**2.2 Fixed-Point Iteration**

**Question 1**

> f = function(x) {x^4 + 2\*x^2 + x -3}

1. g1(x) = (3+x-2\*x^2) ^ (1/4)
2. g2(x) = ((x+3-x^4)/2) ^ (1/2)
3. g3(x) = ((x+3)/ (x^2 + 2)) ^ (1/2)
4. g4(x) = (3\*x^4+2\*x^2+3) / (4\*x^3+ 4\*x -1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 2**

A) Perform four iterations, if possible, on each of the functions *g* defined in question 1. Let P0 = 0, *Pn+1* = *g(Pn)* for n = 0,1,2,3

a.

P0 = 0

When n= 0:

P = P0 =0

When n= 1:

P = g1(P0=0)

So, P = = 1.3160740

When n= 2:

P = g1(P=1.3160740)

P = = 0.9607411

When n= 3:

P=g1(P=0.9607411)

p = = 1.2059016

Doing it with R code:

***f*** *= function(x) {****x^4 + 2\*x^2 - x -3****}*

> g1 = function(x){(3+x-2\*x^2) ^ (1/4)}

> fixedPoint (1, g1, n=4, iter=true)

p\_Old abs (P - p\_Old)

0 1.000000 0.00000000

1 1.189207 0.18920712

2 1.080058 0.10914936

3 1.149671 0.06961368

4 1.107821 0.04185090

b.

> g2 = function(x){((x+3-x^4)/2) ^ (1/2)}

> fixedPoint (1, g2, n=4, iter=true)

p\_Old abs (P - p\_Old)

0 1.0000000 0.0000000

1 1.2247449 0.2247449

2 0.9936662 0.2310787

3 1.2285686 0.2349025

4 0.9875064 0.2410622

c.

> g3 = function(x) {((x+3)/ (x^2 + 2)) ^ (1/2)}

> fixedPoint (1, g3, n=4, iter=true)

p\_Old abs (P - p\_Old)

0 1.000000 0.000000000

1 1.154701 0.154700538

2 1.116427 0.038273129

3 1.126052 0.009624823

4 1.123639 0.002413348

d.

> g4 = function(x) {(3\*x^4+2\*x^2+3) / (4\*x^3+ 4\*x -1)}

> fixedPoint (1, g4, n=4, iter=true)

p\_Old abs (P - p\_Old)

0 1.000000 0.000000e+00

1 1.142857 1.428571e-01

2 1.124482 1.837545e-02

3 1.124123 3.585261e-04

4 1.124123 1.342358e-07

B) Which function do you think gives the best approximation to the solution.

Clearly, g4 is converging faster. It already has an absolute error of less than 1. 124123 x by the fourth iteration!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 4**

**The following four methods are proposed to compute . Rank them in order, based on their apparent speed of convergence, assume P0 = 1.**

We have

The sequence produces the square root of a negative numbers in the second iteration so fails to converge right away. It leads to divergence pretty quickly. More than likely, the conditions of the fixed-point theorem have not been met.

This one quickly diverges as well. By iteration 6, it’s gone off to -

f2 = function(x){ x - (x^5 -7)/(x^2)}

fixedPoint (1, f2, n=5, iter=true)

P\_init abs(P - P\_init)

0 1.000000e+00 0.000000e+00

1 7.000000e+00 6.000000e+00

2 -3.358571e+02 3.428571e+02

3 3.788436e+07 3.788469e+07

4 -5.437256e+22 5.437256e+22

5 1.607457e+68 1.607457e+68

6 -Inf Inf

This one converges very nicely and quickly in only 7 iterations (if we’re looking for an error of less than10−7). Here are the iterations

f3 = function(x){ x - (x^5 -7)/(5\*(x^4))}

fixedPoint (1, f3, n=7, iter=true)

P\_init abs(P - P\_init)

0 1.000000 0.000000e+00

1 2.200000 1.200000e+00

2 1.819764 3.802363e-01

3 1.583475 2.362888e-01

4 1.489461 9.401386e-02

5 1.476022 1.343854e-02

6 1.475773 2.491904e-04

7 1.475773 8.418205e-08

This one does converge, but it is VERY, VERY, VERY slow. In order for it to converge with an error of less than 10−7, we need 500 iterations.

