No.10 積分と数値計算

NAKADA Masayuki

Kobe University Secondary School

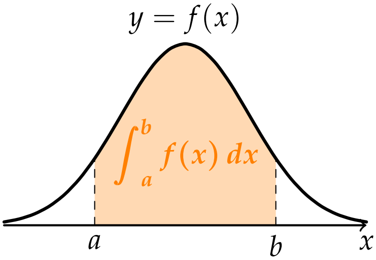
October 3, 2021

積分

関数 $y=f(x)$ の $a \leqq x \leqq b$ の部分と x 軸とで挟まれた部分の（符号付き）面積を，$f(x)$ の区間 $[a,b]$ における**定積分**といい，

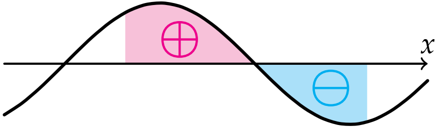
$$\boxed{\int_a^b f(x)\,dx}$$

で表す．



定積分において，関数 $f(x)$ を**被積分関数**，区間 $[a,b]$ を**積分区間**，a を**下端**，b を**上端**という．

※ 面積が「符号付き」であるとは，関数のグラフが x 軸より上方にある部分については**正**の面積，グラフが x 軸より下方にある部分については**負**の面積として扱うという意味である．

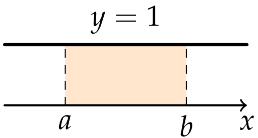


また，積分区間には向きがあるとみなすため，上端と下端が逆になると，定積分の値は (-1) 倍となる: $\int_b^a = -\int_a^b$.

このような現象は，後述の定積分の定義によるものである．一見これまでの知識からは不自然とも思えるが，負の面積を許容しておくことで様々な計算を「自然に」行えるようになる．

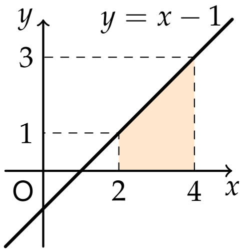
e.g.

1. 定数関数 $f(x) = 1$:
定数関数においては，高さが一定である長方形の面積を求めることになる．よって



$$\left(\int_a^b dx =\right) \int_a^b 1\,dx = (b-a)\cdot 1 = b-a$$

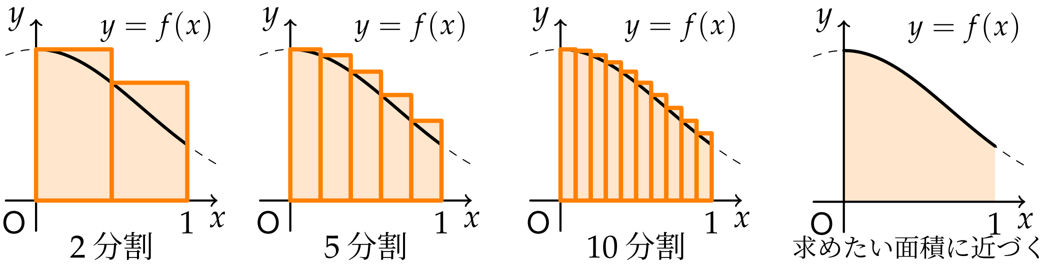
2. 1 次関数 $f(x) = x - 1$:
1 次関数においては，求める面積は図のような台形や三角形を組み合わせたものとなる．よって



$$\int_2^4 (x-1)\,dx = \frac{1}{2}\cdot(1+3)\cdot 2 = 4$$

グラフが複雑な形であっても，基本は上記の操作を洗練させて求積する．

グラフが曲線を描く場合，積分区間を分割して，長方形の和に近似して求める.

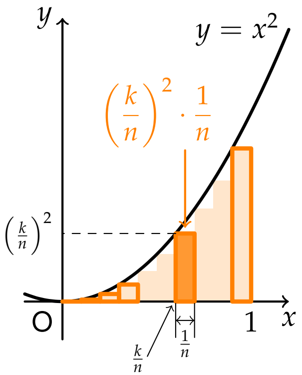


$$f(0)\frac{1}{2}+f\left(\frac{1}{2}\right)\frac{1}{2} \qquad f(0)\frac{1}{5}+\cdots+f\left(\frac{4}{5}\right)\frac{1}{5} \qquad f(0)\frac{1}{10}+\cdots+f\left(\frac{9}{10}\right)\frac{1}{10} \qquad \to \int_0^1 f(x)\,dx$$

このような求積の方法を**区分求積法**という.

