Package 'isingLenzMC'

October 13, 2022

Version 0.2.5

Date 2015-04-08

Title Monte Carlo for Classical Ising Model

Maintainer Mehmet Suzen <mehmet.suzen@physics.org>

Depends R (>= 3.0)

NeedsCompilation yes

BuildVignettes yes

Description Classical Ising Model is a land mark system in statistical physics.The model explains the physics of spin glasses and magnetic materials, and cooperative phenomenon in general, for example phase transitions and neural networks.This package provides utilities to simulate one dimensional Ising Model with Metropolis and Glauber Monte Carlo with single flip dynamics in periodic boundary conditions. Utility functions for exact solutions are provided.

License GPL (>= 3)

Author Mehmet Suzen [aut, cre]

Repository CRAN

Date/Publication 2016-07-02 03:03:29

R topics documented:

`flipConfig1D` *Given Flip a site randomly*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, flip any of the site randomly. The function uses default RNG (Marsienne-Twister) unless changed by the user, within R, to generate a vector that contains 1s or -1s. This function calls 'flipConfig1D' C function.

Usage

```
flipConfig1D(x)
```

Arguments

x	1D Spin sites on the lattice

Value

Returns vector that contains 1s or -1s.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n          <- 10 # 10 spin sites
mySites    <- genConfig1D(n) # Generate sites
# now flip
mySitesNew <- flipConfig1D(mySites)
```

flipConfig1Dmany *Flip a single site randomly many times*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, flip any of the site randomly, repeat it many times. The function uses default RNG (Marsienne-Twister) unless changed by the user, within R, to generate a vector that contains 1s or -1s. This function calls 'flipConfig1Dmany' C function.

Usage

```
flipConfig1Dmany(x, upperF)
```

Arguments

x	1D spin sites on the lattice.
upperF	The number of times

Value

Returns vector that contains 1s or -1s.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n          <- 10 # 10 spin sites
mySites    <- genConfig1D(n) # Generate sites
# now flip 100 times
mySitesNew <- flipConfig1Dmany(mySites, 100)
```

flipConfig1D_R *Given Flip a site randomly*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, flip any of the site randomly. The function uses default RNG (Marsienne-Twister) unless changed by the user, within R, to generate a vector that contains 1s or -1s. This function is a pure R implementation

Usage

```
flipConfig1D_R(x)
```

Arguments

x	1D Spin sites on the lattice

Value

Returns vector that contains 1s or -1s.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n          <- 10 # 10 spin sites
mySites    <- genConfig1D_R(n) # Generate sites
# now flip
mySitesNew <- flipConfig1D_R(mySites)
```

---

genConfig1D *Generate one dimensional spin sites randomly*

Description

The function uses default RNG (Marsienne-Twister) unless changed by the user, within R, to generate a vector that contains 1 or -1. This reflects spin sites. This function calls 'genConfig1D' C function.

Usage

```
genConfig1D(n)
```

Arguments

n	The number of spin sites on the lattice.

Value

Returns vector that contains 1s or -1s.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n <- 10 # 10 spin sites
genConfig1D(n)
```

genConfig1D_R *Generate one dimensional spin sites randomly*

Description

The function uses default RNG (Marsienne-Twister) unless changed by the user, within R, to generate a vector that contains 1 or -1. This reflects spin sites. This function is pure R implementation.

Usage

```
genConfig1D_R(n)
```

Arguments

n	The number of spin sites on the lattice.

Value

Returns vector that contains 1s or -1s.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n <- 10 # 10 spin sites
genConfig1D_R(n)
```

genUniform *Get uniformly a spin state*

Description

Generate a single spin state from uniform distribution.

Usage

```
genUniform(n)
```

Arguments

n	dummy argument

Value

Returns randomly 1 or -1 from uniform distribution.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
genUniform()
```

`isPerform1D` *Perform metropolis MC on 1D Ising model*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, and an other flip sites, perform Metropolis Monte Carlo applying periodic boundary conditions, i.e., cyclic. This function calls the C function 'isPerform1D'.

