設計補助計算のための簡易インタプリタ言語の開発

西松建設(株)　正員　根本隆栄

§1 はじめに

日頃設計にたずさわっている人達から、「電卓で処理するには少々手間のかかる計算があるので電算機で処理したい。しかしながら、プログラム言語を修得してプログラムを組んで処理するだけの時間はない。もっと手軽に電算機を利用することはできないものだろうか」といった相談をしばしば受ける。こうした状況をふまえ、筆者は従来のプログラム言語(FORTRAN, PL/I等)のもつ細かい文法上の規則を大幅に取り除き、プログラミングの経験が殆どなくても数式を簡単に表現し、かつ実行できるインタプリタ形式の簡易言語の開発を試みた。ここに紹介する言語は、内容的には通常の算術代入文, 入出力文, LOOP文, 配列計算文、IF文等ごく基本的な機能しかもたないプリミティブなものである。しかしながら、設計および構造解析の補助計算用としてまずまず使用に耐えるものである。また、FORTRANによる開発済サブルーチンを付加することにより容易に機能拡張をはかることができる点に大きな特徴をもっている。

§2 言語の概要

(1) 言語の特徴

この言語は初心者もしくは補助計算として時々プログラムを作成して使用する程度のユーザーを対象とした。したがって、通常の言語にみられる細かい文法上の規則は大幅に取り除いた。この言語の特徴は次のように要約される。

1) インタプリタ形式を採用しているのでコンパイル、リンケージといった概念が不要であり、初心者にはなじみやすい。
2) 定数、変数は共に整数型、実数型の区別はないので、通常使いなれた変数名で算術式を記述できる。
3) 各種配列計算はすべて一行で記述し実行することができる。
4) 言語はすべてFORTRANで書かれているので、開発済サブルーチンを組み込むことにより容易に機能アップをはかることができる。

(2) モード

この言語は定義モードと実行モードの2つのモードをもつ。

1) 定義モード
プログラム文を読み込み解釈して実行可能なスタックをつくり出す。一般に、IF文, LOOP文を含む場合はこのモードの状態で入力する。

2) 実行モード
定義モードで作成されたスタックを適当な実行コマンドにより実行する。

(3) プログラム文の種類

使用可能なプログラム文とその機能を表-2・1に示す。

表-2・1 プログラム文とその機能

文の種類	機　　能	例	備　　考
代入文	算術式を定義する.	Y=Z*X+Z Y=(COS(X)+COS(-X))/2	三角関数の引数の単位は原則として度である.
READ文	変数および配列要素の値を読み込む	READ X Y A(1 1)	
PRINT文 NLIST文	変数および配列要素の値をプリントする.	PRINT X Y A(1 1) NLIST X Y	NLISTは結果を変数名と共にプリントする
IF文	条件判断を行う	IF(X>=Y) JUMP 10	算術IFとしての機能のみをもつ
JUMP文	指定された文番号へ飛ぶ	JUMP 20	
LOOP文 NEXT文	LOOP, NEXTではさまれた区間を制御変数の値が満足されるまで繰り返す	LOOP 10 X=1 20 1.5 ⋮ NEXT 10	LOOP文の制御変数は整変数でなくてもよい. また増分値は負の値でもよい.
ARRAY文	配列を定義する	ARRAY A(10 10) B(10)	配列はそれが使用される以前であればどの位置で定義してもよい
MREAD文	配列全体の読み込みを行う	MREAD A(* *) MREAD B(I J) J=1 5 I=1 5	
MPRINT文	配列全体のプリントを行う	MPRINT A(* *) MPRINT B(I J) J=1 5 I=1 5	
配列計算文	マトリックスの加減、乗算および各種のマトリックス操作を行う.	A(* *)=B(* *)+C(*) A(*)=B(* *)*C(*) A(* *)=INVERT(A(* *), N) B(* *)=TRANS(B(* *), N) - - - - - - - - - -	[A]の逆マトリックスを求める [B]の転置行列をつくる
基本関数文	解析に必要な種々の関数機能を与える	X=MAX(X1, X2, X3 ····) X(*)=CARDANO(A1, A2, A3, A4) - - - - - - - - - -	3次方程式 $A_1X^3+A_2X^2+A_3X+A_4=0$ の根を求める.
ストリング文	タイトル, コメントをプリントする		

(4)実行コマンド

定義モードで作成されたプログラムは、表-2.2 に示す実行コマンドにより実行させる。

表-2.2 実行コマンド名とその機能

コマンドの種類	機　　能
RUN	定義したプログラムを単に実行する。
LOOP $X=m_1, m_2, m_3$	プログラム全体を X が m_1 から m_2 まで増分値 m_3 により繰返し実行する。
DO $X=X_1\ X_2\ X_3$ ----	プログラム全体を X が X_1, X_2, X_3, ---- の値に対して実行する。
X=1.5, B=10, ----	プログラム内の未定義の変数に対して値を定義する

