

アルゴリズム記法

東洋大 情報工学科 中村慶一

§1. はじめに

電子計算機教育はプログラミング教育から始められ
るべきではない。最初に教えられるべきことは、自
分の頭の中にある処理手順(アルゴリズム)を、正確且つ
簡単に相手に伝達する方法でなければならぬ。こ
れなしに特定のプログラミング言語によって処理手順
を記述する方法を教えられると、その言語でしか通用
しない規則の印象ばかりを強く受けて、たとえばFOR-
TRANでは $3 * A$ は禁止だとみまぐれ、I * 2 番目
という指定は駄目だとみ、4.6 ききみはいけなくて正
の整数についてでなければくり返せないとみ、あるいは
GO TO の連発で論理の流れが自分でもわからなくな
ってしまったといふようなばかばかしい判断が頭
を支配してしまって、自由な発想で自分の処理手順を
書き下す上に重大な障礙を及ぼし勝ちである。

昔毎秒1000回の処理をやつてゐた電子計算機に対しては少しでも計算機の労力と人間的労力とを削減の多い言語で書く意味もあつたが、毎秒百萬回の処理が普通になり、しかも人間次元での1秒の価値が増大してゐる今日では当然要求されるべきものであつた。

このFORTRANに対してすぐれてゐるとしてIBM
が切り替へたPL/Iも、配列の構造データの構造を勝
てはALGOLを受へつゝだものであり、そのALGOL
の改良案に近いPASCALなどによつて教育された場合
でも少いとはいへ特有の有限事項の影響を強く受けた
といふ点ではやはり問題が残ると考えられる。

ここに提案するアルゴリズム記法は他人への伝達あるいは自分へのメモとして指図書を作成する一つの方法であり、これが出来上った段階では高級言語に変換する平均的な5は70~80%、FORTRANでも50%はその作業が完了していると考えても良い位のものであり、いわゆるフローチャートよりも簡単に全体の構造がわかりやすいことが何長である。この方法で導入を行うことをAIEIT(アジア工科大学院)や東洋大工学部で試み成功しているのでその概要を述べる。

§2. アルゴリズム記法の例

個数 N の情報を先頭に N 個の値を並べているデータ
用紙よりヘクトルXに読み込み、

平均值 $MX = \sum_{i=1}^N X_i / N = \bar{X}$

不偏分散の平方根

$$RV = \sqrt{\frac{\sum (X_i - \bar{X})^2}{N-1}} = \sqrt{\frac{\sum X^2 - (\sum X)^2/N}{N-1}}$$

と求め、その値と残差 $X_i - \bar{X}$ ならぬに $MEAN =$
 の左に MX 、 $RMS =$ の右に RV の値を印刷する手順
 をアルゴリズム記法で書いてみよう。

```

配列 X
      (N×1)
read (N, X)
      (N×1)
1)  $N \leftarrow 1$ 
   [  $SX \leftarrow \sum X_i$  ;  $SXX \leftarrow \sum X_i^2$ 
      $MX \leftarrow SX/N$ 
      $RV \leftarrow \sqrt{\frac{SXX - SX^2/N}{N-1}}$ 
     1)  $N \leftarrow N + 1$ 
       [ print ( $X_i$ ,  $X_i - MX$ )
         print ('MEAN=',  $MX$ )
         print ('RMS=',  $RV$ )

```

表現は小学校以来おなじみの数字記号そのままで、
特に説明することもないが、強いてつけ加えれば、い
くつかの文と角かつかで一つの文にまとめられること、
←によって後で説明する登録用紙上の左辺の名前に対
応する値を右辺の式によって求めた値と書き直すこと、
| (1) N は最初から 1 から決まるとしお初めから N をこ
えるまでくり返すこと、I はこれより返しの中心
になる特定の名前であることを示すこと、'RMS='
とはこの字そのものの印字を指令すること、print
は () 内を 1 行に印字すること、read はデータ用紙
から順に登録用紙上の対応場所に値を読み入ることな
どは自然にくみ取ることができる。

