Homework 3

Math 3607, Summer 2021

Spenser Smith

Table of Contents

Problem 1.

This problem asks for a script that verifies that the number that follows 8 is 8+8eps by computing 8+4eps and 8+4.01eps. Since 8+4eps=8 this means that it clearly does not follow 8. Since 8+4.01eps=8+8eps this means that 8+8eps must be the number immediately following the number 8.

```
format long

ans1=8+eps*4;
fprintf('8 + 4*eps = %.24f \n',ans1)
```

```
8 + 4*eps = 8.000000000000000000000000
```

```
ans2=8+eps*4.01;
fprintf('8 + 4.01*eps = %.24f \n',ans2)
```

```
8 + 4.01*eps = 8.000000000000001776356839
```

```
ans3=8+8*eps;
fprintf('8 + 8*eps = %.24f \n',ans3)
```

```
8 + 8*eps = 8.000000000000001776356839
```

```


ans5=16-4*eps; %does not give previous number
fprintf('16 - 4*eps = %.24f \n',ans5)
```

```
16 - 4*eps = 16.000000000000000000000000
```

```
ans4=16-4.01*eps; %number right before 16
fprintf('16 - 4.01*eps = %.24f \n',ans4)
```

```
16 - 4.01*eps = 15.999999999999998223643161
```

```


ans7=1024-256*eps; %does not give previous number
fprintf('1024 - 256*eps = %.24f \n',ans7)
```

```
1024 - 256*eps = 1024.000000000000000000000000
```

```
ans6=1024-256.01*eps; %gives previous number
```

```
fprintf('1024 - 256.01*eps = %.24f \n',ans6)
```

```
1024 - 256.01*eps = 1023.999999999999886313162278
```

```
ans8=1024-512*eps; %number right before 1024
fprintf('1024 - 512*eps = %.24f \n',ans8)
```

```
1024 - 512*eps = 1023.999999999999886313162278
```

Problem 2.

This problem asks for a script that approximates a function f(x) for three different expressions.

Part A:

TS of $\log(x+1)$: $x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots$

Then $\frac{\log(1+x)}{x} = \frac{x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots}{x}$

$$= 1 - \frac{x}{2} + \frac{x^2}{3} - \ldots$$

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

Part B:

```
for k=1:1:20
    x=10^(-k);
    if x <= 1e-16
        f1=1;
    else
        f1=(log(x+1)) ./ x;
    end

    f2=(log(x+1)) ./ ((1+x)-1);
    f3=log1p(x) ./ x;
    fprintf('%.24f %.24f %.24f %.24f \n',x,f1,f2,f3)
end
```

```
0.100000000000000005551115 0.953101798043249348602046 0.953101798043248460423627 0.953101798043248571445929
0.010000000000000000208167 0.995033085316809229325941 0.995033085316808341147521 0.995033085316808341147521
0.001000000000000000020817 0.999500333083423231350650 0.999500333083533254452391 0.999500333083533143430088
0.000100000000000000004792 0.999950003332973125225180 0.999950003333083370371526 0.999950003333083370371526
0.000009999999999999999124 0.999995000039884174292126 0.999995000033332970268418 0.999995000033333081290721
0.000000999999999999999955 0.999999499918066803161310 0.999999500000333330085311 0.999999500000333441107614
0.000000099999999999999995 0.999999950583870478304505 0.999999950000003301475715 0.999999950000003412498017
0.000000010000000000000000 0.999999988922529059465205 0.999999995000000030387355 0.999999995000000030387355
0.000000001000000000000000 1.000000082240370957720188 0.999999999499999958629815 0.999999999499999958629815
0.000000000100000000000000 1.000000082690370994953355 0.999999999949999995862981 0.999999999949999995862981
0.000000000010000000000000 1.000000082735370776632067 0.999999999994999888563996 0.999999999994999888563996
0.000000000001000000000000 1.000088900581840967163316 0.999999999999500066572011 0.999999999999499955549709
0.000000000000100000000000 0.999200722162590926345160 0.999999999999949928941589 0.999999999999950039963892
0.000000000000010000000000 0.999200722162635890377658 0.999999999999994892974087 0.999999999999995003996389
0.000000000000001000000000 1.110223024625155874289817 0.999999999999999444888488 0.999999999999999444888488
0.000000000000000100000000 1.000000000000000000000000 NaN 1.000000000000000000000000
0.000000000000000010000000 1.000000000000000000000000 NaN 1.000000000000000000000000
0.000000000000000001000000 1.000000000000000000000000 NaN 1.000000000000000000000000
0.000000000000000000100000 1.000000000000000000000000 NaN 1.000000000000000000000000
0.000000000000000000010000 1.000000000000000000000000 NaN 1.000000000000000000000000
```

Problem 3.

This problem asks for a script that compares the accuracy of approximations to the actual values.

```
format long g
t=-4:-4:-16;
x=cosh(t);

%Part A
kf= (x.*(1./sqrt(x.^2-1))) ./ acosh(x);
fprintf('The condition values are: \n')
```

```
The condition values are:
```

```
disp(kf)
```

```
  Columns 1 through 3

       0.250167787600421         0.125000028133797        0.0833333333396252

  Column 4

      0.0625000000000016
```

```
%Part B
t2=log(x-sqrt(x.^2-1));
abs_err1=abs(t2-t);
rel_err1=abs_err1 ./ t;
fprintf('The absolute error values for the star function are: \n')
```

```
The absolute error values for the star function are:
```

```
disp(abs_err1)
```

```
  Columns 1 through 3

     4.61852778244065e-14      1.71089808986835e-10      1.37072186490172e-07

  Column 4

       0.0013751287983812
```

```
fprintf('The relative error values for the star function are: \n')
```

```
The relative error values for the star function are:
```

```
disp(rel_err1)
```

```
  Columns 1 through 3

    -1.15463194561016e-14     -2.13862261233544e-11     -1.14226822075144e-08

  Column 4

    -8.59455498988249e-05
```

```
%Part C
t3=-2*log(sqrt((x+1)./2)+sqrt((x-1)./2));
abs_err2=abs(t3-t);
rel_err2=abs_err2 ./ t;
fprintf('The absolute error values for the cross function are: \n')
```

```
The absolute error values for the cross function are:
```

```
disp(abs_err2)
```

```
     0     0     0     0
```

```
fprintf('The relative error values for the cross function are: \n')
```

```
The relative error values for the cross function are:
```

```
disp(rel_err2)
```

```
     0     0     0     0
```

```
%Part D
%The second one is unstable as it's too good of an approx. since the error
%is 0.
```