61A Extra Lecture 1
Announcements

• If you want 1 unit (pass/no pass) of credit for this CS 98, stay tuned for a Piazza post
Announcements

• If you want 1 unit (pass/no pass) of credit for this CS 98, stay tuned for a Piazza post
  ▪ Only for people who really want extra work that's beyond the scope of normal CS 61A
Announcements

• If you want 1 unit (pass/no pass) of credit for this CS 98, stay tuned for a Piazza post
  • Only for people who really want extra work that's beyond the scope of normal CS 61A
• Anyone is welcome to attend the extra lectures, whether or not they enroll
Announcements

• If you want 1 unit (pass/no pass) of credit for this CS 98, stay tuned for a Piazza post
  ▪ Only for people who really want extra work that's beyond the scope of normal CS 61A
• Anyone is welcome to attend the extra lectures, whether or not they enroll
• Permanent lecture times are TBD, but probably Wednesday evening or Monday evening
Newton's Method
Newton's Method Background

Quickly finds accurate approximations to zeroes of differentiable functions!
Newton's Method Background

Quickly finds accurate approximations to zeroes of differentiable functions!

\[ f(x) = x^2 - 2 \]
Newton's Method Background

Quickly finds accurate approximations to zeroes of differentiable functions!

\[ f(x) = x^2 - 2 \]
Newton's Method Background

Quickly finds accurate approximations to zeroes of differentiable functions!

f(x) = x^2 - 2

A "zero" of a function f is an input x such that f(x)=0
Newton's Method Background

Quickly finds accurate approximations to zeroes of differentiable functions!

\[ f(x) = x^2 - 2 \]

A "zero" of a function \( f \) is an input \( x \) such that \( f(x) = 0 \)

\[ x = 1.414213562373095 \]
Newton's Method Background

Quickly finds accurate approximations to zeroes of differentiable functions!

A "zero" of a function $f$ is an input $x$ such that $f(x)=0$

Application: a method for computing square roots, cube roots, etc.
Newton's Method Background

Quickly finds accurate approximations to zeroes of differentiable functions!

\[ f(x) = x^2 - 2 \]

A "zero" of a function \( f \) is an input \( x \) such that \( f(x) = 0 \).

Application: a method for computing square roots, cube roots, etc.

The positive zero of \( f(x) = x^2 - a \) is \( \sqrt{a} \). (We're solving the equation \( x^2 = a \).)
Newton's Method

Given a function $f$ and initial guess $x$, 
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$: 
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$: 
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

Compute the value of $f$ at the guess: $f(x)$
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

- Compute the value of $f$ at the guess: $f(x)$
- Compute the derivative of $f$ at the guess: $f'(x)$
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

1. Compute the value of $f$ at the guess: $f(x)$
2. Compute the derivative of $f$ at the guess: $f'(x)$
3. Update guess $x$ to be:
   \[ x = x - \frac{f(x)}{f'(x)} \]
Newton's Method

Given a function \( f \) and initial guess \( x \),

Repeatedly improve \( x \):

Compute the value of \( f \) at the guess: \( f(x) \)

Compute the derivative of \( f \) at the guess: \( f'(x) \)

Update guess \( x \) to be:

\[
x - \frac{f(x)}{f'(x)}
\]
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

Compute the value of $f$ at the guess: $f(x)$

Compute the derivative of $f$ at the guess: $f'(x)$

Update guess $x$ to be:

$$x - \frac{f(x)}{f'(x)}$$
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

- Compute the value of $f$ at the guess: $f(x)$
- Compute the derivative of $f$ at the guess: $f'(x)$
- Update guess $x$ to be:

$$x = \frac{f(x)}{f'(x)}$$
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

- Compute the value of $f$ at the guess: $f(x)$
- Compute the derivative of $f$ at the guess: $f'(x)$
- Update guess $x$ to be:
  $$x - \frac{f(x)}{f'(x)}$$
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

Compute the value of $f$ at the guess: $f(x)$

Compute the derivative of $f$ at the guess: $f'(x)$

Update guess $x$ to be:

$$x - \frac{f(x)}{f'(x)}$$
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

Compute the value of $f$ at the guess: $f(x)$

Compute the derivative of $f$ at the guess: $f'(x)$

Update guess $x$ to be:

$$x = \frac{-f(x)}{f'(x)}$$
Newton's Method

Given a function $f$ and initial guess $x$,

Repeatedly improve $x$:

Compute the value of $f$ at the guess: $f(x)$

Compute the derivative of $f$ at the guess: $f'(x)$

Update guess $x$ to be:

$$x - \frac{f(x)}{f'(x)}$$

Finish when $f(x) = 0$ (or close enough)
Newton's Method

Given a function $f$ and initial guess $x$, repeatedly improve $x$:

1. Compute the value of $f$ at the guess: $f(x)$
2. Compute the derivative of $f$ at the guess: $f'(x)$
3. Update guess $x$ to be:
   \[ x \leftarrow x - \frac{f(x)}{f'(x)} \]
4. Finish when $f(x) = 0$ (or close enough)

