領域の三角分割による２次元適応型積分則

2003MM055 牧哲弘
指導教員: 杉浦洋

1　はじめに

本研究では二次元三角領域上の定積分を近似する適応型数値積分則の構成を行う．

二次元の一般的な領域は，適当な領域分割と変数変換により三角領域上の積分に帰着できる．この意味で三角領域上の数値積分法は基本的で重要である．

点(0,0),(1,0),(0,1)を頂点とする三角形を基本三角形と呼ぶ．基本三角形上の近似積分公式を基本公式と呼ぶ．任意の三角領域上の積分は，アフィン変換により基本三角形上の積分に変換できるので,基本公式によりすべての三角領域における積分の近似値が計算できる．

適応型積分則とは，要求精度にしたがって三角領域を小三角領域に分割し，各小三角領域に同じ基本公式を用いる計算法である．分割が細かくなればなるほど精度がよくなる．その際，積分関数の変化の緩やかな部分は粗く分割し，変化の激しい部分は細かく分割することによって，一様均等な分割法と比べて，より少ない分割数で同じ精度が達成できると思われる．そのためには関数の局所的な変化の激しさを評価する数値的な指標が必要である．

2　基本公式について

xy-平面上の3点$\boldsymbol{\alpha}=(0,0)$,$\boldsymbol{\beta}=(1,0)$,$\boldsymbol{\gamma}=(0,1)$を頂点とする三角形領域$\Delta$を基本三角領域と呼ぶ．$n$個の標本点$\boldsymbol{\pi}_1=(\xi_1,\eta_1)$,$\boldsymbol{\pi}_2=(\xi_2,\eta_2)$,$\cdots$, $\boldsymbol{\pi}_n=(\xi_n,\eta_n)\in\Delta$と重み$\rho_1$,$\rho_2$, $\cdots$, ρ_nによる積分公式を特別に

$$I_n f=\sum_{l=1}^{n}\rho_l f(\boldsymbol{\pi}_l)\cong\iint_{\Delta} f(\boldsymbol{p})dxdy$$

と書き基本公式と呼ぶ.

基本公式は，変数変換により任意の三角領域$D=D(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})$の近似積分に用いることができる．$D$の面積を$S$とする，基本領域$\Delta$から$D$へのアフィン変換

$$\varphi: q=s\boldsymbol{\alpha}+t\boldsymbol{\beta}+u\boldsymbol{\gamma}\mapsto \boldsymbol{p}=s\boldsymbol{a}+t\boldsymbol{b}+u\boldsymbol{c}$$

により

$$\iint_D f(p)dxdy=2S\iint_{\Delta} f(\varphi(q))dtdu$$

この右辺に積分則I_nを用いて積分公式

$$\begin{aligned}Q_n(D)f&=2S\sum_{i=1}^{n}\rho_i f(\varphi(\xi_i,\eta_i))\\&\cong Q(D)f=\iint_D f(x,y)dxdy\end{aligned}\qquad(1)$$

を得る．

$s(s\geq 0)$次以下の2変数多項式全体の集合を$\prod_s=\{p(x,y):\deg p\leq s\}$と書く．次数は全次数である．任意の$f\in\Pi_s$で$I_n(D)f=I(D)f$，かつ$I_n(D)f\neq I(D)f$となる$f\in\Pi_{s+1}$が存在するとき，積分公式$I_n(D)$は次数$s$であると言う．アフィン変換で，$x,y$の$k$次以下の多項式は$t,u$の$k$次多項式に変換されるから，$I_n$が$s$次公式なら$I_n(D)$も$s$次公式である．三角形を含む最小円板の半径を三角形の半径とよぶ.

3　適応型積分則の領域分割法と分割の判断指標について

与えられた許容誤差$\varepsilon>0$に対し真の積分値$Q(D)f$の近似積分$\tilde{Q}(D)f$を

$$|\tilde{Q}(D)f-Q(D)f|\leq\varepsilon$$

を満たすように計算することを目標とする．
まず，D全体にs次積分則$Q_n(D)$を用い$\tilde{Q}(D)f=Q_n(D)f$とする．もし，

$$|\tilde{Q}(D)f-Q(D)f|>\varepsilon$$

なら，領域Dをその頂点と重心$\boldsymbol{g}$を通る直線で小三角領域D_0,D_1に2分し(**図1**)，それぞれに同じ積分則(1)を用いる．各小領域には許容誤差$\varepsilon/2$ を割り当て，

$$|\tilde{Q}(D_i)f-Q(D_i)f|\leq\varepsilon/2(i=0,1)\qquad(2)$$

なら

$$\tilde{Q}(D)f=\tilde{Q}(D_0)+\tilde{Q}(D_1)f$$

として終了する．もし,いずれかのiで(2)が成立しなければその領域をさらに2分する．この再帰的な操作をくりかえし，すべての小領域で割り当てられた許容誤差を満たした時点で近似積分が完了する．

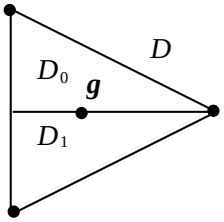

図 1: 三角形の分割

ここで問題となるのは誤差評価である．用いるn点基本公式をI_n，標本点の集合をX_nとする．X_nの部分集合を$X_m(m<n)$としそれを標本点とする低次基本公式をI_mとする．

