Bisection

Math 426

University of Alaska Fairbanks

September 2, 2020

Last Class

Given a function f we wish to find a solution of

$$f(x) = c.$$

Last Class

Given a function f we wish to find a solution of

$$f(x) = c.$$

By defining $F(x) = f(x) - c$ we can assume $c = 0$.

Last Class

Given a function f we wish to find a solution of

$$f(x) = c.$$

By defining $F(x) = f(x) - c$ we can assume $c = 0$.

Find functions $F(x)$ that

- A solution of $F(x) = 0$ is $\sqrt{2}$.

Last Class

Given a function f we wish to find a solution of

$$f(x) = c.$$

By defining $F(x) = f(x) - c$ we can assume $c = 0$.

Find functions $F(x)$ that

- A solution of $F(x) = 0$ is $\sqrt{2}$.
- A solution of $F(x) = 0$ is π.

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 In A Picture

Bisection Algorithm

Given:

- A continous function F.
- Numbers a, b.
- $F(a)$ and $F(b)$ have opposite signs.
- δ, 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;
```

Recall

Analysis of Bisection

Given an application of bisection:

1. How good an approximation is the result?
2. How much work is needed to compute the result?

Analysis of Bisection

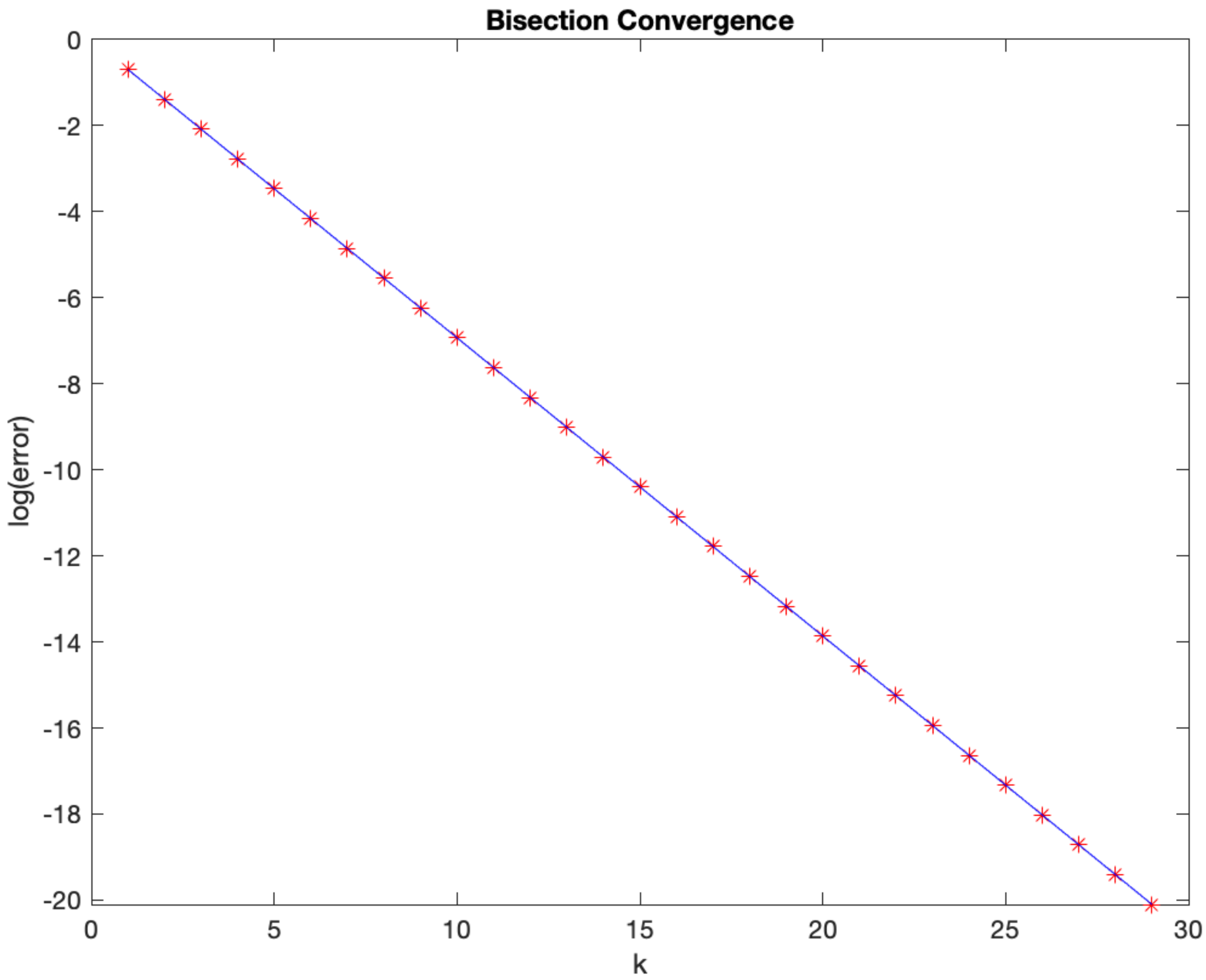
Given an application of bisection:

1. How good an approximation is the result?
2. How much work is needed to compute the result?

Notation:

- a_k, b_k: the interval endpoints at step k
- $m_k = (a_k + b_k)/2$: the midpoint of interval k
- $e_k = |x_{\text{exact}} - m_k|$: the **absolute error** at step k

Convergence Rate



Convergence Rate

Labor per Digit

What can go wrong?

- Need to have a guess for the initial interval.
- Some (rare) root cannot be found: $F(x) = x^2$ never changes sign.
- Workload seems fair: every 3.3 steps we gain a digit. But we can do better!