: Lattice methods, convergence
See also: Option valuation, European call
See also: PDE methods, convergence
European put options
   and basis functions, 465
   as control variate for Bermudan puts, 491–492, 498
Event, 120
Events, 39, 56, 59, 101, 119–121, 133
See also: Workbook events
Catching events, 121
Class_Initialize(), 59–60, 120, 557, 613
   and encapsulation, 555
   and initialization, 72
Class_Terminate(), 59–60, 120, 557, 614
defining events, 120
Event, 120
firing events. See Events, raising events
RaiseEvent, 120
raising events, 39, 55, 119, 120, 121
WithEvents, 39, 120, 121
Evil, programming, 614
inappropriate use of End, 495, 511
inappropriate use of globals, 62
inappropriate use of GoTo, 43, 614
inappropriate use of names, 60
inappropriate use of Object, 65
inappropriate use of Variants, 262
lame Case-Else, 223
raising class events, 120
See also: Telepathy; Psychic connection
Evolution, direction
   backwards, 294–295
   and American options. See LSLS method
   binary chop, 294
   implementation, 298, 312
   partial binary chop, 294–295
   See also: Stratified sampling
   forward, 13, 294
   and barrier options, 97
   and plain Monte Carlo. See Index to implementations, Plain Monte Carlo
Evolution, types, 15–16
   element-wise
     level 1, (Chapter 3), 29–30
     level 2, (Chapter 4), 48–49
     level 4, (Chapter 6), 93–100
   holistic, storage constraints, 296
   path-wise
     level 3, (Chapter 5), 67–76, 76–77
     level 5, (Chapter 10), 153–174
     and moment matching, 394, 395
   performance comparisons, 254–258
   slice-wise, 107, 261
   and American option valuation, 449
   and antithetic variates, 285
   and bridge methods, 293–295
   level 4, (Chapter 7), 107–116
   level 5, (Chapter 8), 122–133
   level 6, (Chapter 12), 193
   level 6a, (Chapter 13), 211
   and moment matching, 394, 395
   and storage constraints, 486
Exact simulation. See Simulation, exact simulation
Exact solutions, to SDEs
GBM, 11, 32, 118
Ornstein-Uhlenbeck, 427

Excel
Application methods. See Application methods
Application object, 242
buttons. See Buttons
Cells(). See Cells()
compiler. See VBA compiler
developer tab, 180, 517, 571
error messages, 519–521
Excel 2003, xvi, 180, 623
Excel 2007, 180, 234, 517, 521
Excel settings, 517
See also: VBA settings
front-end. See Front-end
quick access toolbar, 517
range names, 29
ribbon, 180, 517
ThisWorkbook. See ThisWorkbook
VBA settings, 517
Workbook events. See Workbook events
Excel 2003, xvi, 180, 623
Excel 2007, 180, 234, 517, 521
Excel settings, 517
See also: VBA settings
Exception safe, 188, 217, 614
Exercise streams
the complex number exercise, 79
implied volatility stream, 38, 51, 79, 104, 133, 152, 176, 191, 209
lattice stream, 38, 51, 80, 104, 133, 152, 176, 191, 210
PDE stream, 38, 51, 80, 104, 133, 152, 176, 191, 209
pi stream, 37, 51, 79, 103, 133, 152, 191, 209
See also: Pi series expansions
the String exercise, 79
Exiting from a For loop, 256
Exotic processes, 416
Expose, 614
encapsulation, 57
interface, 180, 551
Public and Friend methods, 50
Extended trapezium rule, 543
library object, IntegratorExtended, 542
See also: Numerical integration
Extreme options, 315, 352, 368
Façade pattern, 560, 614
and DLLs, 546
evolver object example, 113
option wrapper object example, 494
wrapping a state, 270
See also: Design patterns
Factory pattern, 55, 119, 562, 614
embryonic factory, 122, 123, 124, 127–130
and Enums, 164, 585
and I/O, 141
and environmental settings, 198, 212, 218, 228
factory layer, 19–21
performance, 254–258
polymorphic factory
fully polymorphic, (Chapter 12), 193–210
base classes, 193, 614
the factory objects, 198–203
the ICreateable interface, 194–195
and input, 203
parameter classes, 194
and setting parameter values, 206–208
and the singleton pattern, 563
synonym classes, 195–196
utility casts, 538
and VBA settings, 179, 517
semi-polymorphic, (Chapter 13), 211–229
the factory object, 218
instantiation, 216, 222–225
meta-class objects, 212–216, 219–222
prototype instantiations, 214, 216, 223, 226
single interface, 183–190
object creation, 187–188
object registration, 184–187, 188
and the VBE object library, 179–183
private factory method, 356
See also: Design patterns
FD. See Finite difference methods
Feller condition
Cox, Ingersoll and Ross, 408, 417–424
Heston, 435–436, 445
Feynman-Kac representation, 581
Files, (Chapter 9), 137–152
and error trapping, 147
files, types, 137–139, 144–145
binary, 137
delimited, 144
random access, 137, 145–151
sequential, 137, 144, 147–148
space-delimited, 144
tab-delimited, 144
the FileSystemObject, 138–140
input from
costs of, 174–176
and Enums, 153–155
for an option book, 164–166
and random access files, 145, 150–151
and sequential files, 144–145, 147–148
and the TextStream object, 140–141
and VBA intrinsic file functions, 144–145
handle number, 144, 147
Files (continued)

output to, 137, 138

costs of, 174–176

for an option book, 166–170

and random access files, 148–150

and sequential files, 66, 170

and the TextStream object, 66, 141

and VBA intrinsic file functions, 144–145

serialization, 127, 616

the TextStream object, 140–141

and UDTs, 145

VBA file objects

See FileSystemObject object

See TextStream object

See also: VBA intrinsic file functions and statements

FileSystemObject object, 66, 138–140, 616

and VBA intrinsic functions, 143

Filtration, 3, 4

Finite difference methods (FD). See PDE methods

Fixed length Strings, 59, 263, 545–546

Flexible barrier option, 488

Fong and Vasiczek model, 325

and an auxiliary model, 325–326

and a delta control variate 326–327

simulation. See Simulation, Fong and Vasiczek process

For-Each, looping through

a Collection, 263, 275–276

the VBComponents object, 189, 198

a Dictionary, illegality of, 262

For-Next

and application building, 29–30, 574–575

default value of loop increment, 29

Exiting from a For loop, 256

and type of loop counter, 245

performance, 240–241

and looping through POD-like objects, 263

and Step, 29, 247

Foresight bias, 392, 452, 459

Form, 164

Forward evolution. See Evolution, direction, forward

Friend data, 31, 50, 55, 58, 614, 621

Front-end, Excel

application building, 25–26, 570–572

for Crank-Nicolson, 586

and the Environment object, 218

input from, 533

in the Crank-Nicolson application, 586

in the lattice application, 570–572

in the Monte Carlo application, 28, 29, 69, 92, 122

and validation, 28, 43–45, 533

for the Monte Carlo applications

level 1, 26

level 4, 89

level 5, 120

level 6a, 213

output to

in the Crank-Nicolson application, 585–586

in the lattice application, 571, 572

in the Monte Carlo application, 26, 45

and utility functions, 533

for the pi application, 184

for the trinomial lattice, 571

Fudge, 420

Functions, 35, 45, 47, 48–50


See also: Procedures; Subs

Functor, 71, 79, 191, 553, 557, 565, 614

Fundamental pricing equation, 3, 567, 581

Gandhi and Hunt, 426, 428

Gang of four, 559

Gaussian affine models

2-factor Gaussian, 278

explicit solutions, 10, 426

and observables, 12

GBM. See Geometric Brownian motion

Gegenbauer polynomials, 464

See also: Basis functions

Genz. See Index to library functions, n_probs(); bi_variate_t_dist()

Genz and Ge. See Index to library functions, bi_variate_n_dist_upper()

