Math128a - Project1

February 26, 2020

[ ]:
```
3033230423                                                       Minkwan Kim
```

[ ]:
```
P1. Function findzero(f, a, b, tol)
```

[ ]:
```
function p = findzero(f, a, b, tol)
w = 1;
count = 0;
#to calculate at most 100 times
p = a + ((w*f(a)*(a-b))/(f(b)-w*f(a)));
#If this p value satisfy the condition, the progress finishes.
fprintf('a b p f(p)\n')
fprintf('%d %d %d %f\n', a, b, p, f(p));
while abs(b-a) >= tol && abs(f(p)) >= tol && count < 100
    #All conditions.
    if f(p)*f(b) > 0
        w = (1/2);
    else
        w = 1;
        a = b;
    end
    b = p;
    p = a + ((w*f(a)*(a-b))/(f(b)-w*f(a)));
    fprintf('%d \t %d \t %d \t %f\n', a, b, p, f(p));
    count = count + 1;
end
end
```

```
--------------------------------------------------------------
```

[ ]:
```
P2. Test the function by findzero(cos(x)-x, 0, 1, 10^(-10))
```

[ ]:
```
f = @(x) cos(x)-x;
p = findzero(f, 0, 1, 10^(-10))


a              b              p              f(p)
0              1              6.850734e-01  0.089299
1              6.850734e-01  7.362990e-01  0.004660
1              7.362990e-01  7.415391e-01  -0.004109
7.362990e-01  7.415391e-01  7.390836e-01  0.000003
7.415391e-01  7.390836e-01  7.390851e-01  0.000000
7.415391e-01  7.390851e-01  7.390851e-01  -0.000000
7.390851e-01  7.390851e-01  7.390851e-01  0.000000
p = 0.739085133215161

cos(p)
ans = 0.739085133215161
```

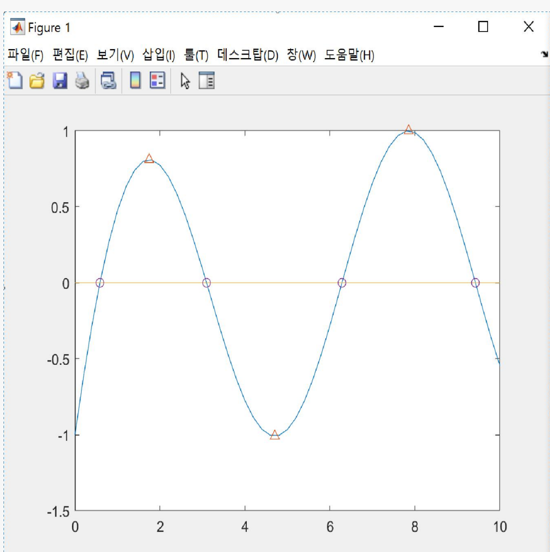
[ ]:
```
#comment


Not surprisingly, we aim to reduce f(p). Each time through the loop, we see␣
 ↪that f (p) decreases,
which is the process of changing points and recalculating to find the optimal␣
 ↪point close to the solution
within the first [a, b] interval. By changing the value of b to the value of␣
 ↪the previous p, a more optimized interval can be set,
and if the sign of f (p) and f (b) is different, by setting (a) to (b), the␣
 ↪target interval can be narrowed.









------------------------------------------------------------------------------
```

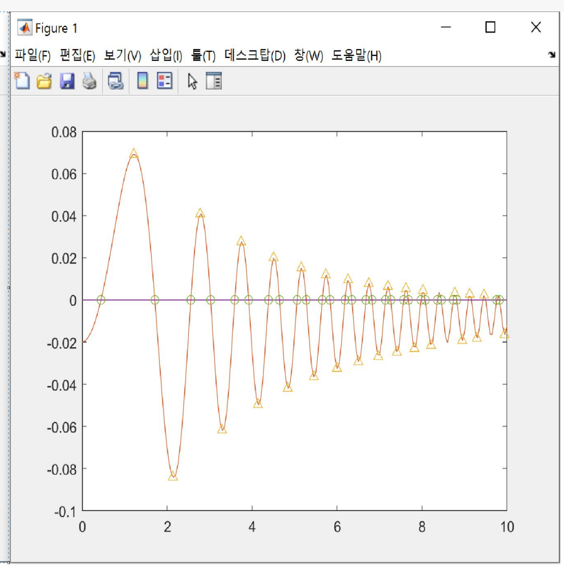
[ ]:
```
P3. Function findmanyzeros(f, a, b, n, tol)
```

