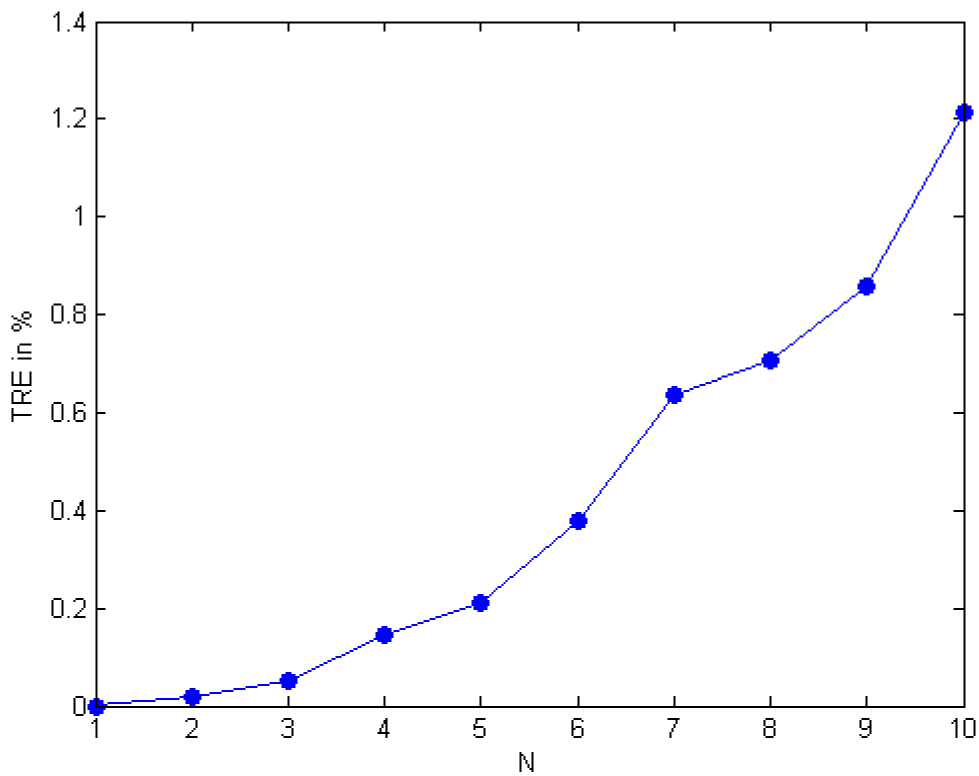
Prob 1 Solution and Matlab example #1 (prepared by HPH)

Many have obtained the solution for this problem by hand, which is perfectly fine. With the toy calculator, one can "keep 3 digits" by chopping or rounding. Either is acceptable. This first example uses chopping. Matlab examples #2 and #3 use rounding and chopping, respectively.

```
A = 0.873; B = A; power = [1:10]; TRE(1) = 0;
for ipower = 2:10
    Btrue = A^ipower;
    B = B*A;
    B = (B*1000000-mod(B*1000000,1000))/1000000;
    TRE(ipower) = 100*abs(B - Btrue)/abs(Btrue);
end
plot(power,TRE,'-o','MarkerFaceColor','b');xlabel('N');ylabel('TRE in %')
```



Prob 1 Matlab example #2 (Thanks to Andrew Shabilla)
This program considers "rounding" instead of "chopping".

