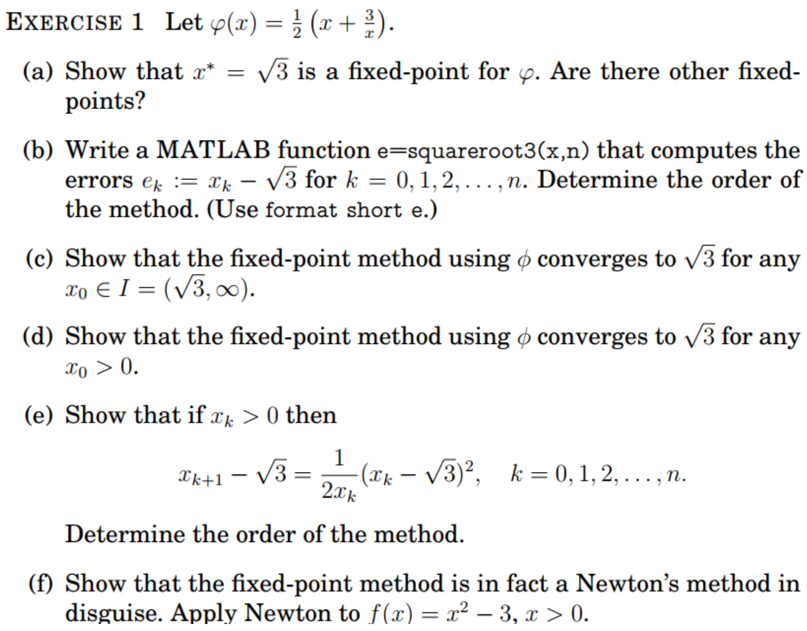


EXERCISE 1 Let $\varphi(x) = \frac{1}{2}\left(x + \frac{3}{x}\right)$.

(a) Show that $x^* = \sqrt{3}$ is a fixed-point for φ. Are there other fixed-points?

(b) Write a MATLAB function e=squareroot3(x,n) that computes the errors $e_k := x_k - \sqrt{3}$ for $k = 0, 1, 2, \ldots, n$. Determine the order of the method. (Use format short e.)

(c) Show that the fixed-point method using ϕ converges to $\sqrt{3}$ for any $x_0 \in I = (\sqrt{3}, \infty)$.

(d) Show that the fixed-point method using ϕ converges to $\sqrt{3}$ for any $x_0 > 0$.

(e) Show that if $x_k > 0$ then

$$x_{k+1} - \sqrt{3} = \frac{1}{2x_k}(x_k - \sqrt{3})^2, \quad k = 0, 1, 2, \ldots, n.$$

Determine the order of the method.

(f) Show that the fixed-point method is in fact a Newton's method in disguise. Apply Newton to $f(x) = x^2 - 3$, $x > 0$.

(a) A fixed point of a function is an element that is mapped to itself by the function. That is, c is a fixed point of a function f if c belongs to both the domain and the codomain of f, and f(c) = c

=> $x* = \sqrt{3}\ and\ \varphi(x) = \frac{1}{2}\left(x + \frac{3}{x}\right)$. Replacing $x* = \sqrt{3}$ into the function, we have:

$\varphi\left(\sqrt{3}\right) = \frac{1}{2}\left(\sqrt{3} + \frac{3}{\sqrt{3}}\right) = \frac{1}{2}\left(\frac{6}{\sqrt{3}}\right) = \frac{3}{\sqrt{3}} = \sqrt{3} = x* \Rightarrow x* = \sqrt{3}$ is a fixed point for $\varphi(x)$

Solving the equation $x = \frac{1}{2}\left(x + \frac{3}{x}\right)$ to find all fixed points of the function $\varphi(x)$. Condition: $x \neq 0$.

=> $x = \frac{1}{2}\left(x + \frac{3}{x}\right) \Rightarrow 2x^2 = x^2 + 3 \Rightarrow x^2 = 3 \Rightarrow x = \pm\sqrt{3}$ => Another fixed point is $x = -\sqrt{3}$

