**FixedPoint Method**

这次介绍的是不动点迭代法，由于这种迭代法的收敛性与方程的迭代格式有关，所以有的迭代格式不会导致结果收敛。**所以这种方法了解即可，一般在实际计算中不会用到。**

不动点迭代的思想很简单，如下所示。

将方程写成f(x)=x的迭代格式，给定迭代初始值x0，对函数f(x)进行如下迭代。

**不动点迭代**

x0 = 初始估计值

x = f(x)

直到前后两次的迭代值的误差小于预先的给定值，停止迭代。

下面给出具体实例，求方程x^3 + x – 1 = 0的根。给出此方程的三种迭代格式。

第一种：x = 1 - x^3，此种迭代格式不收敛

第二种：x = ( 1 - x^3 )^(1/3) ，此种迭代格式收敛

第三种：x = ( 1 + 2\*x^3 ) / ( 1 + 3\*x^2 ) ，此种迭代格式不仅收敛速度快，精度还高。下面的代码用的是第三种格式。

**fortran code**

Program FixedPointIteration

!// 使用不动点迭代方法求解方程 x^3 + x - 1 = 0 的零点

Implicit none

Real(kind=8), parameter :: eps = 1.d-8

Real(kind=8), external :: func

Real(kind=8) :: xtemp, x0 = 0.5d0 !// x0为迭代初始值

Do

xtemp = func ( x0 )

If ( abs( xtemp - x0 ) < eps ) exit

x0 = xtemp

End do

Write ( \*,'(a,f13.9)' ) ' the result of fixed point iteration is', xtemp

End program FixedPointIteration

Real(kind=8) function func ( x )

Implicit none

Real(kind=8), intent(in) :: x

func = ( 1.d0 + 2.d0 \* x \* x \* x ) / ( 1.d0 + 3.d0 \* x \* x )

End function func

**Python code**

|  |  |
| --- | --- |
|  | """ |
|  | Created on Thu Apr 12 20:10:20 2018 |
|  |  |
|  | @author: luk |
|  | """ |
|  |  |
|  | # 使用不动点迭代方法求解方程 x^3 + x - 1 = 0 的零点 |
|  | eps = 1e-8 |
|  | x0 = 0.5 # 初始迭代值 |
|  |  |
|  | def func( x ): |
|  | value = ( 1.0 + 2.0 \* x\*\*3 ) / ( 1.0 + 3.0 \* x\*\*2 ) |
|  | return value |
|  |  |
|  | while True: |
|  | xtemp = func( x0 ) |
|  | if ( abs( xtemp - x0 ) < eps ): |
|  | break |
|  | x0 = xtemp |
|  |  |
|  | print ( " The solution is ", xtemp ) |
|  |  |

**结果如下**

