牛顿迭代解非线性方程组

PB16030899 - 朱河勤
2018-5-20

书上 p186, 附录 1 程序 10

程序源码

```
import sympy
import numpy as np
from math import sqrt

def solveNonlinearEquations(funcs:[sympy.core],init_dic:dict,epsilon:float=0.001,maxtime:int=5
    '''solve  nonlinear equations:'''
    li = list(init_dic.keys())
    delta = {i:0 for i in li}
    ct = 0
    while 1:
        ct+=1
        ys = np.array([f.subs(init_dic) for f in funcs],dtype = 'float')
        mat = np.matrix([[i.diff(x).subs(init_dic) for x in li] for i in funcs ],dtype = 'float
        delt = np.linalg.solve(mat,-ys)
        for i,j in enumerate(delt):
            init_dic[li[i]] +=j
            delta[li[i]] = j
        print("第{}次迭代: {}".format(ct,init_dic))
        if ct>maxtime:
            print("after iteration for {} times, I still havn't reach the accurrency.\
                    Maybe this function havsn't zeropoint\n".format(ct))
            return init_dic
        if sqrt(sum(i**2 for i in delta.values()))<epsilon:return init_dic

if __name__ == '__main__':
    x,y = sympy.symbols('x y')
    funcs=[x**2+y**2-1,x**3-y]
    init = {x:0.8,y:0.6}
    epsilon = 1e-5
    print("迭代法求解非线性方程组")
    for i in funcs:
        print('| ',i, '= 0')
    print("初值: {}".format(init))
    print("精度: {}".format(epsilon))
    res = solveNonlinearEquations(funcs,init,epsilon)
    print("结果: {}".format(res))
```

算法描述

对 $f(x,y), g(x,y)$ 在 (x_0, y_0) 作二元 Taylor 展开，并取其线性部分，得到方程组

$$\begin{cases} f(x,y) \approx f(x_0,y_0) + (x-x_0)\dfrac{\partial f(x_0,y_0)}{\partial x} + (y-y_0)\dfrac{\partial f(x_0,y_0)}{\partial y} = 0 \\ g(x,y) \approx g(x_0,y_0) + (x-x_0)\dfrac{\partial g(x_0,y_0)}{\partial x} + (y-y_0)\dfrac{\partial g(x_0,y_0)}{\partial y} = 0 \end{cases}$$

令 $x - x_0 = \Delta x, y - y_0 = \Delta y$, 则有

$$\begin{cases} \Delta x \dfrac{\partial f_1(x_0,y_0)}{\partial x} + \Delta y \dfrac{\partial f_1(x_0,y_0)}{\partial y} = -f_1(x_0,y_0) \\ \Delta x \dfrac{\partial f_2(x_0,y_0)}{\partial x} + \Delta y \dfrac{\partial f_2(x_0,y_0)}{\partial y} = -f_2(x_0,y_0) \end{cases} \tag{3.7}$$

如果

$$\det(J(x_0,y_0)) = \left| \begin{array}{cc} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_1}{\partial y} \end{array} \right|_{(x_0,y_0)} \neq 0$$

解出 $\Delta x, \Delta y$

$$w_1 = w_0 + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x_0 + \Delta x \\ y_0 + \Delta y \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

再列出方程组

$$\begin{cases} \dfrac{\partial f(x_1,y_1)}{\partial x}(x-x_1) + \dfrac{\partial f(x_1,y_1)}{\partial y}(y-y_1) = -f(x_1,y_1) \\ \dfrac{\partial g(x_1,y_1)}{\partial x}(x-x_1) + \dfrac{\partial g(x_1,y_1)}{\partial y}(y-y_1) = -g(x_1,y_1) \end{cases}$$

解出

$$\Delta x = x - x_1, \quad \Delta y = y - y_1$$

$$w_2 = \begin{pmatrix} x_1 + \Delta x \\ y_1 + \Delta x \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$$

继续做下去，每一次迭代都是解一个类似式 (3.7) 的方程组

$$J(x_k, y_k) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} -f(x_k,y_k) \\ -g(x_k,y_k) \end{pmatrix}$$

$$\Delta x = x_{k+1} - x_k, \quad \Delta y = y_{k+1} - y_k$$

直到 delta x, delta y 的绝对值小于 epsilon

测试结果