名前は大文字を倣用しており添字はふりやちくする
ために小文字にしているが別に区別はない。

もう一つニュートンの近似によって $\sqrt[3]{A}$ を求めた

手順を思いしてみる。ニュートンの近似によれば立方根の*n*回目の近似値 X_n は

$$X_n = \frac{1}{3} \left(2 \cdot X_{n-1} + \frac{A}{X_{n-1}^2} \right)$$

である。また $|X_n - X_{n-1}| \leq 10^{-4}$ となったら計算を打ち切るものとし、データ用紙上には A の値として 45.8, 65 がカンマで切つて並べてあり、データの終りにはその中にならぬ値 0 を入れておくものとする。また後のために関数 RE (次のデータを読みその値を RE に入れたる関数)、関数 CR (X の立方根を精度 E で求め CR に入れたる関数) を作ってテストとしてそれらを使用してみることになる。

関数 RE

[read (X); RE $\leftarrow X$

関数 CR (X, E)

[$X1 \leftarrow X/3$

[$X2 \leftarrow \frac{1}{3} \left(2 \cdot X1 + \frac{X}{X1^2} \right)$ while $|X1 - X2| > E$

[$X1 \leftarrow X2$

CR $\leftarrow X2$

[$A \leftarrow RE$ while $A \neq 0$

[print ($A, CR(A, 10^{-4})$)

45.8, 65, 0;

1個関数はこのように一定の手順全体に名前をつけて辞書に格納し式の中で呼び出して使用できる。文の切れ目が明白でないときは ; で区切っても良い。while の後の条件が成立している間は [] の値についてくり返すもので精度を 10^{-4} としたのは $|X10^{-4}|$ の値のつもりである。

第2機数個の場合は関数でなく手順にする。手続の例として、データサイズ N 、ベクトル名 V を与えて平均値 ME と不偏分散の平方根 RV を導き出す手続 MER は次のようなものになる。

手続 MER (N, V, ME, RV)

[$SX, SXX \leftarrow 0$

[$I \leftarrow 1(1)N$

[$SX \leftarrow SX + V_i$

[$SXX \leftarrow SXX + V_i^2$

ME $\leftarrow \frac{SX}{N}$

RV $\leftarrow \sqrt{\frac{SXX - SX^2/N}{N-1}}$

和をとる部分は $SX \leftarrow \sum_{i=1}^N X_i$ $SXX \leftarrow \sum_{i=1}^N X_i^2$

などとしても同じである。

手続は関数式式中で引用されるのに加えて、文としてその名前を呼ぶ。たとえば大きさ M のベクトル X を用い関数 MEAN と R に入れたら

MER ($M, X, MEAN, R$)

とすればよい。関数手続共に宣言の時の仮パラメータと実行の時の実パラメータは、その何番目ということに対応する名前同志の間で値を交換することになる。

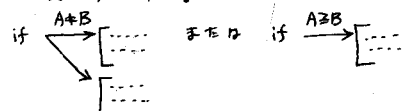
§3. アルゴリズム記法の原則

- (1) 計算を命ずる値は その手順をアルゴリズム記法で示した指図書と、その中で read () という指定に従って読み込まべきデータを順にコマで区切つて並べたデータ用紙を相手に渡す。これに加えて計算を命ぜられた側は、実行の途中で値を保持すべき変数名とそれに対する値を記入するための登録用紙、式の計算途中の値を記入するための計算用紙を用意して計算を実行し、指図書中の print () という指定に示された形式に従って報告用紙に記入する。すでに用意された手続き、関数については別に用意した辞書に記入してあるので、これらは互いに自由を利用できるとし、今後の為にかきつけた手順中有用な部分を一般化した新しい手続きや関数を試みてみるときは、指図書中でその宣言を行つてから引用すればよい。
- (2) 個々の作業単位を文といふが、文の区切りを明白にするために ; を用いても良い。いくつかの文を角かっこでまとめて一つの文とすることもある。角かっこの中で宣言された名前はその範囲内で適用する。これをブロック構造という。
- (3) 論理の流れは原則として指図書中の記入順序に従うが、次の場合に限って順序変更を認める。