options (digits = 13)

f4 = function(x){ x - (x^5 -7)/12}

fixedPoint (1, f4, n=600, iter=true)

485 1.475773380678 4.434708913426e-07

486 1.475772947689 4.329895233468e-07

487 1.475773370445 4.227558929593e-07

488 1.475772957681 4.127641211049e-07

489 1.475773360689 4.030085143381e-07

490 1.475772967206 3.934834689190e-07

491 1.475773351389 3.841835565233e-07

492 1.475772976286 3.751034365340e-07

493 1.475773342524 3.662379328695e-07

494 1.475772984942 3.575819549351e-07

495 1.475773334072 3.491305680114e-07

496 1.475772993193 3.408789204240e-07

497 1.475773326016 3.328223068255e-07

498 1.475773001059 3.249561026930e-07

499 1.475773318335 3.172758220593e-07

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 5 (original question was changed)**

Use a fixed-point method to determine a solution accurate within 10-2 for function on [, ]. Use P0 =1

Solve for x:

The original domain [, ], when n =1, it located on [ , ]

Thus, the new function , where n=1 it has the range of ( , )

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 6**

If we choose then it fails to work.

If we choose , then we got this:

> g = function(x) { (x +1)^(1/3) }

> fixedPoint (1, g)

[1] 1.324718

> fixedPoint (1, g, iter = true)

p\_Old abs (P - p\_Old)

0 1.000000 0.000000e+00

1 1.259921 2.599210e-01

2 1.312294 5.237279e-02

3 1.322354 1.005998e-02

4 1.324269 1.914925e-03

5 1.324633 3.638807e-04

6 1.324702 6.912326e-05

7 1.324715 1.312993e-05

8 1.324717 2.493995e-06

9 1.324718 4.737265e-07

10 1.324718 8.998282e-08

With the root accurate to within 10-2, I choose P3 = 1.322354

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 10**

Use a fixed-point method to find an approximation to accurate within 10-4.

since

Then

> g= function(x){sqrt(25/x)}

> fixedPoint(1.5, g)

[1] 2.924018

> fixedPoint (1.5, g, iter = true)

p\_Old abs(P - p\_Old)

0 1.500000 0.000000e+00

1 4.082483 2.582483e+00

2 2.474616 1.607867e+00

3 3.178455 7.038392e-01

4 2.804542 3.739132e-01

5 2.985651 1.811088e-01

6 2.893680 9.197098e-02

7 2.939306 4.562579e-02

8 2.916404 2.290212e-02

9 2.927832 1.142867e-02

10 2.922112 5.719920e-03

11 2.924971 2.858562e-03

12 2.923541 1.429630e-03

13 2.924256 7.147278e-04

14 2.923899 3.573857e-04

15 2.924077 1.786874e-04

16 2.923988 8.934507e-05

17 2.924033 4.467219e-05

18 2.924010 2.233618e-05

19 2.924021 1.116807e-05

20 2.924016 5.584040e-06

21 2.924019 2.792019e-06

22 2.924017 1.396010e-06

23 2.924018 6.980048e-07

24 2.924018 3.490024e-07

25 2.924018 1.745012e-07

26 2.924018 8.725060e-08

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 11**

, determine an interval [a, b] on which fixed-point iteration will converge.

To find the value k, we use **Corollary 2.4 |** to estimate how many iterations it will take to converge. The first derivate of function g(x) is:

From , we can see that it’s always negative, which means function is decreasing.

Taking the 2nd derivative of function , we get:

Let , then we get , which is the critical value of .

Taking the 3rd derivative of function , we get:

Since , so is a local max for

We know , and

Then it follows that ,

So there exist a fixed point in [0, 1]

Next, the value of k is the maximum of | over the interval [0, 1]

By the **Extreme Value Theorem**, the maximum on [0, 1] will exist either at the critical point or the endpoints of the interval.

