数値微分と数値積分

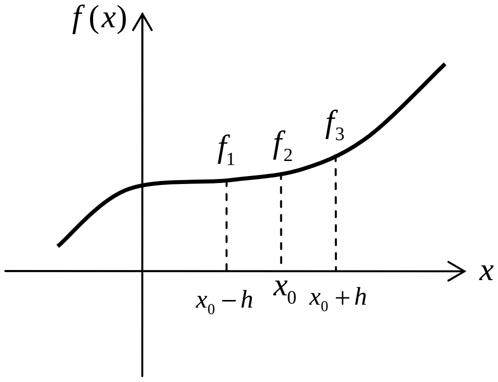
2007/5 平野拓一(東京工業大学)

1. はじめに

数値微分と数値積分について説明する。

2. 数値微分

数値計算では無限小を扱うことができないので、小さいが有限な値を用いた微分演算の近似である差分を用いる。ここで、微分はその点での傾き $f'(x)=\lim_{\Delta x\to 0}\dfrac{f(x+\Delta x)-f(x)}{\Delta x}$ だが、差分は有限距離離れた２点での傾き $\dfrac{f(x+\Delta x)-f(x)}{\Delta x}$ である。差分は下図に示すように前進差分、後退差分、中央差分の３つの方法がある。



$$
\begin{cases}
\left.\dfrac{df}{dx}\right|_{x=x_0} \approx \dfrac{f_3-f_2}{h} & \text{(前進差分)} \\
\left.\dfrac{df}{dx}\right|_{x=x_0} \approx \dfrac{f_2-f_1}{h} & \text{(後退差分)} \\
\left.\dfrac{df}{dx}\right|_{x=x_0} \approx \dfrac{f_3-f_1}{2h} & \text{(中央差分)}
\end{cases}
$$

[テイラー展開による差分の精度の考察]

$$f(x_0+h)=f(x_0)+hf'(x_0)+O(h^2) \;\to\; f'(x_0)=\frac{f(x_0+h)-f(x_0)}{h}+O(h)$$

上式は前進差分そのものであり、$O(h)$ の誤差を含むことがわかる。

$h\to -h$ と書き換えると、

$$f(x_0-h)=f(x_0)-hf'(x_0)+O(h^2) \;\to\; f'(x_0)=\frac{f(x_0)-f(x_0-h)}{h}+O(h)$$

となり、後退差分そのものである。これも $O(h)$ の誤差を含むことがわかる。つぎに、$O(h^2)$ までテイラー展開した次の2式の差を取ると、

$$f(x_0+h)=f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+O(h^3)$$

$$-)\; f(x_0-h)=f(x_0)-hf'(x_0)+\frac{h^2}{2}f''(x_0)+O(h^3)$$

$$f(x_0+h)-f(x_0-h)=2hf'(x_0)+O(h^3) \;\to\; f'(x_0)=\frac{f(x_0+h)-f(x_0-h)}{2h}+O(h^2)$$

となり、中央差分そのものである。これは $O(h^2)$ の誤差しかなく、前進差分、後退差分

と比較して精度が良いことがわかる。

差分による定式化は、微分方程式の初期値問題（ある時間における関数の値を与え、時間発展を追跡する問題）においてよく用いられる。例えば電磁界問題では FDTD 法で用いられる。**ルンゲ・クッタ法**は得に精度が良く、よく用いられる。電気力線の方程式を数値的に解くのにも用いられる。Mathematica では NDSolve コマンドは内部でこれらのいろいろなアルゴリズムを選択できるようになっている。