1. Matlab code:

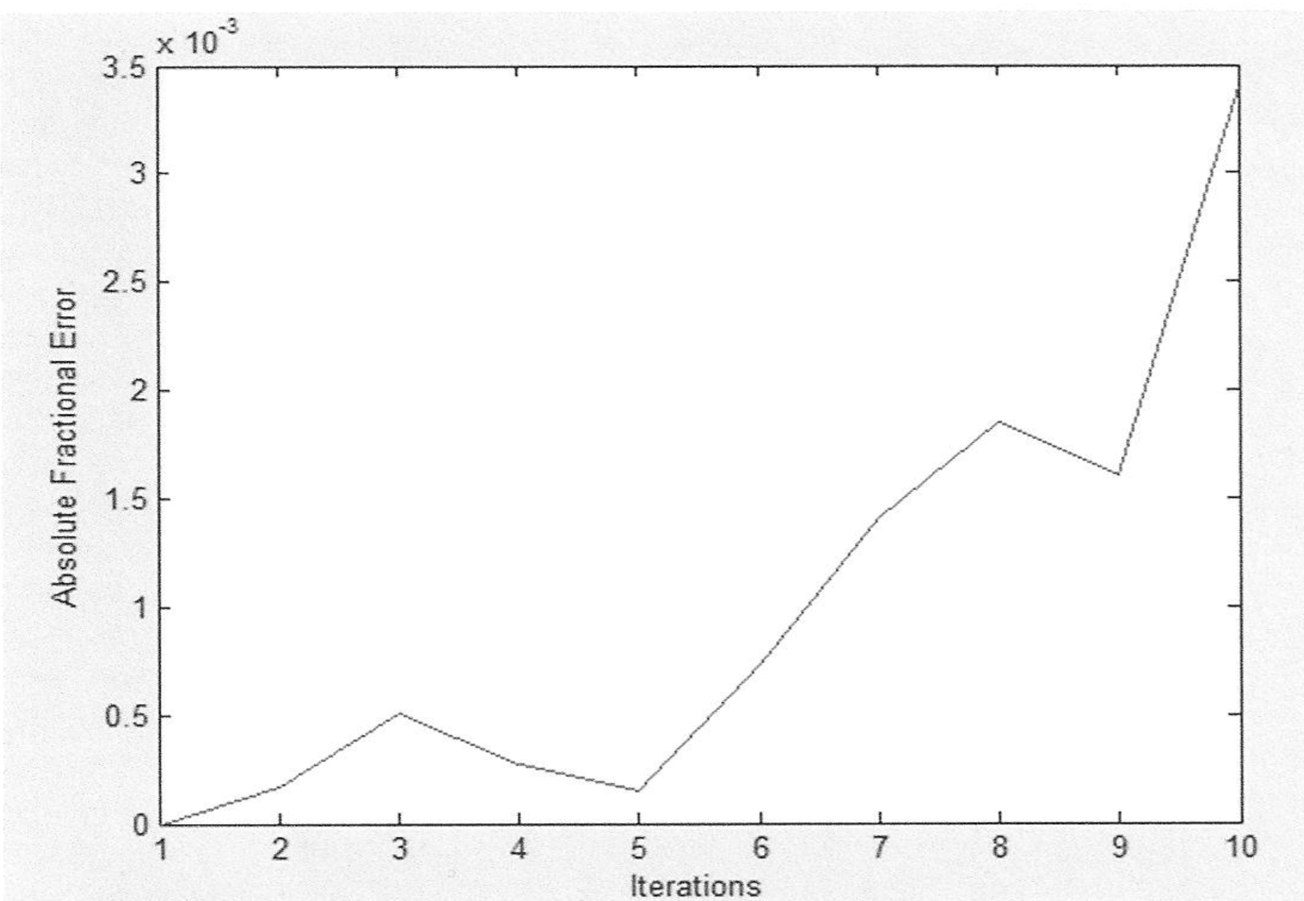
Problem Script

```
clear all
clc
a = .873;
atrue = a;
x=[1:10];
for exp = [1:9]
    a(exp + 1) = a(1)*a(exp);
    atrue(exp + 1) = atrue(1)*atrue(exp);
    a(exp + 1) = Trunc(a(exp + 1),3);
%    n(exp) = exp;
end
a
atrue
error = abs((atrue - a)./(atrue));
plot(x,error)
ylabel('Absolute Fractional Error');
xlabel('Iterations');
```

Trunc Function

```
function [ ntrunc ] = Trunc( n,decs)
%Rounds a number to the specified number of decimal places

nint = n*(10^decs);
nint2 = round(nint);
ntrunc = nint2*(10^-decs);
end
```



Prob 1 Matlab example #3 (Thanks to Audrey Nash)
This program uses "chopping". It produces a plot (omitted here) similar to that in Example #1.

```
% Problem 1: Matlab Code

% toy calculator values
A=0.873;
A2=1;
c=1;

for i=1:10
        AE=A*A2;
        AE=floor(AE*10^3)/(10^3);
        A2=AE;
        y(i)=A2;
end
y

% true values
for t=1:10
    ATV(t)=A^t;
end
ATV

% error

for n=1:10
    error(n)=(ATV(n)-y(n))/(ATV(n));
end
error

plot(error)
xlabel('exponent');
ylabel('true relative error');
```

Prob 2 Solution (prepared by HPH)

Since the initial interval has a length of 2, we need to repeatedly half it until its length falls below 0.04 in order to guarantee (without knowing what the true solution is) that the true error is bounded by ± 0.02. This requires the minimum number of iteration, N, to satisfy $2/2^N < 0.04$, which leads to N = 6. After performing bisection 6 times, the final interval is [1.1875, 1.21875] and the solution is 1.203125 ± 0.015625.

