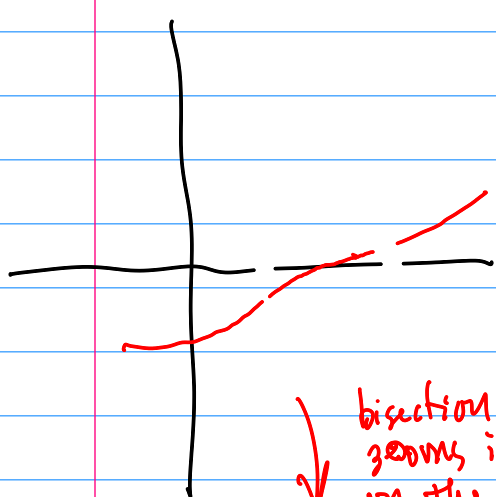
Last time we used the bisection method to find an approximation of $\sqrt{2}$.

```
f(x)=x^2-2
a=0
b=2
cold=0
while true
    c=(a+b)/2
    if cold==c
        break
    end
    global cold=c
    println("sqrt(2)=",c," p/m ",(b-a)/2)
    if f(c)>0
        global b=c
    else
        global a=c
    end
end
```

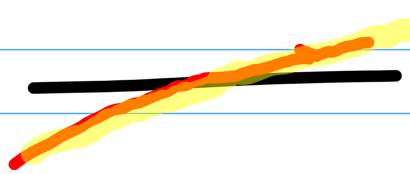
check for when guesses repe
no more precision possible c

how accurately we can compute f(c) determines how accurate we can find an approximation of x such that $f(x)=0$.

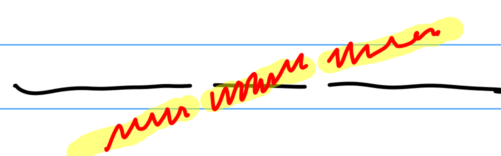


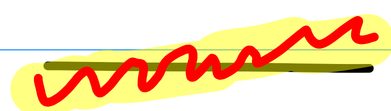
bisection zooms in on the graph

It could happen that there are lots of x's that when plugged into $f(x)$ the rounded computation of f leads to zero. More likely in a poorly conditioned problem there is a range of x's that when plugged into the rounded computation of f lead to answers slightly greater or less than 0 in what looks like a random way due to the rounding...



rounding error makes the line wiggley





can't find x exactly

4.2 Fixed Point methods

$f(x) = x^2 - 2$

Used bisection 52 times to approximate $\sqrt{2}$

assume g is continuous and differentiable.

New idea find a function $g(x)$ such that the sequence defined by $x_{n+1} = g(x_n)$ satisfies $x_n \to \sqrt{2}$

If $x_n \to \sqrt{2}$ what are some necessary properties g must have?

assume this

cont

$$x_{n+1} = g(x_n)$$

as $n \to \infty$

$$\sqrt{2} = g(\sqrt{2}) \qquad \text{Thus } g(\sqrt{2}) = \sqrt{2}.$$

Error: $\varepsilon_n = x_n - \sqrt{2}$. Then by the mean value theorem

$$\varepsilon_{n+1} = x_{n+1} - \sqrt{2} = g(x_n) - g(\sqrt{2}) = g'(\xi_n)(x_n - \sqrt{2}) = g'(\xi_n)\,\varepsilon_n$$

where ξ_n is between x_n and $\sqrt{2}$

Thus $|\varepsilon_{n+1}| = |g'(\xi_n)|\,|\varepsilon_n|$

If the error is going to decrease, then we need $|g'(\xi_n)| < 1$.

Question: How do I find a function g such that $g(\sqrt{2}) = \sqrt{2}$ and $|g'(\xi_n)| < 1$ for all those n?

parameter to choose.

Idea: Let $g(x) = x + c f(x)$ ← cont. and differentiable $\quad f(x) = x^2 - 2$

$$g(\sqrt{2}) = \sqrt{2} + c f(\sqrt{2}) = \sqrt{2} \qquad f(\sqrt{2}) = 0$$

$$g'(x) = 1 + c f'(x) = 1 + c(2x)$$

Since we are looking for a positive root (in this case) then we can assume $x > 0$ and so $c < 0$ for $g'(x) < 1$.

But we don't want $g'(x)$ to be too negative otherwise $|g'(x)|$ would be large..

I know $1 < \sqrt{2} < 2$ it doesn't make sense to approximate $\sqrt{2}$ by anything larger than 2.

lets assume $x < 2$ so that $1 + c(2x) = 1 - 2|c|x > 1 - 4|c|$

negative

I need $g'(x) > -1$ but it looks easy to make a stronger condition and choose c so $g'(x) > 0$.

Let $c = -\frac{1}{4}$ then for $0 < x \leq 2$

we have

$$0 < \frac{x}{2} \leq 1$$

$$0 > -\frac{x}{2} \geq -1$$

$$1 > \underbrace{1 - \frac{x}{2}}_{} \geq 0$$

$g'(x) = 1 + c(2x) = 1 - \frac{1}{2}x$

Thus $g'(x) \in [0,1)$ if $x \in (0,2]$.

Let $x_0 = 2$ and $x_{n+1} = g(x_n)$ where $g(x) = x - \frac{1}{4}f(x)$ and $f(x) = x^2 - 2$

Then $x_n \to \sqrt{2}$ as $n \to \infty$. Let's check...

