Numerical Analysis

Math 370 Fall 2002 MWF 9:30am - 10:25pm
 Fowler 127

Worksheet 7: **Friday September 20**

SUMMARY Continuation of MATLAB programming; Machine Precision Algorithm

CURRENT READING Recktenwald pp. 209-211

Warm-up
Write down (in your own words) the meaning of the following terms:
ALGORITHM :

PSEUDOCODE :

The **Machine Precision** is the number ϵ_m which makes the following statement on a computer to be TRUE:

$$1 + \epsilon_m = 1$$

Consider the following algorithm to compute ϵ_m, the machine precision:

```
LET epsilon = 1
LET COUNTER = 0
LET MAXCOUNTER = 100
WHILE COUNTER < MAXCOUNTER
   LET B = 1 + EPSILON
   IF (B EQUALS 1) QUIT PROGRAM
   LET EPSILON = EPSILON/2  % halve epsilon each iteration
   LET COUNTER = COUNTER + 1 % update thecounter
END WHILE
OUTPUT (EPSILON, COUNTER)
```

Exercise
Find the machine precision of your calculator.

Implications
When designing an algorithm one should NOT USE the logical construct **Are** x **and** y **equal?** but instead **Are** x **and** y **close?** or **Is** $x - y$ **small enough**?

Here is how the "MACHINE PRECISION" ALGORITHM would be implemented in MATLAB (found in `Q:\mfiles\math370\myeps.m`

```
epsilon = 1;
it = 0;
maxit = 100;
while it < maxit,
   b = 1 + epsilon;
   if b == 1 break; end
   epsilon = epsilon/2;
   it = it + 1;
end
fprintf('epsilon = %12.8e in %d steps',epsilon,it);
```

NOTE the machine precision ϵ_m for MATLAB is found in the command `eps`.
Therefor how many bits is MATLAB using to store floating point numbers?

Therefor how many bits is **your calculator** using to store floating point numbers?

What would the output of the following MATLAB code be? (Example found on page 211 of Recktenwald)

```
x = tan(pi/6);
y = sin(pi/6)/cos(pi/6);
if x==y
   fprintf('x and y are equal\n');
else
   fprintf('x and y are not equal: x - y = %e\n');
end
```

Go to MATLAB and check your answer!