MATH 446/OR 481
Project 2

February 19, 2023

Part 1. The equation to be solved is

$$f(x) = x^n - a = 0$$

The derivative $f'(x)$ is

$$f'(x) = nx^{n-1}$$

The fixed point iteration function is thus

$$g(x) = x - f(x)/f'(x) = x - \frac{x^n - a}{nx^{n-1}}$$

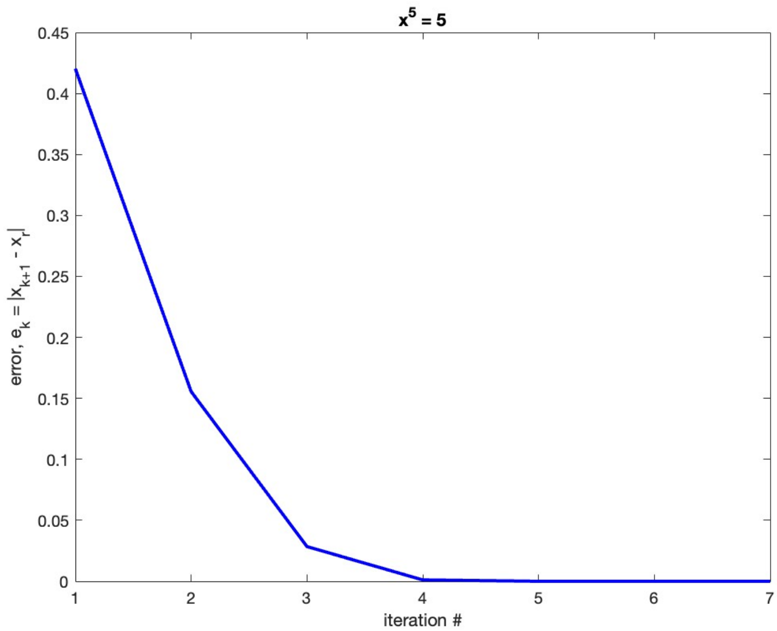
Simplifying,

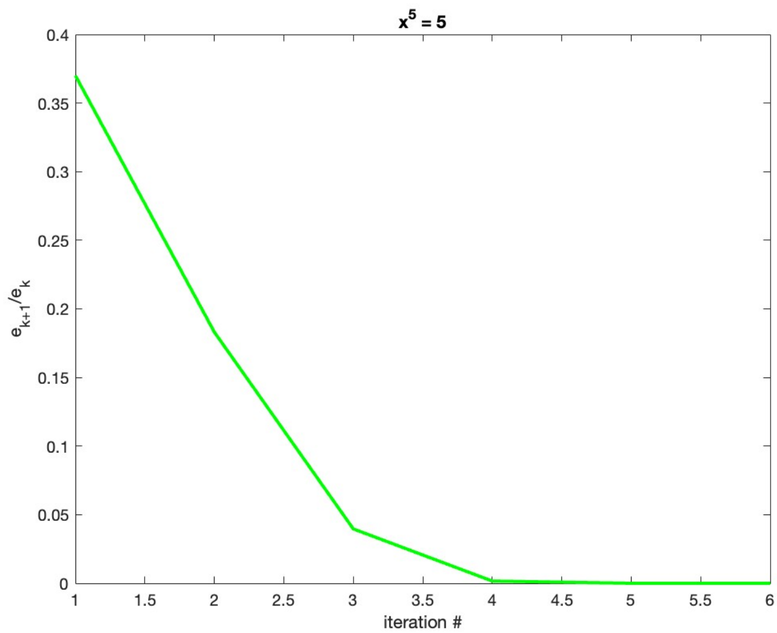
$$\boxed{g(x) = \left(\frac{n-1}{n}\right) x + \left(\frac{a}{n}\right) \frac{1}{x^{n-1}}}$$

