

Index

- '#' (comments), Python basics 195
- '+' (concatenation operator), Python basics 194
- '=' (assignment operator), Python basics 194–5
- '[]' (lists), Python basics 196–7
- abscissae 31–4, 63–7
 - see also* curves
- abstraction 201–3, 208–9
- access mechanisms, Python basics 203, 209
- accrual periods, concepts 77–82
- accrual_basis 79–82, 178–87
- accrual_def 77–9
- accrual_end_date 79–82, 117–22
- accrual_holiday_centres 80–2
- accrual_shift_method 80–2
- accrual_start_date 79–82, 90
- accrued_coupon 153–6
- acos 48–9
- ActiveState Python distribution 8, 133
 - see also* Python
- actual_cpn 154–6
- add 66–7, 101–22, 170, 172–6
- AddConstant 170, 172–6
- add_constant 66–7, 106–22, 170, 172–6
- AddCurve 170–6
- add_curve 66–7, 101–22, 170–6
- add_event 94–8
- AddSurface 170, 173–6
- add_surface 66–7, 101–22, 170, 173–6
- adjuvants, concepts 82–4, 147–52, 153–6, 177–87
- adjuvant_table 82–4, 147–52, 153–6, 180–7
- aliases, Python basics 196–7
- all_observables 72–9
- alphas 132–42
- American options 219–20
 - see also* options
- analytic/semi-analytic pricing formulae 99
- antithetic 112–22
- append 46, 47–9, 54–7, 73–9, 81–3, 87–90, 94–8, 124–8, 132–42, 151–2, 155–6
- appendices 4, 191–220
- arbitrage-free pricing 217–18
- args 121, 188
- array 38–41, 57, 115–22, 133–42, 152–6, 179–87
- arrays 2–3, 17–26, 38–44, 67, 115–22, 133–42, 152–6, 170–6, 179–87, 196–7
 - see also* NumPy
 - associated arrays 196–7
 - C++ 17–26
 - concepts 17–26, 152–6, 196–7
 - dictionaries 119–22, 181, 196–7, 215–16
 - manipulation examples 23–6
 - MultiArray multidimensional array types 17–19
- ASCII 172
- Asian options 99
- assert 30–1, 61, 67, 110–22, 150–2
- assert_seq_close 110–22, 150–2, 155–6
- assignment operator ('=') 194–5
- assign_zero 23–6
- associated arrays, Python basics 196–7
- at-the-money option prices 61
- atm_option_test 61
- attributes 70–9
- back substitution 42–6
- barrier options 99
- Base 209–11
- basis_30/360 14–16, 81–2, 89–91, 96–8
- basis_act_360 14–16, 73–4, 88–91, 111–22
- basis_act_365 14–16
- basis_act_act 14–16
- batch interpreter mode, Python 193–4
- berm 126–8, 130–42, 150–2
- Bermudan swaptions 4, 101–2, 115–16, 126–8, 132–42, 145–52, 157, 168–89
 - concepts 145–52, 157, 168–89

- Bermudan swaptions (*cont.*)
 definition 145–6
 Microsoft Excel 168–89
 pricing models 4, 101–2, 115–16, 126–8,
 132–42, 145–52, 157, 168–89
- bermudan_tests 149–52
- bibliography 221
- bid–offer spreads 100–1
- bind.hpp 20
- bisect 35–6
- bisection root-finding method
see also root-finding algorithms
 concepts 35–6, 37
- bjam.exe 8
- Black–Scholes option pricing formula
 COM servers 165–8
 concepts 5–9, 28–9, 111–22, 165–8
 Microsoft Excel 165–8
- BlackScholes, COM server 165–8
- black_scholes 165–8
- Blitz++ libraries
see also C++
 concepts 7–8, 19–20
 definition 19
- bonds
 coupons 79–82, 89–91, 96–8, 123–8, 145–57,
 177–87
 zero coupon bonds 100–1, 110–22, 146–57,
 177–87, 217–18
- bool 212
- Boost.Build system 8
- Boost.Date_Time library, concepts 11–16,
 159–63
- Boost.MultiArray multidimensional array types,
 concepts 17–19
- BOOST_PREVENT_MACRO_SUBSTITUTION
 18–19
- Boost.Python library 4, 7–9, 19–26, 36, 159–63,
 207–16
see also C++
 basics 207–16
 call policies 212–14
 classes 207–9
 concepts 7–8, 19–26, 159–63, 207–16
 constructors 207–9
 embedding 161–3, 214–16
 enums 214
 functions 212–14
 Getting Started Tutorial 207
 ‘Hello World’ C++ function 207
 inheritance 209–11
 methods 207–9
 operators 212
 tools 207–16
- BOOST_PYTHON... 207–16
- BOOST_TEST 163
- bound 29–31, 52–7, 64–7
- bounds 53–7
- Box, Don 165
- Brownian motion
 concepts 57, 113–14, 217–18
 semi-analytic conditional expectations 57–61
- building ppf packages, concepts 7–9
- built-in data types, Python basics 1, 195–7
- business days 14–16
- C++ 2, 3–4, 7–9, 11–26, 36, 98, 121–2, 157, 203,
 207–16
see also ppf package
 access mechanisms 203, 209
 arrays 17–26
 Blitz++ libraries 7–8, 19–20
 Boost.Python library 4, 7–9, 19–26, 36,
 159–63, 207–16
 C++/Python ‘Hybrid Systems’ 4, 159–63
 enum types 14–16, 214
 Indexable concepts 3
- nth_inma_of_year 161–3
- overloaded functions 212–14
- pointers 212–14
- policies 121–2
- Python interoperability benefits 2, 3–4, 7–9,
 11–26, 157
 references 212–14
 STL functions 29
- C, Python interoperability benefits 2
- calibration routine, Hull–White model 100–1
- call 32–4, 45–6, 63–7, 83–5, 98, 116–22,
 125–8, 130–42, 146–57, 180–7
- call options 4, 5–9, 86–91, 93–8, 115–22, 123–43
see also options
- call policies, Boost.Python library 212–14
- callable 86–91, 96–8, 123–43, 150–2
- callable structures 4, 86–91, 93–8, 115–22,
 123–43
- cancellable 86–8
- candidate_date 89–91
- cap floaters 115–16
- cash flows *see* flows
- ccys 90, 102–22, 182–7
- class, Python basics 201–3
- classes 2–3, 119–22, 151–2, 155–6, 188–9,
 201–3
 Boost.Python library 207–9
 concepts 201–3
 examples 201–3
 factory classes 119–22, 151–2, 155–6, 188–9
 pricing models 3
 Python basics 2–3, 201–3
 trade representations 3
- cle_exercise 115–22