Geometric Brownian motion

bridge distribution, of the maximum, sampling from, 397

exact solution to SDE, 32, 118

simulation. See Simulation, Geometric Brownian motion

See also: Black-Scholes model

Getters, 60–61, 64, 556, 614

and encapsulation, 554

and the factories, 127, 198–200, 218, 222

and input, 69, 533

and POD-like objects, 263, 266

and Properties, 60–61

for results, 109, 112, 124, 184

Glasserman, P., xvi, 3, 281, 291, 307, 348, 368, 393, 399, 457

Global variables, 21

evil, 62

and state, 40, 50, 550, 553–554


Go-faster stripes, 247

Golden section search. See Minimization algorithms, golden section search

GoTo, 28, 43, 614

and error handling, 42

Gubbins, 42

Hack, 50

Handles, 614
to files, 144, 147
to objects, 356, 361
Heath, Jarrow and Morton (HJM) model, 8
Hedging, 10–11
calibration to hedging instruments, 10, 393–396, 610
computing hedge ratios, path-wise differentiation, 331–332
delta control variate, 321–322, 324, 342–343, 445
Heptanomial lattice, 105, 152
Heston model, 277–278, 328–330, 401, 431
calibration, 12
explicit solution, European call, 328
Feller condition, 435–436, 445
See also: Zero lock-out Heston
GBM, as perturbation of, 328
implementation
delta control variate, 445
explicit solution, 328
Monte Carlo, 438–439
the little Heston trap, 329
option valuation, 436–446
arithmetic average rate option, 443–444
and barrier options, 446
European call, benchmark, 439
quadratic option, 442–443
simulation. See Simulation, Heston process
variance reduction
and antithetic variates, 290
control variates
and auxiliary instrument, 319
auxiliary model, 319, 330, 438–445
delta control variate, 330, 445
and stratification, 316, 441
zero lock-out Heston (ZLH), 435–436, 438–445
See also: Cox, Ingersoll and Ross (CIR) model;
Black-Scholes model
Hitting time. See also: Stopping time
to a barrier, 454, 511
reification of, 97, 113
distribution, 396
to the early exercise boundary, 492, 498, 511
Holistic evolution. See Evolution, types, holistic evolution
Honte, 245, 615
Horror, 62
House-keeping, 101
Hubris. See Platonic application
I/O. See also: Files; Input; Output channels. See Channels
See also: Streams, I/O
console, 164
form, 164
streams. See Streams, I/O
IDE, 121, 517
Idioms, 615
array expansion idiom, 147, 166, 174
catch and re-throw idiom, 44, 76, 531
error handling idiom, 42, 50
Not first time. See Static initialisation idiom
RAII idiom, 72, 76, 95, 141, 559, 616
singleton idiom, 63, 216, 563
Static initialization idiom. See Static initialization idiom
See also: Design patterns
If-Then, If-Then-Else
block If, 239
compound If, comparison with Select, 240
in-line If, 239
Ignorance, 517, 559
IID, 14, 32
Implemented, 84–86, 557–558
and registration, 197, 201
Implied volatility, 38
See also: Exercises, implied volatility stream
See also: Root finding algorithms
volatility surface, 10, 12
Importance sampling, (Chapter 22), 345–369
average rate option, (GBM), 364, 366, 379–380, 381–382
bank of, 352, 368
combined with control variates, 371–372
digital option, standard error analysis, 346–348
IS densities
constant GBM density, 350–351
terminally modified density, 351–352
ITM options, 345–369
OTM options, 345–369
quadratic options, 363, 366, 379, 381
registration of, 352–356, 373, 377
See also: Variance reduction
In the money (ITM) options
and importance sampling, 345–369
and the LSLS method, exercise in, 479
and problems for Monte Carlo, 282, 315
and stratified sampling, 302
Indenting, 28, 621
Independent identically distributed. See IID
Indirection, 67, 81, 122, 266, 615
and wrapper objects, 271
See also: Dispatch
Industrial strength, 21, 101, 613, 616
Ingratiation, xvii, 29, 102
Inheritance, 614
in C++, 558
in VBA, 55
Initialization, 615
in application building, 28–30, 572–574, 585–589
default, 28, 553–554, 557
in-line, 28, 29
of objects. See Initialization of objects
and the Static initialization idiom. See Static initialization idiom
Initialization of objects, 553–556
Class_Initialize(). See Events, Class_Initialize()
and Dim-New, 58
in the factory
embryonic, 127–130
polymorphic
ICreatable, 194
IReuseable, 195
and POD-like objects, 263
and Properties, 60–61
and RAI. See RAI idiom
with a SetValues() method, 68–73
with a SetValues() Property, 84–85, 91–97, 127
in VBA, 556–557
In-line comments, 622
Input. See also: Files, input from; Front-end, input from; I/O, channels
and decoupling, 90–91, 93
and encapsulation, 69, 75
and environmental data. See Environment
and the factory 20, 21
the embryonic factory, 123, 127–130
the polymorphic factory, 187, 197–206
the semi-polymorphic factory, 212, 216, 218, 222, 225, 226
formatted input, 145, 148, 151
and the InputManager object, 69, 92, 127
and UDTs, 39, 46–47, 145, 155, 591
Instantiation, 56, 57–58, 615
and the Dictionary object, 262
and Dim-New, 57–58
and the factory
embryonic, 127–130
and an option book, 162–163
polymorphic, 184–188, 198–203
semi-polymorphic, 216, 222–225
and the FileSystemObject object, 139
on the fly, 185, 196, 207, 532, 544
pop an object, 196, 532, 544
and RAI. See RAI idiom
and reference counting, 61–63
and side effects, 183, 200
and the singleton pattern, 563–565
as a structural property, 553–556
in VBA, 556–559
Integer, 621
casting, 244
and DLLs, 545
literals, 33
Integrated development environment. See IDE
Interest rate models
See Cox, Ingersoll and Ross (CIR) model
See Fong and Vaseck model
See Gaussian affine models
See HJM model
See Libor market model (LMM) model
See Vasicek model
Interfaces, 84–86
See also: Polymorphism
application interface, 193, 228
conforming object. See Conforming object
mix-in interfaces, 193, 477, 615
ICreatable, 194–195
IReuseable, 195
Invariant, 615
Inverse transform method. See Normal variate generators, inverse transform
Invocation chain, 155, 183, 184, 615
Invoker, 19, 187
IS. See Importance sampling
Is, 66
ITM. See In the money options
Iterated Itô integrals, 403–406, 425
moments of, 403, 405, 425
simulation of, 404, 405, 406, 425
See also: Itô-Taylor discretization schemes
Itô-Taylor discretization schemes.
0.5 strong. See Euler discretization scheme
1.0 strong. See Milstein discretization scheme
1.5 strong, 404–406
in 1-dimension, 103, 408–410
with CIR, 410
bodges, 420
order of convergence, 420–424
with exotic processes, 416
with GBM, benchmark, 409
order of convergence, 413–414
with Vasicek, benchmark, 409
order of convergence, 414
2.0 weak, in 1-dimension, 410–411
with CIR, 411
bodges, 420
order of convergence, 420–424
and exotic processes, 416
with GBM, benchmark, 410
order of convergence, 413–414
with Vasicek, benchmark, 411
order of convergence, 414
relationship with predictor-corrector, 413
Jäckel, P., xvi, 3, 281, 307, 312, 399, 417
Jamshidian formula, 326
Jump-diffusion models, 319
See also: Merton jump-diffusion model
Knock-in barrier option, 7, 97, 98, 113, 255, 316
Knock-out barrier option, 16, 95–97, 98, 113, 255, 290, 316, 396
Labels
and “:”, 28
and error trapping, 42–45, 50
and GoTo, 28
Laguerre polynomials, 464–465, 479–483
See also: Basis functions
Late bound, 139, 244, 262, 263
Lattice methods
binomial, 210, 567
branching probabilities, 105, 210, 457, 567, 569
and moment matching, 570
convergence, 575–576
American put, 452–453, 576
continuation values, 450, 568, 575
eyearly exercise boundary, 452–453
comparison with PDE method, 593
European call, 576
heptanomial, 105, 152
moment matching, 570
pruning, 104, 576
Richardson extrapolation, 452
trinomial, 567–570
exercise stream, 38, 104–105, 152
implementation, 570–575
performance, 575–576
Layout, 68
indenting, 28, 621
Laziness, 275, 298, 495, 531, 550
LD. See Low discrepancy sampling
Legendre polynomials, 464–465, 466, 479–482
See also: Basis functions
Level, programming. See Programming level
Libor market model (LMM)
calibration, 12
state variables, 12, 268, 449
Lifetime
of composites objects, 559, 613
of function arguments, 185
of Statics, 47
Linear equation methods
ordinary least squares regression (OLS), 445, 463, 466, 468
singular value decomposition (SVD), 445, 468
tri-diagonal solvers
See also: PDE methods, SOR; PDE methods, LU
tri-diagonal solver, 589, 595
Linear growth condition, 401–402
and CIR, 408
Lipschitz condition, 401–402, 416
and CIR, 402, 408, 410, 433
and Heston, 432
Liquid prices, and calibration, 10, 11, 610
Literals, 33, 554, 615
and performance, 236, 237
and telepathy, 194
LMM. See Libor market model
Local variables, 39, 47, 50
Log-Euler discretization schemes, with CIR, 419
blows up, 420, 421, 424
order of convergence, 420–424
Log-normal approximation discretization schemes,
CIR, 418–419
convergence, 420–424
with Heston, convergence, 435–436
Log-normal density, 331, 526
central moments, 418
moments of, 331
Long, 621
and ^, 237
casting, 244
casting to Long, 238, 554
and comparison tests, 239
and Err.Raise, 39, 40
as a handle, 361
and input, 44, 203
as keys (in a Dictionary), 275
as loop counter, 245
and validation, 41, 44, 533
Long-step Monte Carlo, 26, 418, 429
with barrier options, 257
and efficiency, 378, 383, 384
and step frequency, 112, 261
Longstaff and Schwartz least squares (LSLS) Monte Carlo method. See LSLS method
Lookback call option, discretely reset, 305
Low discrepancy (LD) sampling, (Chapter 19), 307–317
See also: Stratified sampling
average rate options, 314–315, 315–316
globally averaged, 315
Bermudan puts, 482–483, 487–488
and bias, 314–315
and standard error, 314–315
See also: Sobol’ sequences
See also: Variance reduction methods
LSLS method
2-pass method, 505–509
American option, general, 456–457, 458
American put options, 459–460
See also, Option valuation, American put bias in, 474
bond options, 460–461
cash-flow formulation, 459
error in EEB estimate, 491
error in mimicking barrier option value, 491
using control variates, for early exercise boundary, 498–505
using control variates, valuation
Bermudan-early control variate, 493
Bermudan put control variate, 492
Bermudan-T1 control variate, 493
Bermudan terminal control variate, 493
LSLS method (continued)
  European put control variate, 492
  performance of, 497–498
  stock control variate, 492
  stopping time control variates, 491
  See also: Continuation values; Early exercise boundary
LU tri-diagonal solver, 589, 595

