E08a 编程作业解答

姓名：任云玮　　学号：516030910586

1　问题 1 的解答

求解下列方程的实根

$$x^2 - 3x + 2 - e^x = 0 \tag{1}$$

$$x^3 + 2x^2 + 10x - 20 = 0 \tag{2}$$

1.1　设计不动点迭代

对方程 (1) 和 (2), 分别设计不动点迭代，即给出相应的 $\varphi(x)$，并说明其收敛性.

(1)

$$\varphi(x) = \frac{1}{3}\left(x^2 + 2 - e^x\right).$$

令 $f(x) = x^2 - 3x + 2 - e^2$，$f(0.25) \approx -0.028 > 0$，$f(0.3) \approx -0.16 < 0$，所以零点 x_* 在 $[0.25, 0.3]$ 中. 而在该区间上 $\varphi'(x) = (2x - e^x)/3 \in (-0.3, -0.2)$. 所以一阶收敛. ■

(2)

$$\varphi(x) = -\frac{1}{20}(x^3 + 2x^2 - 20x - 20).$$

令 $f(x) = x^3 + 2x^2 + 10x - 20$，$f(1) = -7 < 0$，$f(1.5) = 2.875 > 0$，所以零点 x_* 在 $[1, 1.5]$ 中. 而在该区间上 $\varphi'(x) = -(3x^2 + 4x - 20)/20 \in (0.3, 0.7)$. 所以一阶收敛. ■

1.2　Steffensen 加速迭代

1.2.1　对之前设计的不动点迭代格式，编写直接迭代的程序，命名为 direct_iteration.m，迭代停止准则为 $|x_k - x_{k-1}| < 10^{-8}$；

```
eps = 10^(-8);

phi = @(x)((x.^2 + 2 - exp(x)) ./ 3);
x0  = 0.25;

```

```
direct_iteration(phi, x0, eps)
% ans = 0.2575

phi = @(x)(-(x.^3 + 2*x^2 - 10*x - 20) ./ 20);
x0  = 1;

direct_iteration(phi, x0, eps)
% ans = 1.3688
```

```
function [res, cnt] = direct_iteration(phi, x0, eps)
    res = phi(x0);
    cnt = 1;
    while abs(res - x0) >= eps
        x0 = res;
        res = phi(x0);
        cnt = cnt + 1;
    end
end
```

1.2.2 编写 Steffensen 加速程序，命名为 Steffensen.m，在相同条件下重复方程（1）的计算：给出不同初值下，加速前和加速后的迭代步数（列出表格）.

```
eps = 10^(-8);
phi = @(x)((x.^2 + 2 - exp(x)) ./ 3);
N = 11;
x0s = linspace(0, 1, N)';
cnti = zeros(1, N)';
for i = 1:N
    [~, cnti(i)] = direct_iteration(phi, x0s(i), eps);
end

cnts = zeros(1, N)';
for i = 1:N
    [~, cnts(i)] = Steffensen(phi, x0s(i), eps);
end
```

```
function [res, cnt] = Steffensen(phi, x0, eps)
    cnt = 1;
    psi = @(x)( x - (phi(x) - x).^2 ./ (phi(phi(x)) - 2 .* phi(x) + x));
    res = psi(x0);
    while abs(res - x0) >= eps
        x0 = res;
        res = psi(res);
        cnt = cnt + 1;
    end
end
```

表 1: 迭代步数比较

x_0	direct iteration	Steffensen
0	14	4
0.1	14	4
0.2	13	3
0.3	13	3
0.4	14	3
0.5	14	4
0.6	14	4
0.7	15	4
0.8	15	4
0.9	15	4
1	15	4

1.3 Newton 迭代

1.3.1 针对方程（2），讨论其实根的范围，并判断其是单根还是重根；

解 对于 (2) 的 $f(x) = \text{L.H.S.}$ 由于 $f(1) = -7 < 0$，$f(2) = 16 > 0$. 所以在 $(1, 2)$ 上有一实根. 同时有

$$f'(x) = 3x^2 + 4x + 10$$

其判别式 $\Delta < 0$，所以 $f'(x) > 0$ 恒成立. 从而仅有一实根，且为单根.

1.3.2 编写基于课本 P227 式 (4.14) 的 Newton 迭代程序，命名为 Newton_iteration.m，迭代停止准则同前；

```
p   = [1 2 10 -20];
eps = 10^(-8);
x0  = 1;
Newton_iteration(p, x0, eps)
% ans = 1.3688
```

```
function [res, cnt] = Newton_iteration(p, x0, eps)
    f   = @(x)(polyval(p, x));
    df  = @(x)(polyval(polyder(p), x));
    ddf = @(x)(polyval(polyder(polyder(p)), x));
    varphi = @(x)(x - f(x) .* df(x) ./ (df(x).^2 - f(x) .* ddf(x)));
    res = varphi(x0);
    cnt = 1;
    while abs(res - x0) >= eps
        x0 = res;
        res = varphi(x0);
        cnt = cnt + 1;
```

```
    end
end
```