3. 数値積分

3.1 数値積分の必要性

数値積分は多くの科学技術計算などで必要となるツールである。例えば、定積分

$$A = \int_a^b f(x)dx$$

を計算するとしよう（このような問題は本当に頻繁に必要になる）。$f(x)$ の不定積分 $F(x)$ が簡単に求まるときは全く問題なく、

$$A = [F(x)]_a^b = F(b) - F(a)$$

と計算すればよい。

（例） $f(x) = x$ のとき、

$F(x) = \dfrac{x^2}{2}$ だから、

$A = (b^2 - a^2)/2$

しかし、実際には $f(x)$ の不定積分を求めるのがものすごく大変なことがある。不定積分をしようとしただけで気が遠くなり、研究が何も進まないのでは世の中発展しない。また、積分が複雑でやる気が起きないということだけが問題なのではなく、もっと本質的な問題がある。実は、微分は必ず求まるが、不定積分は必ずしも求まるとは限らないのである。より正確に言うと、初等関数(冪乗, 三角関数, 指数関数, 対数関数)の四則演算および合成関数で表された表現は必ず微分して初等関数の四則演算および合成関数で表された表現になるが、逆、つまり初等関数の四則演算および合成関数で表された表現の不定積分は必ずしも求まるとは限らないのである。もうこの段階でコンピュータを利用した近似的な数値積分の必要性がわかったと思う。

3.2 不定積分可能性と特殊関数について

理論に興味無い人は 3.2 節を読み飛ばしてもよいが、気になる人のために必ずしも不定積分が求まるとは限らないことを簡単に説明しよう。

<u>微分について</u>

微分演算子 D に対して次の公式が成り立つ。

(D1) 線形性: $D[f+g]=D[f]+D[g]$

(D2) 積の微分: $D[fg]=D[f]g+fD[g]$

(D3) 除算の微分: $D[f/g]=\dfrac{D[f]g-fD[g]}{g^2}$ (D2 and D4 からも導ける)

(D4) 合成関数の微分: $D[f(g)]=(df/dg)D[g]$

今、初等関数の四則演算および合成関数で表された表現の微分を考えると、(D1)〜(D4)を再帰的に適用して初等関数の微分に辿り着くことができる(数学的帰納法、プログラミング言語の再帰呼出、リカーシブ・コールを思い出すとわかりやすい)。なぜならば、(D1), (D2), (D3)は四則演算をばらして微分することを可能にし、(D4)は合成関数をばらして微分することを可能にするからである。そして、最後には初等関数の表現となる（なぜならば元の関数は初等関数の四則演算および合成関数で表された表現だったからである）。初等関数は必ず微分でき、その表現はまた初等関数で表されるので、初等関数の四則演算および合成関数で表された表現の微分表現は必ず初等関数の四則演算および合成関数で表現できるのである。

積分について

さて、微分はこのようにして絶対に行うことができるのであるが、何回も述べているように積分は必ずしもできるとは限らない。例えば、次の関数を見てみよう。

$\dfrac{\sin x}{x}$ (信号処理でよく登場する sinc 関数)

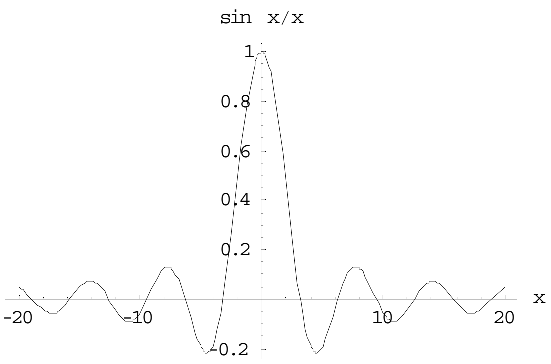


図 1 sinc 関数(x=0 は除去可能な特異点)

一見簡単そうに見えるが、実は不定積分$\int \dfrac{\sin x}{x}dx$が求まらない。「不定積分が求まらない」と言う表現は実は曖昧で、正確に言うならば「不定積分は初等関数で表現することができない」というのが正確である。だから、初等関数で表現することができないのだが、よく使う積分ならば新たにこの不定積分に名前を付けて定義してしまえば良いのである。実際に$\int \dfrac{\sin x}{x}dx$の不定積分は

初等関数とは別の関数として定義されて正弦積分関数 $S_i(x)=\int_{t=0}^{x}\frac{\sin t}{t}dt$ と定義される関数を用いて $\int\frac{\sin x}{x}dx=S_i(x)$ と表すことができる（$t=0$ による定数項は不定積分だからあってもなくても同じと考える)。このように、科学技術計算をしていていろいろなところで良く出現する方程式の解などが初等関数の四則演算および合成関数で簡潔に表せないとき、$S_i(x)$ のようにその解に名前を付けて定義することがよくある。このように特別に名前を与えられた関数を特殊関数(Special Function)と呼ぶ。他にも余弦積分関数、指数積分関数、ガンマ関数、ツェータ関数、ベッセル関数、ルジャンドル関数、マシュー関数、楕円関数などがある。

