Chapter 1

線形代数

線形代数は、高次元に立ち向かうための強力な道具となる。

どれだけ高次元に話を広げたとしても、「関係」を語る言葉の複雑さが増すことはない。
この章では、そんな状況を実現するための理論を追いかけていく。

1.1　ベクトルと座標

1.1.1　移動の表現としてのベクトル

平面上のある点の位置を表すのに、よく使われるのが直交座標である。
直交座標では、x 軸と y 軸を垂直に張り、

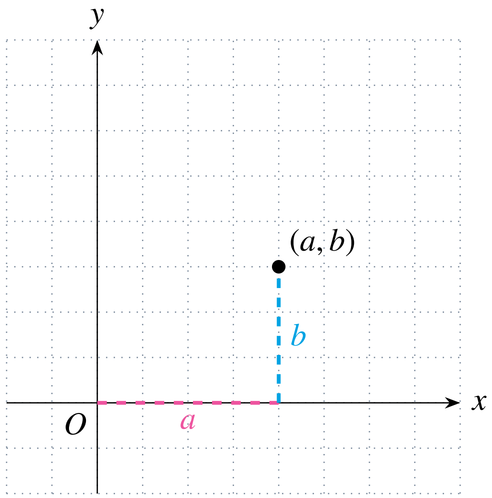
- 原点 O からの x 軸方向の移動量（x 座標）
- 原点 O からの y 軸方向の移動量（y 座標）

という 2 つの数の組で点の位置を表す。

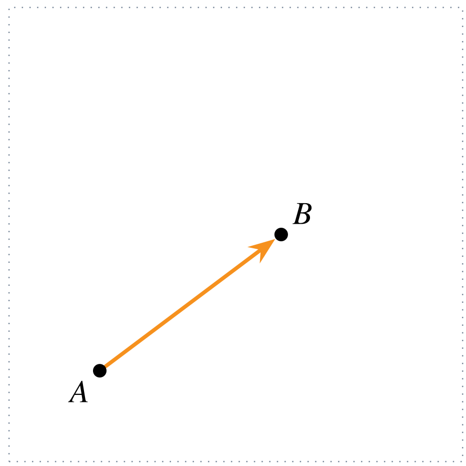
座標とは、「x 軸方向の移動」と「y 軸方向の移動」という 2 回の移動を行った結果である。
右にどれくらい、上にどれくらい、という考え方で平面上の「位置」を特定しているわけだが、単に「移動」を表したいだけなら、点から点へ向かう矢印で一気に表すこともできる。

