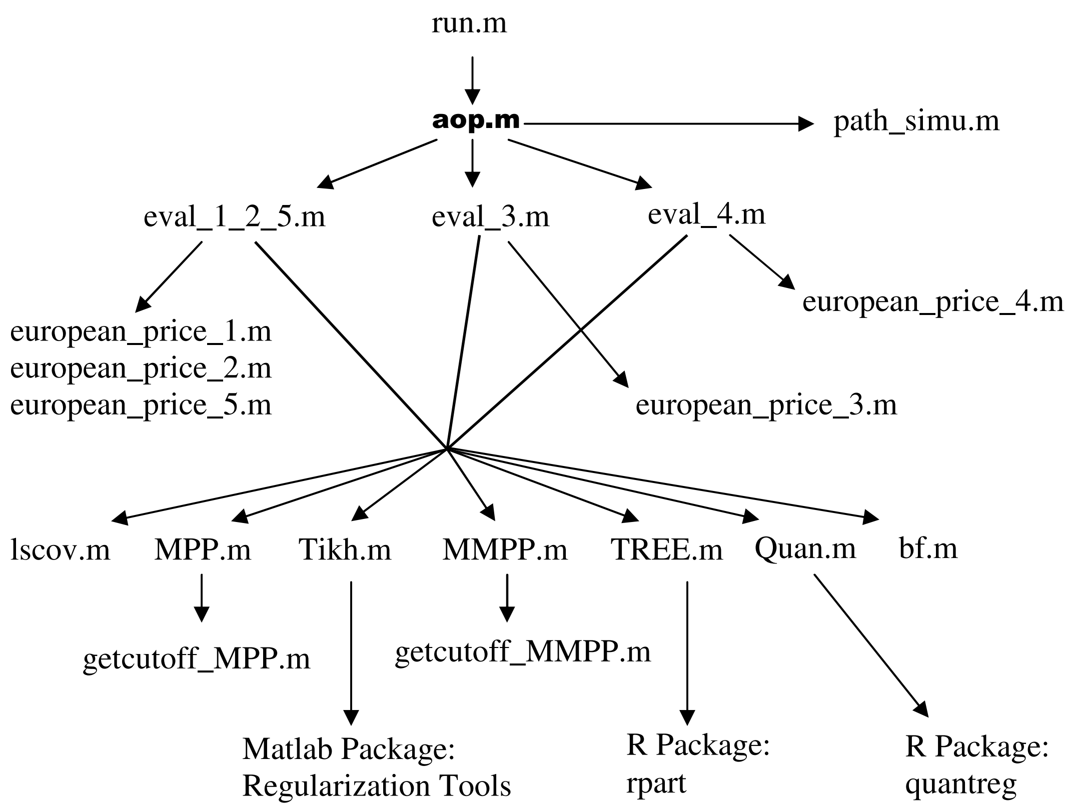
Pricing American Style Options by Monte Carlo Simulation

Chunyu (Ben) Yang, June 2010

Flowchart



• aop.m	the main function, "American Option Pricing"
• run.m	a shell that sets parameters and schedules a batch of jobs
• path_simu.m	Monte Carlo simulation of paths of underlying assets
• eval_1_2_5.m	Pricing case 1, 2, and 5
• eval_3.m	Pricing case 3
• eval_4.m	Pricing case 4
• lscov.m	OLS regression method
• MPP.m	Matching Projection Pursuit method
• Tikh.m	Tikhonov Regularization method
• MMPP.m	Modified Matching Projection Pursuit method
• TREE.m	Regression Tree method
• Quan.m	Quantile regression method
• bf.m	Basis functions generator
• getcutoff_MPP.m	Compute the cutoff value of the MPP method
• getcutoff_MMPP.m	Compute the cutoff value of the MMPP method
• european_price_#.m	Compute the European option price of Case #

aop.m

Description: the main function, "American Option Pricing", which simulates paths and pricing under completely specified parameters.