電磁界解析で用いる特殊関数はほとんどの場合、ヘルムホルツの波動方程式を解いたときに出てくる特殊関数である。ヘルムホルツの波動方程式をカルテシアン座標で解くと sin, cos (, exp) となり、初等関数で表される。しかし、円筒座標や球座標で解くと解は初等関数で表現できない。ヘルムホルツの波動方程式を円筒座標で解くとベッセル関数で表現される。導出過程からベッセル関数は別名円筒関数とも呼ばれる。ベッセル関数の線形結合であるハンケル関数もよく用いられる。ベッセル関数は円筒波の定在波を表し、ハンケル関数は円筒波の進行波を表現するのに用いられる。ヘルムホルツの波動方程式を球座標で解くと球ベッセル関数およびルジャンドル関数で表現できる。ヘルムホルツの波動方程式は電磁波に限らず、音波などいろいろな波動現象を数式で表現したときに、同じ数式表現として姿を表すことは想像できるだろう。だから、このようによく登場する微分方程式の解は特別に数学者が注目してその解を級数で表現したりして解いて名前を付けて特殊関数とするのである。慣例的によく研究した人の名前を関数名にすることがよく行われる。ベッセルやルジャンドルは人の名前である。工学ではそれらの特殊関数をツールとして用い、応用している。特殊関数が出現する微分方程式や方程式の解を知りたい場合には公式集[1]を調べると良い。また、最近は記号数式処理ソフトも充実しており、Mathematica などを用いても特殊関数を含む複雑な不定積分を行ってくれる。鶏と卵のような話になるが、特殊関数の値を数値計算するのにもいくつか方法があり、級数表現で計算する方法と積分表現のまま数値積分を行う方法などがある。後者を用いる場合、特殊関数の値を求めるのにも数値積分が必要になることがわかるだろう。

さて、話が逸れたが、今度はなぜ初等関数の四則演算および合成関数で表された表現の不定積分は必ずしも初等関数の四則演算および合成関数で表現されるとは限らないのかを説明しよう。不定積分演算子 I に対して次の公式が成り立つ。

(I1) 線形性: $I[f+g]=I[f]+I[g]$

(I2) 積の積分: $I[fg]=\left[f\ I[g]\right]-I\left[D[f]\ I[g]\right]$ (部分積分)

(I3) 除算の積分: $I[f/g]=f\ I[1/g]-I\left[D[f]\ I[1/g]\right]$ (部分積分)

(I4) 合成関数の積分: $I[f(g)]=I\left[f/D[g]\right]$ (置換積分)

(I1)～(I4)はそれぞれ(D1)～(D4)に対応する積分の公式である。微分の公式と違って、これは再帰

的に被積分関数に適用しても必ずしも積分できるとは限らない。なぜならば、(I1)は完全に分解してくれるが、(I2)～(I4)の公式は微分公式のときとは違って積分演算子 I や微分演算子 D の積の形が存在しているからである。幸運ならば積分できることもある。

