

数学クォータ科目「応用解析」第 1 回 / ベクトル解析 (1)

# ベクトル関数の微分と積分

佐藤 弘康 / 日本工業大学 共通教育学群

# イントロダクション)「応用解析」について

この科目では次の3つのテーマについて学習する。

- (1) ベクトル解析
- (2) 複素関数論 (または, 複素解析)
- (3) 微分方程式

- 解析学  $\equiv$  微分積分学

つまり, 3つとも「微分積分学」と関連。

- 「微分積分学」の対象は関数。これまで扱っていた関数は, 実数値関数
  - 1変数関数  $y = f(x)$  (「基礎数学 I」「II」または, 高校数学)  
変数  $x$  の値を定めると, 変数  $y$  の値がただひとつ決まる。
  - 2変数関数  $z = f(x, y)$  (「数学」)  
変数  $x, y$  の値を定めると, 変数  $z$  の値がただひとつ決まる。

# イントロダクション)「応用解析」について

---

この科目では次の関数を扱う．

## (1) ベクトル解析

- (1 変数) ベクトル関数 : 1 変数関数の三つ組
- スカラー場 : 3 変数関数
- ベクトル場 : 3 変数関数の三つ組
- (2 変数) ベクトル関数 : 2 変数関数の三つ組

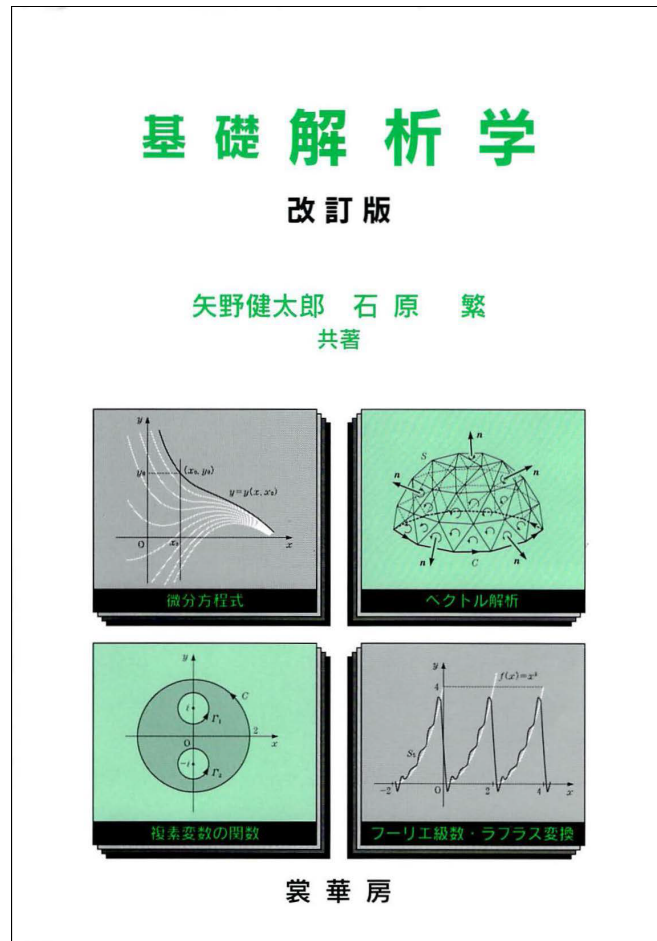
## (2) 複素関数論

- 複素関数 (複素 1 変数複素数値関数) : 2 変数関数の組
- 複素平面内の曲線 (実 1 変数複素数値関数) : 1 変数関数の組

## (3) 微分方程式

- 1 変数関数

# 「応用解析」の教科書・参考書



# 今回の授業で理解してほしいこと

---

- (1変数) ベクトル値関数 の
  - 概念とその幾何学的な解釈
  - 微分の計算と微分の幾何学的な解釈
  - 不定積分と定積分の計算

# 【復習 1】（空間）ベクトルとは

- 有向線分 線分の両端点に「始点」と「終点」の情報を付与したモノ。
  - 2つの有向線分が平行移動により始点と終点が重なるならば, それらを同じモノとみなす. これを **ベクトル** とよぶ.
- ベクトル  $A$  に対し,  $A = \overrightarrow{OP}$  となるような点  $P(l, m, n)$  がひとつ定まる.  
※始点が原点  $O$  になるように  $A$  を平行移動したときの終点が  $P$ .
- ベクトルの表し方
  - 成分表示:  $P$  の座標で  $A$  を表したものの.  $A = (l, m, n)$
  - 基本ベクトル表示: ベクトルの線形演算 (和とスカラー倍) の性質から,  $A = li + mj + nk$  と書ける.
- 上の対応により, 以後は点とベクトルを同一視して扱うことがある.