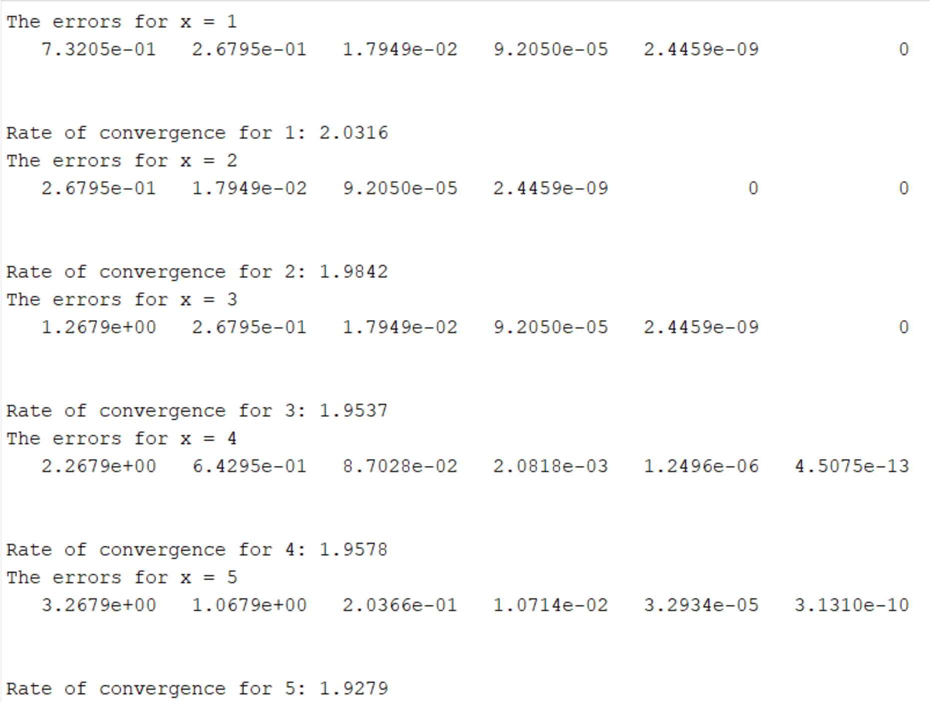
(b) The Matlab code for squareroot3.m is

```
%{
This function uses recursion to implemen the iteration method
We need to find the square root of 3. In other words, we should find
the
root of the function x^2 = 3.
We should convert this function to the form: x = f(x) so that
it will become an iteration method
=> x = 1/2 * (x + 3/x)
In the function, the output is a matrix of errors for n =
0,1,2,3,...
%}
function [errors] = squareroot3(x,n)
    format short e
    errors = inner([],x,n);
end

function [errorsLoop] = inner(errorsInput,x,n)
    if n ~= 0
        x_nextiteration = 1/2 * (x + 3/x);
        error = [errorsInput, abs(x - sqrt(3))];
        errorsLoop = inner(error, x_nextiteration, n - 1);
    else
        errorsLoop = [errorsInput, abs(x - sqrt(3))];
    end
end
```

The Matlab code for squareroot3_order.m is:

```
clc;
% I choose n = 5 to obtain a correct rate of convergence
% The rate of converge to sqrt(3) for x = 1 to 5 are
for x = 1:5
errors = squareroot3(x,5);
disp("The errors for x = " + x)
disp(errors);
disp(" ");
disp("Rate of convergence for " + x + ": " + detectrate(errors));
end
```



```
The errors for x = 1
   7.3205e-01   2.6795e-01   1.7949e-02   9.2050e-05   2.4459e-09            0


Rate of convergence for 1: 2.0316
The errors for x = 2
   2.6795e-01   1.7949e-02   9.2050e-05   2.4459e-09            0            0


Rate of convergence for 2: 1.9842
The errors for x = 3
   1.2679e+00   2.6795e-01   1.7949e-02   9.2050e-05   2.4459e-09            0


Rate of convergence for 3: 1.9537
The errors for x = 4
   2.2679e+00   6.4295e-01   8.7028e-02   2.0818e-03   1.2496e-06   4.5075e-13


Rate of convergence for 4: 1.9578
The errors for x = 5
   3.2679e+00   1.0679e+00   2.0366e-01   1.0714e-02   3.2934e-05   3.1310e-10


Rate of convergence for 5: 1.9279
```

=> The rate of convergence is around $\alpha = 2$ => Order of the method is 2

(c) Proof

$$\lim_{x\to+\infty} \frac{1}{2}\left(x+\frac{3}{x}\right) = \frac{1}{2}(x+0) = \frac{1}{2}x$$

$$\lim_{x\to\sqrt{3}} \frac{1}{2}\left(x+\frac{3}{x}\right) = \frac{1}{2}\left(\sqrt{3}+\frac{3}{\sqrt{3}}\right) = \sqrt{3}$$

As we can see, big number converges by being halved in each iteration => All x converges to smaller value when x is large. When x is near sqrt(3), it converges to sqrt(3). This can be seen in the limits

=> The fixed point methos using φ converges to $\sqrt{3}$ for any $x_0 \in I = \left(\sqrt{3},\infty\right)$

(d) Proof

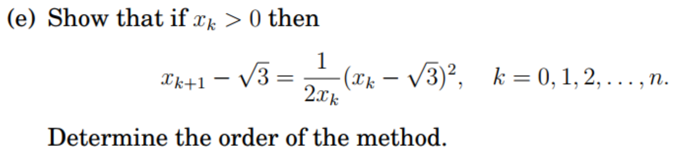
First iteration: $\lim_{x\to 0}\frac{1}{2}\left(x+\frac{3}{x}\right)=\frac{1}{2}\left(0+\infty\right)=\infty$

Second iteration: $\lim_{x\to +\infty}\frac{1}{2}\left(x+\frac{3}{x}\right)=\frac{1}{2}\left(x+0\right)=\frac{1}{2}x$

We can see that as x approaches 0, the result will reach positive infinity in the second iteration. And we know that large values will converge to sqrt(3) again as proved in (c) from second iteration onwards.