- CLSIDs 166–8, 177–87, 188–9
 coeffs 50–1
 collars 123–8
 com 5–6, 165–89
 COM servers 4, 5–6, 98, 165–89
 Black–Scholes option pricing formula
 165–8
 BlackScholes 165–8
 class registration 169
 common utilities 168–9
 concepts 165–89
 exceptions 166–8, 169
 market server 169–76
 pricer server 187–9
 trade server 176–87
 COMException 166–8, 169
 comments ('#'), Python basics 195
 common market information
 see also environment; observables
 concepts 3, 63–7, 69, 99–100, 169–76
 compilers 7–8
 Component Object Module *see* COM...
 component-based programming in Python
 99–122
 see also Hull–White model
 concatenation operator ('+') 194
 conditional expectations, integration 57–61
 constants 65–7, 83–5, 106–22, 147–52, 153–6,
 170, 172–6, 177–87, 212
 constructors, Boost.Python library 207–9
 control flow statements, Python basics 197–200
 controller 97–8, 124–8, 130–42, 146–57
 controller module
 concepts 93, 97–8, 124–8, 130–42, 146–57
 definition 93
 copy 135–42
 core 5–6, 69–84, 93–8, 124–8, 149–56, 165–8,
 176–87
 correlation surfaces 63, 65–7, 101
 cos 48–9, 205
 coupon_leg 155–6
 coupon_observables 89–91, 96–8, 120–2
 coupon_rate 89–91, 96–8, 120–2, 146–57,
 177–87
 coupons 79–83, 89–91, 96–8, 123–8, 145–57,
 177–87
 coupon_shift_method 89–91, 96–8, 177–87
 cpn 126–8, 131–42, 146–57
 create_cached_moments 57–61
 create_callable_pricer 150–2
 CreateEnvironment 170–1
 create_environment 149–57, 171
 create_exercise_schedule 149–52
 create_fixed_leg 149–57
 create_funding_leg 149–57
 CreateHullWhiteLatticePricer 188–9
 CreateLeg 177, 180–7
 create_pricer 149–57
 create_tarn... 155–6
 CreateTrade 177, 180–7
 create_variable 102–22, 146–57
 cs 52–61
 cscript.exe 167
 csi 55–7
 cT 58–61
 cubic roots, concepts 3, 46–9, 53–7
 cubic spline interpolation
 see also interpolation schemes
 concepts 33–4, 171–6
 cubic_roots 47–9, 53–7
 cubic_spline 34
 curve 63–7, 72–9, 100–22, 148–57, 170–6
 curves
 see also abscissae; interpolation...;
 ordinates
 concepts 3, 63–7, 100–22, 170–6
 discount-factor functions 3, 6, 63–7, 100–22

 data model: concepts 3, 69–91
 data structures
 see also exercise...; flows; observables; trade...
 adjuvants 82–4, 147–52, 153–6, 177–87
 concepts 3, 69–91, 176–87
 date 64–5, 89–91, 96–8, 111–22, 151–2,
 160–3, 168–9
 date conversions, COM servers 168–9
 date_time, concepts 5–9, 11–16, 64–7, 70–84,
 86–91, 95–8, 110–22, 151–2, 159–63,
 168–9, 176–87
 Date_Time library, concepts 11–16, 159–63
 DAX 70
 day count conventions 14–16, 72–9, 80–3,
 89–91, 111–22
 day_count_basis 14–16, 72–9, 80–3
 decref 19–26
 deep_in_the_money_test 150–2
 deep_out_the_money_test 150–2
 def statements, Python basics 200–5
 default arguments, constructors 207–9
 delta 36–7
 derivatives 36–7
 see also forward...; options; swap...
 Derived 209–11
 dictionaries, Python basics 119–22, 181, 196–7,
 215–16
 dimensions 24–6
 discount-factor functions
 see also curves
 concepts 3, 6, 63–7, 97–8, 100–22, 131–42,
 146–57, 217–18
 Hull–White model 100–22, 146–57
 discount_factor 100–22

- doctest testing module, concepts 6–7
dot 38–9, 42–4
duration 78–9, 80–2, 86–91, 96–8, 120–2,
178–87
dynamic type system of Python 2–3
- early exercise premiums 4, 85–91, 132–3,
219–20
elem 83–5
elif examples 14–16, 30–1, 35–7
else statements, Python basics 199–200
embedding, Boost.Python library 161–3, 214–16
encapsulation support, Python 2–3, 58–61, 209
end 80–91, 96–8, 120–2, 177–87
endif 19–23
end_of_start_year 16
enforce_no_exercise_stubs 90, 125–8,
129–42
enforce_single_currency 90–1
enum types 14–16, 214
env 67, 97–8, 100–22, 125–8, 129–42, 146–57
env_id 188–9
environment 65–7, 97–8, 100–22, 169–76
see also common market information; curves;
surfaces
epsilon 18–19
equal_range 30–1, 55–7
equity spot prices 63, 69–70
errors.hpp 20
Essential COM 165
Euler's scheme 113–14
EUR 70, 87–91
European options 5–9, 149–52
see also options
europeans 150–2
european_symbol_table_listener 128,
150–2
evaluate_regression 133–42
event 93–8, 126–8, 130–42, 146–57
events
see also timeline
concepts 93–8, 126–8, 129–42, 146–57
definition 93
event_type 93–4
evolve 112–22, 129–42
evolve component, pricing models 112–15,
129–42
except 27–8
exercise 85–91, 94–8, 115–22, 132–42,
177–87
exercise component, pricing models 115–22,
123–8
exercise dates 88–91
exercise schedules
concepts 88–91, 177, 179–87
definition 69
- exercise_boundary_regression
137–42
exercise_date 86–91
exercise_event 93–8, 138–42
exercise_helper 134–42
exercise_info 88–91
exercise_opportunity 94–8
exercises
see also notification dates
concepts 3, 69, 85–91, 93–8, 115–22, 123–8,
131–42, 176–87, 219–20
definition 69, 85–7
early exercise premiums 4, 85–91, 132–3,
219–20
exercise_schedule 88–91, 95–8, 180–7
exercise_type 86–91, 94–8, 120–2, 142,
150–2, 180–7
exotic financial instruments, pricing approaches
99, 123–8, 145–57
exotic_first_flow_pv 156
exp 28–9, 52–61, 109–22, 170–6
expectation calculations
see also integration
concepts 3, 49–61, 108–9
conditional expectations 57–61
expected_libor 117–22
expected_swap 117–22
Expecting 6–7
expect_non_null 21–6
expiries 64–7, 101–22, 170–6
explanatory_variables 135–42,
150–2
expressiveness aspects of Python 1
ex_sched 88–91, 96–8, 120–2, 150–2
extensibility aspects of Python 1–4, 7–9, 11–26
extract 162–3, 215–16
- fabs 37, 60–1, 114–22
factors 63–7, 109–22
factory 151–2, 155–6, 188–9
factory classes 119–22, 151–2, 155–6, 188–9
fee 85–7
fee_ccy 85–7
fill 45–6, 101–22, 133–42, 146–57
filler component, pricing models 45–6, 101–22,
133–42, 146–57
filtration 57–61, 145–52, 217–18
final_date 89–91
final_important_date 88–91
financial contracts
see also trades
concepts 69, 88–91
financial engineering, Python uses 1–4, 11–26
financial instruments
see also bonds; derivatives; options; swap...
path-dependency issues 99, 123–43, 157

