

# 二分法による 1 変数方程式の解

青木繁伸

2020 年 3 月 17 日

## 1 目的

一変数関数  $f(x) = 0$  について、指定した範囲内にある全ての解を二分法により求める。

**Python** には `scipy.optimize.bisect` という関数がある。

## 2 使用法

```
import sys
sys.path.append("statlib")
from misc import bisection
bisection(func, lower, upper, ndiv=50, epsilon=1e-14, max_iteration=500, verbose=True)
```

### 2.1 引数

<code>func</code>	関数定義
<code>lower</code>	解を求める下限値
<code>upper</code>	解を求める上限値
<code>ndiv</code>	区間を細区分する個数 (デフォルトは 50) 近接した解を求めるためには、区分数を大きくしないといけない (むやみに大きくする必要はない、まえもって、図を描くなりして解の存在範囲を押さえておいて決める方がよい)。
<code>epsilon</code>	解の許容誤差 (デフォルトは $1e-14$ すなわち $10^{-14}$ )
<code>max_iteration</code>	解を反復により求める反復回数上限値 (デフォルトは 500)
<code>verbose</code>	必要最小限のプリント出力をする (デフォルトは <code>True</code> )

### 2.2 戻り値

解のリスト

### 3 使用例

#### 3.1 例 1

$f(x) = (x + 6.7)(x - 3.4)(x - \sqrt{2}) = 0$  の解を求める。

```
import numpy as np
import sys
sys.path.append("statlib")
from misc import bisection

def func(x):
    return (x+6.7)*(x-3.4)*(x-np.sqrt(2))

bisection(func, -10, 10)
```

```
ans. = -6.7000000000000008
```

```
ans. = 1.414213562373094
```

```
ans. = 3.4000000000000005
```

```
[-6.7000000000000008, 1.414213562373094, 3.4000000000000005]
```

**Python** には `scipy.optimize.bisect()` があるが、指定範囲の解は 1 個しか求まらない。全ての解を求めるためには、探索区間を変えて複数回の実行が必要になる

```
import numpy as np
from scipy.optimize import bisect
bisect(func, -10, 10)
```

```
-6.6999999999999991
```

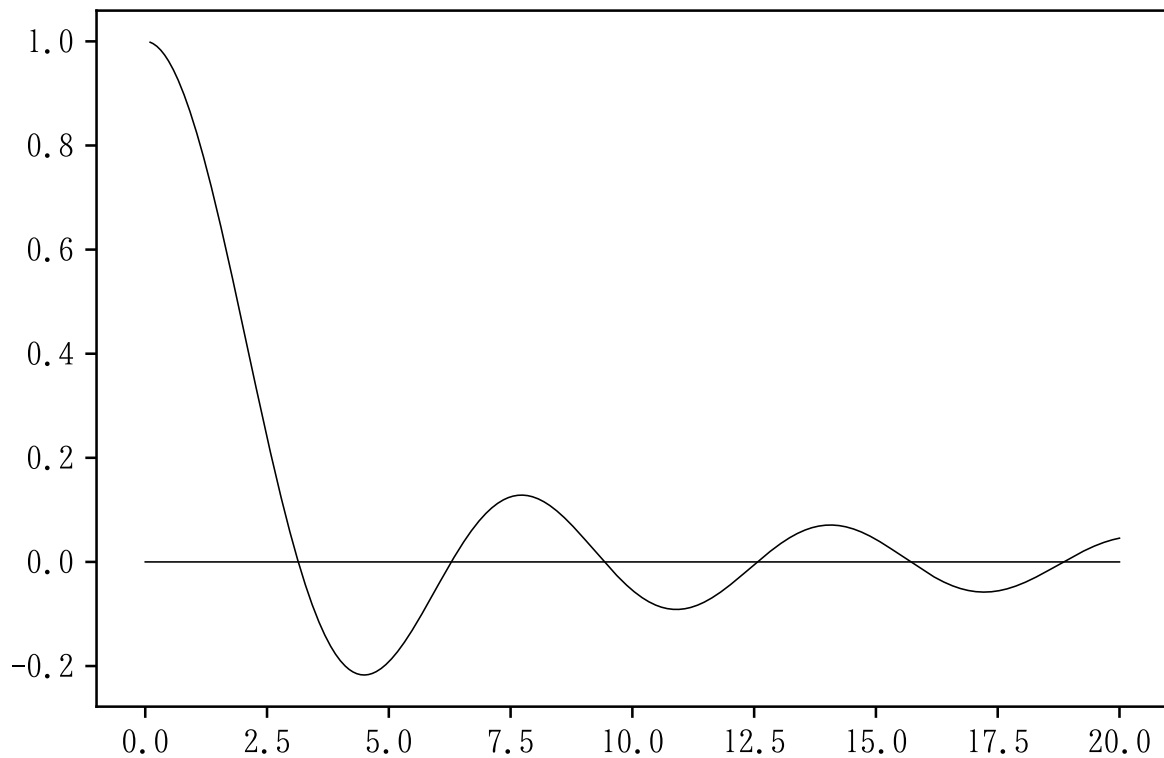
#### 3.2 例 2

$f(x) = \frac{\sin(x)}{x} = 0$  の解を求める。

```
import numpy as np
import matplotlib.pyplot as plt

def func(x):
    return np.sin(x)/x

x = np.linspace(0.1, 20, 1000)
y = [func(z) for z in x]
plt.plot(x, y, color="black", linewidth=0.5)
plt.plot([0, 20], [0, 0], color="black", linewidth=0.5)
```



```
import sys
sys.path.append("statlib")
from misc import bisection

a = bisection(func, 0.001, 20, ndiv=50)
```

```
ans. = 3.1415926535897722
```

```
ans. = 6.2831853071795605
```

```
ans. = 9.42477796076944
```

```
ans. = 12.566370614359133
```

```
ans. = 15.707963267949014
```

```
ans. = 18.849555921538894
```

```
b = [print("x = {0:8.5f}, f(x) = {1:3g}".format(x, func(x))) for x in
a]
```

```
x = 3.14159, f(x) = 6.68281e-15
```

```
x = 6.28319, f(x) = -4.13836e-15
```

```
x = 9.42478, f(x) = -6.36925e-15
```

```
x = 12.56637, f(x) = -3.14886e-15
```

```
x = 15.70796, f(x) = -3.01435e-15
```

```
x = 18.84956, f(x) = 7.12316e-15
```