ガロア理論について

さて、初等関数の四則演算および合成関数で表された表現の不定積分は必ずしも初等関数の四則演算および合成関数で表現されるとは限らないということを説明した。ここまでは理解できたと思うが、もう少し深く考察してみよう。どのようなときに被積分関数の不定積分ができて、どのようなときにできないのだろうか？この問題に対する明快な（しかし、難解である）解を与えたのはフランスの数学者ガロアである。この理論は難しく、原論文は非常に難解だったため、現代では数学者によって群(Group)・環(Ring)・体(Korper, Field)などのガロア理論に至るまでの基礎理論が整備されている。ガロア理論はある関数の不定積分は可能かどうかという問題としてよりは、方程式の解の公式が存在するかしないかを判定する理論として有名である。例えば、中学の数学でも習うように 2 次方程式 $ax^2+bx+c=0$ の解の公式 $x=\dfrac{-b\pm\sqrt{a^2-4ac}}{2a}$ は存在する。3 次方程式（カルダーノの公式など)、4 次方程式の解の公式も存在する。さて、ではそれ以上の高次の方程式につても解の公式は存在するのだろうか？実はガロア理論によると 5 次以上の方程式に対する一般の解の公式は存在しない。しかし、「代数学の基本定理」によれば確かに複素数の範囲で解は存在するのである。でも解の公式がないということは、すなわち「初等関数の四則演算および合成関数で表現できない」ということである。もし、また新たに特殊関数を定義すれば、もちろん解の公式は作れるということになる。このように、「解の公式が存在するか？」とか「不定積分可能か？」というようなことは実は曖昧な表現であり、正確に言うならば「初等関数の四則演算および合成関数で表現できるかどうか？」と言わなければならない。新たに特殊関数を定義して、その関数を以後用いてもよいとするならば表現できる世界は広がるのである。このように、ある集合を定義し、その集合に対する 2 項演算を定義してその演算を集合に施して得られる集合、またその集合に新たな演算を定義して・・・というように集合を拡張していく考え方が、群論、あるいは群・環・体の考え方である。5 次以上の方程式の解は一般にその集合に収まることができないのである。ガロア理論を用いると、ある方程式が与えられたときに解の公式が存在するかどうかを判別することができる。また応用として、ある関数が与えられたときに積分できるかどうか判別することができるのである。そのことを聞くだけで非常に興味深く、美しそうな理論だと思わないだろうか。

3.3 定積分とリーマン積分

さて、以上のように初等関数の四則演算および合成関数で表された表現の不定積分は必ずしも初等関数の四則演算および合成関数で表現されるとは限らない。しかし、定積分は不定積分の表現が得られなくてもある値を持つのである。すなわち、高校までは積分とは微分の逆演算であり、不定積分とは定積分に上端を代入した値から下端を代入した値を引いたものであると定義されて

いた。しかし、これでは実用上問題がある。なぜならば、不定積分の表現が得られない場合でも定積分は得られることもあるからである。この問題に対処するためにリーマンは不定積分の新たな定義を与えた。その概念は高校で習う**区分求積**の概念と同じと言ってよい。

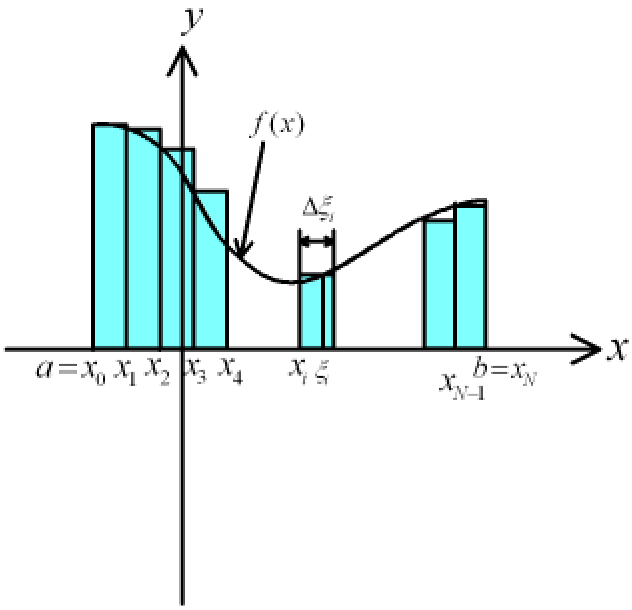


図 2 リーマン積分

ある関数 $f(x)$ の区間 $[a:b]$ でのリーマン積分は図 2 のように短冊和として面積を求めるのが定義である。各短冊の高さはその区間内の f の値だったらどの値を使ってもよい。そして短冊の幅を無限に小さくしたときの値はある一定の値に収束し、その値がリーマン積分の値とするのである（収束しない場合や、この定義でも問題ある場合があるかもしれない。他の積分の定義の例としては広義積分、ルベーグ積分、スチルチェス積分などいくつかある。必要ならば新しく定義すればよいのである）。

$$\int_a^b f(x)dx = \lim_{\substack{\Delta\xi_i \to 0 \\ (N\to\infty)}} \sum_{i=0}^{N-1} f(\xi_i)\Delta\xi_i$$

