

## 数式処理機能付き数式エディタの開発

内山 靖文

工学院大学 大学院 電気工学専攻

三好 和憲

工学院大学 電子工学科

パーソナルコンピュータ上で、我々が日常使用する表現で数式を表示編集できる数式エディタを装備し、入力された数式に対して評価、可視化を行うシステムを開発した。ここでは前回報告した本システムをさらに拡張したものについて報告する。

Development of formula editor  
with formula manipulating system

Yasufumi Uchiyama

Graduate school of electrical engineering, Kogakuin Univ.

Kazunori Miyoshi

Department of electronics, Kogakuin Univ.

A formula editor which can show and edit formulae in usual textbook format is developed. This system was reported the last time, and expanded one is reported here.

## 1. はじめに

現在存在する数式処理システムは非常に高機能ではあるが、そのために操作が難解で、とても初心者が手軽に使えるものではない。一方、教育目的としてこれらのシステムを使用する場合、ほとんどのユーザーはコンピュータにはあまりなじみがないといった初心者であり、教育という目的に至るまでにそのシステム自体を理解しなければならないという問題が生じると思われる。

このような背景から、初心者にも手軽に使えることをめざして以下のような機能を持つシステムを開発し、参考文献[1]で報告した。

- ① 数式を2次元表示し、編集できる数式エディタ
- ② エディタにより入力された数式をグラフ化する
- ③ 多倍長演算による簡単な演算機能
- ④ 数式の構造を構造木を用いて視覚化する
- ⑤ 数式をポーランド、逆ポーランド記法に変換して表示

本稿では、これをさらに拡張し、機能の充実をはかったシステムの開発について報告する。

## 2. システム構成

本システムでは各機能はモジュール化されており、各ライブラリモジュール内ではそれぞれの機能がオブジェクトとして定義されている。本システムの利用法としては、図1のように実行ファイルをそのまま利用する方法と、図2のようにユーザーが必要なライブラリを各自のアプリケーションに組み込んで利用する方法がある。(参考文献[1] 3.「マッスル」のシステム構成 参照)

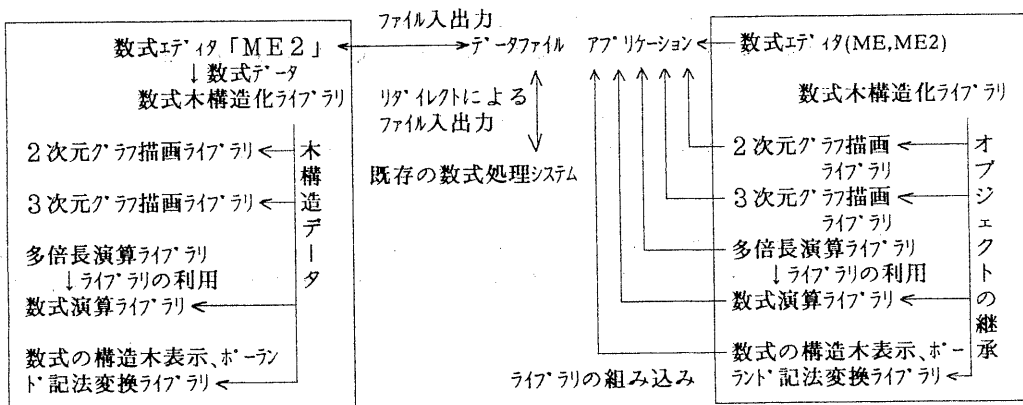


図1 「マッスル」のシステム構成 (その1)

図2 「マッスル」のシステム構成 (その2)

### 3. 数式エディタ

#### 3.1 数式のエディット

- ① 数式の2次元形式表示でのエディット  
数式を

$$\frac{x^2 + y^2}{x^2 - y^2}$$

のように2次元的に表示し編集することが出来る。本システムでは次のような数式が2次元形式表示で入力可能である。

- ・分数
- ・指数 (上付き文字)
- ・ルート
- ・下付き文字
- ・積分

- ② キリシヤ文字の入力

カナキーをロックして英字キーを入力することでギリシヤ文字を入力可能。これは後で述べるTeX出力機能に伴って追加された機能である。通常の出力の際には、 $\alpha$ ならalphaというように読みで出力される。

- ③ エディタ終了時の数式の文法エラーチェック機能

数式の文法的誤り (括弧が対応していない等) をチェックする機能を持つ。エラーが発生した場合はその位置にカーソルを移動し、修正を促す。

- ④ マーク機能

1個または複数個の数式にマークをつけて出力すると、エディタ内のすべての数式のうち、マークされたものが出力される。これは、後で述べる2次元グラフ描画オブジェクトの複数式化に伴って追加された機能であるが、今後はこの機能を用いて3次元グラフの複数式化も実現する予定である。

