$$ax^2 + bx + c = 0$$

familiar with the formula for its roots:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Question: Can you rewrite that formula with the $\sqrt{\ }$ in denominator?

Lot's of equations don't have a closed form solution:

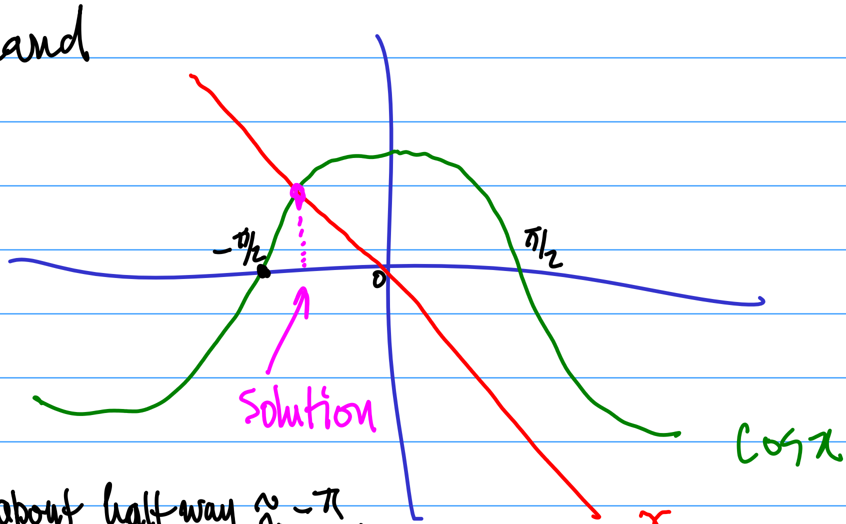
$$f(x) \equiv x + \cos x = 0$$

Goal approximate solution on a computer

Idea: Graph it

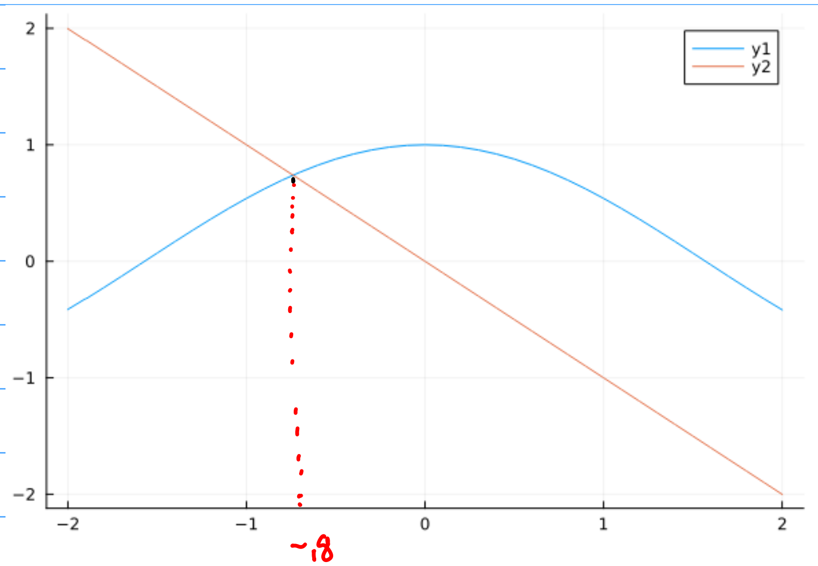
$\cos x = -x$

By hand



about halfway $\approx -\frac{\pi}{4}$

Using the computer




```
julia> using Plots

julia> xs=-2:0.1:2
-2.0:0.1:2.0

julia> plot(xs,cos.(xs))

julia> plot!(xs,-xs)
```

Note $-\frac{\pi}{4} \approx -.78539...$

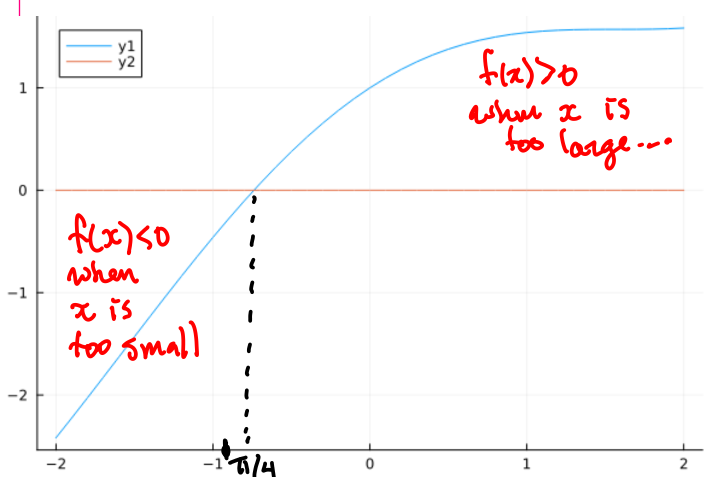
Rule of thumb, If you want some information about the error in an approximation, solve the problem in different ways and see how much the answers differ...