$x_0\in I=\left(0,\sqrt{3}\right)=> x+\frac{3}{x}\in\left(2\sqrt{3},\infty\right)=>\frac{1}{2}\left(x+\frac{3}{x}\right)-x\in\left(\sqrt{3},\infty\right)$ is strictly positive, so we can see that after each iteration, values of x strictly increases. If x exceeds sqrt(3), it will starts to converge down as seen from (c).

=> For all x > 0, the fixed point method converges to sqrt(3)



(e) Show that if $x_k > 0$ then

$$x_{k+1}-\sqrt{3}=\frac{1}{2x_k}(x_k-\sqrt{3})^2, \quad k=0,1,2,\ldots,n.$$

Determine the order of the method.

We have: $x_{k+1}=\frac{1}{2}\left(x_k+\frac{3}{x_k}\right)$

$$=> x_{k+1}-\sqrt{3}=\frac{1}{2}\left(x_k+\frac{3}{x_k}\right)-\sqrt{3}=\frac{1}{2}\left(x_k-2\sqrt{3}+\frac{3}{x_k}\right)=\frac{1}{2x_k}\left(x_k^2-2\sqrt{3}x_k+3\right)=\frac{1}{2x_k}\left(x_k-\sqrt{3}\right)^2$$

=> LHS = RHS (proven)

Order of the method

Suppose $\left\{x_k\right\}_{k=0}^{\infty}$ is a sequence that converges to x*, with $x_k\neq x*$ for all n. If positive constants λ and α exist with

$$\lim_{k\to\infty}\frac{\left|x_{k+1}-x*\right|}{\left|x_k-x*\right|^{\alpha}}=\lambda$$

Then $\left\{x_k\right\}_{k=0}^{\infty}$ converges to p of order α, with asymptotic error constant λ

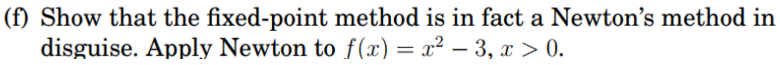
By definition, we know that $\lim_{k\to\infty}x_k=\sqrt{3}$, since the iteration method is convergent to sqrt(3) as found in (d) for all x > 0

$$=> \lim_{k\to\infty}\frac{\left|x_{k+1}-\sqrt{3}\right|}{\left|x_k-\sqrt{3}\right|^{\alpha}}=\lim_{k\to\infty}\frac{\frac{1}{2x_k}\left(x_k-\sqrt{3}\right)^2}{\left|x_k-\sqrt{3}\right|^{\alpha}}=\lim_{k\to\infty}\frac{1}{2x_k}\left(x_k-\sqrt{3}\right)^{2-\alpha}$$

We need to find α such that the result of the limit is a constant.

Observation: $\lim_{k\to\infty}\frac{1}{2x_k}=\frac{1}{2}\lim_{k\to\infty}\frac{1}{x_k}=\frac{1}{2\sqrt{3}}$ which is constant => We can make the term $\left(x_k-\sqrt{3}\right)^{2-\alpha}$ vanished => $\alpha=2$

=> $\lambda=\lim_{k\to\infty}\frac{\left|x_{k+1}-\sqrt{3}\right|}{\left|x_k-\sqrt{3}\right|^2}=\frac{1}{2\sqrt{3}}$. The order of the method is thus 2

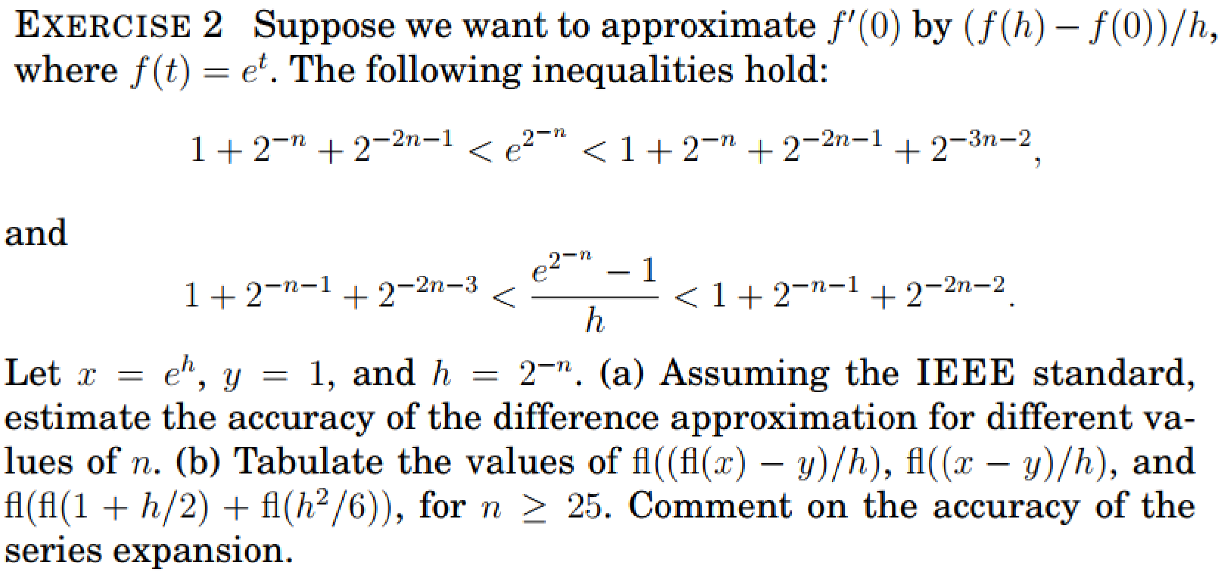


(f) Show that the fixed-point method is in fact a Newton's method in disguise. Apply Newton to $f(x)=x^2-3$, $x>0$.

