

土木研究所報告

第 132 号

越流型余水吐の水理機能に関する研究 (3)

——非標準型越流頂の流量係数について——……………藤本 成…(1)

建設技術に適応した電子計算システムに関する研究 (1)

——NEW ALGOL-H を中心とするシステム——……………中村慶一, 小池長春…(25)



昭和 43 年 5 月

建設省土木研究所

越流型余水吐の水利機能に関する研究(3)

—非標準型越流頂の流量係数について—

(昭和42年10月)

藤 本 成*

要 旨

本文は、越流頂の一般流量公式について考えたものであって、さきに研究(2)で導いた越流頂の流量係数の基本式および標準型越流頂に対する流量公式を考慮し、これに基づく一般公式を提案するとともに、非標準型越流頂に対するこの公式の適用性に検討を加えたものである。

その結果、ここに提案した越流頂の一般流量公式は、円弧および放物線で構成された越流頂に対して十分な適用性を有することが明らかになり、従来それぞれの形状に対して模型実験を行なうことが必要であったこれらの越流頂の流量係数についても、一般公式により取扱うことが可能となった。

目 次

1. 序 論.....(1)	[1] 岩崎の流量公式と著者の提案式の比較.....(14)
2. 従来の研究方向.....(4)	[2] 放物線越流頂の形状.....(16)
3. 越流頂の一般的な流量公式.....(6)	[3] 放物線越流頂の流量係数.....(20)
4. 円弧で構成された越流頂の流量係数.....(9)	6. 結 論.....(23)
5. 放物線越流頂の流量係数.....(14)	

1. 序 論

越流頂の流量係数については、従来より数多くの人々により検討され、各種の公式が提案されてきた。しかしこれらの大部分は個々の形状に対する実験公式であって、越流頂の形状と流量係数の一般的な関係について検討を加え、越流頂の流量係数の一般公式を導いたものはほとんどないのが現状である。これは越流頂の形状そのものが一定した形状でないために、これらを統一して取扱うことが困難なためと考えられる。

しかし、最近のように治水、利水の高度化に伴って河川流量の正確な把握の問題が生じ、各種の越流頂を流量測定用利用することが多くなると、現在の一般性に乏しい実験公式では推定誤差が大きく、再検討を要するものと思われる。

そこで、越流頂の水利機能、すなわち流量係数に及ぼす越流頂形状の影響について、理論ならびに実験の両面より追求し、越流頂の流量係数の一般式について検討することになった。

このような意図のもとに越流型余水吐の水利機能に関する研究として一連の研究を開始し、すでに本研究の成果の一部を研究(1)**および(2)**として発表してきた。

すなわち、研究(1)では、越流頂の流量係数の理論的な取扱いの可能性について追求した。まず堤頂付近の縦断面が放物線であるような越流頂を基本的なモデルとして想定し、この堤頂を越える流れについて従来の2次元ポテンシャル運動理論に基づく考え方にしたがって理論的に解析してみた。

この解析の結果導かれる越流頂の水位—流量関係は、よく知られているように次の2式で表わされる。

$$q = (R+h) \log \left(1 + \frac{h}{R} \right) \\ \times \sqrt{\frac{g(R+h) \log(1+h/R)}{1 + \log(1+h/R)}} \quad \dots\dots\dots (1)$$

および

$$H_0 = \frac{(R+h) \log(1+h/R)}{2[1 + \log(1+h/R)]} + h \quad \dots\dots\dots (2)$$

また、一般に用いられる越流頂の流量公式にしたがっ

* ダム部ダム水利研究室研究員

** 研究(1)—越流型余水吐の水利機能に関する研究(1) 土研報告 第127号 (昭和41年3月)

** 研究(2)—越流型余水吐の水利機能に関する研究(2) 土研報告 第130号 (昭和42年8月)

て、流量係数 C を導入し、これをこの(1)および(2)式より求めれば、次のようになる。

すなわち、

$$2(H_0 - h) \left\{ 1 + \log \left(1 + \frac{h}{R} \right) \right\} \\ = (R + h) \log \left(1 + \frac{h}{R} \right) \dots \dots \dots (3)$$

の条件において

$$C = \frac{q}{H_0^{3/2} g^{1/2}} = \sqrt{2g} \left(1 - \frac{h}{H_0} \right)^{1/2} \\ \times \left(\frac{R + h}{H_0} \right) \log \left(1 + \frac{h}{R} \right) \dots \dots \dots (4)$$

である。ここに、 q ：単位幅当りの放流量（越流量）、 H_0 ：支配断面（ここでは堤頂）を基準面とする全水頭、 R ：堤頂における越流面の曲率半径、 h ：従来の理論における堤頂の水深であり、研究(1)ではこれを実際の水深と区別して h_* と表わしている。また、 g ：重力加速度である。

研究(1)では、さらにこのように理論より導かれる越流頂の流量係数について実験的検証を試みている。

この場合、実験上の模型としては、Randolph によって与えられた標準型越流頂を用いているが、標準型越流頂はすでに知られているように鉛直刃形せきの自由ナップ下側水脈線に一致する形状であり、堤頂上、下流の越流面曲線は堤頂を接点とする2つの関数形で表わされ、堤頂では関数の2次微分が不連続に変化し、結果として堤頂の底面曲率半径は確定した値を持たないため、直接理論の検証をすることができないので、研究(1)では流れはある連続した性質を有する曲面上に沿って流れているものと仮定し、さらに放物線越流頂に対する解析結果がそのまま適用できるものと仮定して、堤頂底面の曲率半径を流速分布より推定し、この推定値を用いることにより実験値と比較する方法を採用している。

まず、Randolph の標準型越流頂に対して、水理的にみた堤頂の越流面の曲率半径 R を推定したところ、 R は流量に関係なく一定した値を示しており、越流頂の設計水頭 (H_d は本来設計水深と定義すべきであるが、本研究では標準型越流頂として接近水頭のはほとんど無視しえる Randolph および Harrold の越流面曲線にのみ限定して取扱っているため、 H_d を設計水頭として用いている。) に対して、次のように表わされることが明らかになった。

$$R = 0.9173 H_d \dots \dots \dots (5)$$

さらに、この堤頂の底面曲率半径を用いて計算される Randolph の標準型越流頂の堤頂における水理量は実験結果と良好な一致をみせ、越流頂の水位-流量関係すなわち流量係数も同様この理論で十分表わし得ることが明らかになった。

研究(2)では、この成果を実用化するために理論式に対して十分な精度が維持できる近似式について考察を加えることにした。

これは、さきの(1)および(2)式あるいは(3)および(4)式に示される理論的関係は、通常われわれの用いる H_0 と q あるいは C の直接の関係を示していないため、実用上不便であり、さらに簡単に計算できることが実用公式として不可欠な条件だからである。

このような意図のもとに、さきの理論式に示される関数形を再検討し、かつ対数関数を数学的に展開近似して、この理論的関係と同等な値ならびに性状を示す実用公式として次式を誘導し提案した。すなわち、

$$C = C_0 \left(1 + K_R \frac{H_0}{R} \right)^{1/2} \dots \dots \dots (6)$$

および

$$\frac{H_0}{h} = \frac{3R + 4K_R H_0}{2R + 3K_R H_0} \dots \dots \dots (7)$$

であり、Randolph あるいは Harrold の標準型越流頂に対しては、(6)式中の R のかわりに H_d を用い次のように表わされる。

$$C = C_0 \left(1 + K_H \frac{H_0}{H_d} \right)^{1/2} \dots \dots \dots (8)$$

ここに、 C_0 ： $H_0=0$ における流量係数の理論値であって、m-sec 単位系においては重力加速度 $g=9.80\text{m/s}^2$ として $C_0=1.704$ 、ft-sec 単位系においては $g=32.17\text{ft/s}^2$ として $C_0=3.086$ である。また、 K_R および K_H はともに係数であって、理論式よりそれぞれ $K_R=0.648$ 、 $K_H=0.704$ と計算された。

一方このように理論式より導かれる(6)あるいは(8)式に示す流量係数はいずれも一様な流れに設置された越流頂に対するものであって、実際に問題とする越流頂は橋脚、橋台のように堤頂の流れの一様性を阻害する構造物を有しており、幅方向に一様な流れとはかなり異なった流量分布を示すものと考えられることから、これらの影響を求め流量係数を補正することも流量推定上重要になってくる。しかし、このような構造物の影響は現段階においては理論的に求めることが困難であり、多くの実験資料を収集して論議することが必要になる。

そこで、研究(2)では、比較的資料の多くかつ比較の容易な標準型越流頂の場合に対して、実際の余水吐水理模型実験資料を収集し、この影響を求めて、次のような標準型越流頂の流量推定法を提案した。

すなわち、橋脚、橋台のある標準型越流頂に対する流量公式として、

$$Q = N S_0 n \cdot L S_0 C' \cdot H_0^{3/2} \dots \dots \dots (9)$$

と表わすものとすれば、流量係数 C' は次のように表わすことができる。

すなわち、

$$\begin{aligned}
 C' &= C(1-M) \\
 C &= C_0 \left(1 + K_H \frac{H_0}{H_d} \right)^{1/2} \\
 M &= M_d \left(\frac{H_0}{H_d} \right)^{1.5} \\
 M_d &= 0.0756 \left(\frac{H_0}{L_s} \right)^{0.50} \\
 &\left. \begin{aligned}
 (n=1 \text{ あるいは } n \geq 2 \text{ かつ } b/S' \geq 0.80 \text{ に対して}) \\
 &= 0.0756 \left(\frac{H_d}{L_s} \right)^{0.50} \left\{ \frac{1}{n} + 1.465 \right. \\
 &\quad \times \left(\frac{n-1}{n} \right) \left(\frac{b}{S'} \right)^{1.7} \Big\} \\
 &\left. (n \geq 2 \text{ かつ } b/S' < 0.80 \text{ に対して}) \right\} \dots (10)
 \end{aligned}
 \right\}
 \end{aligned}$$

ここに、 Q : 余水吐の全越流量、 Ns : 余水吐のブロック数、 n : 1ブロック当りの径間数、 L_s : 1径間当りの越流幅、 H_0 : 堤頂を基準面とする全水頭（越流水頭）、 C' : 橋脚、橋台の影響を補正した流量係数、 C : 一樣流れに対する流量係数、 C_0 : $H_0=0$ における流量係数、 K_H : 係数 $=0.704$ 、 H_d : 越流頂の設計水頭、 M : 越流水頭 H_0 における流量係数の減少率、 M_d : $H_0=H_d$ における流量係数の減少率、 b : 橋脚幅、 S' : 堤頂から橋脚先端までの水平距離 $S+L$ 、 S : ダム上流面から橋脚先端までの水平距離（突き出し長）、 L : 堤頂からダム上流面までの水平距離、であり、Randolphの形状では、 $L=0.287H_d$ 、Harroldでは $L=0.282H_d$ である。

なお、ここに M および M_d を示す実験式は、 $H_0=0$ において橋脚、橋台の影響はまったくなくなるもの（ $H_0=0$ において $M=0$ ）および橋脚の影響は橋台よりも常に小さいものと仮定して導いている。

よって、この(9)および(10)式を用いれば、標準型越流面曲線を採用した越流頂に対しては橋脚、橋台の影響を補正することが可能となり、一応その流量推定法は確立されたものと考えられよう。

