非线性方程的数值解作业

一、 问题（方程组）：

实验 1-3：设映射 F：$R^n \to R^n$ 定义为

$$F_1(x) = x_1 - e^{\cos(\frac{1}{n+1}(x_1+x_2))}$$

$$F_i(x) = x_i - e^{\cos(\frac{1}{n+1}(x_{i-1}+x_i+x_{i+1}))}$$

$$F_n(x) = x_n - e^{\cos(\frac{1}{n+1}(x_{n-1}+x_n))}$$

实验 4：

$$f_1(x1, x2, x3) = 3x_1 - \cos(x2 * x3) - 0.5$$

$$f_2(x1, x2, x3) = x1^2 - 81(x2 + 0.1)^2 + \sin(x3) + 1.06$$

$$f_3(x1, x2, x3) = e^{-x1*x2} + 20x3 + \frac{1}{3}(10\pi - 3)$$

二、 数值实验

实验 1：用牛顿迭代法求解上述非线性方程组的解

牛顿迭代法：

$$\begin{cases} x^{k+1} = x^k + \Delta x^k \\ F'(x^k)\Delta x^k + F(\Delta x^k) = 0 \end{cases}$$

用 Matlab2018b 进行牛顿迭代法的编程，方程的维数为 100，初始值x^0=[1,……,1]，精度 $\varepsilon = 10^{-6}$。

停止准则为以下两者

(1) $\|x^{k+1} - x^k\| < \varepsilon * \|x^k\|$

(2) $\|F(x^k)\| < \varepsilon$

实验 2：用 Newton-SOR 迭代法和 SOR-Newton 迭代法，求解上述非线性方程组的解。

用 Matlab2018b 进行 Newton-SOR 迭代法和 SOR-Newton 迭代法的编程，方程的维数为 100，初始值为 x0=[0,⋯,0] [1,⋯,1] [20,⋯,20]三种，精度 $\varepsilon=10^{-6}$，内迭代次数均为 2，松弛因子 w 均为 1。

停止准则为：$\|F(x^k)\|<\varepsilon$

实验 3：用一点割线法、两点割线法和改进 n 步迭代法，求解上述非线性方程组的解。

用 Matlab2018b 进行一点割线法、两点割线法和改进 n 步迭代法的编程，方程的维数为 100，初始值为 x0=[0,⋯,0] [1,⋯,1] [20,⋯,20]三种，精度 $\varepsilon=10^{-6}$，h=0.5。

停止准则为：$\|F(x^k)\|<\varepsilon$

实验 4：用延拓牛顿法和循环中点求积牛顿法，选择初始点[0,0,0]’ 和[20,20,20]’，求解上述 3 维非线性方程组的解。其中精度 $\varepsilon=10^{-6}$。

收敛准则为 $\|F(x^k)\|<\varepsilon$ 和 $\|x^{k+1}-x^k\|<\varepsilon$

三、 实验4的程序(sess_run5.m)(continuation_newton.m)(recurrent_newton.m)

File: sess_run5.m(主程序)

```
function sess_run5
x0=[100;100;100];
N=3000;
err=1e-6;
M=10;
%[x,iter]=continuation_newton(@f,@df,x0,N,err)
[x,iter]=recurrent_newton(@f,@df,x0,N,M,err,err)

function y=f(x)
f1=3*x(1)-cos(x(2)*x(3))-0.5;
f2=x(1)^2-81*(x(2)+0.1)^2+sin(x(3))+1.06;
f3=exp(-x(1)*x(2))+20*x(3)+10*pi/3-1;
y=[f1;f2;f3];

function y=df(x)
y=[3  x(3)*sin(x(2)*x(3))  x(2)*sin(x(2)*x(3));
   2*x(1)  -162*(x(2)+0.1)  cos(x(3));
   -x(2)*exp(-x(1)*x(2))  -x(1)*exp(-x(1)*x(2))  20];
```

File: continuation_newton.m(延拓牛顿法)

```
function [x1,iter]=continuation_newton(f,df,x0,N,err)

%延拓牛顿法

x=x0;

itermax=100;

f0=f(x0);

for i=0:N-1

    x=x-df(x)\(f(x)+(i/N-1)*f0);

end

for iter=1:itermax

    x1=x-df(x)\f(x);

    if norm(x1-x)<err

        break

    end

    x=x1;

end
```

File: recurrent_newton(循环中点求积牛顿法)

```
function [x,iter] = recurrent_newton(f,df,x0,N,M,err1,err2)
```

```
%循环中点求积牛顿法¨
for iter=1:M
    f0=f(x0);
    x=x0-df(x0)\f0/N;
    for j=1:N-1
        x_half=x+(x-x0)/2;
        x0=x;
        x=x-df(x_half)\f0/N;
    end
    z0=0.1;
    z1=df(x)\f(x);
    while norm(z1)<norm(z0)
        z0=z1;
        x=x-z1;
        if norm (z1)<=err1
            return
        end
        if norm (f(x))<=err2
            return
        end
        z1=df(x)\f(x);
    end
end
```

四、 数值实验结果

实验 1(牛顿迭代法):

