**第二次上机作业**

**问题描述：**

编写非线性方程求根的不动点迭代算法程序。

从不同的初始点开始，分别采用以下三种迭代方式求解方程的正根，记录迭代过程并说明原因。

（1） （2） （3）

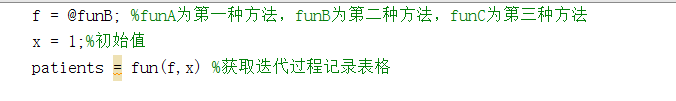
**问题分析：**

令f(x)=x²-x-1，在x=1时，f(x)=-1<0;x=2时，f(x)=1>0，则根在（1，2）范围内。不动点法中，x=g(x)存在不动点条件是在所有x∈（1，2）范围内，都有|g‘(x)|<1。（1）式中g’(x)=2x,在该范围内，不满足条件，不会收敛到不动点处。（2）式|g‘(x)|=，在该范围内，满足条件，能够成功收敛到不动点处。（3）式|g‘(x)|=，在该范围内满足条件，能够成功收敛到不动点处。

在此次作业中，我选择让结果出现5位有效数字，容限设置为5e-6。

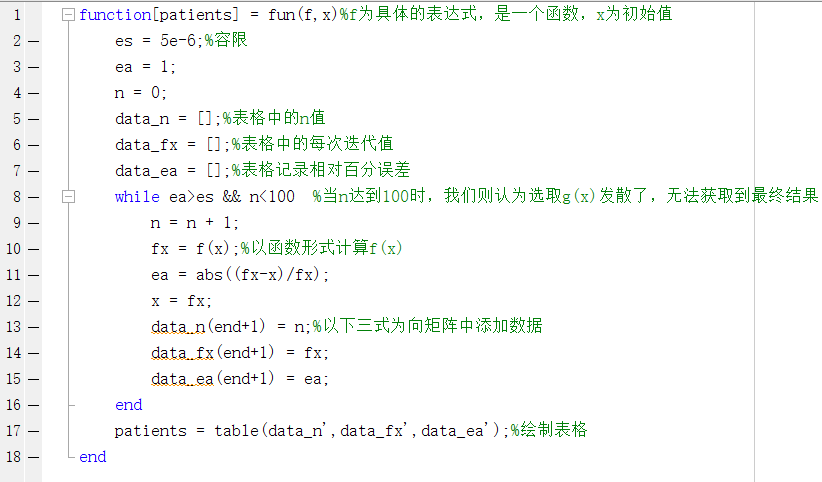
**Matlab程序：**

**主程序：**

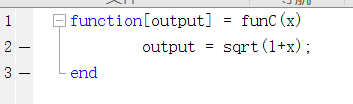
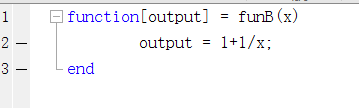
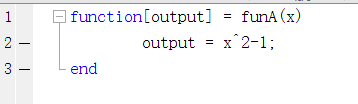


**函数文件:**

**循环计算求值并绘制表格的函数：fun()**



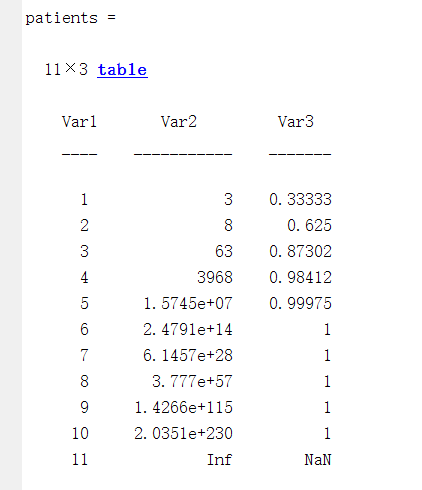
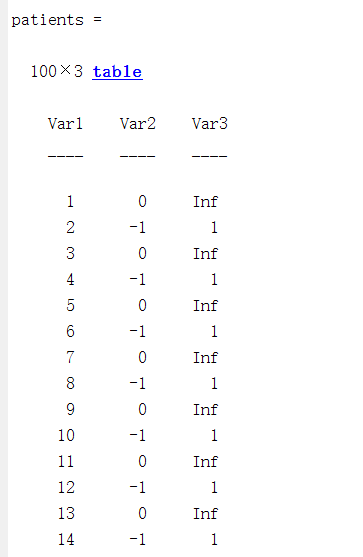
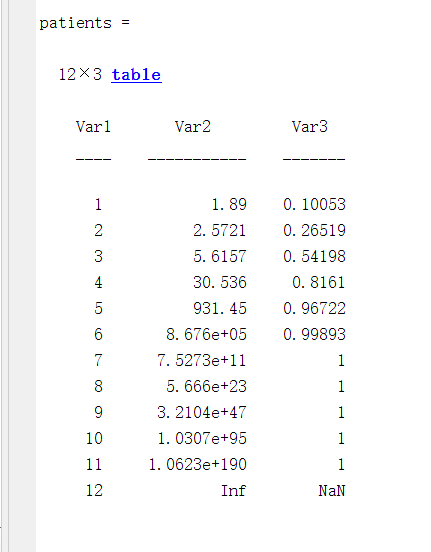
**表达式1，2，3对应函数分别为funA(),funB(),funC(),具体代码如下：**



**运行结果与分析：**

第一种方法:

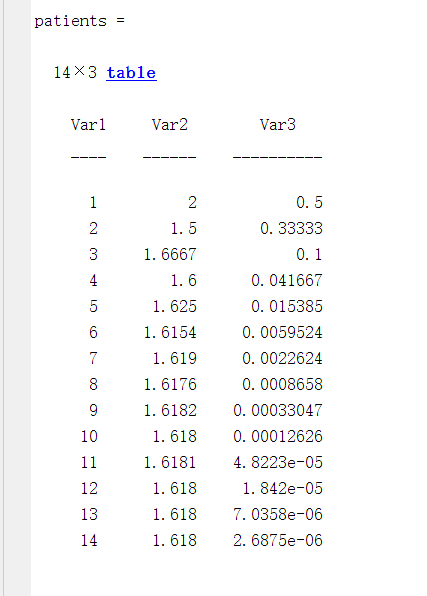
取值x在1到2内的步进为0.1的十个值，都无法得到正确的结果，运行时出现了两种情况，一种是f（x）在0和-1之间循环，另一种是f（x）达到了inf，发散了。



选择不同的初始值，会有着不同的发散快慢。

第二种方法：

同样的。我们选取在1到2之间步进长度为0.1的10个值，运行结果如下：



其中。Var1为迭代次数，Var2为x值，Var3为相对百分误差，

左图是初始值x=1的情况

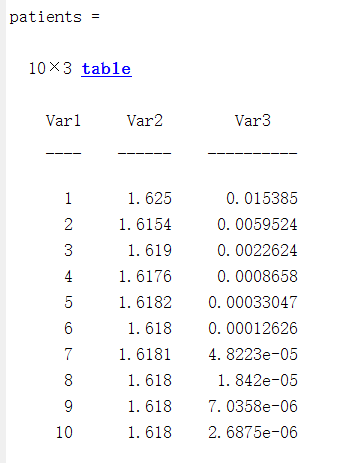
将不同初始值情况汇总如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 初始值x | 最终迭代次数 | fx值 | 相对百分误差 |
| 1.0 | 14 | 1.6180 | 2.6875e-06 |
| 1.1 | 14 | 1.6180 | 2.1215e-06 |
| 1.2 | 13 | 1.6180 | 4.2355e-06 |
| 1.3 | 13 | 1.6180 | 3.0543e-06 |
| 1.4 | 13 | 1.6180 | 1.9902e-06 |
| 1.5 | 12 | 1.6180 | 2.6875e-06 |
| 1.6 | 10 | 1.6180 | 2.6875e-06 |
| 1.7 | 11 | 1.6180 | 4.4643e-06 |
| 1.8 | 12 | 1.6180 | 3.6291e-06 |
| 1.9 | 13 | 1.6180 | 2.0627e-06 |
| 2.0 | 13 | 1.6180 | 2.6875e-06 |

从以上结果中我们可以看出，越靠近不动点值1.6180的初始值，所需最终迭代次数越少。在相同最终迭代次数时，相对百分误差也越小，收敛速度越快。

其中x=1.6时，收敛速度最快

运行结果如下：



运行时，每次迭代使得根越来越接近于真实根1.6180，并且迭代解是振荡的。方法收敛时，误差大致与前一次的迭代误差成比例，并且小于前一次的迭代误差。（线性收敛）

第三种方法：

同样的。我们选取在1到2之间步进长度为0.1的10个值。

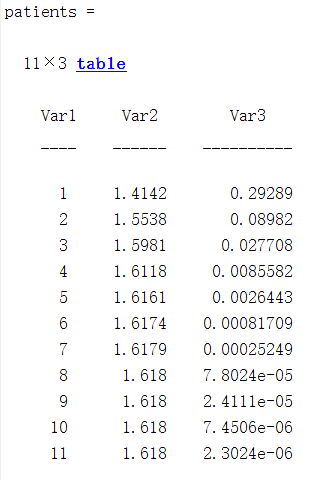
将不同初始值情况汇总如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 初始值x | 最终迭代次数 | fx值 | 相对百分误差 |
| 1.0 | 11 | 1.6180 | 2.3024e-06 |
| 1.1 | 11 | 1.6180 | 1.8981e-06 |
| 1.2 | 10 | 1.6180 | 4.8782e-06 |
| 1.3 | 10 | 1.6180 | 3.6543e-06 |
| 1.4 | 10 | 1.6180 | 2.4681e-06 |
| 1.5 | 9 | 1.6180 | 4.2614e-06 |
| 1.6 | 8 | 1.6180 | 2.0775e-06 |
| 1.7 | 9 | 1.6180 | 2.8781e-06 |
| 1.8 | 10 | 1.6180 | 1.9483e-06 |
| 1.9 | 10 | 1.6180 | 2.9801e-06 |
| 2.0 | 10 | 1.6180 | 3.9863e-06 |

从以上结果中我们同样可以看出，越靠近不动点值1.6180的初始值，所需最终迭代次数越少。在相同最终迭代次数时，相对百分误差也越小，收敛速度越快。

其中x=1时，距离真实值1.6180最远，收敛速度最慢

运行结果如下：



运行时，每次迭代使得根越来越接近于真实根1.6180，并且迭代解是单调增加的。方法收敛时，误差大致与前一次的迭代误差成比例，并且小于前一次的迭代误差。（线性收敛）

**总结：**

三种方法中后两种方法成功得到了最终结果，第一种方法最终发散了，而第三种方法要比第二种方法收敛的更迅速。这提示着我们，不动点法的使用和g（x）的选取有着很大关联，选择一个能够收敛的函数才能得到最终结果。除此以外，初始值的选择也和我们的运算速度挂钩，越接近，得到结果的速度越快，相对百分误差越小。