

汎用数式処理システム (KFMS) について

慶大 工学部 小林 利臣

§ 1. はじめに

最近，各研究所および大学などにおいて電子計算機によって数式処理を行なったという報告がよく聞かれるようになってきたが，慶応義塾大学でも汎用数式処理システム（KFMS と呼ぶ）を開発しているので，ここにその概要を述べることにする。数式処理という事柄自体に数多くの問題点があるが，この報告ではKFMSの紹介ということを主目的としたので，数式処理の問題点については触れないことにする。さらに，KFMSの内部構成および処理方法などについては紙面の都合上省略することにしたが，興味のある方は参考文献[4]を参照していただきたい。

§ 2. 汎用システムについて

数式処理システムを設計するにあたってまず最初に問題

になるのが、汎用にするか特殊目的用にするかということがある。確かにある特定の問題を解くためには特殊目的用の方が能率よりが、他の問題を解くためにはまたそれ用の数式処理システムを作らねばならない。このようなシステムしか存在しないとすると、何かの必要にせまられてある数式処理の問題を解こうとするときには、まず長い時間をかけてそれ用のシステムを作らねばならないことになる。これでは一般の人は電子計算機で数式処理を行なおうと思わなくなってしまう。数式処理という対象に興味を持つ人たちの研究分野になってしまう。そのために、一般の数式処理にとって本質的に必要な基本オペレーションを簡単にプログラムできるようにしている汎用システムが必要になってくる。どのようなものが基本オペレーションなのかとか、そのような基本オペレーションをすべて含むような汎用システムが実現可能かなどといった問題点があるが、電子計算機による数式処理をもっと一般的にするためには、以上の問題点を考慮しつつ、汎用システムの開発を行なわなければならないだろう。

§3. KFMSの特徴

(1) 汎用システムである。

現在のところ基本オペレーションは13種(式定義,

積分，微分，置換，展開，単純化，整理項，文字の整理，係数抜取，式の消去，統計，印刷，複写）であるが，さらに開発中であり，これからも増える予定である。

(2)処理方法がインタプリティブ（interpretive）である。

処理速度などの面から見れば長所とは言えないうが，人間との対話形式を取り入れたり，TSS用にするためには必要条件となる。

(3)簡単なKFMS言語を使用する。

1つのスタートメントと1つの基本オペレーションがはっきり対応しているので，ある数式処理の問題を解くにあたっては，どのような順序で基本オペレーションを施すかさえ決めれば，簡単にKFMS言語を用いてプログラムを組むことができる。

(4)積分機能が存在する。

処理の方法はSAINTで用いているアルゴリズムとほぼ同じで，与えられた数式を，あらかじめ組み込まれている22個の基本公式と一致するまで置換法などを用いて変形してゆくといった方法を採用している。もちろん現状では，すべての式を積分できるわけではなく，部分積分や簡単な置換積分を行なうこともできる。さらに現在，積分できない式については適当と思われる置換方法を提案する。

(5) 会話形式も可能である。

タイプライタからプログラムを入力することもできるので、いくつかの処理結果を見てから次のステートメントを決めることもできる。その点で、固定したプログラムしか処理できないシステムに比べてはるかに有利である。

§ 4. K F M S の使用法

(1) 処理形式

(a) バッチ処理形式

ステートメントをカードに穿孔しておき、そのカードをカード・リーダーから入力する。このとき処理結果はライン・プリンタに打ち出される。

(b) 会話形式

ステートメントをタイプライタから入力する。このとき処理結果はタイプライタに打ち出される。

なお使用計算機は T O S B A C 3 4 0 0 - 3 0 (1 6 K 語) で、周辺機器類は処理形式によって異なる。システム作成言語はアセンブラである。

(2) K F M S 言語

(a) 一般的規約

(i) 変数名

8文字以内の英数字。ただし最初の文字は英字に限る。K F M Sでは、式変数と原子変数との区別はなく、式が割り充てられている (assigned) 変数は式変数と見做され、式が割り充てられていない変数は原子変数と見做される。

