

数式処理ワークステーション機能 の開発

対馬勝英, 広田健児 (大阪電気通信大)

1. 序

我々は長年にわたり数式処理の良好なヒューマンインタフェースについて研究を進めてきた。数式処理システムの持つ強力な機能と応用へのポテンシャルティにもかかわらず広範な利用が実現されていないのが現状である。その理由として数式処理システムのヒューマンインタフェースが貧弱であることがあげられる。

元来、特殊な研究者のツールとして開発された数式処理システムはインタフェースの不足をユーザの叡知により補うかたちで利用されてきた。しかし、広範な利用を期待するならば、利用者にこの種の叡知を期待することはできない。従って、まず、良好なヒューマンインタフェースを作成し、それをもとに数学処理システムの普及を図るという二段階のステップをとる必要がある。

本論文は数式処理システムの広範な普及をはかる為に我々の行った良好なヒューマンインタフェースの開発の報告である。

数式処理システムのヒューマンインタフェースの改善は次の二点に分類される。¹⁾

- ① 数式処理そのものの改善
- ② 入出力に関する改善

我々は人間が紙の上で行う全ての操作が数式処理システムの上でも可能になることが数式処理システムのヒューマンインタフェースの完成された姿であると考えた。それを実現するには①、②の双方についての改善が必要であり、どちらか一方の改善ではうまくゆかない。

①に関する研究は数式処理システムの扱える数学的な実体の拡大であり、また、形式的な操作を可能にする試みでもある。これに関しては I C A S 機能として完成している。^{2), 3), 4)}

一方、②に関する研究が、今回、完成した入出力に関するワークステーション機能を実現するソフトウェア EWS である。それらの各々について、2章、3章で詳述し、4章において両者の機能を複合した応用例について述べる。

2. 数式処理機能の拡大—— I C A S 機能の作成

REDUCE, muMATH 等の広く利用されている数式処理システムにおいては、整式の処理に重点がおかれ、以下に述べる様に関数の形式的取扱やオペラ、微係数、等を含む解析的な操作については強力ではない。

従って、例えば、 $F(x)=x^3$ の差分を求める際に

$$\begin{aligned} F(x) &= F(x+h)-F(x) & 1) \\ &= (x+h)^3 - x^3 & 2) \\ &= 3hx^2 + 3h^2x + h^3 & 3) \end{aligned}$$

に関して

$$\text{FUNCTION } F(x), x^3, \text{ENDFUN\$} \quad 4)$$

と $F(x)$ を定義したとして

$$F(x+h)-F(x) \text{ -----} \rightarrow 3*h*x^2+3*h^2*x+h^3 \quad 5)$$

を求めることは出来る。

しかし、1), 2) を得ることは困難である。更に、高階の差分にかんしては

$$\Delta^2 F(x) \quad 6)$$

$$\text{-----} \rightarrow \Delta(F(x+h)-F(x)) \quad 7)$$

$$\text{-----} \rightarrow \Delta F(x+h) - \Delta F(x) \quad 8)$$

$$\text{-----} \rightarrow F(x+2h)-F(x+h)-(F(x+h)-F(x)) \quad 9)$$

$$\text{-----} \rightarrow F(x+2h)-2F(x+h)+F(x) \quad 10)$$

$$\text{-----} \rightarrow (x+2h)^3 - 2(x+h)^3 + x^3 \quad 11)$$

$$\text{-----} \rightarrow 6h^2x + 6h^3 \quad 12)$$

標準的な処理の内でこれを行うことは困難である。

7), 8) に於いては出力側にオペラが出現する。また、整式として評価されていない関数 $F(x)$ が関数形として出現する。これらが数式処理システムの上で可能であるためには

