CSC336 A4

Kaiqu Liang

December 5, 2018

Problem 1

(a)

MatLab Program:

```
function question1a
n = [0, 1, 2, 3, 4, 5];
x = 1;
y = x - sqrt(2);
fprintf("n%14s%25s\n", "x(n)", "x(n) - sqrt(2)");
for i = 0:5
    fprintf("%d%20.15f%20.15f\n", i, x, y);
    x = x - (x^2 - 2) / (2 * x);
    y = x - sqrt(2);
end
```

Program Output:

```
>> question1a
n           x(n)          x(n) - sqrt(2)
0    1.000000000000000   -0.414213562373095
1    1.500000000000000    0.085786437626905
2    1.416666666666667    0.002453104293572
3    1.414215686274510    0.000002123901415
4    1.414213562374690    0.000000000001595
5    1.414213562373095    0.000000000000000
```

(b)

MatLab Program:

```
function question1b
x_n = 1;
x_m = 2;
y = x_n - sqrt(2);
f = @(x) x^2 - 2;
fprintf("n%14s%25s\n", "x(n)", "x(n) - sqrt(2)");
fprintf("%d%20.15f%20.15f\n", 0, x_n, y);
fprintf("%d%20.15f%20.15f\n", 1, x_m, x_m - sqrt(2));
for n = 2 : 7
    new_x = x_m - f(x_m) * (x_m - x_n) / (f(x_m) - f(x_n));
    y = new_x - sqrt(2);
    fprintf("%d%20.15f%20.15f\n", n, new_x, y);
    x_n = x_m;
    x_m = new_x;
end
```

Program Output:

```
>> question1b
n           x(n)           x(n) - sqrt(2)
0    1.000000000000000   -0.414213562373095
1    2.000000000000000    0.585786437626905
2    1.333333333333333   -0.080880229039762
3    1.400000000000000   -0.014213562373095
4    1.414634146341463    0.000420583968368
5    1.414211438474870   -0.000002123898225
6    1.414213562057320   -0.000000000315775
7    1.414213562373095    0.000000000000000
```

Problem 2

(a)

(1)
$$g_1(x) = \frac{x^2+2}{3}$$
$$g_1'(x) = \frac{2}{3}x$$
$$|g_1'(2)| = \left|\frac{2}{3} \times 2\right| = \left|\frac{4}{3}\right| > 1$$
Hence, fixed-point iteration diverges

(2)
$$g_2(x) = \sqrt{3x-2}$$
$$g_2'(x) = \frac{3}{2} \cdot \frac{1}{\sqrt{3x-2}}$$
$$|g_2'(2)| = \left|\frac{3}{2} \cdot \frac{1}{\sqrt{3 \times 2 - 2}}\right| = \frac{3}{4} < 1$$
Hence, fixed-point iteration converges linearly.

(3)
$$g_3(x) = 3 - \frac{2}{x}$$
$$g_3'(x) = \frac{2}{x^2}$$
$$|g_3'(2)| = \frac{2}{2^2} = \frac{1}{2} < 1$$
Hence, fixed-point iteration converges linearly.

(4)
$$g_4(x) = \frac{x^2-2}{2x-3}$$
$$g_4'(x) = \frac{2x \cdot (2x-3) - 2(x^2-2)}{(2x-3)^2} = \frac{2x^2-6x+4}{(2x-3)^2}$$
$$|g_4'(2)| = \left|\frac{2 \times 2^2 - 6 \times 2 + 4}{(2^2-3)^2}\right| = |8-12+4| = 0$$
$$g_4''(x) = \frac{(4x-6) \cdot (2x-3)^2 - (2x^2-6x+4) \cdot 4(2x-3)}{(2x-3)^4}$$
$$g_4''(2) = 2 \times 1^2 - 0 = 2 \neq 0$$
Hence, fixed-point iteration converges quadratically.