A. 選択 ~~← 文の文~~

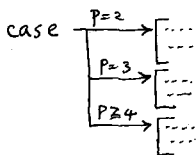
(1) 二選択 (if 文)

条件が真か偽かによって二つのうちの一つの文を選択するもので、それが終れば当然この if 文の次の文を実行する。



(2) 多著撰一 (case 文)

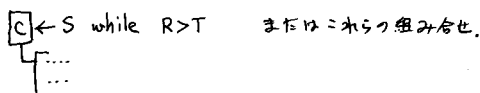
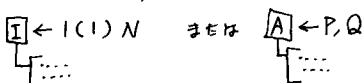
条件によって一つの文を選択するもので、それが終われば当然この case 文の次の文を実行する。



B. 繰返し

(1) 1 個の変数(制御変数)の値を変化させた繰返し

制修変態に次々と代入すべき値がなくなつた場合
 点またはある条件を満足する場合はこの繰り直し
 文 2 次文に移る。(洋ブロッコ構造)



(2) 任意個数の値を累化させた累原し(1回を序と)

あらかじめ計算に格納してある系統であるのは、関数を実行することと命令するもので、それぞれ対応する変パラメタで、計算中の値パラメタを置き換えて実行する^が、終了すれば系統ならその系統と次の文、関数の場合はその呼び出した式の次の項に移る。(彈ゴロツク構造)

C 無条件と仮とし

これは全体の構造を明らかにくちまのてちまり使
わなりちが良いが、あちかじめ ST: なでとコ
ロンを介して文に付けてある各札を指してとびこす
ことを所すする場合もある。この場合は今までと違
つて、とびこした先でどうなるかは行つてみなければ
わからず。

go to ST

[4] 定数は式中に直接書いても良いが、これでも何回も使用するならば ← を用いて登録用紙上の名前に記入できる。しかし測定結果に基づいて変更したい値は read(A, B, C) などという指令によってデータ用紙から順に読み込むことも出来る。

この入力文が出ると共に、データの値はあくらぬ。

[5] 報告様式の指定は `print( )` によって行うが、
1行の印刷に好しては一つの出力文が対応する。

そのまゝ印字を要求するのは 'A=' と引用符に
つづみ、値の印刷を要求するものは式を直接記入。

[6] いくつかの値(行列等)をまとめ、その何番目の要素として使用するには、あらかじめ角めつに次の

配列 $A_{100 \times 1}$, $C_{10 \times 10}$, $D_{100 \times 1}$, $E_{100 \times 1}$

などに登録用紙上に既列として登録済であることを要求し、

read (N, A, D)

$$A \leftarrow 0 \quad ; \quad S \leftarrow A_i + B_{ij}$$
$$E_{N \times 1} \leftarrow A_{N \times 1} + D_{N \times 1}$$

```
print (BNxN)
```

に自由に利用できる

[7] 誤解のない限り普通に用いられる記号は利用して
良いが、種々なように直して指定する。

$$\frac{A*B}{C+D} \quad \sqrt{A} \quad \sin(A) \quad \ln(F) \quad A^B$$

また値の入れかえは $A \leftrightarrow B$ でよいか $E \leftarrow A$

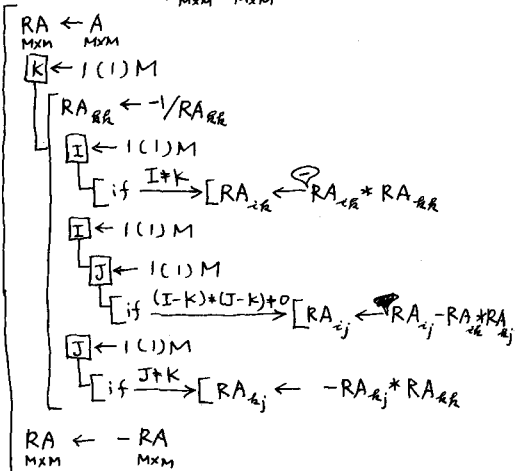
$A \leftarrow B; B \leftarrow E$ という実施になることは注意。また初値は初期値0、累乗は初期値1を入れることは実施の段階(プログラムの変換の時)で要である。