しかし、既設の多くの越流頂の流量推定問題を考える場合、これらの越流頂形状が非常に多種多様であることに留意する必要がある。

これは、越流頂の縦断形状を刃形ぎの自由ナップ下側水脈線に一致させると越流面の圧力状態を改良する上に非常に有利であるという考え方は古くからあったけれども、標準型越流頂形状というものはまだ一般的でなく、かつこれに統一することの有利さも当時はほとんどなかったためと考えられる。

いま、さらにこの理由についてその主たるものを挙げてみると、次のような理由をあげることができる。

(1) ダムの越流頂の設計においては、従来より越流

面に負圧を生ぜしめないことがその基準となっており、刃形ぎの自由ナップ下側水脈線を越流頂の基本形状とすることは古くから一般的であったが、実際に設計する場合にはこのような水利条件ばかりでなく、ダムの構造上の安定条件を満たすことが必要であり、この刃形ぎの自由ナップをさらに重厚にして使用することが多い。

また、RandolphあるいはHarroldの標準型越流面曲線はまだあまり重視されておらず、自由ナップの測定結果そのものが座標として与えられており、設計者はこの座標にしたがって自由に曲線をあてはめ使用するのが通例とされていた。たとえば、わが国のダム設計基準*ではDavisの“Handbook of Applied Hydraulics”に記載されている鉛直刃形ぎのナップ形状が座標として与えられており、設計者はこの座標に基づいて越流頂を設計すべきであるとしていることから、この事情を容易に想像できるであろう。

このように、越流頂の標準性の維持についてはあまり重視されておらず、ただ判定の基準にすぎなかったようである。

(2) 越流頂に水門を設置する場合には、自由越流頂において刃形ぎの自由ナップ下側水脈線を用いるのと同様な発想から、越流面に負圧を生ぜしめないように水門半開時の水脈の軌跡を越流面曲線の基本形状とする設計法があり、このような越流頂では、刃形ぎの自由ナップよりはるかにふくらんだ曲線を使用することになる。

この設計法はかなり一般的のものであって、ダムの設計上の基礎的文献である米国開拓局の“Treatise on Dams, Chap. 12, Spillways”によれば、水門を設置した越流頂の場合にはゲート上流の形状は刃形ぎの自由ナップ下側水脈線と一致させ、下流は設計水位でゲート開きを1ftに設置した状態での放水脈の軌跡を基本形状とすることを提唱しており、次の曲線を下流面曲線として使用すべきであると述べている。

$$X^2 = 4hY \dots \dots \dots (11)$$

であって、ここに、 X および Y は堤頂を原点とし、それぞれ水平下流方向および鉛直下向方向にとっており、 h は貯水池の水位を最高水位とし、ゲート開きを1ftにした場合の水脈中心を基準面とする水頭である。この(11)式に示される水脈の軌跡は、初期速度が $V_0 = C_v \sqrt{2gh}$ で水平に水脈が放出されるものとして導かれており、水脈の軌跡 $Y = (X/V_0)^2 g/2$ において速度係数 C_v を $C_v = 1.0$ としたものである。

このような論議は、わが国においては岩崎によつてさ

* ダム設計基準（昭和32年制定）には越流面形状の標準設計のために、刃形ぎの自由ナップの下側水脈の形状を座標で示している。しかし、最近改訂されたダム設計基準（昭和42年改訂版）では、このような表現を改め、越流頂の基本形状としてHarroldの標準型越流頂を記載している。

らに具体化され、水門より上流側の越流面曲線を刃形ぜき自由ナップ下側水脈線と一致させ、堤頂に楕円リップあるいは刃形リップをもった水門を設置し、このナップを測定して実験式にまとめて発表するとともに、水門を設置した越流頂の下流面曲線についてはこの実験式と対応させてその安全性を検討すべきであるとして、その方法を提案している。

これは、越流型余水止を設計する上に非常に重要な事項であり、従来かなり重視されていた点であるが、近年いわれる自由越流に対する標準型越流頂ではゲートの半開操作に対して越流面に極度の負圧が生じないことが多くの水理模型実験の結果明らかにされ、このような設計法は最近除却に採用しない方向にあるが、越流頂の設計上この点に留意する必要があることは言うまでもない。

(3) (2)と同様に水門を設置した越流頂で常に自然越流を望む場合に使用されるドラムゲートを有する越流頂では、ゲートの構造上越流頂は標準型に比して非常に平坦な形に変形される。また、このように、ゲートの構造ならびに機能上の理由から堤頂の形状が著しく平坦になる例は非常に多い。

このように各種の理由により、実際のダムの越流面曲線は刃形ぜきの自由ナップに一致させた形状からはかなり異なった越流頂になることがしばしば生じている。また、著者らの越流頂形状に対する見方からすれば、標準型越流頂は Randolph あるいは Harrold を含めた越流面曲線に限定されることになり、これ以外を非標準型越流頂 (Irregular Overflow Crest)* と呼称することにすれば、既設の大部分の越流頂はこの範囲に入るものと考えられる。

著者らは主として越流頂の水理設計に対する必要性から、標準型越流頂の流出特性に検討を加え、その方法を確立することができた。しかし、既設の越流頂の流量推定問題に直面すると、それらの多くは非標準型であるために標準型越流頂に対する流量公式が直接適用できないものが大部分である。さらにこの多くは水理模型実験によらず経験的な方法によりその流量を推定しており、かつその方法にもかなり問題があるものと考えられる。

そこで、ここではこのような非標準型越流頂に対しても標準型越流頂と同様な流量公式を導くために、標準型越流頂の流量公式をさらに一般的な形に変換し、実際の非標準型越流頂形状と流量係数の関係を明らかにするとともに、各種の越流頂に関する水理模型実験の結果と対

比させながら、その公式の適用性について考えることにする。

2. 従来の研究方向

非標準型越流頂 (1.において定義したように、ここでは Randolph あるいは Harrold によって与えられる標準型越流頂以外のすべての越流頂を取扱う。) の流量係数について一般的な取扱いを試みる場合、越流頂の形状特性の表示方法にまず問題を生ずる。

これは標準型越流頂では設計水頭 H_d によりすべての越流頂の相似が成立するため、この唯一の形状特性量 H_d によって越流頂形状を代表させることが可能であったが、非標準型越流頂の場合には形状が統一されないことが前提となっており、このような越流頂全般に共通した特性量がないのが普通だからである。

したがって、非標準型越流頂の流量係数について一般的な議論を進めるためには、かなりの実験資料を必要とするため、統一的な流量公式に関する研究は非常に数少ないのが現状である。しかし、非標準型越流頂に対する一般的な流量公式の必要性は以前よりあり、いくつかの試みが発表されているが、実際の非標準型越流頂の流量推定法に関する実用的な研究としては、J. N. Bradley (1952)⁽¹⁾、B. Buehler (1954)⁽²⁾、K. W. Kirkpatrick (1955)⁽³⁾ および現在わが国で唯一の実用公式として使用されている岩崎 (1957)⁽⁴⁾ の研究をあげることができる。

そこで、まずこれらの研究が非標準型越流頂の形状の多様性をどのように一般化して取扱っているかについて検討してみることにする。

まず、J. N. Bradley は米国開拓局における越流頂の水理模型実験結果を収集して、次のような非標準型越流頂に対する流量係数推定法を提案している。

すなわち、その推定方法は次のとおりである。

(1) 数多くの越流頂に関する実験資料を収集し、これらの越流頂形状を同一の縮尺 (Standard Scale) に無次元化して描く。

(2) 問題の越流頂を同一縮尺に描き、上、下流面を同時に満たす (類似する) 形状を選び、この類似した越流頂の実験資料より設計水頭時の流量係数 C_d を得る。

(3) 設計水頭時の C_d が得られたならば、実験資料より求められた図-1に示す $C/C_d \sim H_0/H_d$ 曲線より任意の H_0 に対する流量係数を得る。

* 同様な研究を行なった J. N. Bradley は、越流頂を datum shape と irregular shape とに分類し、datum shape としては自由ナップに一致あるいは類似した越流頂を示し、非常に広範囲の越流頂形状を含んでおり、irregular shape はこれ以外の越流頂の意味に使用している。また、K. W. Kirkpatrick は Standard Crest と Irregular Spillway Crest とに分類し、Standard Crest としては越流頂が刃形ぜきの自由ナップに類似した形状すべてを包摂して取扱ひ、Irregular Spillway Crest としては広頂を含めた堤頂を取扱っている。

また、ダム水理研究室の従来からの分類法では Randolph あるいは Harrold の越流頂を標準型と呼び、これに類似した越流頂 (主として放物形越流頂) を準標準型とし、これ以外を非標準型と呼んでいる。

このように、非標準型あるいは "Irregular" という呼称は研究者により異なっている。

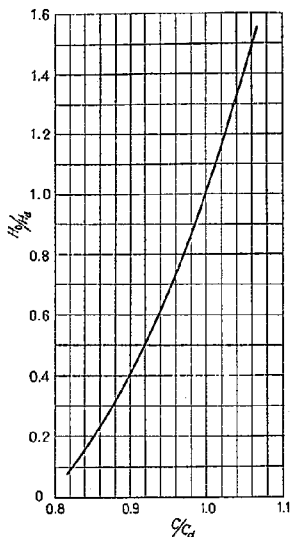


図-1 Bradley の流量推定法における $C/C_d \sim H_0/H_d$ 曲線

これが、Bradley の提案した越流頂の流量係数推定法であり、高ダムの越流頂およびアースダム等の低ダムの越流頂の2種について非常に多くの実験資料を収集しており、かつ越流頂の多様性をそのまま推定法にまとめている点。非常に素朴な方法ではあるが、かなり確実な方法でありかつ実用上有利な方法と考えられる。しかし、この方法で用いられる越流頂の設計水頭については不明確な点があることおよび類似の越流頂の選定に主観の入りやすいことなどさらに検討する必要があるであろう。

B. Bueler は、非標準型越流頂でも設計水頭を決定すれば標準型越流頂と同様な取扱いが可能であるとの前提

のもとに、非標準型越流頂の設計水頭決定方法として次のような具体的な方法を提案している。

すなわち、図-2に示すように同一縮尺で各設計水頭に対する標準型越流頂を描き、これに問題とする越流頂を描くことにより、最もよく合致する設計水頭を選定する方法であり、合致する標準型の設計水頭が堤頂上、下流で異なる場合はその大なる方を選定すべきであるとしている。このようにして、非標準型越流頂においても設計水頭が決定すれば、その流量係数は標準型越流頂の流量公式を用いて推定することが可能であり、B. Bueler は Brudnell の導いた次式を採用している。

$$C = 3.97 \left(\frac{H_0}{H_d} \right)^{0.13} \quad (\text{ft-sec 単位}) \dots\dots\dots (12)$$

B. Bueler のこの非標準型越流頂であっても、設計水頭を適切な方法により選定すれば、標準型越流頂と同様な取扱いが可能であるとする考え方は、さきの Bradley における $C/C_d \sim H_0/H_d$ 曲線は越流頂の形状に左右されないとするのと同様なものであって、非標準型越流頂でも標準型越流頂に対して定義されると同様な設計水頭が存在することを仮定したものであり、いかなる越流頂でも流量係数を支配する唯一の代表寸法が存在することを仮定したものである。しかし、非標準型越流頂において求められる設計水頭のもつ物理的意義は標準型越流頂とはかなり異なり、現実には仮想設計水頭にすぎないことに注意する必要がある。

