Previous Lecture:
- Probability and random numbers
- 1-d array—vector

Today’s Lecture:
- More examples on vectors and simulation
- Color computation
- Linear interpolation

Announcement:
- Project 3 due on Friday 3/1
- Prelim 1 on Thurs 3/7 at 7:30pm

Simulation
- Imitates real system
- Requires judicious use of random numbers
- Requires many trials
- \( \rightarrow \) opportunity to practice working with vectors!

Loop patterns for working with a vector

```matlab
% Given a vector v
for k = 1:length(v)
    % Work with v(k)
    % E.g., disp(v(k))
end

% Given a vector v
k = 1;
while k <= length(v)
    % Work with v(k)
    % E.g., disp(v(k))
    k = k+1;
end
```

2-dimensional random walk

Start in the middle tile, (0,0).
For each step, randomly choose between N,E,S,W and then walk one tile. Each tile is 1x1.
Walk until you reach the boundary.

Another representation for the random step

- Observe that each update has the form
  \[ \begin{align*}
  x &= x + \Delta x \\
  y &= y + \Delta y
  \end{align*} \]
- no matter which direction is taken.
- So let’s get rid of the if statement!
- Need to create two "change vectors" deltaX and deltaY

```matlab
deltaX 1
deltaY 1
```
Example: polygon smoothing

Can store the x-y coordinates in vectors x and y

function [xNew,yNew] = Centralize(x,y)
% Translate polygon defined by vectors x,y such that the centroid is on the
% origin. New polygon defined by vectors xNew,yNew.
  n = length(x);
  xBar = sum(x)/n;
  yBar = sum(y)/n;
  xNew = x-xBar;
  yNew = y-yBar;

Second operation: normalize

function [xNew,yNew] = Normalize(x,y)
% Resize polygon defined by vectors x,y such that distance of the vertex
% furthest from origin is 1
  d = max(sqrt(x.^2 + y.^2));
  xNew = x/d;
  yNew = y/d;

Third operation: smooth

Obtain a new polygon by connecting the midpoints of the edges
function \([x_{\text{New}}, y_{\text{New}}] = \text{Smooth}(x, y)\)

% Smooth polygon defined by vectors \(x, y\)
% by connecting the midpoints of
% adjacent edges

\(n = \text{length}(x);\)
\(x_{\text{New}} = \text{zeros}(n,1);\)
\(y_{\text{New}} = \text{zeros}(n,1);\)

for \(i=1:n\)

\(\text{Compute the midpt of ith edge.}\)
\(\text{Store in } x_{\text{New}}(i) \text{ and } y_{\text{New}}(i)\)
end

\(x_{\text{New}}(1) = (x(1)+x(2))/2\)
\(y_{\text{New}}(1) = (y(1)+y(2))/2\)

\((x_1,y_1)\)
\((x_2,y_2)\)
\((x_3,y_3)\)
\((x_4,y_4)\)
\((x_5,y_5)\)

\(x_{\text{New}}(2) = (x(2)+x(3))/2\)
\(y_{\text{New}}(2) = (y(2)+y(3))/2\)

\(x_{\text{New}}(i) = (x(i) + x(i+1))/2;\)
\(y_{\text{New}}(i) = (y(i) + y(i+1))/2;\)

end

Polygon Smoothing

% Given \(n, x, y\)
for \(i=1:n\)

\(x_{\text{New}}(i) = (x(i) + x(i+1))/2;\)
\(y_{\text{New}}(i) = (y(i) + y(i+1))/2;\)
end

Does above fragment compute the new \(n\)-gon?

A: Yes
B: No
Color computation

- Color is a 3-vector, sometimes called the RGB values
- Any color is a mix of red, green, and blue
- Example:
  \[ c = [0.4 \ 0.6 \ 0] \]
- Each component is a real value in [0,1]
- [0 0 0] is black
- [1 1 1] is white

Making an x-y plot

- \[ a = [0 \ 4 \ 3 \ 8]; \quad \% \text{x-coords} \]
- \[ b = [1 \ 2 \ 5 \ 3]; \quad \% \text{y-coords} \]
- \[ \text{plot}(a, b, '*') \]

Making an x-y plot with multiple graphs (lines)

- \[ a = [0 \ 4 \ 3 \ 8]; \]
- \[ b = [1 \ 2 \ 5 \ 3]; \]
- \[ f = [0 \ 4 \ 6 \ 8 \ 10]; \]
- \[ g = [2 \ 6 \ 4 \ 3]; \]
- \[ \text{plot}(a, b, '-*', f, g, '.') \]

Linear interpolation

<table>
<thead>
<tr>
<th>(x)</th>
<th>(g(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>110</td>
</tr>
<tr>
<td>10</td>
<td>118</td>
</tr>
<tr>
<td>10.25</td>
<td>?</td>
</tr>
<tr>
<td>10.50</td>
<td>?</td>
</tr>
<tr>
<td>10.75</td>
<td>?</td>
</tr>
<tr>
<td>11</td>
<td>126</td>
</tr>
<tr>
<td>12</td>
<td>134</td>
</tr>
</tbody>
</table>

\[ g(10.5) = \frac{1}{2} g(11) + \frac{1}{2} g(10) \]

function paintChips(c1,c2,n)

% n tiles from color c1 to c2

\[ f = \text{???} \]
\[ v = (1-f)*c1 + f*c2; \]

% Use linear interpolation to obtain the colors. Each chip has a color \(v\) that is a linear combination of colors \(c1\) and \(c2\). Let \(f\) be a fraction in \((0,1)\) ...

% Draw kth tile

end