The Newton's Method is given by: $x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$. Now we applied Newton's method to the function $f(x)=x^2-3, x>0$

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}=x_k-\frac{x_k^2-3}{(x_k^2-3)'}=x_k-\frac{x_k^2-3}{2x_k}=x_k-\frac{x_k}{2}-\frac{3}{2x_k}=\frac{x_k}{2}-\frac{3}{2x_k}=\frac{1}{2}\left(x_k-\frac{3}{x_k}\right)=\varphi(x)$$

=> The fixed point method is truly the Newton's method in disguise.



EXERCISE 2 Suppose we want to approximate $f'(0)$ by $(f(h)-f(0))/h$, where $f(t)=e^t$. The following inequalities hold:

$$1+2^{-n}+2^{-2n-1}<e^{2^{-n}}<1+2^{-n}+2^{-2n-1}+2^{-3n-2},$$

and

$$1+2^{-n-1}+2^{-2n-3}<\frac{e^{2^{-n}}-1}{h}<1+2^{-n-1}+2^{-2n-2}.$$

Let $x=e^h$, $y=1$, and $h=2^{-n}$. (a) Assuming the IEEE standard, estimate the accuracy of the difference approximation for different values of n. (b) Tabulate the values of $\mathrm{fl}((\mathrm{fl}(x)-y)/h)$, $\mathrm{fl}((x-y)/h)$, and $\mathrm{fl}(\mathrm{fl}(1+h/2)+\mathrm{fl}(h^2/6))$, for $n\geq 25$. Comment on the accuracy of the series expansion.

a) The estimation for the accuracy of the difference approximationis given by the floating-point number: $fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)$, $where\ h=2^{-n}, x=e^h=e^{2^{-n}}, y=1$.

We have: fl(y) = fl(1) = 1, fl(2^-n) = 2^-n, since they are integers and can be expressed exactly in floating-point. The formula can be simplified as:

$$fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=fl\left(\frac{fl\left(fl(x)-1\right)}{2^{-n}}\right)$$

The rounding formula for fl(x), x = e^(2^-n) is given by

$$1+2^{-n}+2^{-2n-1}<e^{2^{-n}}<1+2^{-n}+2^{-2n-1}+2^{-3n-2}$$

- For n = 25, we have:

$$1+2^{-25}+2^{-51}<e^{2^{-25}}<1+2^{-25}+2^{-51}+2^{-77}$$

We know that for n >= 53, 2^-n in floating-point cannot be recognized by computers and thus, it equals to 0.

=> $1+2^{-25}+2^{-51}<e^{2^{-25}}<1+2^{-25}+2^{-51}+0$

=> $fl(e^{2^{-25}})=1+2^{-25}+2^{-51}$

=> Difference approximation for n = 25:

$$fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=fl\left(\frac{fl\left(1+2^{-25}+2^{-51}-1\right)}{2^{-25}}\right)=fl\left(\frac{2^{-25}+2^{-51}}{2^{-25}}\right)=1+2^{-26}$$

Similarly for n = 26:

$$1+2^{-26}+2^{-53}<e^{2^{-26}}<1+2^{-26}+2^{-53}+2^{-80}$$

=> $1+2^{-26}+0<e^{2^{-26}}<1+2^{-26}+0+0$

=> $fl(e^{2^{-25}})=1+2^{-27}$

$$fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=fl\left(\frac{fl\left(1+2^{-26}-1\right)}{2^{-26}}\right)=fl\left(\frac{2^{-26}}{2^{-26}}\right)=1$$

n = 27:

$$1+2^{-27}+2^{-55}<e^{2^{-27}}<1+2^{-27}+2^{-55}+2^{-83}$$

=> $1+2^{-27}+0<e^{2^{-27}}<1+2^{-27}+0+0$

=> $fl(e^{2^{-27}})=1+2^{-27}$

$$fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=fl\left(\frac{fl\left(1+2^{-27}-1\right)}{2^{-27}}\right)=fl\left(\frac{2^{-27}}{2^{-27}}\right)=1$$

n = 53:

=> $1+0+0<e^{2^{-27}}<1+0+0+0$ => $fl(e^{2^{-53}})=1$

$$fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=fl\left(\frac{fl\left(1-1\right)}{2^{-27}}\right)=fl\left(\frac{0}{2^{-27}}\right)=0$$

=> For n = 25, $fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=1+2^{-26}$

=> For n > 25 and n < 53, $fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=1$