Since the problem did not clarify how the "uncertainty" is defined, those who used other measures of numerical error and obtained a slightly different answer also receive full credit. For example, if one considers the criterion, $|f(x_N)| < 0.02$, $x_N = 1.1875$ may be a satisfactory answer already.

Prob 3   Solution and Matlab example #1 (prepared by HPH)

(a) The two solutions within 0 < x < 5 are 0.45173 and 3.30565. See Matlab program.

(b) Most of the initial guesses within $0 < x_0 < 5$ converge to either 0.45173 or 3.30565 as the final solution. There are exceptions: (i) Near the location where df(x)/dx = 0, the process can converge to a solution very far away from the interval [0,5]. This is because the initial "tangent" is almost horizontal, therefore it intersects with the zero line at a very large value of x. An example is the case with $x_0 = 1.75$ which converges to 216.9712. (ii) If an initial guess is located at near the boundary of two major domains within which the initial guesses converge to 0.45173 and 3.30565, it might converge to neither of them. An example is $x_0 = 4.4$ which converges to 15.90932. The attached Matlab program seeks the solutions given the initial guesses $x_0 = 0, 0.05, 0.1, ..., 4.95, 5.0$, with 1000 iterations for each initial guess. Note that for $x_0 = 4.7$, it takes 245 iterations for the process to converge to the common solution of 0.45173. If only a small number of iterations are performed, one would obtain a very large value for that case.

```
f = inline('exp(-x)-sin(x)-0.2','x');
fprime = inline('-exp(-x)-cos(x)','x');
for k = 1:100
    x0 = (k-1)*0.05;
    x00 = x0;
    for iter = 1:1000
        x1 = x0 - f(x0)/fprime(x0);
        x0 = x1;
    end
    fprintf('initial guess = %8.5f   solution = %8.5f \r',x00,x1)
end
```

Results:

```
initial guess =  0.00000   solution =  0.45173
initial guess =  0.05000   solution =  0.45173
initial guess =  0.10000   solution =  0.45173
initial guess =  0.15000   solution =  0.45173
initial guess =  0.20000   solution =  0.45173
initial guess =  0.25000   solution =  0.45173
initial guess =  0.30000   solution =  0.45173
initial guess =  0.35000   solution =  0.45173
initial guess =  0.40000   solution =  0.45173
initial guess =  0.45000   solution =  0.45173
initial guess =  0.50000   solution =  0.45173
initial guess =  0.55000   solution =  0.45173
initial guess =  0.60000   solution =  0.45173
initial guess =  0.65000   solution =  0.45173
initial guess =  0.70000   solution =  0.45173
initial guess =  0.75000   solution =  0.45173
initial guess =  0.80000   solution =  0.45173
initial guess =  0.85000   solution =  0.45173
initial guess =  0.90000   solution =  0.45173
initial guess =  0.95000   solution =  0.45173
initial guess =  1.00000   solution =  0.45173
initial guess =  1.05000   solution =  0.45173
initial guess =  1.10000   solution =  0.45173
initial guess =  1.15000   solution =  0.45173
initial guess =  1.20000   solution =  0.45173
initial guess =  1.25000   solution =  0.45173
initial guess =  1.30000   solution =  0.45173
initial guess =  1.35000   solution =  0.45173
```

