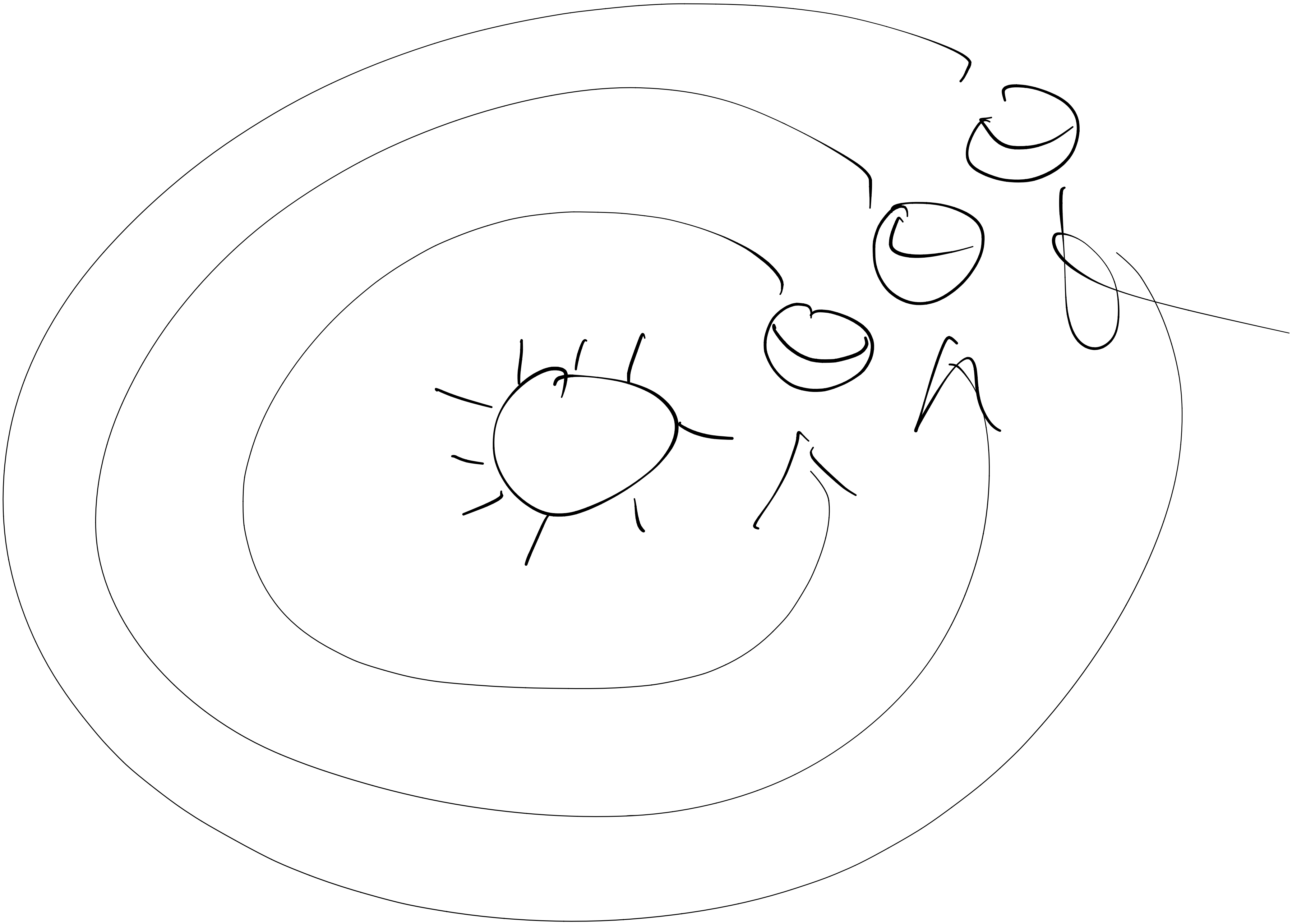
Lecture 17

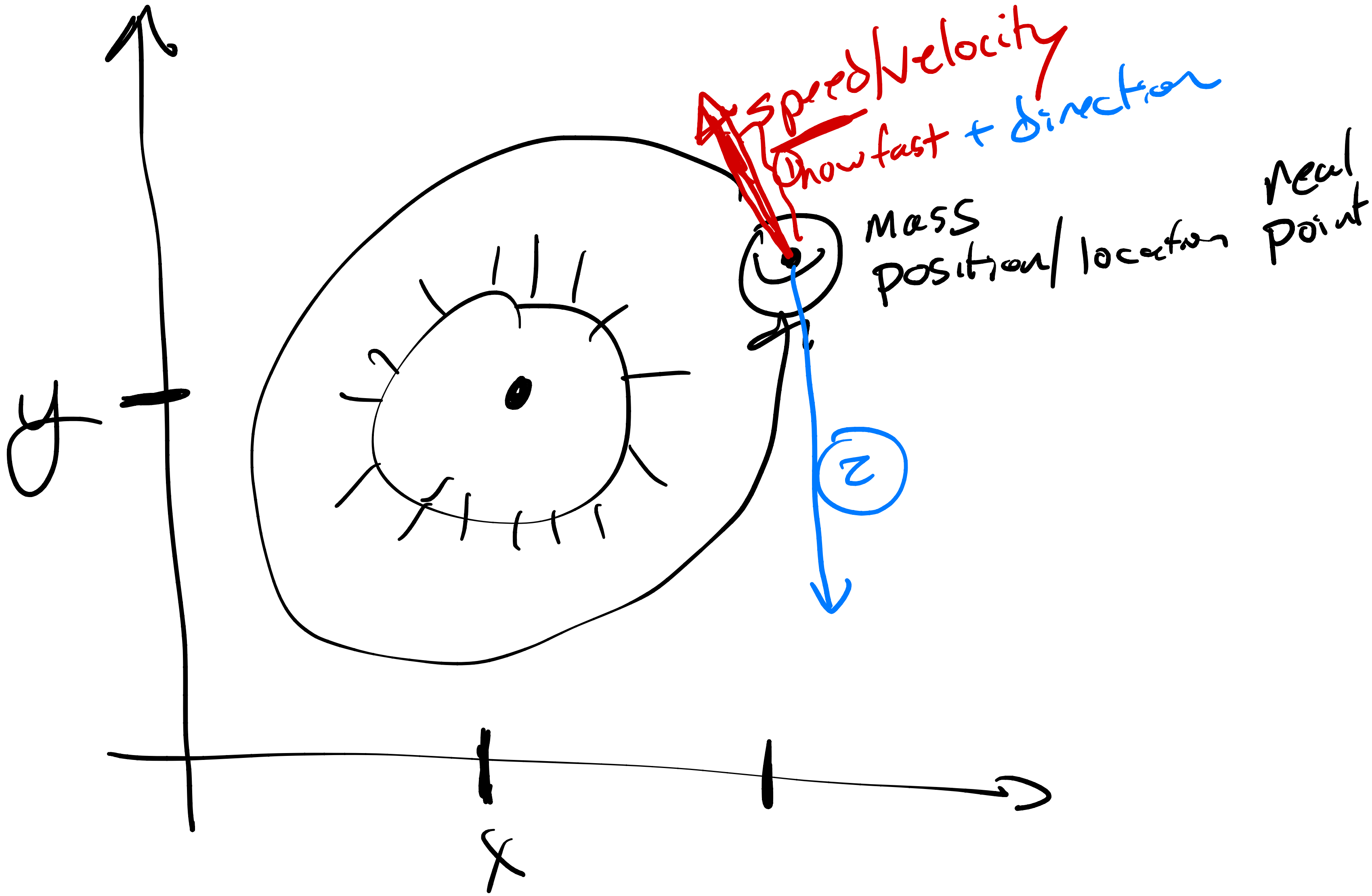
N-body simulation



Simulate motion

vectors

points → (x, y) coords