Bisection method... If we know one value is too small and the other too large... guess the one in the middle (and check).

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

```
julia> plot(xs,f.(xs))

julia> plot!(xs,0*xs)
```



general for when f(x) is increasing... it's reversed if f is decreasing...

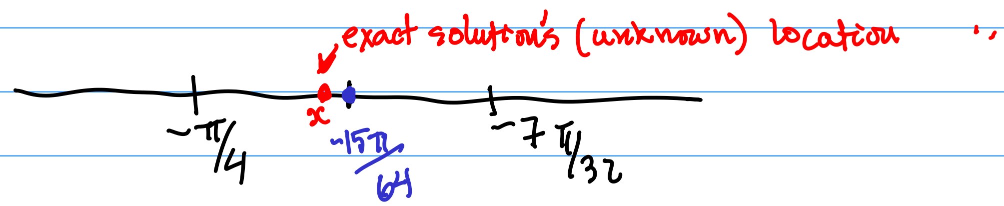
$$f(x) \equiv x + \cos x$$

not zero... negative...

a	b	c	f(c)	
$-\pi/2$	0	$-\pi/4$	$-.07829...$	too small
$-\pi/4$	0	$-\pi/8$	$.53118...$	too big
$-\pi/4$	$-\pi/8$	$-3\pi/16$	$.2424...$	too big
$-\pi/4$	$-3\pi/16$	$-7\pi/32$	$.085787$	...
$-\pi/4$	$-7\pi/32$	$-15\pi/64$	0.00464	too big

the correct solution is somewhere in this interval..

What about the error in the approximation $x^* = -15\pi/64$?



$$e_{abs} = |x - x^*| \le \frac{1}{2}|b-a| = \frac{1}{2}\left|-\frac{7\pi}{32} - -\frac{\pi}{4}\right| \approx .049...$$

```
julia> 1/2*abs(-7*pi/32+pi/4)
0.04908738521234052
```

Thus, I have 1D accuracy guaranteed at this point.

In code the bisection method...

```
julia> a=-pi/2; b=0
       for n=1:5
           c=(a+b)/2
           fc=f(c)
           println("[$a,$b] $c $fc")
           if fc>0
               b=c
           else
               a=c
           end
       end
[-1.5707963267948966,0] -0.7853981633974483 -0.0782913822109007
[-0.7853981633974483,0] -0.39269908169872414 0.5311804508125626
[-0.7853981633974483,-0.39269908169872414] -0.5890486225480862 0.24242098975445903
[-0.7853981633974483,-0.5890486225480862] -0.6872233929727672 0.08578706038996975
[-0.7853981633974483,-0.6872233929727672] -0.7363107781851077 0.004640347169851511
```

reverse this if f is a decreasing function

lot's of digits because using 64-bit floats

Tidy up the output

```
julia> using Printf

julia> a=-pi/2; b=0
       for n=1:5
           c=(a+b)/2
           fc=f(c)
           @printf("%14.6e %14.6e %14.6e %14.6e\n",a,b,c,fc)
           if fc>0
               b=c
           else
               a=c
           end
       end
 -1.570796e+00   0.000000e+00  -7.853982e-01  -7.829138e-02
 -7.853982e-01   0.000000e+00  -3.926991e-01   5.311805e-01
 -7.853982e-01  -3.926991e-01  -5.890486e-01   2.424210e-01
 -7.853982e-01  -5.890486e-01  -6.872234e-01   8.578706e-02
 -7.853982e-01  -6.872234e-01  -7.363108e-01   4.640347e-03
```

Label the output

```
julia> a=-pi/2; b=0
       @printf("%14s %14s %14s %14s\n","a","b","c","f(c)")
       for n=1:5
           c=(a+b)/2
           fc=f(c)
           @printf("%14.6e %14.6e %14.6e %14.6e\n",a,b,c,fc)
           if fc>0
               b=c
           else
               a=c
           end
       end
             a              b              c           f(c)
 -1.570796e+00   0.000000e+00  -7.853982e-01  -7.829138e-02
 -7.853982e-01   0.000000e+00  -3.926991e-01   5.311805e-01
 -7.853982e-01  -3.926991e-01  -5.890486e-01   2.424210e-01
 -7.853982e-01  -5.890486e-01  -6.872234e-01   8.578706e-02
 -7.853982e-01  -6.872234e-01  -7.363108e-01   4.640347e-03
```