- pricing approaches 99, 108–9, 123–43, 145–57
types 99, 123–8
- financial payoffs 49–61, 85–6, 98, 145–56,
176–87
- finite differences 123
see also lattice-based pricing framework
- first_axis 64–5
- first_imm_after 14
- first_imm_before 13–14
- fit 133–42
- fit_fos 44–6, 133–42
- fitted_fos 132–42
- fix 70–9, 105–22, 147–57
- fixed leg, swaps 74–9, 145–52, 181–7
- fixed_flows 75–9, 105–22
- fixed_leg_payoff 145–52, 181–7
- fixed_pay_basis 75–9
- fixed_pay_holiday_centres 75–9
- fixed_period... 75–9
- fixed_pv 77, 106–22, 149–52
- fixed_rate 145–57
- fixed_shift_convention 75–9
- fixing 69–79, 147–57
- float 43–4, 212
- float_flows 75–9, 106–22
- floating leg, swaps 74–9, 145–52, 181–7
- floating point numbers, Python basics 194–6
- float_leg_payoff 146–57, 181–7
- float_pay_basis 75–9
- float_pay_holiday_centres 75–9
- float_period... 75–9
- float_pv 106–22, 149–52
- float_shift_convention 75–9
- floor 155–6
- flow 79–83, 93–8, 146–57, 177–87
- flow_id 69–79
- flows 79–83, 85–6, 88–9, 90–1, 93–8, 105–22,
123–43, 146–57, 177–87
- flows
see also legs; observables
concepts 3, 6, 69, 79–83, 88–91, 93–8,
105–42, 176–87
definition 69, 74, 79
- for statements, Python basics 197–200
- foreign exchange rates 63, 70–9, 80–3, 86–7,
90–1, 93–8, 105–22, 135–42, 146–57,
177–87
see also reset currencies
- Fortran 77/90 19
- forward 72–9, 149–57
- forward contracts 72–9, 104–5
- fos 132–42
- fsT 60–1
- ftT 57–61
- functional programming idioms, Python
support 1
- functionally orthogonal components,
Hull–White model 99–122
- functions
Boost.Python library 212–14
overloaded functions 212–14
Python basics 2–3, 200–1
- funding_first_flow_pv 156
- funding_leg... 148–57
- fund_pv 77
- gauss 27–8, 46, 113–22
- Gaussian distributions 27–8, 46, 51–7,
102–22
- GBP 70–9
- generalized linear least squares, concepts 3,
44–6, 133–42
- generalized_least_squares 44–6,
133–42
- GenerateAdjuvantTable 177,
179–87
- generate_adjuvant_table 82–4, 147–52,
153–6, 179–87
- GenerateExerciseSchedule 177,
179–87
- generate_exercise_table 86–91, 95–8,
120–2, 151–2
- GenerateFixedCoupon Observables
177, 181, 182
- generate_fixed_coupon_observables
89–1, 96–8, 120–2
- GenerateFlows 177, 181, 184–5
- generate_flows 74–9, 80–3, 89–1, 96–8,
120–2, 178–87
- GenerateLiborCoupon Observables
181
- GenerateLiborObservables 177–8, 181,
182–3
- generate_libor_observables 72–9,
89–91, 96–8, 178–87
- generate_observables 72–9, 95–8, 120–2
- generate_swap_observables 77–9
- get 22–6, 94–8, 114–22
- get_adjuvant_table 147–52, 153–6
- get_date 13–16, 160–3
- get_environment 97–8, 146–57
- get_event 97–8, 146–57
- get_model 97–8, 146–57
- getopt 27–8
- get_override 160–3
- Getting Started Tutorial, Boost.Python library
207
- get_trade 97–8, 154–6
- Girsanov's theorem 218
- GNUPlot 2
- the Greeks 142–3
- gregorian 160–3

