Package: AsianOption (via r-universe)

June 6, 2026

Type Package

Title Asian Option Pricing under Price Impact

Version 0.2.0

Date 2026-03-09

Maintainer Priyanshu Tiwari <tiwari.priyanshu.iitk@gmail.com>

Description Implements the framework of Tiwari and Majumdar (2025) <doi:10.48550/arXiv.2512.07154> for valuing arithmetic and geometric Asian options under transient and permanent market impact. Provides three pricing approaches: Kemna-Vorst frictionless benchmarks, exogenous diffusion pricing (closed-form for geometric, Monte Carlo for arithmetic), and endogenous Hamilton-Jacobi-Bellman valuation via a tree-based Bellman scheme producing indifference bid-ask prices.

License GPL (>= 3)

URL https://github.com/plato-12/AsianOption

BugReports https://github.com/plato-12/AsianOption/issues

Encoding UTF-8

Depends R (>= 4.0.0)

Imports Rcpp (>= 1.0.0)

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), covr

RoxygenNote 7.3.3

Repository https://plato-12.r-universe.dev

Date/Publication 2026-03-08 22:04:04 UTC

RemoteUrl https://github.com/plato-12/asianoption

RemoteRef HEAD

RemoteSha cd02830358310a50118618f3f663737cb10eee67

Contents

`price_arithmetic_asian_diffusion`
Arithmetic Asian Option Price via Euler-Maruyama Monte Carlo (Exogenous Diffusion)

Description

Prices an arithmetic Asian option under exogenous transient price impact using Euler-Maruyama Monte Carlo simulation.

Usage

```
price_arithmetic_asian_diffusion(
  S0,
  K,
  r,
  sigma,
  T,
  lambda_T,
  I0,
  kappa,
  eta,
  rho = 0,
  option_type = "call",
  n_steps = 252,
  n_sims = 1e+05,
  use_control_variate = TRUE,
  seed = 0,
  n_quad = 1000
)
```

Arguments

- `S0` Initial stock price (positive).
- `K` Strike price (positive).
- `r` Risk-free rate (positive).
- `sigma` Volatility parameter (positive).
- `T` Time to maturity (positive).
- `lambda_T` Transient impact coefficient (non-negative).
- `I0` Initial transient impact state (real number).
- `kappa` Mean reversion rate for transient impact (positive).
- `eta` Noise amplitude for transient impact process. Can be: - A single non-negative number (constant eta) - A function of time t in [0,T] returning a non-negative value
- `rho` Correlation between stock and impact Brownian motions (in [-1,1]). Default is 0.
- `option_type` Character string: "call" (default) or "put".
- `n_steps` Number of time steps in the Euler-Maruyama discretisation (default: 252).
- `n_sims` Number of Monte Carlo simulation paths (default: 100000).
- `use_control_variate` Logical. If TRUE (default), uses the geometric Asian diffusion closed-form price as a control variate to reduce variance.
- `seed` Integer seed for reproducibility. 0 means no seed (default: 0).
- `n_quad` Number of quadrature points for the geometric closed-form computation when using control variates (default: 1000).

Details

In the exogenous regime with no active trading ($\nu \equiv 0$), the stock price dynamics are:

$$dS_t = S_t(r + \bar{\lambda}_T I_t)\, dt + \sigma S_t\, dW_t$$

$$dI_t = -\kappa I_t\, dt + \eta(t)\, dW_t^I$$

where W and W^I are Brownian motions with instantaneous correlation ρ.

The arithmetic Asian payoff is $\Phi_A(Y_T) = (Y_T/T - K)^+$ where $Y_t = \int_0^t S_u\, du$.

The Euler-Maruyama scheme discretises the SDEs on a uniform grid with step $\Delta t = T/N$. The log-Euler method is used for S to ensure positivity.

When `use_control_variate = TRUE`, the geometric Asian diffusion closed-form from `price_geometric_asian_diffusio` is used as a control variate, which typically reduces the standard error substantially.

Value

An object of class "arithmetic_asian_diffusion" (a list) with:

price Estimated option price.

std_error Standard error of the estimate.

lower_ci Lower bound of 95% confidence interval.

upper_ci Upper bound of 95% confidence interval.

geometric_price Closed-form geometric Asian price (benchmark).

correlation Correlation between arithmetic and geometric MC payoffs.

use_control_variate Whether control variate was used.

n_sims Number of simulations.

n_steps Number of time steps.

References

Tiwari, P., & Majumdar, S. (2025). Asian option valuation under price impact. *arXiv preprint*. doi:10.48550/arXiv.2512.07154

See Also

price_geometric_asian_diffusion for the geometric Asian closed-form in the same diffusion limit.

