《工程硕士数学》第三次计算实习

> 软硕232 丁浩宸 2023213911

第一题

理论基础

- 不动点迭代法
- 不动点迭代法的Steffensen加速法
- Newton迭代法

算法描述

- 对于不动点迭代法$x=\varphi(x)$，Steffensen加速法的算法为：$\begin{cases} m=\varphi(x_k) \\ n=\varphi(m) \\ x_{k+1}=x_k-\frac{(m-n)^2}{n-2m+x_k} \end{cases}$。
- Newton迭代法的算法为：$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$

计算程序

注意到方法2很可能是不收敛的，因此如果迭代超过1万次则强行停止。

```
format long
metric = 1.368808107;

% method 1
i = 0;
x = 1;
while (true)
    xnext = 20 / (x ^ 2 + 2 * x + 10);
    x = xnext;
    i = i + 1;
    if (abs(x - metric) < 1e-9)
        break
    end
end
i

% method 2
i = 0;
x = 1;
while (true)
    xnext = (20 - 2 * x ^ 2 - x ^ 3) / 10;
    x = xnext;
    i = i + 1;
    if (abs(x - metric) < 1e-9)
        break
    end
    if (i > 10000)
        break
```

```
    end
end
i
if (i > 10000)
    disp("gasp... it's still not converging...")
    disp("i need some break...")
end

% steffensen 1
i = 0;
x = 1;
while (true)
    m = 20 / (x ^ 2 + 2 * x + 10);
    n = 20 / (m ^ 2 + 2 * m + 10);
    xnext = x - (m - n) ^ 2 / (n - 2 * m + x);
    x = xnext;
    i = i + 1;
    if (abs(x - metric) < 1e-9)
        break
    end
end
i

% steffensen 2
i = 0;
x = 1;
while (true)
    m = (20 - 2 * x ^ 2 - x ^ 3) / 10;
    n = (20 - 2 * m ^ 2 - m ^ 3) / 10;
    xnext = x - (m - n) ^ 2 / (n - 2 * m + x);
    x = xnext;
    i = i + 1;
    if (abs(x - metric) < 1e-9)
        break
    end
    if (i > 10000)
        break
    end
end
i

% newton
i = 0;
x = 1;
while (true)
    xnext = x - (x ^ 3 + 2 * x ^ 2 + 10 * x - 20) / (3 * x ^ 2 + 4 * x + 10);
    x = xnext;
    i = i + 1;
    if (abs(x - metric) < 1e-9)
        break
    end
end
i
```

计算结果分析

方法	收敛到目标结果所用的迭代次数
不动点迭代法1	24
不动点迭代法2	无法收敛
不动点迭代法1的Steffensen加速	3
不动点迭代法2的Steffensen加速	4
Newton迭代法	4

由此证明了课本中的如下结论：

- “Steffensen加速法不但可以提高速度，有时也能把不收敛的方法改进为二阶收敛的方法。”（P105）
- “（Newton迭代法）至少是二阶收敛”。（P107）

第二题

理论基础

- 压缩映射定理
- 非线性方程组的Newton迭代法

算法分析

- 压缩映射定理：函数Φ定义在区域$D \subset \mathbb{R}^n$，假设：
    - 存在闭集$D_0 \subset D$，实数$L \in (0,1)$使$||\Phi(x)-\Phi(y)|| \le L|y-x||$，$\forall x,y \in D_0$
    - $\Phi(x) \in D_0$，$\forall x \in D_0$

  则Φ在D_0有唯一的不动点x^*
- 非线性方程组的Newton迭代法：$\Phi(x) = x-(F'(x))^{-1}F(x)$，式中$F'(x)$为导数矩阵$\begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \cdots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix}$。

计算代码

第一问

```
% b.m
function result = b(X)
    x1 = X(1);
    x2 = X(2);
    x3 = X(3);
```

```
    result =  [cos(x2 * x3 + 1 / 2) / 3; sqrt(x1 ^ 2 + sin(x3) + 1.06) / 9 -
0.1; (1 - 10 * pi / 3 - exp(-x1 * x2)) / 20];
end

% 脚本
format long

X = [0; 0; 0]
Y = b(X)

i = 1;
x = 1;
while (true)
    Y = b(X)
    i = i + 1;
    if (abs(Y(1) - X(1)) < 1e-9)
        if (abs(Y(2) - X(2)) < 1e-9)
            if (abs(Y(3) - X(3)) < 1e-9)
                break
            end
        end
    end

    if (i > 10000)
        break
    end

    X = Y;
end
i
```