この B. Bueler の考え方は K. W. Kirkpatrick の検討を得て、後述の岩崎の方法にさらに具体化されることになる。

K. W. Kirkpatrick は、堤頂より上流側の曲面形にのみ着目して、これを標準型越流頂形状と合わせることであり非標準型越流頂の設計水頭を推定することを提案するとともに、このようにして推定された設計水頭時の流量係数は越流頂下流面曲線のふくらみによりその値が変化することを TVA の 9 ケのダムの水理模型実験資料を

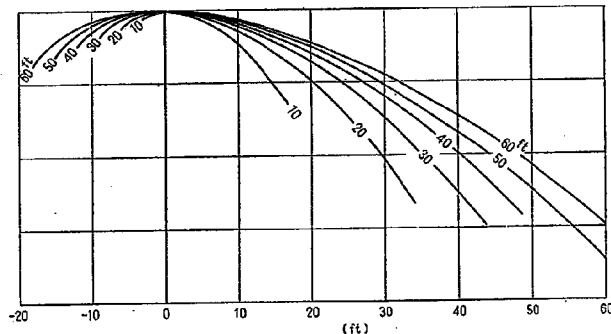


図-2 種の H_d 値に対する標準型越流頂形状

整理して明らかにした。

この場合、堤頂下流面の曲線のふくらみの定量的な表示法として堤頂上流の曲線形より求められた設計水頭 H_d を越流頂の基本寸法とし、この値を用いて無次元化された堤頂下流面の $X=0.5H_d$ における曲線の無次元縦座標 Y/H_d をこのふくらみを指示する値としており、この値が小さいほど流量係数も小さくなることを指摘している。

この Kirkpatrick の提案は、非標準型越流頂においても標準型と同様な設計水頭概念を導入するという不明確な点を有しながら、曲線のふくらみの流量係数に及ぼす影響をかなり具体的に明示した点非常に画期的なものと考えられよう。

岩崎はこの Kirkpatrick の提案をさらに具体的に数式化し、実用公式として次のようにまとめている。

すなわち、岩崎は標準型越流頂 (Harrold 型) においては、堤頂と上流鉛直面との水平距離 W は設計水頭 H_d に対して $W=0.282H_d$ と表わされることを利用し、非標準型越流頂でも Kirkpatrick の提案にしたがひ、この設計水頭 H_d が堤頂と上流鉛直面との距離 W より $H_d=W/0.282$ として計算することを提案するとともに、さきの $X/H_d=0.5$ における無次元縦座標 $(Y/H_d)_0$ と $H_0=H_d$ における C_d との関係式を Kirkpatrick の TVA の資料および土木試験所の竹内の実験資料より次のように導いている。

すなわち、

$$\left. \begin{aligned} C_d &= 1.971 + 0.498\xi + 6.63\xi^2 \\ C &= 1.60 \frac{1+A(H_0/H_d)}{1+0.5A(H_0/H_d)} \end{aligned} \right\} \dots\dots\dots (13)$$

ここに $\xi=(Y/H_d)_0$ であり、係数 A は $H_0=H_d$ における $C=C_d$ を与えることにより次式によって計算される。

$$A = \frac{C_d - 1.60}{1.60 - 0.5C_d} \dots\dots\dots (14)$$

この岩崎の提案した非標準型越流頂の流量係数を求める実用的方法は、現在わが国における唯一の方法として水理公式集にも採用され広く利用されているものであり、越流頂と流量係数の関係すなわち越流頂形状の流量係数に及ぼす影響を平明に表わしている点実用上非常に有利なものと考えられる。しかし、堤頂に水平部を有する場合の設計水頭の計算法あるいは(13)式により求められる C_d の値の性格等明確でない点がいくつかある。

すなわち、堤頂に水平部分のある場合岩崎はこれを堤頂下流面の一部として取扱っているが、実際上このような取扱いは行なうと、 $X/H_d=0.5$ における $(Y/H_d)_0$ の値が堤頂下流面のふくらみの具体的な表示とは考えられなくなることおよび(13)式より求められる C_d に対する橋脚、橋台の影響の有無が明確でないことなど、実際の使用上混乱をきたしている点である。

このように非標準型越流頂の流量係数に対する従来の研究成果をまとめてみると、非標準型であっても標準型と同様な意義を有する設計水頭を何んらかの方法により算定し、問題の非標準型越流頂に対応する標準型越流頂を求めて、その流量公式を準用しようとするものが大部分をしめており、これに改良を加えて越流頂の形状の流量係数に及ぼす影響を求めたものには、Kirkpatrick の提案あるいは岩崎の実験式があるにすぎない。しかし、それらもまだ実用上かなり問題が生じている点を考えると、提案の域をでないものと考えられ、さらに多くの実験資料を用いた検討が必要のようである。

3. 越流頂の一般的な流量公式

2.において述べたように、従来の研究では非標準型越流頂の形状の複雑さのために越流頂形状と流量係数との関係を実験的にもまとめることが困難となり、非標準型越流頂と標準型越流頂との形状の類似点を利用し、標準型越流頂の流量公式を準用しようとするものが大部分であった。

これは、標準型越流頂の実験資料が比較的多く、この資料より導かれる標準型越流頂の流量公式の信頼度はかなり高いものと予想されることから当然な方向と考えられよう。

しかし、非標準型越流頂の形状の特異性が流出特性に及ぼす影響は標準型越流頂におけるそれとは当然異なるものと考えられ、従来においても一部試みられているように、非標準型の影響の補正がその流出特性を取扱う場合最も重要になってくるものと考えられる。

ここでは、このような点を重視して、非標準型越流頂を含む越流頂の一般的な流量公式について、標準型越流頂における研究成果を考慮して、若干の考察を加えてみる。

まず、標準型越流頂の流量公式は、すでに述べたように、(9)および(10)式において表わすことができたが、この公式においては標準型越流頂の形状特性は、設計水頭 H_d という唯一の形状寸法に代表されている。

これは、標準型越流頂の形状がその大小にかかわらず常に相似であり、かつこのように相互に相似な形状に対する流量係数は、フルードの相似則が成立する範囲においては、形状と同様な相似性が成立するため、ある1つの標準型越流頂に対して導かれた流出特性は、他のあらゆる標準型越流頂に対して同様適用することが可能になることから考えても、標準型越流頂の流量公式が形状の基本寸法である設計水頭 H_d の関数として表わされることは当然の結果と考えられよう。

一方、越流頂の流出特性について理論的に検討を行なったところ、研究(1)および(2)において述べたよう

に、その特性すなわち越流頂の流量係数は、堤頂における底面の境界条件に支配され、堤頂部における底面の曲率半径 R のみの関数として (1) および (2) 式あるいは (3) および (4) 式に示す関数により表わされることが明らかになった。

このことから考えれば、標準型越流頂の流量係数表示式における唯一の形状特性量すなわち設計水頭 H_d は、堤頂における底面曲率半径 R と同等の意義を有することになる。

これは、研究 (1) において実験的に明らかにしたように標準型越流頂の堤頂における水理的にみた底面曲率半径 R は、流量に関係なく一定しており、かつ設計水頭との比率が常に一定していることから明らかである。

よって、標準型越流頂の流量公式に含まれる堤頂形状の基本量 H_d は、越流頂の一般的な流量公式の立場からみれば、堤頂における底面曲率半径と同等の意義をもつ値と見なすことができる。

これは、越流頂の中で標準型越流頂が一つの特殊な形状と見なせることであり、この意味から越流頂の一般的な流量公式は標準型越流頂に対しても適用できるものであることが必要になる。

このような点を考慮して、さきに導いた (10) 式に示す標準型越流頂に対する流量公式を一般的な越流頂の流量公式に拡張するために、研究 (1) で実験的に求めた堤頂の底面曲率半径 R と設計水頭 H_d の (5) 式の関係を用いて、流量公式を R の関数として導いてみる。

すなわち、標準型越流頂の流量係数は R の関数として次のように導かれる。

$$\left. \begin{aligned} C &= C \left[1 - M_r \left(\frac{H_0}{R} \right)^{1.5} \right] \\ C &= 1.704 \left(1 + 0.648 \frac{H_0}{R} \right)^{1/2} \\ M_r &= 0.0693 \left(\frac{R}{L_s} \right)^{0.50} \\ (n=1 \text{ あるいは } n \geq 2 \text{ かつ } b/S' \geq 0.80 \text{ に対して}) \\ &= 0.0693 \left(\frac{R}{L_s} \right)^{0.50} \left\{ \frac{1}{n} + 1.465 \right. \\ &\quad \times \left. \left(\frac{n-1}{n} \right) \left(\frac{b}{S'} \right)^{1.7} \right\} \\ (n \geq 2 \text{ かつ } b/S' < 0.80 \text{ に対して}) \end{aligned} \right\} \dots (15)$$

ここに、 R は、越流頂形状が、堤頂を原点とし、水平下流方向に X 軸、鉛直下向き方向に Y 軸をとり、 $Y=F(X)$ と表わされるものとすれば、

$$R = \left\{ 1 + \left(\frac{dY}{dX} \right)^2 \right\}_{X=Y=0}^{3/2} / \left(\frac{d^2Y}{dX^2} \right)_{X=Y=0} \dots (16)$$

である。

この (15) 式に示す越流頂の流量公式は、越流頂形状の中で非常に特殊な例と見なせる標準型越流頂の流量公式

を一方的に拡張して、公式に一般性を持たせたものであるけれども、標準型越流頂という比較的实验資料の多いすなわち流量公式としての信頼度の高いものより導かれている点、越流頂の一般的な流量公式として十分耐えうる公式になるものと思われる。

なお、標準型越流頂に対しては、この変換を逆行なうことにより、容易にさきの (10) 式を導くことが可能であり、越流頂の流量公式としてはこの (15) 式を用いれば十分であろう。

この越流頂の一般流量公式の適用性については、後に実験資料と対比させ検討することにして、この式に含まれる R の性状について、標準型越流頂に対する検討結果より整理してみる。

すなわち、この越流頂の一般的な流量公式を曲面で構成された堤頂すなわちあらゆる越流頂に適用できるものとすれば、この堤頂底面の曲率半径と定義づけられる R の性状についてさらに明確にすることが必要だからである。

これは、堤頂曲線の関数形より計算される堤頂上の実際の曲率半径は、この関数形に左右され種種のケースが生じるから、この各々のケースについて (15) 式に用いる流出特性を示す R との関係を明らかにし、かつこの R の選定基準を明確にしておくことが必要になるからである。

標準型越流頂の場合には、堤頂底面曲率半径が堤頂上流側 ($X \rightarrow -0$) に対して $R_u = 0.5 H_d$ 、堤頂下流側 ($X \rightarrow +0$) に対して $R_d = 0$ と計算され、実験により定めた流出特性を支配する曲率半径は、 $R = 0.9173 H_d$ となり、實際上同一堤頂に対しても各々の R が存在するのが普通だからである。

そこで、まずこのような観点から堤頂を分類してみると、越流頂は一般に次の 5 ケースに分類できる。すなわち、堤頂上流側の関数形より $X \rightarrow -0$ として求められる堤頂底面曲率半径を R_u 、同様に $X \rightarrow +0$ として求められる堤頂底面曲率半径を R_d としてその分類を示せば次のようである。