So

Thus,

According to Corollary 2.4, we get

Which means that,

If we choose initial guess P0 = , then the number of iterations to get the error within 10-5:

n > =

Find the approximation with R:

g = function(x){(2- exp(x) + x^2)/3}

> fixedPoint(1, g)

[1] 0.2575303

> fixedPoint(1, g, iter=true)

p\_Old abs (P - p\_Old)

0 1.00000000 0.000000e+00

1 0.09390606 9.060939e-01

2 0.30345393 2.095479e-01

3 0.24585170 5.760223e-02

4 0.26057771 1.472601e-02

5 0.25674040 3.837314e-03

6 0.25773538 9.949833e-04

7 0.25747706 2.583232e-04

8 0.25754410 6.704493e-05

9 0.25752670 1.740228e-05

10 0.25753122 4.516859e-06

11 0.25753004 1.172383e-06

12 0.25753035 3.042999e-07

13 0.25753027 7.898311e-08

We predicted n = 10 iterations and observed 9.

(a more refined value for the max of |g’(x)| would give a more accurate iteration count.)

Rewrite the function as,

Take the 1st derivative of we get,

The 1st derivative is always negative, which means the function is always decreasing.

Take the 2nd derivative of we get,

The 2nd derivative is always positive, which means the function is always increasing, and the graph of the function is concave up.

since

And function is always decreasing,

Then it follows that

It follows that a fixed point exits in [0, 1]

Next, the value of k can be determined.

By the **Extreme Value Theorem**, the maximum on [0, 1] will exist either at the critical point or the endpoints of the interval. So,

Since the , so the inequality does not hold on the interval [0, 1].

So, we need to find a tighter bound on the interval than [0,1].

By choosing a point closer to the fixed point and where

First, let| ,

Then solve for x,

It follows that we need to choose a point between 0.325 and a fixed point so we can estimate the k.

Let us try x = 0.44,

Since , then x = 0.44 is less than the fixed point.

**and**

It follows that a fixed point exits in [0.44, 1], so the new bound is on the interval [0.44, 1].

It follows that

According to Corollary 2.4, if we choose initial guess P0 = , then the number of iterations to get the error within 10-5 is at least:

n > =

We need at least 37 iterations to guarantee an error of 10-5.

Using R on the function results with only 28 being required.

Find the approximation with R:

g = function(x){ 6^(-x) }

> fixedPoint(0.45, g)

[1] 0.4480657

> fixedPoint(0.45, g, iter=true)

p\_Old abs(P - p\_Old)

0 0.4500000 0.000000e+00

1 0.4465108 3.489231e-03

2 0.4493110 2.800269e-03

3 0.4470623 2.248731e-03

4 0.4488672 1.804930e-03

5 0.4474179 1.449292e-03

6 0.4485813 1.163357e-03

7 0.4476472 9.340743e-04

8 0.4483971 7.498260e-04

9 0.4477950 6.020204e-04

10 0.4482783 4.832863e-04

11 0.4478903 3.880109e-04

12 0.4482018 3.114915e-04

13 0.4479517 2.500797e-04

14 0.4481525 2.007643e-04

15 0.4479913 1.611811e-04

16 0.4481207 1.293975e-04

17 0.4480168 1.038844e-04

18 0.4481002 8.339976e-05

19 0.4480333 6.695565e-05

20 0.4480870 5.375307e-05

21 0.4480439 4.315433e-05

22 0.4480785 3.464507e-05

23 0.4480507 2.781389e-05

24 0.4480730 2.232952e-05

25 0.4480551 1.792665e-05

26 0.4480695 1.439187e-05

27 0.4480579 1.155412e-05

28 0.4480672 9.275882e-06

29 0.4480598 7.446882e-06

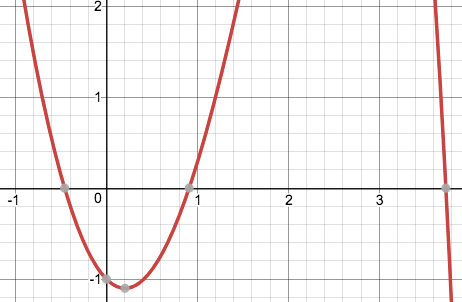
30 0.4480657 5.978511e-06

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 12**

For each of the following equations, determine a function **g** and an interval [a, b] on which fixed-point iteration will converge to a positive solution of the equation.

