**《数值分析》实验报告：实验七**

学号：10172911 姓名： 梁天一 班级： 数172 成绩：

|  |  |
| --- | --- |
| 实验名称：方程求根 | 实验地点：奉贤数学系机房 |
| 所使用的工具软件及环境：Matlab | |
| 一、实验目的：  1. 利用Matlab 实现非线性方程求根； | |
| 二、实验内容：  求下列方程的实根：  要求：（1）设计一种不动点迭代法，要使迭代序列收敛，然后再用斯特芬森加速迭代，计算到为止。（2）用牛顿迭代，同样计算到为止。输出迭代初值及各次迭代值和迭代次数，比较方法的优劣。 | |

|  |
| --- |
| 三、操作步骤：  程序：  function r=BuDongDieDai(DieDaiFun,x0)  r=inf;  k=0;%迭代次数  while abs(r-x0)>1e-8  r=x0;  y=DieDaiFun(x0);  x0=y;  fprintf('\n次数k=%d;方程的根x=%.11f;f(x)=%.8f;误差为%f\n',k,r,y,abs(r-x0));%保留11位小数  k=k+1;    end  end  function r=newton(f,df,x0)  r=inf;  k=0;%迭代次数  while abs(r-x0)>1e-8  r=x0;  y=f(x0);  x0=x0-f(x0)/df(x0);  fprintf('\n次数k=%d;方程的根x=%.11f;f(x)=%.8f;误差为%f\n',k,r,y,abs(r-x0));%保留11位小数  k=k+1;  abs(r-x0)  end  end  function[x\_star,index,it]=steffensen(phi,x,ep)  %斯特芬森加速迭代方法  % x为初始点  % x\_star为当迭代成功时，输出方程的根  % 当迭代失败时，输出最后的迭代值;  % index为指标变量，当index=1时，表明迭代成功  % it为迭代次数  index=0;k=1;  fprintf('\n次数k=0;方程的根x=%.11f;f(x)=%.11f\n',x,phi(x));  while k< 500  x1=x;y=feval(phi,x);z=feval(phi,y);  x=x-(y-x)^2/(z-2\*y+x);  fprintf('\n次数k=%d;方程的根x=%.11f;f(x)=%.8f;误差为%f\n',k,x,y,x-x1);%保留11位小数  if abs(x-x1)<ep  index=1;break;  end  k=k+1;  end  x\_star=x;it=k;  end    调用函数  BuDongDieDai(DieDai1,1.5);  BuDongDieDai(DieDai2,1.5);  [x\_star,inedx,it] = steffensen(DieDai1,100,1e-8);  [x\_star,inedx,it] = steffensen(DieDai2,100,1e-8);  r=newton(f1,df1,1.5);  r=newton(f2,df2,1.5);  扩展实验  修改后的函数  function k=BuDongDieDai(DieDaiFun,x0)  r=inf;  k=0;%迭代次数  while abs(r-x0)>1e-8  r=x0;  y=DieDaiFun(x0);  x0=y;  %fprintf('\n次数k=%d;方程的根x=%.11f;f(x)=%.8f;误差为%f\n',k,r,y,abs(r-x0));%保留11位小数  k=k+1;  if abs(y)>=1e29  k=500;  break;%防止数值过大  end  end  end  function k=newton(f,df,x0)  r=inf;  k=0;%迭代次数  while abs(r-x0)>1e-8  r=x0;  y=f(x0);  x0=x0-f(x0)/df(x0);  %fprintf('\n次数k=%d;方程的根x=%.11f;f(x)=%.8f;误差为%f\n',k,r,y,abs(r-x0));%保留11位小数  k=k+1;  if abs(y)>=1e29  k=500;  break;%防止数值过大  end  end  end  function k=steffensen(phi,x,ep)  %斯特芬森加速迭代方法  % x为初始点  % x\_star为当迭代成功时，输出方程的根  % 当迭代失败时，输出最后的迭代值;  % index为指标变量，当index=1时，表明迭代成功  % it为迭代次数  index=0;k=1;  %fprintf('\n次数k=0;方程的根x=%.11f;f(x)=%.11f\n',x,phi(x));  while k< 500  x1=x;  y=feval(phi,x);  z=feval(phi,y);  x=x-(y-x)^2/(z-2\*y+x);  %fprintf('\n次数k=%d;方程的根x=%.11f;f(x)=%.8f;误差为%f\n',k,x,y,x-x1);%保留11位小数  if max(abs(y),abs(z))>=1e9  k=500;  break;%防止数值过大  end  if abs(x-x1)<ep  index=1;break;  end  k=k+1;  end  x\_star=x;it=k;  end    修改后的调用程序  扩展实验:比较不同算法的迭代次数优劣  