数理解析研究所講究録 1335 巻 2003 年 196-203

数学公式データベースとG関数

森永 昌義
MASAYOSHI MORINAGA
愛媛大学大学院理工学研究科
GRADUATE SCHOOL OF SCIENCE & ENGINEERING EHIME UNIVERSITY[*]

村上 裕美
YUMI MURAKAMI
愛媛大学大学院理工学研究科
GRADUATE SCHOOL OF SCIENCE & ENGINEERING EHIME UNIVERSITY[†]

野田 松太郎
MATU-TAROW NODA
愛媛大学工学部
SCHOOL OF ENGINEERING EHIME UNIVERSITY[‡]

1 はじめに

数式処理のアルゴリズムの中で、記号積分に関する研究は極めて重要である。被積分関数の型を有理関数のみとしていた初期のアルゴリズムの拡張がはかられている。しかし、より広範な問題に数式処理を活用するためには、数学公式データベースを作成し、パターンマッチングを行うことにより記号積分を行う方法に頼らざるを得ない。現在までの数式処理に関する研究では、必ずしもデータベースとの結合は十分には検討されてはいない。この種の研究として、数式処理システム GAL [9] を対象として、関係データベースの上で、数学公式の構造とインデキシング手法 [10] について述べたものが発表されているのみである。そこで、本研究では、数式処理システムの固有の数式表現や命令形態に依ることなく、数学公式データベースにより記号積分を行えるようなシステム作成を行うことを目標とする。このため、数式表現には、OpenMath 表現を採用した [1, 2, 3]。OpenMath による数式表現は XML 形式 [4] で表されており、XML の名前が、記号、演算子、関数名の数学オブジェクトである OpenMath オブジェクトに対応する。このような数学オブジェクトを木構造を用いて表現することによって、結果として公式の検索を容易にすることが出来る。記号積分を実行するために、超幾何関数を一般化した Maijer の G 関数 [6, 8] を実装し、公式の自動作成・検索を行う。G 関数固有な演算によって、G 関数相互間の変換が可能である。この技術を用い、任意の数式処理システムで計算することにより、G 関数を変形し、新しい数学公式を導出し、数学公式データベースに格納することが出来る。この操作によって大量の数学公式を生成することが可能になる。

*morinaga@hpc.cs.ehime-u.ac.jp
†cumi@hpc.cs.ehime-u.ac.jp
‡noda@cs.ehime-u.ac.jp

2 Meijer の G 関数

2.1 Meijer の G 関数と関数定理

Meijer の G 関数 [6, 8] は次式によって定義される。

$$G(\vec{a};\vec{b};\vec{c};\vec{d};z)=G_{pq}^{mn}\left(z\left|\begin{matrix}a_1,\ldots,a_p\\ b_q,\ldots,b_q\end{matrix}\right.\right)=\frac{1}{2\pi i}\oint_L\Gamma\left(\begin{matrix}1-\vec{a}+y,\vec{c}-y\\ \vec{b}-y,-\vec{d}+y\end{matrix}\right)e^{yz}dy \tag{1}$$

ここで、$\vec{a}=(a_1,\ldots,a_n),\vec{b}=(a_{n+1},\ldots,a_p),\vec{c}=(b_1,\ldots,b_m),\vec{d}=(b_{m+1},\ldots,b_q)$ である。また、

$$\Gamma\left(\begin{matrix}a_1,\ldots,a_m\\ b_1,\ldots,b_n\end{matrix}\right)=\frac{\prod_{i=1}^{m}\Gamma(a_i)}{\prod_{i=1}^{n}\Gamma(b_i)}$$