- gregorian_calendar_base 15–16, 160–3
 GUI toolkits, Python integration benefits 2
- Hammond, Mark 165
- handle_zero_derivative 36–7
- has_constant 66–7
- has_curve 66–7
- has_exercise_schedule 131–42
- has_key 94–8
- has_surface 66–7
- ‘Hello World’ C++ function, Boost.Python library 207
- help 27–8
- helper functions 93–8, 161–3
- high-bias estimators 131–2
- high-level aspects of Python 1
- historical_df 97–8, 146–57
- holiday_centres 15–16, 75–84, 86–8
- holidays 14–16, 75–84
- Hull–White model 3–4, 6, 65–7, 99–122, 145–57, 168–89, 217–18
 brief outline 217–18
 calibration routine 100–1
 component-based design 99–122
 concepts 4, 99–122, 145–57, 168–89, 217–18
 constants 65–7
 evolve component 112–15
 exercise component 115–22
 filler component 101–22, 146–57
 functionally orthogonal components 99–122
 mathematics 4, 99–100, 217–18
 Microsoft Excel 168–89
 model factories 118–22, 149–52, 155–6, 188–9
 requestor component 100–22
 rollback component 108–12
 state component 101–22, 145–52
- hull_white... 67, 100–22, 151–2, 155–6, 188–9
- hull_white_monte_carlo_model_factory 121, 151–2, 155–6, 188–9
- hwmodel 121–2
- hybrid financial instruments, pricing approaches 99
- ‘Hybrid Systems’ 4, 159–63
- if statements, Python basics 199–200
- IMM dates, concepts 12–16, 159–63
- immutable types, Python basics 195–7
- impl 63–7
- import directive, Python basics 203–5
- import_array 23–4
- in-the-money option prices 150–2
- incremental_distribution 102–22
- incremental_fill 102–22
- incremental_vol 102–22
- indented code, Python basics 198–200
- Indexable concepts 3
- indicator 54–7, 134–42, 154–6
- inflation rates 104–5
- information, common market information 3, 63–7, 69, 99–100, 169–76, 177–87
- inheritance, Python basics 122, 202–3, 209–11
- init 31–4, 51–7, 63–7, 69–91, 97–8
- input 21–2
- insert_symbol 97–8, 125–8, 130–42
- install 9
- installing ppf packages, concepts 7–9
- integers, Python basics 194–6
- integral 51–7
- integral_indicator 54–7
- integral_max 55–61
- integration
 Brownian motion 57–61
 concepts 49–61
 piecewise constant polynomial fitting 49–51, 57–61
 piecewise polynomial integration 51–61
 semi-analytic conditional expectations 57–61
- integrator 58–61, 108–22
- integrator_tests 59–61
- interactive interpreter mode, Python 193–4
- interest rates 6, 69–70, 70–9, 89–91, 101–2, 104–22, 143, 145–56, 177–87
 LIBOR rate 6, 69, 70–9, 89–91, 104–22, 143, 145–56, 177–87
 swap rates 70–9, 104–5
- international money market 12–16, 159–63
see also IMM...
- interoperability aspects of Python 1–4, 7–9, 11–26, 157
- interp 63–7
- interpolation... 31–4, 63–7, 106–22, 170–6
- interpolation schemes
see also curve...
 concepts 3, 29–34, 63–7, 106–22, 170–6
 cubic spline interpolation 33–4, 171–6
 linear interpolation 31–3, 171–6
 linear on zero interpolation 32–3, 171–6
 loglinear interpolation 32–4, 106–22, 171–6
- interpolation_base 31–4
- interpreters 2, 24–6, 45–6, 193–4, 208–9
- inv 38–9
- inverse floaters 115–16
- InvokePricer 189
- is_business_day 15–16
- is_exercise_event 138–42
- is_last_flow 153–6
- is_pay_event 138–42

- JPY 73–9, 81–3, 89–91, 96–8
- key 66–7, 100–22, 148–57
- keys, Python basics 196–7
- Kiusalaas, Jaan 38
- Kolmogorov’s strong law of large numbers 129
- lags, LIBOR rates 70–9
- lambda 19, 31, 36–7, 133–42, 152–6, 188–9
- Langtangen, Hans Peter 2
- last_important_date 69–79
- lattice 102–3, 108–22, 124–8
- lattice-based pricing framework 3, 6, 49–51, 67, 99, 101–22, 123–31, 142–3, 149–52, 168–89
- concepts 99, 102–3, 108–22, 123–31, 142–3, 149–52, 168–89
- rollback 108–22, 124–8
- lattice_pricer 125–8, 188–9
- leg 84–5, 90–1, 94–8, 120–2, 126–8, 130–42, 148–57, 177–87
- leg_id 93–8, 150–6
- leg_pv 126–8, 131–42
- legs 88–91, 95–8, 130–42, 177–87
- legs
- see also* flows
- concepts 3, 69, 74–9, 84–5, 88–91, 93–8, 120–2, 125–8, 130–42, 148–57, 176–87
- definition 69, 84
- swaps 74–9
- len 43–6, 50–1, 52–7, 81–5, 133–42
- leverage 156
- libor 147–57
- LIBOR rate 6, 69, 70–9, 89–91, 104–22, 123–8, 143, 145–56, 177–87
- concepts 70–9, 104–22, 123–8, 143, 145–52
- definition 71
- libor_basis 75–9
- libor_holiday_centres 75–9
- libor_observables 96–8, 105–22
- libor_rate 70–9
- libor_shift_convention 75–9
- likelihood ratios 143
- LIMITS 17–18
- linalg 38–44
- linear 31–4, 45–6
- linear algebra 3, 17–19, 33–4, 38–44, 49–51
- concepts 38–44
- matrix inversion 38–9
- matrix multiplication 38
- matrix pseudo-inverse 39
- singular value decomposition of a matrix 42–6
- solving linear systems 39–40
- solving tridiagonal systems 2–3, 34, 39–40
- solving upper diagonal systems 17–19, 40–4, 49–51
- linear equations 17–19, 33–4
- linear interpolation
- see also* interpolation schemes
- concepts 31–3, 171–6
- linear on zero interpolation
- see also* interpolation schemes
- concepts 32–4, 171–6
- linear_algebra 31, 34, 40–1, 50–1
- linear_fo 132–42
- linear_on_zero 32–4
- linspace 111–22
- Linux 8
- list comprehensions, Python basics 196–7
- lists, Python basics 3, 196–7, 215
- local volatility, Hull–White model 100–22
- local_vol 101–22
- locate 31–4
- log 32–4
- loglinear 32–4, 63–7, 106–22
- loglinear interpolation
- see also* interpolation schemes
- concepts 32–4, 106–22, 171–6
- lognormal distributions, random number generation 27–8
- lognormal_martingale_test 59–61
- lognormvariate 27–8
- Longstaff, F.A. 219
- low-bias estimators 131–2
- lower_bound 29–31, 52–7
- main 162–3, 166–8, 214–16
- make_array 23–6
- market 6, 63–7, 101–22, 169–76
- market server, COM servers 169–76
- MarketServer COM component 169–76, 181, 188–9
- market_server 169–76, 188–9
- Markovian models 123–8
- martingales 59–61, 110–22, 145–52, 217–18
- math 2–3, 6–9, 17–19, 23–6, 28–9, 31–4, 35–6, 40–1, 44–9, 50–61, 100–22, 132–42, 170–6, 205
- mathematics 3, 4, 6–9, 23–6, 27–61, 99–102, 217–18
- see also* interpolation schemes; linear algebra
- basic tools 3, 27–61
- concepts 27–61
- cubic roots 3, 46–9, 53–7
- generalized linear least squares 3, 44–6, 133–42
- Hull–White model 4, 99–100, 217–18
- integration 49–61
- quadratic roots 3, 46–9
- random number generation 3, 27–8, 45–6, 112–22
- root-finding algorithms 3, 35–7