(b)
MatLab Program:

```
function question2b
g1 = @(x) (x^2 + 2) / 3;
g2 = @(x) sqrt(3 * x - 2);
g3 = @(x) 3 - 2 / x;
g4 = @(x) (x ^ 2 - 2) / (2 * x - 3);
fprintf("Table for g1(x): \n");
verify_linear_convergence(g1);
fprintf("Table for g2(x): \n");
verify_linear_convergence(g2);
fprintf("Table for g3(x): \n");
verify_linear_convergence(g3);
fprintf("Table for g4(x): \n");
verify_quadratic_convergence(g4);
end

function verify_linear_convergence(g)
x = 2.3;
error = 2 - x;
fprintf("k%17s%32s\n", "e(k)", "|e(k + 1)| / |e(k)|");
for k = 0: 10
    x = g(x);
    new_error = 2 - x;
    ratio = abs(new_error) / abs(error);
    if k ~= 10
        fprintf("%d%24.10d%24.10d\n", k, error, ratio);
    else
        fprintf("%d%23.10d%24.10d\n", k, error, ratio);
    end
    error = new_error;
end
fprintf("\n");
end

function verify_quadratic_convergence(g)
x = 2.3;
error = 2 - x;
fprintf("k%17s%32s%26s\n", "e(k)", "|e(k + 1)| / |e(k)|", "|e(k + 1)| / |e(k)|^2");
for k = 0: 10
    x = g(x);
    new_error = 2 - x;
    ratio1 = abs(new_error) / abs(error);
    ratio2 = abs(new_error) / (abs(error))^2;
    if k ~= 10
        fprintf("%d%24.10d%24.10d%24.10d\n", k, error, ratio1, ratio2);
    else
        fprintf("%d%23.10d%24.10d%24.10d\n", k, error, ratio1, ratio2);
    end
    error = new_error;
end
fprintf("\n");
end
```

Program Output:

```
>> question2b
Table for g1(x):
k            e(k)              |e(k + 1)| / |e(k)|
0      -3.0000000000e-01        1.4333333333e+00
1      -4.3000000000e-01        1.4766666667e+00
2      -6.3496666667e-01        1.5449888889e+00
3      -9.8101644481e-01        1.6603388149e+00
4      -1.6288196814e+00        1.8762732271e+00
5      -3.0561107601e+00        2.3520369200e+00
6      -7.1880853394e+00        3.7293617798e+00
7      -2.6806970735e+01        1.0268990245e+01
8      -2.7528052097e+02        9.3093506990e+01
9      -2.5626829103e+04        8.5436097011e+03
10     -2.1894562574e+08        7.2981876579e+07

Table for g2(x):
k            e(k)              |e(k + 1)| / |e(k)|
0      -3.0000000000e-01        7.1198120706e-01
1      -2.1359436212e-01        7.2215241551e-01
2      -1.5424768454e-01        7.2947963836e-01
3      -1.1252054514e-01        7.3481120312e-01
4      -8.2681357148e-02        7.3872004486e-01
5      -6.1078375861e-02        7.4160212474e-01
6      -4.5295853314e-02        7.4373622251e-01
7      -3.3688166839e-02        7.4532152072e-01
8      -2.5108515739e-02        7.4650197450e-01
9      -1.8743556576e-02        7.4738255285e-01
10     -1.4008607163e-02        7.4804032081e-01

Table for g3(x):
k            e(k)              |e(k + 1)| / |e(k)|
0      -3.0000000000e-01        4.3478260870e-01
1      -1.3043478261e-01        4.6938775510e-01
2      -6.1224489796e-02        4.8514851485e-01
3      -2.9702970297e-02        4.9268292683e-01
4      -1.4634146341e-02        4.9636803874e-01
5      -7.2639225182e-03        4.9819059107e-01
6      -3.6188178528e-03        4.9909692956e-01
7      -1.8061408790e-03        4.9954887218e-01
8      -9.0225563910e-04        4.9977453780e-01
9      -4.5092439501e-04        4.9988729431e-01
10     -2.2541137576e-04        4.9994365351e-01

Table for g4(x):
k            e(k)              |e(k + 1)| / |e(k)|      |e(k + 1)| / |e(k)|^2
0      -3.0000000000e-01        1.8750000000e-01         6.2500000000e-01
1      -5.6250000000e-02        5.0561797753e-02         8.9887640449e-01
2      -2.8441011236e-03        2.8280148035e-03         9.9434397042e-01
3      -8.0431600802e-06        8.0430282044e-06         9.9998360398e-01
4      -6.4691363377e-11              0000000000               0000000000
5             0000000000                     NaN                      NaN
6             0000000000                     NaN                      NaN
7             0000000000                     NaN                      NaN
8             0000000000                     NaN                      NaN
9             0000000000                     NaN                      NaN
10            0000000000                     NaN                      NaN
```