x0=-50:1:50;  n=length(x0);  K\_BuDongDieDai=zeros(n,1);  K\_steffensen=zeros(n,1);  K\_newton=zeros(n,1);  for i=1:n  K\_BuDongDieDai(i)=BuDongDieDai(DieDai1,x0(i));  K\_steffensen(i)=steffensen(DieDai1,x0(i),1e-8);  K\_newton(i)=newton(f1,df1,x0(i));  end  plot(x0,K\_BuDongDieDai,x0,K\_steffensen,x0,K\_newton)  xlabel('初值')  ylabel('迭代次数,上限500次')  legend('不动点法','斯特芬森加速迭代','牛顿迭代')  C:\Users\tianyilt\AppData\Local\Temp\ConnectorClipboard7803224921874469268\image15754633332210.png  x0=-200:1:200;  n=length(x0);  K\_BuDongDieDai=zeros(n,1);  K\_steffensen=zeros(n,1);  K\_newton=zeros(n,1);  for i=1:n  K\_BuDongDieDai(i)=BuDongDieDai(DieDai2,x0(i));  K\_steffensen(i)=steffensen(DieDai2,x0(i),1e-8);  K\_newton(i)=newton(f2,df2,x0(i));  end  plot(x0,K\_BuDongDieDai,x0,K\_steffensen,x0,K\_newton)  xlabel('初值')  ylabel('迭代次数,上限500次')  legend('不动点法','斯特芬森加速迭代','牛顿迭代')  C:\Users\tianyilt\AppData\Local\Temp\ConnectorClipboard7803224921874469268\image15754633332291.png  DieDai3=@(x)(exp(x)-2)/(x-3)  DieDai3 = *包含以下值的 function\_handle:*  @(x)(exp(x)-2)/(x-3)  x0=-200:1:200;  n=length(x0);  K\_BuDongDieDai=zeros(n,1);  K\_steffensen=zeros(n,1);  K\_newton=zeros(n,1);  for i=1:n  K\_BuDongDieDai(i)=BuDongDieDai(DieDai3,x0(i));  K\_steffensen(i)=steffensen(DieDai3,x0(i),1e-8);  K\_newton(i)=newton(f1,df1,x0(i));  end  plot(x0,K\_BuDongDieDai,x0,K\_steffensen,x0,K\_newton)  xlabel('初值')  ylabel('迭代次数,上限500次')  legend('不动点法','斯特芬森加速迭代','牛顿迭代')  C:\Users\tianyilt\AppData\Local\Temp\ConnectorClipboard7803224921874469268\image15754633332342.png  x0=-50:0.5:0;  n=length(x0);  K\_BuDongDieDai=zeros(n,1);  K\_steffensen=zeros(n,1);  K\_newton=zeros(n,1);  for i=1:n  K\_BuDongDieDai(i)=BuDongDieDai(DieDai3,x0(i));  K\_steffensen(i)=steffensen(DieDai3,x0(i),1e-8);  K\_newton(i)=newton(f1,df1,x0(i));  end  plot(x0,K\_BuDongDieDai,x0,K\_steffensen,x0,K\_newton)  xlabel('初值')  ylabel('迭代次数,上限500次')  legend('不动点法','斯特芬森加速迭代','牛顿迭代')  C:\Users\tianyilt\AppData\Local\Temp\ConnectorClipboard7803224921874469268\image15754633332423.png |
| 四、实验结果：  先上结论,迭代速度: 斯特芬森加速迭代>牛顿迭代>不动点法(甚至不动点法还不一定收敛),具体参数放置在最后  但是扩展实验粉碎了以上结论, 斯特芬森加速迭代与牛顿迭代与牛顿法的好坏不能简单下结论,事实上斯特芬森迭代依赖于不动点法的迭代公式的好坏.而且不如不动点法与牛顿法稳定,甚至在实验时发生**混沌现象**  首先给出扩展实验  公式1  C:\Users\tianyilt\AppData\Local\Temp\ConnectorClipboard7803224921874469268\image15754620706100.png  公式2  C:\Users\tianyilt\AppData\Local\Temp\ConnectorClipboard7803224921874469268\image15754620906700.png  可以看出newton法胜在稳定,但当迭代公式选择恰当时,史蒂芬森能够迭代地非常快,甚至超越牛顿法.  