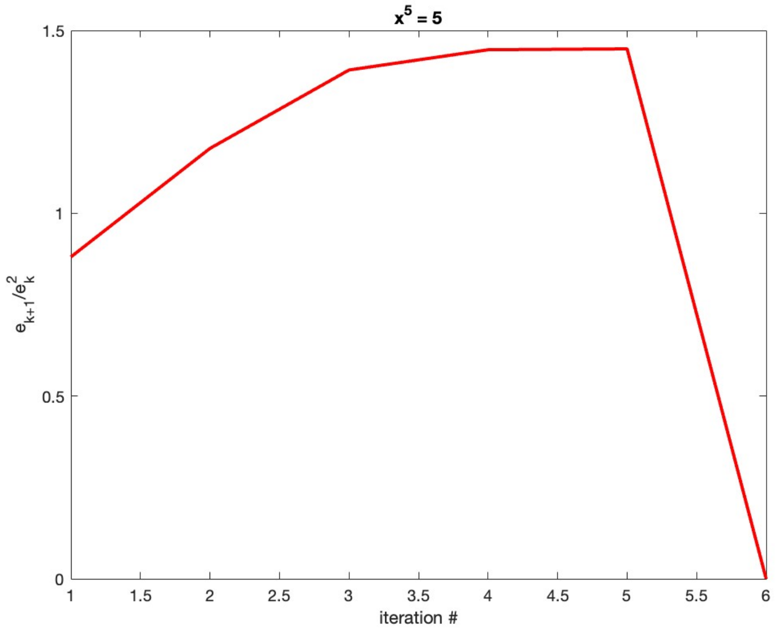
a. Computing $5^{1/5}$ From MATLAB,

$$\boxed{5^{1/5} = 1.3797296614612149}$$

The plot of errors and error ratios are shown below.