e.g. $\int_0^1 x^2\,dx$ の値を，区分求積法の考え方で求める．

区間 $[0,1]$ を n 等分（右図）すると，長方形の面積の和は



$$0^2\cdot\frac{1}{n}+\left(\frac{1}{n}\right)^2\cdot\frac{1}{n}+\left(\frac{2}{n}\right)^2\cdot\frac{1}{n}+\cdots+\left(\frac{n-1}{n}\right)^2\cdot\frac{1}{n}$$

$$=\frac{1}{n^3}\left(0^2+1^2+\cdots+(n-1)^2\right)=\frac{1}{n^3}\cdot\frac{1}{6}(n-1)n(2n-1)$$

$$=\frac{1}{6}\left(1-\frac{1}{n}\right)\left(2-\frac{1}{n}\right)$$

上式において，n の値が膨らむと，$\frac{1}{n}$ の影響は無視できるほど小さくなる．よって，区間分割の細分化の極限として

$$\int_0^1 x^2\,dx=\lim_{n\to\infty}\frac{1}{6}\left(1-\frac{1}{n}\right)\left(2-\frac{1}{n}\right)=\frac{1}{6}\cdot1\cdot2=\mathbf{\frac{1}{3}}$$

一般の関数 $f(x)$ についても，積分区間 $[a,b]$ の任意の分割の細分による右辺の極限値が一意的に定まる場合には，同様の手順で積分を定義される．

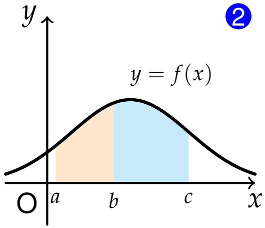
Riemann 積分（参考）

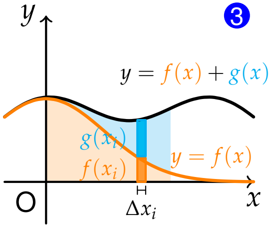
$$\int_a^b f(x)\,dx = \lim_{\text{区間細分}} \sum_{k=0}^{n-1} f(\xi_k)(x_{k+1} - x_k)$$

※ ここで $\lim_{\text{区間細分}}$ とは，各 n について，積分区間 $[a,b]$ の n 個の小区間 $[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]$ への分割（つまり $a = x_0 < x_1 < x_2 < \cdots < x_n = b$）と，各区間の値 $\xi_k \in [x_k, x_{k+1}]$ についての和 $\sum f(\xi_k)(x_{k+1} - x_k)$ をとり，その極限を考えることを表すのであるが，区間の細分全体には「束」と呼ばれる構造が入っており，極限のとり方には注意を要する．

積分について，以下の計算法則がよく用いられる．

1. $$\int_b^a f(x)\,dx = -\int_a^b f(x)\,dx$$
2. $$\int_a^b f(x)\,dx + \int_b^c f(x)\,dx = \int_a^c f(x)\,dx$$
3. $\alpha, \beta \in \mathbb{R}$ を定数とするとき
$$\int_a^b (\alpha f(x) + \beta g(x))\,dx = \alpha \int_a^b f(x)\,dx + \beta \int_a^b g(x)\,dx$$





また，被積分関数の x は積分の中では束縛変数であるから，次が成り立つ．

$$\int_a^b f(x)\,dx = \int_a^b f(y)\,dy = \int_a^b f(t)\,dt$$

数値計算

実際に積分値を計算するには，様々なテクニックを駆使する必要があったり，そもそも初等的には不可能である場合が多い（それゆえに豊富な数学の理論が発展してきたのであるが）.
ここでは，区分求積法の考え方に python を利用して定積分の近似値を計算しよう.
手順は，以下の通りである.

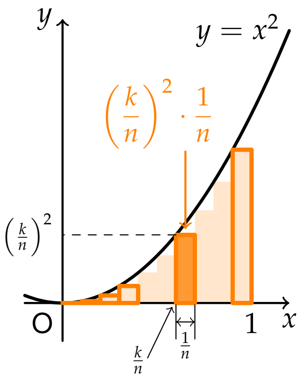
1. 関数を定義する.
2. 積分区間を指定する.
3. 積分値の分割方法を指定する.
4. 積分値を長方形の和として計算する.

e.g. 関数 $f(x) = x^2$ の区間 $[0,1]$ における定積分の計算

コード: squared.py

```
squared = lambda x: x**2 # 関数の定義
bottom, top = 0.0, 1.0 # 積分区間の下端と上端
divnum = 10 # 積分区間の等分数

integral_left = 0.0 # 積分値の初期値
interval = top - bottom # 積分区間の長さ
for i in range(divnum):
  integral_left += \
  squared(bottom + interval * i / divnum) \
  * (interval / divnum)
print(integral_left)
```



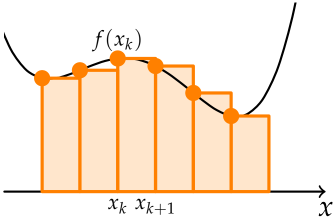
`divnum` の値を `100`，`1000`，... と入れ替えて，理論値（1/3）との差を観察してみよ．