Magic number, 615
main(), 19–21, 26
  See also: Calling procedure
  See also Index to implementations, main()
Maintenance
  and comments, 26, 28
  and data representation, 265
  and objects, 549
  and re-factoring, 616
  and references, 58
  and tricky code, 234, 253
  and VBA compiler problems, 518
  and yukky code, 39, 578
Market
  computing values of observables, 389
  market value, 8, 38, 392
  money market, 449
Market component, of models, 8, 10
Martingale, in no-arbitrage pricing framework, 3, 424, 567
Martingale correction
  Libor market model, 393
  simulation and, 393–394
Martingale simulation, 393
MC. See Monte Carlo
Me, 66, 92, 158
Measure
  Borel measure, 4
  equivalent martingale measure, 424, 567
  pricing measure. See Pricing measure
  risk neutral measure. See Pricing measure
Member functions. See Methods, of an object
Memory manager, 174
Mersenne twister, 251
Merton jump-diffusion model, 23
  explicit solution, European call, 23
Message
  See also: Request
  calling an object’s interface method, 556, 559–560,
  615
Meta-classes, (Chapter 13), 211–229, 558–559, 615
  in C++, 559
  performance, 254–255
  reference counting, 212
  role of, 212–214
Method of false position, 605
  in exercises, 80, 104
  FalsePosition(), 606

Methods, of an object
  and objects’ implementation, 615
    Private methods, 59, 555
    and objects’ interface, 555, 615
    Public interface, 555, 615
  See also: Request
  and POD-like objects, 263–266
Milstein discretization scheme, 37, 402–404
  in 1-dimension, 406–407
  in 2-dimension, 404
  with CEV, 103
  with CIR, 406
    order of convergence, 420–424
    positive values, 408
  convergence, theoretical order, 404
    with exotic processes, 416
    with GBM, benchmark, 37, 406
    order of convergence, 413–414
    with Heston, 432–433
    order of convergence, 435–436
    and iterated Itô integral, 403
    with Vasicek, benchmark, 406
    order of convergence, 414
  Minimization algorithms, 610–612
    golden section search, 152, 611
    exercises, 152
    GoldenSection(), 612
Mix-in interfaces. See Interfaces, mix-in interfaces
Models. See also: Stochastic differential equations;
  Discretization; Simulation
  See also: Calibration
  Black-Scholes framework, 3, 7, 25, 324
    See Black-Scholes model
  CEV (Constant elasticity of variance) model, 102–103, 114
  Cox, Ingersoll and Ross model. See Cox, Ingersoll
  and Ross model
  Fong and Vasicek model. See Fong and Vasicek
  model
  Gaussian affine models. See Gaussian affine models
  Heath, Jarrow and Morton (HJM) model, 8
  Heston model. See Heston model
  interest rate models
    See Cox, Ingersoll and Ross model
    See Fong and Vasicek model
    See Gaussian affine models
    See Heath, Jarrow and Morton model
    See Libor market model
    See Vasicek model
  jump-diffusion models. See Jump-diffusion models
  Libor market model (LMM). See Libor market model
market component of models, 8, 10
Merton jump-diffusion. See Merton jump-diffusion
SABR, 8, 12
Shimko, Tejima and Van Deventer model, 277, 279
stochastic volatility models
See Heston model
See SABR model
See also: Fong and Vasicek model
stock models
See Black-Scholes model
See CEV model
See Jump-diffusion models
See Merton jump-diffusion model
See Shimko, Tejima and Van Deventer model
Vasicek model. See Vasicek model

Modules
class modules. See Class modules
code modules. See Standard modules
standard modules. See Standard modules
Moment freezing discretization schemes, 327, 329
with CIR, 419–420
order of convergence, 420–424
with Heston, 434
order of convergence, 435–436
Moment matching, lattices, 570
Moment matching correction, 394–396
and antithetic variates, 291
and calibration, 394
and holistic evolution, 16, 294
and put-call parity, 398
and stratified sampling, 291
and storage, 394, 395
Moment matching discretization schemes
See Log-normal approximation discretization schemes
See Moment freezing discretization schemes
See Normal approximation discretization scheme
Moments
central moments. See Central moments
of CIR process, 418
of iterated Itô integrals, 403, 405, 425
and weak discretization schemes, 411
Money market account numeraire, 3
See also: Accumulator account numeraire
Monitor, 72, 91
See also: Index to implementation, Counter, step
Monte Carlo. See also: Simulation; Index to implementations, Monte Carlo
and American options. See Option valuation,
American put
See also: Bermudan option valuation, by Monte Carlo
Bermudan option valuation. See Bermudan option valuation, by Monte Carlo
bias. See Bias
See also: Discretization, convergence
discretization. See Discretization
efficiency gain, 282
evolution, direction. See Evolution, direction
evolution, types. See Evolution, types
long-step. See Long-step Monte Carlo
option valuation. See Option valuation
plain Monte Carlo. See Index to implementations,
Plain Monte Carlo
Richardson extrapolation, and the American put.
See Richardson extrapolation
sample paths. See Sample paths
short step, 112, 254, 431
simulation. See Simulation
slices. See Slices
standard error. See Standard error
variance reduction
antithetic variates. See Antithetic variates
control variates. See Control variates
in the Heston model. See Heston model, variance reduction
importance sampling. See Importance sampling
low discrepancy sampling. See Low discrepancy sampling
spectral decomposition, 294
stratified sampling. See Stratified sampling
Moro. See Index to library functions, cndev()
Morris. See Index to library functions,
gamma_inverse()
Moshier. See Index to library functions,
beta_inverse()