Examples

```
# Basic call pricing
price_arithmetic_asian_diffusion(
  S0 = 100, K = 100, r = 0.05, sigma = 0.2, T = 1,
  lambda_T = 0.01, I0 = 0, kappa = 1, eta = 0.1, rho = 0,
  n_steps = 100, n_sims = 10000, seed = 42
)

# With time-dependent eta
eta_func <- function(t) 0.1 * (1 + 0.5 * t)
price_arithmetic_asian_diffusion(
  S0 = 100, K = 100, r = 0.05, sigma = 0.2, T = 1,
  lambda_T = 0.01, I0 = 0.5, kappa = 2, eta = eta_func, rho = 0.3,
  n_steps = 100, n_sims = 10000, seed = 42
)
```

price_arithmetic_asian_hjb
Price Arithmetic Asian Option via HJB Bellman Scheme (Endogenous Impact)

Description

Computes bid and ask prices for an arithmetic Asian option under transient price impact using a Bellman (HJB) scheme.

Usage

```
price_arithmetic_asian_hjb(
  S0,
  K,
  T,
  N,
  sigma,
  r_cont,
  kappa,
  lambda_bar_T,
  lambda_bar_P,
  k_A,
  k_B,
  psi_cost,
  eta = 1,
  p = 0.5,
  I0 = 0,
  control_set = NULL,
  nu_min = -5,
  nu_max = 5,
  n_controls = 31,
  n_logS = NULL,
  n_I = 51,
  n_Y = 51,
  option_type = "call",
  validate = TRUE
)
```

Arguments

S0	Initial stock price (positive).
K	Strike price (positive).
T	Time to maturity (positive).
N	Number of time steps (positive integer).
sigma	Volatility (positive).

r_cont	Continuous risk-free rate.
kappa	Mean reversion rate for impact (non-negative).
lambda_bar_T	Transient impact coefficient (non-negative).
lambda_bar_P	Permanent impact coefficient (non-negative).
k_A, k_B	Cost coefficients (non-negative).
psi_cost	Cost exponent (in (0, 2]).
eta	Noise trader intensity (scalar or length-N vector).
p	Probability of up move (in (0, 1)).
I0	Initial impact state.
control_set	Optional numeric vector of controls; otherwise built from nu_min, nu_max, n_controls.
nu_min, nu_max, n_controls	Control grid (used if control_set is NULL).
n_logS, n_I	Grid sizes for log-price and impact state.
n_Y	Grid size for running state (integral of log S). Ignored if n_Z is provided.
option_type	"call" or "put".
validate	Whether to validate inputs.

Details

Bid and ask are defined via value-function differences: a baseline problem (no option), a long-option problem, and a short-option problem are solved internally in C++.

Value

A list with S3 class "hjb_asian" containing:

ask_price Ask (seller's indifference) price at t=0

bid_price Bid (buyer's indifference) price at t=0

mid_price Mid price

spread Ask minus bid

optimal_nu Optimal trading rate (seller/short) per period, length N

optimal_volumes Seller volumes per period (optimal_nu * dt)

optimal_nu_buyer Optimal trading rate (buyer/long) per period

optimal_volumes_buyer Buyer volumes per period

asian_type Type of Asian option ("arithmetic")

option_type Option type

params List of input parameters

grid_sizes Grid sizes used in the computation `price_geometric_asian_diffusion` *Geometric Asian Option Price in Exogenous Diffusion Limit*

Description

Computes the closed-form price for a geometric Asian call option in the exogenous diffusion limit with transient price impact.

Usage

```
price_geometric_asian_diffusion(
  S0,
  K,
  r,
  sigma,
  T,
  lambda_T,
  I0,
  kappa,
  eta,
  rho = 0,
  option_type = "call",
  n_quad = 1000
)
```

Arguments

S0	Initial stock price (positive).
K	Strike price (positive).
r	Risk-free rate (positive, typically close to 1 in discrete models).
sigma	Volatility parameter (positive).
T	Time to maturity (positive).
lambda_T	Transient impact coefficient (non-negative).
I0	Initial transient impact state (can be any real number).
kappa	Mean reversion rate for transient impact (positive).
eta	Noise amplitude for transient impact process. Can be: - A single positive number (constant eta) - A function of time t in [0,T] returning a non-negative value
rho	Correlation between stock and impact Brownian motions (in [-1,1]).
option_type	Character string: "call" (default) or "put".
n_quad	Number of quadrature points for numerical integration (default: 1000).

Details

In the exogenous regime with no trading control (nu = 0), the stock price dynamics are:

dS_t = S_t * (r + lambda_T * I_t) dt + sigma * S_t * dW_t dI_t = -kappa * I_t dt + eta(t) * dW^I_t

where W and W^I are Brownian motions with correlation rho.