=> For n >= 53, $fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=0$

b) Accuracy of series expansion

*For $fl\left(\frac{fl\left(fl(x)-fl(y)\right)}{fl(h)}\right)=fl\left(\frac{fl\left(fl(x)-1\right)}{2^{-n}}\right)$: Done in part A

* For $fl\left(\frac{x-y}{h}\right)=fl\left(\frac{e^{2^{-n}}-1}{h}\right)$

- For n = 25, $1+2^{-26}+2^{-53}<\frac{e^{2^{-25}}-1}{h}<1+2^{-26}+2^{-52}$ => $1+2^{-26}+0<\frac{e^{2^{-25}}-1}{h}<1+2^{-26}+2^{-52}$

$fl\left(\frac{x-y}{h}\right)=fl\left(\frac{e^{2^{-n}}-1}{h}\right)=1+2^{-26}+2^{-52}$

- For n > 26 and n < 52, $fl\left(\frac{e^{2^{-n}}-1}{h}\right)=1+2^{-n-1}$

- For >= 52, $fl\left(\frac{e^{2^{-n}}-1}{h}\right)=1$

* For $fl\left(fl\left(1+\frac{h}{2}\right)+fl\left(\frac{h^2}{6}\right)\right)=fl\left(1+2^{-n-1}+\frac{2^{-2n}}{6}\right)=fl\left(1+2^{-n-1}+2^{-2n-\log_2 6}\right)$

- For n = 25, $fl\left(fl\left(1+\frac{h}{2}\right)+fl\left(\frac{h^2}{6}\right)\right)=fl\left(1+2^{-26}+2^{-52.584}\right)=1+2^{-26}+2^{-52.584}$

- For n > 26 and n < 52, $fl\left(fl\left(1+\frac{h}{2}\right)+fl\left(\frac{h^2}{6}\right)\right)=1+2^{-n-1}$

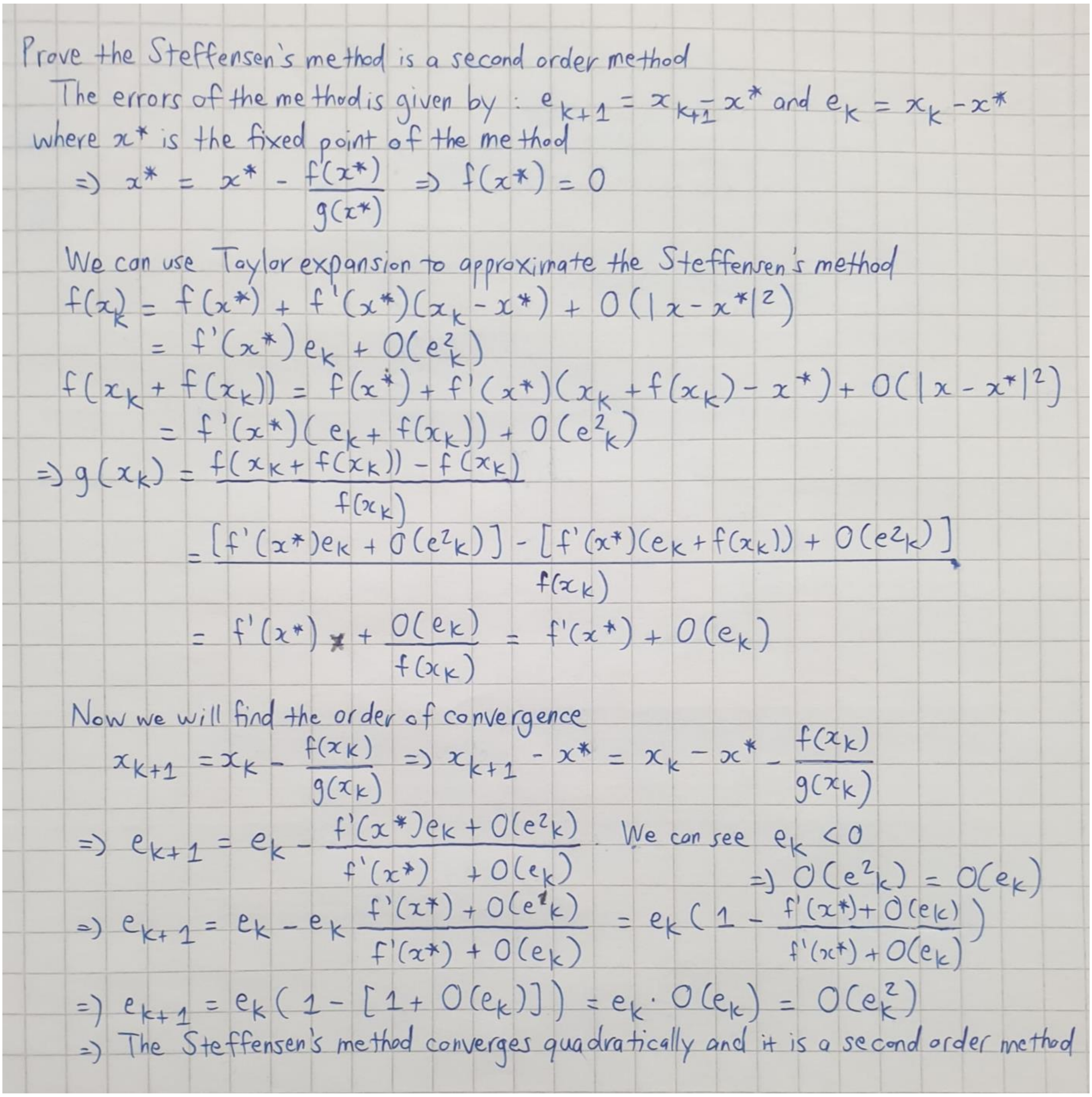
- For >= 52, $fl\left(fl\left(1+\frac{h}{2}\right)+fl\left(\frac{h^2}{6}\right)\right)=1$