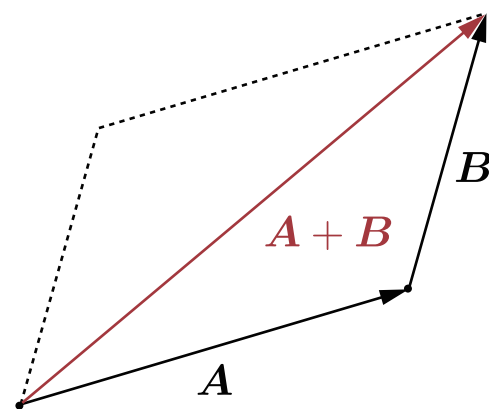
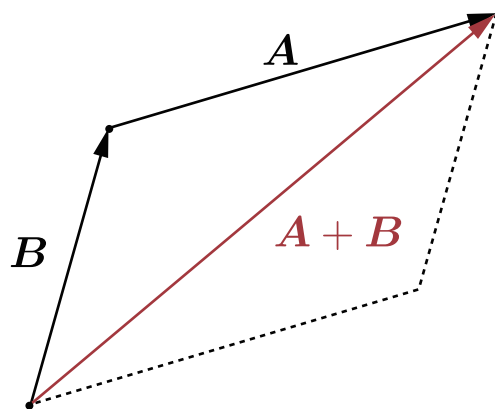
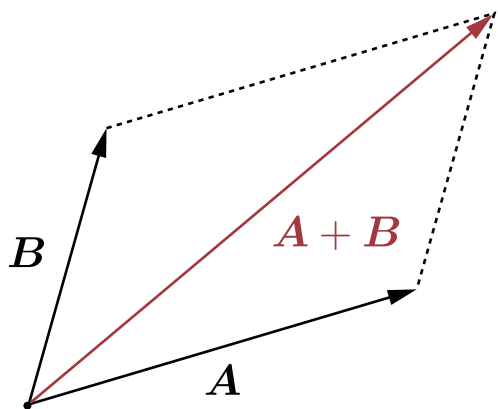
$$P(l, m, n) \leftrightarrow A = (l, m, n) \leftrightarrow A = li + mj + nk$$

## 【復習 2】 ベクトルの大きさ，線形演算

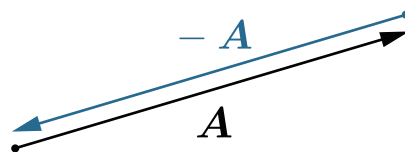
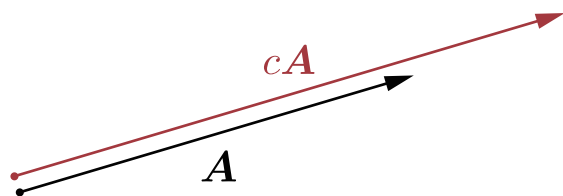
- ベクトル  $A = li + mj + nk$  の**大きさ**；  $|A| = \sqrt{l^2 + m^2 + n^2}$

※線分としての長さのこと.