Remark: Newton's method can be seen as a way of automatically tuning the value of c. Right now $c = -\frac{1}{4}$.

```
julia> f(x)=x^2-2
f (generic function with 1 method)

julia> g(x)=x+c*f(x)
g (generic function with 1 method)

julia> c=-1/4
-0.25
```

```
julia> xn=x0
       for n=1:20
           xn=g(xn)
           println("x_$n=",xn)
       end
x_1=1.5
x_2=1.4375
x_3=1.4208984375
x_4=1.4161603450775146
x_5=1.4147828143349983
x_6=1.4143802114005837
x_7=1.4142623658001936
```

```
x_8=1.4142278559705035
x_9=1.4142177488197718
x_10=1.414214788550556
x_11=1.4142139215117826
x_12=1.4142136675623491
x_13=1.4142135931823114
x_14=1.4142135713969055
x_15=1.4142135650161078
x_16=1.4142135631472157
x_17=1.4142135625998298
x_18=1.414213562439504
x_19=1.4142135623925458
x_20=1.4142135623787921
```

```
x_1=1.5 ϵ_1=0.08578643762690485
x_2=1.4375 ϵ_2=0.023286437626904855
x_3=1.4208984375 ϵ_3=0.006684875126904855
x_4=1.4161603450775146 ϵ_4=0.001946782704419503
x_5=1.4147828143349983 ϵ_5=0.0005692519619031611
x_6=1.4143802114005837 ϵ_6=0.00016664902748853017
x_7=1.4142623658001936 ϵ_7=4.8803427098453867e-5
x_8=1.4142278559705035 ϵ_8=1.4293597408343572e-5
x_9=1.4142177488197718 ϵ_9=4.186446676572406e-6
x_10=1.414214788550556 ϵ_10=1.2261774609001463e-6
x_11=1.4142139215117826 ϵ_11=3.5913868745574007e-7
x_12=1.4142136675623491 ϵ_12=1.0518925397384749e-7
x_13=1.4142135931823114 ϵ_13=3.080921628928479e-8
x_14=1.4142135713969055 ϵ_14=9.023810365604845e-9
x_15=1.4142135650161078 ϵ_15=2.6430126975895973e-9
x_16=1.4142135631472157 ϵ_16=7.741205454436795e-10
x_17=1.4142135625998298 ϵ_17=2.2673463107025782e-10
x_18=1.414213562439504 ϵ_18=6.640887839637344e-11
x_19=1.4142135623925458 ϵ_19=1.9450663302222893e-11
x_20=1.4142135623787921 ϵ_20=5.696998428561528e-12
```

```
julia> c=-1/3
-0.3333333333333333
```

```
julia> xn=x0
       for n=1:20
           xn=g(xn)
           println("x_$n=",xn," ϵ_$n=",xn-sqrt(2.0))
       end
x_1=1.3333333333333335 ϵ_1=-0.08088022903976166
x_2=1.4074074074074074 ϵ_2=-0.006806154965687705
x_3=1.413808870598994 ϵ_3=-0.0004046917741011846
x_4=1.4141903630708597 ϵ_4=-2.3199302235488162e-5
x_5=1.414212235403363 ϵ_5=-1.3269697320961171e-6
x_6=1.4142134864818374 ϵ_6=-7.589125772256011e-8
x_7=1.4142135580327995 ϵ_7=-4.3402956784177604e-9
x_8=1.4142135621248695 ϵ_8=-2.482256622471368e-10
x_9=1.4142135623588987 ϵ_9=-1.4196421815881877e-11
x_10=1.4142135623722831 ϵ_10=-8.120171202108395e-13
x_11=1.4142135623730485 ϵ_11=-4.6629367034256575e-14
x_12=1.4142135623730925 ϵ_12=-2.6645352591003757e-15
x_13=1.414213562373095 ϵ_13=-2.220446049250313e-16
x_14=1.4142135623730951 ϵ_14=0.0
x_15=1.414213562373095 ϵ_15=-2.220446049250313e-16
x_16=1.4142135623730951 ϵ_16=0.0
x_17=1.414213562373095 ϵ_17=-2.220446049250313e-16
x_18=1.4142135623730951 ϵ_18=0.0
x_19=1.414213562373095 ϵ_19=-2.220446049250313e-16
x_20=1.4142135623730951 ϵ_20=0.0
```

$$c = \frac{-1}{f'(\sqrt{2})}$$

$$f'(x) = 2x$$

$$f'(\sqrt{2}) = 2\sqrt{2}$$

$$c = \frac{-1}{2\sqrt{2}}$$

```
julia> c=-1/(2*sqrt(2))
-0.35355339059327373

julia> xn=x0
       for n=1:20
           xn=g(xn)
           println("x_$n=",xn," ϵ_$n=",xn-sqrt(2.0))
       end
x_1=1.2928932188134525 ϵ_1=-0.12132034355964261
x_2=1.409009742330268 ϵ_2=-0.005203820042827045
x_3=1.4142039882381274 ϵ_3=-9.574134967715864e-6
x_4=1.4142135623406868 ϵ_4=-3.240829826722802e-11
x_5=1.4142135623730951 ϵ_5=0.0
x_6=1.414213562373095 ϵ_6=-2.220446049250313e-16
```

Newton's method uses x_n to approximate the ideal value for c automatically...

$$c = \frac{-1}{2x_n}$$

and $$g(x_n) = x_n - \frac{1}{2x_n}(x_n^2 - 2)$$