Nam Nguyen

npn190000

**Exercises**

Section 3.1: 3, 8, 14, 15, 16, 19 (b,c,d)

3. Demonstrate graphically that the equation 50π + sin x = 100 arctan x has infinitely many solutions.

**Answer:**

A picture containing diagram

Description automatically generated

Let y1 = 50π + sin x and y2 = 100 arctan x. In the picture above, we can consider that have infinitely many solutions. Because observe that two equations intersect at infinite many points.

8. If a = 0.1 and b = 1.0, how many steps of the bisection method are needed to determine the root with an error of at most 1 /2 × 10−8?

**Answer:**

. a = 0.1 and b = 1.0

. Determine the root with an error of at most (1 /2) × 10−8

. If f(a)f(b) < 0 then after n step an appropriate root error (b-a)/2n+1 is obtained.

We Have: (b-a)/2n+1 < (1 /2) × 10−8 (with a = 0.1 and b = 1.0)

Then 0.9/2n+1 < (1 /2) × 10−8

Then 0.9/2n < 10−8

Then 2n > 0.9\*108

Then n > log2(0.9\*108)

Then n > 26.43

Hence, after 27 steps then the root will be with an error of at most 0.5\* 10−8

14. Denote the successive intervals that arise in the bisection method by [a0, b0], [a1, b1], [a2, b2], and so on.

a. Show that a0 ≤ a1 ≤ a2 ··· and that b0 ≥ b1 ≥ b2 ···.

b. Show that bn − an = 2−n(b0 − a0).

c. Show that, for all n, anbn + an−1bn−1 = an−1bn + anbn−1.

**Answer**:



. The bisection of [a0, b­0 ]. Let c0 = (a0 + b0)/2 is the middle.

. successive intervals that arise in the bisection method by [a0, b0], [a1, b1], [a2, b2], … [an, bn]

. So two possible interval [a1, c0] or [c0, b­1].

If f(b0)f(c0) < 0 then [c0, b­1], then a1  = c0 and b1 = b0.

If f(a0)\*f(c0) < 0 then [a1, c0], then b1  = c0 and a1 = a0.

**a.** Show that a0 ≤ a1 ≤ a2 ··· and that b0 ≥ b1 ≥ b2 ···.

By taking [a1, c0] = [a1, b1], That mean a0 ≤ a1 and b0 ≥ c0 = b1.

Similarly we have: a1 ≤ a2 and b1 ≥ b2.

So {a0,a1, .. an} will be the monotonic increasing sequence.

So {b0,b1, .. bn} will be the monotonic decreasing sequence.

Hence: Show that a0 ≤ a1 ≤ a2 ··· and that b0 ≥ b1 ≥ b2 ···.

b. Show that bn − an = 2−n(b0 − a0).

If f(b0)f(c0) < 0 then [c0, b­1], then a1  = c0 and b1 = b0.

b1 – a1 = (b0 – a0)/ 2

If f(a0)\*f(c0) < 0 then [a1, c0], then b1  = c0 and a1 = a0.

b1 – a1 = (b0 – a0)/ 2

In both cases, b1 – a1 = (b0 – a0)/ 2. Now consider

b1 – a1 = 2−1(b0 – a0).

b2 – a2 = 2−1(b1 – a1) = 2−2(b0 – a0). …

bn – an = 2n-1(b1 – a1) = 2−n(b0 – a0).

Hence bn − an = 2−n(b0 − a0).

c. Show that, for all n, anbn + an−1bn−1 = an−1bn + anbn−1.

anbn + an−1bn−1 = an−1bn + anbn−1

Then anbn + an−1bn−1 - an−1bn - anbn−1 = 0

Then an (bn + bn−1)- an−1 (bn + bn−1) = 0

Then (bn + bn−1) (an+ an-1) = 0

When [an-1, bn-1], then next step will remain either an-1 or bn-1, So the equation above will always true when one of both b or a size will remain 0, then the equation always true for n ≥ 1.

15. (Continuation) Can it happen that a0 = a1 = a2 =···

**Answer**:

Only If f(an)\*f(cn) < 0 then [an+1, cn], then bn+1  = cn and an+1 = an.

In this case, a0 can remain then a0 = a1 = a2 =··· can happen.

16. (Continuation) Let cn = (an + bn)/2. Show that

lim n→∞ cn = lim n→∞ an = lim n→∞ bn .

**Answer**:

From above:

Consider: bn − an = 2−n(b0 − a0).