现在给出公式1的第二种迭代法  DieDai3=@(x)(exp(x)-2)/(x-3)  plot(x0,K\_BuDongDieDai,x0,K\_steffensen,x0,K\_newton)  xlabel('初值')  ylabel('迭代次数,上限500次')  legend('不动点法','斯特芬森加速迭代','牛顿迭代')  C:\Users\tianyilt\AppData\Local\Temp\ConnectorClipboard7803224921874469268\image15754628624440.png  可以看出史蒂芬森在-50-0范围内剧烈震荡,然后x0<-50时看出史蒂芬森比牛顿法更好,但是在大于0时,该方法反而不如牛顿法.更有趣的是,加速法发挥在-50,0范围内反而更加稳定  有趣的混沌现象    以下为非扩展部分  发现斯特芬森加速迭代,牛顿迭代>不动点法  BuDongDieDai(DieDai1,1.5);  次数k=0;方程的根x=1.50000000000;f(x)=-0.07722969;误差为1.577230  次数k=1;方程的根x=-0.07722969011;f(x)=0.36009574;误差为0.437325  次数k=2;方程的根x=0.36009573825;f(x)=0.23206743;误差为0.128028  次数k=3;方程的根x=0.23206743171;f(x)=0.26421684;误差为0.032149  次数k=4;方程的根x=0.26421684057;f(x)=0.25579999;误差为0.008417  次数k=5;方程的根x=0.25579998590;f(x)=0.25797975;误差为0.002180  次数k=6;方程的根x=0.25797974926;f(x)=0.25741365;误差为0.000566  次数k=7;方程的根x=0.25741364782;f(x)=0.25756056;误差为0.000147  次数k=8;方程的根x=0.25756056113;f(x)=0.25752243;误差为0.000038  次数k=9;方程的根x=0.25752242729;f(x)=0.25753233;误差为0.000010  次数k=10;方程的根x=0.25753232508;f(x)=0.25752976;误差为0.000003  次数k=11;方程的根x=0.25752975604;f(x)=0.25753042;误差为0.000001  次数k=12;方程的根x=0.25753042285;f(x)=0.25753025;误差为0.000000  次数k=13;方程的根x=0.25753024977;f(x)=0.25753029;误差为0.000000  次数k=14;方程的根x=0.25753029470;f(x)=0.25753028;误差为0.000000  次数k=15;方程的根x=0.25753028304;f(x)=0.25753029;误差为0.000000  BuDongDieDai(DieDai2,1.5);  次数k=0;方程的根x=1.50000000000;f(x)=1.31147541;误差为0.188525  次数k=1;方程的根x=1.31147540984;f(x)=1.39441634;误差为0.082941  次数k=2;方程的根x=1.39441633877;f(x)=1.35747562;误差为0.036941  次数k=3;方程的根x=1.35747562065;f(x)=1.37384422;误差为0.016369  次数k=4;方程的根x=1.37384421645;f(x)=1.36657422;误差为0.007270  次数k=5;方程的根x=1.36657421582;f(x)=1.36979982;误差为0.003226  次数k=6;方程的根x=1.36979982286;f(x)=1.36836801;误差为0.001432  次数k=7;方程的根x=1.36836800600;f(x)=1.36900345;误差为0.000635  次数k=8;方程的根x=1.36900344720;f(x)=1.36872141;误差为0.000282  次数k=9;方程的根x=1.36872141259;f(x)=1.36884659;误差为0.000125  次数k=10;方程的根x=1.36884658600;f(x)=1.36879103;误差为0.000056  次数k=11;方程的根x=1.36879103019;f(x)=1.36881569;误差为0.000025  次数k=12;方程的根x=1.36881568737;f(x)=1.36880474;误差为0.000011  次数k=13;方程的根x=1.36880474380;f(x)=1.36880960;误差为0.000005  次数k=14;方程的根x=1.36880960087;f(x)=1.36880745;误差为0.000002  次数k=15;方程的根x=1.36880744517;f(x)=1.36880840;误差为0.000001  次数k=16;方程的根x=1.36880840193;f(x)=1.36880798;误差为0.000000  次数k=17;方程的根x=1.36880797729;f(x)=1.36880817;误差为0.000000  次数k=18;方程的根x=1.36880816576;f(x)=1.36880808;误差为0.000000  