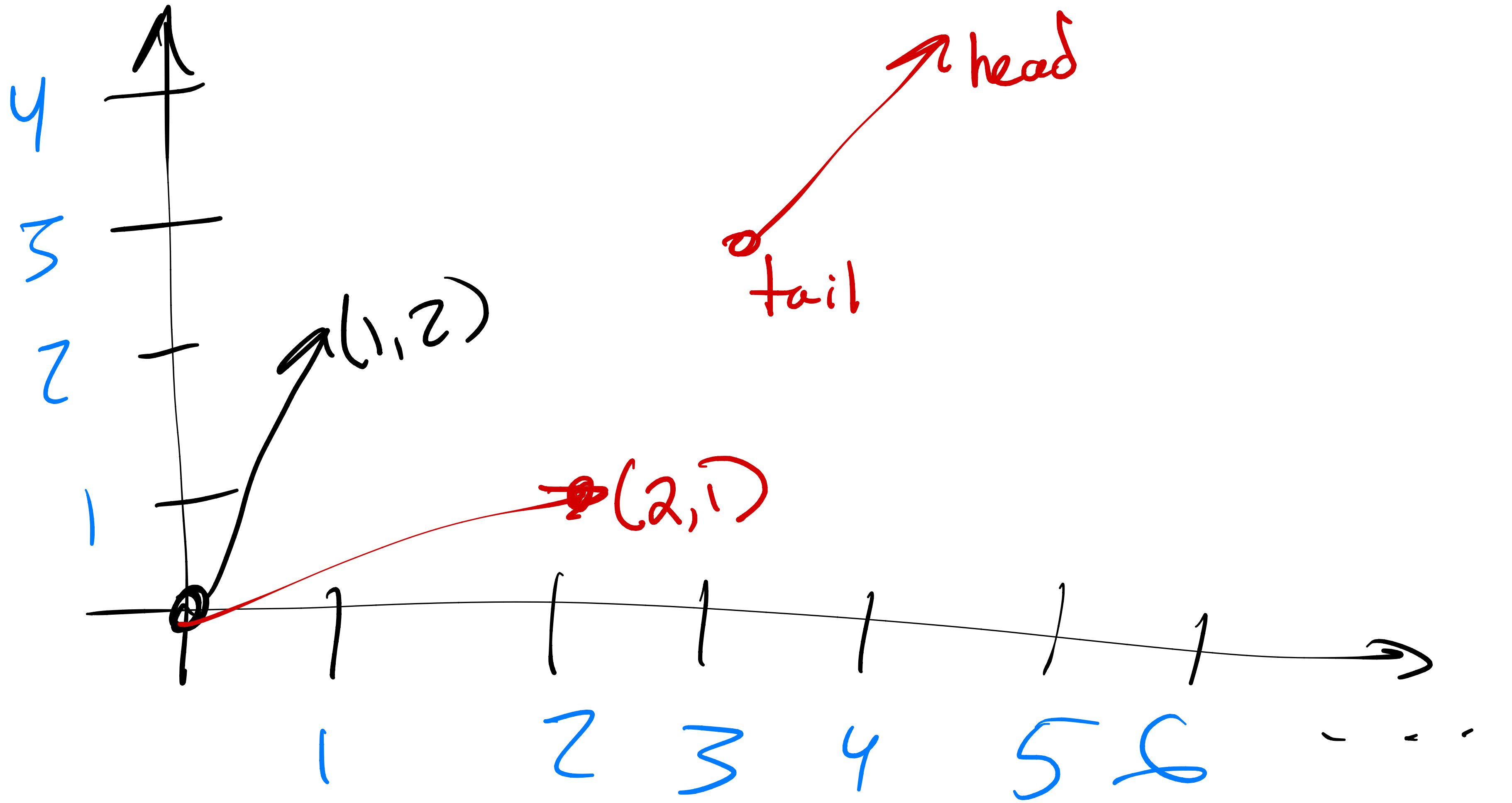


```
(*      (x,y)   coords      *)
type point = real * real


(* represent a vector whose tail is at (0,0) *)
type vec = real * real


(*                  mass    position   velocity (speed) *)
type body = real  * point * vec

type bodies = body Seq.seq
```