- | | |
|-------|--|
| 第 1 種 | $R_u = C_1, R_d = C_2 \quad (C_1 = C_2)$ |
| 第 2 種 | $R_u = C_1, R_d = C_2 \quad (C_1 \neq C_2)$ |
| 第 3 種 | $R_u = \infty \text{ or } 0, R_d = C_2$ |
| 第 4 種 | $R_u = C_1, R_d = \infty \text{ or } 0$ |
| 第 5 種 | $R_u = \infty \text{ or } 0, R_d = \infty \text{ or } 0$ |

ここに、 C_i は有意な値を示している。

この 5 ケースについて、(15) 式に示す流量公式に用いる堤頂底面曲率半径 R の選定基準について考えてみる。

この場合、まず同様な問題が生じ、実際に実験により (15) 式の R を推定した Randolph 型越流頂を例にとり再検討してみる。

Randolph 型越流頂では、堤頂は堤頂上流側で $(Y - 0.5H_d)^2 + X^2 = (0.5H_d)^2$ 、堤頂下流側で $Y = 0.53X^{1.8}/H_d^{0.8}$ と表わされるから、 $X \rightarrow 0$ で $R_d = 0.5H_d$ 、 $X \rightarrow +0$ で $R_d = 0$ と計算され、上記の分類では第4種に相当する。この Randolph 型越流頂を越える流れを実際に測定し、流速分布を理論式と対比させて流出特性を支配する堤頂の曲率半径を求めたところ、この値は $R = 0.9173H_d$ と求められた。よって、この Randolph 型堤頂では、 $R_u = 0.5H_d$ 、 $R_d = 0$ 、 $R = 0.9173H_d$ となったわけである。

いま、この $R = 0.9173H_d$ を実際に数値的にも与え、かつ Randolph 型越流頂下流面曲線の指数1.8に近い曲線として放物線を考えるものとして、その係数を求めたところ、Randolph 型越流頂に相当する放物線越流頂として $Y = 0.545X^2/H_d$ を得た。

筆者は、この Randolph 型越流頂に対する R 値の推定結果より、越流頂の流量係数に支配的な影響を及ぼす越流頂の形状効果を次のようにまとめた。

すなわち、

(1) 越流頂の流量係数に支配的な影響を及ぼすのは、堤頂下流面の形状であり、この影響は $X \rightarrow +0$ より求められる堤頂底面曲率半径により代表できる。

(2) 数値的に堤頂において曲率半径が確定しなくても、堤頂 ($X=0$) において曲率半径が有意な値を示し、かつ問題の越流頂下流面曲線に類似の曲線形を用いれば、(15)式の流量公式に用いる堤頂底面曲率半径を推定することは可能である。

このように、非常に大胆な類推であり仮定ではあるけれども、Randolph 型越流頂の流量係数に関する実験結果を用いれば、上記(1)および(2)のような堤頂底面曲率半径 R の選定基準を導くことができる。

これは、さきの越流頂形状の分類において、第1、2および3種については、流量公式に用いる R として $R = R_d = C_2$ を用いれば十分であることを示しており、第4および5種については、問題の越流頂の堤頂付近の形状を何らかの手段を用いて堤頂で曲率半径が有意な値を示す曲線形に変換し、その曲線形より R を求めることが必要になるわけである。

しかし、実際にはこのような第4および5種の越流頂に対して、上述(2)で示すような R の算定を目的とする曲線の当てはめ問題は非常に困難なようである。これは、曲線の当てはめ計算自体は、最小2乗法等の手段を用いることにより容易に行なえるが、問題の越流頂に対する曲線の当てはめ範囲がさらに問題であって、 R の値そのものがこの範囲に依りてかなり敏感に変動する性質のものであり、実際には安定した値とは言いがたいからである。すなわち、問題の越流頂の $X \rightarrow +0$ における曲

率半径の妥当な推定値は、必ずしも流量公式における R とは一致しないようであり、 R は越流頂下流面のかなりの範囲に対して推定すべき値であると考えられるからである。

たとえば、越流頂下流面曲線が $Y_1 = MX^N (N \neq 2)$ と表わされる一般的な越流頂に放物線 $Y_2 = AX^2$ を当てはめ、係数 A より $R = 1/2A$ として R を推定することを考えてみる。

この $Y_1 = MX^N$ を $0 \sim X_1$ の範囲で $Y_2 = AX^2$ に近似せしめるために、ここでは最小2乗法の考え方にしたがってこの範囲の誤差 ϵ の2乗の総和 E を最小ならしめるように係数 A を定めよう。

すなわち

$$\begin{aligned} E &= \int_0^{X_1} \epsilon^2 dX = \int_0^{X_1} (Y_1 - Y_2)^2 dX \\ &= \frac{M^2}{2N+1} X_1^{2N+1} - \frac{2MA}{N+3} X_1^{3+N} \\ &\quad + \frac{A^2}{5} X_1^5 \dots\dots\dots (17) \end{aligned}$$

より、

$$\frac{\partial E}{\partial A} = \frac{2A}{5} X_1^5 - \frac{2M}{N+3} X_1^{3+N} = 0$$

となるから、

$2A = 1/R$ は次のように求められる。

$$2A = \frac{10M}{N+3} X_1^{N-2}, \quad R = \frac{N+3}{10M} X_1^{2-N} \dots\dots\dots (18)$$

この(18)式において、長さの尺度を一般的に取扱うために

$$\left. \frac{dY_1}{dX} \right|_{X=X_1} = \tan \theta_1 = MN X_1^{N-1}$$

と表わされる $\tan \theta_1$ を用いて、 X_1 を消去すれば、 R は $\tan \theta_1$ の関数として次のように表わされる。

$$R = \frac{N+3}{10M} \left(\frac{\tan \theta_1}{MN} \right)^{\frac{2-N}{N-1}} \dots\dots\dots (19)$$

よって、(19)式において、近似範囲を示す無次元量 $\tan \theta_1$ を与えれば、任意の指数関数 $Y_1 = MX^N$ に対して、放物線近似を前提として、堤頂の曲率半径を計算することは容易である。

しかし、この $\tan \theta_1$ を選定するに際しては、次のような問題が生ずる。

すなわち、いま実験的に R 値を求めた Randolph 型越流頂に対して(19)式を適用し、この式より R を求めようとする場合、 $M = 0.53$ 、 $N = 1.8$ 、 $R = 0.9173$ として逆算すれば明らかなように、この $\tan \theta_1$ として1.0程度の値を考えることが必要になる。これは、 X_1 として無次元座標系において1.0程度すなわち実際には $X_1 \approx H_d$ の範囲を考えていることに相当し、近似範囲としてかなりの区間を要することがわかる。しかし、 N が2に近づ

けば、(19)式の右边は次第に $\tan \theta_1$ によって急激な変化を示さなくなり容易に安定した値に近づくこと、および、 $N=2$ となれば R は $\tan \theta_1$ に無関係に決定されることを考慮すれば、この $\tan \theta_1$ をある一定した値で与えることは適当でないように考えられる。

このように、(19)式において用いる $\tan \theta_1$ を問題の指数関数の指数 N に無関係に与えることは適切でなく、実際には $\tan \theta_1$ の選定基準として、 $\tan \theta_1$ が N の関数となるように与えることが必要であろう。

これらについては、多くの実験資料を用いて論議することが必要であるが、現在のところこのような検討を試みるに十分な実験資料はまったくないので、ここでは R の計算式として (19) 式を示すにとどめ、この $\tan \theta_1$ の選定方法については後日さらに検討を加えることにする。

よって、この研究(3)においては、さきの越流頂の堤頂形状による分類のうち、第4および5種の越流頂に対する一般的取り扱いが省き、第1、2および3種の堤頂の底面曲率半径の既知な越流頂のみを取りあげ、これらの越流頂に対するさきの(15)式の適用性について、実験資料と推定曲線とを比較し、検討を加えることにする。

4. 円弧で構成された越流頂の流量係数

非標準型越流頂として、最も単純な形状すなわち堤頂付近の形状が単一あるいは複合円弧で表わされた越流頂の流量係数について、まず考えてみる。

この場合、堤頂が単一の円弧で構成された越流頂は非常に数少なく実際のではないが、図-3に示すような堤頂の上下流面がいずれも円弧で表わされる複合円弧で構成された越流頂、あるいは堤頂にドラムゲートを設置した越流頂でこのゲートが倒伏位置 (Gate Down) にある場合などはいずれもこの範囲に相当する。

なお、図-3に示す複合円弧で構成された越流頂は、Creager⁽⁵⁾が鉛直刈形ぜき下側水脈線に類似の越流頂として導いたものであり、著者らの標準型越流頂と同意義を有する形状であるが、この形状では堤頂の曲率半径が $R=0.9167H_d$ となっており、さきに著者らの導いた Randolph 型越流頂に対する R の推定値 $R=0.9173H_d$ とほぼ完全に一致している点、非常に興味あることだと思われる。

ここでは、これら堤頂付近の形状が単一あるいは複合円弧で表わされ

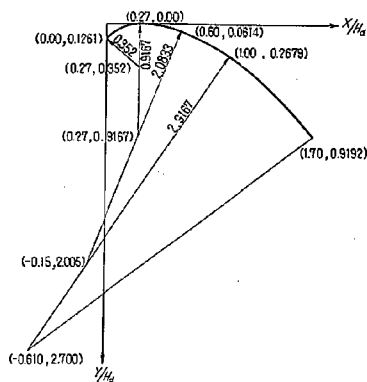


図-3 複合円弧で構成された Creager の標準型越流頂

る越流頂に対する越流頂の一般流量公式の適用性について、実験資料と比較検討することにする。

なお、複合円弧で構成された越流頂の実験資料あるいはドラムゲートに関する実験資料は非常に数少なく、筆者もこの実験経験を有しないが、ここでは J. N. Bradley により集録された実験資料⁽⁶⁾を用いて検討することにする。

Bradley により集録されたドラムゲートは表-1に示すとおりであり、その形状は図-4に示すとおりである。

なお、表中、Grand Coulee, Bhakra, Shasta および Hamilton の各ダムドラムゲートの半径は、Bradley の発表した値とは異なり、同論文の付図およびさきに紹介した Bradley が非標準型越流頂の流量係数を取扱って米国開拓局より発表した資料⁽¹⁾をもとに、堤頂付近のみを取り扱うことにして、その値を修正している。

これらドラムゲートの倒伏位置における流量係数の実験結果は、さきの(15)式による計算結果とともに図-5お

表-1 Bradley により集録されたドラムゲートの形状

No.	ダ ム 名	ゲート数	ゲート長(幅)(ft)	ゲート半径(ft)	接近水深(ft)	最大水頭高さ(ft)	模 型 尺 寸
1	Grand Coulee	11	135	23.00	360	31.65	1:30
2	Bhakra	2	135	23.00	410	28	1:80
3	Shasta	3	110	23.00	460	28	1:88
4	Hamilton	1	300	40.00	50	32	1:30
5	Hoover(4-M3)	4	100	26.80	50	26.6	1:20
6	Hoover(8-M5)	4	100	36.00	50	26.6	1:20
7	Hoover(7-C4)	4	100	26.00	50	26.6	1:60
8	Friant	3	100	20.50	140	19.0	1:25
9	Norris	3	100	34.00	200	27.0	1:72
10	Madden	4	100	30.00	120	30.0	1:72
11	Capilano	1	70	24.58	200	23.0	1:60