=> As n increases, the accuracy gets close to 1 => The series expansion becomes more accurate

EXERCISE 3 Show that Steffensen's method

$$x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)}, \quad g(x_n) = \frac{f(x_n + f(x_n)) - f(x_n)}{f(x_n)}$$

is a second order method.



Prove the Steffensen's method is a second order method

The errors of the method is given by : $e_{k+1} = x_{k+1} - x^*$ and $e_k = x_k - x^*$ where x^* is the fixed point of the method

$$\Rightarrow x^* = x^* - \frac{f(x^*)}{g(x^*)} \Rightarrow f(x^*) = 0$$

We can use Taylor expansion to approximate the Steffensen's method

$$f(x_k) = f(x^*) + f'(x^*)(x_k - x^*) + O(|x - x^*|^2)$$
$$= f'(x^*) e_k + O(e_k^2)$$
$$f(x_k + f(x_k)) = f(x^*) + f'(x^*)(x_k + f(x_k) - x^*) + O(|x - x^*|^2)$$
$$= f'(x^*)(e_k + f(x_k)) + O(e^2_k)$$
$$\Rightarrow g(x_k) = \frac{f(x_k + f(x_k)) - f(x_k)}{f(x_k)}$$
$$= \frac{[f'(x^*)e_k + O(e^2_k)] - [f'(x^*)(e_k + f(x_k)) + O(e^2_k)]}{f(x_k)}$$
$$= f'(x^*) \cancel{*} + \frac{O(e_k)}{f(x_k)} = f'(x^*) + O(e_k)$$

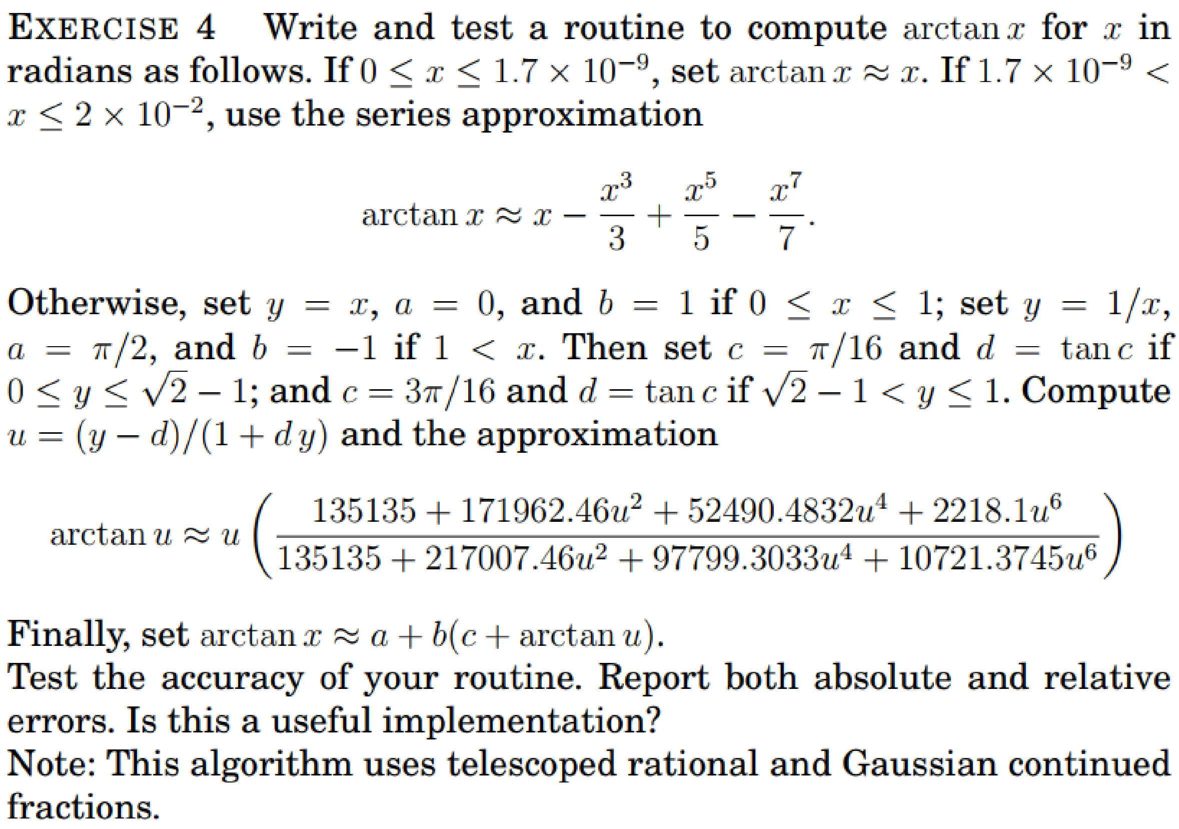
Now we will find the order of convergence

