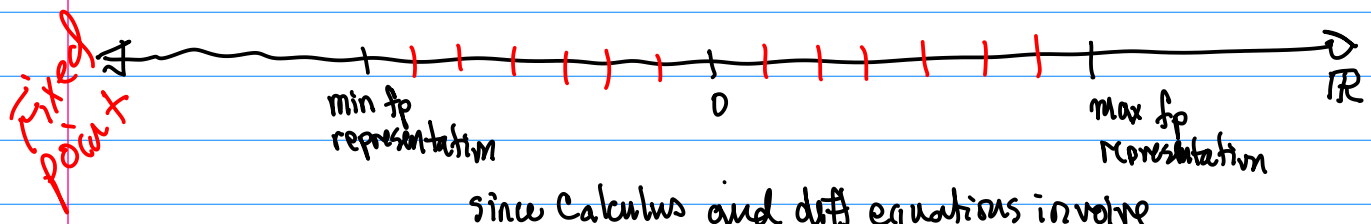
Floating point arithmetic.

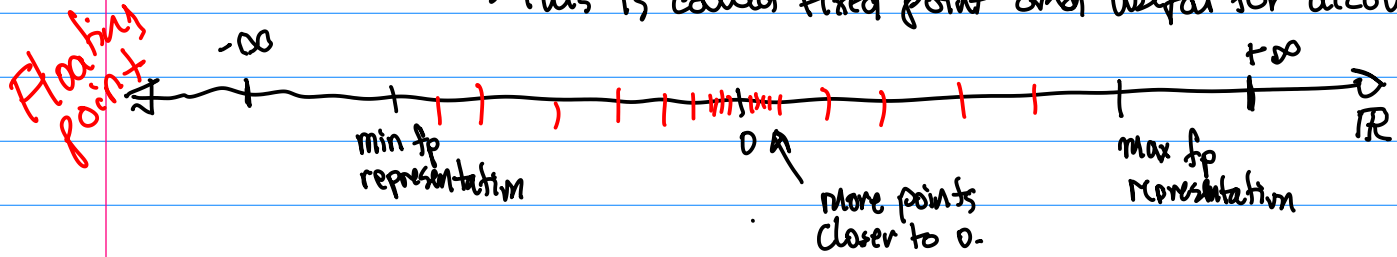
Real Numbers — infinite set — uncountably infinite set

Digital Computer — finite — 64-bit floating point



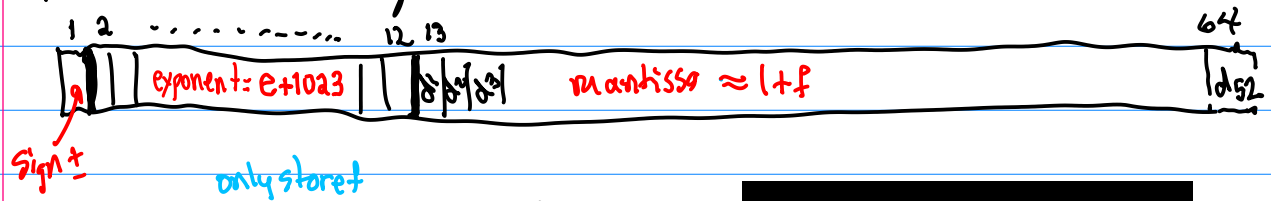
since Calculus and diff equations involve limits to zero, equally spacing is a bad idea.

- This is called fixed point and useful for accounting.



since Calculus and diff. equations involve limits to zero, put more points close to zero..

Represent numbers using 64 bits



$$\pm \underbrace{d_0}_{1}.\underbrace{d_1 d_2 d_3 \cdots}_{f} d_{p-1} \times 2^e$$

only stored

```
julia> (1+0/2+1/2^2)*2^2
5.0
```

Example

$$1.01 \times 2^2 = \left(1 + \frac{0}{2} + \frac{1}{2^2}\right) \times 2^2 = 5$$

Can always arrange the mantissa so the first digit is non-zero.

$$0.11010 \times 2^0 = 1.10101 \times 2^{-1} = 0.7890625$$

```
julia> (1+1/2+0/2^2+1/2^4+0/2^5+1/2^6)*2^-1
0.7890625
```

prefix means written binary...

$e = 2 \times 0b10$

```
julia> 0b10
0x02
julia> Int(0b10)
2
```

(0x02 ↖ hexidecimal; 2 ↖ decimal)

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
⋮	⋮
1023	1111111111
1020	10000000000

bits = $e + 1023 = 0b10 + 0b1111111111$

10

10000000001

1 2 3 4 5 6 11

```
julia> Int(0b10000000000)
1024
```

← fits in the 11-bits for the exponent...

Next time relative and absolute error. Then overflow.