Homework 2016-03-04

Chuan Lu
13300180056

March 4, 2016

Problem 1.

$$Given \quad G(x) = e^{-x} \quad and \quad F(x) = x - e^{-x},$$

compare Fixed-point iteration and Newton-Raphson iteration.

Proof. **0.1 The code is shown as follows.**

```
function[root, root_list] = fixed_point_iter(func, x0, tol, order)
% FIXED_POINT_ITER Extract root ROOT of function FUNC with init
% point X0 given, using fixed point iteration
% author: chuanlu
% 2016-03-04

if nargin < 3
    error('More arguments are needed --newton_raphson_iter');
elseif nargin == 3
    tol = 1e-6;
    order = 100;
end

count = 0;
root_list = zeros(1, 1);
while 1
    count = count + 1;
    root_list(count) = x0;
    x1 = feval(func, x0);
    if abs(x1 - x0) < tol || count >= order
        root = x1;
        return
    elseif count >= 100
        warning('Count over 100, may not be convergence');
        root = x1;
```

```
            return
        end
        x0 = x1;
end
```

```
function[root, root_list] = newton_raphson_iter(func1, func2, x0, tol, order)
% Extract root ROOT of function FUNC1 with its derivative FUNC2
% and init point X0 given

if nargin < 3
    error('More arguments are needed --newton_raphson_iter');
elseif nargin == 3
    tol = 1e-6;
    order = 100;
elseif nargin == 4
    order = 100;
end

count = 0;
root_list = zeros(1, 1);
while 1
    count = count + 1;
    root_list(count) = x0;
    f1 = feval(func1, x0);
    f2 = feval(func2, x0);
    x1 = x0 - f1 / f2;
    if abs(x1 - x0) < tol
        disp(count);
        root = x1;
        return
    elseif count >= order
        root = x1;
        return
    elseif count > 100
        warning('Count over 100, may not be convergence');
        root = x1;
        return
    end
    x0 =x1;
end
```

```
% homework 1.2.1
format long

f = @(x)(x - exp(-x));
```

```
f2 = @(x)(1 + exp(-x));
g = @(x)(exp(-x));

x0 = 1;
tol = 1e-16;
n1 = 4;
n2 = 24;
[root1, root_list1] = fixed_point_iter(g, x0, tol, n2);
[root2, root_list2] = newton_raphson_iter(f, f2, x0, tol, n1);
disp('root1')
disp(root1)
disp('root_list1')
disp(root_list1)
disp('root2')
disp(root2)
disp('root_list2')
disp(root_list2)
```

0.2 The result is shown as follows.

Fixed-point iter	Newton-Raphson iter
1.000000000000000	1.000000000000000
0.367879441171442	0.537882842739990
0.692200627555346	0.566986991405413
0.500473500563637	0.567143285989123
0.606243535085597	
0.545395785975027	
0.579612335503379	
0.560115461361089	
0.571143115080177	
0.564879347391050	
0.568428725029061	
0.566414733146883	
0.567556637328283	
0.566908911921495	
0.567276232175570	
0.567067898390788	
0.567186050099357	
0.567119040057215	
0.567157044001298	
0.567135490206278	
0.567147714260119	
0.567140781458298	
0.567144713346570	
0.567142483401307	
Root	Root
0.567143748099411	0.567143290409784

0.3 Analysis

When using Newton-Raphson iteration to extract the root of an equation, according to (1.2.22), the error is of second order, while the error of fixed-point iteration is of first-order. Hence, the speed of convergence is much quicker for Newton-Raphson iteration.

□

Problem 2.
For a n-dimension vector $\mathbf{x}$, when

$$1 \leqslant p \leqslant q,$$

prove

$$\|\mathbf{x}\|_q \leqslant \|\mathbf{x}\|_p \leqslant n^{\frac{1}{p}-\frac{1}{q}} \|\mathbf{x}\|_q$$

Proof. On one hand, assume $\|\mathbf{x}\|_q = 1$, then

$$\|x_i\| \leqslant 1, \quad for \quad 1 \leqslant i \leqslant n$$

Because $1 \leqslant p \leqslant q$, then $\|x_i\|^q \leqslant \|x_i\|^p$. Hence, $(\sum_{i=1}^n \|x_i\|^p)^{\frac{1}{p}} \geqslant (\sum_{i=1}^n \|x_i\|^q)^{\frac{1}{q}}$, which means $\|x\|_q \leqslant \|x\|_p$.
On the other hand, according to *Hölder* inequation,

$$\frac{(\sum_{i=1}^n \|x_i\|^p)^{\frac{1}{p}}}{n^{\frac{1}{p}}} \leqslant \frac{(\sum_{i=1}^n \|x_i\|^q)^{\frac{1}{q}}}{n^{\frac{1}{q}}},$$

hence

$$\|\mathbf{x}\|_p \leqslant n^{\frac{1}{p}-\frac{1}{q}} \|\mathbf{x}\|_q$$

$\square$