Amelia Rotondo
CWID: 887925113

Homework: Chapter 1

5 33.

At the bottom of the `floatgui.m` document I wrote in the following lines:

```
- total_floats = 2^(t+1) * (emax - emin + 1);
- text(.9*xmax,2,num2str(total_floats));
```

Adding the formula for the total_floats allows us to see the total amount of available floating points generated within the parameters of the floatgui application.

3 34. Show the output & explain why some entries are zero but some are not.

The output produced by `t = 0.1; n = 1:10; e = n/10 - n*t` emphasizes the existence of rounding errors in computer math. Since the output produces nonzero floats around -1.0e-15, we can assume that the numbers calculated by `n*t` are just barely larger than the numbers produced by `n/10`. This subtle yet visible difference in values highlights how computer math inevitably produces some rounding error through its use of different operators and functions.

5 35.

a. `x = 1; while 1+x > 1, x = x/2, pause(.02), end`
   i. Produces 53 lines of output.
   ii. The last two lines of output represent x reaching and surpassing the eps in size. Since the eps is passed, 1+x is not greater than 1, and the function ends.
b. `x = 1; while x+x > x, x = 2*x, pause(.02), end`
   i. Produces 1024 lines of output.
   ii. The last two lines show the realmax being surpassed and x being regarded as infinite. Therefore, x+x is no longer greater than x, and the function ends.
c. `x = 1; while x+x > x, x = x/2, pause(.02), end`
   i. Produces 1075 lines of output.
   ii. The last two lines show the realmin being surpassed and x being regarded as 0. Therefore, x+x is no longer greater than x, and the function ends.

5 38.

When calculating the roots for a, b, and c, I got the roots x = [100000000, 7.4506e-09]. When I ran the method roots([a b c]) I got the results x = [1.0e+08, 0]. When trying to compute the roots by hand, I found that the variables make it difficult to write-out. When using a calculator, I got the same results as using the quadratic formula in matlab. After considering how `$x_1x_2 = c/a$` I found some error in the calculator derived answer. However, the answer provided by the roots( ) method has no error. If I use the values from the calculator through the formula `$x_1x_2 = c/a$` I get the same result as from the roots( ) method.

Amelia Rotondo
CWID: 887925113

Homework: Chapter 1

4

39.

In the program `powersin.m` the while loop is responsible for checking if s+t is about equal to s; if so, the loop will terminate. For the values x = π/2, 11π/2, 21π/2, and 31π/2:

a. Respectively, the answer given fluctuates from:
   i. 1, to -1, to 0.9999, to -5.822e+03

b. Respectively, the largest term fluctuates from:
   i. 0.646, to 3.0665e+06, to 1.4673e+13, to 7.989e+19

c. Respectively, the amount of terms used fluctuates from:
   i. 11, to 37, to 60, to 78

My conclusion about using floating-point arithmetic and power series to evaluate functions is that these algorithmic methods are efficient when calculating values in a certain range. However, when calculating across a wide range of input values, their accuracy and efficiency diminishes.