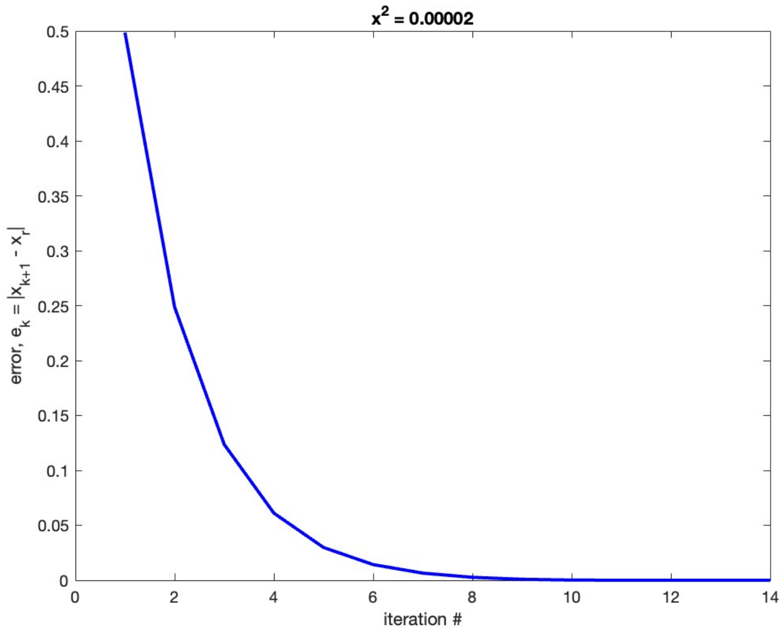


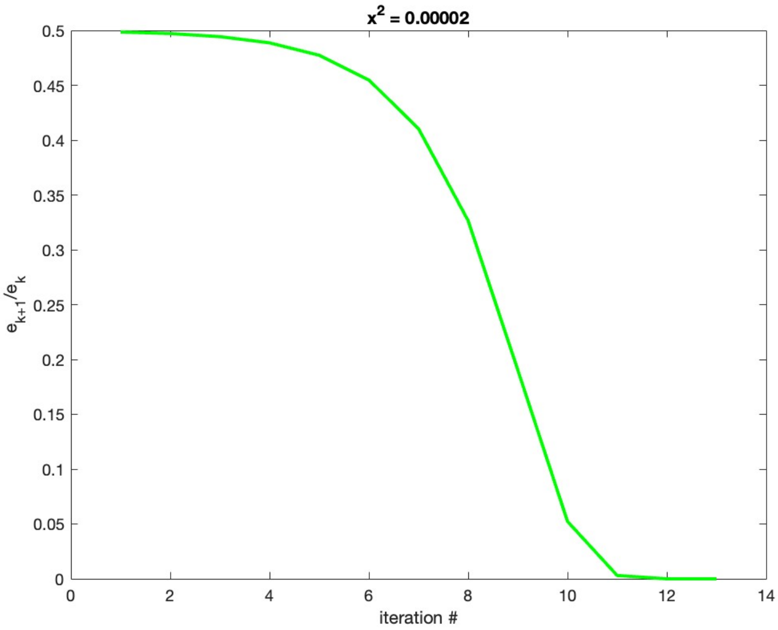


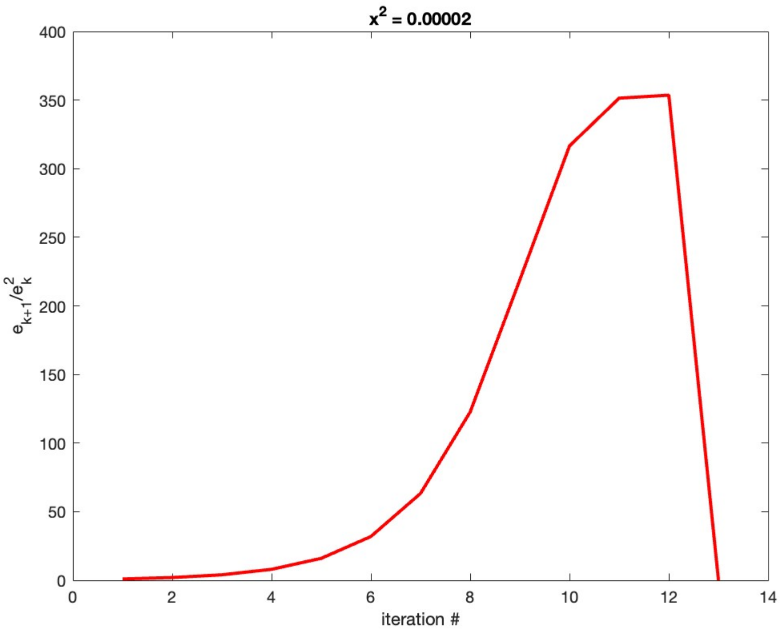
b. Computing $\sqrt{0.000002}$ From MATLAB,

$$\boxed{\sqrt{0.000002} = 0.0014142135623731}$$

The plot of errors and error ratios are shown below.