Names
disambiguation, 44, 58, 181
inappropriate names, 60
of loop counter, 468
name clash, 44, 531, 546
See also: Names, disambiguation
naming conventions, Private data members, 59
qualified names, 44, 130
analogy with object syntax, 550, 556, 565
and library functions, 531
and With, 66
range names, 59
Namespace, 183, 531
Natural basis functions, 464, 465, 479–482
See also: Basis functions
New, 57–58, 61, 556
and Dim, 57–58, 68, 72, 76
and polymorphism, 84–86, 558
Newton-Raphson method, 608–610
in exercises, 209
NewtonRaphson(), 609
NIG. See Normal inverse Gaussian process
Nitty-gritty, coding, 13, 233, 234
Non-central χ² distribution, 391, 417
Non-polymorphic interface, 153
Non-virtual interface, in VBA. See Non-polymorphic interface
Normal approximation discretization scheme, CIR, 418
Normal density, 293, 536
Normal distribution, 22, 25, 251, 536
Normal inverse Gaussian (NIG) process, 292
Normal variate generators
from 12 uniforms, 23
inverse transform, 6, 246, 251
and the bridge distribution, 391
performance, 246, 249–251, 254–255
and stratified sampling, 292, 312, 441
See also: Index to library functions, cndev()
See also: Application methods,
Application.NormSInv()
polar rejection, 33
and ^, 238
GetNormal(), 33
normal1(), 22, 535
performance, 246, 249–251, 254–255
Not first time idiom. See Static initialisation idiom
Nothing, 57–58, 60, 556
and anon(), 196
and reference counting, 61–62
Numeraire, 3, 567
accumulator account numeraire, 325, 424, 581
money market account numeraire, 3
See also: Accumulator account numeraire
Numerical integration
extended trapezium rule. See Extended trapezium rule
and LD sampling, 307
library quadrature objects, 542
and the Monte Carlo method, 4–6, 11, 14, 389, 390
and option pricing in the Heston model, 317, 328–329
Simpson’s rule. See Simpson’s rule
trapezium rule. See Trapezium rule
Numerical methods. See also: Index to implementations
bodge. See Bodge
convergence, of numerical methods
Monte Carlo. See Option valuation
See also: Discretization, convergence
See Lattice methods, convergence
See PDE methods, convergence
See Root finding algorithms, convergence
See also Simulation
discretization error. See Discretization, convergence
efficiency. See Efficiency gain
See also: Performance; Variance reduction
lattice methods. See Lattice methods
linear equation methods. See Linear equation methods
minimization algorithms. See Minimization algorithms
Monte Carlo. See Monte Carlo

See also: Simulation
numerical integration. See Numerical integration
PDE methods. See PDE methods
pre-computing, 233, 234
quadrature. See Numerical integration
random number generators
See Normal variate generators
See Uniform variate generators
Richardson extrapolation, and the American put
See Richardson extrapolation
root finding algorithms. See Root finding algorithms
simulation. See Simulation
See also: Discretization; Random number generators
speed. See Speed
Numerical Recipes
library functions, 531, 533, 536
low discrepancy sampling (LD) sampling, 307, 310
minimization, 610, 611
quadrature methods, 543
random number generators, 33, 74, 246, 251
singular value decomposition (SVD), 468
tri-diagonal solvers, 584, 596, 598
root finding algorithms, 603, 606
vade mecum, xvi
NVI. See Non-virtual interface

Object. 65
and anon(), 196
and the factory, 223
Object declaration, references, 56, 57–59, 556–557, 615
Object definition, in VBA, 56, 59–61
StopWatch example, 63–65
See also: Class modules
Object organization. See also Design patterns
aggregation, 613
association, 613
bidirectional association, 613
Object oriented programming. See also: Programming concepts; Object organization
dispatch. See Dispatch
encapsulation. See Encapsulation
exception safety, 188, 217, 614
functor, 71, 79, 191, 553, 557, 565, 614
interfaces. See Interfaces
message. See Message
meta-classes. See Meta-classes
object declaration, references, 56, 57–59, 556–557, 615
object definition. See Object definition
object organization. See Object organization
objects. See Objects
plain old data (POD). See Plain old data
polymorphism. See Polymorphism
reference counting. See Reference counting
reification. See Reification
request. See Request
Simula 67, 549
strong typing. See Strong typing
tracking. See Tracking
in VBA, 55–102
See also: Class modules
See also: VBA intrinsic objects
See also: VBA OOP reserved words
See also: VBE object library
Objects. See also: Excel; VBA intrinsic objects; VBA OOP reserved words
binding. See Binding
casting, 196
constructors. See Constructors
copy assignment, 55, 553, 556
data members. See Data members
declaration, references, 56, 57–59, 556–557, 615
definition. See Object definition
destructor. See Destructor
decrease. See Decrease
getters. See Getters
initialization. See Initialization
instantiation. See Instantiation
member functions. See Methods
methods. See Methods
references. See References
setters. See Setters
state. See State, of an object
See also: Object oriented programming
OLS. See Ordinary least squares regression
On Error, 42–43, 50, 517
Ontology, 95
OOP. See Object oriented programming
operator(), 71, 614
Operator overloading, 556, 557, 615
Optimization