Then: lim n→∞ (bn ) − lim n→∞ ( an) = lim n→∞ 2−n(b0 – a0) = 0

Then: **lim n→∞ (bn ) = lim n→∞ ( an)**

Now, consider: cn = (an + bn)/2

Then **lim n→∞ cn** = lim n→∞ (an + bn)/2 = 0.5 (lim n→∞ an + lim n→∞ bn) = 0.5(2\* lim n→∞ an ) = **lim n→∞ an**

Hence: lim n→∞ cn = lim n→∞ an = lim n→∞ bn .

19. (True–False) Using the notation of the text, determine which of these assertions are true and which are generally false:

b. an ≤ r ≤ cn

c. cn ≤ r ≤ bn

d. |r − an| ≤ 2−n

Answer:

b. an ≤ r ≤ cn

False – Since r can lie in the interval (cn, bn) if If(bn)\*f(cn) < 0

c. cn ≤ r ≤ bn

False – Since r can lie in the interval (an, cn) if If(an)\*f(cn) < 0

d. |r − an| ≤ 2−n

True - Since r − an ≤ bn – an­ = 2-n (b0 – a0­). Hence:|r − an| ≤ 2−n.

Section 3.2: 5, 6, 14

5. The equation x − Rx−1 = 0 has x = ±R1/2 for its solution. Establish Newton’s iterative scheme, in simplified form, for this situation. Carry out five steps for R = 25 and x0 = 1.

**Answer**:

f(x) = x − Rx−1 = 0

Then f(x) = x2 – R = 0 (when multiple x into the equation we still get f(x) = 0)

Newton’s method: A picture containing text, antenna

Description automatically generated

f(x) = x2 – R, then f’(x) = 2x

xn+1 = xn – ((xn 2 – R)/(2xn)) = (xn 2 + R)/(2xn)

R = 25 and x0 = 1

Step 1: x1 = (x0 2 + R)/(2x0) = 13

Step 2: x2 = (x1 2 + R)/(2x1) = 194/23 = 97/13

Step 3: x3 = (x2 2 + R)/(2x2) = 6817/1261 ~ 5.406

Step 4: x4 = (x3 2 + R)/(2x3) ~ 5.015

Step 5: x5 = (x4 2 + R)/(2x4) ~ 5

Hence: x1  = 13; x2  = 97/13; x3  ~ 5.406; x4  ~ 5.015; x5  ~ 5;

6. Using a calculator, observe the sluggishness with which Newton’s method converges in the case of f (x) = (x − 1)m with m = 8 or 12. Reconcile this with the theory. Use x0 = 1.1.

Answer:

f (x) = (x − 1)m with m = 8 or 12 Use x0 = 1.1

then f’(x) = m (x − 1)m-1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| m = 8 | | | | |
| n | xn | f(xn) | f'(xn) | xn+1 |
| 0 | 1.1 | 10-8 | 8\*10-7 | 1.088 |
| 1 | 1.088 | 3.436\* 10-8 | 3.142\* 10-7 | 1.077 |
| 2 | 1.077 | 1.185\* 10-8 | 1.238\* 10-7 | 1.067 |
| 3 | 1.067 | 4.061\* 10-8 | 4.849\* 10-8 | 1.089 |
| 4 | 1.059 | 1.391\* 10-8 | 1.898\* 10-8 | 1.051 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| m = 12 | | | | |
| n | xn | f(xn) | f'(xn) | xn+1 |
| 0 | 1.1 | 10-12 | 8 \* 10-11 | 1.088 |
| 1 | 1.088 | 2.014\* 10-13 | 2.762\* 10-11 | 1.08 |
| 2 | 1.08 | 7.081\* 10-14 | 1.06\* 10-11 | 1.074 |
| 3 | 1.074 | 2.486\* 10-14 | 4.058\* 10-12 | 1.067 |
| 4 | 1.067 | 8.789\* 10-15 | 1.565\* 10-12 | 1.062 |

When exponent is large then it moving towards the solution with slow speed. Otherwise.

14. Determine the formulas for Newton’s method for finding a root of the function f (x) = x – (e/x). What is the behavior of the iterates?

Answer:

f (x) = x – (e/x).

then f’ (x) = 1 + (e/x2).