を意味する。

積分路 L は $L_{\gamma+i\infty}$,L_{∞},$L_{-\infty}$ の 3 つのいずれかになる。$n,p-n,m,q-m$ は各々$\vec{a},\vec{b},\vec{c},\vec{d}$ の要素数を示している。Meijer の G 関数は次のような性質を持つ。

$$G(\mu,\vec{a};\vec{b};\vec{c};\mu,\vec{d};z) = G(\vec{a};\vec{b};\vec{c};\vec{d};z) \tag{2}$$

$$G(\vec{a};\mu,\vec{b};\mu,\vec{c};\vec{d};z) = G(\vec{a};\vec{b};\vec{c};\vec{d};z) \tag{3}$$

$$G(\vec{a};\vec{b};\vec{c};\vec{d};-z) = G(1-\vec{a};1-\vec{b};1-\vec{c};1-\vec{d};z) \tag{4}$$

$$e^{tz}G(\vec{a};\vec{b};\vec{c};\vec{d};z) = G(\vec{a}+t;\vec{b}+t;\vec{c}+t;\vec{d}+t;z)\ . \tag{5}$$

以上の性質を用いることにより G 関数の積分定理を導くことができる [6, 8]。以下に、G 関数の積分定理を示す。

1. 不定積分

$$\int G(\vec{a};\vec{b};\vec{c};\vec{d};z)dz=G(1,\vec{a};\vec{b};\vec{c};0,\vec{d};z) \tag{6}$$

2. 定積分

$$\int_0^y x^{\alpha-1}G_{pq}^{mn}\left(\omega x\left|\begin{matrix}a_1,\ldots,a_p\\ b_1,\ldots,b_q\end{matrix}\right.\right)dx = y^{\alpha}G_{p+1,q+1}^{m,n+1}\left(\omega y\left|\begin{matrix}a_1,\ldots,a_n,1-\alpha,a_{n+1},\ldots,a_p\\ b_1,\ldots,b_m,-\alpha,b_{m+1},\ldots,b_q\end{matrix}\right.\right) \tag{7}$$

3. 無限積分

$$\int_0^{\infty} z^tG(\vec{a};\vec{b};\vec{c};\vec{d};\log(z)+v)dz = \frac{1}{u}e^{-\alpha v}\Gamma\left(\begin{matrix}-\vec{a}+1-\alpha,\vec{c}+\alpha\\ \vec{b}+\alpha,-\vec{d}+1-\alpha\end{matrix}\right) \tag{8}$$

ただし、$\alpha=\frac{t+1}{u}$ である。

また、G 関数の積を用いる場合の無限積分は以下のようになる。

$$\int_0^{\infty} z^tG_1G_2dz=\frac{1}{u}e^{-\alpha v_2}G_3 \tag{9}$$

ただし、$\alpha=\frac{t+1}{u}$ とし、

$$\begin{aligned}G_1 &= G(\vec{a_1};\vec{b_1};\vec{c_1};\vec{d_1};u\log(z)+v_1),\\ G_2 &= G(\vec{a_2};\vec{b_2};\vec{c_2};\vec{d_2};u\log(z)+v_2),\\ G_3 &= G(\vec{a_1},-\vec{c_2}-\alpha+1;\vec{b_1},-\vec{d_2}-\alpha+1;\vec{c_1},-\vec{a_2}-\alpha+1;\vec{d_1},-\vec{b_2}-\alpha+1;v_1-v_2)\end{aligned}$$

である。

4. Cauchy の主値積分

$$\int_0^\infty \frac{G(\vec{a};\vec{b};\vec{c};\vec{d};\log(z)+v)}{z-\mu}dz = -\pi G\left(0,\vec{a};-\frac{1}{2},\vec{b};0,\vec{c};-\frac{1}{2},\vec{d};v+\log(\mu)\right) \tag{10}$$

2.2 公式の生成

Meijer の G 関数は次のような関係式によって新たに公式が生成できる。

2.2.1 Shift Operators

$$\begin{aligned} A_i &= D+(-a_i+1) \\ B_i &= -D+(b_i-1) \\ C_i &= -D+c_i \\ D_i &= D-d_i \end{aligned}$$