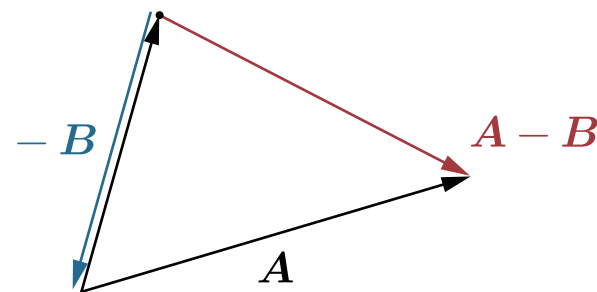
- 和**  $A + B$



- スカラー倍**  $cA$



- 差**  $A - B = A + (-B)$



# (1変数) ベクトル関数

## 定義

変数  $t$  の値を決めると、その値に応じてベクトル  $A(t)$  がただ一つ定まるとき、 $A(t)$  を独立変数  $t$  のベクトル関数という。

- $A(t)$  はベクトルなので、基本ベクトル表示が得られる；

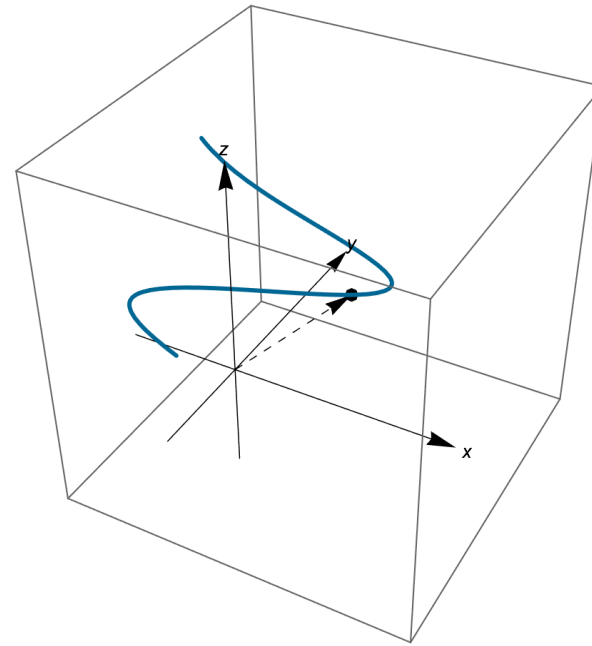
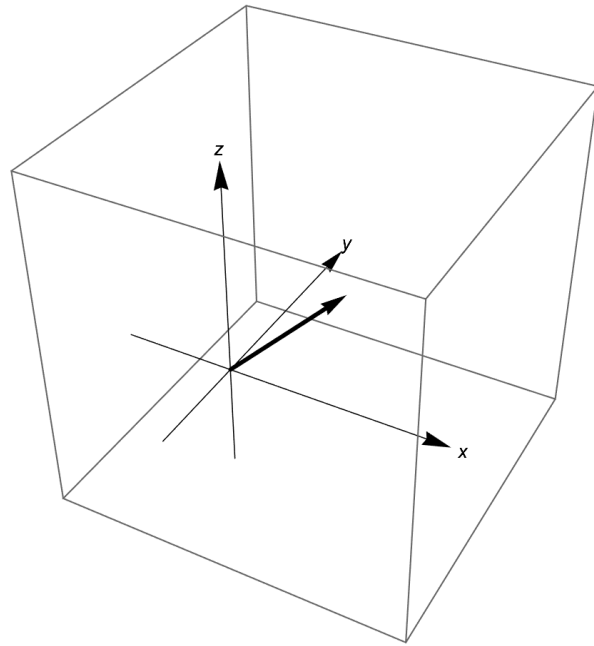
$$A(t) = A_x(t) \mathbf{i} + A_y(t) \mathbf{j} + A_z(t) \mathbf{k}$$

- つまり、ベクトル関数  $A(t)$  を考えることは、  
3つの1変数関数の組  $A_x(t), A_y(t), A_z(t)$  を考えることと同じである。
- $A_x(t), A_y(t), A_z(t)$  のことをベクトル関数  $A(t)$  の成分とよぶ。



# (1変数) ベクトル関数

例1)  $A(t) = \sin 3t \, i + \cos t \, j + \frac{1}{t+1} \, k$



- ベクトル関数  $A(t)$  の始点を定点  $O$  に固定すれば,  $A(t)$  の終点  $P$  は一般に1つの曲線を描く. この曲線を  $A(t)$  の **ホドグラフ** という.