ある地点から別のある地点への「移動」を表す矢印をベクトルという。

ベクトルが示す、ある地点からこのように移動すれば、この地点にたどり着く…といった「移動」の情報は、相対的な「位置関係」を表す上で役に立つ。



「位置の特定」という視点



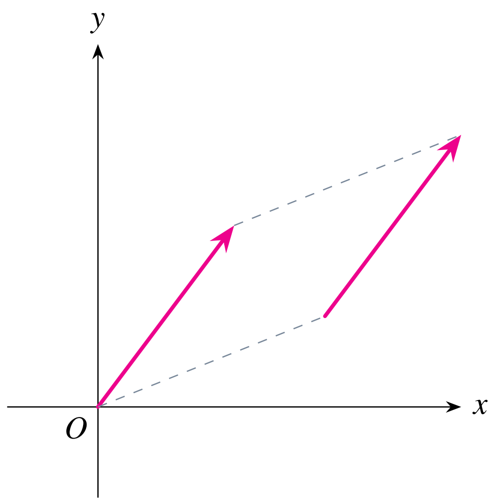
「移動」という視点

平行移動してもベクトルは同じ

座標は「位置」を表すものだが、ベクトルは「移動」を表すものにすぎない。

座標は「原点からの」移動量によって位置を表すが、ベクトルは始点の位置にはこだわらない。

たとえば、次の 2 つのベクトルは始点の位置は異なるが、同じ向きに同じだけ移動している矢印なので、同じベクトルとみなせる。

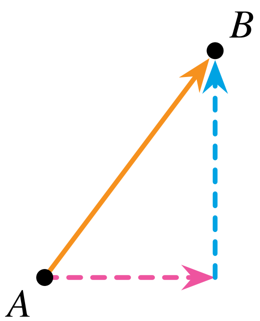


このような「同じ向きに同じだけ移動している矢印」は、平面内では平行な関係にある。

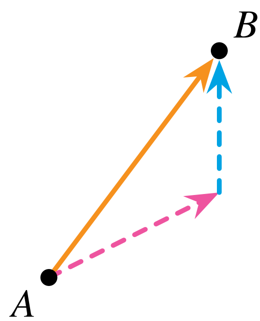
つまり、平行移動して重なる矢印は、同じベクトルとみなすことができる。

移動の合成とベクトルの分解

ベクトルは、各方向への移動の合成として考えることもできる。
純粋に「縦」と「横」に分解した場合は直交座標の考え方によく似ているが、必ずしも直交する方向のベクトルに分解する必要はない。



「縦」と「横」に分解

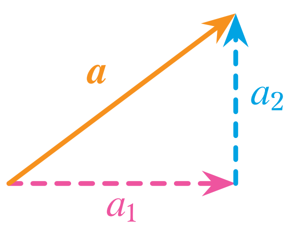


他の分解も考えられる

1.1.2　高次元への対応：数ベクトル

2 次元以上の空間内の「移動」を表すには、「縦」と「横」などといった 2 方向だけでなく、もっと多くの方向への移動量を組み合わせて考える必要がある。
また、4 次元を超えてしまうと、矢印の描き方すら想像がつかなくなってしまう。それは、方向となる軸が多すぎて、どの方向に進むかを表すのが難しくなるためだ。

そこで、一旦「向き」の情報を取り除くことで、高次元に立ち向かえないかと考える。
移動を表す矢印は「どの方向に進むか」と「どれくらい進むか」という向きと大きさの情報を持っているが、その「どれくらい進むか」だけを取り出して並べよう。



$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

こうして単に「数を並べたもの」もベクトルと呼ぶことにし、このように定義したベクトルを数ベクトルという。

数を並べるとき、縦と横の 2 通りがある。それぞれ列ベクトル、行ベクトルとして定義する。

列ベクトル　数を縦に並べたものを 列ベクトル という。

$$\boldsymbol{a} = [a_i] = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

行ベクトル　数を横に並べたものを 行ベクトル という。

$$\boldsymbol{a} = [a_i] = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}$$

単に「ベクトル」と言った場合は、列ベクトルを指すことが多い。

行ベクトルは、列ベクトルを横倒しにしたもの（列ベクトルの転置）と捉えることもできる。

転置による行ベクトルの表現

行ベクトルは、列ベクトル $\boldsymbol{a}$ を 転置 したものとして表現できる。

$$\boldsymbol{a}^\top = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}$$

1.1.3　ベクトルの和

ベクトルによって数をまとめて扱えるようにするために、ベクトルどうしの演算を定義したい。

ベクトルどうしの足し算は、同じ位置にある数どうしの足し算として定義する。

ベクトルの和 2 つの n 次元ベクトル $\boldsymbol{a}$ と $\boldsymbol{b}$ の和を次のように定義する。

$$\boldsymbol{a}+\boldsymbol{b}=[a_i]+[b_i]=\begin{bmatrix} a_1+b_1 \\ a_2+b_2 \\ \vdots \\ a_n+b_n \end{bmatrix}$$