普通、簡単のために $\xi_i = x_i$, $\Delta\xi_i = \Delta x = (b-a)/N$ とする。

この面積を定積分、いや、もうこの言葉を使わずにリーマン積分と呼ぶのである。このようにリーマン積分を定義したら、不定積分の表現を基づく定義になっていないから不定積分の表現が得られない場合でも面積を計算するのに矛盾が生じない。リーマン積分は図 2 のようなイメージから短冊和と呼ばれることもある。

3.4 数値積分アルゴリズム

3.4.1 様々な数値積分法

1 節の話から、数値積分の重要性がわかったと思う。数値積分は求積法(quadrature)とも呼ばれる。本節では、ディジタルコンピュータを使ってどのように数値積分を行うか説明する。最も

簡単な方法は 3.3 節のリーマン積分の定義をそのまま実行することである。
しかし、図 2 を見てもわかるように、短冊（長方形）の面積を用いて近似すると相当分割数を上げないと精度が得られないことがわかると思う。

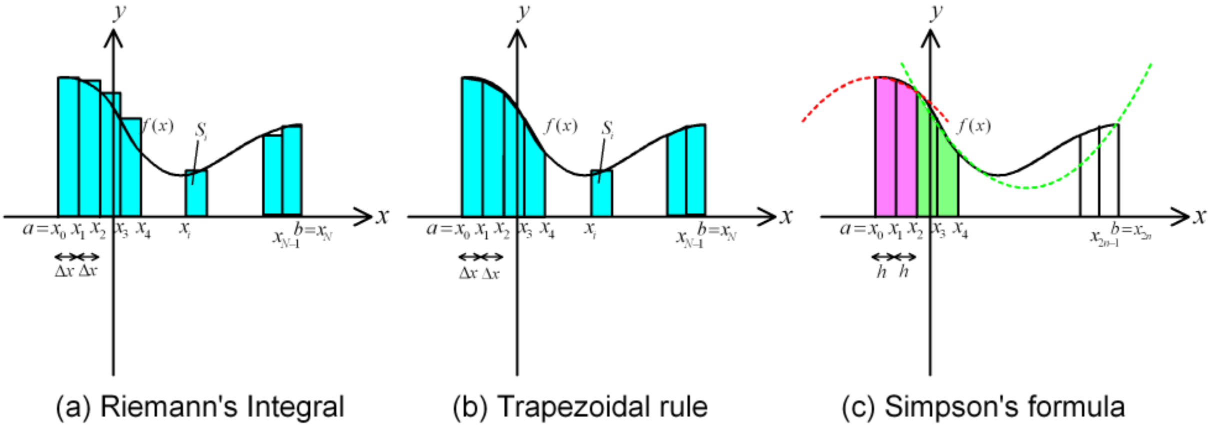


図 3 いろいろな数値積分法

そこで、図 3(b)に示すように、各区間を台形で近似すればより精度の高い数値計算が行えることがわかると思う。これを台形公式、あるいは台形則(Trapezoidal Rule)と言う。しかし、被積分関数 f が非常に滑らかな場合にはより精度の良い方法がある。それが図 3(c)に示すシンプソンの公式、あるいはシンプソン則である。ここまで説明すればもうわかると思うが、より高次の多項式で展開する方法がある。このように多項式補間を用いて数値積分する方法を Newton-Coes 公式と言う（台形則やシンプソン則はその特殊な場合である）。

シンプソンの公式は被積分関数を 2 次関数で補間するのでリーマン積分（短冊和）や台形公式を使うよりも滑らかな被積分関数に対しては一般に精度がいい。ただ、被積分関数が滑らかだという前提があることに注意しておく必要がある[2]。