§3 使用例

以下、実際に使用された例を示す。

(1)例題-1

```
MODE=DEFINE
ARRAY CD(120 3)
10) READ N ALFA X0 Z0
IF(N=0) JUMP 20
L=H*TAN(SITA)
X=L*SIN(ALFA)
Z=L*COS(ALFA)
Y=-H
X=X+X0+XG
Z=Z+Z0+ZG
CD(N 1)=X
CD(N 2)=Y
CD(N 3)=Z
JUMP 10
20) STOP
/
      H=0 SITA=0 XG=0 ZG=18
RUN
2 180 0 -3.5
3 120 3.03 -1.75
4 60 3.03 1.75
5 0 0 3.5
6 -60 -3.03 1.75
7 -120 -3.03 -1.75
8 150 0.75 -1.30
9 30 0.75 1.30
10 -90 -1.5 0
29 150 1.515 -2.62
112 -150 -1.515 -2.62
0 E
    :
    :
SITA=0
RUN
98 30 0.75 1.30
100 150 0.75 -1.30
102 -90 -1.5 0
0 E
/
MODE=DEFIN
SPACE(3)
LOOP 30 I=1 120
SPACE 1
PRINT I CD(I 1) CD(I 2) CD(I 3)
NEXT 30
/
RUN
```

図-3.2 プログラムリスト

図-3.1 の平面図に示されるような杭基礎構造物の立体解析を行った。基礎は直杭、斜杭をもちモデル化のための節点数は約120個となり、図-3.2 に示すプログラムにより節点座標を求めた。座標値の計算方法は図-3.3 を参照されたい。

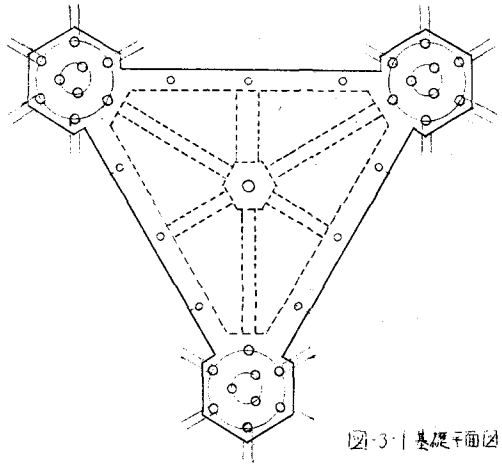

図-3.1 基礎平面図

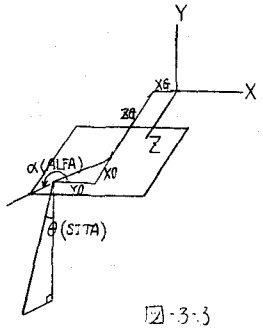

図-3.3

(2) 例題 - 2

```
MODE=DEFINE
* TO DB(A) FROM DB(FLAT)
ARRAY B(10) C(10)
READ B(I) I=1 8
C(1)=B(1)-26
C(2)=B(2)-16
C(3)=B(3)-9
C(4)=B(4)-3
C(5)=B(5)
C(6)=B(6)+1
C(7)=B(7)+1
C(8)=B(8)-1
SUMA=0
SUMC=0
LOOP 10 I=1 8
SUMA=SUMA+10**(B(I)/10)
SUMC=SUMC+10**(C(I)/10)
NEXT 10
LA=10*LOG10(SUMA)
LB=10*LOG10(SUMC)
SPACE(3)
PRINT B(I) I=1 8
SPACE(1)
PRINT C(I) I=1 8
SPACE
NLIST LA LB
/
RUN
84 78 74 73 72 72 65 57
RUN
87 86 79 79 79 79 74 62
RUN
88 87 83 80 80 80 74 65
RUN
98 97 93 90 90 90 84 75
RUN
82 81 76 71 71 70 66 62
```

図-3-4 プログラムリスト

B特性で与えられた周波数分析結果をA特性に変換すると共にオールパスおよび騒音レベルを求めるために、図-3-4 に示すようなプログラムを作成し計算を行った.

[B] —— B特性による音圧レベル

[C] —— A特性による音圧レベル

LA —— オールパス

LB —— 騒音レベル

§4 あとがき

元来、この種の言語は端末機を用いて対話的に処理することが原則であろう。当社にはまだディスプレイが導入されておらず、すべてのジョブはバッチ形式で処理されている。このため、言語の機能不足と相まって利用者にはいま一つ身近に感じられないようである。ハードおよび言語機能の充実が今後の課題である。

現在、APLに代表されるように、端末機の利用を前提として表現が簡潔でかつ強力な機能をもついわゆるエンドユーザー向け言語が際々に普及しつつある。したがって、このような言語の開発は専門のメーカーにまかせるべきものであるかも知れない。しかしながら、新しい言語によるプログラムの開発は、基本サブルーチンを含めてすべて一から作成してゆかなければならず、既に開発した資産としてのプログラムを有効に生かすことはできない。筆者の形式は、開発済のサブルーチン(例えば、逆マトリックス、高次方程式の解法 etc.)を組み込むことにより各種の機能を提供している。すなわち、単にコマンドおよびサブルーチンにデータを引き渡すためのインターフェースを作成すればよく、過去の資産を生かすことができる。

今後の展望としては、種々の設計および解析のためのプログラムをサブルーチンの形で組み込み、適当なコマンドにより実行させると共に、これら目的別プログラム間のインターフェースをこの言語によって記述するといったPCLとしての性格を持たせることが、この言語を更に有用ならしめるために必要となると思われる。

参考文献

中田育男 : コンパイラの技法　　竹内書店新社