The running log-integral Z_T = integral_0^T log(S_u) du is Gaussian, so G_T = exp(Z_T/T) is lognormal. This yields a Black-Scholes type closed form:

U(0) = exp(-r*T) * [exp(mu_G + sigma_G^2/2) * Phi(d1) - K * Phi(d2)]

where: - mu_G = m_Z / T - sigma_G^2 = v_Z / T^2 - m_Z and v_Z are the mean and variance of Z_T - Phi is the standard normal CDF - d1 = (mu_G - log(K) + sigma_G^2) / sigma_G - d2 = d1 - sigma_G

Value

The price of the geometric Asian option in the exogenous diffusion limit.

References

Tiwari, P., & Majumdar, S. (2025). Asian option valuation under price impact. *arXiv preprint*. doi:10.48550/arXiv.2512.07154

Examples

```
# Example 1: Constant eta, no correlation
price_geometric_asian_diffusion(
  S0 = 100, K = 100, r = 0.05, sigma = 0.2, T = 1,
  lambda_T = 0.01, I0 = 0, kappa = 1, eta = 0.1, rho = 0
)

# Example 2: Time-dependent eta
eta_func <- function(t) 0.1 * (1 + 0.5 * t)
price_geometric_asian_diffusion(
  S0 = 100, K = 100, r = 0.05, sigma = 0.2, T = 1,
  lambda_T = 0.01, I0 = 0.5, kappa = 2, eta = eta_func, rho = 0.3
)
```

`price_geometric_asian_hjb` *Price Geometric Asian Option via HJB Bellman Scheme (Endogenous Impact)*

Description

Computes bid and ask prices for a geometric Asian option under transient price impact using a Bellman (HJB) scheme. Same interface as `price_arithmetic_asian_hjb`; the running average is geometric (average of log-price, then exp).

Usage

```
price_geometric_asian_hjb(
  S0,
  K,
  T,
  N,
  sigma,
  r_cont,
  kappa,
  lambda_bar_T,
  lambda_bar_P,
  k_A,
  k_B,
  psi_cost,
  eta = 1,
  p = 0.5,
  I0 = 0,
  control_set = NULL,
  nu_min = -5,
  nu_max = 5,
  n_controls = 31,
  n_logS = NULL,
  n_I = 51,
  n_Y = 51,
  n_Z = NULL,
  option_type = "call",
  validate = TRUE
)
```

Arguments

S0	Initial stock price (positive).
K	Strike price (positive).
T	Time to maturity (positive).
N	Number of time steps (positive integer).
sigma	Volatility (positive).
r_cont	Continuous risk-free rate.
kappa	Mean reversion rate for impact (non-negative).
lambda_bar_T	Transient impact coefficient (non-negative).
lambda_bar_P	Permanent impact coefficient (non-negative).
k_A, k_B	Cost coefficients (non-negative).
psi_cost	Cost exponent (in (0, 2]).
eta	Noise trader intensity (scalar or length-N vector).
p	Probability of up move (in (0, 1)).

I0	Initial impact state.
control_set	Optional numeric vector of controls; otherwise built from nu_min, nu_max, n_controls.
nu_min, nu_max, n_controls	Control grid (used if control_set is NULL).
n_logS, n_I	Grid sizes for log-price and impact state.
n_Y	Grid size for running state (integral of log S). Ignored if n_Z is provided.
n_Z	Grid size for running state (alias for geometric case). If provided, used instead of n_Y.
option_type	"call" or "put".
validate	Whether to validate inputs.

Value

List with S3 class "hjb_asian" (same structure as arithmetic), with asian_type = "geometric".

---

price_kemna_vorst_arithmetic

Kemna-Vorst Arithmetic Average Asian Option

Description

Calculates the price of an arithmetic average Asian option using Monte Carlo simulation with variance reduction via the geometric average control variate. This implements the Kemna & Vorst (1990) method WITHOUT price impact.

Usage

```
price_kemna_vorst_arithmetic(
  S0,
  K,
  r,
  sigma,
  T0,
  T_mat,
  n,
  M = 10000,
  option_type = "call",
  use_control_variate = TRUE,
  seed = NULL,
  return_diagnostics = FALSE
)
```

Arguments

S0	Numeric. Initial stock price at time T0 (start of averaging period). Must be positive.
K	Numeric. Strike price. Must be positive.
r	Numeric. Continuously compounded risk-free rate (e.g., 0.05 for 5%). Use `log(r_gross)` to convert from gross rate.
sigma	Numeric. Volatility (annualized standard deviation). Must be non-negative.
T0	Numeric. Start time of averaging period. Must be non-negative.
T_mat	Numeric. Maturity time. Must be greater than T0.
n	Integer. Number of averaging points (observations). Must be positive.
M	Integer. Number of Monte Carlo simulations. Default is 10000. Larger values give more accurate results but take longer.
option_type	Character. Type of option: "call" (default) or "put".
use_control_variate	Logical. If TRUE (default), uses the geometric average as a control variate for variance reduction. This dramatically improves accuracy.
seed	Integer. Random seed for reproducibility. Default is NULL (no seed).
return_diagnostics	Logical. If TRUE, returns additional diagnostic information including confidence intervals, correlation, and variance reduction factor. Default is FALSE.