次数k=19;方程的根x=1.36880808211;f(x)=1.36880812;误差为0.000000  次数k=20;方程的根x=1.36880811923;f(x)=1.36880810;误差为0.000000  次数k=21;方程的根x=1.36880810276;f(x)=1.36880811;误差为0.000000  [x\_star,inedx,it] = steffensen(DieDai1,1.5,1e-8);  次数k=0;方程的根x=1.50000000000;f(x)=-0.07722969011  次数k=1;方程的根x=0.26515987448;f(x)=-0.07722969;误差为-1.234840  次数k=2;方程的根x=0.25753168845;f(x)=0.25555679;误差为-0.007628  次数k=3;方程的根x=0.25753028544;f(x)=0.25752992;误差为-0.000001  次数k=4;方程的根x=0.25753028544;f(x)=0.25753029;误差为-0.000000  [x\_star,inedx,it] = steffensen(DieDai2,1.5,1e-8);  次数k=0;方程的根x=1.50000000000;f(x)=1.31147540984  次数k=1;方程的根x=1.36907537571;f(x)=1.31147541;误差为-0.130925  次数k=2;方程的根x=1.36880810892;f(x)=1.36868949;误差为-0.000267  次数k=3;方程的根x=1.36880810782;f(x)=1.36880811;误差为-0.000000  r=newton(f1,df1,1.5);  次数k=0;方程的根x=1.50000000000;f(x)=-4.73168907;误差为1.055783  次数k=1;方程的根x=0.44421745996;f(x)=-0.69459276;误差为0.189219  次数k=2;方程的根x=0.25499815976;f(x)=0.00957034;误差为0.002532  次数k=3;方程的根x=0.25752968467;f(x)=0.00000227;误差为0.000001  次数k=4;方程的根x=0.25753028544;f(x)=0.00000000;误差为0.000000  r=newton(f2,df2,1.5);  次数k=0;方程的根x=1.50000000000;f(x)=2.87500000;误差为0.126374  次数k=1;方程的根x=1.37362637363;f(x)=0.10178868;误差为0.004812  次数k=2;方程的根x=1.36881481962;f(x)=0.00014159;误差为0.000007  次数k=3;方程的根x=1.36880810783;f(x)=0.00000000;误差为0.000000 用极端取值证明斯特芬森加速迭代>牛顿迭代 steffensen(DieDai2,50,1e-8)  次数k=0;方程的根x=50.00000000000;f(x)=0.00766283525  次数k=1;方程的根x=1.92080138969;f(x)=0.00766284;误差为-48.079199  次数k=2;方程的根x=1.37349968442;f(x)=1.14083098;误差为-0.547302  次数k=3;方程的根x=1.36880844776;f(x)=1.36672696;误差为-0.004691  次数k=4;方程的根x=1.36880810782;f(x)=1.36880796;误差为-0.000000  次数k=5;方程的根x=1.36880810782;f(x)=1.36880811;误差为-0.000000  ans = 1.3688  newton(f2,df2,50);  次数k=0;方程的根x=50.00000000000;f(x)=130480.00000000;误差为16.923476  次数k=1;方程的根x=33.07652399481;f(x)=38686.46219453;误差为11.297048  次数k=2;方程的根x=21.77947626941;f(x)=11477.48438203;误差为7.550208  次数k=3;方程的根x=14.22926795976;f(x)=3408.26610396;误差为5.054275  次数k=4;方程的根x=9.17499250542;f(x)=1012.46624168;误差为3.383443  次数k=5;方程的根x=5.79154980027;f(x)=299.26004388;误差为2.236750  次数k=6;方程的根x=3.55479970370;f(x)=85.74178416;误差为1.380061  次数k=7;方程的根x=2.17473915341;f(x)=21.49178022;误差为0.653495  次数k=8;方程的根x=1.52124383076;f(x)=3.36124019;误差为0.145966  次数k=9;方程的根x=1.37527762818;f(x)=0.13673776;误差为0.006457  次数k=10;方程的根x=1.36882020325;f(x)=0.00025517;误差为0.000012  次数k=11;方程的根x=1.36880810786;f(x)=0.00000000;误差为0.000000 |

任课教师： 20 年 月 日