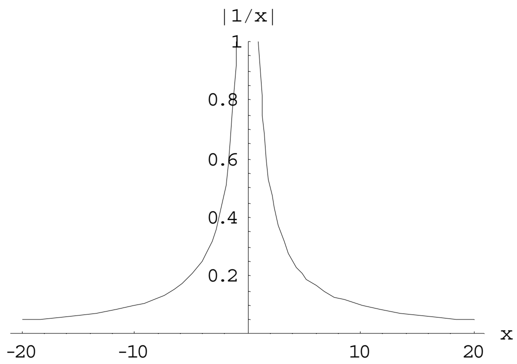


図 4 |1/x|のグラフ

例えば、図 4 のグラフのように被積分関数が急激に変化する場合、積分区間が急激な変化点を含む場合にはシンプソンの公式を使うと誤差が大きくなってしまう。この場合にはまだ工夫の無い台形則の方が精度が良くなったりするので注意が必要である。このような問題は電磁界解析のモーメント法の自己リアクションの計算などでもよく起きる。

また、今説明した数値積分法は非常に古典的なものであり、分割が不等間隔のものや自動的に分割の仕方を判断するものもある。しかし、数値計算の問題ではどこでミスしているかわからないため、わからなかったらとりあえず手の込んだルーチンを使うよりもアルゴリズムと動作が良くわかっているシンプルな台形則などを使うことが有用な場合が多い。電磁界のモーメント法解析を行うならばせめてシンプソンの公式ぐらいまでは知っていて欲しい。他のより高度な数値積分法に関してはサブルーチン集を利用してプログラム中で呼び出すことができればよい。

3.4.2 リーマン積分（短冊和）

$$S_i = f(x_i)\Delta x$$

$$\int_a^b f(x)dx \cong \sum_{i=0}^{N-1} S_i = \sum_{i=0}^{N-1} f(x_i)\Delta x$$

ただし、$\Delta x = (b-a)/N$, $x_i = i\,\Delta x$

3.4.3 台形則

i 番目の台形の面積は

$$S_i = \frac{\{f(x_i) + f(x_{i+1})\}\Delta x}{2}$$

$$\int_a^b f(x)dx \cong \sum_{i=0}^{N-1} S_i$$

$$= \frac{\{f(x_0) + f(x_1)\}\Delta x}{2} + \frac{\{f(x_1) + f(x_2)\}\Delta x}{2} + \cdots + \frac{\{f(x_{N-1}) + f(x_N)\}\Delta x}{2}$$

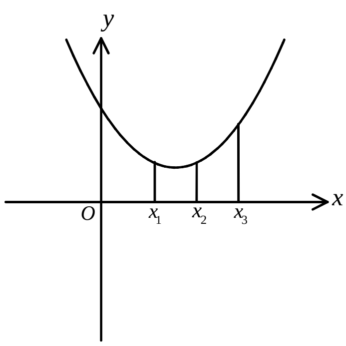
$$= \frac{\{f(x_0) + f(x_N)\}\Delta x}{2} + \sum_{i=1}^{N-1} f(x_i)\Delta x$$

ただし、$\Delta x = (b-a)/N$, $x_i = i\,\Delta x$

3.4.4 シンプソンの公式

補助定理１ 3 点 $(x_1, y_1), (x_2 = \frac{x_1 + x_3}{2}, y_2), (x_3, y_3)$ を通る 2 次関数（放物線）の方程式

$$f(x) = y = ax^2 + bx + c$$



証明

$$\begin{cases} a{x_1}^2 + bx_1 + c = y_1 \\ a{x_2}^2 + bx_2 + c = y_2 \\ a{x_3}^2 + bx_3 + c = y_3 \end{cases}$$

$$\begin{bmatrix} {x_1}^2 & x_1 & 1 \\ {x_2}^2 & x_2 & 1 \\ {x_3}^2 & x_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \qquad \text{(Vandermonde 行列)}$$

Mathematica

```
In[1]:= Solve[{{x1^2, x1, 1}, {x2^2, x2, 1}, {x3^2, x3, 1}}.{a, b, c} == {y1, y2, y3} /. {x2 → (x1 + x3) / 2}, {a, b, c}] //
        FullSimplify
```

$$\text{Out[1]= } \left\{\left\{a \to \frac{2\,(y1 - 2\,y2 + y3)}{(x1 - x3)^2},\ b \to -\frac{x3\,(3\,y1 - 4\,y2 + y3) + x1\,(y1 - 4\,y2 + 3\,y3)}{(x1 - x3)^2},\ c \to \frac{x3\,(x1 + x3)\,y1 - 4\,x1\,x3\,y2 + x1\,(x1 + x3)\,y3}{(x1 - x3)^2}\right\}\right\}$$