# (1変数) ベクトル関数

例2) 成分が独立変数  $t$  の1次関数のベクトル関数；

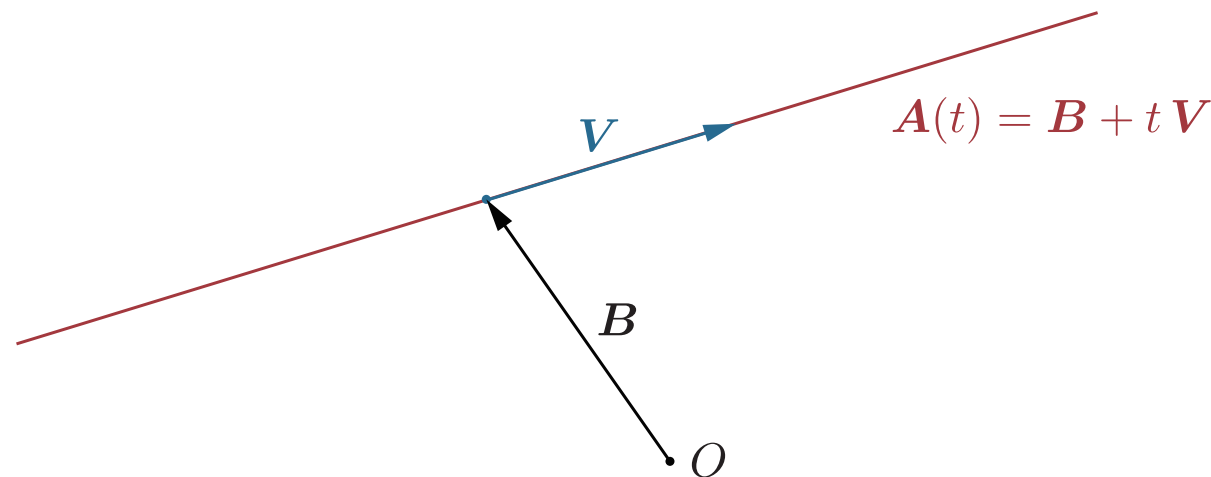
$$A(t) = (b_1 + v_1 t) \mathbf{i} + (b_2 + v_2 t) \mathbf{j} + (b_3 + v_3 t) \mathbf{k}$$

- これは

$$A(t) = (b_1 \mathbf{i} + b_2 \mathbf{j} + b_3 \mathbf{k}) + t (v_1 \mathbf{i} + v_2 \mathbf{j} + v_3 \mathbf{k}) = \mathbf{B} + t \mathbf{V}$$

と書ける. ただし,  $\mathbf{B} = b_1 \mathbf{i} + b_2 \mathbf{j} + b_3 \mathbf{k}$ ,  $\mathbf{V} = v_1 \mathbf{i} + v_2 \mathbf{j} + v_3 \mathbf{k}$ .

- このベクトル関数のホドグラフは, 直線である.