[ ]:
```
function p = findmanyzeros(f, a, b, n, tol)
p = [];
#Make p empty space
count = 1;
#to make a index order
for i = 0 : (n-1)
    #i from 0 to n-1
    f_i = a + (i*(b-a)/n);
    s_i = ((i+1)*(b-a)/n);
    #the first and second x values to focus on
    #the last values are (a+((n-1)*(b-a)/n)) and (b)
    #x has 51 values [a, b]
    if abs(f(f_i) + f(s_i)) ~= (abs(f(f_i)) + abs(f(s_i)))
    #To filter out f(f_i) and f(s_i) has different sign(one negative, one
    #positive)
    p(count) = findzero(f, f_i, s_i, tol);
    #To stack all values f(p) = 0
    count = count + 1;
    end
end
```

[ ]:
```
P4. Test the function by findmanyzeros(sin(x) - e^(-x), 0, 10, 50, 10^(-10))
and findmanyzeros(f2, 0, 10, 50, 10^(-10))
```

[ ]:
```
----------------------------------------------------
```





```
f = @(x) sin(x) - exp(-x);
p1 = findmanyzeros(f, 0, 10, 50, 10^(-10));
p1 = 0.588532744041087   3.096363932410559   6.285049273390893   9.
 ↪424697254737579

df = @(x) cos(x) + exp(-x);
p2 = findmanyzeros(df, 0, 10, 50, 10^(-10));
p2 = 1.746139530434118   4.703323759452176   7.854369686580018


x = linspace(0, 10, 51);
y = sin(x) - exp(-x);
plot(x, y)
dx = p2;
dy = sin(dx) - exp(-dx);
hold on
plot(dx, dy, '^')
x1 = linspace(0, 10, 51);
y1 = x1 - x1;
plot(x1, y1)
dx1 = p1;
dy1 = dx1 - dx1;
plot(dx1, dy1, 'o')
hold off

-------------
```

[ ]:
```
f = @(x) sin(x^2)/(10+x^2) - (1/50)*exp(-x/10);
p1 = findmanyzeros(f, 0, 10, 50, 10^(-10));
p1 = 0.443110789251605   1.709385822474302   2.557782495860945   3.
 ↪022905099683073   3.590516586869940
3.918946782269551   4.385913256726604   4.645747749903903   5.057295018102749 ␣
 ↪5.273681028403169
5.649120496083582   5.834699427043790   6.184321249881802   6.346532707752022 ␣
 ↪6.676564162573747
6.820233458663715   7.134772634763719   7.263243688300312   7.565153865182300 ␣
 ↪7.680857912370610
7.972232067120999   8.077002073441022   8.359425442934908   8.454682904760533 ␣
 ↪8.729391358369847
8.816263654880965   9.755171766503366   9.821728751045532

df = @(x) (2*cos(x^2)*(10+x^2)-2*x*sin(x^2))/(10+x^2)^2 + (1/500)*exp(-x/10);
p2 = findmanyzeros(df, 0, 10, 50, 10^(-10));
p2 = 1.214679103911531   2.134281815988549   2.776997396884052   3.
 ↪289996442540363   3.741499741640262
```

```
4.136378351105458   4.504644599169932   4.837120329071166   5.156079936538756   ␣
↪5.448475934968741
5.733911404038826   5.997796213288747   6.258545523478870   6.500820672335095   ␣
↪6.742415843377305
6.967586926134158   7.193759933256041   7.404959579185324   7.618367817743434   ␣
↪7.817877968338688
8.020492640395814   8.210036356184405   8.769517212984869   8.942870158681284   ␣
↪9.120959776489643
9.287612762526338   9.459336782142138   9.941275480555698


x = linspace(0, 10, 201);
y = [];
for i = 0 : 200
    t = (i*0.05);
    y(i+1) = sin(t^2)/(10+(t^2)) - (1/50)*exp(-t/10);
    end
plot(x, y)
dx = p2;
dy = 0;
for i = 1 : length(p2)
    q = p2(i);
    dy(i) = sin((q^2))/(10+(q^2)) - (1/50)*exp(-q/10);
    end
hold on
plot(dx, dy, '^')
x1 = linspace(0, 10, 51);
y1 = x1 - x1;
plot(x1, y1)
dx1 = p1;
dy1 = dx1 - dx1;
plot(dx1, dy1, 'o')
hold off
```

[ ]:
```
#comment
For the completeness of the graph, I used linspace(0, 10, 201), not linspace(0,␣
↪10, 51) in graph 2.
As the result, we can see in the graph solutions that couldn't be calculated␣
↪when n was 50 and where the slope becomes zero.
In the first graph, we can see that the graph and the calculation result match,
but in the second graph, we can see that they do not coincide in some intervals.
I think that such discrepancy can be cleared by setting n high.
```