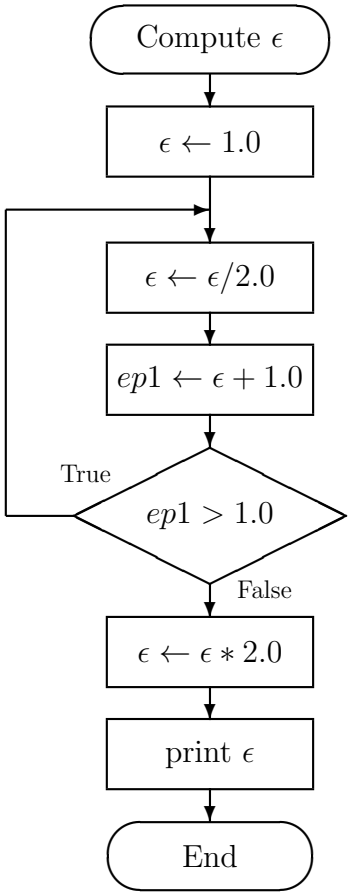
MECH2700 Engineering Analysis I, 2015
Exercise Sheet 3

Algorithms Revisited

1. When considering floating-point values, the machine precision is related to the number of digits stored in the fractional part of the value. One way of estimating the machine precision is to find the smallest value ϵ that can be added to the value 1.0 such that $1.0+\epsilon$ can be distinguished from 1.0. Implement the algorithm in the flow chart below as a Python program using "float" numbers and compute the value of ϵ. How many bits are in the fractional part of a float number?



   For the post-tested loop, you may use a "while 1: ..." loop and break on an appropriate condition. Paste your code into your workbook. State the value that your code calculates for ϵ and estimate the number of bits in the fractional part of the of a number of type "float". Hint: Consider log_2(error) or count the number of iterations.

2. Use the following pseudo-code description to write a program to compute the cube root of a given number N which is in the range zero to one. Before writing the program, check that the algorithm should work (by hand) and adjust it if necessary.

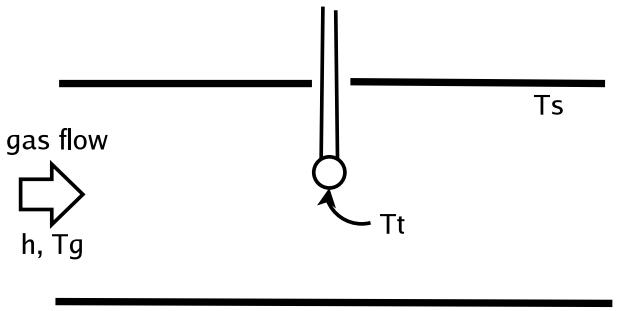
```
Begin with N given;

   Set the trial value x = 0.1;
   Set the stepsize dx = 0.1;

   Loop while dx > 1.0e-6
      If x cubed is less than N then
         increment x by dx;
      Else
         decrement x by dx;
         decrease the magnitude of dx by a factor of 10;
      End If
   End Loop

   print x as the (approximate) cube root of N;
End
```

Set up a table in your workbook and work through the algorithm for ten iterations of the algorithm for $N = 0.6$ and a starting guess of 0.1 as given above. Write python code to solve this for N = 0.6, use a starting guess of 0.1 for the cube root. How many iterations does this code require to converge and what is the cube root of 0.6.

Solve the same problem this time using Newton's method (Look back to your entries for Exercise Sheet 2). Paste your code and any rough working into your workbook. Make sure to count the number of iterations.

How many iterations did the Newton's method take to obtain a solution and how does this compare to that generated by the algorithm above and your calculator? Compare the performance of the algorithms.

3. A thermometer is placed in a flow of gas in a large duct in order to measure the temperature of the gas. The temperature indicated by the thermometer is not necessarily the same as that of the gas but is determined by the overall energy balance on the sensing element.



We will label the temperature of the gas as T_g, the effective radiation temperature of the duct wall as T_S and the temperature indicated by the thermometer as T_t. Assuming that $T_g > T_s$, energy will be transferred to the thermometer by convection and then dissipated to its surroundings by radiation. The energy balance can be expressed as

$$hA(T_g - T_t) = \sigma A \epsilon (T_t^4 - T_s^4)$$

where A is the surface area of the sensing element, $h = 8.3\,\mathrm{W/m^2{\cdot}K}$ is the convection heat transfer coefficient, $\epsilon = 0.9$ is the emissivity of the sensing element, and $\sigma = 5.669 \times 10^{-8}\,\mathrm{W/m^2 \cdot K^4}$ is the Stefan-Boltzmann constant.

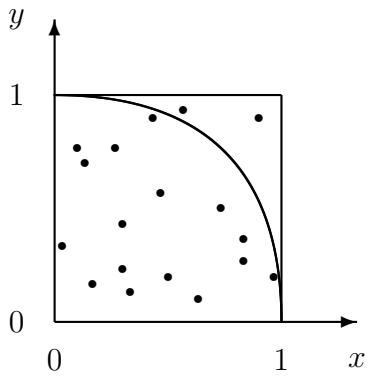
If the walls of the duct are at $T_s = 278\,\mathrm{K}$ and the gas temperature is specified as $T_g = 302\,\mathrm{K}$, our goal is to determine the temperature indicated by the thermometer.

(a) Rewrite the energy balance equation in the form $f(x) = 0$.

(b) Write a Python program that solves this equation using the bisection method.

(c) What is the error in the temperature measurement?

4. Monte Carlo simulation is a numerical technique based on random sampling. We shall use the method to estimate the area of a quarter circle (as shown in the figure) and thus compute an estimate for π.



The approach is to generate N randomly placed sample points (x, y), with each coordinate in the range 0.0 ... 1.0, and to count the number of points that lie within the unit circle. The probability that a point will lie within the quarter circle is the ratio of the areas, $\pi/4$.

Write a Python program to implement this method and use it to make estimates of π for N= 50,000, 100,000, ... 5,000,000. Plot the magnitude of error between the estimate of π and the true value as a function of the number of sample points N. Use logarithmic scales for both axes. From the trend shown on the graph, can you estimate how many sample points will be required to get an approximation for π that is accurate to 6 decimal digits? Compare the computer time required for this method with that required for Machin's formula from Exercise Sheet 1.

To allow convenient plotting of the results, accumulate the intermediate results (at the values of N specified above) in lists that can be turned into plotted data sets.

In your workbook, detail all of the parts of solving the problem including:

- How to generate a random number between 0 and 1 in python
- How to determine whether your a point in the unit square lies inside in quarter circle of radius 1
- Your code
- Plot of the error estimates
- Answer any of the questions in the problem description