AMSC 460 - Computational Methods

Table of Contents

HOMEWORK 1 - Problem 1

Yizhan Ao

Let f(x) = e^x + x^2 _ 5x.

(a) The bracket [1.5, 2] contains a root. Explain why using the Intermediate Value Theorem. For this bracket, estimate the number of iterations N that would be needed to compute the root to an accuracy of 10^-4.

```
clear all
syms f(x) x y

f(x) = exp(x) + x^2 - 5*x;
double(f(1.5))

ans =
   -0.7683

double(f(2))

ans =
    1.3891

%Since f(x) is continuous on [1.5, 2], and f(1.5) = -0.7683 < 0 and
 f(2) = 1.3891 > 0.
%By the Intermediate Theorem, f(x) has at least one root on the
 interval [1.5, 2].
```

(b) The bracket given in (a) contains a root, but there is another root. Find a bracket for it. Then use the bisection method to find the two roots to an accuracy of 10^-4.

```
A = double(f(0));
fprintf(['Because f(0)= %g > 0 f(1.5)= %g < 0. So find a root in', ...
' [0, 1.5]'], A ,double(f(1.5)))

Because f(0)= 1 > 0 f(1.5)= -0.768311 < 0. So find a root in [0, 1.5]

bisect(@(x) exp(x) + x^2 - 5*x, 0, 1, 0.0001)
```

```
ans =
    0.2805

bisect(@(x) exp(x) + x^2 - 5*x, 1.5, 2, 0.0001)

ans =
    1.7339
```

So roots are at x = 0.2805 and at x = 1.7339.

HOMEWORK 1 - Problem 2

Consider the cubic f(x) = x^3 _ x _ 1. (a) Use the MATLAB command fzero to find a root in the interval [1, 2].

```
fzero(@(x) x^3 - x - 1,1)

ans =
    1.3247
```

(b) Show that f(x) = 0 can be rewritten as a fixed point problem for both the functions (i) g1(x) = x^3 _ 1, and (ii) g2(x)=(1+x)^(1/3).

```
g1= x^3 - 1;
g2=(1+x)^(1/3);
g1' == diff(g1)
g2' == diff(g2)
%Let f(x) = x^3 # x # 1 = 0, then we can have x^3 # x # 1 = 0 then x^3
 # 1 = x
%So we can write g1(x) = x^3 # 1.
%x^3 # x # 1 = 0
%x^3 = 1 + x
%x = (1 + x)^(1/3)
%Then g2(x) = (1 + x)^(1/3).

ans =
conj(x)^3 - 1 == 3*x^2
ans =
conj((x + 1)^(1/3)) == 1/(3*(x + 1)^(2/3))
```

(c) Which of the functions g1 and g2 is a contraction mapping near the root r from part (a)? Which of g1 or g2 will be successful in making the iteration xi+1 = g(xi) converge locally to the root r?

```
diff((1 + x)^(1/3))

ans =
1/(3*(x + 1)^(2/3))

%Derative of g1 and g2, g1'= 3 * x^2 and g2'= 1/(3*(x + 1)^(2/3)).
%Since g1'(1) = 3 and g1'(2) = 12 and g1' is strictly increasing on
 [1,2]. So g1' diverges.
%g2' is continuous and decreasing on [1,2], g2'(1) = 0.21 and g2'(2) =
 0.1602
```

```
%So # L,  0 # L < 1 s.t |g2'(x)| # L < 1   # x # [1,2], g2'converges.
%Thus by the Contraction Mapping Theorem only g2 is a constraction on
 [1,2],
%g2 will be making the iteration xi+1 = g(xi) converge locally to the
 root r.
```