- mathematics (*cont.*)
 standard normal cumulative distributions 3, 27–9, 31
 Matlab 2
 matrices 33–4, 38–44
 see also linear algebra
 matrix 38, 41–4
 matrix inversion, concepts 38–9
 matrix multiplication, concepts 38
 matrix pseudo-inverse, concepts 39
 max 135–42, 152–6
 max_dim 21–2
 maxflt 35–7
 maxits 35–7
 mean 51–61, 114–22, 127–8, 131–42
 mean reversion 65–7
 mean/variance 114–15
 memory management issues, Python benefits 1
 Mersenne Twister 19939 (MT19939) 27
 methods
 Boost.Python library 207–9
 Python basics 201
 Microsoft Excel 4, 165–89
 Microsoft Excel VBA 165, 167–8, 172–4, 181, 188–9
 Microsoft VBScript 167–8
 Microsoft Visual Studio C++ compiler versions 8
 Microsoft Windows
 COM servers 4, 5–6, 98, 165–89
 ppf package 8
 min 152–6
 min_dim 21–2
 mingw/gcc... 8
 misconceptions about Python 2–3
 model 97–8, 100–22, 124–8, 129–42, 146–57, 187–9
 model factories
 Hull–White model 118–22, 149–52, 155–6, 188–9
 Monte-Carlo simulations 121, 151–2, 155–6
 model_factories module 119–22
 modified_following 73–9, 80–2, 83–4, 86–91, 96–8, 111–22, 151–2
 module.hpp 11–12
 modules, Python basics 203–5
 moments 52–7
 moneyness_tests 150–2
 monotonic_with_target_test 156–7
 Monte-Carlo simulations
 callable trades 131–42
 concepts 3, 4, 6, 99, 101–2, 108–22, 123, 128–43, 150–7, 219–20
 evolve component 112–15, 129–42
 exercise component 115–22
 model factories 121, 151–2, 155–6
 non-callable structures 129–31, 142–3
 pricing framework 123, 128–43, 150–2
 monte_carlo... 112–22, 129–42, 150–2
 months 64–5, 74–9, 89–91, 111–22, 159–63
 MT19939 27
 multi-asset products 123
 multi_array, concepts 17–19
 MultiArray multidimensional array types 17–19
 mut 59–61
 mutable types, Python basics 195–7
 muT 59–61

 N 28–9, 31, 34, 52–7
 see also standard normal cumulative distributions
 namespaces 2–3, 11–26, 150–3, 208–9, 215–16
 ‘natural spline’ 34
 Newton–Raphson root-finding method
 see also root-finding algorithms
 concepts 36–7
 newton_raphson 36–7
 *NIX users 8
 non-callable structures
 Monte-Carlo simulations 129–31, 142–3
 pricing framework 129–31, 142–3
 normal_distribution 51–7, 59–61, 102–22
 normalisation 137–42
 notification dates, exercises 88–91, 95–8
 notification_date 85–91, 95–8
 notional 79–82, 146–57
 n_quadratic_fo 132–42
 nt 58–61
 nth_imm_of_year 13–14, 159–63
 nth_kday_of_month 13–14, 159–63
 nT 58–61
 nT 58–61
 num... 133–42
 numeraire 104–22, 135–42, 146–57
 numeraire_rebased_bonds 67, 104–22, 146–57
 numerical analysis 3–4, 27–61, 99–122, 123–43, 168–89
Numerical Methods in Engineering with Python (Kiusalaas) 38
 numerical techniques 99–122, 123–43, 145–57, 168–89
 see also Hull–White model; lattice-based...; Monte-Carlo...
 num_explanatory_variables 115–22
 num_obs 133–42
 NumPy 3, 8–9, 19–26, 38–46, 50–1, 64–7, 102–22, 127–8, 133–42, 152–7
 see also Python; SciPy