1.3.3 在相同条件下，比较 Steffensen 加速法与 Newton 迭代法的迭代步数.

```
eps = 10^(-8);
p   = [1 2 10 -20];
phi = @(x)(polyval(p, x));
N   = 11;
x0s = linspace(1, 2, N)';
cntn = zeros(1, N)';
for i = 1:N
    [~, cntn(i)] = Newton_iteration(p, x0s(i), eps);
end

cnts = zeros(1, N)';
for i = 1:N
    [~, cnts(i)] = Steffensen(phi, x0s(i), eps);
end
```

表 2: 迭代步数比较

x_0	Newton iteration	Steffensen
1	5	8
1.1	4	7
1.2	4	7
1.3	4	8
1.4	4	5
1.5	4	6
1.6	4	7
1.7	5	9
1.8	5	11
1.9	5	13
2	5	15

1.3.4 选做：若方程有复根，应如何求解？可能的话，求出方程 (2) 的一个复根.

解 由于压缩映射定理对于 $\mathbf{R}^n$ 都是成立的，所以对于用于迭代的映射 φ，只需要在不动点 x_* 附近满足 $\nabla\varphi(x) \leq l < 1$，即可保证局部收敛. 从而只需要选择恰当的 φ 和复平面上的恰当初始点 x_0，即可求得一个复根. 对于方程 (2)，取初始点 $x_0 = -1 + 3i$，得复根 $x_* \approx -1.6844 + 3.4313i$.

2　问题 2 的解答

多项式求根是一个病态问题，考虑多项式

$$p(x) = (x-1)(x-2)\cdots(x-10) = a_0 + a_1 x + \cdots + a_9 x^9 + x^{10}$$

求解扰动方程 $p(x) + \varepsilon x^9 = 0$.

2.1　病态性分析

任意给定系数，对 $\varepsilon = 10^{-6}, 10^{-8}, 10^{-10}$，用 MATLAB 求根函数计算扰动方程的根，分析 ε 对根的影响（注意 MATLAB 多项式的存储方式）.

解　代码与求解结果如下.

```
p = [1, -55, 1320, -18150, 157773, -902055, 3416930, ...
    -8409500, 12753576, -10628640, 3628800];
eps = [10^(-6), 10^(-8), 10^(-10)];
rts = zeros(3, 10)';

for i = 1:3
    pret = zeros(1, 11);
    pret(2) = eps(i);
    rts(1:end, i) = roots(p + pret)';
end
```

表 3: 扰动方程的根

ε	10^{-6}	10^{-8}	10^{-10}
x_{10}	9.99722948588222	9.99997244092440	9.99999972451783
x_9	9.00954408664740	9.00009608123229	9.00000096030286
x_8	7.98668735225904	7.99986684514984	7.99999866983353
x_7	7.00940116158683	7.00009342012864	7.00000093241953
x_6	5.99651655390372	5.99996500763840	5.99999965125337
x_5	5.00067909066280	5.00000678245187	5.00000006736228
x_4	3.99993932906276	3.99999939306495	3.99999999402577
x_3	3.00000195269050	3.00000001953720	3.00000000018562
x_2	1.99999998730198	1.99999999987237	1.99999999999926
x_1	1.00000000000276	1.00000000000006	0.999999999999987

对于每一个零点 $r_i = i$，相对于 $a_9 = -55$ 的相对条件数成立公式

$$\kappa_i = \frac{|a_9 \times r_i^8|}{|p'(r_i)|}.$$

所以每个零点对应的条件数为，

$$\kappa_1 = 1.51 \times 10^{-4}, \quad \kappa_2 = 0.3492, \quad \ldots, \quad \kappa_9 = 9.15 \times 10^4, \quad \kappa_{10} = 1.51 \times 10^4.$$

首先在原多项式的零点处，$p'(x)$ 都没有十分小或十分大，同时 $a_9 = -55$ 的大小也可以接收. 所以条件数的大小主要由 r_i^8 决定. 当 r_i 较小，为 1 或 2 时，条件数较小，由系数的扰动而产生的误差较小. 而当 r_i 较大时，由于有 r_i^8 的存在，条件数十分大，从而有微小扰动产生的误差亦会十分大.

2.2 解决方案思考

选做：可能的话提出一个稳定化求解的策略（可以查阅文献）.

解 设多项式被表示为

$$p(x) = \sum_{i=0}^{n} b_i p_i(x).$$

其中 $\{p_i\}$ 是一组基. 则多项式零点 r 相对于系数 b_i 的相对条件数成立公式

$$\kappa = \frac{|b_i p_i(r)|}{|rp'(r)|}.$$

由于 $p'(x)$ 是给定的，无法修改，所以考虑选取恰当的基使得在零点处的 $p_i(r)$ 的取值较小即可. 从而可以选取 Chebyshev 多项式作为基，然后测量对应的系数即可.[1]

[1] 参考文献：Tsai, Edison, ”A Method for Reducing Ill-Conditioning of Polynomial Root Finding Using a Change of Basis” (2014). University Honors Theses. Paper 109.