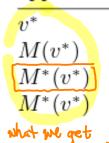
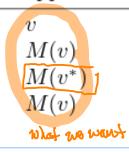
Ways of measuring errors:  $x \in \mathbb{R}$  is the exact value and  $x^*$  the approximation

(1) Obsolute error  $Cas_1 = |x - x^*|$ (2) relative error ...,  $C_{rel} = \frac{Cab_2}{|x|}$ 

Types!

approximation approximand type of error





initial error propagated error generated error total cumulative error

M is the function or algorithm we want to apply.

M4 is an approximation of that function programmed into a compution.

already initial error in the inputs

julia> 4.27\*3.68 15.7136

$$(4.27 \times 10^{1}) \times (3.68 \times 10^{1}) = 15.7136 \times 10^{2} \rightarrow 1.57 \times 10^{3}$$

julia> 4.27e1\*3.68e1 1571.36 rounding step 35 generalles more error.

round back to 35

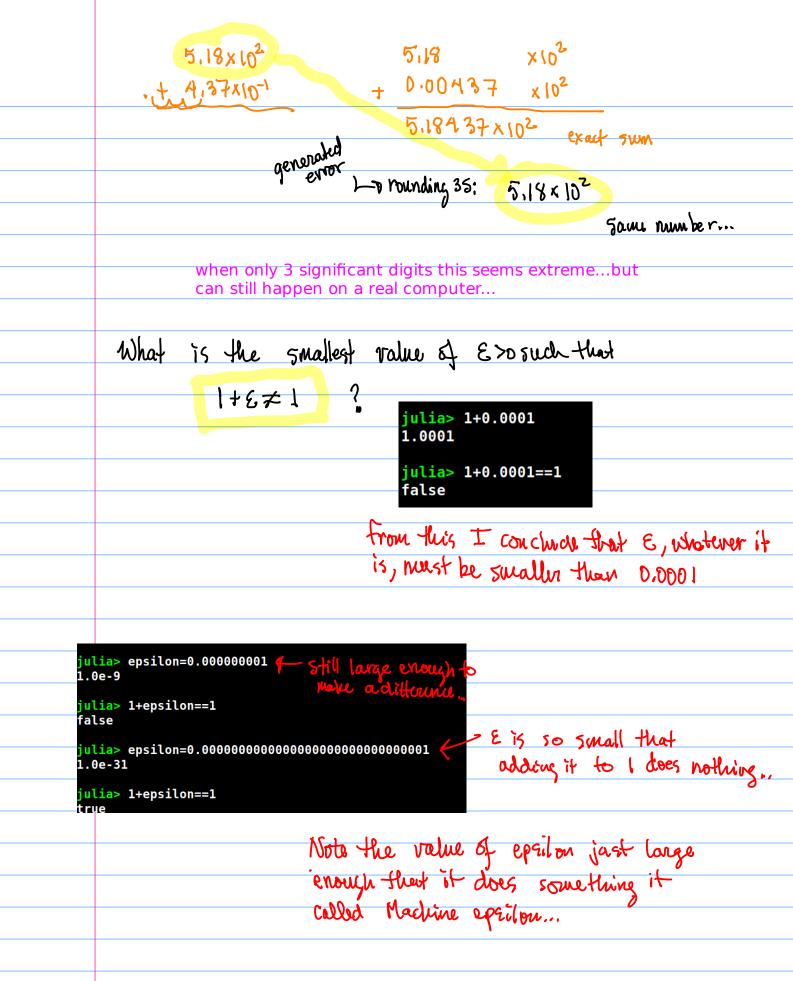
grun 1570 = 1,57×103

Because of the rounding the computer doesn't perform multiplication exactly...and so ...

addition of a small to a large # may have no effect ",

$$5.18 \times 10^2 + 4.37 \times 10^{-1} = 5.18 \times 10^2 + 0.00437 \times 10^{20}$$
  
=  $5.18437 \times 10^2 \rightarrow 5.18 \times 10^2$ 

How much smaller does the have to be for no effect?



```
julia> epsilon=1e-16 🗲 🚾 🚾
1.0e-16
julia> 1+epsilon==1
julia> epsilon=1e-15
1.0e-15
julia> 1+epsilon==1
false
```

makes a difference.

1.0 0.00000000000000000

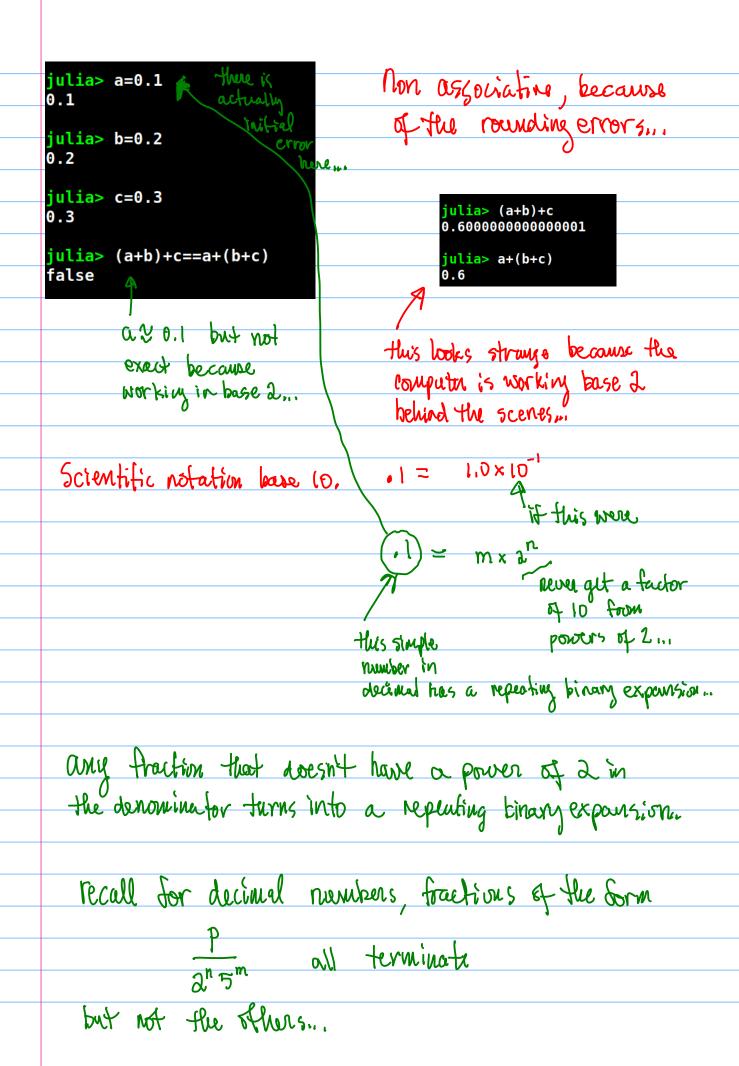
indicates about 15 significant digits are available...

the associative law for addition does not always hold.

(a+b)+c = 
$$\alpha$$
+(b+c)  
If approximate addition... if can happen that
$$(\alpha + b) + c \neq \alpha + (b + c)$$

```
<mark>julia> a=1.0</mark>
1.0
julia> b=2.0
2.0
julia> c=3.0
julia> (a+b)+c==a+(b+c)
```

In this case associativity holds,,



- Step 01
- Step 02
- Step 03, Non-Associativity of Addition, Types of Errors
- Step 04, Realistic Floating Point Data Types, Rounding Examples
- Step 05, Taylor's Theorem

Mad to have for wednesday,,,

already initial error in the inputs 
$$15.7136$$
  $15.7136$   $15.7136$