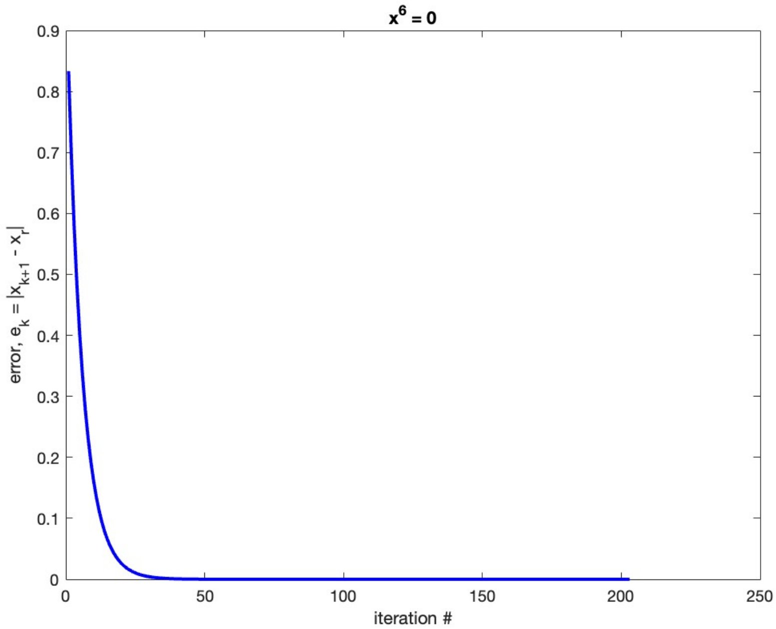


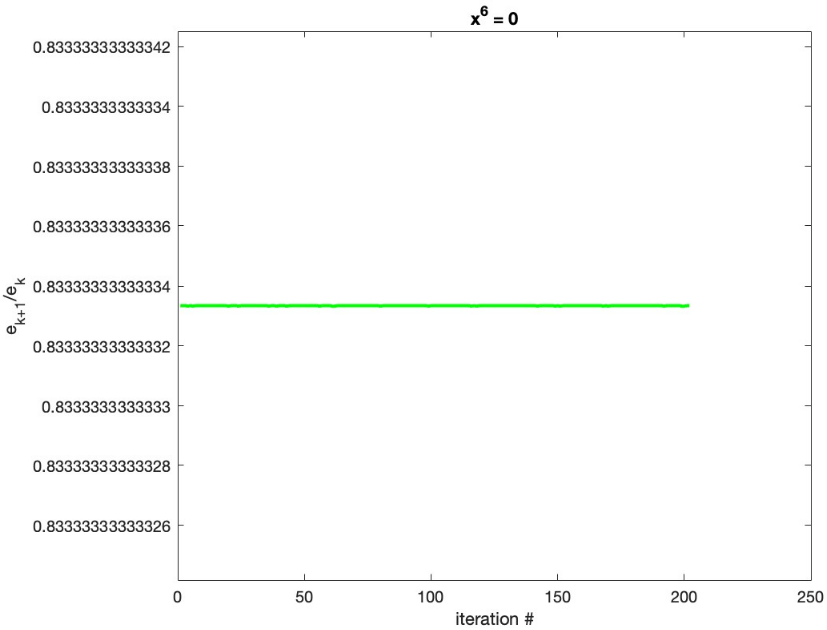
From the plots, we can see that the method has quadratic convergence.

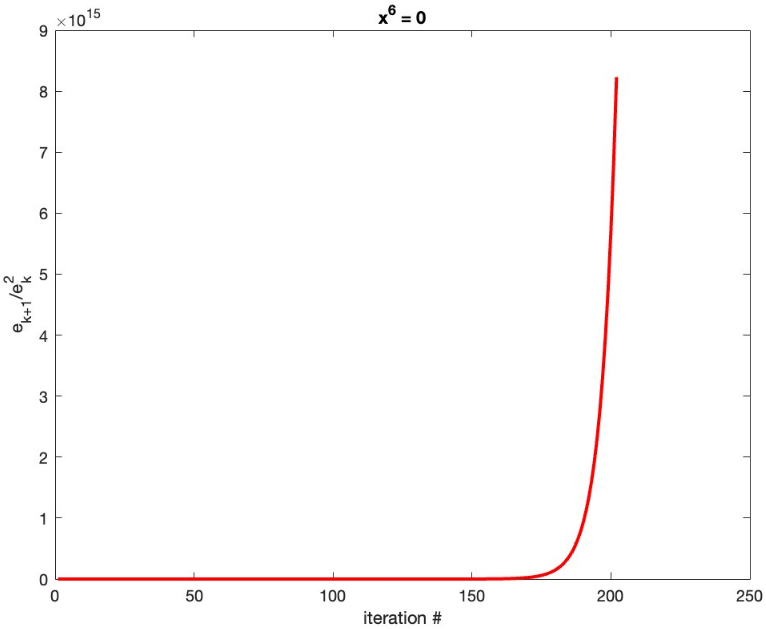
c. Computing $a^{1/n} with a = 0 and n = 6$ From MATLAB,

$$\boxed{0^{1/6} = 8.437369182026138e-17}$$

The plot of errors and error ratios are shown below.







From the plots, we can see, there is linear convergence, which is expected since 0 is a repeated root.