補助定理２

補助定理１の放物線を区間$[x_1, x_3]$で積分すると

$$\int_{x_1}^{x_3} f(x)dx = \frac{1}{6}(x_3 - x_1)(y_1 + 4y_2 + y_3)$$

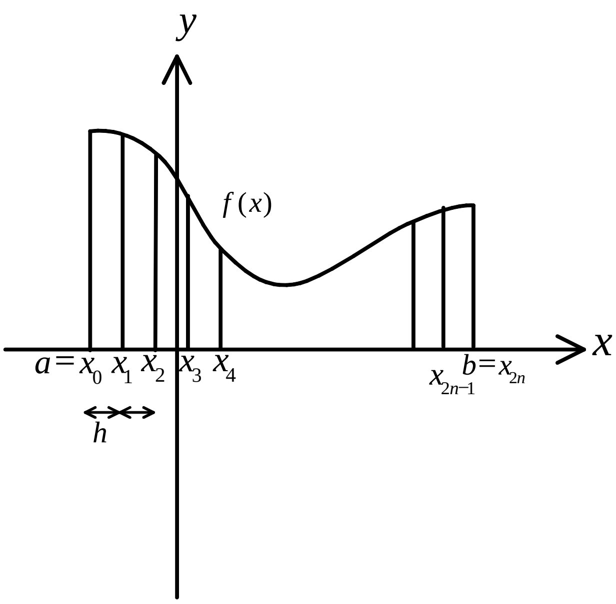
証明

Mathematica

```
In[2]:= f[x_] := a*x^2 + b*x + c;
        Integrate[f[x], {x, x1, x3}] /.
          {a → 2 (y1 - 2 y2 + y3)/(x1 - x3)^2, b → -(x3 (3 y1 - 4 y2 + y3) + x1 (y1 - 4 y2 + 3 y3))/(x1 - x3)^2,
           c → (x3 (x1 + x3) y1 - 4 x1 x3 y2 + x1 (x1 + x3) y3)/(x1 - x3)^2} // Simplify
```

$$\text{Out[3]= } -\frac{1}{6}\,(x1 - x3)\,(y1 + 4\,y2 + y3)$$

シンプソンの公式



図のように区間$[a,b]$を$2n$等分し、各区間の長さをhとする。そして補助定理２の公式を点$(x_0,x_1,x_2),(x_2,x_3,x_4),\cdots,(x_{2n-2},x_{2n-1},x_{2n})$の$n$組に対して適用すると$\int_a^b f(x)dx$の近似値$S_{2n}^{\mathrm{simp}}$が求まる。

$$
\begin{aligned}
S_{2n}^{\mathrm{simp}} &= \frac{h}{3}\sum_{i=0}^{n-1}(y_{2i}+4y_{2i+1}+y_{2i+2}) \\
&= \frac{2h}{6}\{(y_0+4y_1+y_2)+(y_2+4y_3+y_4)+\cdots+(y_{2n-2}+4y_{2n-1}+y_{2n})\} \\
&= \frac{h}{3}\{y_0+4(y_1+y_3+\cdots+y_{2n-1})+2(y_2+y_4+\cdots+y_{2n-2})+y_{2n}\}
\end{aligned}
$$

問題

短冊和、台形公式、シンプソンの公式を使った数値積分プログラムを作成せよ。
自分で適当な関数を選び、作成したプログラムを使ってその関数を数値積分した結果が解析的に定積分した結果とよく一致することを確かめよ。また、分割数による収束の傾向をグラフにして示せ。

References

[1] 森口繁一・宇田川銈久・一松信, 数学公式, 岩波書店, 東京, 1997.
[2] 中井三留, 微分法と積分法, 学術図書出版社, pp.120-121, 1994.
[3] W.H. Press et al., Numerical Recipes in C [日本語版], 技術評論社, 東京, 2003.