(1,2)
(2,1)
head
tail
1
2
3
4
1
2
3
4
5
6

new position is a function of

- old position
- velocity
- accelerations

$$\vec{S}' = \vec{S} + \vec{V} * t + \frac{1}{2}\vec{a} * t^2$$

new pos — old pos — velocity — accel

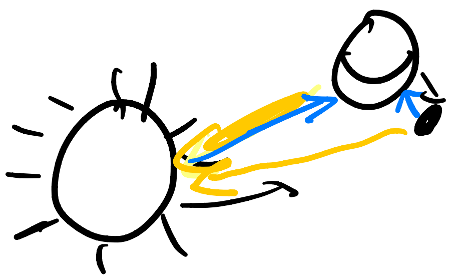
new velocity is a fn of

- old velocity
- accel

$$\vec{V}' = \vec{V} + \vec{a} * t$$

old vel — accel

acceleration is sum of accelerations due to all of the other bodies



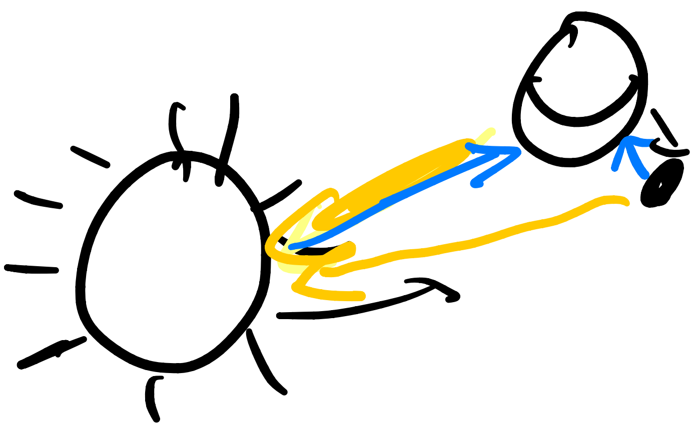
$\vec{a}$ vector

add vectors ?

$$\vec{a}_{earth} = \vec{a}^{sun}_{earth} + \vec{a}^{moon}_{earth}$$

$$\vec{a}_{moon} = \vec{a}^{sun}_{moon} + \vec{a}^{earth}_{moon}$$

$$\vec{a}_{sun} = \vec{a}^{earth}_{sun} + \vec{a}^{moon}_{sun}$$



$$F = ma$$

$$\frac{G m_1 m_2}{d^2} = F$$

G ← gravitational constant

in the direction of: $\hat{dist}_{earth, sun}$

$$\vec{a}^{\,sun}_{earth} = \hat{dist}_{earth,\,sun} * \frac{G * m_{sun}}{|\vec{dist}_{earth,\,sun}|^2}$$

$$\vec{a}^{\,moon}_{earth} = \hat{dist}_{earth,\,moon} * \frac{G\, M_{moon}}{|\vec{dist}_{earth,\,moon}|^2}$$