Usage

```
isPerform1D(ikBT, x, J, H, nstep, ensembleM, probSel)
```

Arguments

`ikBT`	1/kB*T (Boltzmann factor)
`x`	1D Spin sites on the lattice.
`J`	Interaction strength
`H`	External field
`nstep`	Number of MC steps requested
`ensembleM`	Value of the theoretical magnetization (could be thermodynamic limit value)
`probSel`	Which transition probability to use. 1 for Metropolis 2 for Glauber

Value

Returns a pair list containing values for omegaM, Fluctuating metric vector for Magnetisation (length of naccept), naccept, number of MC steps accepted and nreject, number of MC steps rejected and times as accepted time steps. Times corresponds to times where flips occur, this is so-called transition times ('metropolis time' or 'single flip time') to judge the timings between two accepted steps.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n        <- 10 # 10 spin sites
mySites  <- genConfig1D(n) # Generate sites
output   <- isPerform1D(1.0, mySites, 1.0, 0.0, 10, 0.5, 1) # Metropolis
output   <- isPerform1D(1.0, mySites, 1.0, 0.0, 10, 0.5, 2) # Glauber
```

isStep1D *Carry one step Metropolis Monte Carlo on 1D ising model*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively and the usual thermodynamic parameters ikBt, J and H. Perform 1 step metropolis Monte Carlo, applying periodic boundary conditions, i.e., cyclic. This function calls the C function 'isStep1D'. Importance sampling is applied.

Usage

```
isStep1D(ikBT, x, J, H, probSel)
```

Arguments

ikBT	1/kB*T (Boltzmann factor)
x	1D Spin sites on the lattice.
J	Interaction strength
H	External field
probSel	Which transition probability to use. 1 for Metropolis 2 for Glauber

Value

A pair list, flip states (vec) and if step is accepted (accept).

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n             <- 10 # 10 spin sites
mySites       <- genConfig1D(n) # Generate sites
# only short-range part
isStep1D(1.0, mySites, 1.0, 0.0, 1) # Metropolis
isStep1D(1.0, mySites, 1.0, 0.0, 2) # Glauber
```

lattice1DenergyNN *Nearest-Neighbour energy in periodic boundary conditions in 1D*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, return nearest neighbour energy, applying periodic boundary conditions, i.e., cyclic. This function calls the C function 'lattice1DenergyNN'.

Usage

```
lattice1DenergyNN(x)
```

Arguments

x	1D Spin sites on the lattice

Value

Returns the nearest neighbour energy.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n          <- 10 # 10 spin sites
mySites    <- genConfig1D(n) # Generate sites
# now flip
mySitesNew <- lattice1DenergyNN(mySites)
```

lattice1DenergyNN_R *Nearest-Neighbour energy in periodic boundary conditions in 1D*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, return nearest neighbour energy, applying periodic boundary conditions, i.e., cyclic. This function is a pure R implementation.

Usage

```
lattice1DenergyNN_R(x)
```

Arguments

x	1D Spin sites on the lattice

Value

Returns the nearest neighbour energy.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n          <- 10 # 10 spin sites
mySites    <- genConfig1D_R(n) # Generate sites
nnEnergy <- lattice1DenergyNN(mySites)
```

---

sumVec *Sum given vector*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, return the sum. This function calls the C function 'sumVec'.

Usage

```
sumVec(x)
```

Arguments

x	1D Spin sites on the lattice

Value

Returns the sum, corresponding the long-range part.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n          <- 10 # 10 spin sites
mySites    <- genConfig1D(n) # Generate sites
sumVecs    <- sumVec(mySites)
```

sumVec_R *Sum given vector*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, return the sum. This function calls the C function 'sumVec'.

Usage

```
sumVec_R(x)
```

Arguments

x	1D Spin sites on the lattice

Value

Returns the sum, corresponding the long-range part.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n          <- 10             # 10 spin sites
mySites    <- genConfig1D_R(n) # Generate sites
sumVecs    <- sumVec_R(mySites)
```

totalEnergy1D *Total energy in periodic boundary conditions in 1D*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, return total energy, applying periodic boundary conditions, i.e., cyclic. This function calls the C function 'totalEnergy1D'.