Syntax: aop(CASE, r, T, M, D, delta, S_0, K, SIGMA, N_r, N, opt_type, n, rho, t_e, t_a, lambda, bf_num, method, bf_type, Euro)

Input:

CASE	case # (=1, 2, 3, 4, 5)
r	risk-free rate
T	terminal time
M	# of exercise points
D	discrete dividend rate
delta	continuous dividend rate
S_0	starting price
K	strike price
SIGMA	volatility
N_r	# of replication
N	# of simulation paths
opt_type	option type (1 call, -1 put)
n	# of underlying assets (required only for case 3)
rho	correlation between assets (required only for case 3)
t_e	available exercise points (required only for case 4)
t_a	# of days average over (required only for case 4)
lambda	jump process parameter (required only for case 5)
bf_num	# of basis functions
method	projection methods method=1 OLS Regression method=2 Quantile Regression method=3 Tikhonov Regularization method=4 MPP method=5 MMPP method=6 TREE
bf_type	type of basis functions bf_type=1 Powers bf_type=2 Laguerre bf_type=3 Legendre bf_type=4 Hermite A bf_type=5 Hermite B bf_type=6 Chebyshev 1st kind A bf_type=7 Chebyshev 1st kind B bf_type=8 Chebyshev 1st kind C bf_type=9 Chebyshev 1st kind A bf_type=10 Chebyshev 1st kind B
Euro	indicator of using European price basis (0: turn off; 1: turn on)

Output: A file named "out.txt" that contains the following information on each row: "CASE; n; K; N; bf_num; method; bf_type; elapse time; price; s.e."

Command window displays the following information:
"elapse time; price; s.e."

Example: aop(1, 0.05, 3, 7, 5, 0, 100, 100, 0.2, 20, 1000, 1, 1, 0, 0, 0, 0, 5, 1, 1, 0) computes the price of a 3-year call option with strike $100 on a stock that has a current price of $100, pays discrete dividends $5 every six months with volatility 20%, risk-free rate 5%. The other parameters are: 20 replications, 1000 paths, 5 basis functions, OLS Regression, power polynomials, and no European price basis.

run.m

Description: A shell that sets default parameters for each test case and schedules a batch of jobs by looping and calling the main function aop(...)

Syntax: run(CASE, M, N, Euro, bf_num_l, bf_num_u, method_l, method_u, n, K)

Input:

CASE	case # (=1, 2, 3, 4, 5)
M	# of exercise points
N	# of simulation paths
Euro	indicator of using European price basis (0: turn off; 1: turn on)
bf_num_l	# of basis functions lower bound
bf_num_u	# of basis functions upper bound
method_l	evaluation methods: lower bound (1 to 6)
method_u	evaluation methods: upper bound (1 to 6)
n	# of underlying assets (required only for case 3)
K	strike price

Output: A file named “out.txt” that contains the following information on each row:
“CASE; n; K; N; bf_num; method; bf_type; elapse time; price; s.e.”

Command window displays the following information:
“elapse time; price; s.e.”

Example: run(3, 13, 1000, 1, 1, 10, 1, 2, 3, 80) computes the price of a max-call option with 3 underlying assets, strike price $80, and 13 exercisable times using 1000 paths, polynomial basis functions of number 1 to 10, European price basis functions, OLS Regression and Quantile Regression. The remaining parameters are set to the default values.

path_simu.m

Description: Simulate N paths for n correlated assets and M exercise times.

Syntax: path = path_simu (r, T, M, D, delta, S_0, SIGMA, N, opt_type, n, rho, lambda)

Input:

r	risk-free rate
T	terminal time
M	# of exercise points
D	discrete dividend rate
delta	continuous dividend rate
S_0	starting price
SIGMA	volatility
N	# of simulation paths
opt_type	option type (1 call, -1 put)
n	# of underlying assets
rho	correlation between assets
lambda	jump process parameter

Output: Matrix "path" with dimension (n, M, N).

