Functions and for Looping

Tommy Luckner
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 accessing any element of a matrix
- Creating a function
- Usage of the `for` loop (repetition) structure

Recall

- To define a matrix and vector,

$$A = \begin{pmatrix} 2 & 3 & 5 & 7 \\ 11 & 13 & 17 & 19 \\ 23 & 29 & 31 & 37 \end{pmatrix} \quad \text{and} \quad u = \begin{pmatrix} -4 & 3 & 12 \end{pmatrix},$$

  type

```
>> A=[2 3 5 7; 11 13 17 19; 23 29 31 37]
>> u=[-4 3 12]
```

New Commands

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

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

2. To access the i entry of a vector type

```
>> u(i)
```

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.

M-files

We use m-files or script files when creating a complicated program or function in MATLAB. These allow us to execute several lines of command at once. A MATLAB function file can be called upon in the command window and can act just like a built-in function. To open an m-file, you can type ctrl+N or go to file -> new -> script.

Creating Functions

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
```

In order to run this function from the command line or another m-file you must save it as ****.m, where **** should be the name of your function. For example, in the function above, it would be "myfunction.m".

As typed, the above function only works for a single input. How could we change this function to work for a vector input?

Note: Always put comments throughout an m-file to say what the code does so that it is clear to you when you come back to use it, or to someone else who may use it.

for loop

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

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

If no increment value is given, then the default is 1. For example,

```
a = 0;   % Initialize a.
for i = 1:1:5    % Variable i takes on values 1, 2, 3, 4, and then 5.
    a = a + i;   % What does this do?
end
```

The context of a `for` loop should always be indented between the `for` and `end` lines of the statement for readability.

In-Class Exercise

1. Let $u = \begin{pmatrix} 2 & 8 & 3 & 4 & 9 & 10 \end{pmatrix}$. Create a for loop that adds up the entries of the vector u.
2. Create a function called `vectorsum` that takes as input a vector of arbitrary length and sums up its entries. The loop should look similar to the one you just created.
3. Test your function with the following vectors.
   - u from problem 1
   - `x=linspace(0,1,300)`
   - some vector of your choosing

   You can check to see if your program is running correctly by using the MATLAB command

   ```
   >> sum(u)
   ```

   The command `h=sum(u)` makes h a scalar that is the sum of the elements of the vector u. If u is a matrix, h is a row vector with the sum over each column.