# (1変数) ベクトル関数の微分 (導関数)

## 定義

ベクトル関数  $A(t) = A_x(t) \mathbf{i} + A_y(t) \mathbf{j} + A_z(t) \mathbf{k}$  に対し,

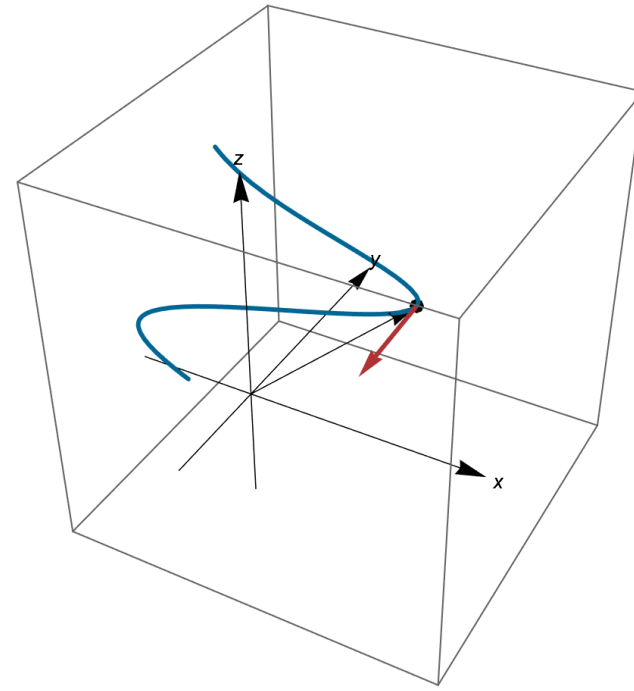
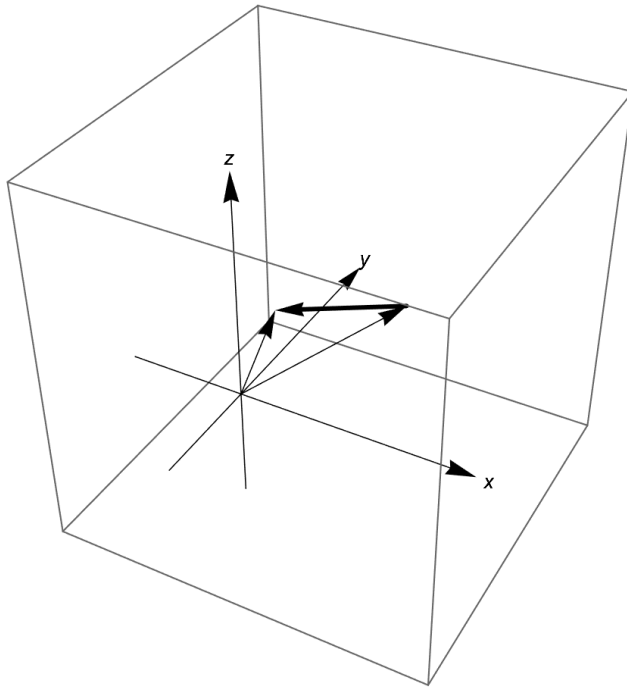
$$\lim_{h \rightarrow 0} \frac{A(t+h) - A(t)}{h}$$

を  $A(t)$  の微分または導関数とよび,  $A'(t)$  や  $\frac{dA}{dt}(t)$  と表す.

**計算方法** 各成分を微分すればよい.

$$\begin{aligned} \frac{A(t+h) - A(t)}{h} &= \frac{A_x(t+h) - A_x(t)}{h} \mathbf{i} + \frac{A_y(t+h) - A_y(t)}{h} \mathbf{j} + \frac{A_z(t+h) - A_z(t)}{h} \mathbf{k} \\ &\xrightarrow{h \rightarrow 0} A'_x(t) \mathbf{i} + A'_y(t) \mathbf{j} + A'_z(t) \mathbf{k} \end{aligned}$$

# (1変数) ベクトル関数の微分の幾何学的な解釈



- $A'(t)$  はベクトル関数  $A(t)$  の ホドグラフに接するベクトル である.

# (1 変数) ベクトル関数の積分

$A(t) = A_x(t) \mathbf{i} + A_y(t) \mathbf{j} + A_z(t) \mathbf{k}$  をベクトル関数とする.

- $D(t)$  の導関数が  $A(t)$  であるとき,  $D(t)$  を  $A(t)$  の不定積分といい, 次のような記号で表す ;

$$D(t) = \int A(t) dt$$

- 実際の計算は  $\int A(t) dt = \int A_x(t) dt \mathbf{i} + \int A_y(t) dt \mathbf{j} + \int A_z(t) dt \mathbf{k}$
- $A(t)$  の定義域内の区間  $a \leq t \leq b$  に対して, 1 変数関数の場合と同様にしてリーマン和の極限として, 定積分  $\int_a^b A(t) dt$  が定義できる.
- 実際の計算は  $\int_a^b A(t) dt = \int_a^b A_x(t) dt \mathbf{i} + \int_a^b A_y(t) dt \mathbf{j} + \int_a^b A_z(t) dt \mathbf{k}$

# まとめと復習（と予習）

---

- ベクトル関数とは何ですか？
  - ベクトル関数のホドグラフとは何ですか？
  - ベクトル関数の微分（導関数）はどのように計算しますか？
  - ベクトル関数の微分（導関数）はどのようなベクトルですか？
  - ベクトル関数の不定積分はどのように計算しますか？
  - ベクトル関数の定積分はどのように計算しますか？

教科書 p.73～77

問題集 187, 188, 189

予 習 ベクトルの内積 「基礎数学Ⅰ」, 合成関数の微分 「数学」