Using graphing calculator to graph the function



There are three different fixed point that can be found.

First, since

Make a choice for function g(x):

Let us choose P0 =3.5, using R code to find the solution:

g = function(x){ log(3\*x^2) } # **log** is natural log in R. For log base 10 using **log10** in R.

> fixedPoint(3.5, g)

[1] 3.733079

> fixedPoint(3.5, g, iter=true)

p\_Old abs (P - p\_Old)

0 3.500000 0.000000e+00

1 3.604138 1.041382e-01

2 3.662778 5.863945e-02

3 3.695056 3.227819e-02

4 3.712604 1.754777e-02

5 3.722079 9.475492e-03

6 3.727177 5.097997e-03

7 3.729915 2.737453e-03

8 3.731383 1.468376e-03

9 3.732170 7.871960e-04

10 3.732592 4.218881e-04

11 3.732818 2.260691e-04

12 3.732939 1.211288e-04

13 3.733004 6.489835e-05

14 3.733039 3.477034e-05

15 3.733058 1.862853e-05

16 3.733068 9.980330e-06

17 3.733073 5.346993e-06

18 3.733076 2.864662e-06

19 3.733077 1.534747e-06

20 3.733078 8.222425e-07

21 3.733079 4.405172e-07

22 3.733079 2.360075e-07

23 3.733079 1.264412e-07

24 3.733079 6.774098e-08

There is a fixed point on the interval [0, 1]

since

Make a choice for function g(x):

Let us choose P0 =0.5, using R code to find the solution:

g = function(x){ sqrt(exp(x)/3) }

> fixedPoint (0.5, g)

[1] 0.9100075

> fixedPoint (0.5, g, iter=true)

p\_Old abs (P - p\_Old)

0 0.5000000 0.000000e+00

1 0.7413324 2.413324e-01

2 0.8364070 9.507459e-02

3 0.8771277 4.072073e-02

4 0.8951694 1.804169e-02

5 0.9032811 8.111716e-03

6 0.9069522 3.671019e-03

7 0.9086184 1.666248e-03

8 0.9093757 7.573073e-04

9 0.9097201 3.444037e-04

10 0.9098768 1.566690e-04

11 0.9099481 7.127751e-05

12 0.9099805 3.243000e-05

13 0.9099953 1.475545e-05

14 0.9100020 6.713720e-06

15 0.9100050 3.054754e-06

16 0.9100064 1.389922e-06

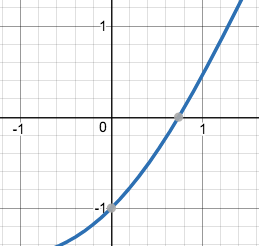
17 0.9100070 6.324192e-07

18 0.9100073 2.877530e-07

19 0.9100075 1.309287e-07

20 0.9100075 5.957304e-08

Using graphing calculator to graph the function



Since

Then a fixed point exits in the interval [0, 1]

Let and the initial guess P0 = 0.5

g = function(x){ cos(x) }

> fixedPoint (0.6, g)

[1] 0.7390842

Warning message:

In fixedPoint (0.6, g): Failed to converge after 30 iterations

> fixedPoint (0.6, g, iter=true)

p\_Old abs (P - p\_Old)

0 0.6000000 0.000000e+00

1 0.8253356 2.253356e-01

2 0.6783104 1.470252e-01

3 0.7786340 1.003236e-01

4 0.7118735 6.676047e-02

5 0.7571393 4.526576e-02

6 0.7268038 3.033547e-02

7 0.7473020 2.049818e-02

8 0.7335253 1.377677e-02

9 0.7428189 9.293640e-03

10 0.7365649 6.254011e-03

11 0.7407805 4.215574e-03

12 0.7379421 2.838374e-03

13 0.7398546 1.912541e-03

14 0.7385666 1.288047e-03

15 0.7394343 8.677631e-04

16 0.7388499 5.844813e-04

17 0.7392436 3.937383e-04

18 0.7389784 2.652156e-04

19 0.7391570 1.786575e-04

20 0.7390367 1.203436e-04

21 0.7391178 8.106591e-05

22 0.7390632 5.460650e-05

23 0.7390999 3.678381e-05

24 0.7390752 2.477792e-05

25 0.7390919 1.669075e-05

26 0.7390806 1.124307e-05

27 0.7390882 7.573475e-06

28 0.7390831 5.101580e-06

29 0.7390865 3.436487e-06

30 0.7390842 2.314858e-06

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 14**

Use a fixed-point iteration method to determine a solution accurate to within 10-4 for

