CS 105

Introduction to Scientific Computing

Lecture #4 – Vectors and Matrices

Matt Burlick
Stevens Institute of Technology

ASSIGNMENT 3

- Make a grayscale crop of an image
- Add noise to an audio file

NECESSARY SKILLS

- Loading images and audio into matrices
- Performing matrix operations
  - Including accessing matrix data

TOPICS

- Vectors and Matrices
  - Assigning
  - Accessing
  - Operations
  - Sub-matrices

READING

- Sections 2.1-2.4, 2.8

GROUPS OF DATA

- Often we may want to group values together
  - A "set" of numbers
  - Text
- In Computer Science we call a group of values an *array* or *vector*
  - In particular, an array or vector of characters is called a *string*.
- We can also have 2-D arrays or vectors
  - These are called *Matrices*

CREATING MATRICES IN MATLAB

- We use square brackets to group members of arrays
  - Each member is also called an *element* of the array.
  - You can separate elements either by spaces or commas.
- Example:
  - [4 3 5 0]
  - [4,3,5,0]
- To create a 2D array (i.e matrix) we separate rows using a semicolon
  - [4 3 5 0; 1 2 3 4]

$$\begin{matrix} 4 & 3 & 5 & 0 \\ 1 & 2 & 3 & 4 \end{matrix}$$

CREATING MATRICES: THE COLON OPERATOR

- We can use the colon operator to make arrays
  - 1:1:4 → [1 2 3 4]
    - This means, start at 1, increase by steps of 1 up to 4
  - 1:2:8 → [1 3 5 7]
  - 1:3 → [1 2 3]
    - This is a shortcut for 1:1:3

ACCESS DATA IN MATRICES

- There are additional operations we may want to do on arrays.
- In particular we may want to *access* certain elements
- To do this we use
  VarName(location)
  - Where VarName is the variable storing the data and location is the numerical location in the array
  - NOTE: In MATLAB the first location is location 1 (most other programming languages start with location 0)

ACCESSING DATA IN MATRICES

```
X = [4 3 8 0];
```

- We may also want to select several locations at once
- We do this by providing a location *array*
  - X([1 3 4])
  - X(1:3)
- We can use the colon operator on its own to mean "all"
  - X(:)
- There is also a special *end* location
  - X(end)
- If we have a Matrix, we can specify both the rows and the columns by separating the locations with a comma
  - Y(1,2)
  - Y(1:2,[2 3])

```
Y = [4 3 5; 0 2 3];
```

ARRAY/MATRIX OPERATIONS

- Do you recall what it means to multiply two matrices?
  - X = Y*Y
- Same restrictions on matrix division and exponents
- We don't have to worry about this for scalar multiplication/division or addition and subtraction
  - X = 2*Y;
  - X = Y + Y;

```
Y = [4 3 5; 0 2 3];
```

ELEMENT-BY-ELEMENT OPERATIONS

- But what if we want multiplication, division, and exponents to behavior more like the addition/subtraction with matrices?
- Why would we want this?
  - Useful when doing computations on a bunch of values
  - Evaluate $y=x^2$ for x=1,2,3,…1000
- Matlab allows for *element-by-element* operations
  - To do this we precede the operator with a dot, .
    - [3 2 1].*[1 2 3]

EX: EVALUATE FOR SET OF VALUES

- Evaluate $y=x^2+2x$ for $x=1,2,3,4$