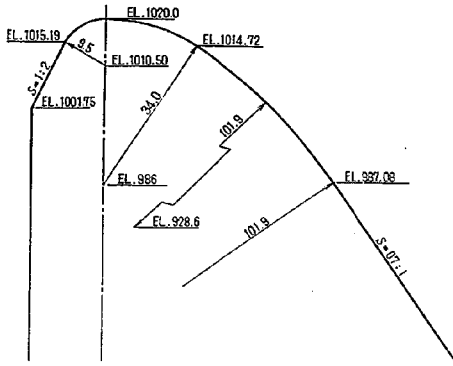


図-4-7 Norris ダムの越流頂

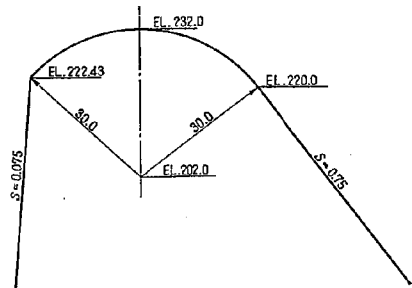


図-4-8 Madden ダムの越流頂

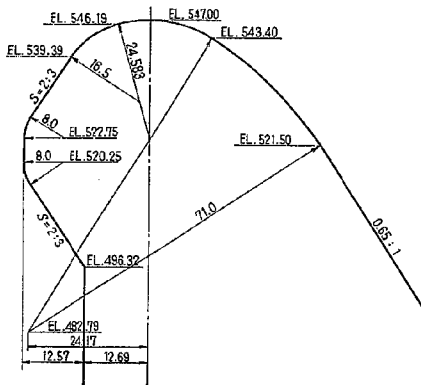


図-4-9 Capilano ダムの越流頂

よび表-2に示される。

なお、これらの図表における流量係数の推定計算は、橋脚幅 b および橋脚先端位置を示すファクター S' の資料がないため、いづれも $b/S' \geq 0.80$ の範囲に入り、その影響がないものと仮定して、次式により計算している。

$$\left. \begin{aligned} C' &= C \left\{ 1 - M_r \left(\frac{H_0}{R} \right)^{1.5} \right\} \\ C &= 3.086 \left(1 + 0.648 \frac{H_0}{R} \right)^{1/2} \\ M_r &= 0.0693 \left(\frac{R}{L_s} \right)^{0.50} \end{aligned} \right\} \dots (20)$$

(ft-sec 単位)

ここに、 L_s は純越流長であり、ここでは表-1のゲート長さが相当し、 R は堤頂の越流底面の曲率半径であり、ドラムゲート面の曲率半径である。

いま図-5、表-2に示されるドラムゲートの倒伏位置における流量係数に対する(20)式の適用結果を検討してみ

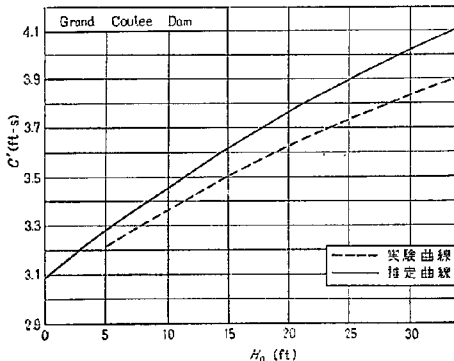


図-5-1 Grand Coulee ダムにおける実験曲線と推定曲線

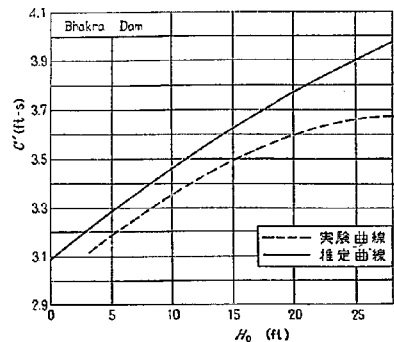


図-5-2 Bhakra ダムにおける実験曲線と推定曲線

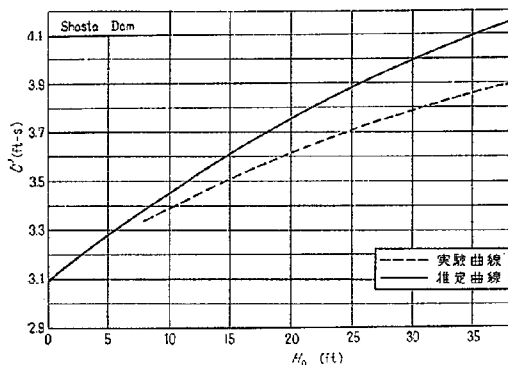


図-5-3 Shasta ダムにおける実験曲線と推定曲線

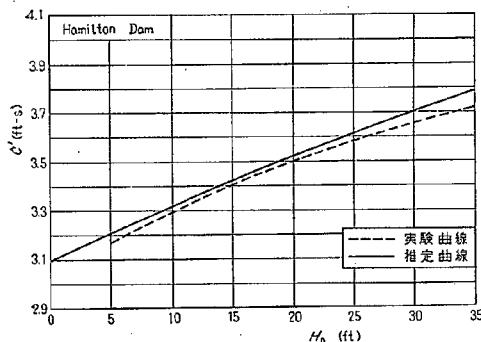


図-5-4 Hamilton ダムにおける実験曲線と推定曲線

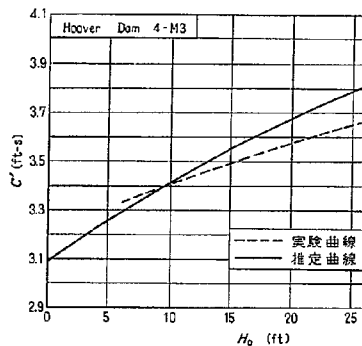


図-5-5 Hoover ダム (4-M3) における実験曲線と推定曲線

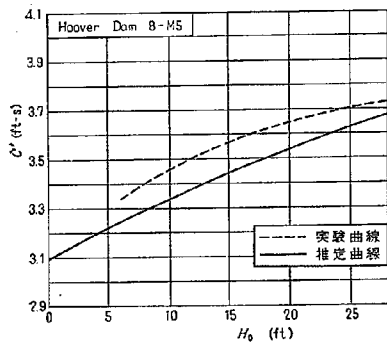


図-5-6 Hoover ダム (8-M5) における実験曲線と推定曲線

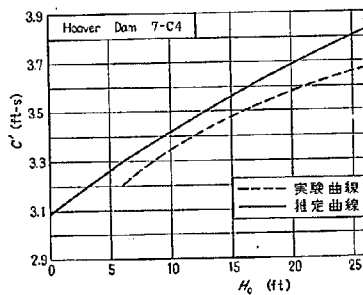


図-5-7 Hoover ダム (7-C4) における実験曲線と推定曲線

ると、Norris ダムおよび Hoover ダムモデル 8-M5 の場合を除き、いずれも推定結果の方が実験値より大きい値を示している。しかし、Madden ダムあるいは Hamilton ダムのように比較的に良い一致を示す例もあり、全体的には流量係数の性状をかなり適確に表わしているものと考えられる。

しかし、 H_0 の増大にともない、推定結果と実験結果との差が単調に増大する例の多いことを考えれば、流量係数の基本値に対する補正項が(20)式に示す M_r では不十分であることが予想される。

この点については、これらドラムゲートの実験条件、とくに接近水路の条件について、詳細な検討が必要であるとともに、越流底面

表-2 ドラムゲートの流量係数の実験値と推定値

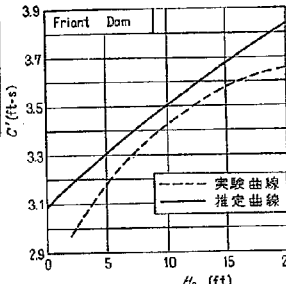
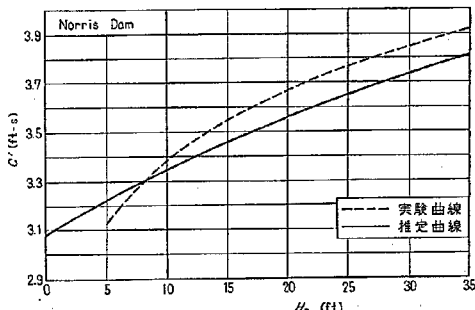
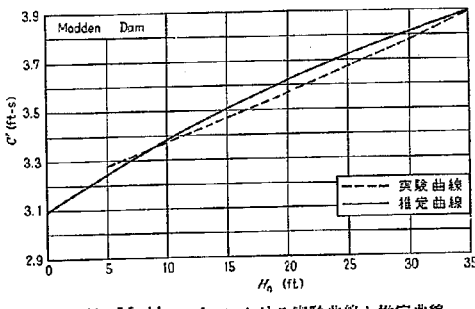
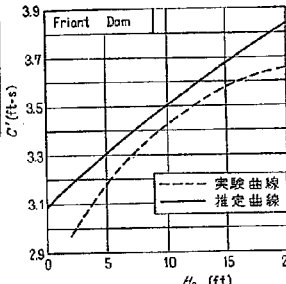
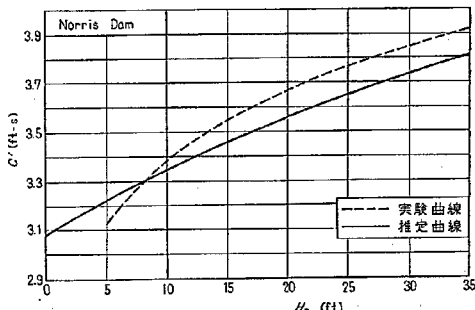
H_0 (ft)	C' (ft-s 単位)			H_0 (ft)	C' (ft-s 単位)			H_0 (ft)	C' (ft-s 単位)						
	実験値	推定値	誤差(%)		実験値	推定値	誤差(%)		実験値	推定値	誤差(%)				
1. Grand Coulee Dam															
5	3.220	3.286	2.05	5	3.125	3.222	3.10	8	3.250	3.370	3.69				
10	3.352	3.465	3.37	10	3.390	3.345	-1.33	13	3.415	3.520	3.07				
15	3.510	3.625	3.28	15	3.550	3.457	-2.62	18	3.530	3.652	3.46				
20	3.635	3.769	3.69	20	3.670	3.560	-3.00	23	3.625	3.768	3.94				
25	3.745	3.898	4.09	25	3.765	3.654	-2.95	28	3.705	3.867	4.37				
30	3.842	4.013	4.45	30	3.845	3.739	-2.76	33	3.775	3.953	4.72				
35	3.920	4.116	5.00	35	3.915	3.816	-2.53								
2. Bhakra Dam															
3	3.120	3.209	2.85	10. Madden Dam											
8	3.275	3.396	3.69	5	3.280	3.240	-1.22								
13	3.420	3.563	4.18	10	3.365	3.378	0.37								
18	3.550	3.713	4.59	15	3.460	3.503	1.24								
23	3.645	3.848	5.57	20	3.560	3.617	1.60								
28	3.680	3.969	7.85	30	3.770	3.811	1.09								
				35	3.900	3.893	-0.18								
3. Shasta Dam															
8	3.335	3.394	1.77	11. Capilano Dam											
13	3.465	3.558	2.68	8	3.250	3.370	3.69								
18	3.575	3.706	3.66	10	3.365	3.378	0.37								
23	3.675	3.836	4.38	15	3.460	3.503	1.24								
28	3.760	3.952	5.11	20	3.560	3.617	1.60								
33	3.835	4.053	5.68	30	3.770	3.811	1.09								
38	3.895	4.141	6.32	35	3.900	3.893	-0.18								
4. Hamilton Dam															
5	3.160	3.205	1.42	5. Hoover Dam 4-M3											
10	3.290	3.316	0.79	6	3.338	3.290	-1.44								
15	3.400	3.421	0.62	10	3.405	3.411	0.18								
20	3.500	3.519	0.54	14	3.472	3.522	1.44								
25	3.580	3.612	0.89	18	3.540	3.624	2.37								
30	3.645	3.700	1.51	22	3.605	3.718	3.13								
35	3.710	3.783	1.97	26	3.670	3.803	3.62								
6. Hoover Dam 8-M5															
5	3.335	3.302	-0.99	7. Hoover Dam 7-C4											
10	3.460	3.331	-3.73	6	3.200	3.296	3.00								
15	3.565	3.439	-3.53	10	3.360	3.420	1.79								
20	3.650	3.537	-3.09	14	3.450	3.534	2.43								
25	3.705	3.626	-2.13	18	3.540	3.638	2.77								
28	3.735	3.677	-1.55	22	3.615	3.734	3.29								
				26	3.665	3.822	4.28								
8. Friant Dam															
2	2.965	3.179	7.22	9. Norris Dam											
5	3.175	3.308	4.19	5	3.125	3.222	3.10								
8	3.340	3.428	2.63	10	3.390	3.345	-1.33								
11	3.460	3.537	2.22	15	3.550	3.457	-2.62								
14	3.550	3.641	2.56	20	3.670	3.560	-3.00								
17	3.625	3.735	3.03	25	3.765	3.654	-2.95								
20	3.650	3.823	4.74	30	3.845	3.739	-2.76								
				35	3.915	3.816	-2.53								