Example: path = path_simu(0.05, 1, 13, 0, 0.1, 100, 0.2, 1000, 1, 3, 0.3, 0) simulates 1000 paths at 13 exercise times over a year for 3 assets with correlation 30%. The other parameters are: risk-free rate 5%, continuous dividend rate 10%, volatility 20%, and no jumps.

eval_1_2_5.m

Description: Pricing Case 1 (Call Option with Discrete Dividends), Case 2 (Call Option with Continuous Dividends), and Case 5 (Put Option on a Jump-Diffusion Asset).

Syntax: opt_val = eval_1_2_5(CASE, bf_num, method, bf_type, Euro, r, t_int, opt_type, K, path, SIGMA, delta, D, lambda)

Input:

CASE	case # (=1, 2, 3, 4, 5)	
bf_num	# of basis functions	
method	projection methods	
	method=1	OLS Regression
	method=2	Quantile Regression
	method=3	Tikhonov Regularization
	method=4	MPP
	method=5	MMPP
	method=6	TREE
bf_type	type of basis functions	
	bf_type=1	Powers
	bf_type=2	Laguerre
	bf_type=3	Legendre
	bf_type=4	Hermite A
	bf_type=5	Hermite B
	bf_type=6	Chebyshev 1st kind A
	bf_type=7	Chebyshev 1st kind B
	bf_type=8	Chebyshev 1st kind C
	bf_type=9	Chebyshev 1st kind A
	bf_type=10	Chebyshev 1st kind B
Euro	indicator of using European price basis (0: turn off; 1: turn on)	
r	risk-free rate	
t_int	intermediate time between exercise points	
opt_type	option type (1 call, -1 put)	
K	strike price	
path	matrix of simulation paths generated by function path_simu(…)	
SIGMA	volatility	
delta	continuous dividend rate	
D	discrete dividend rate	
lambda	jump process parameter (required only for case 5)	

Output: The option price "opt_val".

Example: path = path_simu (0.05, 3, 7, 5, 0, 100, 0.2, 1000, 1, 1, 0, 0)
opt_val = eval_1_2_5(1, 5, 1, 1, 0, 0.05, 0.5, 1, 100, path, 0.2, 0.1, 0, 0)
compute the price of a 3-year call option with strike price $100 on a stock that has a current price of $100, pays discrete dividends $5 every six months with volatility 20%, risk-free rate 5%. The other parameters are: 1000 paths, 5 basis functions, OLS Regression, power polynomials, and no European price basis.

eval_3.m

Description: Pricing Case 3 (Max-Call Option on Multiple Underlying Assets)

Syntax: opt_val = eval_3(bf_num, method, bf_type, Euro, r, t_int, opt_type, K, path, SIGMA, delta, rho)

Input:

bf_num	# of basis functions	
method	projection methods	
	method=1	OLS Regression
	method=2	Quantile Regression
	method=3	Tikhonov Regularization
	method=4	MPP
	method=5	MMPP
	method=6	TREE
bf_type	type of basis functions	
	bf_type=1	Powers
	bf_type=2	Laguerre
	bf_type=3	Legendre
	bf_type=4	Hermite A
	bf_type=5	Hermite B
	bf_type=6	Chebyshev 1st kind A
	bf_type=7	Chebyshev 1st kind B
	bf_type=8	Chebyshev 1st kind C
	bf_type=9	Chebyshev 1st kind A
	bf_type=10	Chebyshev 1st kind B
Euro	indicator of using European price basis (0: turn off; 1: turn on)	
r	risk-free rate	
t_int	intermediate time between exercise points	
opt_type	option type (1 call, -1 put)	
K	strike price	
path	matrix of simulation paths generated by function path_simu(…)	
SIGMA	volatility	
delta	continuous dividend rate	
rho	correlation between assets	

Output: The option price “opt_val”.