Library for 2D Vectors

type vec

type point

val ++ : vec * vec → vec

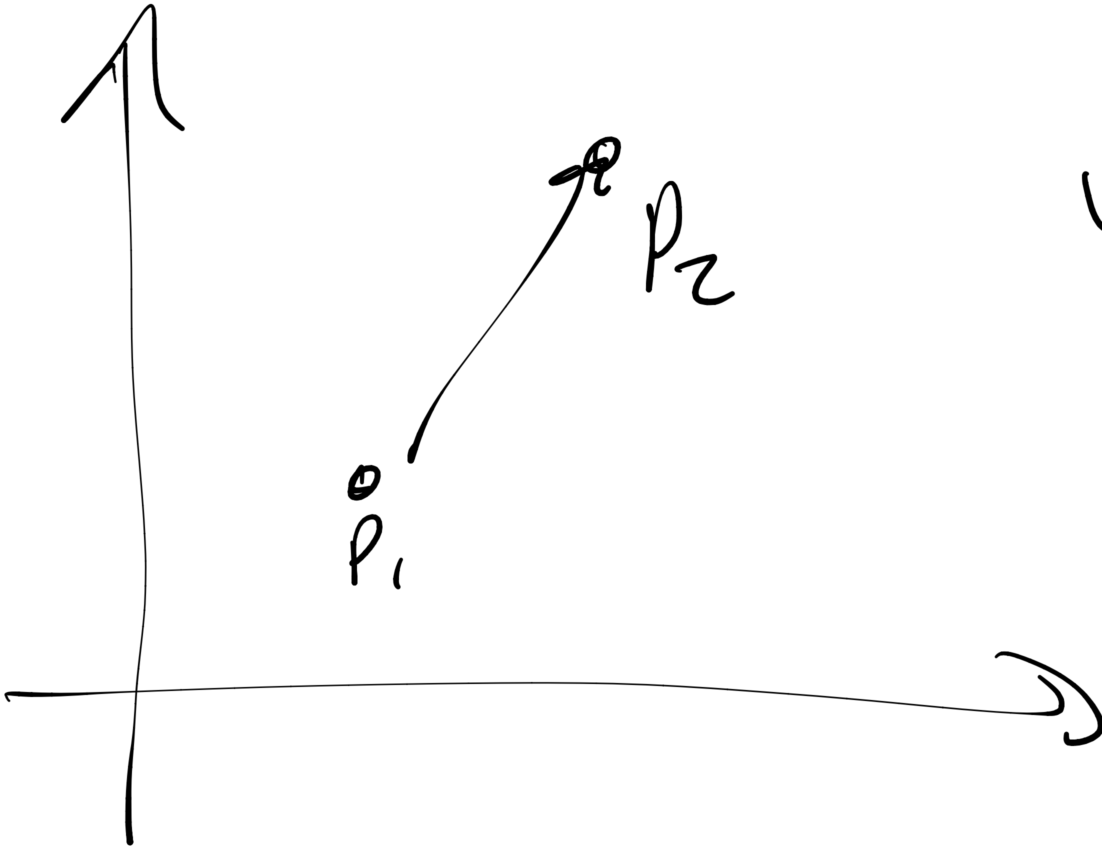
val unitVec : vec → vec (*hat*)