図-5-8 Friant ダムにおける実験曲線と推定曲線

図-5-9 Norris ダムにおける実験曲線と推定曲線

図-5-10 Madden ダムにおける実験曲線と推定曲線

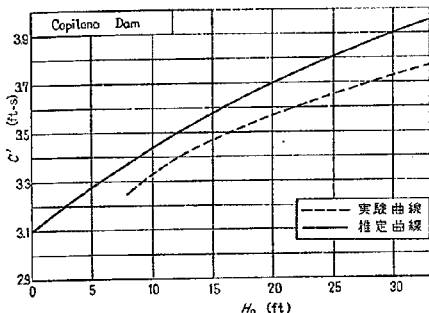


図-5-11 Capilano ダムにおける実験曲線と推定曲線

に流れが沿うことを前提とする(20)式の H_0/R に対する適用範囲をさらに明確にする必要があるものと考えられる。

しかし、従来まったく実験的にのみ論じられたこれらの越流頂の流出特性についても、著者らの提案した流量公式を用いれば、ある程度の推定が可能となったものと思われる。

5. 放物線越流頂の流量係数

堤頂下流側の越流面曲線が放物線で表わされる越流頂(放物線越流頂)は、すでに1.において紹介したゲート付越流頂の設計法にみられるように、非常に一般的な越流頂形状であり、わが国において Randolph あるいは Harrold の標準型越流頂を採用していないダムはすべてこの放物線越流頂であると言っても過言ではない。よって、このような放物線越流頂に対する流量公式の要請度も従来より非常に高く、現在では岩崎により提案された実用的な実験公式⁸⁾が広く使用されている。

そこで、ここではまず岩崎の実用公式と著者らの提案式とを比較検討し、両者の関係を明らかにしてのち、主として土木研究所が実施した多くの放物線越流頂の物理模型実験資料を用いて、著者らの提案した流量公式の適用性を検討してみる。

〔1〕岩崎の流量公式と著者の提案式の比較

著者らの導いた流量公式を現在広く使用されている岩崎公式と比較してみることは非常に興味あることと思われる。

岩崎の提案した放物線越流頂に対する実用流量公式は、すでに示したように次式で表わされた。

$$\left. \begin{aligned} C_d &= 1.971 + 0.498\xi + 6.63\xi^2 \\ C &= 1.60 \frac{1 + A(H_0/H_d)}{1 + 0.5A(H_0/H_d)} \end{aligned} \right\} \dots\dots\dots(13)$$

ここに $\xi = (Y/H_d)_0$ であって、 C_d は、堤頂と越流面始端(堤体上流面)との水平距離 W より $H_d = W/0.282$ として求められる非標準型越流頂の仮想設計水頭時の流量係数であり、 $(Y/H_d)_0$ は、上述の仮想設計水頭を用いて計算される $X/H_d = 0.5$ における無次元縦座標である。

また、この(13)式における係数 A は、 $H_0/H_d = 1.0$ すなわち $H_0 = H_d$ における $C = C_d$ を与えることにより、すでに示した(14)式で与えられる。すなわち、

$$A = \frac{C_d - 1.60}{1.60 - 0.5C_d} \dots\dots\dots(14)$$

いま、放物線越流頂の堤頂下流面曲線が図-6に示すように堤頂を原点とし、水平下流方向に X 軸、鉛直下向き方向に Y 軸をとった直角座標 (X, Y) において、 $Y = KX^2$ と表わされる場合をモデルとして取り上げ、岩崎の提案した(13)式に示される実用公式の構造を検討し、その物理的意義を調べてみよう。

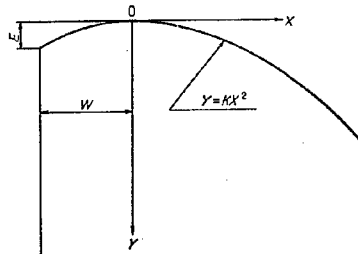


図-6 放物線越流頂記号説明図

まず、堤頂と越流面始端との水平距離を W とすれば、仮想設計水頭 H_d は W の関数として次のように表わされた。

$$H_d = W/0.282 \dots\dots\dots(21)$$

一方、この設計水頭を用いて計算される $X = 0.5H_d$ における無次元縦座標 $(Y/H_d)_0$ は、越流面曲線 $Y = KX^2$ が堤頂 $X=0$ における曲率半径を R として $Y = 1/2RX^2$ と変形しうることを考慮すれば、次のように表わされる。すなわち、

$$\left(\frac{Y}{H_d} \right)_0 = KH_d \left(\frac{X}{H_d} \right)^2_{X=0.5H_d} = \frac{(0.5)^2}{2 \times 0.282} \frac{W}{R} = 0.443 \frac{W}{R} \dots\dots\dots(22)$$

⁸⁾ 岩崎の実用流量公式は放物線越流頂にかぎらず、すべての標準型越流頂に類似の越流頂に対して導かれたものであり、公式を導くに用いた資料も Kirkpatrick が収録した TVA のダムに関する実験資料が大部分であって、実際に放物線越流頂の実験資料は Wilson グムの1例を含むにすぎない。よって、この公式を放物線越流頂に対する流量公式と限定することは誤謬であろうと思われる。
しかし、わが国の場合、標準型に類似する越流頂として復合円弧を用いたものは非常に少なく、大半がいわゆる放物線越流頂であり、かつ実際にも放物線越流頂に対する流量公式として利用されることが多いので、ここではこの実験式を放物線越流頂に対する公式として取扱うことにする。
しかし、この公式に対する論議は、放物線越流頂に限らず、標準型越流頂に類似のすべての越流頂に適用できるものと考えられる。

よって、(13)式に(22)式の関係代入すれば、 $H_d=W/0.282$ における流量係数 C_d は W/R の関数として、次のように表わすことができる。

$$C_d = 1.971 + 0.221 \frac{W}{R} + 1.303 \left(\frac{W}{R} \right)^2 \dots\dots (23)$$

ここに、 W および R は越流頂の形状寸法であり、 W/R は越流頂の堤頂上流形状の影響を表わすファクターと考えることができる。そこで、 W/R を m と表わし、仮想設計水頭 H_d を通常の水頭 H_0 と同様に取り扱い扱えば、 $H_0=R \cdot m/0.282$ における流量係数 C は次のように表わされることになる。

$$C = 1.971 + 0.221m + 1.303m^2 \dots\dots (24)$$

ここに $m=W/R$ である。

そこで、これらの関係をさきの流量係数の変化を示す(13)式の第2式に代入し、かつ H_0/H_d のかわりに H_0/R の関数に置換すれば、岩崎の提案式は次のように変形できる。

$$\begin{aligned} C &= 1.60 \frac{1+A(H_0/H_d)}{1+0.5A(H_0/H_d)} \\ &= 1.60 \frac{1+B(H_0/R)}{1+0.5B(H_0/R)} \dots\dots (25) \end{aligned}$$

ここに、

$$B = \frac{1}{m} \left\{ \frac{1}{1.362 - 0.245m - 1.444m^2} - 0.564 \right\}$$

である。

すなわち、岩崎の提案した実用流量公式は、仮想設計水頭を含む公式として(13)式のように提案されているけれども、実際には流量係数 C を H_0/R の関数として表わすことが可能であり、かつ堤頂上流部の形状の影響を含む公式であることがわかる。

つぎに、岩崎が(13)式を導くにあたり用いた実験資料を調べてみると、実験資料としては Kirkpatrick の集録した TVA のダムに関するものが大部分であり、これに土木試験所の竹内らの実験結果を加えた全部で12のダムに関するものである。そこで、この資料より、 $(Y/H_d)_0$ の範囲を求めてみると、 $(Y/H_d)_0$ は極端に小さい値を示す土試73-II および III の資料を除き、他はすべて $0.072 \leq (Y/H_d)_0 \leq 0.141$ の範囲にあり、大部分は $(Y/H_d)_0 = 0.1$ 前後の資料である。

いま、 $(Y/H_d)_0$ と W/R の間には、堤頂下流面曲線を放物線と仮定した場合に成立する(22)式の関係が常に成立するものと仮定して、この $(Y/H_d)_0$ の範囲より W/R の範囲を求めてみると、この岩崎式の W/R の資料の範囲は $0.16 \leq W/R \leq 0.32$ に相当することがわかる。

そこで、岩崎の流量公式に表わされる $m(=W/R)$ の流量係数に及ぼす影響を知るために、 m をこの範囲に限定して実際に流量係数を計算してみた。この計算結果は表-3および図-7に示すとおりである。

これをみると、岩崎公式においては $m=0.16 \sim 0.32$ の範囲において、流量係数は5%程度の変動を示すことがわかる。

一方、著者らの導いた越流頂の流量係数は、流れが越流底面に沿って流れることを前提として、放物線越流頂に対しても(15)式に示すように1次式で表わされる。

$$C = 1.704 \left(1 + 0.648 \frac{H_0}{R} \right)^{1/8} \dots\dots (26)$$

この(26)式によって与えられる流量係数は、さきの表-3図-7に併記されるが、この両者を比較してみると、著者

表-3 岩崎の実験公式と著者の提案式の比較

$\frac{H_0}{R}$	流 量 係 数 C (m-sec 単位系)									著者の式 Eq(26)
	岩 崎 の 提 案 式 Eq(25)									
	m=0.16	0.18	0.20	0.22	0.24	0.26	0.28	0.30	0.32	
0.0	1.600	1.600	1.600	1.600	1.600	1.600	1.600	1.600	1.600	1.704
0.1	1.700	1.693	1.688	1.684	1.681	1.679	1.677	1.676	1.675	1.758
0.2	1.789	1.776	1.767	1.760	1.754	1.750	1.747	1.745	1.744	1.811
0.3	1.867	1.850	1.838	1.828	1.820	1.815	1.811	1.808	1.807	1.862
0.4	1.937	1.917	1.902	1.890	1.881	1.874	1.869	1.866	1.864	1.912
0.5	2.001	1.978	1.960	1.947	1.936	1.929	1.923	1.919	1.917	1.961
0.6	2.058	2.033	2.014	1.999	1.987	1.979	1.973	1.968	1.966	2.008
0.7	2.110	2.083	2.063	2.047	2.034	2.025	2.018	2.014	2.011	2.054
0.8	2.157	2.130	2.108	2.091	2.078	2.068	2.057	2.056	2.053	2.100
0.9	2.201	2.172	2.150	2.132	2.118	2.108	2.101	2.096	2.093	2.144
1.0	2.241	2.212	2.188	2.170	2.156	2.145	2.137	2.132	2.129	2.187
1.1	2.278	2.248	2.224	2.206	2.191	2.180	2.172	2.166	2.164	2.230
1.2	2.312	2.282	2.258	2.239	2.224	2.213	2.204	2.199	2.196	2.272