#### 3.2 ファイルへの入出力機能

- ① REDUCEからの入力

従来の内部形式数式データのファイルからの読み込み、保存、出力形式数式データの出力 (参考文献 [1] 4.2 数式エディタ「ME2」参照) に加え、新たにMS-DOSのリダイレクトによるREDUCEの出力をエディタ内に取り込むことが可能となった。

- ② TeX出力機能

エディットした数式を組版ソフトであるTeXのソースで出力することが可能。TeXでの数式の表現は何種類かあるので、ユーザーが出力表現を自由に設定できるようにした。(デフォルトでは ¥[ 数式 ¥])

設定例

\$ 数式 \$+

各数式の先頭の文字列を\$, 後ろの文字列を\$に設定

¥[ 数式 ¥]

各数式の先頭の文字列を¥[、後ろの文字列を¥]に設定

`\begin{eqnarray*}` 数式ブロックの先頭の文字列を`\begin{eqnarray*}`に  
`&& 数式` `¥¥` 各数式の先頭の文字列を`&&`、後ろの文字列を`¥¥`に  
`&& 数式` `¥¥`  
`\end{eqnarray*}` 数式ブロックの後ろの文字列を`\end{eqnarray*}`に設定  
 (eqnarray\*は&と&で囲まれた文字の位置をそろえる。この場合  
 は数式先頭で位置揃えを行う。)

### 3.3 出力式の設定

数式を出力する際の出力表現を従来よりも柔軟に設定可能となった。具体的には式を構成する部分式をアルファベットに置き換えて、出力の表現を文字列の形で設定する。例えば、分数の場合、分子の式をA、分母の式をBと置いて、デフォルトではA/Bとなっているが、`div(A, B)`のように関数の形で出力することが可能である。

### 3.4 数式の内部データ表現と出力データ表現

数式エディタ内では数式データはString型(文字列型)で与えられている。数式データの内、1行で表現できる部分についてはそのままの表現となっているが、「ME」では数式の2次元表示を採用しているため、内部数式データの中にこれを認識するための専用のコード(2次元表示の始まり、終わり、パラメータの区切りを表すコード)が必要となる。そこで、それを考慮して定義したものが表1のようなデータ表現である。

また、内部データの例と、それに対応した数式の出力データの例も示す。ファイルへの出力、プログラム中の関数によるデータの受け渡しにはこの表現が用いられる(これは前述の通りユーザーが再設定可能である)。なお、従来は出力式の中に不必要な括弧が存在する場合があったが、これは下の例でも示したように修正した。これは、括弧内の部分式の本を作り、そのルートとなる部分、つまり括弧内の最も結合の弱い演算子と、まわりの演算子の結合の強さを比較することで、その括弧が必要かどうかを判断している。

表1 数式のデータ表現

	始まり	データの区切り	終わり
分 数	\$80	\$90	\$A0
指 数	\$81	—	\$A1
ル ー ト	\$82	\$92	\$A2
積 分	\$83	\$93	\$A3
下付き文字	\$84	—	\$A4

#### 内部データの例

$$\frac{x}{y} \rightarrow \$80 x \$90 y \$A0$$

$$\frac{x+y}{a+\frac{b}{c}} \rightarrow \$80 x+y \$90 a+ \$80 b \$90 c \$A0 \$A0$$

$$\{(a+b)/c\}^{x+y} \rightarrow \{(a+b)/c\} \$81 x+y \$A1$$

#### 出力データの例

$$\$80 x \$90 y \$A0 \rightarrow x/y$$

$$\$80 x+y \$90 a+ \$80 b \$90 c \$A0 \$A0 \rightarrow (x+y)/(a+b/c)$$

$$\{(a+b)/c\} \$81 x+y \$A1 \rightarrow ((a+b)/c)^{(x+y)}$$

※ \$ は 16 進数を示す

#### 4. 数式木構造化ライブラリ

数式木構造化ライブラリは、与えられた数式をポインタの鎖を用いた木構造データに変換するものである（木構造データについては参考文献 [1] 5. 数式木構造化ライブラリ 参照）。本システムでは、各オブジェクトは数式木構造化オブジェクトの下位に定義されており、ここで作られた木構造データを利用してそれぞれの機能を実現する。