$$x_{k+1} = x_k - \frac{f(x_k)}{g(x_k)} \Rightarrow x_{k+1} - x^* = x_k - x^* - \frac{f(x_k)}{g(x_k)}$$
$$\Rightarrow e_{k+1} = e_k - \frac{f'(x^*)e_k + O(e^2_k)}{f'(x^*) + O(e_k)}$$

We can see $e_k < 0$

$$\Rightarrow O(e^2_k) = O(e_k)$$
$$\Rightarrow e_{k+1} = e_k - e_k \frac{f'(x^*) + O(e^1_k)}{f'(x^*) + O(e_k)} = e_k \left(1 - \frac{f'(x^*) + O(e_k)}{f'(x^*) + O(e_k)}\right)$$
$$\Rightarrow e_{k+1} = e_k(1 - [1 + O(e_k)]) = e_k \cdot O(e_k) = O(e_k^2)$$

$\Rightarrow$ The Steffensen's method converges quadratically and it is a second order method



Exercise 4 Write and test a routine to compute $\arctan x$ for x in radians as follows. If $0 \leq x \leq 1.7 \times 10^{-9}$, set $\arctan x \approx x$. If $1.7 \times 10^{-9} < x \leq 2 \times 10^{-2}$, use the series approximation

$$\arctan x \approx x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7}.$$

Otherwise, set $y = x$, $a = 0$, and $b = 1$ if $0 \leq x \leq 1$; set $y = 1/x$, $a = \pi/2$, and $b = -1$ if $1 < x$. Then set $c = \pi/16$ and $d = \tan c$ if $0 \leq y \leq \sqrt{2} - 1$; and $c = 3\pi/16$ and $d = \tan c$ if $\sqrt{2} - 1 < y \leq 1$. Compute $u = (y - d)/(1 + d\,y)$ and the approximation

$$\arctan u \approx u \left(\frac{135135 + 171962.46u^2 + 52490.4832u^4 + 2218.1u^6}{135135 + 217007.46u^2 + 97799.3033u^4 + 10721.3745u^6} \right)$$

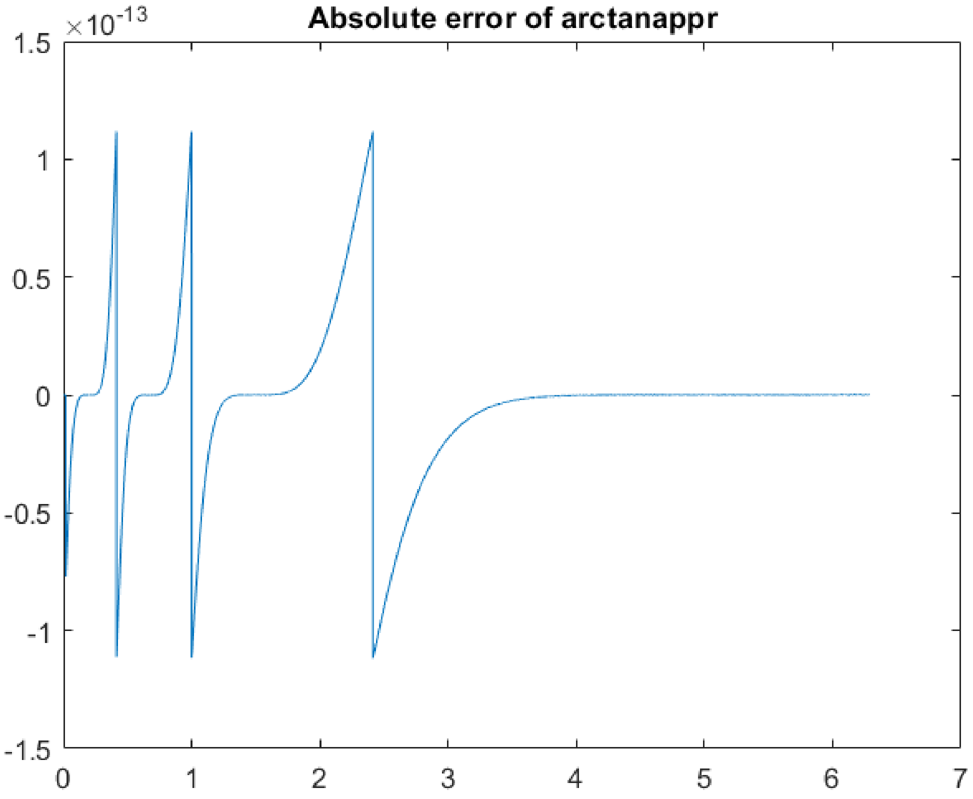
Finally, set $\arctan x \approx a + b(c + \arctan u)$.
Test the accuracy of your routine. Report both absolute and relative errors. Is this a useful implementation?
Note: This algorithm uses telescoped rational and Gaussian continued fractions.

