Package ‘MG1StationaryProbability’

June 13, 2023

Type Package

Title Computes Stationary Distribution for M/G/1 Queuing System

URL https://github.com/MashroomMole/MG1StationaryProbability

Version 0.1.2

Description The idea of a computational algorithm described in the article by Andronov M. et al. (2022) <https://link.springer.com/chapter/10.1007/978-3-030-92507-9_13>.
The purpose of this package is to automate computations for a Markov-Modulated M/G/1 queuing system with alternating Poisson flow of arrivals. It offers a set of functions to calculate various mean indices of the system, including mean flow intensity, mean service busy and idle times, and the system's stationary probability.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

BugReports https://github.com/MashroomMole/MG1StationaryProbability/issues

Imports parallel, stats, doParallel(>= 1.0.17), foreach(>= 1.5.2), memoise(>= 2.0.1),

Depends R (>= 4.0.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Olga Zoldaka [aut, cre]

Maintainer Olga Zoldaka <zoldaka@gmail.com>

Repository CRAN

Date/Publication 2023-06-13 07:00:12 UTC

R topics documented:

---

b *Service continuous density distribution*

Description

Service continuous density distribution

Usage

```
b(t)
```

Arguments

`t`	time value

Value

density function value (double) given `t`

`densityOfSojournTimeAtState_i` *The density of the sojourn time in state i with probability that*

Description

The density of the sojourn time in state i with probability that

Usage

```
densityOfSojournTimeAtState_i(i, j, t, dt, m = c(0.2, 0.3), mMax = 14)
```

Arguments

`i`	MC i-th state
`j`	MC j-th state
`t`	time value
`dt`	time increment
`m`	distribution parameters vector of sojourn times in alternating environment states
`mMax`	max number of addends in sums

Value

double

Examples

```
densityOfSojournTimeAtState_i(1, 0, 10, 1, m=c(1, 2), mMax=5)
```

`EN` *Expectation of number of arriving claims depending on i and j*

Description

Expectation of number of arriving claims depending on i and j

Usage

```
EN(i, j, t, m = c(0.2, 0.3), lambda = c(1, 2))
```

Arguments

`i`	MC i-th state
`j`	MC j-th state
`t`	time value
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector

Value

double

Examples

```
EN(1, 1, 2)
```

`ENU` *Expectation of number of arriving claims*

Description

Expectation of number of arriving claims

Usage

```
ENU(i, t)
```

Arguments

`i`	MC i-th state
`t`	time value

Value

double

Examples

```
ENU(1, 3)
```

`finalStateProbability` *Probability of the final state*

Description

Probability of the final state

Usage

```
finalStateProbability(i, j, t, m = c(0.2, 0.3))
```

Arguments

`i`	MC i-th state
`j`	MC j-th state
`t`	time value
`m`	distribution parameters vector of sojourn times in alternating environment states

Value

double

Examples

```
finalStateProbability(0, 1, 10)
```

`flowIntensityMean` *The mean intensity of the arrived flow*

Description

The mean intensity of the arrived flow

Usage

```
flowIntensityMean(lambda = c(1, 2))
```

Arguments

`lambda`	Poisson flow intensity vector

Value

mean intensity value (double) of the arrived flow h *Density of empty time for initial state i jointly with probability of final state j*

Description

Density of empty time for initial state i jointly with probability of final state j

Usage

```
h(i, j, t, m = c(0.2, 0.3), lambda = c(1, 2))
```

Arguments

`i`	MC i-th state
`j`	MC j-th state
`t`	time value
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector

Value

double

Examples

```
h(1, 1, 2, m = c(2.5, 0.2))
```

loadCoefficient *Load coefficient*

Description

Load coefficient

Usage

```
loadCoefficient(m, lambda)
```

Arguments

`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector

Value

load coefficient value (double) of the arriving flow

Examples

```
loadCoefficient(m = c(0.2, 0.3), lambda = c(1,2))
```

---

meanSojournTimeWithFSP   *Mean sojourn time in the initial state i jointly with final probability of state j*

Description

Mean sojourn time in the initial state i jointly with final probability of state j

Usage

```
meanSojournTimeWithFSP(i, j, t, m = c(0.2, 0.3))
```

Arguments

`i`	MC i-th state
`j`	MC j-th state
`t`	time value
`m`	distribution parameters vector of sojourn times in alternating environment states

Value

double

Examples

```
meanSojournTimeWithFSP(1, 0, 3)
```

`meanSoujournTime` *Mean sojourn time in the initial state i (without final probability of state j)*

Description

Mean sojourn time in the initial state i (without final probability of state j)

Usage

```
meanSoujournTime(i, t)
```

Arguments

`i`	MC i-th state
`t`	time value

Value

double

Examples

```
meanSoujournTime(0, 10)
```

`meanTimeEmptyFixed` *Mean time of empty period in fixed state i*

Description

Mean time of empty period in fixed state i

Usage

```
meanTimeEmptyFixed(i)
```

Arguments

`i`	MC i-th state

Value

complex

`meanTimeOfBusyPeriodETW` *Mean time of busy period*

Description

Mean time of busy period

Usage

```
meanTimeOfBusyPeriodETW(m = c(0.2, 0.3), lambda = c(1, 2))
```

Arguments

`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector description

Value

complex

`meanTimeOfBusyPeriodEW` *Mean time of busy period multiplied by load coefficient*

Description

Mean time of busy period multiplied by load coefficient

Usage

```
meanTimeOfBusyPeriodEW(m = c(0.2, 0.3), lambda = c(1, 2))
```

Arguments

`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector

Value

complex

`meanTimeOfEmptyPeriod` *Mean time of empty period given the stationary probability*

