Lecture 4 outline

I. Vector example: cumulative sum
   a. Look for opportunities to reuse results you’ve already computed

II. Scenario: statistics of rolling dice
   a. Can count occurrences of each result in a vector (one element per outcome)
   b. Plot histogram using bar() function
   c. Generate random rolls
      i. Uniform probability: equally likely to choose a number in two ranges of the same width
      ii. Divide total range into N regions for N outcomes
      iii. Scale to make region widths integers, then “round” using ceil() or floor()
         1. Using round() will lead to extra outcome, non-equal likelihoods
   d. Look up bin from outcome
      i. If outcomes are integers, can use directly as vector indices
      ii. Accumulation pattern: be sure to initialize vector to zero

III. More plotting
   a. title(), xlabel(), ylabel(): Label your plots! (include units!)
   b. Can plot lines between points
   c. Can plot many line segments or points at once by plotting longer vectors of x and y coordinates
      i. Much faster than `hold on`
   d. Can change font sizes, line widths, etc.
e. Can plot multiple lines with one command (without `hold on`)

f. Can add legend with `legend()`

IV. Vector preallocation

a. Much faster in MATLAB to pre-allocate a vector before populating it if you know how long it will eventually be

b. Typically preallocate using `zeros()` function

c. If final length is unknown, appending is fine (ignore warning)

V. 2D arrays

a. Example 2D datasets: topography (height map), images, diffraction patterns

b. Must be rectangular (all rows have same # of columns)

c. Index using `M(r,c)`

d. Create using same tools a vectors, but not constrained to single row or column

   i. Concatenations must be rectangular, though

e. Query number of rows, columns using `size()`

VI. Nested loops

a. Traverse matrix with nested loops (one over rows, one over columns)

b. “Row-major” traversal: outer loop over rows, inner loop over columns

c. It matters where you write code:

   i. Between loops: Action at start of each row

   ii. Inside inner loop: Action for each column (within the row)

   iii. Between `end`'s: Action at end of each row