# Bisection

Math 426

University of Alaska Fairbanks

August 31, 2020

# Solve for $x$

A common task: "solve for $x$".

- $3x + 7 = 9$

# Solve for $x$

A common task: "solve for $x$".

- $3x + 7 = 9$
- $7x^2 + 2x + 5 = 9$

# Solve for $x$

A common task: "solve for $x$".

- $3x + 7 = 9$
- $7x^2 + 2x + 5 = 9$
- $12x^3 + 7x^2 - 2x + 5 = 12$

# Solve for $x$

A common task: "solve for $x$".

- $3x + 7 = 9$
- $7x^2 + 2x + 5 = 9$
- $12x^3 + 7x^2 - 2x + 5 = 12$
- $e^x + x = 3$

# Solve for $x$

A common task: "solve for $x$".

- $3x + 7 = 9$
- $7x^2 + 2x + 5 = 9$
- $12x^3 + 7x^2 - 2x + 5 = 12$
- $e^x + x = 3$
- $x^6 + 12x^2 + 3x = 4$

# Our task

We want to solve $f(x) = c$ for some function $f$ and some value $c$.

# Our task

We want to solve $f(x) = c$ for some function $f$ and some value $c$.

Rather than find an exact solution $x_{\text{exact}}$, we'll use the computer to

1. Find an approximate solution $x_{\text{est}}$ so that $f(x_{\text{est}}) \approx c$.

# Our task

We want to solve $f(x) = c$ for some function $f$ and some value $c$.

Rather than find an exact solution $x_{\text{exact}}$, we'll use the computer to

1. Find an approximate solution $x_{\text{est}}$ so that $f(x_{\text{est}}) \approx c$.
2. Find an estimate for the size of the **error**

$$\text{error} = \left|x_{\text{est}} - x_{\text{approx}}\right|$$

# Basic transformation

We want to solve

$$f(x) = c.$$

# Basic transformation

We want to solve

$$f(x) = c.$$

Let $F(x) = f(x) - c$. So $f(x) = c$ if and only if $F(x) = 0$.

# Basic transformation

We want to solve

$$f(x) = c.$$

Let $F(x) = f(x) - c$. So $f(x) = c$ if and only if $F(x) = 0$.

So we can always transform the equation so that $c = 0$. We'll use $F$ for the name of the function. A solution of $F(x) = 0$ is call a **root** of $F$.

# Idea of Bisection

Suppose we know numbers $a$ and $b$ with $a < b$ and

$$F(a) < 0$$
$$F(b) > 0$$

Then there should be a $c$ somewhere in the middle so that $F(c) = 0$.

# Idea of Bisection

Suppose we know numbers $a$ and $b$ with $a < b$ and

$$F(a) < 0$$
$$F(b) > 0$$

Then there should be a $c$ somewhere in the middle so that $F(c) = 0$.

Not so fast:

- $F(x) = \dfrac{1}{x}$
- $a = -1,\ F(a) = -1$
- $b = 1,\ F(b) = 1$

# Idea of Bisection

Suppose we know numbers $a$ and $b$ with $a < b$ and

$$F(a) < 0$$
$$F(b) > 0$$

Then there should be a $c$ somewhere in the middle so that $F(c) = 0$.

Not so fast:

- $F(x) = \begin{cases} 1 & x > 0 \\ -1 & x \leq 0 \end{cases}$
- $a = -1,\ F(a) = -1$
- $b = 1,\ F(b) = 1$

# Intermediate Value Theorem

Extra ingredient: **continuity**.

### Theorem

*Suppose $f$ is a continuous function on an interval $[a, b]$. Then for each value of $y$ between $f(a)$ and $f(b)$ there exists $c \in [a, b]$ such that*

$$f(c) = y.$$

# Intermediate Value Theorem

Extra ingredient: **continuity**.

### Theorem

*Suppose $f$ is a continuous function on an interval $[a, b]$. Then for each value of $y$ between $f(a)$ and $f(b)$ there exists $c \in [a, b]$ such that*

$$f(c) = y.$$

So if $F$ is continuous, $F(a) < 0$ and $F(b) > 0$ there is $c$ somewhere in between such that $F(c) = 0$. This guarantees a root.

# Bisection Algorithm

Given:

- A continous function $F$.
- Numbers $a$, $b$.
- $F(a)$ and $F(b)$ have opposite signs.
- $\delta$, an error tolerance

# Bisection Algorithm

## Bisection Iteration

```
F_a = F(a)
F_b = F(b)

while abs(b-a)<2*delta

   c = (b+a)/2;

   F_c = F(c);

   if sign(F_a) = sign(F_c)
      a = c;
      F_a = F_c;
   else
      b = c;
      F_b = F_c;
   end
end

root = (a+b)/2;
```