brent-search Documentation

Release 1.0.18

Danilo Horta

April 26, 2017

Contents

1 Install **3**

2 Examples **5**

2.1 Functions 5

Python Module Index **9**

You can get the source and open issues on Github.

CHAPTER 1

Install

The recommended way of installing it is via conda

```
conda install -c conda-forge brent-search
```

An alternative way would be via pip

```
pip install brent-search
```

CHAPTER 2

Examples

```
>>> from brent_search import bracket
>>> def f(x):
...     return (x-2)**2
>>>
>>> bracket(f)
((1.2499997019767761, 2.499999701976776, 4.999999701976776, 0.5625004470349246, 0.24999970197686494,
```

Functions

brent_search.**bracket**(*f*, *x0=None*, *x1=None*, *a=-inf*, *b=inf*, *gfactor=2.0*, *rtol=1.4902e-08*, *atol=1.4902e-08*, *maxiter=500*)

Find a bracketing interval.

A bracket is defined as any three strictly increasing points `(x0, x1, x2)` such that `f(x0) > f(x1) < f(x2)`.

Exit code:

- 0: unknown
- 1: found bracket
- 2: hit the boundary
- 3: too close points
- 4: maxiter reached
- 5: not strictly convex function

Parameters

- **f** (*callable*) – function of interest.
- **x0** (`float`, optional) – first point.
- **x1** (`float`, optional) – second point.
- **a** (`float`, optional) – interval's lower limit. Defaults to `-inf`.
- **b** (`float`, optional) – interval's upper limit. Defaults to `+inf`.
- **gfactor** (`float`, optional) – growing factor.
- **rtol** (`float`, optional) – relative tolerance. Defaults to `1.4902e-08`.

- **atol** (float, optional) – absolute tolerance. Defaults to `1.4902e-08`.
- **maxiter** (int, optional) – maximum number of iterations. Defaults to `500`.

Returns A tuple containing the found solution (if any) in the first position and the exit code in the second position: `((x0, x1, x2, f0, f1, f2), ecode)`.

`brent_search.`**`brent`**(*f*, *a=-inf*, *b=inf*, *fa=None*, *fb=None*, *x0=None*, *f0=None*, *rtol=1.4902e-08*, *atol=1.4902e-08*, *maxiter=500*)

Seeks a local minimum of a function f in a closed interval [a, b] via Brent's method.

Given a function f with a minimum in the interval the a <= b, seeks a local minima using a combination of golden section search and successive parabolic interpolation.

Let `tol = rtol * abs(x0) + atol`, where `x0` is the best guess found so far. It converges if evaluating a next guess would imply evaluating `f` at a point that is closer than `tol` to a previously evaluated one or if the number of iterations reaches `maxiter`.

Parameters

- **f** (*object*) – Objective function to be minimized.
- **a** (float, optional) – interval's lower limit. Defaults to `-inf`.
- **b** (float, optional) – interval's upper limit. Defaults to `+inf`.
- **rtol** (*float*) – relative tolerance. Defaults to 1.4902e-08.
- **atol** (*float*) – absolute tolerance. Defaults to 1.4902e-08.
- **maxiter** (*int*) – maximum number of iterations

Returns best guess for the minimum of f. float: value of f evaluated at the best guess. int: number of iterations performed.

Return type float

References

- http://people.sc.fsu.edu/~jburkardt/c_src/brent/brent.c
- Numerical Recipes 3rd Edition: The Art of Scientific Computing
- https://en.wikipedia.org/wiki/Brent%27s_method

`brent_search.`**`minimize`**(*f*, *x0=None*, *x1=None*, *a=-inf*, *b=inf*, *gfactor=2*, *rtol=1.4902e-08*, *atol=1.4902e-08*, *maxiter=500*)

Function minimization.

Applies *brent_search.bracket()* and then *brent_search.brent()* to find the minimum.

Parameters

- **f** (*callable*) – function of interest.
- **x0** (float, optional) – first point.
- **x1** (float, optional) – second point.
- **a** (float, optional) – interval's lower limit. Defaults to `-inf`.
- **b** (float, optional) – interval's upper limit. Defaults to `+inf`.
- **gfactor** (float, optional) – growing factor.

- **rtol** (`float`, optional) – relative tolerance. Defaults to `1.4902e-08`.
- **atol** (`float`, optional) – absolute tolerance. Defaults to `1.4902e-08`.
- **maxiter** (`int`, optional) – maximum number of iterations. Defaults to `500`.

Returns A tuple containing the found solution (if any) in the first position, the function evaluated at that point, and the number of function evaluations.

Python Module Index

b

brent_search, 5

Index

B

bracket() (in module brent_search), 5
brent() (in module brent_search), 6
brent_search (module), 5

M

minimize() (in module brent_search), 6