Example:

path = path_simu (0.05, 1, 13, 0, 0.1, 100, 0.2, 1000, 1, 5, 0.3, 0);
opt_val = eval_3(4, 1, 1, 1, 0.05, 1.0/12, 1, 80, path, 0.2, 0.1, 0.3)
compute the price of a 1-year max-call option with strike price $80 over 5 stocks with current price $100, volatility 20%, risk-free rate 5%, and continuous dividend rate 10%. The correlation between each stock is 30%. The option is monthly exercisable. The other parameters are: 1000 paths, 4 basis functions (see details below), OLS Regression, power polynomial basis functions, and European price basis functions.

Basis functions of case 3 are specially defined as follows:

Suppose there are 5 underlying assets and $X_1<X_2<X_3<X_4<X_5$

bf_num=0: 1 $X_{(5)}$ $X_{(4)}$

bf_num=1: X_1 X_2 X_3 X_4 X_5

bf_num=2: X_1^2 X_2^2 X_3^2 X_4^2 X_5^2

bf_num=3: $X_1 X_2$ $X_1 X_3$ $X_1 X_4$ $X_1 X_5$ $X_2 X_3$ $X_2 X_4$ $X_2 X_5$ $X_3 X_4$ $X_3 X_5$ $X_4 X_5$

bf_num=4: X_1^3 X_2^3 X_3^3 X_4^3 X_5^3

bf_num=5:

$X_1^2 X_2$	$X_1^2 X_3$	$X_1^2 X_4$	$X_1^2 X_5$
$X_2^2 X_1$	$X_2^2 X_3$	$X_2^2 X_4$	$X_2^2 X_5$
$X_3^2 X_1$	$X_3^2 X_2$	$X_3^2 X_4$	$X_3^2 X_5$
$X_4^2 X_1$	$X_4^2 X_2$	$X_4^2 X_3$	$X_4^2 X_5$
$X_5^2 X_1$	$X_5^2 X_2$	$X_5^2 X_3$	$X_5^2 X_4$
$X_1 X_2 X_3$	$X_1 X_2 X_4$	$X_1 X_2 X_5$	
$X_1 X_3 X_4$	$X_1 X_3 X_5$	$X_1 X_4 X_5$	
$X_2 X_3 X_4$	$X_2 X_3 X_5$	$X_2 X_4 X_5$	$X_3 X_4 X_5$

Higher bf_num contains all the basis functions of lower bf_num with some new ones added in.

eval_4.m

Description: Pricing Case 4 (American-Asian Call Option)

Syntax: opt_val = eval_4(bf_num, method, bf_type, Euro, r, t_int, opt_type, K, path, t_e, t_a, SIGMA, delta)

Input:

bf_num	# of basis functions	
method	projection methods	
	method=1	OLS Regression
	method=2	Quantile Regression
	method=3	Tikhonov Regularization
	method=4	MPP
	method=5	MMPP
	method=6	TREE
bf_type	type of basis functions	
	bf_type=1	Powers
	bf_type=2	Laguerre
	bf_type=3	Legendre
	bf_type=4	Hermite A
	bf_type=5	Hermite B
	bf_type=6	Chebyshev 1st kind A
	bf_type=7	Chebyshev 1st kind B
	bf_type=8	Chebyshev 1st kind C
	bf_type=9	Chebyshev 1st kind A
	bf_type=10	Chebyshev 1st kind B
Euro	indicator of using European price basis (0: turn off; 1: turn on)	
r	risk-free rate	
t_int	intermediate time between exercise points	
opt_type	option type (1 call, -1 put)	
K	strike price	
path	matrix of simulation paths generated by function path_simu(…)	
t_e	available exercise points	
t_a	# of days to average over	
SIGMA	volatility	
delta	continuous dividend rate	

Output: The option price “opt_val”.