Value

If `return_diagnostics = FALSE`, returns a numeric value (the estimated option price). If `return_diagnostics = TRUE`, returns a list with components:

price Estimated option price

std_error Standard error of the estimate

lower_ci Lower 95% confidence interval

upper_ci Upper 95% confidence interval

geometric_price Analytical geometric average price (control variate)

correlation Correlation between arithmetic and geometric payoffs

variance_reduction_factor Ratio of variances (with/without control)

n_simulations Number of Monte Carlo simulations used

n_steps Number of time steps in each simulation

References

Kemna, A.G.Z. and Vorst, A.C.F. (1990). "A Pricing Method for Options Based on Average Asset Values." *Journal of Banking and Finance*, 14, 113-129.

Examples

```
price_kemna_vorst_arithmetic(
  S0 = 100, K = 100, r = 0.05, sigma = 0.2,
  T0 = 0, T_mat = 1, n = 50, M = 10000
)
```

---

price_kemna_vorst_geometric

Kemna-Vorst Geometric Average Asian Option

Description

Calculates the price of a geometric average Asian call option using the closed-form analytical solution from Kemna & Vorst (1990). This is the standard benchmark implementation WITHOUT price impact.

Usage

```
price_kemna_vorst_geometric(S0, K, r, sigma, T0, T_mat, option_type = "call")
```

Arguments

`S0`	Numeric. Initial stock price at time T0 (start of averaging period). Must be positive.
`K`	Numeric. Strike price. Must be positive.
`r`	Numeric. Gross risk-free interest rate per period (e.g., 1.05 for 5 Must be positive.
`sigma`	Numeric. Volatility (annualized standard deviation). Must be non-negative.
`T0`	Numeric. Start time of averaging period. Must be non-negative.
`T_mat`	Numeric. Maturity time. Must be greater than T0.
`option_type`	Character. Type of option: "call" (default) or "put".

Details

The geometric average at maturity is defined as:

$$G_T = \exp\left(\frac{1}{T - T_0}\int_{T_0}^{T} \log(S(\tau))d\tau\right)$$

For the discrete case with n+1 observations:

$$G_T = \left(\prod_{i=0}^{n} S(T_i)\right)^{1/(n+1)}$$

The closed-form solution for a call option is:

$$C = S_0 e^{d^*} N(d) - KN(d - \sigma_G \sqrt{T - T_0})$$

where:

$$d^* = \frac{1}{2}(r - \frac{\sigma^2}{6})(T - T_0)$$

$$d = \frac{\log(S_0/K) + \frac{1}{2}(r + \frac{\sigma^2}{6})(T - T_0)}{\sigma\sqrt{(T - T_0)/3}}$$

and $N(\cdot)$ is the cumulative standard normal distribution function.

Value

Numeric. The analytical price of the geometric average Asian option.

References

Kemna, A.G.Z. and Vorst, A.C.F. (1990). "A Pricing Method for Options Based on Average Asset Values." *Journal of Banking and Finance*, 14, 113-129.

Examples

```
price_kemna_vorst_geometric(
  S0 = 100, K = 100, r = 0.05, sigma = 0.2,
  T0 = 0, T_mat = 1, option_type = "call"
)
```

`print.hjb_asian` *Print method for HJB Asian option results*

Description

Print method for HJB Asian option results

Usage

```
## S3 method for class 'hjb_asian'
print(x, ...)
```

Arguments

`x`	Object of class `hjb_asian` from `price_arithmetic_asian_hjb`.
`...`	Additional arguments (unused).

Value

Invisible `x`.

print.kemna_vorst_arithmetic
Print Method for Kemna-Vorst Arithmetic Results

Description

Print Method for Kemna-Vorst Arithmetic Results

Usage

```
## S3 method for class 'kemna_vorst_arithmetic'
print(x, ...)
```

Arguments

`x`	Object of class "kemna_vorst_arithmetic"
`...`	Additional arguments (ignored)

Value

Invisibly returns the input object `x`. Called for side effects (printing).

summary.kemna_vorst_arithmetic
Summary Method for Kemna-Vorst Arithmetic Results

Description

Summary Method for Kemna-Vorst Arithmetic Results

Usage

```
## S3 method for class 'kemna_vorst_arithmetic'
summary(object, ...)
```

Arguments

`object`	Object of class "kemna_vorst_arithmetic"
`...`	Additional arguments (ignored)

Value

Invisibly returns the input object `object`. Called for side effects (printing).

Index