

Package: amsSim (via r-universe)

June 1, 2026

Type Package

Title Adaptive Multilevel Splitting for Option Simulation and Pricing

Version 0.1.0

Description Simulation and pricing routines for rare-event options using Adaptive Multilevel Splitting and standard Monte Carlo under Black-Scholes and Heston models. Core routines are implemented in C++ via Rcpp and RcppArmadillo with lightweight R wrappers.

License MIT + file LICENSE

URL <https://github.com/RiccardoGozzo/amsSim>,
<https://riccardogozzo.github.io/amsSim/>

BugReports <https://github.com/RiccardoGozzo/amsSim/issues>

Encoding UTF-8

Language en-US

Depends R (>= 4.1)

Imports Rcpp (>= 1.0.0)

LinkingTo Rcpp, RcppArmadillo

SystemRequirements C++17

ByteCompile true

NeedsCompilation yes

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository <https://riccardogozzo.r-universe.dev>

Date/Publication 2025-10-28 16:20:35 UTC

RemoteUrl <https://github.com/riccardogozzo/amssim>

RemoteRef HEAD

RemoteSha 11950c2a90d57c7b1e2be33948ce7104c77653ba

Contents

AMS	2
simulate_AMS	3
Index	5

AMS	<i>AMS Adaptive Multilevel Splitting estimator for rare-event option payoffs.</i>
-----	---

Description

Pipeline per iteration:

- Simulate n paths under the chosen model (BS/Heston-family).
- Compute continuation scores $a_{i,j}$ via `function_AMS_Cpp`.
- Set level $L = K$ -th order statistic of $\max_j a_{i,j}$.
- Identify survivors (top $n - K$) and parents (K indices that cleared the level).
- For each parent, cut at first index that exceeds L and resimulate the suffix.
- Repeat until $L \geq L_{\max}$. Then compute discounted payoff on the final population.

Usage

```
AMS(
  model,
  type,
  funz,
  n,
  t,
  p,
  r,
  sigma,
  S0,
  rho = NULL,
  rim = 0L,
  v0 = 0.04,
  Lmax = 0,
  strike = 1,
  K = 1L
)
```

Arguments

<code>model</code>	1 = Black–Scholes; 2,3,4 = Heston variants (as in <code>simulate_AMS</code>).
<code>type</code>	Payoff type passed to <code>payoff()</code> and <code>function_AMS_Cpp</code> (1..6).

funz	1 = BS digital proxy in continuation; 2 = raw feature (signed).
n	Population size ($> K$).
t	Maturity in years (>0).
p	Total time steps (>0).
r	Risk-free rate.
sigma	BS volatility (used by continuation; >0 if <code>funz == 1</code>).
S0	Initial spot.
rho	Correlation for Heston models (required for <code>model >= 2</code> , in $[-1, 1]$).
rim	Left-trim for simulation (keep last <code>p - rim</code> steps; $0 \leq \text{rim} < p$).
v0	Initial variance for Heston models (≥ 0).
Lmax	Stopping level: iterate while $L < L_{\max}$.
strike	Strike K used by continuation and final payoff.
K	Number of resampled offspring per iteration ($1..n-1$).

Value

List with price and std.

Examples

```
out <- AMS(model = 2, type = 3, funz = 1, n = 500, t = 1, p = 252, r = 0.03,
           sigma = 0.2, rho = -0.5, S0 = 1, rim = 0, Lmax = 0.5, strike = 1.3, K = 200)
str(out)
```

simulate_AMS	<i>simulate_AMS Monte Carlo simulation of price paths under: 1 = Black-Scholes (exact solution) 2 = Heston (Euler discretisation) 3 = Heston (Milstein discretisation) 4 = Heston (Quadratic-Exponential scheme, Andersen 2008)</i>
--------------	---

Description

simulate_AMS Monte Carlo simulation of price paths under: 1 = Black-Scholes (exact solution) 2 = Heston (Euler discretisation) 3 = Heston (Milstein discretisation) 4 = Heston (Quadratic-Exponential scheme, Andersen 2008)

Usage

```
simulate_AMS(model, n, t, p, r, sigma, S0, rho = NULL, rim = 0L, v0 = 0.04)
```

Arguments

model	Integer in {1, 2, 3, 4} selecting the model.
n	Number of simulated paths (>0).
t	Maturity in years (>0).
p	Total time steps (>0).
r	Risk-free rate.
sigma	Black-Scholes volatility (≥ 0 , used only when model == 1).
S0	Initial spot price (>0).
rho	Correlation between asset and variance Brownian motions (required for Heston models, finite in $[-1, 1]$).
rim	Left-trim: discard the first rim time steps ($0 \leq \text{rim} < p$). Returned matrices keep $p - \text{rim} + 1$ columns including the initial time.
v0	Initial variance for Heston models (≥ 0).

Value

List: for model 1 returns S ($n \times (p - \text{rim} + 1)$); for Heston models returns S and V .

Examples

```
b <- simulate_AMS(1, n = 50, t = 1, p = 10, r = 0.01, sigma = 0.2, S0 = 100, rho = NULL)
str(b)
```

Index

AMS, [2](#)

simulate_AMS, [3](#)