Newton’s method: A picture containing text, antenna

Description automatically generated

xn+1 = xn – ((xn – (e/xn))/( 1 + (e/xn2)))

= xn– (((xn2 -e) xn)/( xn2 +e))

= (xn3 + exn - xn3 +e xn)/(xn2 +e)

= (2exn)/(xn2 +e)

Then xn+1 = (2exn)/(xn2 +e). When xn  > 1, then use the formula to get the iterates decrease when 0 < xn  ≤ 1, then the iterates increase when xn  <0.

Section 3.3: 2, 5, 13 (b,d)

2. If we use the secant method on f (x) = x 3 − 2x + 2 starting with x0 = 0 and x1 = 1, what is x2?

**Answer:**

Secant method: Diagram

Description automatically generated

with x0 = 0 and x1 = 1.

f (x) = x 3 − 2x + 2 then f(0) = 2; and f(1) = 1

x2 = x1 – ((x1 – x0)/(f(x1)-f(x0))) = 1 - ((1-0)/(1-2)) = 2

Hence, x2 = 2.

5. Using the bisection method, Newton’s method, and the secant method, find the largest positive root correct to three decimal places of x 3 − 5x + 3 = 0. (All roots are in [−3, +3].)

**Answer**:

x 3 − 5x + 3 = 0. (All roots are in [−3, +3].)

f(1) = -1, f(2) = 1

As f(1)f(2) = -1 < 0

Use Bisection interval [1,2]

c1 = (1+2)/2 = 1.5

f(c1) = -1.125 < 0. Then interval [1.5,2]

c2 = (1.5+2)/2 = 1.75

f(c2) = -0.391 < 0. Then interval [1.75,2]

c3 = (1.75+2)/2 = 1.875

f(c3) = 0.217 > 0. Then interval [1.75,1.875]

c4 = (1.75+1.875)/2 = 1.8125

f(c4) = - 0.1082 > 0. Then interval [1.8125,1.875]

c5 = (1.8125+1.875)/2 = 1.844

f(c5) = 0.05 > 0. Then interval [1.8125,1.844]

So on…

Newton’s Method

A picture containing text, antenna

Description automatically generated

Let x0 = 2

f(x) = x 3 − 5x + 3

f’(x) = 3X2 -5

x1 = 1.875 then f(x1) = 0.216

x2 = 1.835 then f(x2) = 0.004

x3 = 1.834 then f(x3) = -0.001

x4 = 1.834 then f(x4) = -0.001

x5 = 1.834 then f(x5) = -0.001

Secant method

Diagram

Description automatically generated

x0= 1; x1 = 2.

x2 = 1.5 then f(x2) = -1.125

x3 = 1.765 then f(x3) = -0.3266

x4 = 1.873 then f(x4) = 0.2057

x5 = 1.831 then f(x5) = -0.0165

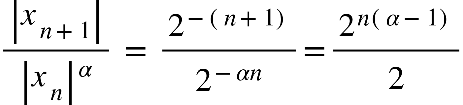
13. Test the following sequences for different types of convergence (i.e., linear, superlinear, or quadratic), where n = 1, 2, 3 ... .

b. xn = 2−n

d. xn = 2−an with a0 = a1 = 1 and an+1 = an + an−1 for n ≥ 2.

**Answer**:

b. xn = 2−n



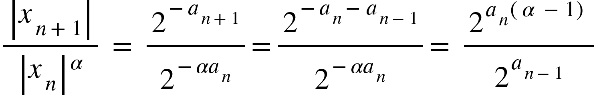
If α < 1, the exponent is negative, and the ratio converges to 0.

If α = 1, the exponent is 0, and the ratio converges to 1/2.

If α > 1, the exponent is diverges.

Hence the order of convergence is linear.

d. xn = 2−an with a0 = a1 = 1 and an+1 = an + an−1 for n ≥ 2.



If α < 1, the exponent is negative, and the ratio converges to 0.

If α = 1, the exponent is 0, and the ratio converges to .

If α > 1, the exponent is diverges.

Hence the order of convergence is linear.

**Computing Exercises**

Section 3.1: 3, 7

3. Find a root of the equation tan x = x on the interval [4, 5] by using the bisection method. What happens on the interval [1, 2]?