Explanation: By the table for $g_1(x)$,we can see that when $k > 6$, $\frac{|e_{k+1}|}{|e_k|}$ increases very fast as k grows and the rate of increase looks strange. Therefore the fixed-point iteration is very likely to diverge.
By the table for $g_2(x)$, we can see that as k increases, $\frac{|e_{k+1}|}{|e_k|}$ is relatively stable and the ratio becomes closer to 0.75. Therefore the fixed-point iteration is very likely to converge linearly with constant C = 0.75.
By the table for $g_3(x)$, we can see that as k increases, $\frac{|e_{k+1}|}{|e_k|}$ is relatively stable and the ratio becomes closer to 0.5. Therefore the fixed-point iteration is very likely to converge linearly with constant C = 0.5.
By the table for $g_4(x)$, we can see that as k increases, $\frac{|e_{k+1}|}{|e_k|^2}$ is relatively stable and the ratio becomes closer to 1. Therefore the fixed-point iteration is very likely to converge quadratically.

Problem 3

(a)

MatLab Program:

```
function question3
R = 0.082054;
a = 3.592;
b = 0.04267;
T = 300;
fprintf("%3s%23s%23s\n", "p", "v Waals", "v ideal gas law");
for i = 0:2
    p = 10^i;
    f = @(v) (p + a / v^2) * (v - b) - R * T;
    v_ideal = R * T / p;
    v = fzero(f, v_ideal);
    fprintf("%3.d%23.15f%23.15f\n", p, v, v_ideal);
end
```

Program output:

```
>> question3
  p                v Waals        v ideal gas law
  1      24.512588128441500     24.616199999999999
 10       2.354495580702039      2.461620000000000
100       0.079510827813453      0.246162000000000
```

Problem 4

(a)

We know $f(x) = \frac{1}{x} - b$, therefore $f'(x) = -\frac{1}{x^2}$
If I apply Newton's method to (3), we have

$$\begin{aligned} r_1 &= r_0 - \frac{f(r_0)}{f'(r_0)} \\ &= r_0 - \frac{\frac{1}{r_0} - b}{-\frac{1}{r_0^2}} && (\text{Since } f(r_0) = \frac{1}{r_0},\ f'(r_0) = -\frac{1}{r_0^2}) \\ &= r_0 + r_0 - br_0^2 \\ &= 2r_0 - br_0^2 \end{aligned}$$

(b)

$$\begin{aligned} \frac{r - r_1}{r} &= \frac{r - (2r_0 - br_0^2)}{r} \\ &= \frac{r^2 - 2r_0 r + br_0^2 \cdot r}{r^2} \\ &= \frac{r^2 - 2r_0 r + r_0^2}{r^2} && (\text{Since } r = \frac{1}{b}) \\ &= \left(\frac{r - r_0}{r}\right)^2 \end{aligned}$$

(c)

We know $\frac{r - r_1}{r}$ is the relative error of r_1 in approximating r and $\frac{r-r_0}{r}$ is the relative error of r_0 in approximating r.
Let's denote $Relative(r_1) := \frac{r-r_1}{r}, \quad Relative(r_0) := \frac{r-r_0}{r}$
We know the correct digits of r_1 is about $\log_{10} Relative(r_1)$ and the correct digit of r_0 is about $\log_{10} Relative(r_0)$.
By (4), we know $Relative(r_1) = [Relative(r_0)]^2$
Therefore $\log_{10} Relative(r_1) = 2 \log_{10} Relative(r_0)$
It follows that r_1 has roughly twice as many correct digits as r_0 has.