compiler optimization
of loop contents, 235
of stub procedures, 242
and OLS regression, 463, 468
premature, 233
See also: Minimization algorithms
Option Explicit, 27, 40, 622
Option pricing formulae
in the Black-Scholes model
 barrier down and out call, 17
Bermudan-2 put, 528–529
compound call option, 529
continuously reset geometrically averaged average rate option, 18
discretely reset geometrically averaged average rate option, 117, 523–526
European call, 25
quadratic payoff option, 526–528
in the CIR model
pure discount bond, 430
in the Heston model
European call, 328–329
in the Merton jump-diffusion model
European call option, 23
in the Vasicek model
bond call option, 326
pure discount bond, 17
Option Private, 58
Option valuation, by Monte Carlo, convergence
American put, 486–488
in basis functions, 487
extrapolation from Bermudan put value, 509
in sample paths, 486
and stratification, 487
average rate options
antithetics, 288–289
combined methods
GBM, 379, 381–382
Heston, 443–446
control variates, 338–342
importance sampling, 364, 366
low discrepancy sampling, 314–315
stratified sampling, 298–304
barrier options, 102
Bermudan puts, by LSLS, 473–474, 484–486
2-pass method, 505–509
in basis functions, 473–474
using rollback control variates, 499–505
in sample paths, 473
and stratification and LD sampling, 482–483, 487–488
using valuation control variates, 497
compound option (2 resets)
in basis functions, 483–484
European call
convergence, Heston, 435–436
importance sampling, 362, 364–366
variance reduction, combined methods
GBM, 378, 380–381, 384–386
Heston, 435–445
out of the money (OTM) options
average rate options
antithetics, 289
combined methods, 381–382
control variates, 339, 342
importance sampling, 366
low discrepancy sampling, 315
stratified sampling, 302
European call
combined methods 380–381
importance sampling, 364–366
Option valuation (continued)
  quadratic option
    combined methods, 381
    importance sampling, 366
  quadratic option, 323, 526–528
    combined methods
      GBM, 379, 381
      Heston, 442–443, 445
  control variates
    delta control variate, 343
    payoff matching control variate, 323–324, 343–344
    performance, 342–343
    importance sampling, 363, 366
 Optional sampling theorem, 492
 Options. See also: Numerical methods; Option pricing
   formula; Option valuation
   American options. See American options
   average rate options. See Average rate options
   barrier options. See Barrier options
   Bermudan options. See Bermudan options
   bond options. See Bond options
   book of options, (Chapter 10), 153–176
   compound option, 483–484, 529
   contract, option, 10
   convertible bond, 450, 461
   digital option, importance sampling example, 346–348
   European call options. See European call options
   European put options. See European put options
   extreme options, 315, 352, 368
   in the money (ITM) options. See In the money
   (ITM) options
   lookback call option, discretely reset, 305
   out of the money (OTM) options. See Out of the
   money (OTM) options
   quadratic option, 323, 526–528
     See also: Option valuation
   redeemable American bond option, 461
   redeemable, callable, convertible bond, 461
   reset dates. See Reset dates
     See also: Put-call parity
 Ordinary least squares regression (OLS), 445, 463, 466, 468
 Ornstein-Uhlenbeck process
     See also: Vasicek model
     bridge distribution of integrated process, 391
     exact solution to SDE, 427
     sampling from, 327
     simulation. See Simulation, Ornstein-Uhlenbeck
     process
 OTM. See Out of the money (OTM) options
 Out of the money (OTM) options
   and combined variance reduction methods, 380–382
   and importance sampling, (Chapter 22), 345–369
   and problems for Monte Carlo, 282
 See also: Option valuation, out of the money
   options
 Output, 22
 See also: Files, output to; Front-end, output to
 See also: I/O, channels; Index to implementations,
   Counter, step
   to the client, 18, 123, 130
   and decoupling, 75, 90–91, 93, 100
   from the destructor, 72, 76, 92, 169
   and encapsulation, 39, 45, 67, 75
   and environmental data. See Environment
   and events, 119
   and the factory, 20–21
     the embryonic factory, 122, 123, 127–133
     the polymorphic factory, 198–200
     the semi-polymorphic factory, 212, 217, 226
     formatting output, 65, 137, 144, 151
   and the OutputManager object, 71–72, 92
   and registration and call-back, 100
 Parameter classes, 194
 Partial binary chop evolution, 294–295
 See also: Evolution, direction, binary chop
 Partial differential equation. See PDE methods
 Path-wise differentiation, and hedge ratios, 331–332
 Path-wise evolution. See Evolution, types, path-wise
   evolution
 PDB. See Pure discount bond
 PDE methods
   comparison with lattice method, 592, 602
   convergence, 582, 591–595, 601–602
     for American put, 591, 593, 601
     continuation values, 584, 591, 598
     for European call, 593
     higher order convergence
       projective SOR (PSOR), 598
       successive over-relaxation (SOR), 595–602
   Crank-Nicolson, (Appendix G), 581–602
   performance, 592–595
   and early exercise, 449, 584, 591, 593
   See also: PDE methods, SOR; PDE methods, PSOR
   problems in higher dimensions, 11
   PSOR, 598, 599, 601
   SOR, 595–602
   transforming the PDE, 584–585
 Performance
   arithmetic functions, 241
   arithmetic operators, 236–238
   arrays. See Arrays, performance
   assignment, 238, 241, 243, 244, 245
   C++, xv, 36, 235, 241
   casts, 244, 248
   comparison with C++, 36
   concatenation, Strings, 241
   Const variables, 236, 245, 247
control statements, 239–241, 245
data structures, (Chapter 16), 261–277
slice representations, 268–277
state representations, 266–268
1-factor, 272–273
n-factor, 273–277
plain state, 273–275
structured state, 275–277
and the factory pattern, 254–258
global variables, and statics, comparison, 238, 245–247
I/O, 174–176
if statement, 239
implicit type conversion, 244
lattice, trinomial, 575–578
literals, 236, 237
loops, 240–241
and meta-classes, 254–255
and Monte Carlo
evolution method. See Evolution, types,
performance comparisons
option valuation. See Option valuation
programming level. See Index to implementations,
Plain Monte Carlo, performance
numerical integration, discount factors, trapezium
rule, 428
objects in VBA
in plain Monte Carlo. See Index to implementations,
Plain Monte Carlo
See also: VBA intrinsic objects
PDE method, Crank-Nicolson, 592–595
plain old data, 266–268, 273–277
power operator, 237
procedure calls
cost of a call, 242
typing arguments, 243
typing return values, 243
programming level. See Index to implementations,
Plain Monte Carlo
random number generators
See Normal variate generators
See Uniform variate generators
rounding a division, 238
rounding a double, 238
Select statement, 240
simulation
of CIR process, 420–424
of GBM, 413–414
of Heston process, 435–436
of Ornstein-Uhlenbeck process, 413–414
static variables, 245–247, 254
statics and globals, comparison, 238, 245–247
static initialization idiom, 247
of a step counter, 250–251, 248, 576
UDTs, 244
variant arrays, 275–276
VBA library functions. See Index to library
functions
VBA primitive types. See VBA primitive types
VBA reserved words. See VBA reserved words
Pi series expansions
Bailey, Borwein and Plouffe, 183
Beeler et al., 37
Euler, 183
Gosper, 37
Leibniz, 183
miscellaneous, 37, 191, 209
Ramanujan, 209
Schellbach, 183
Plain Monte Carlo. See Index of implementations,
Plain Monte Carlo
Plain old data (POD), 263, 555, 615
See also: User defined types
performance, 266–268, 273–277
POD-like objects, 263–265
slice representation, 270, 273–277
state representation, 265–266
and VBA containers, 270, 273
Platonic application, 19–20
POD. See Plain old data
Poisson process
and Merton jump-diffusion process, 23
and rare events, 315
See also: Index to library functions,
rand_Poisson()
Polar rejection method. See Normal variate generators,
polar rejection
Polymorphic applications. See Programming level,
level 4
Polymorphic factory. See Factory pattern, polymorphic
factory
Polymorphic factory applications. See Programming
level, level 6
Polymorphic hierarchy, 615
Polymorphism. See also: Interfaces
base class. See Base classes
derived objects. See Derived objects
inheritance. See Inheritance
polymorphic hierarchy, 615
See also: Programming to an interface
Power functions, 464–465, 466–468, 479–482
See also: Basis functions
Pre-computing, 233, 234
Predictor-corrector method, 411–413
with CIR, 413
order of convergence, 420–424
with exotic processes, 416
with GBM, benchmark, 412
order of convergence, 413–414
with Heston, 433–434
order of convergence, 435–436
partial predictor-corrector, 435
Index 667