オペレタの前置型作用
 オペレタの実体としての存在
 関数, オペレタの評価の抑止

等が実現されている必要がある。

9), 10) は関数 $F(x)$ の評価の抑止を利用し, 6) の左辺は Δ^2 の様にオペレタの関数の前置型の作用の例である。

さらに, オペレタに関しては,

$$(\Delta + 1)^2 F(x) - (\Delta - 1)^2 F(x) \quad 13)$$

$$= [(\Delta^2 + 2\Delta + 1) - (\Delta^2 - 2\Delta + 1)] F(x) \quad 14)$$

$$= 4 \Delta F(x) \quad 15)$$

$$= 4 [F(x+h) - F(x)] \quad 16)$$

の様にもオペレタ代数が利用される場合もある。

1) から 16) に示した全ての処理を可能とする数式処理機能を実現することで紙の上での鉛筆による記号処理と同じ発想, 同じアルゴリズムでの処理が数式処理システム上でも可能となる。

別の観点より要約すると 1 図に示す様に入力, 出力側を問わずオペレタ, 関数, 整式が独立した実体として存在する数式処理機能の実現が, 柔軟な数式処理の実現に欠かせない。

我々はこの様な機能を ICAS とせず, まず, muMATH 上に実現した。現在では, PROLOG で記述した独自のシステムとして ICAS 機能を持つ数式処理システム CASP を 16 ビットパソコン, 32 ビットワークステーションの上に実現している。この種のシステムは数理科学の研究に於いて種々の新しい応用を可能とするがそれについては参考文献をみられたい。⁵⁾

11図 紙上の操作の例

$$(1+x)^3 = 1 + 3x + 3x^2 + x^3$$

$$\int(x+x^5)dx = \frac{x^2}{2} + \frac{x^6}{6}$$

$$(1+3x+3x^2) + \left(\frac{x^2}{2} + \frac{x^6}{6}\right)$$

$$= 1 + 3x + \frac{7}{2}x^2 + \frac{x^6}{6}$$

3 入出力の環境の改善 —— EWSの作成

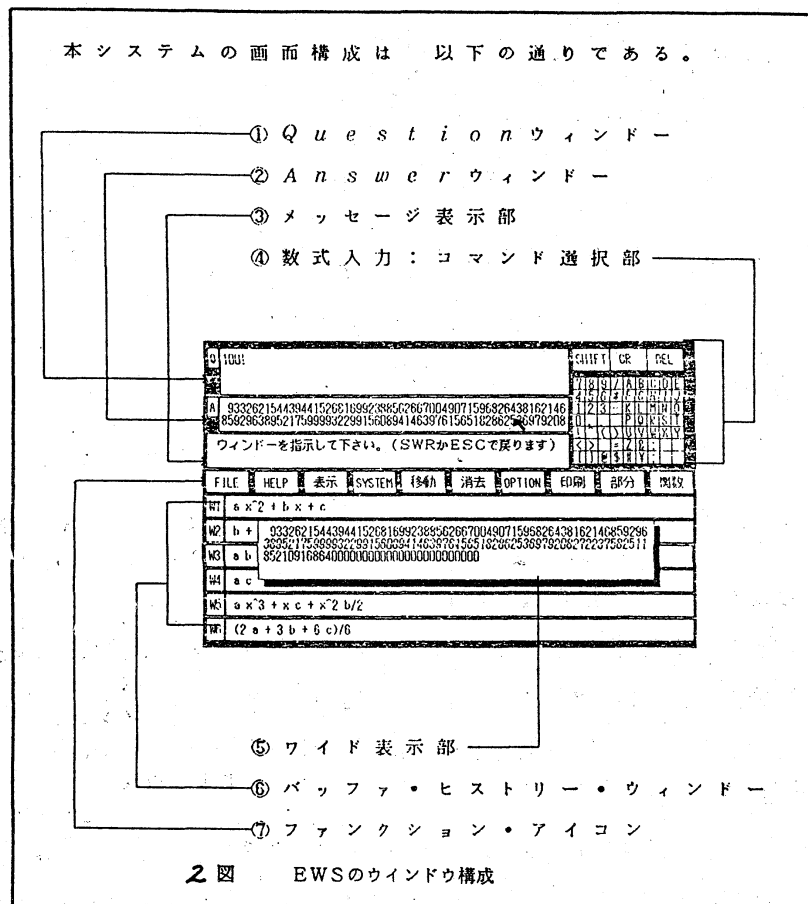
1図に示す様に、人間は紙の上で整式、微係数、関数の、オペレタ等に対し、部分的抜き取り、抜き取り部分に対する命名、転記、部分的に抜き取ったものを作用さす、等の操作を行う。これらは、1)に述べた数学的な操作とは別の区別、移動、一時的保存、命名等の認知に続く空間的な操作と言える。