val ** : vec * real → vec

val zero : vec

val mag : vec → real

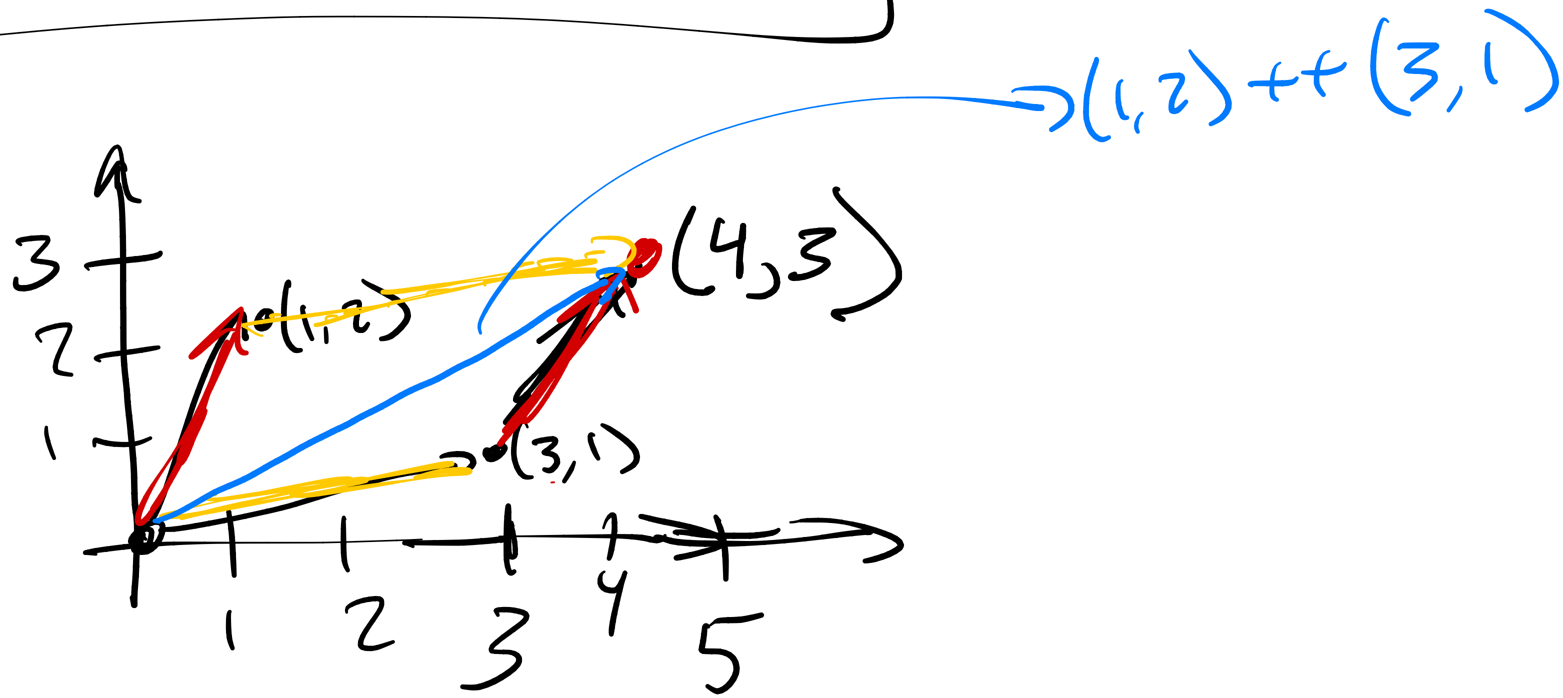
vall --> : point * point → vec



vector from p_1 to p_2

```
fun ((x1, y1) : point) --> ((x2, y2) : point) : vec =
  (x2 - x1, y2 - y1)
```

Adding vectors



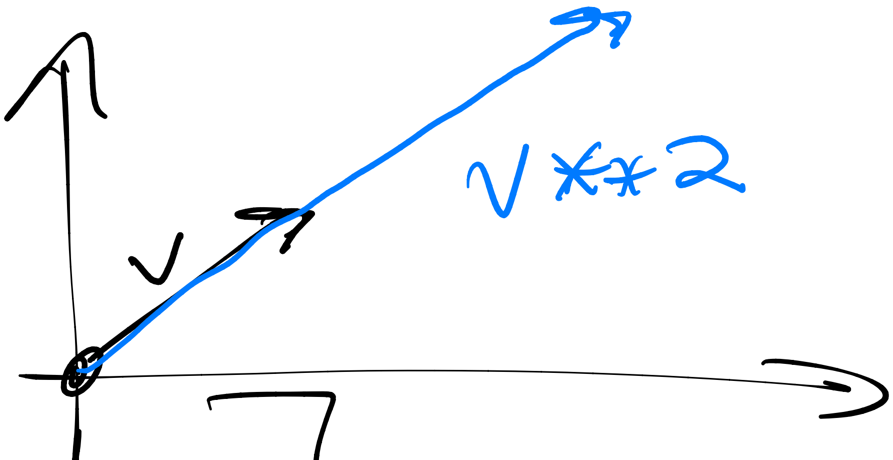
fun ((x1,y1):vec) ++ ((x2,y2):vec):vec =