We know the **domain** is [4, 5], and the period is

Rewrite the function

To make sure our **range** is also in [4, 5], let’s solving for x inside of *tan* field

Since , then it follows that

When n = -1, we have the correct function with the proper range.

We choose , the initial guess P0 = 4

g = function(x){ atan(x)+ pi} # **atan()** is inverse tangent in R. **pi** is in R

> fixedPoint(4, g)

[1] 4.493409

> fixedPoint(4, g, iter=true)

p\_Old abs (P - p\_Old)

0 4.000000 0.000000e+00

1 4.467410 4.674103e-01

2 4.492176 2.476543e-02

3 4.493351 1.175477e-03

4 4.493407 5.548650e-05

5 4.493409 2.618467e-06

6 4.493409 1.235667e-07

7 4.493409 5.831168e-09

The solution accurate to within 10-4 P4 = 4.493407

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ---------------------- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 17**

1. Use Theorem 2.3 to show that the sequence defined by

Converges to whenever > .

* First, we need to find choose a and b, then show function g is continuous on the interval
* Second, we need to find k such that

(if we can, then it converges to a fixed-point p according to the theorem.)

Define:

Let . Further,

since the interval is [, ), then we know that g**’**(x)>0, which means that g is increasing always.

By the Extreme Value Theorem (EVT), the smallest value that g will assume is at the endpoint, .

Sincethen it follows that g(x)∈ [, ).

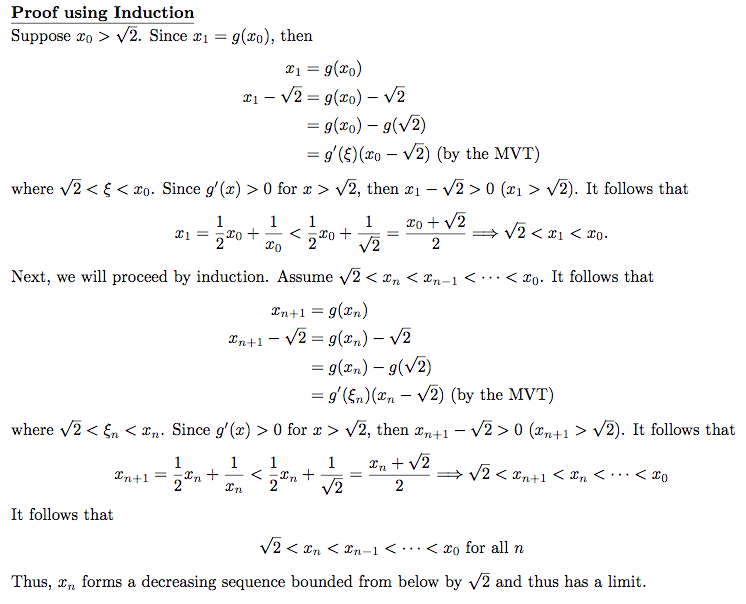
Further, so it follows that k

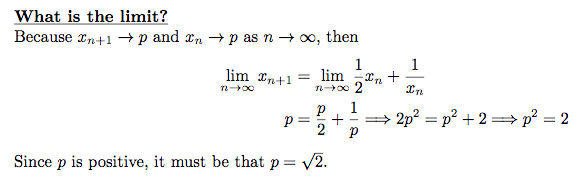
Since our initial guess then it follows that converges to the fixed-point p.

1. **Use the fact that whenever to show that if , then**

Suppose , it follows that

So,





**C. Use the results of parts of (a) and (b) to show that the sequence in (a) converges to , whenever**

\*