(ii) 数値

F O R T R A N で用いられる I タイプ, F タイプ, E タイプ表現が可能である。その他に帯分数を除く分数表現も可能である。

(iii) オペレータ

+ , - , / , * , ** , E X P , L O G , S I N , C O S , T A N , C O T , S E C , C O S E C , A R C S I N , A R C C O S , A R C T A N , A R C C O T , A R C S E C , A R C C O S E C を使うことができる。

(b) ステートメント

(i) 式定義

$U_i = \text{式}$

右辺に書かれた式を変数 U_i に割り充てる。

式の書き方は F O R T R A N と同じで、記号としては § 4.1.a で述べたものだけを使う。

例. $Y = \text{S I N} (A * X + B)$

(ii) 積分

$$U_i = \text{I N T E G} (U_j, U_k)$$

式 U_j を変数 U_k について積分して，その結果を U_i とする。

$$\text{例. } YI = \text{I N T E G} (Y, X)$$

$$\text{結果. } YI = -1 / A * \text{C O S} (A * X + B)$$

(iii) 微分

$$U_i = \text{D I F} (n) (U_j, U_k)$$

式 U_j を変数 U_k について n 回微分して，その結果を U_i とする。ただし $n=1$ のときは，(1) を省略してよい。

$$\text{例. } YD = \text{D I F} (Y, X)$$

$$YDD = \text{D I F} (2) (Y, X)$$

$$\text{結果. } YD = A * \text{C O S} (A * X + B)$$

$$YDD = -A * * 2 * \text{S I N} (A * X + B)$$

(iv) 置換

$$U_i = \text{S U B S T} (U_j) (U_k, U_l) \dots$$

式 U_j に関して，変数 U_k を変数 U_l に置き換えて，その結果を U_i とする。

$$U_i = \text{S U B S T} (U_j) (U_k,) \dots$$

式 U_j に関して，変数 U_k をその対応する式に置き換えて，その結果を U_i とする。

PL/I-FORMACでは式変数が右辺に現われたらすぐに対応する式で置き換えてしまいが、KFMSでは置換ステートメントを使わないう限り、対応する式で置き換えることはしない。即ち式と変数の使い方に因ってユーザーに任意性を残してある。

例. $X = A + B$

$W = C + Z$

$Z = E + F$

$XS = \text{SUBST}(X)(A, W)$

$WS = \text{SUBST}(W)(Z,)$

結果. $XS = W + B$

$WS = C + E + F$

(V) 展開

$U_i = \text{EXPAND}(U_j)$

式 U_j を展開して、その結果を U_i とする。

例. $A = (X + B) ** 2 + (B + C) * D$

$AE = \text{EXPAND}(A)$

結果. $AE = X ** 2 + 2 * X * B + B ** 2 + B * D + C * D$

(VI) 複写

$U_i = \text{COPY}(U_j)$

式 U_j を複写して，その結果を U_i とする。

例． $YC = COPY(Y)$

結果． $YC = SIN(A * X + B)$

(vi) 整理項

$U_i = ORDER(U_j, opt)(U_k, U_l, \dots)$

多項式 U_j に関して，変数 U_k の次数を比べて opt の方向に項を並べかえる。 U_k 以後の指定があれば， U_k に関して同じ次数のいくつかの項について U_k を整理する。以下同様なことを行なう。 opt は INC または DEC である。