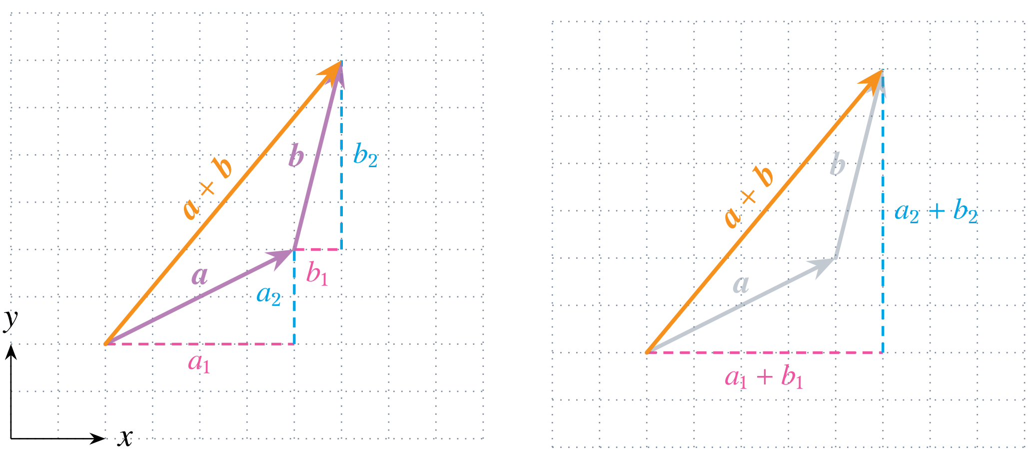
i 番目の数が $\boldsymbol{a}$ と $\boldsymbol{b}$ の両方に存在していなければ、その位置の数どうしの足し算を考えることはできない。

そのため、ベクトルの和が定義できるのは、同じ次元を持つ（並べた数の個数が同じ）ベクトルどうしに限られる。

移動の合成としてのイメージ

数ベクトルを「どれくらい進むか」を並べたものと捉えると、同じ位置にある数どうしを足し合わせるということは、同じ向きに進む量を足し合わせるということになる。

たとえば、x 軸方向に a_1、y 軸方向に a_2 進んだ場所から、さらに x 軸方向に b_1、y 軸方向に b_2 進む…というような「移動の合成」を表すのが、ベクトルの和である。



平行四辺形の法則

[Todo 1: 平行移動しても同じベクトルなので…]

ベクトルの差：逆向きにしてから足す

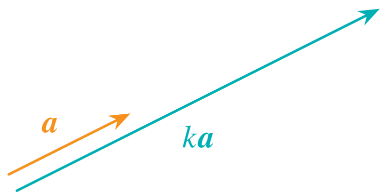
[Todo 2: irobutsu-linear-algebra 2.1.2 ベクトルの差]

矢に沿った移動で考える

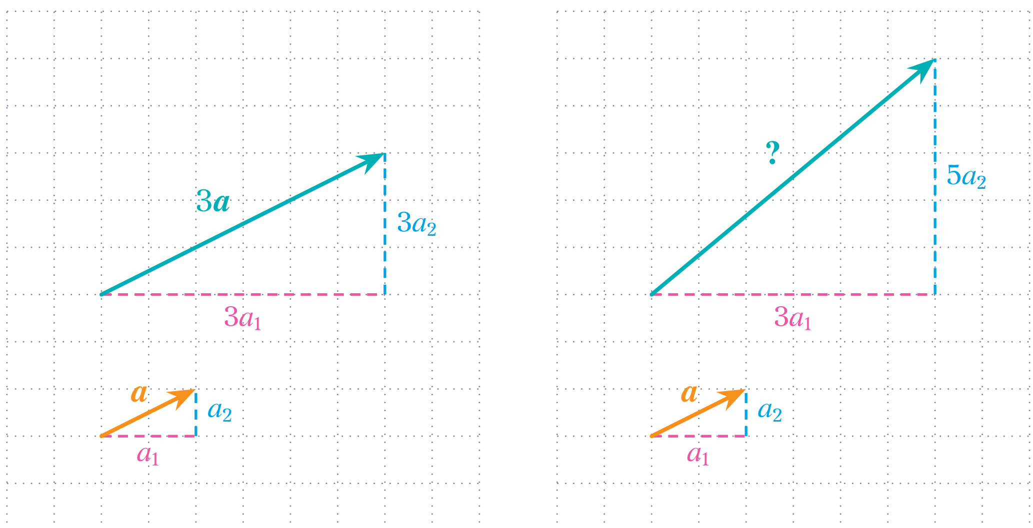
[Todo 3: 手持ちの画像を参考に、和と差の両方について書く]