Description

Mean time of empty period given the stationary probability

Usage

```
meanTimeOfEmptyPeriod()
```

Value

complex

`MET` *Mean idle time if initial state i*

Description

Mean idle time if initial state i

Usage

```
MET(i, m = c(0.2, 0.3), lambda = c(1, 2), tmax = 12)
```

Arguments

`i`	MC i-th state
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector
`tmax`	upper integration limit

Value

double

Examples

```
MET(1)
```

`MST` *Mean empty time sojourn time in the initial state i during the empty period*

Description

Mean empty time sojourn time in the initial state i during the empty period

Usage

```
MST(i, m = c(0.2, 0.3), lambda = c(1, 2), tmax = 12)
```

Arguments

`i`	MC i-th state
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector
`tmax`	upper integration limit

Value

double

Examples

```
MST(1)
```

`not_i` *Helper "not i" function*

Description

Helper "not i" function

Usage

```
not_i(i = 0)
```

Arguments

`i`	MC i-th state

Value

2 if i = 0 and 1 if i = 1

`p0` *The stationary probabilities of the environment state 0*

Description

The stationary probabilities of the environment state 0

Usage

```
p0(m = c(0.2, 0.3))
```

Arguments

`m`	distribution parameters vector of sojourn times in alternating environment states

Value

stationary probability of the environment state 0 (double)

Examples

```
p0()
```

`p1` *The stationary probabilities of the environment state 1*

Description

The stationary probabilities of the environment state 1

Usage

```
p1(m = c(0.2, 0.3))
```

Arguments

`m`	distribution parameters vector of sojourn times in alternating environment states

Value

stationary probability of the environment state 1 (double)

pi — *Stationary probabilities for continuous time environment's state*

Description

Stationary probabilities for continuous time environment's state

Usage

```
pi(m = c(0.2, 0.3))
```

Arguments

m	distribution parameters vector of sojourn times in alternating environment states

Value

double

Examples

```
pi()
```

probabilitiesMatrix — *Probability matrix calculation. Rows represent arriving probabilities at state i and columns represent the same for state j*

Description

Probability matrix calculation. Rows represent arriving probabilities at state i and columns represent the same for state j

Usage

```
probabilitiesMatrix(
  i,
  j,
  m = c(0.2, 0.3),
  lambda = c(1, 2),
  tmax = 12,
  nmax = 5
)
```

Arguments

`i`	MC i-th state
`j`	MC j-th state
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector
`tmax`	upper integration limit
`nmax`	limitation for number of arriving claims

Value

matrix with `nmax` rows and columns

`probabilityOfNArrival` *Probability of n arrival during time t jointly with final state j if initial state is i*

Description

Probability of n arrival during time t jointly with final state j if initial state is i

Usage

```
probabilityOfNArrival(i, j, n, t, m = c(0.2, 0.3), lambda = c(1, 2))
```

Arguments

`i`	MC i-th state
`j`	MC j-th state
`n`	number of arrivals
`t`	upper integration limit
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector

Value

double

Examples

```
probabilityOfNArrival(1, 0, 10, 3, m=c(0.5, 0.3), lambda=c(2, 1))
```

`probabilityOfNArrivalW` *Probability of n arrival during time t (without joint probability of j)*

Description

Probability of n arrival during time t (without joint probability of j)

Usage

```
probabilityOfNArrivalW(i, n, t, m = c(0.2, 0.3), lambda = c(1, 2))
```

Arguments

`i`	MC i-th state
`n`	number of arrivals
`t`	time value
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector

Value

double

Examples

```
probabilityOfNArrivalW(1, 2, 3)
```

`PrTr` *Probability to have state j in the ending of the idle period, if initially we have state i*

Description

Probability to have state j in the ending of the idle period, if initially we have state i

Usage

```
PrTr(i, j, m = c(0.2, 0.3), lambda = c(1, 2), tmax = 12)
```

Arguments

`i`	MC i-th state
`j`	MC j-th state
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector
`tmax`	upper integration limit

Value

double

Examples

```
PrTr(1, 0)
```

`resultingMatrix` *Resulting probabilities matrix calculation*

Description

Resulting probabilities matrix calculation

Usage

```
resultingMatrix(m = c(0.2, 0.3), lambda = c(1, 2), tmax = 12, nmax = 5)
```

Arguments

`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector
`tmax`	upper integration limit
`nmax`	limitation for number of arriving claims

Value

matrix with `2*nmax` rows and columns

`serviceDistribution` *Service distribution function*

Description

Service distribution function

Usage

```
serviceDistribution(t)
```

Arguments

`t`	time value

Value

service function value (double) given `t`

`stationaryProbabilities` *Stationary probability function*

Description

Stationary probability function

Usage

```
stationaryProbabilities(m = c(0.2, 0.3), lambda = c(1, 2), tmax = 12, nmax = 5)
```

Arguments

`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector
`tmax`	upper integration limit
`nmax`	limitation for number of arriving claims

Value

MC stationary probability vector

`stationaryProbabilitiesOfEmptyStates`

Stationary probabilities of the empty states in continuous time model

Description

Stationary probabilities of the empty states in continuous time model

Usage

```
stationaryProbabilitiesOfEmptyStates(i, m = c(0.2, 0.3), lambda = c(1, 2))
```

Arguments

`i`	MC i-th state
`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector

Value

complex

`stationaryProbabilities_cached`

Stationary probability caching function

Description

Stationary probability caching function

Usage

```
stationaryProbabilities_cached(
  m = c(0.2, 0.3),
  lambda = c(1, 2),
  tmax = 12,
  nmax = 5
)
```

Arguments

`m`	distribution parameters vector of sojourn times in alternating environment states
`lambda`	Poisson flow intensity vector
`tmax`	upper integration limit
`nmax`	limitation for number of arriving claims

Value

stationary probability vector cached

Index