	停止条件 1: $\|x^{k+1}-x^k\| < \varepsilon * \|x^k\|$		停止条件 2: $\|F(x^k)\| < \varepsilon$	
初始值	迭代次数	$\|x^{k+1}-x^k\|$	迭代次数	$\|F(x^k)\|$
[0, ⋯, 0]	1	27.18281828	1	27.18281828
	2	9.207523076	2	14.74672184
	3	0.431678826	3	0.743106864
	4	3.04E-05	4	2.77E-05
	5	3.78E-13	5	3.77E-13
[1, ⋯, 1]	⋯	⋯	⋯	⋯
	3	30.33457253	3	27.92264291
	4	11.93883392	4	18.38908032
	5	0.837697626	5	1.446653964
	6	0.000274334	6	0.000473777
	7	2.47E-11	7	4.26E-11
[20, ⋯, 20]	⋯	⋯	⋯	⋯
	9	17.49526454	9	14.531269289013
	10	9.599137236	10	11.24358681
	11	1.609389575	11	2.464247536
	12	0.037460134	12	0.05483113
	13	1.33E-06	13	2.03E-06
			14	2.89E-15

实验 2（Newton-SOR 迭代法和 SOR-Newton 迭代法）结果:

寻找最好的松弛因子：（初始值：[20, ⋯, 20]）

松弛因子 w	Newton-SOR 迭代法的迭代次数	SOR-Newton 迭代法的迭代次数
0.1	90	100（不收敛）
0.2	43	85

0.3	27	53
0.4	19	37
0.5	14	28
0.6	11	21
0.7	8	16
0.8	6	12
0.9	5	9
1.0*	3（最少迭代步数）	4（最少迭代步数）
1.1	5	9
1.2	7	12
1.3	9	16
1.4	11	21
1.5	14	28
1.6	19	38
1.7	27	54
1.8	43	86
1.9	91	100（不收敛）

所以，w=1.0（保留一位小数）是最佳松弛因子。

实验 2 数值实验结果：

松弛因子 w=1	Newton-SOR 迭代法		SOR-Newton 迭代法	
初始值	迭代次数	$\|F(x^k)\|$	迭代次数	$\|F(x^k)\|$
[0, ⋯, 0]	1	8.7700e-02	1	4.8520e-02
	2	1.3012e-06	2	1.0374e-04
	3	5.9591e-12	3	2.2162e-07
[1, ⋯, 1]	1	3.4731e-02	1	3.2851e-02
	2	2.9977e-07	2	7.0219e-05
			3	1.5001e-07
[20, ⋯, 20]	1	2.3296e+00	1	7.4433e-01

	2	6.4473e-04	2	1.5811e-03
	3	3.0037e-09	3	3.3778e-06
			4	7.2159e-09

实验 3（一点、两点割线法和改进 n 步迭代法）数值实验结果：

	一点割线法迭代法		两点割线法迭代法		改进 n 步迭代法	
初始值	迭代次数	$\|F(x^k)\|$	迭代次数	$\|F(x^k)\|$	迭代次数	$\|F(x^k)\|$
$[0,\cdots,0]$	1	5.8529e-02	1	8.2267e-02	1	5.8529e-02
	2	2.7117e-07	2	8.7796e-05	2	6.1283e-05
			3	2.8117e-10	3	6.4473e-08
$[1,\cdots,1]$	1	2.3358e-02	1	3.1333e-02	1	2.3358e-02
	2	4.3191e-08	2	2.1032e-05	2	1.5283e-05
			3	2.5695e-11	3	1.0021e-08
$[20,\cdots,20]$	1	2.6608e+00	1	2.3180e+00	…3	2.7204e-03
	2	2.1912e-03	2	1.5038e-02	4	8.7536e-05
	3	1.4988e-09	3	1.3962e-06	5	2.8170e-06
			4	8.1975e-13	6	9.0734e-08

实验 4（延拓牛顿法和循环中点求积牛顿法）数值实验结果：

其中，N 是中点求积的步数 M 是循环中点求积牛顿法的循环次数

	延拓牛顿法(N=8)			循环中点求积法(N=8,M=10)		
初始值	迭代次数	$\|x^{k+1} - x^k\|$	CPU TIME	迭代次数	$\|F(x^k)\|$	CPU TIME

[0,0,0]	1	9.0491e-02	2.493 ms	1	3.9958e-06	4.552ms
	2	2.8364e-04		2	4.9472e-12	
	3	4.0231e-07		3		

当初始点[20,20,20]距离 x*比较远时，需要取较大的 N 才能使得算法收敛这里我们延拓牛顿法取 N=50，循环中点求积法取 N=500，M=10。

	延拓牛顿法(N=50)			循环中点求积法(N=8,M=10)		
初始值	迭代次数	$\|x^{k+1}-x^k\|$	CPU TIME	迭代次数	$\|F(x^k)\|$	CPU TIME
[20,20,20]	……7	2.8983e-03	9.802 ms	1	1.5665e-06	26.926 ms
	8	4.2050e-05		2	7.6028e-13	
	9	8.8550e-09				

五、结论

实验 1 结论：在使用牛顿迭代法时，在收敛速度上，停止条件 2 往往比停止条件 1 更加严格，可能需要更多的迭代步数。

实验 2 结论： 在解上述非线性方程组时，最好的松弛因子是 w=1.0。在收敛速度上，Newton-SOR 迭代法往往比 SOR-Newton 迭代法更快的收敛。

实验 3 结论：在收敛速度上，一点割线法法最快，两点割线法法次之，改进 n 步迭代法最慢。而两点割线法法因为使用了之前两点的信息，比一点割线法计算的函数值要更少，而改进 n 步迭代法主要是为了克服计算的稳定性而提出的算法。

实验 4 结论：在收敛速度上，循环中点求积法使用的 CPU 时间要比延拓牛顿法使用的 CPU 时间更多。但是循环中点求积法使用的迭代次数会更少，因为他在每次迭代过程中利用中点求积公式更新点的位置更好，但是会损失一定的 CPU 时间。其次，对于离真解 x*较远的点，循环中点求积法的中点求积步数 N 明显多

于延拓牛顿法的中点求积步数 N，也往往更难求出最好的步数 N。