例． $ATT\bar{O} = \bar{O} * D * \bar{O} * R * \bar{O} * K * * 2 *$

$U + \bar{O} - K * * 3 + R * * 3 + K$

$A\bar{O} = ORDER(ATT\bar{O}, DEC)(\bar{O}, K, D)$

結果． $A\bar{O} = \bar{O} * * 3 * K * * 2 * D * R *$

$U + \bar{O} - K * * 3 + K + R * * 3$

(vii) 文字の整理

$U_i = LETORDER(U_j, opt)(U_k, U_l, \dots)$

多項式 U_j に関して，整理項と同じことを行なり，項の中の文字を $U_k, U_l, \dots$ の順に並べる。

例． $AL = LETORDER(ATT\bar{O}, INC)(K, D)$

結果. $AL = R * * 3 + \bar{O} + K + K * * 2 * D * U * \bar{O} * * 3 * R - K * * 3$

(ix) 係数抜取

$U_i = COEF(U_j, U_k)$

多項式 U_j の中にある単項式 U_k の係数を取りだし、それを U_i とする。もちろん U_k は1つの変数であってもよい。

例. $AC = COEF(ATT\bar{O}, R)$

結果. $AC = \bar{O} * * 3 * K * * 2 * D * U$

(x) 単純化

$SIMPLIFY(U_i) \dots$

式 U_i を単純化する。

例. $D = SIN(O) * X + EXP(X) * * 1 - 4 * 8 + (A + A) * 1$

$SIMPLIFY(D)$

結果. $D = EXP(X) - 32 + 2 * A$

(xi) 式の消去

$ERASE(U_i) \dots$

式 U_i の用いていたリストを消去する。そのリストセルは、フリー・リストが足りなくなってきたときにガーベージとして使われる。

(xii) 統計

CENSUS (U_i) ...

式 U_i の用いているリストセルの数を出力する。

例. CENSUS (X) (Y)

結果. X 3

Y 8

(xii)印刷

PRINT U_i

式 U_i を出力する。

PRINT + U_i

式 U_i に関して、すべての変数をその対応する式で置き換えたものを出力する。ただしリストの内部構造は変らない。

例. PRINT Y

結果. $Y = \sin(A * X + B)$

例. PRINT + Y

結果. $Y = \sin(A * (A + B) + B)$

注. 現在のところコアの制約上、入力および出力用のサービスルーチンが全部使えなかったために、数式の入出力はすべてバイナリー・オペレータ表現をとっている。即ち1つのオペレータに対して必ず1組の括弧を必要とする。

なお使用例は印刷の都合上省略したが、御覧になりた

いは参考文献[3]，[4]を参照して下さい。

§5. 将来の方向

(1) プログラム・モード (Program mode) の採用

現在のKFMSの最大の欠陥は手続き (procedure) を書けなかりとることである。そこでFORTRANに似た制御ステートメント (DO文，GOTO文，IF文など) をとり入れたシステム (KFMS-V71と仮称する。) を作る予定である。これに伴って，V71には多くの機能の追加変更があり，ステートメントの形も大きく変わることになるが，これについては別の報告に譲ることにする。

(2) インタラクショナル・モード (Interaction mode) の採用

現在の会話形式ではまだ真の対話とは言えないので，これを改良して，実行中に計算機の出すコメントに対してユーザが何らかの情報を入力できるようにする。これは数式処理において，計算機が得意とする処理については計算機が，人間が得意とする処理は人間が行なうという思想に基づいている。

(3) 単純化の改良

単純化は数式処理の中で最も難しいところで，KFMSでも機能の面で最も遅れている。そこで単純化の度合をい

くつかのレベルに分け，システムが問題に応じて自動的にあるレベルを指定したり，またはユーザーが任意のレベルを指定したりできるようにする。さらに人間とのインタラクションをとり入れれば，かなりうまく単純化できるようになると思われる。

(4) 積分の充実

単純化が改良されれば積分も必然的に改良される。というのは，今のところうまく置換方法は見つけたものの，置換した結果がうまく単純化されなれたために積分に失敗しているからである。さらに基本公式の選び方およびヒューリスティック変換のための特徴抽出などにおいて，SAINTのアルゴリズムを独自のやり方に変更しようとも考えている。

(5) TSS化

一応KFMSはTSS用に設計してあるが，KFMSの方がまだ開発途上で完全なシステムになっていないのでTSSに組み入れていないが，早く実現するようにしたい。TSSに組み込まれれば，端末のタイプライタを自由に使えるのでインタラクションの効果もはるかに大きくなり，数式処理もかなり一般的にそして気楽に使えるようになるだろう。

参考文献

- [1] 大野端夫, "数式処理システムについて",
第9回プログラミング・シンポジウム報告集, 1月, 1968.
- [2] 大野端夫, "数式処理システム",
修士論文, 慶応義塾大学, 1968.
- [3] 小林利臣, "KFMSの概要",
TOSBAC研究会資料, 4月, 1969.
- [4] 小林利臣, "数式処理システム(記号微積分)",
学士論文, 慶応義塾大学, 1969.
- [5] Slagle, J.R., "A Heuristic Program that Solves
Symbolic Integration Problems in Freshmen Calculus",
J.ACM., vol. 10, 1963.