

# Package ‘HybridMC’

February 19, 2015

**Type** Package

**Title** Implementation of the Hybrid Monte Carlo and Multipoint Hybrid Monte Carlo sampling techniques

**Version** 0.2

**Date** 2009-05-27

**Author** Richard D. Morey

**Maintainer** Richard D. Morey <richarddmorey@gmail.com>

**Depends** utils, coda

**Description** This package is an R implementation of the Hybrid Monte Carlo and Multipoint Hybrid Monte Carlo sampling techniques described in Liu (2001): “Monte Carlo Strategies in Computing”.

**License** GPL (>= 2)

**URL** <http://www.r-project.org>

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2009-06-08 16:50:01

**NeedsCompilation** yes

## R topics documented:

hybridMC . . . . . 2

**Index** 4

---

 hybridMC

*Sample from a (log)density using hybrid Monte Carlo*


---

### Description

hybridMC() samples from a (log) joint density function defined up to a proportionality constant. It uses the multipoint hybrid Monte Carlo methods described by Liu (2001). The function supports a full range of tweaking options to minimize autocorrelation.

### Usage

```
hybridMC(y.start, n.samp=1, logDens, dLogDens, epsilon, LFsteps, compWeights=NULL, MPwidth=1, MPweights)
```

### Arguments

y.start	A vector of starting values
n.samp	The number of samples to draw. Each previous value is used as the starting value for the next sample
logDens	The log-density function (up to an additive constant) from which you would like to sample. logDens must return a single value
dLogDens	The function giving the derivative of the log-density function with respect to each variable. dLogDens must return a vector of the same length as y.start
epsilon	Either a single positive value giving the size of the time simulation steps, or a vector of length 2 giving the lower and upper bounds of the interval from which epsilon is uniformly sampled
LFsteps	An integer giving the number of leapfrog simulation steps to do
compWeights	The “masses” of the dimensions. Must be either a single numeric value or a vector of the same length as y.start
MPwidth	The (integer) size of the multipoint window. The default is 1, meaning no multipoint. MPwidth must be less than or equal to LFsteps.
MPweights	A vector of length MPwidth of constants, used to weight the proposal values within the multipoint window. If only a single value is passed, the values are weighted evenly
progress	An integer giving the number of samples between updates to a text progress bar. If progress=0, no progress bar is displayed
...	Arguments to pass to logDens and dLogDens

### Details

The density should have support of the whole real line for every dimension. The quality of the samples can be improved by tweaking epsilon, LFsteps, MPwidth, and to a lesser extent, compWeights and MPweights.

If you use the progress bar, keep in mind that updating the progress bar takes a little time. Too many updates will slow down your sampling, so pick a reasonable value for progress.

**Value**

hybridMC() returns an object of type mcmc (from the coda package). Each row is a sample from the joint density.

**Author(s)**

Richard D. Morey

**References**

Liu (2001) "Monte Carlo strategies in scientific computing"

**Examples**

```
#### Example 1: Jointly sample from two independent double exponentials

## log density function for double exponential
L = function(x)
  -sum(abs(x))

dL = function(x)
  -sign(x)

startVal=c(-1,1)
samples = hybridMC(y.start=startVal, n.samp=5000, logDens=L, dLogDens=dL, epsilon=.2, LFsteps=10)

# Plot the MCMC chains and densities
plot(samples)

# Plot a histogram of the first variable, with true density
hist(samples[,1],freq=FALSE,breaks=50)
x = seq(-5,5,len=100)
lines(x,.5*dexp(abs(x)),col="red") #true density in red

# Autocorrelation function
acf(samples)
```

# Index

\*Topic **distribution**

hybridMC, [2](#)

\*Topic **misc**

hybridMC, [2](#)

\*Topic **multivariate**

hybridMC, [2](#)

hybridMC, [2](#)