Part 2. We have

$$f(x) = xb - 1$$

Then,

$$f'(x) = b$$

The fixed point iteration function is then

$$g(x) = x - f(x)/f'(x) = x - \frac{xb - 1}{b}$$

Simplifying,

$$\boxed{g(x) = \frac{1}{b}}$$

This is a constant function and hence the fixed point iteration cannot be applied to it, as the iterates are all the same ($= 1/b$).

We now try

$$f(x) = \frac{1}{x} - b$$

Then,

$$f(x) = -\frac{1}{x^2}$$

and hence,

$$g(x) = x - f(x)/f'(x) = x + x^2\left(\frac{1}{x} - b\right)$$

Simplifying,

$$\boxed{g(x) = 2x - bx^2}$$

MATLAB Code - nthroot.m

```
function [root, errors] = nthroot(a, n)

% computes a^(1/n) to within 10^-9

initGuess = 1;
trueRoot = a^(1/n);
tol = 1e-16;
maxiter = 100;
errors = zeros(maxiter, 1);

% define fixed point iteration function
g = @(x) ((n-1)/n)*x + (a/n)/(x^(n-1));

% iterate
oldGuess = initGuess;
iter = 0;
while (true)
iter = iter + 1;

newGuess = g(oldGuess);
errors(iter) = abs(newGuess - trueRoot);

if errors(iter) < tol
break
else
oldGuess = newGuess;
end
end

root = newGuess;
errors = errors(1:iter);
end
```

MATLAB Code - part_1.m

```
clc; clear; close all

%% find 5^(1/5)

[root, errors] = nthroot(5, 5);
fprintf('5^(1/5) = %.16f\n', root)

% plot errors
figure()
plot(errors, 'b-', 'linewidth', 2)
xlabel('iteration #')
ylabel('error, e_k = |x_{k+1} - x_r|')
title('x^5 = 5')

figure()
e_k = errors(1:end-1);
e_kplus1 = errors(2:end);
plot(e_kplus1./e_k, 'g-', 'linewidth', 2)
xlabel('iteration #')
ylabel('e_{k+1}/e_k')
title('x^5 = 5')

figure()
plot(e_kplus1./(e_k.^2), 'r-', 'linewidth', 2)
xlabel('iteration #')
ylabel('e_{k+1}/e_k^2')
title('x^5 = 5')


%% find sqrt(0.000002)

[root, errors] = nthroot(0.000002, 2);
fprintf('0.000002^(1/2) = %.16f\n', root)

% plot errors
figure()
plot(errors, 'b-', 'linewidth', 2)
xlabel('iteration #')
ylabel('error, e_k = |x_{k+1} - x_r|')
```

```
title('x^2 = 0.00002')

figure()
e_k = errors(1:end-1);
e_kplus1 = errors(2:end);
plot(e_kplus1./e_k, 'g-', 'linewidth', 2)
xlabel('iteration #')
ylabel('e_{k+1}/e_k')
title('x^2 = 0.00002')

figure()
plot(e_kplus1./(e_k.^2), 'r-', 'linewidth', 2)
xlabel('iteration #')
ylabel('e_{k+1}/e_k^2')
title('x^2 = 0.00002')

%% find 0^(1/6)

[root, errors] = nthroot(0, 6);
fprintf('0^(1/6) = %.16f\n', root)

% plot errors
figure()
plot(errors, 'b-', 'linewidth', 2)
xlabel('iteration #')
ylabel('error, e_k = |x_{k+1} - x_r|')
title('x^6 = 0')

figure()
e_k = errors(1:end-1);
e_kplus1 = errors(2:end);
plot(e_kplus1./e_k, 'g-', 'linewidth', 2)
xlabel('iteration #')
ylabel('e_{k+1}/e_k')
title('x^6 = 0')

figure()
plot(e_kplus1./(e_k.^2), 'r-', 'linewidth', 2)
xlabel('iteration #')
ylabel('e_{k+1}/e_k^2')
title('x^6 = 0')
```