Before moving on to linear algebra, let's summarize the good and bad about the root finding methods in steps 5 — 10.

1. Bisection
2. ~~false position~~
3. Secant method
4. Newton's method.

	Bisection	Secant	Newton
When Convergence	Guaranteed.	unknown, need some condition on x_1 and x_2	If x_1 is close enough to the root
How fast converges	Linear $e_{n+1} \approx \frac{1}{2} e_n$	Super linear but less than quadratic. $e_{n+1} \approx M e_n^{1.6\ldots}$	Quadratic $e_{n+1} \approx M e_n^2$
Hypothesis on f	Don't even need a continuous function, just one that answers the question too big or too small	only used f to iterate but analysis again used f' and f''.	f' needs to exist and $f'(p) \neq 0$ for the quadratic conv. and f'' to be cont for Taylor analysis..

Moving on to linear algebra... LU factorization...

Example:

$A = LU$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 0 & 1 \end{bmatrix}$$

Eliminate below the pivot

this coefficient is greater than 1 and will multiply any error in r_1 by 4

$r_2 \leftarrow r_2 - 4r_1$

row col put 4 there

$r_3 \leftarrow r_3 - r_1$

```
julia> A=[1 2 3; 4 5 6; 1 0 1]
3×3 Matrix{Int64}:
 1  2  3
 4  5  6
 1  0  1

julia> using LinearAlgebra

julia> det(A)
-6.0

julia>
```

invertible matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -2 & -2 \end{bmatrix}$$

$r_3 \leftarrow r_3 - \frac{2}{3} r_2$

use these to make L

$-2 - \frac{2}{3}(-6) = -2+4 = 2$

$$U = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & 2 \end{bmatrix} \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 1 & 2/3 & 1 \end{bmatrix}$$

When doing Gaussian elimination by hand sometime a zero shows up where the pivot was supposed to be... then you have to swap rows to procede...

With a computer we always swap rows so the pivot has the largest value to reduce rounding errors...

Example:

A=LU

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 0 & 1 \end{bmatrix}$$

eliminate below the

~~$r_2 \leftarrow r_2 - 4 r_1$~~

row col put 4 there

fix this by swaping rows first

$r_2 \leftrightarrow r_1$

pivot

$$\begin{bmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \\ 1 & 0 & 1 \end{bmatrix}$$

$r_2 \leftarrow r_2 - \frac{1}{4} r_1$

$r_3 \leftarrow r_3 - \frac{1}{4} r_1$

by swapping rows so the pivot has the largest magnitude, errors are kept under control.

```
julia> Ar=Rational.(A)
3x3 Matrix{Rational{Int64}}:
 1//1  2//1  3//1
 4//1  5//1  6//1
 1//1  0//1  1//1

julia> Ar[[1,2],:]=Ar[[2,1],:];

julia> Ar
3x3 Matrix{Rational{Int64}}:
 4//1  5//1  6//1
 1//1  2//1  3//1
 1//1  0//1  1//1
```

```
julia> Ar[2,:]=Ar[2,:]-1//4*Ar[1,:];

julia> Ar[3,:]=Ar[3,:]-1//4*Ar[1,:];

julia> Ar
3x3 Matrix{Rational{Int64}}:
 4//1   5//1   6//1
 0//1   3//4   3//2
 0//1  -5//4  -1//2
```

$$\begin{bmatrix} 4 & 5 & 6 \\ 0 & 3/4 & 3/2 \\ 0 & -5/4 & -1/2 \end{bmatrix}$$

Note, we are working with factions so no rounding error. So this isn't needed and only an in convenience... However with floating point it's a necessity...

Q: How does swapping rows scramble the LU factorization? How to unscramble it.