$D=\frac{\partial}{\partial z}$ は微分法のための演算子である。A_i と B_i は減算インデックス、C_i と D_i は加算インデックスとみなすことができる。この関係式を用いると、

$$G(\vec{a}-\vec{e_i};\vec{b};\vec{c};\vec{d};z) = A_iG(\vec{a};\vec{b};\vec{c};\vec{d};z) \tag{11}$$
$$G(\vec{a};\vec{b}-\vec{e_i};\vec{c};\vec{d};z) = B_iG(\vec{a};\vec{b};\vec{c};\vec{d};z) \tag{12}$$
$$G(\vec{a};\vec{b};\vec{c}+\vec{e_i};\vec{d};z) = C_iG(\vec{a};\vec{b};\vec{c};\vec{d};z) \tag{13}$$
$$G(\vec{a};\vec{b};\vec{c};\vec{d}+\vec{e_i};z) = D_iG(\vec{a};\vec{b};\vec{c};\vec{d};z) \tag{14}$$

となることは明白である。ここで、$\vec{e_i}$ は単位ベクトルである。Shift Operator を用いた公式生成の例を以下に示す。

例:
$\sin(x)=G(\ ;\ ;\frac{1}{2};0;\frac{x^2}{4})$ の $\vec{c}$(第 3 引数である $\frac{1}{2}$) について Shift Operator を用いることを考える。今、式 (13) を用いると $G(;;\frac{3}{2};0;\frac{x^2}{4})$ となる。

$$\begin{aligned} G(\ ;\ ;\frac{3}{2};0;\frac{x^2}{4}) &= \left(\frac{x}{2}\frac{\partial}{\partial x}-\frac{1}{2}+1\right)G(\ ;\ ;\frac{1}{2};0;\frac{x^2}{4}) \\ &= \left(\frac{x}{2}\frac{\partial}{\partial x}+\frac{1}{2}\right)\frac{\sin(x)}{\sqrt{\pi}} \\ &= \frac{1}{2\sqrt{\pi}}(-x\cos(x)+\sin(x)) \end{aligned}$$

このように Shift Operator を用いることにより新たに公式を求めることが可能である。これらの関係式を用いると、ある G 関数 $G(\vec{a};\vec{b};\vec{c};\vec{d};z)$ の公式が与えられると新たに、$G(\vec{a}-\vec{e_i};\vec{b};\vec{c};\vec{d};z)$ など多くの公式を得ることができる。このような関係を利用して新たに公式を生成するアルゴリズムは、超幾何関数に対する公式生成アルゴリズム [7] を G 関数に拡張することで得られている [8]。

2.3 Marichevの積分アルゴリズム

これらの定理によって積分の求解を行う。G 関数を用いた積分の求解は、次の手順で行われる。

1. 被積分関数を G 関数表現に変換する
   例えば、$\int_0^y e^{-x}dx$ を求めたい場合, $\int_0^y G_{01}^{10}\left(x\left|\begin{matrix}\cdot\\0\end{matrix}\right.\right)dx$ に変形する。
2. G 関数の積分の定理から求めたい積分の解を G 関数表現で求める
   これは上述した G 関数の定理を示した式 (11)-(14) を用いて行う。
3. 得られた G 関数表現の式を通常の関数表現に戻す
   2で得られた式を式 (7)-(10) によって変形を行い、対応する通常の関数表現に変換する。

このアルゴリズムを数式処理システム REDUCE に実装する方法は議論されている [6]。1と3では、通常の関数表現と G 関数表現の関係を参照し、処理を行なっている。2については、積分公式の各引数に、実際の値を代入することで、解の G 関数表現を得ている。

3 システムの実際

3.1 数式の特性化

数学公式データベースを利用するには、数式処理システムが効率的に数学公式を検索できなくてはならない。そこで、検索する際にキーとして用いる公式の特徴を抽出する方法を決定するために、数学公式の特性化を行う。本研究での数学公式データベースはプラットフォーム独立であることに重点をおいているため、数学公式の表現が、数式処理システム固有の表現に依存しないこと、効率的なインデックスを構成すること、数式の意味を保ったまま数学公式の通信が行なえること、という条件を満たさなければならない。このような条件を満たすために、数学的オブジェクトを表現するための標準である OpenMath [1, 2, 3] に従う。格納する数学公式は

$$\textbf{(一般的な表現の式)} \;=\; \textbf{(}G\textbf{関数表現の式)}$$

$$\textbf{例:}\qquad \frac{\sin(x)}{\sqrt{\pi}} \;=\; G_{02}^{10}\left(\frac{x^2}{4}\left|\begin{matrix}\cdot\,\cdot\\ \frac{1}{2},0\end{matrix}\right.\right)$$

