CS319: Scientific Computing (with MATLAB)

CS319 Lab 1: Numbers and Programming

25 Jan, 2023

Goal: to gain familiarity with the concepts of

- basic MATLAB program structure;
- input and output,
- Computer representation of integers and floats.
- Also: some *flow-of-control* (`if`) and *loops* (`for`)

Upload your answer for Q2, and MATLAB script for Q4 by 5pm Tuesday, 31 Jan. See the `Labs ... Lab 1` section of Blackboard.

In this lab, we'll write some matlab **scripts**. So far, all the sample programmes presented in class have been scripts. These are files with names that a letter, and can contain letters, digits, or underscores, and end with "dot m". No other symbols, such as `-` (minus) can be used, not can they being with anything other than an letter. Download and run any script from Week 3.

Q1. (`if/elseif/else`-statements) Write a MATLAB script that prompts the user to enter two integers, x, y, and then reports which in quadrant the point (x, y) is found, or if (x, y) is on an axis (i.e., one or both are zero. (Tip: see `https://en.wikipedia.org/wiki/Quadrant_(plane_geometry)` for a definition of quadrants *I*, *II*, *III* and *IV*.

Q2. The following script finds the largest `int8` that is correctly representable by your computer. It is very similar to an example from Week 2, except that it also computes the time taken. (Full code at Q2_MyIntMax.m).

Q2_MyIntMax.m

```
%% CS319: Lab 1, Q2 (see Eg01_MyIntMax.m from Week 2)
% Who: Niall Madden
% What: Checks the largest integer of type int18
% When: Jan 2023
clear; % clear any previously defined variables
fprintf('\n--------------------------\n');
fprintf('CS319, Lab 1, Q3 \n');

tic; % Start the clock
a=int8(0); % Set a to zero in int8
b=a+1;
while( b>a )
   a=a+1;
   b=a+1;
end
fprintf('Largest int8=%d\n', a);
TimeTaken = toc; % Seconds since "tic"
fprintf('Computional took %f seconds.\n', TimeTaken);
```

Q2(a) Read the code carefully, and make sure you understand it. Do the results agree with the theory covered in class?

Q2(b) There are other types of integers available in MATLAB, for example, `uint8`, `int16` and `uint16`. Modify the script to check that largest values of these. Do you get the expected results?

Q2(c) Suppose you wanted to use this program to test the largest `int32` your MATLAB programs can represent. Estimate how many **seconds** your program would take to run. If it is not too long, compare with the actual time taken (you should find your estimate is a little pessimistic).

Q2(d) Suppose you wanted to use this program to test the largest `int64` your MATLAB programs can represent. Again, estimate how many **years(!)** your program would take to run.

Submit your answers to these questions in essay form (plain text).

Q3. Recall that `double` is the default data type in MATLAB. Write a script to tries to compute the smallest `double` greater than zero that your computer can represent. For example, you could initialise `x=1.0`, and `y=x/2`. Then, for as long as MATLAB thinks that $y > 0$, divide both x and y by 2. Eventually, when y evaluates as 0 (zero), x should be a good approximation of the smallest double representable.

Does the answer given by your code agree with theory? If not, can you give a reason why?

Q4. Next we want to compute the largest `double` representable. This is a little more tricky; where as small floats are eventually rounded to zero (which is a number), large ones tend to infinity (which is not, although MATLAB uses `inf` to represent numbers that are too big). Try a similar approach as in Question 3, but doubling x and y at each step, and using the function `isfinite` to test if y is finite or not.

Submit your solution to Q4 as a MATLAB script, uploaded to Blackboard.