1.1.4 ベクトルのスカラー倍

「どれくらい進むか」を表す数たち全員に同じ数をかけることで、向きを変えずにベクトルを「引き伸ばす」ことができる。



ここで向きごとにかける数を変えてしまうと、いずれかの方向に多く進むことになり、ベクトルの向きが変わってしまう。そのため、「同じ」数をかけることに意味がある。



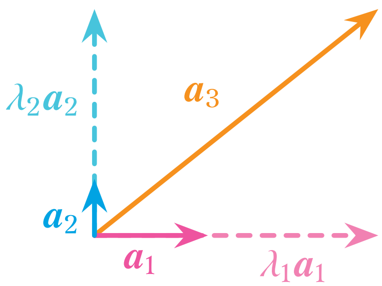
そこで、ベクトルの定数倍（スカラー倍）を次のように定義する。

ベクトルのスカラー倍　n 次元ベクトル $\boldsymbol{a}$ の k 倍を次のように定義する。

$$k\boldsymbol{a} = k[a_i] = \begin{bmatrix} ka_1 \\ ka_2 \\ \vdots \\ ka_n \end{bmatrix}$$

1.1.5　一次結合

ベクトルを「引き伸ばす」スカラー倍と、「つなぎ合わせる」足し算を組み合わせることで、あるベクトルを他のベクトルを使って表すことができる。



$$\boldsymbol{a}_3 = \lambda_1\boldsymbol{a}_1 + \lambda_2\boldsymbol{a}_2$$

このように、スカラー倍と和のみを使った形を一次結合もしくは線形結合という。

1.1.6　基底：座標を復元する

3 次元までのベクトルは、矢印によって「ある点を指し示すもの」として定義できる。

しかし、4 次元以上の世界に話を広げるため、ベクトルを単に「数を並べたもの」として再定義した。「数を並べたもの」としてのベクトルを、数ベクトルと呼んでいる。

さて、2 次元平面や 3 次元空間で点を指し示すためのもう一つの概念として、座標がある。

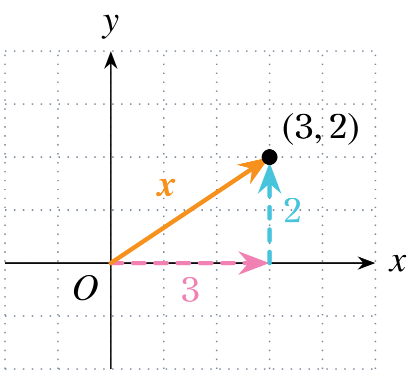
座標は、x 軸方向にこのくらい進み、y 軸方向にこのくらい進む…というように、「進む方向」と「進む長さ」によって表現される。

単なる数の並びである数ベクトルでは、「進む方向」については何も記述されていない。

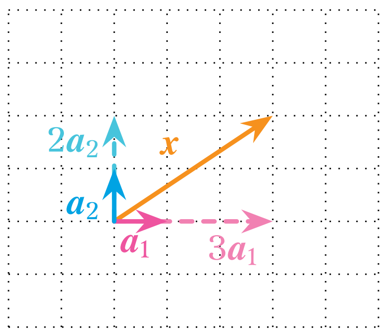
$$\begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