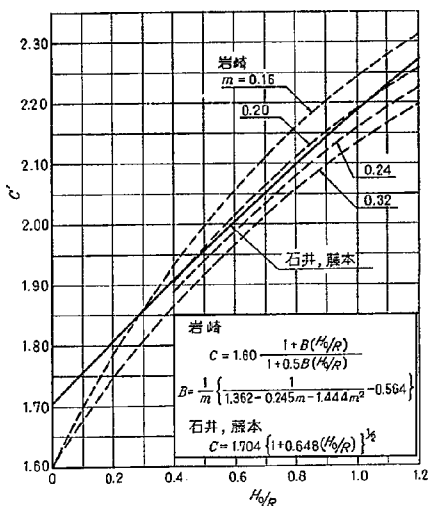


図-7 岩崎の実験公式と著者の提案式の比較
 らの理論式は岩崎公式の $m=0.18\sim0.20$ の流量係数と
 ほぼ一致した値を与えるようである。

しかし、理論式がそのまま適用できることが証明され
 た標準型越流頂では、 m の値がほぼ $0.30\sim0.31$ となり、
 岩崎の提案式ではこの付近の流量係数はいずれも著者ら
 の結果より低い値を示していることから考えて、(13)式
 の $H_d=W/0.232$ における流量係数の値は、橋脚、橋台
 等の影響を含んだ値と見なすことができ、実用上は(15)
 式の C' と同様に取り扱うべきと考えられる。

この点については、岩崎も彼の論文の末尾に、TVA
 のダムに関する Kirkpatrick の集録した実験資料は完全

な2次元模型の資料でないとの推論を示しており、これ
 は著者らの理論式との対比による推定結果と一致するわ
 けである。

しかし、著者らの導いた流量公式と岩崎の提案した実
 験式とに大差のないことも明らかであり、著者らの流量
 公式も岩崎式と同様に十分な適用性を有するものと考え
 てよいであろう。

また、岩崎の提案式に示される W/R 、すなわち堤頂
 下流面のふくらみに対する堤頂上流部の曲面の水平長、
 が減少するほど流量係数が全体的に大きくなるというこ
 とは、堤頂上流の越流面上の負圧の発生と関連があるも
 のと予想され、かなり実際に適合した結果を与えるもの
 と予想されるが、堤頂上流部の形状の影響をさらに厳密
 に論じるためには、岩崎の公式に含まれる橋脚橋台の効
 果を分離して取り扱うことが必要になる。

これは、流れが越流面に沿って流れることを前提とす
 る著者らの流量公式の適用限界とともにさらに実験的な
 検討が必要になるものと思われる。

〔2〕放物線越流頂の形状

放物線越流頂の必要理由については、すでに述べたよ
 うに各種の理由をあげることができるが、実際に実施さ
 れた越流頂より、その自由越流型余水止としての適用性
 を検討してみることは興味あることだと考えられる。

そこで、著者らの流量公式の適用性を検証するために
 収集した放物線越流頂に関する水理模型実験資料より、
 この形状特性と水理条件との関係を整理し、その適用条
 件を検討してみる。

まず、資料として収集されたわが国の放物線越流頂は
 表-4および図-8に示す全部で13ダム、14モデルに関する
 資料であり、このうち、市房(1)、杉田および鹿野川ダ
 ムの3ケースはいずれも原設計案に関するものであり、

表-4 収集した放物線越流頂の形状

No.	ダ ム 名	N_s	n	$L_s(m)$	堤 頂 上 流 部 形 状	堤頂下流部形状	$b(m)$	$S'(m)$	模型縮尺
1	水 沼	1	1	8.000	Stan. $H_d=8.717m$	$Y=0.02X^2$	—	—	1:40
2	笠 掘	1	1	10.000	Stan. $H_d=9.000m$	$Y=0.04329X^2$	—	—	1:50
3	五 郷	1	1	8.000	Stan. $H_d=6.500m$	$Y=0.03424X^2$	—	—	1:30
4	大 野	1	3	9.500	Stan. $H_d=10.700m$	$Y=0.03604X^2$	2.00	3.00	1:30
5	猿 谷	1	4	7.500	$W=3.375m, E=1.346m$	$Y=0.038964X^2$	2.50	5.407	1:25
6	市 房(1)	1	2	10.000	$W=3.500m, E=0.787m$	$Y=0.02417X^2$	4.00	4.50	1:30
7	市 房(2)	1	2	7.100	$W=6.500m, E=3.500m$	$Y=0.022454X^2$	4.00	6.50	1:30
8	永 瀬	1	5	12.000	$W=3.900m, E=1.650m$	$Y=0.03161X^2$	3.00	3.90	1:30
9	藤 原	1	3	9.500	$W=4.150m, E=1.350m$	$Y=0.025362X^2$	3.50	6.30	1:35
10	杉 田	1	4	12.000	$W=3.150m, E=1.535m$	$Y=0.029851X^2$	2.50	5.40	1:40
11	宮 川	1	3	10.000	$W=5.000m, E=1.800m$	$Y=0.027407X^2$	4.50	6.49	1:35
12	鹿 野 川	1	5	10.000	$W=4.500m, E=1.335m$	$Y=0.044899X^2$	2.50	6.942	1:55
13	田 瀬	1	6	10.000	$W=2.000m, E=1.000m$	$Y=0.037202X^2$	2.00	3.00	1:30
14	鳴 子	1	8	11.000	Stan. $H_d=3.000m$	$Y=0.26455X^2$	1.00	0.861	1:35

実際に実施された越流頂形状とは異なる形状である。市房(2)はこの実施設計形状であり、杉田、鹿野川両ダムの実施形状はすでに研究(2)で資料として用いた標準型越流頂がそれである。

また、表中堤頂上流部形状の欄において、“Stan”は

Randolph 型形状を、 W および E の値が記載されたものはそれ以外の形状を示しており、 W は堤頂と堤体上流面との水平距離を、 E は堤頂と越流面曲線始端との標高差を示している。