で表される公式であるが、この公式の左辺と右辺それぞれの特性化を行う。

3.1.1 一般的な表現の式に関する特性化

数学公式は OpenMath に従う XML 形式で表現されているので、XML の特性の中で、

- XML 文書が木構造で表現できること
- タグに囲まれたデータ情報がタグの名前として表現されていること

という特性を用いて、左辺の特性化を行う。ノード名から得られる特性はその数式に含まれる関数名 (sin, cos, log 等) と演算子名 (plus,milus,time 等) である。これらは OpenMath による数式表現では OMS オブジェクト

に分類される。また、木構造より得られる特性は、深さ、葉の数、節ノード数である。これらはOpenMathによってXML形式に変換された数式をDOM [5] を用いて木構造に展開し、DOMによる木構造表に対して、深さや葉の数をカウントすることでそれぞれの特性を抽出することができる。

3.1.2 G関数表現の式に関する特性化

G関数表現の式は上述したようなXMLによる木構造の特徴では識別することができない。しかし、G関数の要素が同じであれば公式が複数存在することはない。そこでG関数表現の式に関する特性は以下のようなものである。

- $\vec{a}, \vec{b}, \vec{c}, \vec{d}$の値そのもの
- $\vec{a}, \vec{b}, \vec{c}, \vec{d}$それぞれの要素数
- zに対応する変数

これらをキーとしてデータベースを設計する。

3.2 数学公式データベースの設計

- 数学公式リレーション (公式番号, 一般的な表現の式,G関数表現の式)
- インデックスリレーション (公式番号, 深さ, 葉の数、OMSの数、シンボル名)
- G関数リレーション (公式番号,Aベクトル,Bベクトル,Cベクトル,Dベクトル,z)

数学公式リレーションでは公式番号を主キーとして検索を行なうことができるため、設計にはリレーショナルデータベースモデルに従った。インデックスリレーションおよびG関数リレーションに関しては一つの公式に関して複数のキーワードが存在するため、オブジェクトリレーショナルモデルに従った。

3.3 公式運用部の流れ

図1は公式運用部の流れを表したものである。それぞれの動作について、以下で説明する。

(A) 入力式の解析
入力式はXML形式で表されているので、DOMパーサを用いて木構造に変換して以下の解析を行なう。

- 不定積分であるか定積分を判定し、積分処理するためにどの定理が使えるかの判定を行なう。
- OMSノードに対する変数名を抽出する。
- 木構造の特徴量を抽出する。

(B) インデックスリレーションによる検索
インデックスリレーションはXML形式で格納する。これを木構造にしたものを用いて、検索を行なう。

```
<formura id="5001">
      <deep>4</deep>
```

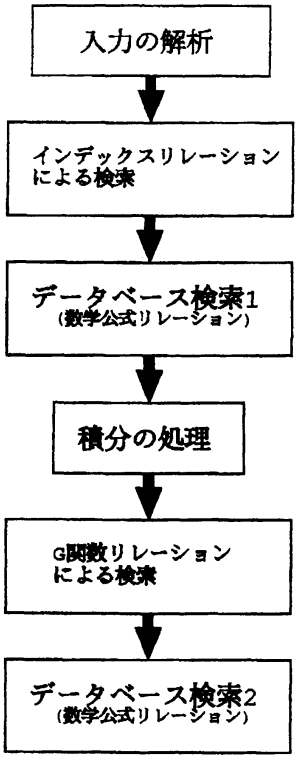

図 1: 公式運用部の流れ

```
        <leaf>7</leaf>
        <OMS>6</OMS>
        <Symbol>plus</Symbol>
        <Symbol>times</Symbol>
        <Symbol>sin</Symbol>
        <Symbol>cos</Symbol>
</formula>
```