(x1 + x2 , y1 + y2)

Scaling

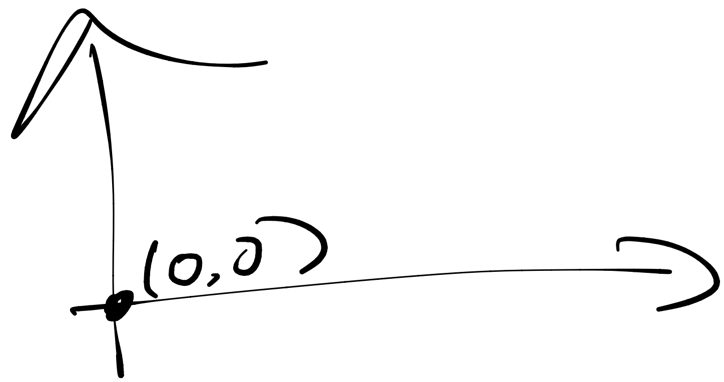
v ** 2.0

times(v, 2.0)




```
[infix ---]
fun ((x,y):vec) ** (s:real) : vec =
    (x*s, y*s)
```

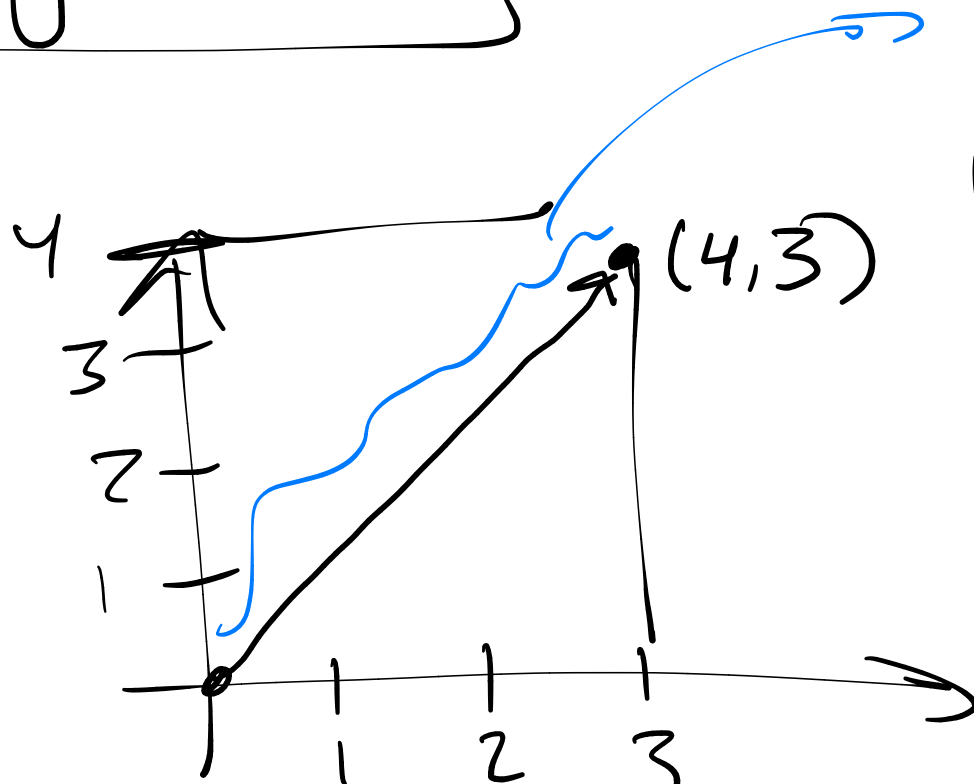
Zero



val zero : vec = (0.0, 0.0)

