Announcements:
- Prelim 1 on Thurs, 2/24, 7:30-9p. Check course website for locations
- Section this week in classrooms (not labs)
- Read over project solutions & grading guides
- Evaluate your TA on-line!

Previous Lecture:
- User-defined functions

Today's Lecture:
- 1-d array—vector
- Easy plots in MATLAB

1-d array: vector
- An array is a named collection of like data organized into rows or columns
- A 1-d array is a row or a column, called a vector
- An index identifies the position of a value in a vector

score
1 2 3 4 5 6
93 92 87 0 90 82

Accessing values in a vector
For the vector score ...
- the kth value: score(k)
- assign value 9 to 4th cell: score(4) = 9
- copy value in 5th cell to 2nd cell: score(2) = score(5)
- copy value in current cell to next cell: score(k+1) = score(k)

Creating vectors
Manual entry of array values:
vecA = [2 3.5 6] %row vector
Enclose values in square brackets. Using a blank or a comma as the separator gives you a row. Using a semi-colon as a separator gives a column vector.
vecB = [2; 3.5; 6] %column vector

Creating vectors
One cell at a time:
vecC(1) = 10; %[10]
vecC(2) = 1; %[10 1]
vecC(5) = 8; %[10 1 0 0 8]
x = vec(6); %ERROR! Can’t access a cell that doesn’t exist
Creating vectors

Use built-in functions:

```matlab
vecC = zeros(1,4) % row of 4 zeros
vecD = ones(4,1) % column of 4 1s
```

Creating vectors

Use short-cut expressions:

```matlab
vecE = 2:7 % [2 3 4 5 6 7]
vecF = 6:-2:1 % [6 4 2]
```

Transpose (change from row to col, or vice versa):

```matlab
vecG = vecF' % [6; 4; 2]
```

Example 1

- Write a program fragment that calculates the cumulative sums of a given vector \( v \).
- The cumulative sums should be stored in a vector of the same length as \( v \).

1, 3, 5, 0 \( v \)

1, 4, 9, 9 cumulative sums of \( v \)

Example 2

- Write a function `evalPoly` to evaluate an \( n^{th} \) order polynomial of \( x \):
  
  \[ a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \]

- Input parameter `coef` has length \( n+1 \), contains the coefficients of the polynomial
- `coef(1)` is the coefficient for the term \( x^0 \)
- Input parameter `x`
- Return the value of the polynomial evaluated at `x`
- No MATLAB predefined function other than `length`

Example 3: Take a hike!

- Write a function `randomWalk` to perform \( n \) steps of random walk starting from \((x_0, y_0)\):
  
  ```matlab
  function randomWalk(n, x0, y0)
  ```

- At every step, the possible movements are left, right, up, or down
- Display the walk