Using Newton's Method
Using Newton's Method

How to find the square root of 2?
Using Newton's Method

How to find the square root of 2?

```python
>>> f = lambda x: x**x - 2
>>> df = lambda x: 2*x
>>> find_zero(f, df)
1.4142135623730951
```
Using Newton's Method

How to find the square root of 2?

```python
>>> f = lambda x: x**2 - 2
>>> df = lambda x: 2*x
>>> find_zero(f, df)
1.4142135623730951
```
Using Newton's Method

How to find the square root of 2?

\[ f(x) = x^2 - 2 \]
\[ f'(x) = 2x \]

```python
>>> f = lambda x: x**2 - 2
>>> df = lambda x: 2*x
>>> find_zero(f, df)
1.4142135623730951
```
How to find the square root of 2?

```python
>>> f = lambda x: x**2 - 2
>>> df = lambda x: 2*x
>>> find_zero(f, df)
1.4142135623730951
```

This applies Newton's method.
Using Newton's Method

How to find the square root of 2?

How to find the cube root of 729?

\[
\begin{align*}
\text{How to find the square root of 2?} & \\
\text{How to find the cube root of 729?} & \\
\end{align*}
\]
Using Newton's Method

How to find the square root of 2?

```python
>>> f = lambda x: x**2 - 2
>>> df = lambda x: 2*x
>>> find_zero(f, df)
1.4142135623730951
```

How to find the cube root of 729?

```
Applies Newton's method
```
Using Newton's Method

How to find the square root of 2?

\[ f(x) = x^2 - 2 \]
\[ f'(x) = 2x \]

\[ f(x) = x^2 - 2 \]
\[ f'(x) = 2x \]

\[
\begin{align*}
\sqrt{2} & = 1.4142135623730951 \\
1 & = 1
\end{align*}
\]

How to find the cube root of 729?

\[ g(x) = x^3 - 729 \]
\[ g'(x) = 3x^2 \]

\[
\begin{align*}
\sqrt[3]{729} & = 9.0 \\
V & = V
\end{align*}
\]
Using Newton's Method

How to find the square root of 2?

\[
\begin{align*}
\text{f}(x) &= x^2 - 2 \\
\text{f}'(x) &= 2x \\
\text{Applies Newton's method}
\end{align*}
\]

\[
\begin{align*}
\triangleright\triangleright & \quad \text{f} = \lambda x: x^2 - 2 \\
\triangleright\triangleright & \quad \text{df} = \lambda x: 2x \\
\triangleright\triangleright & \quad \text{find}_\text{zero}(f, df) \\
& \quad 1.4142135623730951
\end{align*}
\]

How to find the cube root of 729?

\[
\begin{align*}
\text{g}(x) &= x^3 - 729 \\
\text{g}'(x) &= 3x^2 \\
\triangleright\triangleright & \quad \text{g} = \lambda x: x^3 - 729 \\
\triangleright\triangleright & \quad \text{dg} = \lambda x: 3x^2 \\
\triangleright\triangleright & \quad \text{find}_\text{zero}(g, dg) \\
& \quad 9.0
\end{align*}
\]
Iterative Improvement
Special Case: Square Roots
Special Case: Square Roots

How to compute \texttt{square_root}(a)

\textbf{Idea:} Iteratively refine a guess \(x\) about the square root of \(a\)
Special Case: Square Roots

How to compute square_root(a)

**Idea:** Iteratively refine a guess x about the square root of a

**Update:**
Special Case: Square Roots

How to compute $\text{square}_\text{root}(a)$

**Idea:** Iteratively refine a guess $x$ about the square root of $a$

**Update:** \[ x = \frac{x + \frac{a}{x}}{2} \]
Special Case: Square Roots

How to compute square_root(a)

**Idea:** Iteratively refine a guess $x$ about the square root of $a$

**Update:**

$$x = \frac{x + \frac{a}{x}}{2}$$

Babylonian Method
Special Case: Square Roots

How to compute square_root(a)

**Idea:** Iteratively refine a guess $x$ about the square root of $a$

**Update:**

$$x = \frac{x + \frac{a}{x}}{2}$$

(Demo) Babylonian Method
Special Case: Square Roots

How to compute square_root(a)

**Idea:** Iteratively refine a guess $x$ about the square root of $a$

**Update:**

$$x = \frac{x + \frac{a}{x}}{2}$$

Implementation questions:
Special Case: Square Roots

How to compute square_root(a)

Idea: Iteratively refine a guess $x$ about the square root of $a$

Update: $x = x + \frac{a}{x}\quad (Demo)$

Implementation questions:

What guess should start the computation?
Special Case: Square Roots

How to compute square_root(a)

**Idea:** Iteratively refine a guess x about the square root of a

\[ x = \frac{x + \frac{a}{x}}{2} \]  

(Demo)  

**Babylonian Method**

Implementation questions:

What guess should start the computation?