従来の数式処理システムは、二次元的出力等については考慮していても、出力結果の抜き取り、保存、命名を空間的に行うことは考慮していない。これは、人間の紙の上で行う数学に関する操作の一部分しか、数式処理システム上で行えぬことを意味する。ユーザが紙の上から数式処理システムへ移行する際に、処理の等価的転換が必要となり、この煩わしさが数式処理の普及を妨げていた大きな原因と言える。

この種の入出力の改善を行うため、muMATH上にEWSと名付けた数式処理ワークステーション機能を実現した。⁶⁾

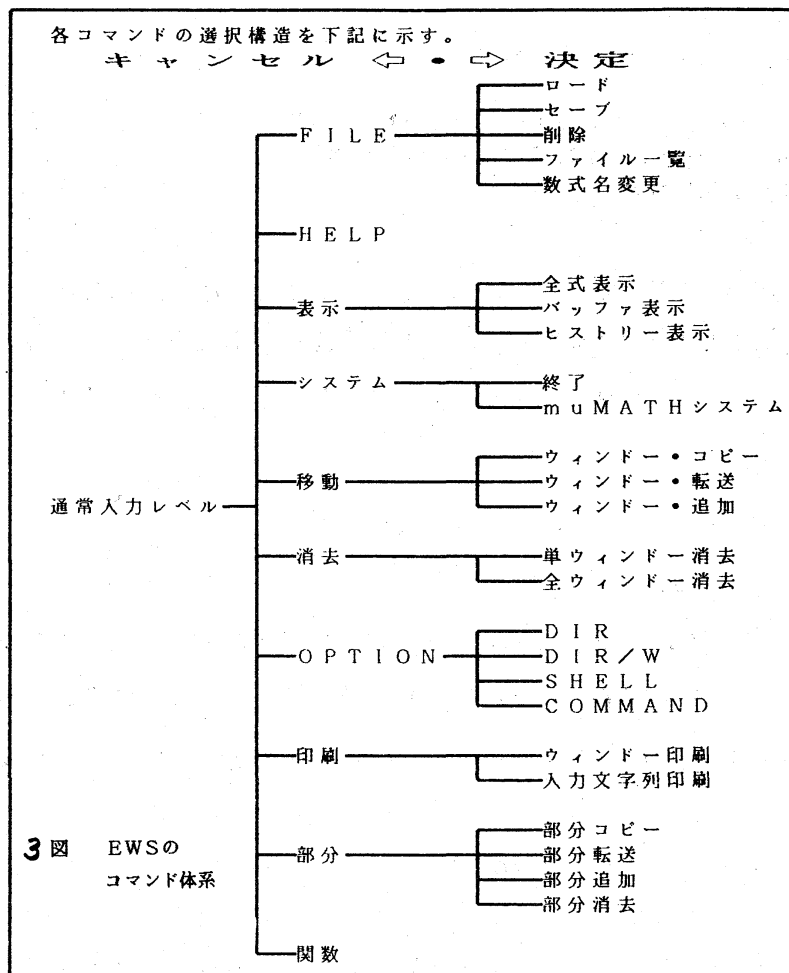
以下に、EWSの設計指針を列記する。

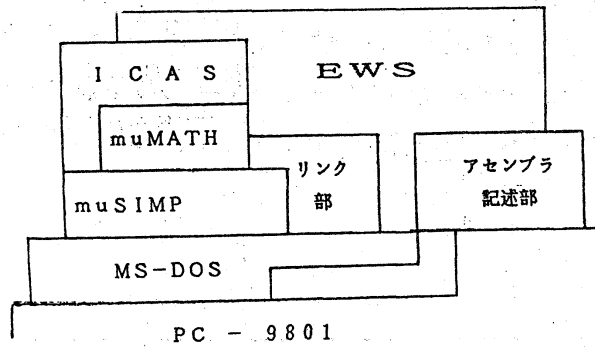
- 1) 全ての作業はマウスとマイコンにより進行する。
- 2) 並列的に見ることのできるスタックを用意し、作業結果を一時的に保存するスタックとして利用する。
- 3) muMATH, ICAS の機能をそのまま保存する。
- 4) 整式、関数、オペレタ等の部分的抜き取りと移動、コピーを可能とする。
- 5) 抜き取った関数、オペレタは評価可能、作用可能である。
- 6) ディスク上のファイルは2)のスタックの延長として、2)と同じやり方で利用できる。



ユーザが対話する画面は2図の様に作られ、Q(入力欄),A(処理結果の欄)において対話的に行った質問と得られた結果を、W1 からW6の6本の作業様スタックに、マウスのクリックにより移動またはコピーすることができる。

また、作業用スタックの上部に10ケのアイコンがあり、これをマウスでクリックすることで、3図に示した10ケのコマンドを選択し、特定のコマンドモードに入ることができる。また、画面の右上にはキーボードの代わりにキャラクタ、数字のマトリクスがあり、これをキックすることで一字ずつの入力を行うこともできる。SHIFT, CR, DELの入力もマウスのみ可能である。





4 図 EWS のソフトウェア構造

本ワークステーションはPC9801のMS-DOS上で作動する。muSIMPとEWSはリンク部を通じてデータを交換する。我々の開発した数式処理機能ICAS (70Kbyte) が利用出来る様に設計されている。

この様にして設計したEWSは4図に示した様にC言語で記述されmuMATH/muSIMPと併用して使用する。数式処理を行うmuMATH, ICAS部はEWSの子プロセスとしてよびだされ実行られ、画面やマウスに関するグラフィックスはアセンブラで記述したが、EWSよりよびだされる。