Answer:

f(x) = tanx -x = 0

**Code**:

clc;

% Display answer

disp('the interval [4, 5]');

root = bisectionMethod(4, 5)

disp('the interval [1, 2]');

root = bisectionMethod(1, 2)

% Bisection function

function root = bisectionMethod(a,b)

f = @(x) (tan(x) - x); % Function

ermin = 0.000001; % limit of answer

error = 3; % check error

c = (a + b)/2; % Middle point

while (error > ermin)

if(f(c)==0)

root = c;

break;

end

if (f(a)\*f(c) < 0)

b = c;

error = abs(b - a);

c = (a+b)/2;

else

a = c;

error = abs(b - a);

c = (a+b)/2;

end

end

root = c;

disp('f(root)');

f(root)

end

**Result**:

Graphical user interface, text, application

Description automatically generated

7. Use the bisection method to determine roots of these functions on the intervals indicated. Process all three functions in one computer run.

f (x) = x 3 + 3x − 1 on [0, 1]

g(x) = x 3 − 2 sin x on [0.5, 2]

h(x) = x + 10 − x cosh(50/x) on [120, 130]

Answer:

Code:

clc;

f = @(x) (x\*x\*x + 3\*x -1);

a = 0;

b = 1;

disp('f(x) = (x\*x\*x + 3\*x -1) on [0,1]')

root = bisection(f,a,b)

g = @(x) (x\*x\*x - 2\*sin(x));

e = 0.5;

f = 2;

disp('g(x) = (x\*x\*x - 2\*sin(x)) on [0.5, 2]')

root = bisection(g,e,f)

h = @(x) (x + 10 - x\*cosh(50/x));

k = 120;

i = 130;

disp('g(x) = (x + 10 - x\*cosh(50/x)) on [120, 130]')

root = bisection(h,k,i)

function root = bisection(f,a,b)

ermin = 0.000001; % limit of answer

error = 3; % check error

c = (a + b)/2; % Middle point

while (error > ermin)

if(f(c)==0)

root = c;

break;

end

if (f(a)\*f(c) < 0)

b = c;

error = abs(b - a);

c = (a+b)/2;

else

a = c;

error = abs(b - a);

c = (a+b)/2;

end

end

root = c;

disp('f(root)');

f(root)

end

Result:

Graphical user interface, text, application

Description automatically generated

Section 3.2: 2, 6, 15, 18

2. Write a simple, self-contained program to apply Newton’s method to the equation x 3 + 2x 2 + 10x = 20, starting with x0 = 2. Evaluate the appropriate f (x) and f’ (x), using nested multiplication. Stop the computation when two successive points differ by 1/ 2 × 10−5 or some other convenient tolerance close to your machine’s capability. Print all intermediate points and function values. Put an upper limit of ten on the number of steps.

**Answer**:

Code:

clc;

% Given

x = [2];

f = @(x) (x^3 + 2\*x^2 + 10\*x - 20);

h = @(x) (x^3 + 2\*x^2 + 10\*x - 20)/ (3\*x^2 + 4\*x + 10);

% Check

stop = 3;

while(stop > 0.0000005)

n = length(x); % Get lenght of x

temp = x(n) - h(x(n)); % Get new xi

x\_new = [x(1:n) temp] ; % add to new list

n\_new = length(x\_new); % get index of the last value in the new index

stop = abs(x(n)-x\_new(n\_new)); % get The diff

x = x\_new; % save the list

stop;

end

% Print

for i = 1: length(x)

fprintf('Index %i have x= %i then f(x)= %i \n', i, x(i),f(x(i)));

end

Result:

Text

Description automatically generated

6. In celestial mechanics, **Kepler’s equation** is important. It reads x = y − ε sin y, in which x is a planet’s mean anomaly, y its eccentric anomaly, and ε the eccentricity of its orbit. Taking ε = 0.9, construct a table of y for 30 equally spaced values of x in the interval 0 ≤ x ≤ π. Use Newton’s method to obtain each value of y. The y corresponding to an x can be used as the starting point for the iteration when x is changed slightly.

**Answe**r:

Code:

clc;

% Given

Y = [0.1];

f = @(Y) (Y - 0.9\*sin(Y));

h = @(Y) (Y - 0.9\*sin(Y))/ (1 - 0.9\*cos(Y));

% Check

stop = 1;

while(stop <= 30)

n = length(Y); % Get lenght of x

temp = Y(n) - h(Y(n)); % Get new xi

Y\_new = [Y(1:n) temp+0.1] ; % add to new list

n\_new = length(Y\_new); % get index of the last value in the new index

stop = stop +1; % get The diff

Y = Y\_new; % save the list

end

% Print

for i = 1: length(Y)

fprintf('Index %i have Y= %i then f(y)= %i \n', i, Y(i),f(Y(i)));

end

Result:

Table

Description automatically generated

15. Find the root of the equation 1 /2 x 2 + x + 1 − ex = 0 by Newton’s method, starting with x0 = 1, and account for the slow convergence.

