

FUNCTIONS AND **FOR** LOOPING

Toby Sanders (sandertl@mailbox.sc.edu)

Department of Mathematics

OVERVIEW

The goal of this week's lab is to develop a MATLAB function and to learn to use a **for** loop.

MATLAB ESSENTIALS

- Defining a matrix and visiting any element of a matrix
- Creating a function
- Usage of the **for** loop (repetition) structure

ACTIVITIES

1. To define a matrix,

$$A = \begin{bmatrix} 2 & 3 & 5 & 7 \\ 11 & 13 & 17 & 19 \\ 23 & 29 & 31 & 37 \end{bmatrix},$$

for example, type

```
>> A=[2 3 5 7; 11 13 17 19; 23 29 31 37]
```

2. To access the i, j entry of A , type

```
>> A(i,j)
```

To access multiple entries of A at once, say the entries (i, j) and (i, k) , type

```
>>A(i,[j,k])
```

3. A few special commands for initializing matrices

- **eye(n)** creates the $n \times n$ identity matrix.
- **zeros(m,n)** creates the $m \times n$ zero matrix.
- **ones(m,n)** creates the $m \times n$ matrix with 1 in every entry.

Functions:

We create functions in M-files. The general form of the **function** statement is:

```
function output = functionname ( input )  
    ...  
end
```

For example,

```
function y = myfunction(x)    % Input: x, Output: y = x^2 + x + 1.
```

```
    y = x^2 + x + 1;
```

```
end
```

for loop:

A **for** loop executes commands repeatedly. The general form of the **for** statement is:

```
    for varname = startvalue : increment : endvalue
        ...
    end
```

For example,

```
>> a = 0;
>> for i = 1:1:5    % variable i takes on values 1, 2, 3, 4, and then 5
    a = a + i;
end
```

******In general, the command "**a:b:c**" will create a set of equally spaced points between **a** and **c**, with a spacing of **b**.