Example: path = path_simu (0.09, 120/365, 121, 0, 0, 100, 0.2, 1000, 1, 1, 0, 0)
opt_val = eval_4(10, 1, 1, 0, 0.09, 1/365, 1, 100, path, [0, 105, 108, 111, 114, 117, 120], 91, 0.2, 0)
compute the price of a 4-month American-Asian call option with strike price $100 on a stock that has current price $100, volatility 20%, risk-free rate 9%, and no dividend. The option is exercisable at day 0, 105, 108, 111, 114, 117, and 120. The other parameters are: 1000 paths, 10 basis functions, OLS Regression, power polynomial basis functions, and no European price basis functions.

european_price_1.m
european_price_2.m
european_price_3.m
european_price_4.m
european_price_5.m

Description: Compute the European option price of each case.

Syntax:

price = european_price_1(opt_type, S, K, sigma, r, delta, D, T, t_int)
price = european_price_2(opt_type, S, K, sigma, r, delta, T)
price = european_price_3(opt_type,S1,S2,K,T,r,delta1,delta2,sigma1,sigma2,rho)
price = european_price_4(opt_type,S,G,K,n,n1,h,r,delta,sigma)
price = european_price_5(opt_type, S, K, sigma, r, delta, T, lambda)

Input:

opt_type	option type (1 call, -1 put)
S	vector of current asset prices
K	strike price
sigma	volatility
r	risk-free rate
delta	continuous dividend rate
D	discrete dividend
T	time to maturity
t_int	intermediate time between payment of discrete dividends
S1	vector of current prices of Asset 1 (case 3)
S2	vector of current prices of Asset 2 (case 3)
delta1	continuous dividend rate of Asset 1 (case 3)
delta2	continuous dividend rate of Asset 2 (case 3)
sigma1	volatility of Asset 1 (case 3)
sigma2	volatility of Asset 2 (case 3)
rho	correlation between assets (case 3)
G	vector of current average asset prices (case 4)
n	the period of time to average over (case 4)
n1	remaining time to maturity (case 4)
h	time between each averaging point (case 4)
lambda	jump process parameter (case 5)

Output: Vector of European option prices

lscov.m

Description: OLS regression method

Syntax: bet = lscov(X,Y)

Input:

X	regressor matrix
Y	data vector

Output:

bet	vector of coefficients estimates

tikh.m

Description: Tikhonov Regularization method

Syntax: bet = Tikh(X,Y)

Input:

X	regressor matrix
Y	data vector

Output:

bet	vector of coefficients estimates

quan.m

Description: Quantile regression method

Syntax: Y_p = Quan(X,Y)

Input:

X	regressor matrix
Y	data vector

Output:

Y_p	vector of predicted values

MPP.m

Description: Matching Projection Pursuit method

Syntax: bet = MPP(X,Y)

Input:

X	regressor matrix
Y	data vector

Output:

bet	vector of coefficients estimates

MMPP.m

Description: Matching Projection Pursuit method

Syntax: bet = MPP(X,Y,P)

Input:

	X	regressor matrix
	Y	data vector
	P	step-size bound

Output: bet — vector of coefficients estimates

TREE.m

Description: Regression Tree method

Syntax: Y_p = Quan(X,Y)

Input:

	X	regressor matrix
	Y	data vector

Output: Y_p — vector of predicted values

getcutoff_MPP.m

Description: Compute the cutoff value of the MPP method

Syntax: cutoff = getcutoff_MPP(X,av_num)

Input:

	X	regressor matrix
	av_num	number of replications

Output: cutoff — cutoff value of the MPP method

getcutoff_MMPP.m

Description: Compute the cutoff value of the MMPP method

Syntax: cutoff = getcutoff_MMPP(X,P,av_num)

Input:

	X	regressor matrix
	P	step-size bound
	av_num	number of replications

Output: cutoff — cutoff value of the MMPP method