and polymorphism, 85–86
and Private data, 60–61
and scope. See Private data
and SetValues() 93, 95, 127
Prototype pattern, 223
See also: Design patterns
Pruning. See Lattice methods, pruning
PSOR. See Projected successive over-relaxation
Psychic connection, 225, 228, 616
See also: Telepathy
Public. See Public data
Public data, 31, 39, 50, 55, 56, 616
and fixed length Strings, 263
and library procedures, 40, 44
and object interfaces, 58–61, 75, 84, 550
and Option Private, 58
and UDTs, 554, 555
See also: Plain old data
Pure discount bond (PDB)
in the CIR model, 430
in the Fong and V asickek model, 325–326
in the Shimko, Tejima and Van Deventer model, 279
in the V asickek model, 17
See also: Bond options
Put-call parity, 393
martingale correction, 393–394
in Monte Carlo method, violation of, 392, 393, 398
Quadratic option, 323, 526–528
See also: Option valuation, quadratic option
See also: Index to implementations, Quadratic option
Quadrature. See Numerical integration
Qualified names. See Names, qualified names
Quantiles
and importance sampling, 347
and stratified sampling, 291–292
Queues, 263
Quick access toolbar, 517
RAII idiom, 72, 76–77, 141, 559, 616
RaiseEvent, 120
Raising errors. See Error handling, raising errors
Random access files, 137, 145, 148, 150
Random number generators
See Normal variate generators
See Uniform variate generators
Range names, 59
Rare events, 315
Rasmussen, N., 447, 459, 470, 494, 498, 509
Rectification functions, 327, 407, 420, 432, 446
Redeemable American bond option, 461
Redeemable, callable, convertible bond, 461
ReDim, 265, 266
ReDim Preserve, 147, 166, 174
See also: Memory management
Re-factor, 616
Reference counting, 61–63, 616
and meta-classes, 212, 214, 225
References, 61–63, 556
assignment, 556
dangling references, 60, 61
declaration of, 56, 57–59, 556–557
and maintenance, 58
reference counting. See Reference counting
self-references, 61
Registration. See also: Registration and call-back
automatic, 188, 197–203
in C++, 183
for control variates, 335, 373, 377
and the factory, 183, 184–188, 562
for importance sampling, 352–356, 373, 377
of objects, 179
for output, 100
side-effects, 183, 184, 200
and the singleton pattern, 563–565
of values, 74, 95, 97, 109
and the VBIDE object, 559
work-arounds, 211–229
See also: Design patterns
Registration and call-back, 55, 559, 616
and the factory, 562
for output, 100
See also: Design patterns
Reification, 616
of a grid, 586
of input data, 46
of options, 97, 109, 305, 562
of a slice, 261
of a state, 266
Request, 21, 556, 616
See also: Message
and the adapter pattern, 560
and control variates, 335, 337, 373, 439
and events, 121
to execute, 90, 107, 113, 159, 285
to the factory
embryonic factor, 123, 130, 162, 172
polymorphic factory, 183, 187–188, 200, 206
semi-polymorphic factory, 214, 216, 218,
221–223, 225, 228
and importance sampling, 352, 353, 356, 358, 361
for output, 92, 166–170
for results, 73, 95
Reset. VBA project, 257
Reset dates, 109–110, 116
and time steps, 110, 116, 261
See also: Options, average rate options; Options,
Bermudan put options
Resource acquisition is initialization. See RAII idiom
Resume, 42, 44, 50
Resume Next, 42, 517
Re-usability. See also: Index to library functions
and files, 138
and functions, 21, 39, 242
IReusable mix-in interface, 195
lack of, 35, 39, 549
and modules, 48
and objects, 56–57, 63, 75, 551
Ribbon, Excel, 180, 517
Richardson extrapolation, and the American put
for the early exercise boundary, 453
with a lattice methods, 452
with Monte Carlo, 489, 491, 627
Ridder’s method, 606–608
in exercises, 80, 104
Ridders(), 608
Risk neutral measure. See Pricing measure
RMSE, 458, 471, 473–474, 499–505, 539
Root finding algorithms, 603–610
bisection method. See Bisection method
bracketing interval, 603
convergence, 604, 608
method of false position. See Method of false position
Newton-Raphson method. See Newton-Raphson method
Ridder’s method. See Ridder’s method
Root mean square error. See RMSE
Routine. See Procedures; Numerical Recipes
Run-time errors, 40, 85, 194–195, 217, 223, 244
SABR model, 8, 12
Sample paths, 4, 6, 7
See Evolution, types, path-wise; Simulation
Sample space, 4, 11
Sanity check, 92, 266
Scaled basis functions, 465
See also: Basis functions
Schiller, 66
Scope, 616
file scope, in C++, 179
module scope, 58, 554
See also: Private data
object scope. See Data members
procedure scope. See Local variables
project scope. See Friend data
of Properties. See Private data
public scope. See Public data
explicit destruction, 58, 60, 62, 101
implicit destruction, 58, 556
statement scope, 196, 544
of variables in blocks, 47
Scripting runtime library, 139, 262, 616
Scripting.Dictionary 262
Scripting.FileSystemObject, 139
SD. See Standard deviation
SDE. See Stochastic differential equations
SE. See Standard error
Select-Case, 616
and “;”, 29
and Enums, 153, 177, 585
and level 1 design, 551, 574, 585
non-polymorphic object creation, 83, 92, 102, 127, 153
performance, 240
polymorphic object creation, 184, 188
semi-polymorphic object creation, 216, 223, 226
Sentinel, 616
and the Static initialisation idiom, 47, 50, 616
Sequential files, 137, 144, 147
Serialization, 127, 153–155, 352–353, 360, 616
Set, 615
and Collection elements, 263
and events, 121
and interfaces, 85, 558
and object instantiation, 56, 57–58, 61, 556
See also: Nothing; Property; RAI idiom
Sets, 222, 263
Setters, 616
See also: Initialization
and encapsulation, 55, 554, 557
and POD-like objects, 263–265
and Properties, 60–61, 556
Shimko, Tejima and Van Deventer model, 278, 279
Short rate models
See Cox, Ingersoll and Ross model
See Fong and Vasicek model
See Gaussian affine models
See Vasicek model
Short step Monte Carlo, 112, 254, 431
Signature. See also: Procedures, stubs
of class events, 59
and DLLs, 545
and events, 121
and interfaces, 84–86
of library functions, 531–542
and option definition, 95
Simpson’s rule, 543
library object, IntegratorSimpson, 542
See also: Numerical integration
Simula 67, 549
Simulation. See also: Discretization; Monte Carlo;
Random number generators
Cox, Ingersoll and Ross process, 417–424
1.5 Itô strong, 410, 421
2.0 Itô weak, 411, 421
convergence rates, 420–424
Euler, 406, 421
exact simulation, 417–418
log-Euler, 419, 421
log-normal approximation, 418–419, 421, 434
Milstein, 406, 408, 421
moment freezing, 419–420
Index

and negative values, 407–408
normal approximation, 418–419, 421
performance, 420–424
predictor-corrector, 413, 421
and rectification functions, 327, 420
of a transformed SDE, 407–408
discount factors, 424–429
convergence rates, 428–429
direct approximation
bias correction, 426
Gandhi and Hunt, 426
integral approximations
histogram, 424
trapezium, 425
Itô-Taylor approximations
0.5, Euler, 425
2.0 weak, 425–426
exact simulation
CIR process, 417–418
GBM, 32
Heston process, Broadie and Kaya, 432, 446
and moment freezing, 434
Ornstein-Uhlenbeck process, 427
exact simulation, absence of
and bias, 390
CEV, 103
and discretization, 401, 407
Fong and Vasicek process
Euler, 328
moment freezing, 328
direct approximation
bias correction, 426
Gandhi and Hunt, 426
integral approximations
histogram, 424
trapezium, 425
Itô-Taylor approximations
0.5, Euler, 425
2.0 weak, 425–426
exact simulation, absence of
Singe, 621
performance, 244
and Rnd(), 34
and Timer(), 56
Singleton idiom, 63, 216, 563
See Singleton pattern
Singleton pattern, 563–565, 616
and the Err object, 40
and the factory, 218
and meta-class data, 216
and standard code modules, 552, 564–565
and tracking, 63
See also: Design patterns
Singular value decomposition (SVD), 445, 468
Slice-wise evolution. See Evolution, types, slice-wise
Slices, 6, 15
See also: Evolution, types, slice-wise
and containers, 268–271
performance
1-factor state, 271–273
3-factor state, 273–277
slice object, 265, 277
Sobol’ sequences, 307–310
Bratley and Fox, 307
direction numbers, 307
SOR. See Successive over-relaxation
Space-delimited files, 144
Spectral decomposition, 294
Speed
of data structures, (Chapter 16), 261–279
of programming level, (Chapter 15), 249–259
of VBA, (Chapter 14), 233–248
See also: Convergence, of numerical methods;
Variance reduction; Performance
Squeaky, 216
Stack
communication, up, 39
events and, 39, 42–44, 60
depth, 101, 296
exiting from, 40, 495, 511
unwinding, 21
Standard deviation. See also: Normal distribution
and bias, 474, 485
and pruning, 577–578
and standard error, 6, 119, 396
with LLSLS, 483, 487, 491, 498
with stratified sampling, 301, 304, 310
Standard error, 6, 31
See also: Variance reduction
and the accumulator object, 72–74
and bias, 508
and error trapping, 111
estimated as standard deviation, 119
Standard error (continued)
and importance sampling, 346–348
and LD sampling, 314–315
and residual error, 422, 428, 436
stratified sampling, 301
Standard modules
and events, 120
globals and persistent state, 40, 50, 551–553
as proto-objects, 553–556
as a singleton object, 228, 564–565
splitting code between, 39, 44, 45, 48
State
arrays and multidimensional state, 263, 265–268
Booleans as state, 97, 107, 111–113, 256
class modules and state, 40
the Collection object and states, 261, 262–263
Doubles state, 265–268, 273–277
and encapsulation, 265, 272
globals and persistent state, 40, 50, 551–553
of an object, 40, 55, 553–554, 616
and slice representations, 268–277
1-factor, 272–273
n-factor, 273–277
plain state, 273–275
structured state, 275–277
state representations, 265–268
state space, 11, 14, 450
and Static variables, 40, 550, 554
user defined types and state, 265–266
wrapping a state, 270
See also: State variables
State space, 11, 14, 450
State variables
and market values, 12
and modelling, 8–10
number of, 11–12, 582, 602
short rate models
See Cox, Ingersoll and Ross model
See Fong and Vasicek model
See Gaussian affine models
See Vasicek model
stochastic volatility, as a state variable
See Heston model; Fong and Vasicek model
traded asset, as a state variables
See Black-Scholes model; Heston model; Merton jump-diffusion model
Static. See Static variables
Static initialization idiom, 47–48, 50, 555, 616
in Stat_ran0(), 33
performance, 244, 245–247
problems with, 48, 254, 257
Static variables
and globals, 33, 245–247
lifetimes, 47, 257
Reset, 257
performance, 245–247, 254
and state, 40, 550, 554
and the Static initialization idiom, 33, 47–48
std::cout(), 138
std::exp(), 241
Step, 29, 241, 247, 576
Stochastic differential equations (SDE)
See also: Discretization; Simulation
CEV (Constant elasticity of variance). See CEV model
Correlation. See Correlation
Cox, Ingersoll and Ross (CIR) process. See Cox, Ingersoll and Ross model
densities. See Distributions
distributions. See Distributions
exact solutions. See Exact solutions, to SDE
exotic processes, 416
Feller condition. See Feller condition
Feynman-Kac representation, 581
Filtration, 3, 4
Fong and Vasicek, 325
geometric Brownian motion. See Geometric Brownian motion
Heston process. See Heston model
hitting time. See Hitting time
iterated Itô integral. See Iterated Itô integral
linear growth condition. See Linear growth condition
Lipschitz condition. See Lipschitz condition
martingale, in no-arbitrage pricing framework, 10
measure. See Measure
Merton jump-diffusion. See Merton jump-diffusion model
moments. See Moments
numeraire. See Numeraire
optional sampling theorem, 492
Ornstein-Uhlenbeck process. See
Ornstein-Uhlenbeck process
Poisson process. See Poisson process
sample space, 4, 11
state space, 11, 14, 450
state variables. See State variables
stochastic process. See Stochastic process
stopping time. See Stopping time
Wiener process. See Wiener process
Stochastic mesh method, Bermudan option valuation, 457
Stochastic process
for asset values, 3, 14
discretization of, 389
and evolution, 261
Lévy processes, 8
See also: Discretization; Models; Simulation;
Stochastic differential equations
Stochastic volatility models
See Heston model; SABR model
See also: Fong and Vasicek model
Stock models
See Black-Scholes model
See CEV (Constant elasticity of variance) model
See Jump-diffusion models
See Merton jump-diffusion model
See Shimko, Tejima and Van Deventer model