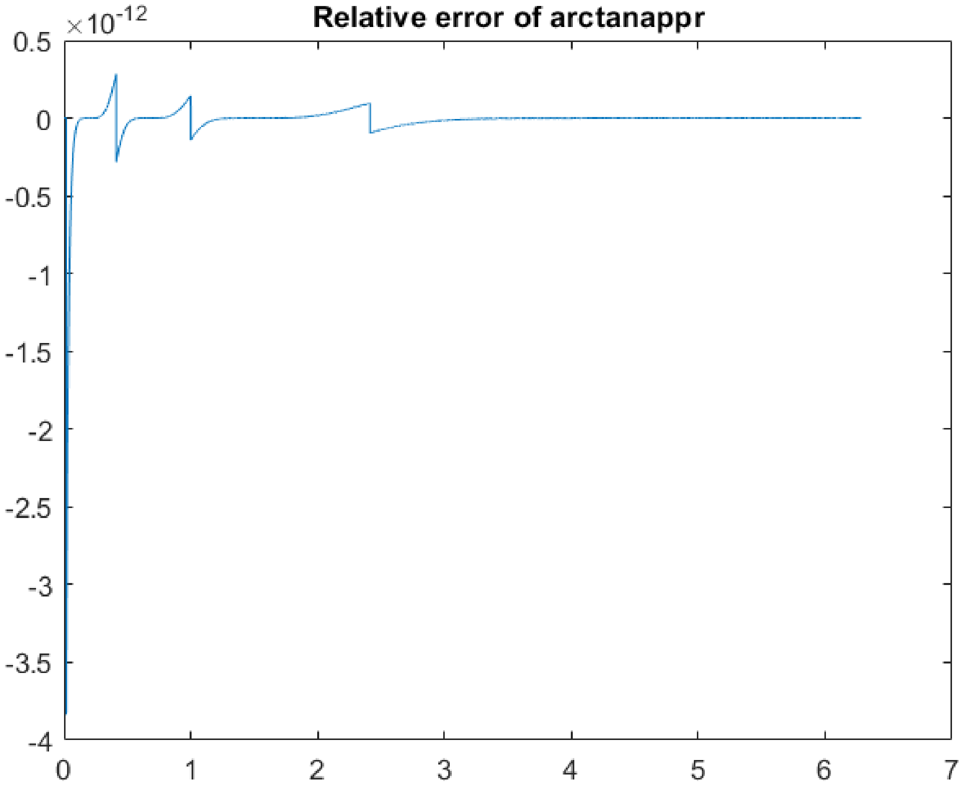
The Matlab code for arctanappr.m

```
function val = arctanappr(x)
    if x >= 0 && x <= 1.7e-9
        val = x;
    elseif x > 1.7e-9 && x <= 2e-2
        val = x - (x^3)/3 + (x^5)/5 - (x^7)/7;
    elseif x >= 0 && x <= 1
        y = x;
        a = 0;
        b = 1;
        if y >= 0 && y <= sqrt(2) - 1
            c = pi/16;
            d = tan(c);
        elseif y > sqrt(2) - 1 && y <= 1
            c = 3*pi/16;
            d = tan(c);
        end
        u = (y - d)/(1 + d * y);
        arctan_u = u * ((135135 + 171962.46*u^2 + 52490.4832*u^4 +
2218.1*u^6)/(135135 + 217007.46*u^2 + 97799.3033*u^4 +
10721.3745*u^6));
        val = a + b*(c + arctan_u);
    elseif x > 1
        y = 1/x;
        a = pi/2;
        b = -1;
        if y >= 0 && y <= sqrt(2) - 1
            c = pi/16;
            d = tan(c);
        elseif y > sqrt(2) - 1 && y <= 1
            c = 3*pi/16;
            d = tan(c);
        end
        u = (y - d)/(1 + d * y);
        arctan_u = u * ((135135 + 171962.46*u^2 + 52490.4832*u^4 +
2218.1*u^6)/(135135 + 217007.46*u^2 + 97799.3033*u^4 +
10721.3745*u^6));
        val = a + b*(c + arctan_u);
    end
end
```

The testing Matlab file for arctanappr_test.m

```
%{
If x is the exact value, x0 is the approximation value
=> Absolute error: Abs_e = x - x0
=> Relative error: Rel_e = Abs_e/x.
%}
clc;
t=linspace(0,2*pi,10000);
rt = atan(t);
for i=1:length(t)
    at(i)=arctanappr(t(i));
end
figure(1);
plot(t,rt-at)
title("Absolute error of arctanappr")
figure(2);
plot(t,(rt-at)./rt)
title("Relative error of arctanappr")
```





We can see that the absolute error of the arctan approximation method is prediodic in 3 cycles for the first pi. After the first pi, the error stabilizes and its very close to 0. The relative error also shows the same characteristic as the absolute error. Since the magnitudes of the erros are extremely small (10 to the power of 13), this approximation is a useful implementation