第二问

```
format long

syms x1 x2 x3;

F = [3 * x1 - cos(x2 * x3) - 1 / 2;
    x1 ^ 2 - 81 * (x2 + 0.1) ^ 2 + sin(x3) + 1.06;
    exp(-x1 * x2) + 20 * x3 + (10 * pi - 3) / 3];

FF = inv([diff(F(1), x1), diff(F(1), x2), diff(F(1), x3);
    diff(F(2), x1), diff(F(2), x2), diff(F(2), x3);
    diff(F(3), x1), diff(F(3), x2), diff(F(3), x3)]);

X = [0.2; 0; -0.5];
i = 0
while (true)
    f = subs(F, [x1, x2, x3], X');
    ff = subs(FF, [x1, x2, x3], X');
    i = i + 1
    Y = vpa(X - ff *  f)
    if (abs(Y(1) - X(1)) < 1e-9)
        if (abs(Y(2) - X(2)) < 1e-9)
            if (abs(Y(3) - X(3)) < 1e-9)
                break
            end
        end
    end

    if (i > 10000)
        break
```

```
    end

    X = Y;
end
i
```

计算结果分析

第一问

没有什么头绪，强行移项得到$\begin{cases} x_1 = \frac{\cos(x_2x_3+\frac{1}{2})}{3} \\ x_2 = \frac{\sqrt{x_1^2+\sin x_3+1.06}}{9} - 0.1 \\ x_3 = \frac{1-\frac{10\pi}{3}-e^{-x_1x_2}}{20} \end{cases}$。以$10^{-9}$为精度，发现这个方法能够在$i=8$收敛，说明这个方法确实满足压缩映射定理（映内性易证，但并不清楚如何证明压缩性质）。

方程组的解为$\begin{cases} x_1 = 0.50000 \\ x_2 = 0.00000 \\ x_3 = -0.52360 \end{cases}$。

第二问

初值	解	收敛到解的迭代次数
$X=\begin{bmatrix}0.5\\0\\-0.5236\end{bmatrix}$	$X=\begin{bmatrix}0.50000\\0.00000\\-0.52360\end{bmatrix}$	1
$X=\begin{bmatrix}0.5\\0\\-0.5\end{bmatrix}$	$X=\begin{bmatrix}0.50000\\0.00000\\-0.52360\end{bmatrix}$	2
$X=\begin{bmatrix}0.4\\0\\-0.5\end{bmatrix}$	$X=\begin{bmatrix}0.50000\\0.00000\\-0.52360\end{bmatrix}$	3
$X=\begin{bmatrix}0.2\\0\\-0.5\end{bmatrix}$	$X=\begin{bmatrix}0.50000\\0.00000\\-0.52360\end{bmatrix}$	4
$X=\begin{bmatrix}0\\0\\0\end{bmatrix}$	$X=\begin{bmatrix}0.50000\\0.00000\\-0.52360\end{bmatrix}$	5
$X=\begin{bmatrix}-0.5\\0.5\\1\end{bmatrix}$	$X=\begin{bmatrix}0.50000\\0.00000\\-0.52360\end{bmatrix}$	6
$X=\begin{bmatrix}-1\\1\\1\end{bmatrix}$、$X=\begin{bmatrix}-1\\1\\1\end{bmatrix}$	$X=\begin{bmatrix}0.50000\\0.00000\\-0.52360\end{bmatrix}$	7

由于$X=\begin{bmatrix}-1\\1\\1\end{bmatrix}$、$X=\begin{bmatrix}-1\\-1\\1\end{bmatrix}$已经是定义域内距离解最远的向量，因此我们认为Newton迭代法解本方程组最多需要7次迭代。这说明Newton迭代法优于我们在第一问中自行提出的迭代法。