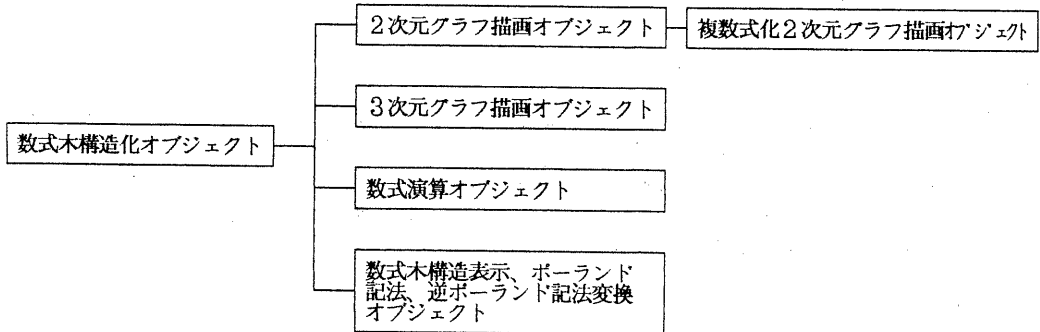


図3 数式木構造化ライブラリを中心としたオブジェクト構成

#### 5. グラフ描画ライブラリ

2次元グラフ描画ライブラリ及び3次元グラフ描画ライブラリは与えられた数式（1変数関数，2変数関数）をグラフ化するものである。グラフ描画ライブラリの機能の詳細は参考文献 [1] 6. 1 グラフ描画ライブラリの機能 すでに解説しているので、ここでは拡張された部分について述べることにする。

##### ① グラフを描く（2次元グラフ）

前バージョンではグラフ化の対象となる数式は1つであった（主変数以外の変数を変化させて重ねて描く機能はあったが、基本的には数式は1つだった）。つまり、数式エディタの式選択カーソル上の数式が2次元グラフ描画オブジェクトに渡されてそれをグラフ化していたが、現バージョンでは数式エディタでマークした1個又は複数個の数式のリストを渡し、これらを重ねて描けるようになった。グラフの複数化を実現するオブジェクトは従来の2次元グラフ描画オブジェクトの下位に定義されている。数式エディタでマークしている数式がある場合は複数化されたオブジェクトのインスタンスを生成し、マークされていない場合は従来のオブジェクトのインスタンスをその都度、動的に生成しているので、機能は向上してもそれほどメモリを圧迫することはない。

##### ② 視点の変更（3次元グラフ）

視点の変更はメニューから値を入力する方法のほかに、グラフが表示されているときにカーソルキーで変更する方法を追加。カーソルキーを押すと、視点が5度ずつ変化し、その都度リアルタイムでグラフが変化する。

③ 隠線処理 (3次元グラフ)

最大最小法による隠線処理を行う際、隠線を点線で表示するか、全く表示しないかを選択することができる。

④ 座標軸の表示 (3次元グラフ)

座標軸の表示、非表示を選択できる。隠線処理を行う場合、座標軸の隠れた部分は点線で表示される。

6. まとめ

今回は参考文献 [1] のシステムをさらに拡張したものを紹介した。本システムの機能の中で、ここで述べていないもの (多倍長演算ライブラリ、数式演算ライブラリ) については前回と変わっていないので参考文献 [1] を参照されたい。今後も本システムの開発において最も重視したコンセプト、「初心者でも容易に扱える多機能なシステムを目指す」を崩さずに次のような課題に取り組みたいと考えている。

・各機能の拡張

数式エディタで2次元表示でエディットできる数式の種類を増やす、グラフ描画で主変数の定義域を指定可能にする、あるいは媒介変数を用いた式のグラフ化を可能にする、等が挙げられる。

・数式の評価に本格的な数式処理を適用する

現段階では数式処理を行いたい場合は、ファイルを介して外部の数式処理システムに渡す必要があるが、これを「マッスル」内部で行えるようにする。その手始めとして、まずはPascal上でLispライクなプログラムを動作可能にする環境を作るリスト処理ライブラリの開発を考えている。

参考文献

- [1] 内山 靖文, 三好 和憲 「統合数式エディタの開発と数学教育への応用」,  
情報処理学会研究報告 93-CE-27, Vol.93, No.47, PP.27-3
- [2] 奥村 晴彦 C言語による最新アルゴリズム辞典 技術評論社
- [3] 奥村 晴彦 LaTeX美文書作成入門 技術評論社
- [4] 宮地 利雄 データ構造とプログラミング 昭晃堂
- [5] R.BRENT MP(A MULTIPLE-PRECISION ARITHMETIC PACKAGE)ソースリスト