これは、

$$-\frac{1}{2}xcos(x)+\frac{1}{2}sin(x)=\sqrt{\pi}G_{02}^{10}\left(\frac{x^2}{4}\middle|\begin{matrix}\cdot\cdot\\ \frac{3}{2},0\end{matrix}\right) \tag{15}$$

という公式の左辺に関する特徴量を示しているものである。このような公式に関するデータの集まりがインデックスリレーションには格納されている。このインデックスリレーションと入力式を解析することによって得られたキーをもとにパターンマッチングを行ない、それに対応する公式番号を抽出する。

(C) データベース検索 1

データベース検索は、Java 言語によるデータベース接続によって SQL を実行することで実現してい

る。データアクセスインターフェースとして JDBC を利用している。(B) で得られた公式番号をキーとしてデータベース検索を行ない XML 形式で書かれた G 関数表現の式を求める。

(D) 積分処理
(C) で得られた XML 形式で書かれた G 関数表現の式に積分の処理を行なうのだが、積分の処理は大別して不定積分と定積分がある。入力式を解析した際にどの積分であるか判定しているので、それによって、3 章で述べた対応する積分の処理を行なう。この処理によって、XML 形式で書かれた新たな G 関数表現の式を得る。

(E) G 関数リレーションによる検索
(B) と作業は同じであるが、G 関数表現の式から一般の表現の式に変換するために行なう。
G 関数リレーションは XML 形式で格納する。これを木構造にしたものを用いて、検索を行なう。

```
<formula id="5001>
      <A name="vector_a"></A>
      <B name="vector_b"></B>
      <C name="vector_c"><OMI>3/2</OMI><OMI>0</OMI></C>
      <D name="vector_d"><OMI>0</OMI></D>
      <Meijer_Z>x^2/4</Meijer_Z>
</formula>
```

これは式 (22) の右辺の情報を記述したものである。このような公式に関するデータの集まりが G 関数リレーションには格納されている。この G 関数リレーションと (D) で得られた G 関数表現の式から得られる特徴量をもとにパターンマッチングを行ない、それに対応する公式番号を抽出する。

(F) データベース検索 2
(C) と作業は同様である。(E) で得られた公式番号をキーとして XML 形式で書かれた一般の表現の式を求める。

4 おわりに

本研究では、数式処理システムに依存しない一部の数学公式データベースの構築と数学公式データベースを利用した積分の演算を行うシステムの構築を行った。
今後の課題としては

- 現在、対象とする関数が初等関数のみであるので、初等関数以外の数学公式データベースを増やす
- 各数式処理システムとつなぐシステムの構築

が挙げられる。

参 考 文 献

[1] The OpenMath Society
http://www.openmath.org

[2] The OpenMath Standard
http://www.nag.co.uk/projects/OpenMath/corecd

[3] OpenMath Core Content Dictionaries
http://www.nag.co.uk/projects/OpenMath/corecd

[4] Extensible Markup Language (XML)
http://www.w3.org/XML/

[5] World-Wide Web Consortium,Document Object Model Level2
http://www.w3.org/TR/DOM-Level2-Core

[6] Adamchik V.S. and Marichev O.I., The Algorithm for calculating integrals of hypergeometric type functions and its realization in reduce system,,proceedings of ISSAC '90,pp.212-221,1990

[7] kelly Rozch,Hypergeometric Function Representations, Proc.ISSAC'96,pp.301-308,1996

[8] Kelly Rozch,Meijer G Function Representations,proceedings of ISSAC '97,pp.205-211,1997

[9] 佐々木建昭、増永良文、阿部昭博、元吉文男、三枝義典、佐々木睦子：数式処理システム GAL における数学公式データベース、第２９回プログラミングシンポジウム、1988.1

[10] 三枝義典、阿部昭博、佐々木建昭、増永良文、佐々木睦子：数式処理システム GAL における数学公式データベースのインデキシング手法、信学論 D-1,Vol.J74-D-I,pp.577-585,1991