CSC336 A4 Report

Tianshu Zhu
1002111225
Utorid: zhutians

Q1

Code:

```
function q1()
    find_root('newton', 5, 1);
    find_root('secant', 7, [1 2]);
end

function find_root(method, iters, x)
    % a): find root using newton's method
    if strcmp(method, 'newton')
        xns = zeros(iters+1, 1); xns(1) = x;
        fprintf('n%20s%20s\n','x(n)','x(n)-sqrt(2)');
        fprintf('%d%20.16f%20.16f\n',0,xns(1),xns(1)-sqrt(2));
        for n = 1:iters
            xnm1 = xns(n);
            xn = xnm1-f(xnm1)/(2*xnm1);
            xns(n+1) = xn;
            fprintf('%d%20.16f%20.16f\n',n,xn,xn-sqrt(2));
        end
        fprintf('\n');
    % b): find root using secant method
    elseif strcmp(method, 'secant')
        xns = zeros(iters+1, 1); xns(1) = x(1); xns(2) = x(2);
        fprintf('n%20s%20s\n','x(n)','x(n)-sqrt(2)');
        fprintf('%d%20.16f%20.16f\n',0,xns(1),xns(1)-sqrt(2));
        fprintf('%d%20.16f%20.16f\n',1,xns(2),xns(2)-sqrt(2));
        for n = 2:iters
            xnm1 = xns(n);
            xnm2 = xns(n-1);
            xn = xnm1-f(xnm1)*(xnm1-xnm2)/(f(xnm1)-f(xnm2));
            xns(n+1) = xn;
            fprintf('%d%20.16f%20.16f\n',n,xn,xn-sqrt(2));
        end
        fprintf('\n');
    end
end

function y = f(x)
    y = x^2-2;
end
```

Results:

a): Newton's method

```
n               x(n)        x(n)-sqrt(2)
0  1.0000000000000000 -0.4142135623730951
1  1.5000000000000000  0.0857864376269049
2  1.4166666666666667  0.0024531042935716
3  1.4142156862745099  0.0000021239014147
4  1.4142135623746899  0.0000000000015947
5  1.4142135623730951  0.0000000000000000
```

b): Secant method

```
n               x(n)        x(n)-sqrt(2)
0  1.0000000000000000 -0.4142135623730951
1  2.0000000000000000  0.5857864376269049
2  1.3333333333333335 -0.0808802290397617
3  1.4000000000000001 -0.0142135623730950
4  1.4146341463414633  0.0004205839683682
5  1.4142114384748701 -0.0000021238982251
6  1.4142135620573204 -0.0000000003157747
7  1.4142135623730954  0.0000000000000002
```

Q2

a):

$g_1(x) = (x^2+2)/3$
$g_1'(x) = 2x/3$
$g_1'(2) = 4/3 > 1.$
May not converge

$g_2(x) = \sqrt{3x-2}$
$g_2'(x) = 3/2\sqrt{3x-2}$
$g_2'(2) = 3/4 < 1$
$\lim_{k \to \infty} |x_{k+1} - x^*| / |x_k - x^*| \simeq |g_2'(2)| = 3/4$

linear Convergence

$g_3(x) = 3 - 2/x$
$g_3'(x) = 2/x^2$
$g_3'(2) = 1/2 < 3/4$
$\lim_{k \to \infty} |x_{k+1} - x^*| / |x_k - x^*| \simeq |g_2'(2)| = 1/2$

linear convergence, farster than g_2

$g_4(x) = (x^2-2)/(2x-3)$
$g_4'(x) = \dfrac{2x(2x-3) - (x^2-2)2}{(2x-3)^2} = \dfrac{2x^2 - 6x + 4}{(2x-3)^2}$
$g_4'(2) = 0$
$g_4''(x) = \dfrac{(4x-6)(2x-3)^2 - (2x^2-6x+4)(4x-6)(2)}{(2x-3)^4}$
$g_4''(2) = 2$
$\lim_{k \to \infty} |x_{k+1} - x^*| / |x_k - x^*| \simeq |g_2''(2)/2| = 1$

Quadratic Convergence

b):

Code:

```
function q2()
    find_root('g1', 10, 2.5);
    find_root('g2', 10, 2.5);
    find_root('g3', 10, 2.5);
    find_root('g4', 10, 2.5);
end

function y = g1(x)
    y = (x^2+2)/3;
end

function y = g2(x)
    y = sqrt(3*x-2);
end

function y = g3(x)
    y = 3-2/x;
end
```

```
function y = g4(x)
    y = (x^2-2)/(2*x-3);
end

function r = rate(c, xn, xnm1)
    r = log(abs(xn-2)/c)/log(abs(xnm1-2));
end

function find_root(method, iters, x)
    % get the corresponding c
    switch method
        case 'g1'
            fprintf('g1\n');
            c = 4/3;
        case 'g2'
            fprintf('g2\n');
            c = 3/4;
        case 'g3'
            fprintf('g3\n');
            c = 1/2;
        case 'g4'
            fprintf('g4\n');
            c = 1;
    end
    xns = zeros(iters+1, 1); xns(1) = x;
    fprintf(' n%24s%24s\n','x(n)','x(n)-2');
    fprintf('%2.d%24.16f%24.16f\n',0,xns(1),xns(1)-2);
    r = 0;
    for n = 1:iters
        % compute xn based on different g
        xnm1 = xns(n);
        if strcmp(method, 'g1')
            xn = g1(xnm1);
        elseif strcmp(method, 'g2')
            xn = g2(xnm1);
        elseif strcmp(method, 'g3')
            xn = g3(xnm1);
        elseif strcmp(method, 'g4')
            xn = g4(xnm1);
        end
        xns(n+1) = xn;
        % print n, xn, xn-2
        if xn<100
            fprintf('%2.d%24.16f%24.16f\n',n,xn,xn-2);
        else
            fprintf('%2.d%24.d%24.d\n',n,xn,xn-2);
        end
        % approximate convergence rate
        if xn == 2 && r == 0 % converged
            r = rate(c, xns(n), xns(n-1));
        elseif n == iters && r == 0
            r = rate(c, xns(n+1), xns(n));
        end
    end
    fprintf('approximate convergence rate: %d\n\n', r);
end
```