(continue to next page)

```
initial guess =  1.40000   solution =  0.45173
initial guess =  1.45000   solution =  0.45173
initial guess =  1.50000   solution =  0.45173
initial guess =  1.55000   solution =  0.45173
initial guess =  1.60000   solution =  0.45173
initial guess =  1.65000   solution =  0.45173
initial guess =  1.70000   solution =  0.45173
initial guess =  1.75000   solution = 216.97125
initial guess =  1.80000   solution = 18.64820
initial guess =  1.85000   solution =  9.62607
initial guess =  1.90000   solution =  3.30565
initial guess =  1.95000   solution =  6.08415
initial guess =  2.00000   solution =  6.08415
initial guess =  2.05000   solution =  9.62607
initial guess =  2.10000   solution =  0.45173
initial guess =  2.15000   solution =  3.30565
initial guess =  2.20000   solution =  3.30565
initial guess =  2.25000   solution =  3.30565
initial guess =  2.30000   solution =  3.30565
initial guess =  2.35000   solution =  3.30565
initial guess =  2.40000   solution =  3.30565
initial guess =  2.45000   solution =  3.30565
initial guess =  2.50000   solution =  3.30565
initial guess =  2.55000   solution =  3.30565
initial guess =  2.60000   solution =  3.30565
initial guess =  2.65000   solution =  3.30565
initial guess =  2.70000   solution =  3.30565
initial guess =  2.75000   solution =  3.30565
initial guess =  2.80000   solution =  3.30565
initial guess =  2.85000   solution =  3.30565
initial guess =  2.90000   solution =  3.30565
initial guess =  2.95000   solution =  3.30565
initial guess =  3.00000   solution =  3.30565
initial guess =  3.05000   solution =  3.30565
initial guess =  3.10000   solution =  3.30565
initial guess =  3.15000   solution =  3.30565
initial guess =  3.20000   solution =  3.30565
initial guess =  3.25000   solution =  3.30565
initial guess =  3.30000   solution =  3.30565
initial guess =  3.35000   solution =  3.30565
initial guess =  3.40000   solution =  3.30565
initial guess =  3.45000   solution =  3.30565
initial guess =  3.50000   solution =  3.30565
initial guess =  3.55000   solution =  3.30565
initial guess =  3.60000   solution =  3.30565
initial guess =  3.65000   solution =  3.30565
initial guess =  3.70000   solution =  3.30565
initial guess =  3.75000   solution =  3.30565
initial guess =  3.80000   solution =  3.30565
initial guess =  3.85000   solution =  3.30565
initial guess =  3.90000   solution =  3.30565
initial guess =  3.95000   solution =  3.30565
initial guess =  4.00000   solution =  3.30565
initial guess =  4.05000   solution =  3.30565
initial guess =  4.10000   solution =  3.30565
initial guess =  4.15000   solution =  3.30565
initial guess =  4.20000   solution =  3.30565
initial guess =  4.25000   solution =  3.30565
initial guess =  4.30000   solution =  3.30565
initial guess =  4.35000   solution =  3.30565
initial guess =  4.40000   solution = 15.90932
initial guess =  4.45000   solution =  0.45173
initial guess =  4.50000   solution =  0.45173
```

(continue to next page)

```
initial guess =  4.55000   solution =  0.45173
initial guess =  4.60000   solution =  0.45173
initial guess =  4.65000   solution =  0.45173
initial guess =  4.70000   solution =  0.45173
initial guess =  4.75000   solution = 22.19251
initial guess =  4.80000   solution = 12.36502
initial guess =  4.85000   solution =  9.62607
initial guess =  4.90000   solution =  9.62607
initial guess =  4.95000   solution = 12.36502
```

Prob 3 Matlab example #2 (Thanks to Vincent Bevilacqua)

This example shows a nice use of the "while" loop.

```
Problem 3

A)

Editor: HW1_Pr3.m

initial_guess = [0.5,1.75,2.5,3.0,4.7];
for i = 1:5
    x0 = initial_guess(i);
    xn = x0 - (exp(-x0) - sin(x0) - 0.2)/(-exp(-x0)-cos(x0));
    uncertainty = abs(xn - x0);
    while uncertainty >= 0.01
        xn1 = xn;
        xn = xn1 - (exp(-xn1) - sin(xn1) - 0.2)/(-exp(-xn1)-cos(xn1));
        %xn = xn-1 - f(xn-1)/f'(xn-1)
        uncertainty = abs(xn - xn1);
    end
    display(strcat('x0=',num2str(initial_guess(i))));
    display(strcat('xn = ',num2str(xn,15)));
    display(strcat('|xn - xn-1| = ',num2str(uncertainty,15)));
    display(' ');
end


Command Window:

>> HW1_Pr3
x0=0.5
xn =0.451731930852224
|xn - xn-1| =0.000848721829151466

x0=1.75
xn =216.971250951186
|xn - xn-1| =0.000813117246565298

x0=2.5
xn =3.30565066324214
|xn - xn-1| =0.00509313103259812

x0=3
xn =3.30565113681156
|xn - xn-1| =0.0042098052164623

x0=4.7
xn =0.451732170122645
|xn - xn-1| =0.00018743807322219
```