**Answer**:

Code:

clc;

% Given

x = [1];

f = @(x) (0.5\*x^2 + x + 1 - exp(x));

f1 = @(x)(x + 1 + - exp(x));

h = @(x) (0.5\*x^2 + x + 1 - exp(x))/ (x + 1 + - exp(x));

% Check

stop = 3;

while(stop > 0.0000005)

n = length(x); % Get lenght of x

temp = x(n) - h(x(n)); % Get new xi

x\_new = [x(1:n) temp] ; % add to new list

n\_new = length(x\_new); % get index of the last value in the new index

stop = abs(x(n)-x\_new(n\_new)); % get The diff

x = x\_new; % save the list

stop;

end

% Print

for i = 1: length(x)

fprintf('Index %i have x= %i then f(x)= %i \n', i, x(i),f(x(i)));

if (abs(f1(x(i))) < 0.00005 && i ~= length(x))

fprintf('Small derivative \n ');

end

if(i == length(x))

fprintf('Convergence \n ');

end

end

**Result**:

Index 1 have x= 1 then f(x)= -2.182818e-01   
Index 2 have x= 6.961056e-01 then f(x)= -6.753850e-02   
Index 3 have x= 4.781129e-01 then f(x)= -2.061871e-02   
Index 4 have x= 3.252851e-01 then f(x)= -6.234993e-03   
Index 5 have x= 2.198578e-01 then f(x)= -1.873025e-03   
Index 6 have x= 1.479339e-01 then f(x)= -5.601354e-04   
Index 7 have x= 9.923640e-02 then f(x)= -1.670002e-04   
Index 8 have x= 6.643295e-02 then f(x)= -4.968763e-05   
Index 9 have x= 4.441177e-02 then f(x)= -1.476321e-05   
Index 10 have x= 2.966279e-02 then f(x)= -4.382407e-06   
Index 11 have x= 1.979969e-02 then f(x)= -1.300099e-06   
Index 12 have x= 1.321069e-02 then f(x)= -3.855329e-07   
Index 13 have x= 8.811982e-03 then f(x)= -1.142949e-07   
Small derivative   
 Index 14 have x= 5.876813e-03 then f(x)= -3.387760e-08   
Small derivative   
 Index 15 have x= 3.918835e-03 then f(x)= -1.004027e-08   
Small derivative   
 Index 16 have x= 2.612983e-03 then f(x)= -2.975380e-09   
Small derivative   
 Index 17 have x= 1.742179e-03 then f(x)= -8.816901e-10   
Small derivative   
 Index 18 have x= 1.161537e-03 then f(x)= -2.612603e-10   
Small derivative   
 Index 19 have x= 7.743954e-04 then f(x)= -7.741430e-11   
Small derivative   
 Index 20 have x= 5.162802e-04 then f(x)= -2.293832e-11   
Small derivative   
 Index 21 have x= 3.441940e-04 then f(x)= -6.796785e-12   
Small derivative   
 Index 22 have x= 2.294640e-04 then f(x)= -2.013945e-12   
Small derivative   
 Index 23 have x= 1.529721e-04 then f(x)= -5.966339e-13   
Small derivative   
 Index 24 have x= 1.019813e-04 then f(x)= -1.767475e-13   
Small derivative   
 Index 25 have x= 6.799318e-05 then f(x)= -5.240253e-14   
Small derivative   
 Index 26 have x= 4.532369e-05 then f(x)= -1.554312e-14   
Small derivative   
 Index 27 have x= 3.019118e-05 then f(x)= -4.440892e-15   
Small derivative   
 Index 28 have x= 2.044721e-05 then f(x)= -1.332268e-15   
Small derivative   
 Index 29 have x= 1.407412e-05 then f(x)= -6.661338e-16   
Small derivative   
 Index 30 have x= 7.348277e-06 then f(x)= -2.220446e-16   
Small derivative   
 Index 31 have x= -8.760156e-07 then f(x)= 0   
Small derivative   
 Index 32 have x= -8.760156e-07 then f(x)= 0   
Convergence

18. Using the bisection method, find the positive root of 2x(1 + x 2)−1 = arctan x. Using the root as x0, apply Newton’s method to the function arctan x. Interpret the results.