- C++ array data accesses 19–26
 concepts 3, 8–9, 19–26, 38–44, 127–8, 152–3, 157
 num_sims 103–22, 150–2
 NumStates 188–9
 num_states 188–9
 num_std_dev 188–9
- object_as_array_impl 21–6
 observable 70–9, 95–8, 104–22, 146–57, 177–87
 observable_base 69–79
 observables
 see also common market information; flows
 concepts 3, 69–79, 89–90, 95–8, 100–1, 104–22, 146–57, 177–87
 definition 69–70
 examples 69–70
 LIBOR rate 6, 69, 70–9, 89–91, 104–22, 123–8, 143, 145–52
 reset dates 69–79
 swap rates 74–9, 104–5
 operator 29–31
 optimal exercise rule 131–42
 options
 see also call...; financial instruments; put...
 at-the-money option prices 61
 Black–Scholes option pricing formula 5–9, 28–9, 111–22, 165–8
 early exercise premiums 4, 85–91, 132–3, 219–20
 exercises 4, 85–91, 131–42, 219–20
 in-the-money option prices 150–2
 out-of-the-money option prices 150–2
 pricing approaches 99, 108–9, 165–8
 types 99
 ordinates 31–4, 63–7
 see also curves
 out-of-the-money option prices 150–2
 overloaded functions, C++ 212–14
 overview of the book 3–4
- packages, Python basics 1, 203–5
 partial proxy scheme 143
 Pascal 201
 path-dependency issues, pricing models 99, 123–43, 157
 pathwise derivatives 143
 pay_currency 79–82, 90, 93–8, 120–2, 135–42, 146–57, 178–87
 pay_date 77–9, 79–82, 106–22, 146–57
 pay_df 146–57
 pay_event 93–8, 138–42
 pay_flows 96–8, 120–2
 pay_leg 90–1, 96–8, 120–2
 payoff 85–6, 98, 145–56, 176–87
 payoffs, path-dependency issues 99, 123–43, 157
 pay_or_receive 84–5, 180–7
 pay_rcv 93–8
 pay_receive 84–5, 93–8, 120–2, 148–57
 see also legs
 pay_shift_method 80–2, 178–87
 PDE techniques 143
 period 80–2, 86–91, 96–8, 120–2, 178–87
 phiTTN 104–22
 pickup value regression 132–42, 150–2, 219–20
 pickup_value_regression 133–42
 piecewise constant polynomial fitting
 see also integration
 concepts 49–51, 57–61
 piecewise polynomial integration
 see also integration
 concepts 51–61
 piecewise_cubic_fit 50–1
 piecewise_polynomial_fitting 50–1
 pointers, C++ 212–14
 policies, concepts 121–2
 polynomials 33–4, 37, 49–61
 see also cubic spline...
- PPF
 concepts 2–4, 5–9, 11–26, 165–8
 topology 3, 5–9
 ppf package
 see also C++; individual modules;
 Python
 building 7–9
 concepts 2–4, 5–9, 11–26, 165–8
 data structures 3
 definition 7–8
 ideas 5–9
 installation 7–9
 Microsoft Excel 165–89
 structure 5–9
 testing 6–7, 9, 17–19
 PPF_AddConstant 175–6
 PPF_AddCurve 174–5
 PPF_AddSurface 175–6
 ppf.com 5–6, 165–89
 ppf.core... 5–6, 69–85, 93–8, 124–8, 149–56, 165–8, 176–87
 PPF_CreateLeg 186
 PPF_CreateMarketEnvironment 174–6
 PPF_CreateTrade 187
 ppf_date_time, concepts 5–9, 11–16, 64–7, 70–84, 86–91, 95–8, 110–22, 151–2, 159–63, 168–9
 ppf_date_time.pyd 11, 159–63
 PPF_GenerateAdjuvantTable 183–4
 PPF_GenerateExerciseSchedule 185–6

- PPF_GenerateFixedCoupon
 Observables 182
 PPF_GenerateFlows 184–5
 PPF_GenerateLiborObservables
 182–3
 ppf.market... 63–7, 101–22, 169–76
 ppf.math... 2–3, 6–9, 17–19, 23–6, 28–9,
 31–4, 35–7, 40–1, 44–9, 50–61, 100–22,
 132–42, 170–6, 205
 ppf.math.pyd 23
 ppf.model... 97–8, 100–22, 124–8, 129–42,
 146–57
 ppf.pricer 6, 134–42, 145–52, 176–87,
 188–9
 ppf.test... 6–7, 9, 17–19, 59–61, 64–7,
 109–22, 148–57
 ppf.utility... 6–7, 29–61, 64–7
 pricer 6, 134–42, 145–52, 176–89
 pricer server, COM servers 187–9
 pricer VBA client 188–9
 PricerServer 187–9
 pricer_server 187–9
 pricing models 3–4, 5–9, 28–9, 49–51, 61, 65–7,
 99–122, 123–43, 145–57, 165–89, 217–18,
 219–20
 see also controller; events
 analytic/semi-analytic pricing formulae 99
 approaches 3–4, 99, 108–9, 123–43,
 145–57
 Bermudan swaptions 4, 101–2, 115–16,
 126–8, 132–42, 145–52, 157, 168–89
 Black–Scholes option pricing formula 5–9,
 28–9, 111–22, 165–8
 C++/Python ‘Hybrid Systems’ 4, 159–63
 classes 3
 Hull–White model 3–4, 6, 65–7, 99–122,
 145–57, 168–89, 217–18
 lattice-based pricing framework 3, 6, 49–51,
 67, 99, 101–22, 123–31, 142–3, 149–52,
 168–89
 Microsoft Excel 165–89
 Monte-Carlo simulations 3, 4, 6, 99, 101–2,
 108–22, 123, 128–43, 150–7, 219–20
 numerical techniques 99–122, 123–43,
 145–57, 168–89
 path-dependency issues 99, 123–43, 157
 TARNs 4, 101–2, 145, 152–7
 pricing_date 66–7, 96–8, 125–8
 productivity benefits of Python 1
 proj_basis 71–9, 105–22
 projection periods
 see also proj...
 LIBOR rate 70–9, 105–22
 proj_end_date 71–9, 105–22
 proj_roll 73–9
 proj_start_date 71–9, 105–22
 pseudo random number generation, concepts 3,
 27–8
 public_methods 169–76, 177–87, 188–9
 put options 5–9
 see also options
 pv 77, 106–22, 126–8, 131–42, 148–57
 ‘py’ file suffix 194, 203–5
 PyArray... 20–6
 PyArray_Check 21–6
 PyArray_ContiguousFromObject
 21–6
 PyArray_DOUBLE 22–6
 PyArrayObject 20–6
 Py_DECREF 20–1
 PyErr_SetString 21–6
 Py_Finalize 214–15
 Py_Initialize 162–3, 214–15
 PyObject 20–6, 215
 Python
 see also ActiveState...; Boost...; mathematics;
 NumPy; *ppf* package
 basics 193–205
 batch interpreter mode 193–4
 benefits 1–4
 built-in data types 1, 195–7
 C++/Python ‘Hybrid Systems’ 4, 159–63
 C API routines 19–26, 161–3
 C/C++ interoperability benefits 2, 3–4, 7–9,
 11–26, 157
 class basics 2–3, 201–3
 COM servers 4, 5–6, 98, 165–89
 concepts 1–4, 193–205
 control flow statements 197–200
 dictionaries 119–22, 181, 196–7, 215–16
 dynamic type system 2–3
 encapsulation support 2–3, 58–61, 209
 expressiveness aspects 1
 extensibility aspects 1–4, 7–9, 11–26
 financial engineering 1–4, 11–26
 function basics 2–3, 200–1
 functional programming idioms 1
 GUI toolkits 2
 high-level aspects 1
 indented code 198–200
 inheritance basics 122, 202–3, 209–11
 interactive interpreter mode 193–4
 interoperability aspects 1–4, 7–9, 11–26, 157
 interpreters 2, 24–6, 45–6, 193–4, 208–9
 list basics 3, 196–7, 215
 Microsoft Excel 4, 165–89
 misconceptions 2–3
 module basics 203–5
 overview of the book 3–4
 package basics 1, 203–5
 productivity benefits 1
 simple expressions 194–5