Result:

```
g1
 n                 x(n)                x(n)-2
        2.5000000000000000      0.5000000000000000
 1      2.7500000000000000      0.7500000000000000
 2      3.1875000000000000      1.1875000000000000
 3      4.0533854166666670      2.0533854166666670
 4      6.1433111120153363      4.1433111120153363
 5     13.2467571396703701     11.2467571396703701
 6     59.1588582391359736     57.1588582391359736
 7                  1e+03                   1e+03
 8                  5e+05                   5e+05
 9                  7e+10                   7e+10
10                  2e+21                   2e+21
diverge

g2
 n                 x(n)                x(n)-2
        2.5000000000000000      0.5000000000000000
 1      2.3452078799117149      0.3452078799117149
 2      2.2440195274852544      0.2440195274852544
 3      2.1753295342213703      0.1753295342213703
 4      2.1274370972285199      0.1274370972285199
 5      2.0933970697613864      0.0933970697613864
 6      2.0688622982896079      0.0688622982896079
 7      2.0509965613985859      0.0509965613985859
 8      2.0378885357633663      0.0378885357633663
 9      2.0282173471524443      0.0282173471524443
10      2.0210522114624681      0.0210522114624681
approximate convergence rate: 1.001471e+00

g3
 n                 x(n)                x(n)-2
        2.5000000000000000      0.5000000000000000
 1      2.2000000000000002      0.2000000000000002
 2      2.0909090909090908      0.0909090909090908
 3      2.0434782608695654      0.0434782608695654
 4      2.0212765957446810      0.0212765957446810
 5      2.0105263157894737      0.0105263157894737
 6      2.0052356020942410      0.0052356020942410
 7      2.0026109660574414      0.0026109660574414
 8      2.0013037809647978      0.0013037809647978
 9      2.0006514657980454      0.0006514657980454
10      2.0003256268316507      0.0003256268316507
approximate convergence rate: 1.000044e+00

g4
 n                 x(n)                x(n)-2
        2.5000000000000000      0.5000000000000000
 1      2.1250000000000000      0.1250000000000000
 2      2.0125000000000002      0.0125000000000002
 3      2.0001524390243901      0.0001524390243901
 4      2.0000000232305739      0.0000000232305739
 5      2.0000000000000009      0.0000000000000009
 6      2.0000000000000000      0.0000000000000000
 7      2.0000000000000000      0.0000000000000000
 8      2.0000000000000000      0.0000000000000000
 9      2.0000000000000000      0.0000000000000000
10      2.0000000000000000      0.0000000000000000
approximate convergence rate: 1.971655e+00
```

Q3

Code:

```
function q3()
    a = 3.592;
    b = 0.04267;
    R = 0.082054;
    T = 300;
    Ps = [1 10 100];
    fprintf('%3s%24s%24s\n', 'P', 'v waals', 'v ideal gas law');
    for i = 1:length(Ps)
        P = Ps(i);
        v0 = ideal_gas_law(P, R, T);
        v = waals(a, b, P, R, T, v0);
        fprintf('%3.d%24.16f%24.16f\n',P,v,v0);
    end
end

% compute v using waals
function v = waals(a, b, P, R, T, v0)
    fun = @(v) (P+a/(v^2))*(v-b)-R*T;
    v = fzero(fun, v0);
end

% compute v using ideal gas law
function v = ideal_gas_law(P, R, T)
    v = R*T/P;
end
```

Result:

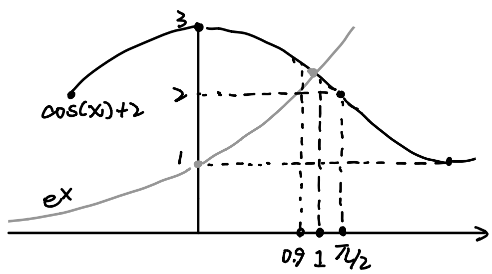
```
  P                 v waals         v ideal gas law
  1      24.5125881284415001     24.6161999999999992
 10       2.3544955807020393      2.4616199999999999
100       0.0795108278134527      0.2461620000000000
```

Q4

a):
$f(x) = 2+\cos(x)-e^x$

Since $\cos(x)$, e^x are continuous over $\mathbb{R}$,
Then f is continuous over $\mathbb{R}$
Since $f(0.9) = 2.62161 - 2.45960 > 0$; $f(1) = 2.54030 - 2.71828 < 0$
Then there is at least one root in $(0.9, 1)$

b):
$f(x)$ has one root



Since $1 \le \cos(x)+2 \le 3$, e^x is strictly increasing
Since $e^0 = 1$, $e^{1.1} = 3.00417 > 3$
Then there is not root in $(-\infty, 0) \cup (1.1, +\infty)$
Since there is only 1 root in $[0, 1.1]$ according to the graph
Then $f(x)$ has only 1 root

c):
Newton's iteration

$x = x - f(x)/f'(x)$
$= x - (2+\cos(x)-e^x)/(-\sin(x)-e^x)$

$x_0 = 0.95$ is a good initial guess. Since root is in $(0.9, 1)$