4. 使用例

5図に整式の処理に関する使用例を示した。展開した整式の一部を抜き取りスタックW1に貯えた。また、Taylor展開した関数の一部を抜き取りスタックW2に貯える。続いて、それらをQ欄に移動し、それらの差を求めた。それらの概念図と処理の実際を5図のA)とB)に各々示した。

次に、6図にオペレタに関する使用例を示した。あらかじめ Dx と定義し、 $(1+Dx)^3$ を作成し、 $3Dx+3Dx^2$ を抜き取り、W1に貯えた。次いで $(1-Dx)^3$ を作成し、 $-3Dx+3Dx^2$ を抜き取り、W2に貯えた。W1とW2をQの上で加算し、それを $F(x)=x^3$ に作用させると、オペレタが作用し、解がA欄に得られた。

この様に、整式、オペレタによらず、抜き取ったものは独立した数学的な実体であり、作用させることができ、また作用を受けることができる。この様に、人間が紙の上で行っている操作をほぼ実現することができた。

$$\begin{array}{l}
 (1+x)^3 \quad \text{-----} \rightarrow 1 + \boxed{3*x + 3*x^2} + x^3 \\
 \text{TAYLOR(\#E x, x, 0, 4)} \quad \text{-----} \rightarrow 1 + \boxed{x + x^2/2} + x^3/6 + x^4/24 \\
 \text{-----} \rightarrow 2*x + 5*x^2/2
 \end{array}$$

(操作)

$(1+x)^3$

W1 <--- A

マウスによるスタックW1への移動

TAYLOR(#E x, x, 0, 4)

W2 <--- A

マウスによるスタックW2への移動

Q <--- (W1)

マウスによる抜き取り移動

-

Q <--- (W2)

マウスによる抜き取り移動(加算的移動)

CR

A 欄

結果の入手

5 図 整式の操作の例

$$\begin{aligned}
 (1 + Dx)^3 &\longrightarrow 1 + \boxed{3*Dx + 3*Dx^2} + Dx^3 \\
 (1 - Dx)^3 &\longrightarrow 1 - \boxed{3*Dx + 3*Dx^2} - Dx^3 \\
 (\bigcirc + \square) \cdot x^3 &\longrightarrow 36 * x
 \end{aligned}$$