Stopping time
for an American put, 454
sampling at stopping times, 491
stopping time control variates, 491–494

Storage, 13, 14
and evolution, 294–295, 296
moment matching, 394, 395
See also: Files; VBA containers

Strategy pattern, 562–563, 617
and StopWatches, 83
See also: Design patterns

Stratified sampling, (Chapter 18), 291–305
See also: Low discrepancy sampling
American put, 487–488
average rate options, 298–304, 443–444
Bermudan put, 482–483, 487–488
Brownian bridge, 292–293
combined with antithetic sampling, 304
evolution, direction of, 294–295
and Heston, 316, 441
ITM options, 302
and moment matching, 291
of normal variates, 292, 312, 441
correlated normals, 441
quadratic option, 442–443
and standard error, 301
stratification times, 302–304
of uniform variates, 291–292, 297–298
Wiener process, 295–298
See also: Variance reduction
Stratonovich integral, 406

Straw man, 176

Streams, I/O, 19, 78, 91, 218–222
in C++, 137, 138
and the TextStream object, 138–141
See also: Channels

Striding, 268–270, 617
performance, 273–274

String
casting
and Dates, 145, 148
from String, 41
to String, Cstr(), 170
concatenation, 241
and DLLs, 545
and Err.Raise, 39, 40
fixed length Strings, 263, 546
input, 18, 92, 127, 164, 200
from file, Input(), 144
as keys
in a Collection, 263
in a Dictionary, 262
output, 170
to file, 141
Line Input #, 144, 148
and Run(), 83–84
and telepathy, 194, 225
and the TextStream object, 140–141
and UDTs, 145, 149
utility functions, 532
variable length Strings, 145, 149

Strong convergence criteria, 399–400
Strong typing, 617
in C++, 40
in VBA, 40, 242, 244

Stubs, 617
and application building, 25–29, 570–572, 585
and interfaces, 84–86, 130, 557
and procedure call performance, 242
and the VBA IDE, 121, 557

Sub-optimal exercise error, 449, 454–455, 471, 474, 487

Subs, 26, 45, 621
See also: Functions; Procedures

Successive over-relaxation (SOR), 595–602
See also: PDE methods

SV. See Stochastic volatility models
SVD. See Singular value decomposition

swap(), 217–218

Synonym classes, 195–196
Syntactic sugar, 42, 531, 556, 617

t-distribution, 536
Tab-delimited files, 144
Telepathic guarantee, 166
Telepathy, 617
and DLLs, 546
and Enums, 155, 166
and input/output, 127, 225
removal of, 194, 265

Template pattern, 562–563, 617
See also: Design patterns

Terminally modified IS density, 351–352
See also: Importance sampling

TextStream object, 140–141
file output, 141
and the FileSystemObject object, 138–139
and streams, 138, 139

this, 66

ThisWorkbook
factory code, 183–188, 197–203
I/O, 166, 170. See also: Front-end

VBProject object, 180–183

Tracking, 62–63, 617
meta-class data, 214, 225
Traded asset, models with
See Black-Scholes model; Heston model; Merton jump-diffusion model
Transformed SDE, discretization of, 407–408
Transforming a PDE, 584–585
Transparency, 271, 557, 615
Trapezium rule, 543
and discount factors, 391, 425
performance, 428
library object, IntegratorTrapezium, 542
See also: Numerical integration
Tricky Code, 234, 248, 253, 272, 273, 617
Tri-diagonal solvers. See Linear equation methods, tri-diagonal solvers
Trinomial lattice. See Lattice methods, trinomial
Type. See User defined types (UDTs)
Type conversion
to Date, 148
to Long, 238
of objects, 553–556
See also: Casts; Conversion constructor
UDT. See User defined types (UDTs)
Uniform distribution
and antithetic sampling, 284
and stratified sampling, 291–292, 297–298
Uniform variate generators
Mersenne twister, 251
performance, 246, 249–251
ran0(), 74, 238, 246–247
ran2(), 246, 247, 535
See also: Rnd()
Unit hypercube
and antithetic variates, 283, 284
and LD sampling, 307, 310
and stratified sampling, 297, 298
User defined types (UDTs)
and complex numbers, 540
as data members, 59
and encapsulation, 45
and file I/O, 145
random access files, 145
serialization, 153–155, 352–353, 360
and input, 46–47
naked UDTs, 265, 267
and objects, 55, 57, 549, 553–556
performance, 266–268, 275–277
and states, 265–266
and VBA containers, 261–263, 270–271
wrapped UDTs, 267
User interface. See Front-end
Validation, 40–45
absence of, 28, 35, 39. See also: Programming level, level 1
by casting, 185, 201
and input, 43–44, 46, 92, 203
and state, 40
validation functions, 41–44, 127, 216, 533
See also: Index to implementations, Utility code modules
See also: Programming level, level 2; Error handling
Variable length Strings, 145, 149
Variance gamma (VG), 292
Variance reduction methods
antithetic variates. See Antithetic variates
control variates. See Control variates
in the Heston model. See Heston model, variance reduction
importance sampling. See Importance sampling
low discrepancy sampling. See Low discrepancy sampling
spectral decomposition, 294
stratified sampling. See Stratified sampling
Variant
and Line Input #, 144
performance
casting, 244
and typing, in procedures, 243
sub-types, 56, 244
and UDTs, 262
and validation, 41, 533
See also: Variant arrays
Variant arrays, 262, 264, 270
and Dictionary object keys, 262
performance, 275–276
Vasicek model, 400
as auxiliary model, 326
and bias correction for CIR, 392
conditional discount factor, 426, 428
exact solution of SDE, 427
explicit solutions
bond options, 326
term structure, 17
Fong and Vasicek, as perturbation of, 325
long rate, 427
simulation. See Simulation, Ornstein-Uhlenbeck process
VB2TheMax, 536
VBA. See also: Excel
comparison with C++. See C++, comparison with VBA
compiler. See VBA compiler
constants. See VBA constants
containers. See VBA containers
control statements. See Control statements
enumerative types. See Enums
See also: VBA enumerative types
error messages, 263, 519, 521
events. See Events
functions
  See Functions; Subs
  See VBA intrinsic file functions and statements
  See VBA intrinsic functions
  See also: Index to library functions
  See also: Procedures

