Previous Lecture:
- Examples on vectors and simulation

Today’s Lecture:
- Finite vs. Infinite; Discrete vs. Continuous
- Vectors and vectorized code
- Color computation with linear interpolation

Announcements:
- Project 3 due Monday at 11pm
- Prelim 1 on Thursday 10/13 at 7:30pm. You must notify us now if you have an exam conflict. Email Randy Hess (rbh27) with your conflict information (course number, instructor contact info).

---

Loop patterns for working with a vector

```matlab
% Given a vector v
k = 1;
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
```

---

Discrete vs. continuous

A plot is made from discrete values, but it can look continuous if there’re many points.

---

Screen Granularity

After how many halvings will the disks disappear?

---

Xeno’s Paradox

- A wall is two feet away
- Take steps that repeatedly halve the remaining distance
- You never reach the wall because the distance traveled after n steps =
  \[ 1 + \frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^n} = 2 - \frac{1}{2^n} \]

---

Example: “Xeno” disks

Draw a sequence of 20 disks where the (k+1)th disk has a diameter that is half that of the kth disk.

The disks are tangent to each other and have centers on the x-axis.

First disk has diameter 1 and center (1/2, 0).
Example: “Xeno” disks

What do you need to keep track of?
• Diameter (d)
• Position
  Left tangent point (x)

<table>
<thead>
<tr>
<th>Disk</th>
<th>x</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>2</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>3</td>
<td>3/4</td>
<td>1/8</td>
</tr>
</tbody>
</table>

% Xeno Disks

DrawRect(0,-1,2,2,'k')
% Draw 20 Xeno disks

for k= 1:20
  % Draw kth disk
  % Update x, d for next disk
end

% Xeno Disks

DrawRect(0,-1,2,2,'k')
% Draw 20 Xeno disks

d= 1;
x= 0; % Left tangent point
for k= 1:20
  % Draw kth disk
  % Update x, d for next disk
end

Here's the output... Shouldn't there be 20 disks?

The "screen" is an array of dots called pixels.
Disks smaller than the dots don't show up.
The 20th disk has radius<.000001

Fading Xeno disks

- First disk is yellow
- Last disk is black (invisible)
- **Interpolate** the color in between

Color is a 3-vector, sometimes called the RGB values

- Any color is a mix of red, green, and blue
- Example:
  \[
  \text{colr} = [0.4 \ 0.6 \ 0]
  \]
- Each component is a real value in [0,1]
- \([0 \ 0 \ 0]\) is black
- \([1 \ 1 \ 1]\) is white
% Draw n fading Xeno disks
d = 1;
x = 0;  % Left tangent point
yellow = [1 1 0];
black = [0 0 0];
for k = 1:n
    % Compute color of kth disk
    colr = ___ * black + ___ * yellow;
    % Draw kth disk
    DrawDisk(x+d/2, 0, d/2, colr)
    x = x+d;
    d = d/2;
end

Example: 3 disks fading from yellow to black
r = 1;  % radius of disk
yellow = [1 1 0];
black = [0 0 0];

% Left disk yellow, at x=1
DrawDisk(1,0,r,yellow)

% Right disk black, at x=5
DrawDisk(5,0,r,black)

% Middle disk with average color, at x=3
colr = 0.5*yellow + 0.5*black;
DrawDisk(3,0,r,colr)

Vectorized code allows an operation on multiple values at the same time

yellow = [1 1 0];
black = [0 0 0];

% Average color via vectorized op
colr = 0.5*yellow + 0.5*black;

% Average color via scalar op
for k = 1:length(black)
    colr(k) = 0.5*yellow(k) + 0.5*black(k);
end

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>

\[ g(10.5) = \frac{1}{4} g(11) + \frac{3}{4} g(10) \]
% Draw n fading Xeno disks
d= 1;
x= 0;  % Left tangent point
yellow= [1 1 0];
black= [0 0 0];

for k= 1:n
    % Compute color of kth disk
    f= (k-1)/(n-1);
color= f*black + (1-f)*yellow;
    % Draw kth disk
    DrawDisk(x+d/2, 0, d/2, color)
    x= x+d;
d= d/2;
end

\[
\frac{k}{n} \quad \frac{k}{n-1} \quad \frac{(k-1)}{n} \quad \frac{(k-1)}{n+1}
\]

\[
A \quad B \quad C \quad D \quad E
\]

Does this script print anything?

k = 0;
while 1 + 1/2^k > 1
    k = k+1;
end
disp(k)

Computer Arithmetic—floating point arithmetic

Suppose you have a calculator with a window like this:

\[
\begin{array}{cccc}
+ & 2 & 4 & 1 & - & 3 \\
\end{array}
\]

representing \(2.41 \times 10^{-3}\)

Floating point addition

\[
\begin{array}{cccc}
+ & 2 & 4 & 1 & - & 3 \\
+ & 1 & 0 & 0 & - & 3 \\
\end{array}
\]

Result:

\[
\begin{array}{cccc}
+ & 3 & 4 & 1 & - & 3 \\
\end{array}
\]
Floating point addition

\[
\begin{array}{c}
+ & 2 & 4 & 1 & - & 3 \\
+ & 1 & 0 & 0 & - & 6 \\
\end{array}
\]

Result: \[+ 2 4 1 - 3\]

Not enough room to represent .002411

The loop DOES terminate given the limitations of floating point arithmetic!

\[
k = 0; \\
\text{while } 1 + 1/2^k > 1 \quad k = k+1; \\
\text{end} \\
disp(k)
\]

1 + 1/2^53 is calculated to be just 1, so "53" is printed.

Patriot missile failure

In 1991, a Patriot Missile failed, resulting in 28 deaths and about 100 injured. The cause?

Inexact representation of time/number

- System clock represented time in tenths of a second: a clock tick every 1/10 of a second
- Time = number of clock ticks \( \times 0.1 \)

Error of .000000095 every clock tick

Resulting error

... after 100 hours

0.000000095 \( \times (100\times60\times60) \)

0.34 second

At a velocity of 1700 m/s, missed target by more than 500 meters!

Computer arithmetic is inexact

- There is error in computer arithmetic—floating point arithmetic—due to limitation in “hardware.” Computer memory is finite.
- What is 1 + 10^{-16}?
  - 1.0000000000000001 in real arithmetic
  - 1 in floating point arithmetic (IEEE)
- Read Sec 4.3