- standard libraries 1
- structure misconceptions 2–3
- tuples 131, 195–7, 200–1, 215
- visualisation software integration 2
- whirlwind tour 193–205
- white space uses 198–200
- Python Distutils package 9
- Python Programming on Win32* (Hammond & Robinson) 165
- Python Scripting for Computational Science* (Langtangen) 2
- python.hpp 214–15
- python -i command 193
- PyUnit testing module, concepts 6–7, 9

- quadratic roots, concepts 3, 46–9
- quadratic.fo 132–42
- quadratic.roots 46–9
- quantitative analysis 1–4, 27–61, 123–43, 165–89

- raise 15–16, 30–1, 35–6, 40–1, 43–5, 50–8, 66–7, 78, 81–3, 85–7, 88, 89, 91, 98, 102, 105–6, 113, 118–19, 131, 133, 134–6, 166–9, 170–6
- random 27–8, 45–6
- random number generation, concepts 3, 27–8, 45–6, 112–22
- random variables, expectation calculations 3, 49–61
- range function, Python basics 197–8
- ratio 48–9
- rcv.flows 89–91, 96–8, 120–2
- redemption.cap 153–6
- redemption.floor 153–6
- reference counts 20–6
- references, C++ 212–14
- reg.clsid 169–76, 177–87, 188–9
- register.com.class 169–76
- register.date... 11–16, 160–3
- register.date.more.cpp 160
- register_numpy.cpp 23
- register_special.functions 18–19
- reg.progid 169–76, 177–89
- regression schemes 4, 132–42, 150–2, 219–20
- regression_model 136–42, 150–2
- regressions 132–42
- regrid 55–61
- regridder 58–61
- regrid.fs 55–7
- regrid.xT 58–61
- regrid.yT 58–61
- relative_date 66–7, 105–22, 125–8, 146–57
- requestor 100–22, 124–8, 129–42, 146–57
- requestor component, pricing models 100–22, 124–8, 129–42
- reset currencies, concepts 70–9, 96–8, 105–22
- reset dates 69–79, 95–8
 - see also observables
- reset.basis 72–9, 89–91, 96–8, 120–2, 178–87
- reset.ccy 69–79, 90
- reset.currency 71–9, 89–90, 96–8, 105–22, 177–87
- reset.date 69–79, 95–8
- reset.duration 73–9, 89–91, 96–8, 120–2
- reset.holiday.centres 72–9
- reset.id 69–79, 93–8, 146–57
- reset.lag 72–9
- reset.period 73–9, 89–91, 96–8, 120–2
- reset.shift.method 72–9, 89–91, 96–8, 120–2, 178–87
- retrieve 66–7, 97–8, 124–8, 169–76, 179–87, 188–9
- retrieve.constant 67, 100–22
- retrieve.curve 66–7, 100–22, 148–57
- retrieve.surface 67, 100–22
- retrieve.symbol... 97–8, 124–8, 131–42, 149–52
- return statements, Python basics 200–5
- risk
 - the Greeks 142–3
 - management systems 4
- Robinson, Andy 165
- rollback 57–61, 108–22, 124–8
- rollback component, pricing models 108–22, 124–8, 177–87
- rollback.max 57–61, 108–22, 124–8
- rollback.tests 109–22
- roll.duration 72–9, 84–5, 89–91, 96–8, 120–2, 177–87
- roll.end 72–9, 81–3, 84–5, 86–91
- roll.period 72–9, 84–5, 89–91, 96–8, 120–2, 177–87
- rolls 14–16, 72–3
- roll.start 77–8, 81–3, 86–91
- root-finding algorithms
 - bisection method 35–6, 37
 - concepts 3, 35–7
 - Newton–Raphson method 36–7
- root.finding 35–7
- roots 46–9, 53–7
- RuntimeError 30–1, 35–6, 40–1, 43–5, 50–8, 66–7, 77, 81–3, 85–7, 88, 89, 90, 98, 102, 104–5, 113, 118–19, 131, 133, 134–6, 147–8, 169, 170–6, 177–8, 202–3
- sausage Monte-Carlo method 143
- Schwartz, E.S. 219