(操作)

DEFINE F(x), x^3, NEDDEF;

DEFOP Dx, DIF(#EXP, x), ENDOP;

(1+Dx)^3

W1 <----- (Q)

(1-Dx)^3

W2 <----- (Q)

Q <----- W1

Q <----- W2

.F(x)

CR

関数F(x)の定義

オペレタDxの定義

オペレタの展開

Q 欄の部分抜き取りと転送

オペレタの展開

Q 欄の部分抜き取りと転送

マウスによる移動

マウスによる加算移動

オペランドの入力

6 図 オペレタの操作の例

2つの例で示したものの他に、 $F_x(x), F_{xx}(x)$ の様な微係数として表記された関数に対して抜き取り、移動、コピーを行いそれらに対して更にオペラを作用させたり、値を評価することもできる。ICAS機能が関数、変数、微係数、オペラを独立した情報量として扱い、その評価をコントロールする機能を持っているために、この様な柔軟な機能が実現できた。

5. 結論と展望

この意味で、ICAS機能とEWSはお互いに助け合って、柔軟なヒューマンインタフェースを実現している。この両者を一体化したものをINTCASと呼んでいる。ICAS機能と等価な機能は、より現代的な数式処理システムにおいては等価的には実現できるが、我々がICAS機能を開発した真の狙いは、本論文で述べた形でのヒューマンインタフェースの実現にあった。我々が主張し、かつ、開発してきた良好なヒューマンインタフェースを持つ数式処理機能INTCASがこの様な形で完成したと言える。

この様な数式処理機能は研究者にとっても使い安く、紙の上の処理を数式処理システムの上に於ける等価的な処理に変換する事を不要とする点で真の問題解決の方に注意をむけることが可能となる。

一方、数式処理の広範な普及という観点より考えても、良好なヒューマンインタフェースは利用者の増加に拍車をかけることは言うまでもない。

また、数学教育への数式処理システムの応用^{(7),(8)}に付いても本システムは新しい可能性を拓くものであり、我々はその方向の研究をすでに開始している。

本システムの持つワークステーションとしての機能は未だ原始的なものであり、マルチウインドウ、オブジェクト指向、本格的なビットマップディスプレイ等の導入が考えられるが、実際にINTCASを使用してみると十分、柔軟なものであることが分かる。教育、普及の観点よりはMS-DOS版の現状のもので十分かもしれない。我々は微積分を主としたICASシステムをNEWSの上のK-PROLOGを用いて開発した。これに上記の現代的なワークステーション機能を付加することを開始している。

〔参考文献〕

- 1) 対馬, 「数式処理に於けるマン—マシンインタフェースの改善」, ヒューマンインタフェースシンポジウム論文集 ('87)
- 2) 対馬, 「小型数式処理システム」, 情報処理, 27, 379, ('86)
- 3) 対馬, 加賀, 吉松, 佐藤, 「数式処理を用いた微分の教育」, CAI 学会誌, 5, 59, ('87)
- 4) 対馬, 加賀, 吉松, 佐藤, 「微分教育システムTDCAとICAS機能」, 信学技報 ('87)
- 5) 対馬, 大勝, 稲岡, 佐藤, 「代入法による非線型微分方程式の記号的べき級数解法」, 信学論(D), J70-D, 7, p.1448 ('87)
- 6) 対馬, 広田, 「数式処理ワークステーションの開発と数学教育への応用」, 電子情報通信学会 ET 87- 5, ('87)
- 7) 対馬, 阪野, 「数式処理テストに於ける誤答診断システムSMDIAGの作成」, CAI 学会誌, 4, 1, 5, ('84)
- 8) K.Tsushima, H.Kaga, Y.Yoshimatsu, T.Satou : Teaching Mathematics Using Computer Algebra System (TMCA), Microcomputer and Secondary Education, Elsevier (North Holland), ('87)