modules. See Modules
OOP reserved words. See VBA OOP reserved words
  See also: VBA reserved words
performance. See VBA reserved words
reserved words. See VBA reserved words
  See also: VBA OOP reserved words
Scripting runtime library. See Scripting runtime library
settings, 517
types. See VBA primitive types
VBA in-built objects. See VBA intrinsic objects
VBE object library, 179–183
VBA compiler
clean recompile, 190, 518–521
compile-time errors, 40, 42, 194, 223, 244
compiler problems, 518–521
VBA constants
vbCrLf, 4
vbObjectError, 41
  See also: Err object
vbTab, 141, 148
  See also: Tab-delimited files
VBA containers
arrays. See Arrays
Collection object. See Collection object
Dictionary object. See Dictionary object
plain old data (POD). See Plain old data
user defined types (UDTs). See User defined types
Variant arrays. See Variant arrays
VBA enumerative types
vbext_ComponentType, 181
vbext_ProcKind, 182
VBA intrinsic file functions and statements, 143–145
ChDir, 143
ChDrive, 143
Close, 143
CurDrive(), 143
Dir(), 143, 148, 149
EOF(), 143
  and file input, 147–148
FileAttr(), 144
FileCopy, 143
FileDateTime(), 144
FileLen(), 144
FreeFile(), 144, 147
Get, 138, 144
  and random access files, 145, 150, 151
GetAttr, 144
Input(), 144
Input #, 144, 145, 148
Kill, 143, 149
Line Input #, 144, 148
Loc(), 143
LOF(), 144
  and random access files, 150
MdDir, 143
Name, 143
Open, 130, 143
  and random access files, 149, 150
  and sequential files, 147
Print #, 144
Put, 144
  and random access files, 145, 150, 151
Reset, 143
RmDir, 143
Seek(), 143
SetAttr, 143
Write #, 144
VBA intrinsic functions
Abs(), performance, 241
Atn(), 536
CDate(), 148
CDbl()
  and casting, 554
  and file input, 148
  and validation, 41
Chr(), 45, 140, 144
  See also: VBA, constants
CLng()
  and bit arithmetic, 310
  performance, 238, 247
  and validation, 41
Cos(), performance, 241
CreateObject()
  and the Dictionary object, 262
  and FileSystemObject, 139
CStr(), casting, 170
Exp(), performance, 238, 241
Fix(), performance, 238
Format(), 144
IIF(), performance, 239
Int(), performance, 238, 247
IsNumeric(), and validation, 41
IsObject(), 66
Len(), and files, 149
Log(), performance, 237–238, 241
MsgBox(), 44 et cetera
Now(), 56
Randomize(), 34
Rnd(), 22, 34
Sgn(), performance, 241
Sin(), performance, 241
Split(), and file input, 148
Sqr(), performance, 237
VBA intrinsic functions *(continued)*

<table>
<thead>
<tr>
<th>Function</th>
<th>Performance</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round()</td>
<td></td>
<td>238</td>
</tr>
<tr>
<td>Tan()</td>
<td></td>
<td>536</td>
</tr>
<tr>
<td>Time()</td>
<td></td>
<td>56</td>
</tr>
<tr>
<td>Timer()</td>
<td></td>
<td>56</td>
</tr>
<tr>
<td>and StopWatch</td>
<td></td>
<td>65, 88</td>
</tr>
</tbody>
</table>

VBA intrinsic objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>See Collection object</td>
</tr>
<tr>
<td>Dictionary</td>
<td>See Dictionary object</td>
</tr>
<tr>
<td>Err</td>
<td>See Err object</td>
</tr>
<tr>
<td>FileSystemObject</td>
<td>See FileSystemObject object</td>
</tr>
<tr>
<td>TextStream</td>
<td>See TextStream object</td>
</tr>
</tbody>
</table>

VBA OOP reserved words. See also: VBA reserved words

<table>
<thead>
<tr>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>120</td>
</tr>
<tr>
<td>Implements</td>
<td>See Implements</td>
</tr>
<tr>
<td>Is,</td>
<td>66</td>
</tr>
<tr>
<td>Me,</td>
<td>66, 92, 158</td>
</tr>
<tr>
<td>New</td>
<td>See New</td>
</tr>
<tr>
<td>Nothing</td>
<td>See Nothing</td>
</tr>
<tr>
<td>Property</td>
<td>See Property</td>
</tr>
<tr>
<td>RaiseEvent</td>
<td>120</td>
</tr>
<tr>
<td>Set</td>
<td>See Set</td>
</tr>
<tr>
<td>With</td>
<td>66</td>
</tr>
<tr>
<td>WithEvents</td>
<td>39, 120, 121</td>
</tr>
</tbody>
</table>

VBA primitive types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>See Boolean</td>
</tr>
<tr>
<td>Date</td>
<td>56, 145, 148</td>
</tr>
<tr>
<td>Double</td>
<td>See Double</td>
</tr>
<tr>
<td>Integer</td>
<td>See Integer</td>
</tr>
<tr>
<td>Long</td>
<td>See Long</td>
</tr>
<tr>
<td>Single</td>
<td>See Single</td>
</tr>
<tr>
<td>String</td>
<td>See String</td>
</tr>
<tr>
<td>Variant</td>
<td>See Variant</td>
</tr>
</tbody>
</table>

VBA project, Reset

<table>
<thead>
<tr>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset</td>
<td>257</td>
</tr>
</tbody>
</table>

VBA reserved words. See also: VBA OOP reserved words

<table>
<thead>
<tr>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByVal</td>
<td>See ByVal</td>
</tr>
<tr>
<td>Call</td>
<td>621</td>
</tr>
<tr>
<td>Const</td>
<td>See Const</td>
</tr>
<tr>
<td>Dim</td>
<td>See Dim</td>
</tr>
<tr>
<td>Enum</td>
<td>153–155, 166, 352–353, 605, 610</td>
</tr>
<tr>
<td>Function</td>
<td>See Functions</td>
</tr>
<tr>
<td>Option</td>
<td>Explicit, 27, 40, 622</td>
</tr>
<tr>
<td>Option</td>
<td>Private, 58</td>
</tr>
<tr>
<td>Private</td>
<td>See Private data</td>
</tr>
<tr>
<td>Property</td>
<td>See Property</td>
</tr>
<tr>
<td>Public</td>
<td>See Public data</td>
</tr>
<tr>
<td>Redim</td>
<td>265, 266</td>
</tr>
<tr>
<td>Redim Preserve</td>
<td>147, 166, 174</td>
</tr>
<tr>
<td>Static</td>
<td>See Static variables</td>
</tr>
<tr>
<td>Step</td>
<td>29, 241, 247, 576</td>
</tr>
<tr>
<td>Sub</td>
<td>See Subs</td>
</tr>
</tbody>
</table>

**Type. See User defined types (UDTs)**

VBA settings

<table>
<thead>
<tr>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbCrLf</td>
<td>44</td>
</tr>
</tbody>
</table>
| VBE object library, 179–183
  access to the library, none, 211, 228
  accessing the library, 179, 517
  CodeModule object, 180–183
  VBCOMPONENTS object, 180
  VBProject object, 180, 182
  vbext_ComponentType, 181
  vbext_ProcKind, 182
  VBIDE object, 179, 180, 188, 559, 562
  vbObjectError, 41
  See also: Err object
  VBProject object, 180, 182
  vbTab, 141, 148
  VG. See Variance gamma
  Virus checkers, 190, 211
  Visibility, 31, 58, 216
  See also: Scope
  Visual basic for applications. See VBA
  Visual basic editor. See VBE object library
  Visual basic integrated development environment. See VBIDE object

Weak convergence criteria, 399–400
Weak will, 67
While-Wend, 240
Wiener process
  and antithetic variates, 284
  and bridge distribution, 292–293
  and geometric Brownian motion, 3, 25
  and stratified sampling, 295–298
With, 66
WithEvents, 39, 120, 121
Workbook events, Workbook_Open()
  and the factory, 183, 188, 193
  and initialization, 554
Wrapper objects, 617
  and containers, 263, 271
  and the decorator pattern, 124
  and DLLs, 546
  option wrapper object example, 494
  and polymorphism, 153
  and RAII, 76
  wrapping a state, 263, 270
  See also: Design patterns

“X”. See Comments

Yucky. See Yukky application
Yukky application. See Programming level, level 0, yukky

Zero lock-out Heston (ZLH), 435–436, 438–445
Zhang, Kai, xvii, 387
ZLH. See Zero lock-out Heston