Magnitude

$\sqrt{3^2+4^2} = 5$

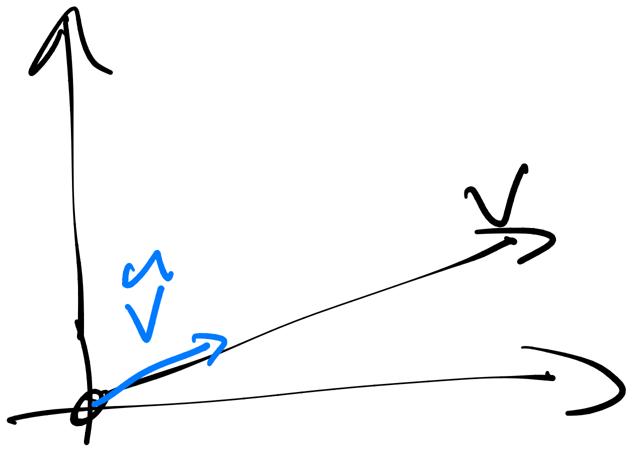


magnitude(v)
is
length of
vec

$\sqrt{x^2+y^2}$

```
fun mag((x,y):vec):real =
    Math.sqrt(x*x + y*y)
```

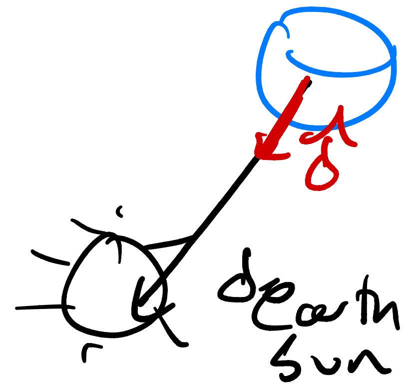
Unit vec



$\hat{v}$ length 1 vector in same dir as v

```
fun unitVec(v : vec) : vec =
    v ** (1.0 / mag v)
```

$$\vec{a}^{sun}_{earth} = \hat{dist}_{earth,sun} * \frac{G * m_{sun}}{|dist_{earth,sun}|^2}$$



m_i p_i v_i

m_j p_j v_j

```
fun accOn ((m_earth, p_earth, v_earth) : body,
           (m_sun, p_sun, v_sun) : body) : vec =
  (if p_earth ≠ p_sun
   then
  let val dearthsun = p_earth --> p_sun in
    unitVec(dearthsun) ** ( G * msun / squared(mag(dearthsun)) )
  end
   else 0.0)
```

```
type vec
type point

val ++ : vec * vec -> vec
val unitVec : vec -> vec
val ** : vec * real -> vec
val zero : vec
val mag : vec -> real
val --> : point * point -> vec
```

$$\vec{a}_{earth} = \vec{a}^{\,sun}_{earth} + + \vec{a}^{\,moon}_{earth} + + \cdots$$

Have n bodies

$$b_1 \quad b_2 \quad b_3 \quad b_4 \cdots \cdots b_n$$

↓

$$\vec{a}_1 \quad \vec{a}_2 \quad \vec{a}_3 \cdots \cdots \vec{a}_n$$

acceleration on all n bodies

```
fun accelerations(bodies: body Seq.seq):
                                vec Seq.seq =

Seq.map(fn body i =>
  Seq.reduce( fn (v1, v2) => v1++v2,
    zero,
    Seq.map(fn bodyj => accOn(bodyi, bodyj)) bodies)
  bodies)
```

acceleration on each body

→ sun on earth ++ → moon on earth ++ ...

i = earth

	input	work	span
inner map	n	$O(n)$	$O(1)$
reduce	n	$O(n)$	$O(\log n)$
inner map then reduce	n	$O(n) + O(n)$ $= O(n)$	$O(1) + O(\log n)$ $= O(\log n)$
map/ overall	n	$O(n^2)$	$O(\log n)$

Approximation

O(n log n) work