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$x^2-x-1=0$的正根，记录迭代过程并说明原因。

（1）$x=x^2-1$　　（2）$x=1+\frac{1}{x}$　　（3）$x=\sqrt{x+1}$

问题分析：

令 $f(x)=x^2-x-1$，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 $x\in$（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$|g'(x)|=\frac{1}{x^2}$，在该范围内，满足条件，能够成功收敛到不动点处。（3）式$|g'(x)|=\frac{1}{2\sqrt{x+1}}$，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：

```
f = @funB; %funA为第一种方法，funB为第二种方法，funC为第三种方法
x = 1;%初始值
patients = fun(f,x) %获取迭代过程记录表格
```

函数文件:

循环计算求值并绘制表格的函数：fun()

```
function[patients] = fun(f,x)%f为具体的表达式，是一个函数，x为初始值
    es = 5e-6;%容限
    ea = 1;
    n = 0;
    data_n = [];%表格中的n值
    data_fx = [];%表格中的每次迭代值
    data_ea = [];%表格记录相对百分误差
    while ea>es && n<100  %当n达到100时，我们则认为选取g(x)发散了，无法获取到最终结果
        n = n + 1;
        fx = f(x);%以函数形式计算f(x)
        ea = abs((fx-x)/fx);
        x = fx;
        data_n(end+1) = n;%以下三式为向矩阵中添加数据
        data_fx(end+1) = fx;
        data_ea(end+1) = ea;
    end
    patients = table(data_n',data_fx',data_ea');%绘制表格
end
```

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

```
function[output] = funA(x)
        output = x^2-1;
end
```

```
function[output] = funB(x)
        output = 1+1/x;
end
```

```
function[output] = funC(x)
        output = sqrt(1+x);
end
```

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。

patients =

100×3 table

Var1	Var2	Var3
1	0	Inf
2	-1	1
3	0	Inf
4	-1	1
5	0	Inf
6	-1	1
7	0	Inf
8	-1	1
9	0	Inf
10	-1	1
11	0	Inf
12	-1	1
13	0	Inf
14	-1	1

patients =

12×3 table

Var1	Var2	Var3
1	1.89	0.10053
2	2.5721	0.26519
3	5.6157	0.54198
4	30.536	0.8161
5	931.45	0.96722
6	8.676e+05	0.99893
7	7.5273e+11	1
8	5.666e+23	1
9	3.2104e+47	1
10	1.0307e+95	1
11	1.0623e+190	1
12	Inf	NaN

patients =

11×3 table

Var1	Var2	Var3
1	3	0.33333
2	8	0.625
3	63	0.87302
4	3968	0.98412
5	1.5745e+07	0.99975
6	2.4791e+14	1
7	6.1457e+28	1
8	3.777e+57	1
9	1.4266e+115	1
10	2.0351e+230	1
11	Inf	NaN

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:

```
patients =

  14×3 table

    Var1     Var2         Var3
    ----    ------    ----------

     1           2           0.5
     2         1.5       0.33333
     3      1.6667           0.1
     4         1.6      0.041667
     5       1.625      0.015385
     6      1.6154     0.0059524
     7       1.619     0.0022624
     8      1.6176     0.0008658
     9      1.6182    0.00033047
    10       1.618    0.00012626
    11      1.6181    4.8223e-05
    12       1.618     1.842e-05
    13       1.618    7.0358e-06
    14       1.618    2.6875e-06
```

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：

```
patients =

  10×3 table

    Var1     Var2       Var3
    ----    ------    ----------

     1       1.625      0.015385
     2      1.6154     0.0059524
     3       1.619     0.0022624
     4      1.6176     0.0008658
     5      1.6182    0.00033047
     6       1.618    0.00012626
     7      1.6181    4.8223e-05
     8       1.618     1.842e-05
     9       1.618    7.0358e-06
    10       1.618    2.6875e-06
```

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:

```
patients =

  11×3 table

    Var1     Var2        Var3
    ----    ------    ----------

     1      1.4142       0.29289
     2      1.5538       0.08982
     3      1.5981      0.027708
     4      1.6118     0.0085582
     5      1.6161     0.0026443
     6      1.6174    0.00081709
     7      1.6179    0.00025249
     8       1.618    7.8024e-05
     9       1.618    2.4111e-05
    10       1.618    7.4506e-06
    11       1.618    2.3024e-06
```

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。