Package 'BarBorGradient'

January 20, 2025

Type Package

Title Function Minimum Approximator

Version 1.0.5

Date 2017-04-18

Author János Hevner

Maintainer János Hevner <hevnerjanos@hotmail.com>

Description Tool to find where a function has its lowest value(minimum). The functions can be any dimensions. Recommended use is with eps=10^-10, but can be run with 10^-20, although this depends on the function. Two more methods are in this package, simple gradient method (Gradmod) and Powell method (Powell). These are not recommended for use, their purpose are purely for comparison.

License GPL-3

Depends R (>= 3.0.0)

Imports stats

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-04-24 12:19:44 UTC

Contents

BarBor — *How to use BarBor function minimum approximator.*

Description

Approximate a functions minimum with double monoton method.

Usage

```
BarBor(exp,eps,x,v,n)
```

Arguments

exp	Expression of the function to be minimized.
eps	Precision of the approximation, recommended value is 10^-10.
x	Starting point of the approximation.
v	A character vector of the functions variables, for instance the two dimension fuction x1*x1+10*x2*x2 needs a c("x1","x2") vector.
n	Maximum setps to make while approximating, if the calculation reaches this number it exits with the current value and point. Recommended to be 10000.

Examples

```
test1 = expression(x1*x1+10*x2*x2)
eps = 10^-15
x = c(3,4)
v = c("x1","x2")
n = 10000
BarBor(test1,eps,x,v,n)
```

BarBorNoPrint — *The BarBor funtcion without printing.*

Description

Same approximation method as the BarBor function, but this doesn't print out anything. Its recommended use is for timing the approximation.

Usage

```
BarBorNoPrint(exp,eps,x,v,n)
```

Arguments

exp	Expression of the function to be minimized.
eps	Precision of the approximation, recommended value is 10^-10.
x	Starting point of the approximation.
v	A character vector of the functions variables, for instance the two dimension fuction x1*x1+10*x2*x2 needs a c("x1","x2") vector.
n	Maximum setps to make while approximating, if the calculation reaches this number it exits with the current value and point. Recommended to be 10000.

Examples

```
test1 = expression(x1*x1+10*x2*x2)
eps = 10^-15
x = c(3,4)
v = c("x1","x2")
n = 10000
BarBorNoPrint(test1,eps,x,v,n)
```

Gradient Method *Gradient method for function minimum approximation.*

Description

Gradient method for approximating a functions minimum value. The purpose of this method is to compare its result with the BarBor method.

Usage

```
Gradmod(exp,eps,G,B,m,x,v,n)
```

Arguments

exp	Expression of the function to be minimized.
eps	Precision of the approximation, recommended value is 10^-10.
G	Inner approximation coefficient, recommended value is 10^-2.
B	Inner approximation coefficient, recommended value is 0.5.
m	Inner steps, recommended value is 20.
x	Starting point of the approximation.
v	A character vector of the functions variables. Exmaple: the two dimension fuction x1*x1+10*x2*x2 needs a c("x1","x2") vector.
n	Maximum setps to make while approximating, if the calculation reaches this number it exits with the current value and point. Recommended to be 10000.

Examples

```
test1 = expression(x1*x1+10*x2*x2)
eps = 10^-10
G = 10^-2
B = 0.5
m = 20
x = c(3,4)
v = c("x1","x2")
n = 10000
Gradmod(test1,eps,G,B,m,x,v,n)
```

`Powell` *Powell's method for finding a functions local minimum.*

Description

Powell's method for finding a function local minimum. The function need not be differentiable, and no derivatives are taken. The function must be a real-valued function of a fixed number of real-valued inputs.

Usage

```
Powell(exp,eps,G,eta,m,k,x,v,n)
```

Arguments

`exp`	Expression of the function to be minimized.
`eps`	Precision of the approximation, recommended value is 10^-10.
`G`	Inner approximation coefficient, recommended value is 10^-2.
`eta`	Inner approximation coefficient, recommended value is G*2.
`m`	Inner steps, recommended value is 20.
`k`	Second inner approximation steps, recommended value is 20.
`x`	Starting point of the approximation.
`v`	A character vector of the functions variables. Exmaple: the two dimension fuction x1*x1+10*x2*x2 needs a c("x1","x2") vector.
`n`	Maximum setps to make while approximating, if the calculation reaches this number it exits with the current value and point. Recommended to be 10000.

Examples

```
test1 = expression(100*(x1*x1-x2)*(x1*x1-x2)+(1-x1)*(1-x1))
eps = 10^-5
G = 10^-2
eta = G *2
m = 20
k = 20
```

```
n = 10000
max = 1000
x = c(1,1)
v = c("x1","x2")
Powell(test1,eps,G,eta,m,k,x,v,n)
```

Index