- SciPy 1, 3, 8
 see also NumPy
 scope guard techniques 20
 SDEs 218
 second_axis 64–5
 seed 112–22, 150–2
 self 31–4, 45–6, 51–61, 63–7, 69–79, 93–122,
 124–8, 130–42, 146–57, 178–89
 semi-analytic conditional expectations, concepts
 57–61
 semi_analytic_domain_integrator
 57–61
 server 166–89
 set_event 126–8, 130–42
 set_last_cfs 135–42
 sgn 46–9
 shape 43–6, 50–1, 58–61, 81–3, 103–22,
 133–42
 shared_ptr.hpp 20–4
 shift 14–16, 72–9, 86–8, 111–22
 shift_convention 14–16, 73–9, 80–2,
 83–91, 96–8, 120–2, 151–2
 shift_method 14–16, 73–9, 80–2, 83–91,
 120–2
 short rates 101–2
 sig 42–6, 65
 sign 35–6
 simple expressions, Python basics 194–5
 sin 205
 singular value decomposition of a matrix
 see also linear algebra
 concepts 42–6
 singular_value_decomposition_
 back_substitution 42–6
 solve_tridiagonal_system 2–3, 34,
 39–40
 solve_upper_diagonal_system 17–19,
 40–4, 50–1
 solving linear systems
 see also linear algebra
 concepts 39–40
 solving tridiagonal systems
 see also linear algebra
 concepts 2–3, 34, 39–40
 solving upper diagonal systems
 see also linear algebra
 concepts 17–19, 40–4, 49–51
 sort 48–9
 special functions 17–18, 27–61
 spread 70–9, 154–6
 sqrt 48–9, 52–7, 59–61, 100–22
 square tridiagonal matrices 33–4, 40–1
 standard deviations 44–6, 51–7, 102–22, 133–42,
 188–9
 standard libraries 1
 standard normal cumulative distributions
 see also N
 concepts 3, 27–9, 31, 51–7, 102–22
 start 80–3, 83–91, 96–8, 120–2, 177–87,
 198–200
 start_date 83–4
 start_of_to_year 16
 state 59–61, 102–22, 124–8, 129–42,
 146–57
 state component, pricing models 101–22, 124–8,
 129–42, 145–52
 stddev 53–7, 102–22
 step, Python basics 198–200
 STL functions, C++ 29
 stochastic volatility 113–14
 stop, Python basics 198–200
 str 71–9, 80–3, 86–7, 94, 166–8, 170–6
 string literals, Python basics 194–6
 structure misconceptions, Python 2–3
 sum_array 23–6
 surf 101–22
 surface 64–7, 101–22
 surfaces
 see also environment
 concepts 3, 63, 64–7, 100–22, 170–6
 definition 64
 volatility surfaces 3, 6, 63–7, 100–22
 surface_tests 64–7
 svd 42–4
 see also singular value decomposition of a
 matrix
 swap rates 70–9, 104–5, 115–22
 swap_obs 105–22
 swap_rate 74–9, 116–22
 swaps 4, 70–9, 101–2, 104–5, 115–22, 123–8,
 132–42, 145–52, 157
 swap_tests 149–52
 swaptions 4, 101–2, 115–16, 126–8, 132–42,
 145–52, 157
 symbol_table 97–8, 124–8, 129–42
 symbol_table_listener 125–8, 129–42
 symbol_value_pair 130–42, 155–6
 symbol_value_pairs_to_add 130–42,
 155–6
 sys 27–8

 table 82–4, 169
 tables, adjuvants 82–4, 147–52, 153–6, 177–87
 tag 169–76, 177–89
 target redemption notes (TARNs) 4, 101–2, 145,
 152–7
 concepts 152–7
 definition 152
 pricing models 4, 101–2, 145, 152–7
 target_indicator 153–6
 tarn_coupon_leg_payoff 152–6
 tarn_funding_leg_payoff 154–6

- TARNs *see* target redemption notes
 tarn_tests 155–6
 templates 18–26, 159–63
 tenor_duration 72–9
 tenor_period 72–9
 tenors 67, 84–5, 101–22, 170–6
 term 28–9, 103–22
 term structure of interest rates *see* yield curves
 term volatility, Hull–White model 100–22
 terminal_T 104–22
 term_var 100–22
 term_vol 100–22
 test 6–7, 9, 17–19, 59–61, 64–7, 109–22, 148–57
 test_bond 115–22
 test_bond_option 111–22
 test_bound 30–1
 test_bound_ci 31
 test_constant 111–22
 test_discounted_libor_rollback 109–22
 test_explanatory_variables 117–22
 test_hull_white 67, 109–22
 testing concepts 6–7, 9, 17–19
 test_lattice_pricer 148–57
 test_market 64–7
 test_math 59–61
 test_mean_and_variance 114–22
 test_monte_carlo_pricer 154–6
 test_value 149–52
 theta 48–9, 205
 throw_error_already_set 21–6
 timeline 94–8, 125–8, 129–42
 see also events
 Tk 2
 tline 96–8
 to_ppf_date 168–9, 178–87
 tower law 60
 tower_law_test 60–1
 trace 23–6
 Traceback 195–7, 202–3
 trade 87–91, 94–8, 125–8, 129–42, 150–7, 177–87
 trade representations, concepts 3, 69–91, 93–8
 trade server, COM servers 176–87
 trade utilities, concepts 88–91
 trade VBA client 181
 trade_id 188–9
 trades
 see also exercise...; flows; legs
 concepts 3, 69, 87–91, 93–8, 123–43, 176–87
 definition 69, 88
 TradeServer 176–87, 188–9
 trade_server 176–87, 188–9
 trade_utils 89–91, 129–42, 153–6, 176–87
 transpose 41–4
 tridiagonal systems 2–3, 33–4, 39–40
 try 27–8, 171–6, 177–87
 Trying 6–7
 tuples, Python basics 131, 195–7, 200–1, 215
 TypeError 195–7
 Ubuntu... 8
 underlying 127–8, 130–42
 unicode 172
 unit fo 132–42
 update_indicator 134–42
 update_symbol 97–8, 126–8, 131–42
 upper_bound 29–31
 USD 70–83, 152
 utility 6–7, 29–61, 64–7
 utility functions 17–26, 29–61
 utils 168–76, 187–9
 values 101–22
 vanilla financial instruments, pricing approaches 99, 123–8, 145–57
 var 102–22
 variance 51–61, 102–22
 variates 103–22
 varT 102–22
 Vasicek models 217–18
 see also Hull–White model
 VB... *see* Microsoft...
 vector 41, 44–6, 133–42, 212
 vectorize 133–42
 visualisation software integration, Python benefits 2
 vol 51–7, 59–61, 114–22
 volatility
 Hull–White model 100–22
 piecewise polynomial integration 51–61
 surfaces 3, 6, 63–7, 100–22
 vols 65
 volt 59–61, 59–61
 weekdays 15–16, 159–63
 while statements, Python basics 199–200
 white space, Python basics 198–200
 win32 165–89
 Win32 Python extensions 165–89
 xh 52–7
 xl 52–7
 xprev 53–7
 xs 56–61
 xsT 60–1
 xT 58–61, 108–22
 xtT 58–61, 108–22
 xt 58–61, 108–22

year fractions 14–16, 71–83, 117–22, 126–8,
146–57
year_based_generator 159–63
year_fraction 15–16, 72–83, 146–57
years 90–1, 151–2
yhs 52–7
yield curves, concepts 63–7, 100–22

y1s 52–7
ys 58–61
yt 58–61, 109–22

zero coupon bonds 100–1, 110–22, 146–57,
217–18
zeros 44–6, 53–61, 64–5, 101–22, 133–42,
156–7

Index compiled by Terry Halliday

<http://www.pbookshop.com>