How do we know when we are finished?
Special Case: Cube Roots
Special Case: Cube Roots

How to compute \( \text{cube\_root}(a) \)

**Idea:** Iteratively refine a guess \( x \) about the cube root of \( a \)
Special Case: Cube Roots

How to compute \( \text{cube\_root}(a) \)

**Idea:** Iteratively refine a guess \( x \) about the cube root of \( a \)

**Update:**
Special Case: Cube Roots

How to compute $cube_root(a)$

**Idea:** Iteratively refine a guess $x$ about the cube root of $a$

**Update:** \[ x = \frac{2 \cdot x + \frac{a}{x^2}}{3} \]
Special Case: Cube Roots

How to compute cube_root(a)

Idea: Iteratively refine a guess x about the cube root of a

Update: \[ x = \frac{2 \cdot x + \frac{a}{x^2}}{3} \] (Demo)
Special Case: Cube Roots

How to compute \( \text{cube\_root}(a) \)

**Idea:** Iteratively refine a guess \( x \) about the cube root of \( a \)

\[
\text{Update: } \quad x = \frac{2 \cdot x + \frac{a}{x^2}}{3} \quad \text{(Demo)}
\]

**Implementation questions:**
Special Case: Cube Roots

How to compute cube_root(a)

**Idea:** Iteratively refine a guess $x$ about the cube root of $a$

$$x = \frac{2 \cdot x + \frac{a}{x^2}}{3}$$

**Implementation questions:**

What guess should start the computation?
**Special Case: Cube Roots**

How to compute \( \text{cube}_\text{root}(a) \)

**Idea:** Iteratively refine a guess \( x \) about the cube root of \( a \)

**Update:** \[
    x = \frac{2 \cdot x + \frac{a}{x^2}}{3}
\]

**Implementation questions:**

- What guess should start the computation?
- How do we know when we are finished?
Implementing Newton's Method

(Demo)
Extensions
Approximate Differentiation
Approximate Differentiation
Approximate Differentiation

Differentiation can be performed symbolically or numerically
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]

\[ f'(x) = 2x \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]

\[ f'(x) = 2x \]

\[ f'(2) = 4 \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]

\[ f'(x) = 2x \]

\[ f'(2) = 4 \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]
\[ f'(x) = 2x \]
\[ f'(2) = 4 \]

\[ f'(x) = \lim_{a \to 0} \frac{f(x + a) - f(x)}{a} \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]
\[ f'(x) = 2x \]
\[ f'(2) = 4 \]
\[ f'(x) = \lim_{a \to 0} \frac{f(x + a) - f(x)}{a} \]
\[ f'(x) \approx \frac{f(x + a) - f(x)}{a} \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]
\[ f'(x) = 2x \]
\[ f'(2) = 4 \]

\[ f'(x) = \lim_{a \to 0} \frac{f(x + a) - f(x)}{a} \]
\[ f'(x) \approx \frac{f(x + a) - f(x)}{a} \quad \text{(if } a \text{ is small)} \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]
\[ f'(x) = 2x \]
\[ f'(2) = 4 \]

\[ f'(x) = \lim_{a \to 0} \frac{f(x + a) - f(x)}{a} \]

\[ f'(x) \approx \frac{f(x + a) - f(x)}{a} \quad \text{(if } a \text{ is small)} \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]

\[ f'(x) = 2x \]

\[ f'(2) = 4 \]

\[ f'(x) = \lim_{a \to 0} \frac{f(x + a) - f(x)}{a} \]

\[ f'(x) \approx \frac{f(x + a) - f(x)}{a} \quad \text{(if } a \text{ is small)} \]
Approximate Differentiation

Differentiation can be performed symbolically or numerically

\[ f(x) = x^2 - 16 \]
\[ f'(x) = 2x \]
\[ f'(2) = 4 \]

\[ f'(x) = \lim_{a \to 0} \frac{f(x + a) - f(x)}{a} \]
\[ f'(x) \approx \frac{f(x + a) - f(x)}{a} \quad \text{(if } a \text{ is small)} \]
Critical Points and Inverses
Critical Points and Inverses

Maxima, minima, and inflection points of a differentiable function occur when the derivative is 0.
Critical Points and Inverses

Maxima, minima, and inflection points of a differentiable function occur when the derivative is 0.
Critical Points and Inverses

Maxima, minima, and inflection points of a differentiable function occur when the derivative is 0

(Demo)

http://upload.wikimedia.org/wikipedia/commons/f/fd/Stationary_vs_inflection_pts.svg
Critical Points and Inverses

Maxima, minima, and inflection points of a differentiable function occur when the derivative is 0

\[(\text{Demo})\]

The inverse $f^{-1}(y)$ of a differentiable, one-to-one function computes the value $x$ such that $f(x) = y$
Critical Points and Inverses

Maxima, minima, and inflection points of a differentiable function occur when the derivative is 0.

The inverse $f^{-1}(y)$ of a differentiable, one-to-one function computes the value $x$ such that $f(x) = y$. 

(Demo)