Usage

```
totalEnergy1D(x, J, H)
```

Arguments

x	1D Spin sites on the lattice.
J	The strength of interaction.
H	The value of the external field.

Value

Returns the total energy.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n             <- 10 # 10 spin sites
mySites       <- genConfig1D(n) # Generate sites
# only short-range part
myTotalEnergy <- totalEnergy1D(mySites, 1.0, 0.0)
```

---

totalEnergy1D_R *Total energy in periodic boundary conditions in 1D*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, return total energy, applying periodic boundary conditions, i.e., cyclic. This function is pure R implementation.

Usage

```
totalEnergy1D_R(x, J, H)
```

Arguments

x	1D Spin sites on the lattice.
J	The strength of interaction.
H	The value of the external field.

Value

Return the total energy.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n             <- 10 # 10 spin sites
mySites       <- genConfig1D_R(n) # Generate sites
# only short-range part
myTotalEnergy <- totalEnergy1D_R(mySites, 1.0, 0.0)
```

`transferMatrix` *Compute theoretical transfer matrix*

Description

Compute transfer matrix

Usage

```
transferMatrix(ikBt, J, H)
```

Arguments

`ikBt`	1/kB*T (Boltzmann factor)
`J`	Interaction strength
`H`	External field

Value

Returns transfer matrix and its eigenvalues in a pair list.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
transferMatrix(1.0, 1.0, 0)
```

`transitionProbability1D` *Compute transition probability using Boltzmann distribution.*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, and an other flip sites, return the transition probability, applying periodic boundary conditions, i.e., cyclic. This function calls the C function 'transitionProbability1D'.

Usage

```
transitionProbability1D(ikBT, x, xflip, J, H, probSel)
```

Arguments

`ikBT`	1/kB*T (Boltzmann factor)
`x`	1D Spin sites on the lattice.
`xflip`	1D Spin sites on the lattice: after a flip.
`J`	Interaction strength
`H`	External field
`probSel`	Which transition probability to use. 1 for Metropolis 2 for Glauber

Value

Returns transition probability.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n             <- 10             # 10 spin sites
mySites       <- genConfig1D(n) # Generate sites
mySitesNew    <- flipConfig1D(mySites)
# only short-range part
transitionProbability1D(1.0, mySites, mySitesNew, 1.0, 0.0, 1) # Metropolis
transitionProbability1D(1.0, mySites, mySitesNew, 1.0, 0.0, 2) # Glauber
```

`transitionProbability1D_R` *Compute transition probability using Boltzmann distribution.*

Description

Given a vector of flip sites, 1s or -1s, representing up and down spins respectively, and an other flip sites, return the transition probability, applying periodic boundary conditions, i.e., cyclic. This function is pure R implementation.

Usage

```
transitionProbability1D_R(ikBT, x, xFlip, J, H)
```

Arguments

`ikBT`	1/kB*T (Boltzmann factor)
`x`	1D Spin sites on the lattice.
`xFlip`	1D Spin sites on the lattice: after a flip.
`J`	Interaction strength
`H`	External field

Value

Returns transition probability.

Author(s)

Mehmet Suzen <mehmet.suzen@physics.org>

Examples

```
n             <- 10 # 10 spin sites
mySites       <- genConfig1D_R(n) # Generate sites
mySitesNew    <- flipConfig1D_R(mySites)
# only short-range part
transitionProbability1D_R(1.0, mySites, mySitesNew, 1.0, 0.0)
```

Index