## Bisection Method:

only use sign of $f(c)$ to determine...

| a | b | c | f(c) | |
|---|---|---|---|---|
| $-\pi/2$ | 0 | $-\pi/4$ | $-.07829\ldots$ | too small |
| $-\pi/4$ | 0 | $-\pi/8$ | $.53118\ldots$ | too big |
| $-\pi/4$ | $-\pi/8$ | $-3\pi/16$ | $.2424\ldots$ | too big |
| $-\pi/4$ | $-3\pi/16$ | $-7\pi/32$ | $.085782$ | ... |
| $-\pi/4$ | $-7\pi/32$ | $-15\pi/64$ | $0.00464$ | too big |

The idea is to maintain a interval $[a, b]$ that contains the solution over each iteration...

1. Guaranteed to converge because the interval shrinks by $1/2$ each iteration..
2. Size of the interval provides an error bound on the solution...

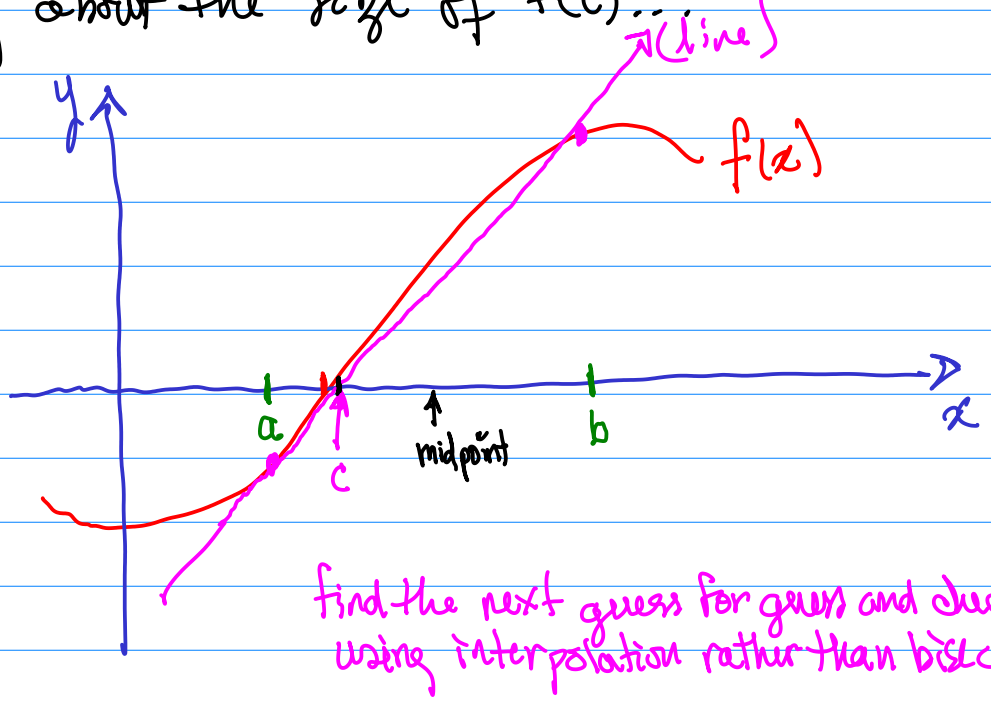
The length of this interval over two $\frac{b-a}{2} = \frac{\frac{-7\pi}{32} - \left(\frac{-\pi}{4}\right)}{2}$

is a bound on the error in the approximation $\frac{-15\pi}{16}$.

3. Though guaranteed the rate of convergence is slow... Could be 100's of iterations to get to 15 digits of precision.

④ Doesn't make any use of the size of $f(c)$.

Try to increase speed of convergence by using something about the size of $f(c)$...




find the next guess for guess and check using interpolation rather than bisection.

line passing through $(a, f(a))$ and $(b, f(b))$

slope: $\frac{f(b)-f(a)}{b-a}$

$$y - f(b) = \frac{f(b)-f(a)}{b-a}(x-b)$$

point slope form for the line

Solve when the line passes through zero... set $y=0$

$$-f(b) = \frac{f(b)-f(a)}{b-a}(x-b)$$

$$\frac{f(b)-f(a)}{b-a}b - f(b) = \frac{f(b)-f(a)}{b-a}x$$

$$c = b - \frac{b-a}{f(b)-f(a)} f(b)$$

next guess...


$$f(x) \equiv x + \cos x$$

guess $c = -0.6110\ldots$

```
julia> f(x)=x+cos(x)
f (generic function with 1 method)

julia> a=-pi/2; b=0.0
0.0

julia> c=b-(b-a)/(f(b)-f(a))*f(b)
-0.6110154703516573
```

check it $f(c) = 0.208\ldots$

too big.

```
julia> f(c)
0.2080503950703395
```

| a | b | c = interpolate | f(c) |
|---|---|---|---|
| $-\pi/2$ | $-0.6110\ldots$ | $-.7232\ldots$ | $0.02637\ldots$ |

```
julia> b=c
-0.6110154703516573

julia> c=b-(b-a)/(f(b)-f(a))*f(b)
-0.7232695414357495
```

```
julia> f(c)
0.026376287678165022
```

too big...

| a | b | c = interpolate | f(c) |
|---|---|---|---|
| $-\pi/2$ | $-.7232\ldots$ | continue on ... | |

The above Idea is called method of false position...

As mentioned in the Prologue, the *method of false position*† dates back to the ancient Egyptians. It remains an effective alternative to the bisection method for solving the equation $f(x) = 0$ for a real root between $a$ and $b$, given that $f$ is continuous and $f(a)$ and $f(b)$ have opposite signs.

① Guaranteed Convergence? In bisection cut in half and then no matter which interval still contains the solution after checking, it's anyway half the size as before...

With false position one interval is bigger than the other and it's not clear which will be kept.

② Often faster than bisection, but might not be if something strange happens..

③ We still have a bracket on the error, given by just the length of the interval...

Idea: Keep the best of both bisection and false position methods. How, by alternation and using one after the other each iteration..

bisect
false position
bisect
false position
⋮

So every other iteration the interval is guaranteed to decrease by 1/2 so it converges...

So every other iteration the information about the size of f(a) and f(b) is used, so there is a potential to converge faster...

---

Another idea that's even simpler... Secant method...

With false position the guess might always update only one of the endpoints.... then the other endpoint reflect old data of old guesses for the solution...

```
fc=f(c)
println("[$a,$b] $c $fc")
if fc>0
    b=c
else
    a=c
end
```

reverse this if

lol

Choose which endpoint based on whether c was too big or too small...

---

Idea always update the oldest data...

a ← b  replace

b ← c  replace...

This does not preserve an interval containing the exact solution. But it always involves the most

recent approximations of the solution...

Input

```
a=-pi/2; b=0.0
for n=1:5
    c=b-(b-a)/(f(b)-f(a))*f(b)
    fc=f(c)
    println("[$a,$b] $c $fc")
    b=a; b=c
end
```

Output

```
[-1.5707963267948966,0.0] -0.6110154703516573 0.2080503950703395
[-1.5707963267948966,-0.6110154703516573] -0.7232695414357495 0.026376287678165022
[-1.5707963267948966,-0.7232695414357495] -0.7372659060759975 0.0030434567148837077
[-1.5707963267948966,-0.7372659060759975] -0.7388777688479117 0.00034703160812754597
[-1.5707963267948966,-0.7388777688479117] -0.7390615216757179 3.951635041565815e-5
```

got small quickly...

Traded guaranteed convergence and definite bounds on the error for faster convergence...