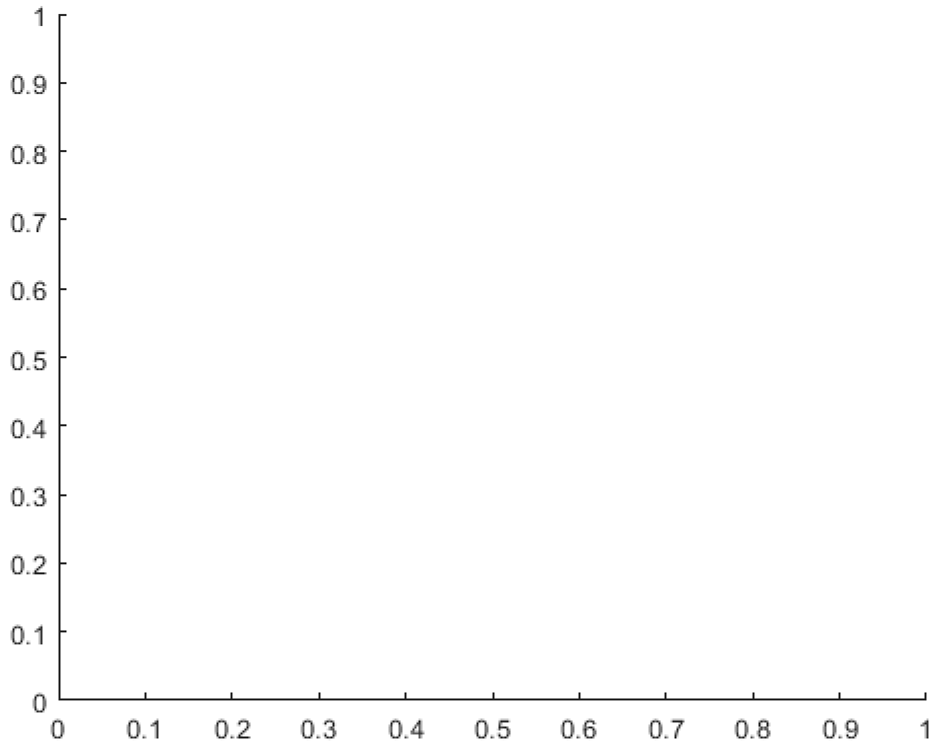
(d) Write a script or function in MATLAB to carry out 10 steps of the fixed point iteration for both g1 and g2, each starting with the guess x0 = 0. What approximate root does your algorithm give for g1? For g2?. Are your results consistent with the analysis from part (c)?

This is the code from the fpi root function[root,tol] = fpi_root(g,x0,N) fprintf('fixed point method g(x)') disp(g) for count= 1:N xx =g(x0); x0=xx; rr(count) = xx; fprintf('\tAfter %d steps, root = %f\n', count, x0) end root = rr(count); tol=abs(g(root)) hold off

```
g1 = @(x) x.^3 - 1;
g2 = @(x) (1 + x).^(1/3);

disp('Fixed point iteration for g1 starts with x0=0:')
fpi_root(g1,0,10)
```

```
Fixed point iteration for g1 starts with x0=0:
fixed point method g(x)    @(x)x.^3-1
 After 1 steps, root = -1.000000
 After 2 steps, root = -2.000000
 After 3 steps, root = -9.000000
 After 4 steps, root = -730.000000
 After 5 steps, root = -389017001.000000
 After 6 steps, root = -58871587162270591457689600.000000
 After 7 steps, root =
 -204040901322752646989478259680513109526757826056202557355691431285390611316736.0
 After 8 steps, root =
 -84947714722373876912426115385994721993330450340708886432958705831500286122585831
 After 9 steps, root = -Inf
 After 10 steps, root = -Inf
tol =
   Inf
ans =
  -Inf
```




```
disp('Fixed point iteration for g2 starts with x0=0:')
fpi_root(g2,0,10)
```

```
Fixed point iteration for g2 starts with x0=0:
fixed point method g(x)    @(x)(1+x).^(1/3)
 After 1 steps, root = 1.000000
 After 2 steps, root = 1.259921
 After 3 steps, root = 1.312294
 After 4 steps, root = 1.322354
 After 5 steps, root = 1.324269
 After 6 steps, root = 1.324633
 After 7 steps, root = 1.324702
 After 8 steps, root = 1.324715
 After 9 steps, root = 1.324717
 After 10 steps, root = 1.324718
tol =
    1.3247
ans =
    1.3247
```

```
%Only g2 is successful in making the iteration xi+1 = g(xi) converge
 locally to the root r.
%The results consistent with the analysis from part (c).
```

Published with MATLAB® R2020b