Answer:

Code:

clc;

f = @(x) (2\*x\*(1/(1+x\*x)) - atan(x));

h = @(x) (2\*x\*(1/(1+x\*x)) - atan(x))/(1/(x\*x +1));

a = 0;

b = 1;

disp('f(x) = (x\*x\*x + 3\*x -1) on [0,1]')

root = bisection(f,a,b)

x = [root];

final = NewTon(f,h,x);

for i = 1: length(final)

fprintf('Index %i have x= %i then f(x)= %i \n', i, final(i),f(final(i)));

end

function final = NewTon (f,h,x)

% Check

stop = 3;

while(stop > 0.0000005)

n = length(x); % Get lenght of x

temp = x(n) - h(x(n)); % Get new xi

x\_new = [x(1:n) temp] ; % add to new list

n\_new = length(x\_new); % get index of the last value in the new index

stop = abs(x(n)-x\_new(n\_new)); % get The diff

x = x\_new; % save the list

final = x;

end

end

function root = bisection(f,a,b)

ermin = 0.000001; % limit of answer

error = 3; % check error

c = (a + b)/2; % Middle point

while (error > ermin)

if(f(c)==0)

root = c;

break;

end

if (f(a)\*f(c) < 0)

b = c;

error = abs(b - a);

c = (a+b)/2;

else

a = c;

error = abs(b - a);

c = (a+b)/2;

end

end

root = c;

disp('f(root)');

f(root)

end

Result:

f(x) = (x\*x\*x + 3\*x -1) on [0,1]  
f(root)  
  
ans =  
  
 0.2146  
  
  
root =  
  
 1.0000  
  
Index 1 have x= 9.999995e-01 then f(x)= 2.146021e-01   
Index 2 have x= 5.707956e-01 then f(x)= 3.423846e-01   
Index 3 have x= 1.168595e-01 then f(x)= 1.142384e-01   
Index 4 have x= 1.061010e-03 then f(x)= 1.061008e-03   
Index 5 have x= 7.962825e-10 then f(x)= 7.962825e-10   
Index 6 have x= 0 then f(x)= 0

Section 3.3: 7, 10

7. Write a simple program to find the root of f (x) = x 3 + 2x 2 + 10x − 20 using the secant method with starting values x0 = 2 and x1 = 1. Let it run at most 20 steps, and include a stopping test as well. Compare the number of steps needed here to the number needed in Newton’s method. Is the convergence quadratic?

Answer:

Code:

clc;

f= @(x) (x\*x\*x + 2\*x\*x + 10\*x -20 );

f1 = @(x) (3\*x\*x + 4\*x + 10);

h = @(x) (x\*x\*x + 2\*x\*x + 10\*x -20 )/ (3\*x\*x + 4\*x + 10);

fprintf('Secant method \n');

x = [2,1];

final1 = Secant(f,x);

for i = 1: length (final)

fprintf('Index %i have x= %i then f(x)= %i \n', i, final(i),f(final(i)));

if (abs(f1(final(i))) < 0.00005 && i ~= length(final))

fprintf('Small derivative \n ');

end

if(i == length(final))

fprintf('Convergence \n ');

end

end

fprintf('\nNewton method \n');

final2 = NewTon (f,h,x);

% Print

for i = 1: length(final2)

fprintf('Index %i have x= %i then f(x)= %i \n', i, final2(i),f(final2(i)));

if (abs(f1(final2(i))) < 0.00005 && i ~= length(final2))

fprintf('Small derivative \n ');

end

if(i == length(final2))

fprintf('Convergence \n ');

end

end

function final = NewTon (f,h,x)

% Check

stop = 3;

while(stop > 0.0000005)

n = length(x); % Get lenght of x

temp = x(n) - h(x(n)); % Get new xi

x\_new = [x(1:n) temp] ; % add to new list

n\_new = length(x\_new); % get index of the last value in the new index

stop = abs(x(n)-x\_new(n\_new)); % get The diff

x = x\_new; % save the list

final = x;

end

end

function final = Secant(f,x)

n = length(x);

i = 1;

while (i <20)

temp = x(n) - ((x(n)-(x(n-1)))/(f(x(n)) - f(x(n-1))))\* f(x(n));

x\_new = [x(1:n) temp] ;

x = x\_new;

n = length(x);

i = i +1;

if(f(x(n))==0)

final = x;

break;

end

end

final = x;

end

Result:

Secant method   
Index 1 have x= 2 then f(x)= 16   
Index 2 have x= 1 then f(x)= -7   
Index 3 have x= 1.304348e+00 then f(x)= -1.334758e+00   
Index 4 have x= 1.376054e+00 then f(x)= 1.531733e-01   
Index 5 have x= 1.368672e+00 then f(x)= -2.872217e-03   
Index 6 have x= 1.368808e+00 then f(x)= -6.018647e-06   
Index 7 have x= 1.368808e+00 then f(x)= 2.372076e-10   
Index 8 have x= 1.368808e+00 then f(x)= 0   
Convergence   
   
Newton method   
Index 1 have x= 2 then f(x)= 16   
Index 2 have x= 1 then f(x)= -7   
Index 3 have x= 1.411765e+00 then f(x)= 9.175656e-01   
Index 4 have x= 1.369336e+00 then f(x)= 1.114812e-02   
Index 5 have x= 1.368808e+00 then f(x)= 1.704487e-06   
Index 6 have x= 1.368808e+00 then f(x)= 3.907985e-14   
Convergence

Explant:

Both the method give the same root of the equation. Hence, Newton method give the fast convergence while secant method give the slow convergence. The above convergence is quadratic.

10. Test the secant method on an example in which r, f’ (r), and f’’ (r) are known in advance. Monitor the ratios en+1/(enen−1)to see whether they converge to −1/ 2 f’’ (r)/ f’ (r). The function f (x) = arctan x is suitable for this experiment.

**Answer**:

Code:

clc;

f= @(x) (atan(x) );

fprintf('Secant method \n');

x = [0,1];

final1 = Secant(f,x);

for i = 1: length (final)

fprintf('Index %i have x= %i then f(x)= %i \n', i, final(i),f(final(i)));

if (abs(f1(final(i))) < 0.00005 && i ~= length(final))

fprintf('Small derivative \n ');

end

if(i == length(final))

fprintf('Convergence \n ');

end

end

function final = Secant(f,x)

n = length(x);

i = 1;

while (i <20)

temp = x(n) - ((x(n)-(x(n-1)))/(f(x(n)) - f(x(n-1))))\* f(x(n));

x\_new = [x(1:n) temp] ;

x = x\_new;

n = length(x);

i = i +1;

if(f(x(n))==0)

final = x;

break;

end

end

final = x;

end

Result:

Secant method   
Index 1 have x= 2 then f(x)= 1.107149e+00   
Index 2 have x= 1 then f(x)= 7.853982e-01   
Index 3 have x= 1.304348e+00 then f(x)= 9.167136e-01   
Index 4 have x= 1.376054e+00 then f(x)= 9.423644e-01   
Index 5 have x= 1.368672e+00 then f(x)= 9.398043e-01   
Index 6 have x= 1.368808e+00 then f(x)= 9.398516e-01   
Index 7 have x= 1.368808e+00 then f(x)= 9.398517e-01   
Index 8 have x= 1.368808e+00 then f(x)= 9.398517e-01   
Convergence

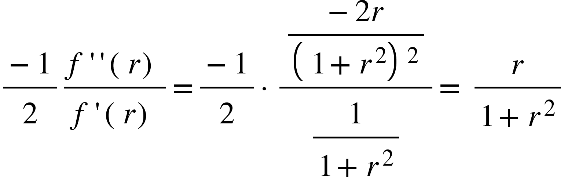
**Explant**:

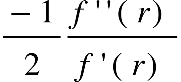
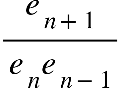
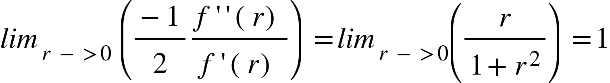
f(x) = arctan x

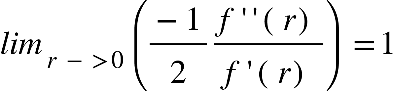
Here r is a root of function f(x).

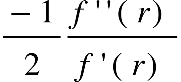
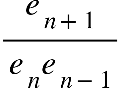
f'(x) = 1/ (1+x2) -> f'’(x) = -(2x)/ (1+x2)2

f'(r) = 1/ (1+r2) -> f'’(r) = -(2r)/ (1+r2)2

Now 

Since  converges to  if 

That is 

Therefore  converges to  . Hence f(x) = arctan x is a convergence function