しかし、「進む方向」を表すベクトル $\boldsymbol{a}_1, \boldsymbol{a}_2$ を新たに用意すれば、一次結合によって「進む方向」と「進む長さ」を持つベクトルを作ることができる。

$$\boldsymbol{x} = 3\boldsymbol{a}_1 + 2\boldsymbol{a}_2$$



$$(3, 2)$$



$$\boldsymbol{x} = 3\boldsymbol{a}_1 + 2\boldsymbol{a}_2$$

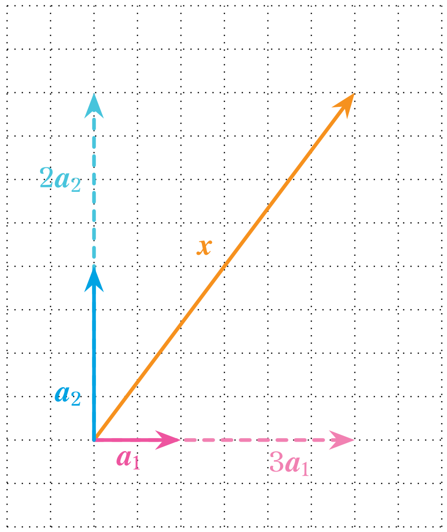
$\boldsymbol{a}_1$ と $\boldsymbol{a}_2$ のように、座標を復元するために向きの情報を付け加えるベクトルを、基底と呼ぶことにする。(「基底」と呼ぶための条件はいろいろあるが、それについては後々解説していく。)

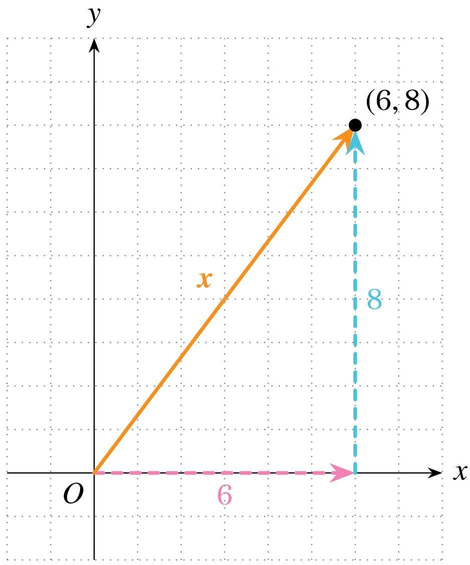
基底が変われば座標が変わる

先ほどの例では、直交座標による点 $(3, 2)$ をベクトルの一次結合 $\boldsymbol{x} = 3\boldsymbol{a}_1 + 2\boldsymbol{a}_2$ で表現するために $\boldsymbol{a}_1$ と $\boldsymbol{a}_2$ を用意した。

$\boldsymbol{a}_1$ を x 軸方向の長さ 1 のベクトル、$\boldsymbol{a}_2$ を y 軸方向の長さ 1 のベクトルとすれば、$\boldsymbol{a}_1$ を 3 倍、$\boldsymbol{a}_2$ を 2 倍して足し合わせることで、点 $(5, 4)$ を指し示すベクトル $\boldsymbol{x}$ を作ることができる。

ここで、一次結合の式 $\boldsymbol{x} = 3\boldsymbol{a}_1 + 2\boldsymbol{a}_2$ は変えずに、$\boldsymbol{a}_1$ と $\boldsymbol{a}_2$ を変更すると、$\boldsymbol{x}$ が指し示す点も変わってしまう。





$$\boldsymbol{x} = 3\boldsymbol{a}_1 + 2\boldsymbol{a}_2$$

$$(6, 8)$$

このことから、

座標は使っている基底の情報とセットでないと意味をなさない

ものだといえる。

1.2 基底にできるベクトルを探す

1.3 線形性を保つ空間

1.4 ベクトルの測り方

1. 内積
2. ノルム

1.5 直交するベクトル

1. 直交基底

Zebra Notes

Type	Number
todo	3