Lecture 3: Control Flow and Loops

Math 98

Reminders and Agenda

- Reminders on Dates
  - Stay on top of that HW due every Wed and Fri.
- Agenda
  - Coursera Videos on `ifelse`, `for`, `while`, `break`, `continue`
  - Exercises
  - Bisection and Newton

Coursera Function Videos

- if statements
  - if, elseif, elseif......elseif, else, end
  - Nested if statements
- for loops
- while loops
- break

In Class Demo

Demonstration of `while`, `continue`, and `break`: `manyFrogs.m`

Exercise: sumCubes.m

Write a program sumCubes.m of the form

```
function S = sumCubes(v)
```

that takes a vector as input and returns the sum of the cubes of its elements. For pedagogical purposes, do this by:

1. Initializing a variable `S = 0` to keep track of the sum
2. Use a `for` loop

Do you know a much simpler way to do this?

Example: testPrime.m

Write a function of the form

```
function [isPrime,divisor] = testPrime(n)
```

that takes in an integer n and returns isPrime $=$ true if n is prime and false otherwise. It should return divisor $=$ NaN if the integer is prime and its smallest divisor otherwise.

(This should be obvious, but don't use the built in `MATLAB` function `isprime`)

128A Assignment

Implement a MATLAB function bisection.m of the form

```
  function p = bisection(f, a, b, tol)
% f: function handle y = f(x)
% a: Beginning of interval [a, b]
% b: End of interval [a, b]
% tol: user provided tolerance for interval width

% p: approximation to the root
```

128A Assignment

Implement a function newton.m of the form

```
function p = newton(f, df, p0, n)
% f: function handle y = f(x)
% df: function handle of derivative y' = f'(x)
% p0: initial estimate of the root
% n: number of steps

% p: approximation to the root
```

HW 3 and Project

With this, you should know everything you need to complete HW3 and the project.