コンピュータによる定積分の区分求積による計算は，あくまで有限分割による近似計算であるから，誤差が生ずることは避けられない．計算の精度を高めるためには，いくつかの方向性が考えられる．

- **積分区間の分割の数を増やす**:
  分割の数を増やすことには，一定の効果がある．しかし，分割を増やせばコンピュータの計算量が増え，コンピュータへの負荷が増大する．オーバーフローの危険もある．区間の分割はほどほどに留めておくべき．
- **近似の方法を工夫する**:
  分割の形状を長方形でなく，他の図形によって行うことも考えられる．詳細は以降のスライドで解説する．
- **その他，個別具体的な工夫**:
  限られたリソースによる積分値の計算の精度を高めることは，それだけで数値計算の数学の一分野として活発に研究されている．変数変換（置換積分）を初めとするさまざまな気の利いた手法が提案されている．

左端点則

$$\sum_{k=0}^{n-1} f(x_k)(x_{k+1} - x_k)$$

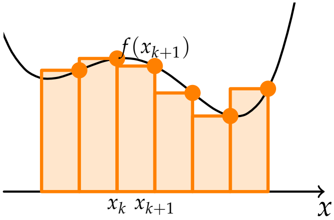


コード: left.py

```
for k in range(divnum):
  integral_left += \
  function(bottom + interval * k / divnum) \
  * (interval / divnum)
```

右端点則

$$\sum_{k=0}^{n-1} f(x_{k+1})(x_{k+1} - x_k)$$

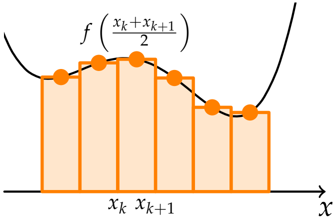


コード: right.py

```
for k in range(divnum):
  integral_left += \
  function(bottom + interval * (k+1) / divnum) \
  * (interval / divnum)
```

中点則

$$\sum_{k=0}^{n-1} f\left(\frac{x_k + x_{k+1}}{2}\right)(x_{k+1} - x_k)$$

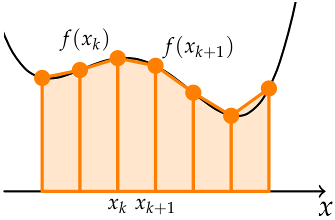


コード: midpoint.py

```
for k in range(divnum):
  integral_left += \
  function(bottom + interval * (k + .5) / divnum) \
  * (interval / divnum)
```

台形則

$$\sum_{k=0}^{n-1} \frac{f(x_k) + f(x_{k+1})}{2} (x_{k+1} - x_k)$$

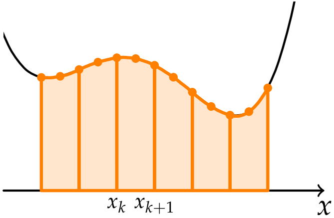


コード: trapezoid.py

```
for k in range(divnum):
  integral_left += \
  .5 * (function(bottom + interval * k / divnum) + function(bottom +
     interval * (k+1) / divnum)) \
  * (interval / divnum)
```

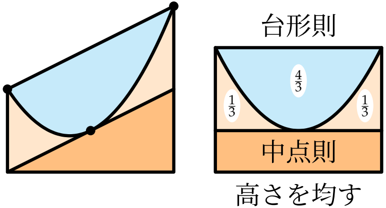
シンプソン則

区間 $[x_k, x_{k+1}]$ 上の積分値を，3 点 $(x_k, f(x_k))$, $\left(\frac{x_k+x_{k+1}}{2}, f\left(\frac{x_k+x_{k+1}}{2}\right)\right)$, $(x_{k+1}, f(x_{k+1}))$ を通る放物線の積分値で近似する．



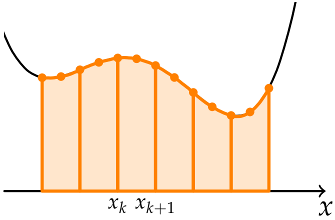
$$\sum_{k=0}^{n-1} \frac{f(x_k) + 4f\left(\frac{x_k+x_{k+1}}{2}\right) + f(x_{k+1})}{6}(x_{k+1} - x_k)$$

$$= \frac{2}{3}(\text{中点則}) + \frac{1}{3}(\text{台形則})$$



シンプソン則

$$\sum_{k=0}^{n-1} \frac{f(x_k) + 4f\left(\frac{x_k + x_{k+1}}{2}\right) + f(x_{k+1})}{6} (x_{k+1} - x_k)$$



コード: simpson.py

```
for k in range(divnum):
  integral_left += \
  (function(bottom + interval * k / divnum) + \
  4 * function(bottom + interval * (k+.5) / divnum) + \
  function(bottom + interval * (k+1) / divnum)) / 6 \
  * (interval / divnum)
```