[8] 注を入力した場合は、<START> など<---->
に包むか、引用するときはつて外りにくくするので
右ちに簡潔に書く。また改行は自らは自由であ
るが空白の印字は 'A##B' と井を用いる。

§4. 逆行列を求めよう

A の逆行列を RA に格納する手順

手續 $\text{MINV}(M, A, RA)$



§5. プログラム言語への変換

以上のアルゴリズム記法により、自らの計算手順が正しく記述できるようになった段階で、現用のプログラム言語への変換が行われる。実は高級プログラム言語 ALGOL, PL/I, PASCAL 等は このアルゴリズム記法と大体同じもので、たとえば ALGOL では 文の終りは else の直前を削いてすべてうご入る、登録用紙に記入する名前をすべて begin の後で宣言する、配列エー括演算する $A_{N \times M} \leftarrow 0$ など認める、とりづね正誤修正し、文の角かっこを begin と end に、 $\leftarrow E :=$ に、 $\square \leftarrow 1(1)N$ を for $I:=1$ step 1 until N do に、 $X_i \in X[I]$ に、read を READ に、print を PRINT に (ただし入出力文は場合によって異なる)、手続きを procedure に、関数を答の型に従って real procedure などと変換すれば、そのまま完全なプログラムになっている。(標準関数は SQRT 等) たとえば §2 の立方根のプログラムを ALGOL で書けば 次のようなものになる:

```
begin
  real procedure RE;
  begin
    real X;
    READ(X); RE := X;
  end;
  real procedure CR(X, E);
  begin
    real X, E;
    begin
      real X1, X2;
      X1 := X/3;
      for X2 := (2*X1 + X/X1↑2)/3
        while ABS(X1 - X2) > E do
          X1 := X2;
      CR := X2;
    end;
  end;
  real A;
  for A := RE while A ≠ 0 do
    PRINT(A, CR(A, .04)); CRLF(1);
  end;
end;
45.8, 65, 0;
変換の上で  $X1^2$  は  $X1 \uparrow 2$  にし、仮パラメタは
```

べて規則を行なったあと、印刷実行改行を命ずる CRLF ならびに、実数の計算は / 整数計算 ($5.2 \div 2$ は 2 となる) は $\div$ を用いることをつけておこう。

PL/I PASCAL も ALGOL とほとんど同じであるが、FORTRAN は本質的に GO TO が必要である ^{ここや} 他 ^の 制限事項のために変換も容易でなければ、後でそのプログラムの構造を追うことも容易でない。

§6. おわりに

我々は電子計算を行うに当たって、自らの頭の中にある手順を、より一般化して系統化 関数化したり、より合理的なものにすることに全力を注ぐべきで、枝葉末節の品にもつまない文法制限などの遺物にわづらわされたり、始めからこのような考え方の枠で練り、自由な発想をさまたげたりするやり方は当然避けるべきである。

また自らの組織で利用できる計算機の種類から、止むを得ず FORTRAN によってプログラムするにしても、このアルゴリズム記法で自由に自らの考え方を展開し、しかる後に FORTRAN の文法制限に合わせて修正を行う方法、いわゆるフローチャートのような書きやすいものをもまねくよりも なるかに簡潔で誤りも少ない筈である。

他人に自らの考えを記してプログラムして貰うことは最も面倒なことであるが、このアルゴリズム記法によるメモを添えることが最も正確かつ簡潔であって、実はここまで出来ていけば ALGOL, PL/I, PASCAL といったプログラムにはほとんど使えないものが出来上ってしまったことはおあり頂けよう。

土木関係の各分野でも、このアルゴリズム記法によって電子計算が普及するようになることを心から望みつつ本文を終ることにしたい。

参考文献

1. Dahl, Dijkstra, Hoare, 野下・川倉・武市訳
構造化プログラミング サイエンス社(1975)
2. Wirth, 野下・筑・武市訳: 系統的プログラミング
北沢科学社(1975)
3. 中村廉一: プログラミング入門 北沢出版(1975)
4. スコット・ラング・中村訳: プログラミングのための PL/I 北沢(1979)