この2つの公式の差でI_nの誤差を推定する．すなわち

$$|I_n(D)f-Q(D)f|\cong E_n(D)f=|I_n(D)f-I_m(D)f|$$

である．これにより，εを許容誤差とする適応型積分$\tilde{Q}(D,\varepsilon)$は次のような再帰関数で表現できる．

$$\tilde{Q}(D,\varepsilon)f = \begin{cases} I_n(D)f \quad (if \quad E_n(D)f \le \varepsilon) \\ \tilde{Q}(D_1,\frac{\varepsilon}{2})f + \tilde{Q}(D_2,\frac{\varepsilon}{2})f \quad (if \quad E_n(D)f > \varepsilon) \end{cases} \tag{3}$$

I_mをI_nの埋め込み公式と言う．

4　標本点の再利用

元の領域で使った標本点を使い捨て，分割後にまた計算することはロスになるので，共通の標本点の再利用を考える．例として三角形の頂点を標本点とする積分則を考える．

元の領域で標本値d[0],d[1],d[2]を標本点の上に書いた図と，分割後の2つの三角形での標本値d0[0],d0[1],d0[2]とd1[0],d1[1],d1[2]を標本上に書いた図を並べて示す．(図2)

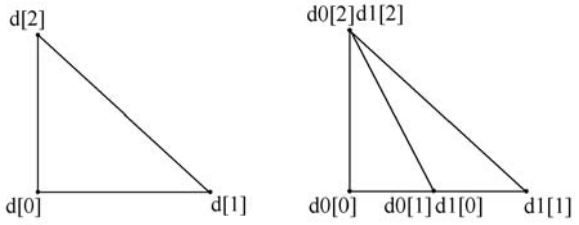

図 2: 分割前，分割後

標本値は6つに増えるが次のように再利用できる．
d0[0]=d[0]
d0[1]=新たに計算
d0[2]=d[2]
d1[0]=d0[1]
d1[1]=d[1]
d1[2]=d[2]
ゆえに，新たに計算しなければならない標本値はd0[1]の1個に抑えることができる．

5　MathematicaによるCプログラム自動生成

n点積分則I_nは標本点のx,y座標と重みを合わせると$3n$個のデータが必要である．また誤差評価のために埋め込む低次公式の標本点番号と重みが必要である．

高次公式ではnが大きくなるので，これらを間違いなくプログラムすることが困難である．

標本値の再利用をプログラム化することはさらに複雑で困難を極める．

そこでMathematicaによりCのプログラムを部分的に自動生成することにした．

6　数値実験結果

基本三角領域Δ上で関数$f(x,y)=e^{x+y}$を積分する．

$$Q(\Delta)f = \iint_{\Delta} f(x,y)dxdy = 1$$

を許容誤差$\varepsilon = 10^{-6}$で近似する．

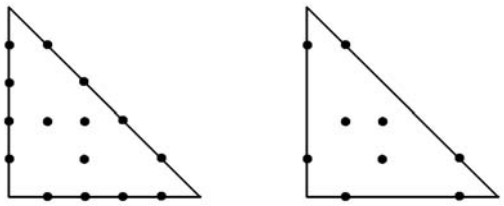

図 3: 標本点　　図 4: 誤差評価用標本点

下の図のような標本点を使いプログラムを実行した．

プログラムの実行結果は1.0000007743376699となった．絶対誤差$|\tilde{Q}(D)f - Q(D)f| = 0.77 \times 10^{-6} < \varepsilon$許容誤差を満たしている．

図5は本研究の方法を適用したときの三角形分割を示す．被積分関数の変化の激しい領域を密に分割していることがわかる．

この場合分割回数は40回なので，標本点を再利用しない場合は600個の標本点が必要となるが，再利用した場合は標本点が増える割合は一回の分割で11個なので，標本点数は444個になる．

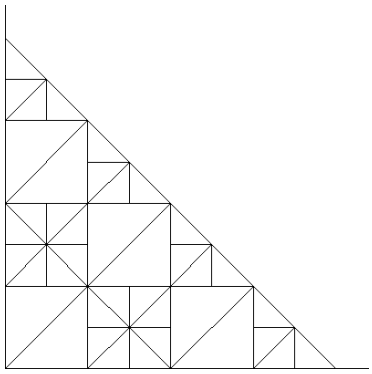

図 5:

7　おわりに

三角領域における数値積分において，求める精度の近似積分を行う適応型積分則をC言語で作成した．標本値の計算回数を減らすため，分割前の親領域の標本値を分割後の子領域で再利用できるものは再利用した．子領域同士で共通な標本値は1度計算して共通に使うようにした．そのためのプログラムは非常に複雑になるためMathematicaで自動生成するプログラムを作成した．

今回は，三角形の最長辺を分割しているが，関数の変化が激しい方向にそった辺を分割する方法[1]も有力な方法と考えられるが，それに関したては今後の課題とする．

参考文献

[1] A.Gentz, R.Cools: An Adaptive Numerical Cubature Algorithm for Simplices, ACM Transactions on Mathematical Software, vol.29, no.3, P.297 - 308(2003).