いま、これら既往の実験結果にみられる放物線越流頂

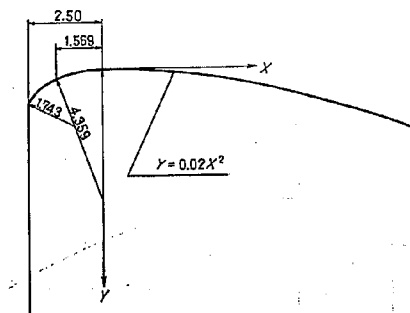


図-8-1 水沼ダムの越流頂

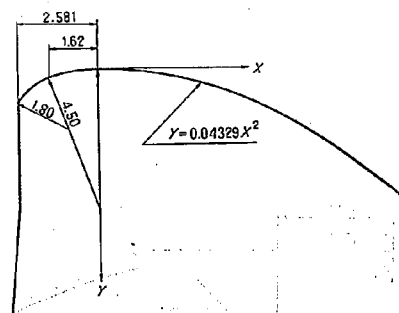


図-8-2 笠掘ダムの越流頂

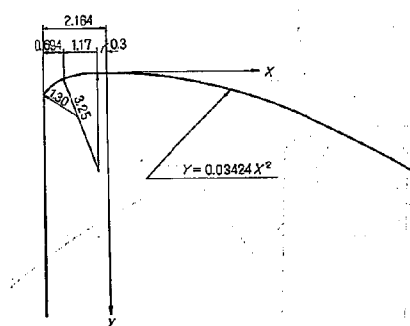


図-8-3 五郷ダムの越流頂

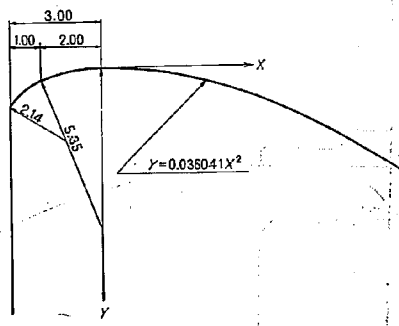


図-8-4 大野ダムの越流頂

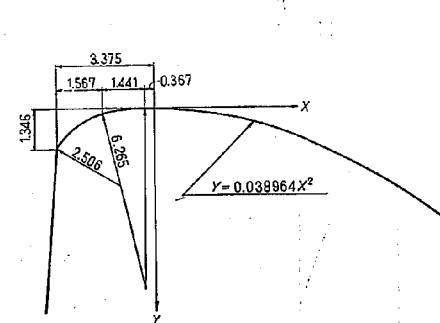


図-8-5 狼谷ダムの越流頂

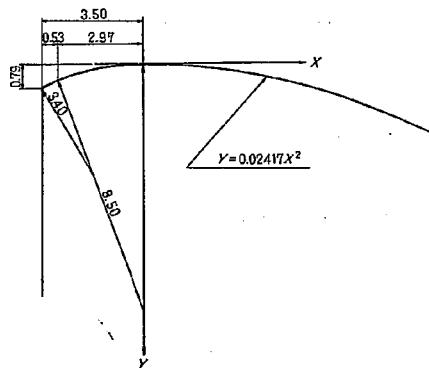


図-8-6 市房ダム(1)の越流頂

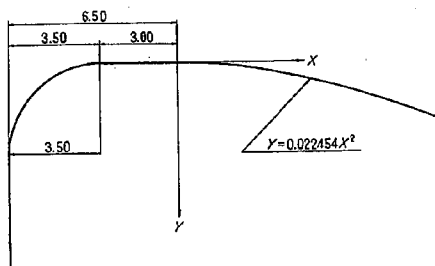


図-8-7 市房ダム(2)の越流頂

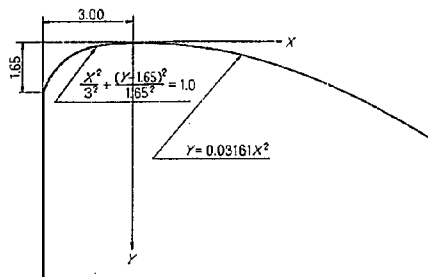


図-8-8 永瀬ダムの越流頂

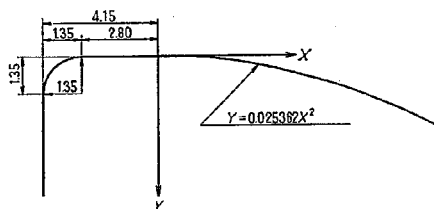


図-8-9 藤原ダムの越流頂

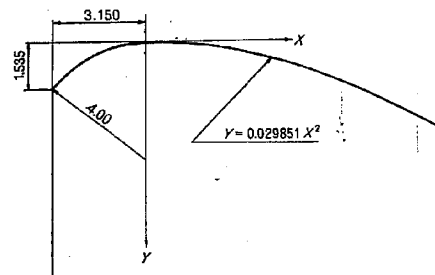


図-8-10 杉田ダムの越流頂

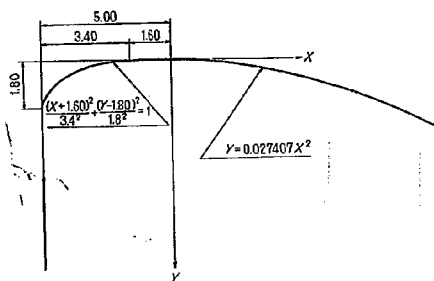


図-8-11 宮川ダムの越流頂

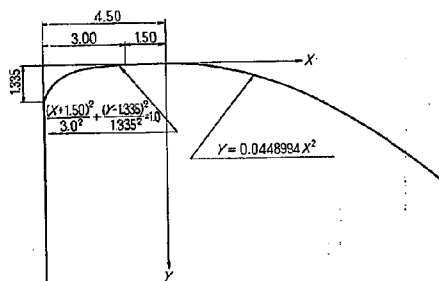


図-8-12 鹿野川ダムの越流頂

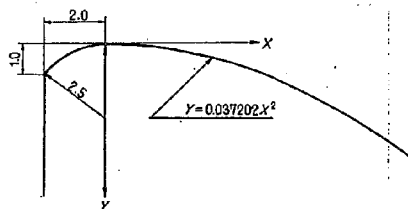


図-8-13 田瀬ダムの越流頂

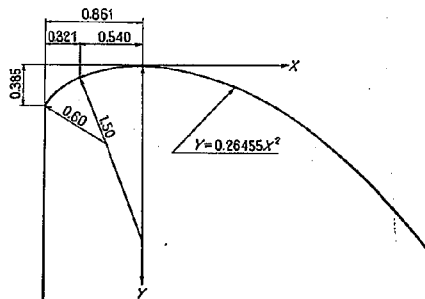


図-8-14 鳴子ダムの越流頂

の水利条件と越流頂形状との関係を整理し、その適用限界について若干の考察を試みよう。

まず、水利条件としては、越流頂の設計において重要な要素となる設計越流水頭 H_d を基準にとることが望ましいが、実験資料からはこの値は明確ではないので、ここでは設計水頭に対応する値として、測定値中の越流水頭の最大値 H_{max} を用いることにする。

なお、越流水頭の最大値 H_{max} は、通常異常洪水流量放流時に相当するものと予想され、計画洪水流量放流時を対象とする設計越流水頭 H_d に対して、 $H_{max}=1.1\sim 1.2H_d$ の関係が成立するものと考えられる。

一方、越流頂形状の特性量としては、放物線越流頂では、堤頂における越流底面の曲率半径 R が代表寸法となり、これに堤頂上流部の形状を示すパラメーターとしてさきの表-4において示したと同様に W および E を用いれば十分であろう。

いま、表-4の放物線越流頂について、 H_{max} および R を明らかにし、それらを用いて越流頂形状を無次元表示すると、表-5のようになる。

この表-5より、放物線越流頂において最も興味ある H_{max}/R の値について既往の形状をまとめてみると、 H_{max}/R は極端に大きい値を示す鳴子ダムの例を除き、いずれも $0.4\sim 1.0$ の範囲にあり、平均的には 0.7 程度の値を示している。これは、Randolph 型越流頂に相当する放物線越流頂として、研究(1)において求めた。

$$\frac{Y}{H_d} = \frac{H_d}{2R} \left(\frac{X}{H_d} \right)^2 = 0.545 \left(\frac{X}{H_d} \right)^2 \quad \dots\dots\dots (27)$$

においては、 $H_d/R=1.090$ となることから考えれば、いずれも標準型越流頂に比してはるかに厚みのある越流面曲線を使用していることがわかる。

一方、堤頂上にゲートを設置した越流頂を想定して、

ゲートの部分開扉時の放出水脈の下面形状を基本形状とする設計法にしたがい、その形状を求めてみると、初速 V_0 を流速係数を C_0 として、

$$V_0 = C_0 \sqrt{2gH_{max}} \quad \dots\dots\dots (28)$$

と表わせば、

$$\frac{Y}{H_{max}} = \frac{H_{max}}{2R} \left(\frac{X}{H_{max}} \right)^2 = \frac{1}{4C_0^2} \left(\frac{X}{H_{max}} \right)^2 \quad \dots\dots\dots (29)$$

と求められる。これより、

$$\frac{H_{max}}{R} = \frac{1}{2C_0^2} \quad \dots\dots\dots (30)$$

であって、 $C_0=0.9, 0.95$ および 1.0 の値に対し、 H_{max}/R は、それぞれ $0.617, 0.554$ および 0.500 と求められる。したがって、このような設計法に対しては既往の形状ははるかにやせた越流頂であるといえる。

よって、実際に放物線越流頂を設計する場合には、自由越流時の流量係数は H_{max}/R に比例して増大することおよび H_{max}/R が逆に大きすぎれば越流底面に負圧の発生するおそれが増大するという相反する2つの性質を勘案して設計することが必要であるが、既往の資料からは H_{max}/R はすべて 1.0 以下であり、かつ $0.6\sim 0.8$ 程度以下のものが大部分であることに留意して、 H_{max}/R がこの範囲以下を利用するように、放物線越流頂の下流面曲線を次式で設計しておけば十分満足できる結果が得られるであろう。

$$Y = \frac{(0.3\sim 0.4)}{H_{max}} X^2 \quad \dots\dots\dots (31)$$

つぎに、既往の放物線越流頂の堤頂上流部形状について、その特徴を調べてみる。

まず、これらの越流頂の堤頂上流部の形状は、標準型のものを除けば、その関数形により単一円弧、複合円弧および長円曲線の3種に分類され、さらに堤頂部の水平

表-5 放物線越流頂の形状特性

No.	ダム名	$H_{max}(m)$	$R(m)$	H_{max}/R	W/H_{max}	E/H_{max}	W/R	E/R
1	水沼	11.388	25.000	0.4555	0.2197	0.1002	0.1001	0.0456
2	笠棚	10.725	11.550	0.9286	0.2408	0.1098	0.2236	0.1020
3	五郷	10.933	14.603	0.7487	0.1706	0.0778	0.1277	0.0583
4	大野	9.560	13.873	0.6891	0.3212	0.1465	0.2214	0.1010
5	猿谷	10.725	12.833	0.8357	0.3147	0.1255	0.2630	0.1049
6	市所(1)	11.909	20.687	0.5757	0.2939	0.0661	0.1692	0.0380
7	市所(2)	16.862	22.267	0.7573	0.3855	0.2076	0.2919	0.1572
8	永瀬	13.545	15.818	0.8563	0.2879	0.1218	0.2466	0.1043
9	藤原	13.062	19.714	0.6626	0.3177	0.1033	0.2105	0.0685
10	杉田	15.312	16.750	0.9141	0.2057	0.1002	0.1880	0.0916
11	宮川	12.915	18.244	0.7079	0.3871	0.1394	0.2741	0.0987
12	鹿野川	8.800	11.136	0.7902	0.5114	0.1540	0.4041	0.1200
13	田瀬	7.325	13.440	0.5450	0.2730	0.1365	0.1488	0.0744
14	鳴子	4.800	1.890	2.5397	0.1794	0.0818	0.4556	0.2078

部の有無を考慮すれば計6種に分類できるが、このような詳細な曲面の構成要素よりも、表-5に示す W あるいは E の無次元値そのものの方がより重要なファクターとなるものと考えられるので、ここでは既往の形状においてこれらがどのような値を示しているかについてまてみる。

まず、堤頂と越流面始端との水平距離 W については、 $W/H_{\max}$ および W/R の形で示してあるが、 W/R は極端に大きい値を示す2つの例を除き、他はいずれも0.1～0.3の範囲にあり、平均値は $W/R=0.22$ である。また、堤頂と越流面始端との標高差 E についても W と同様な表示をしてあるが、 E/R がほぼ0.1程度の値を用いる例が多く、平均値は $E/R=0.09$ であって、 W と E の比はほぼ2.5程度になるようである。

このように、実際の曲面を構成する要素は複雑であるが、 W あるいは E の値そのものは常にある基準内に収まる値を用いていることがわかる。

よって、新しく放物線越流頂を設計する場合には、これらの平均的な諸数値を満たすように設計しておけば、十分に機能的な越流頂と見なすことができるであろう。

[3] 放物線越流頂の流量係数

つぎに、[2]で示した放物線越流頂に関する水理模型実験の資料を用いて、著者らの流量公式の放物線越流頂に対する適用性について検討してみる。

著者らの導いた放物線越流頂に対する流量公式はさきに示した(15)式であり、堤頂下流面曲線が $Y=KX^2$ と表わされる放物線越流頂においては $R=1/2K$ となることを利用し、 R を K に変換すれば、流量公式は次のように変形される。

$$\left. \begin{aligned} C' &= C[1-2.828M_r(KH_0)^{1/4}] \\ C &= 1.704(1+1.296KH_0)^{1/8} \\ M_r &= 0.0490(1/KL_s)^{0.50} \\ (n=1 \text{ あるいは } n \geq 2 \text{ かつ } b/S' \geq 0.80 \text{ に対して}) \\ &= 0.0490(1/KL_s)^{0.50} \left\{ \frac{1}{n} + 1.465 \right. \\ &\quad \times \left(\frac{n-1}{n} \right) \left(\frac{b}{S'} \right)^{1.7} \left. \right\} \\ (n \geq 2 \text{ かつ } b/S' < 0.80 \text{ に対して}) \end{aligned} \right\} \dots (32)$$

ここに、 C' : 橋脚および橋台の影響を補正した流量係数 (m-sec 単位系)、 C : 一様流れにおける流量係数 (m-sec 単位系)、 H_0 : 堤頂を基準面とする全水頭、 L_s : 1 径間当りの越流幅、 n : 径間数、 b : 橋脚幅、 S' : 堤頂から橋脚前面までの水平距離である。

いま、(32)式を用いて、表-4に示す放物線越流頂の流量係数を計算し、これを実験値とともに示したものが図-9である。

この図-9に示す実験値と計算曲線とを比較してみると、著者らの流量公式が十分な適用性を示す例として、笠掘、水沼、杉田および藤原の4ダムをあげることができる。また、計算曲線と実験値とが同一の傾向を示す例として、五郷、大野、猿谷、市房(1)、宮川、鹿野川および鴻子の計7ダムをあげることができるであろう。しかし、実験値と計算曲線とが異なった変化を示す例として、田瀬、市房(2)、および永瀬の3ダムがある。

このように実験値と計算曲線とが良い一致を示す例は少ないけれども、まったく異常な傾向を示す田瀬ダムの例を除けば、他のモデルの実験値はいずれも誤差の許容範囲を $\pm 5\%$ まで抜ければ包含しうる。

よって、このような判定が許容されるならば、著者らの流量係数推定式は放物線越流頂に対して十分な適用性

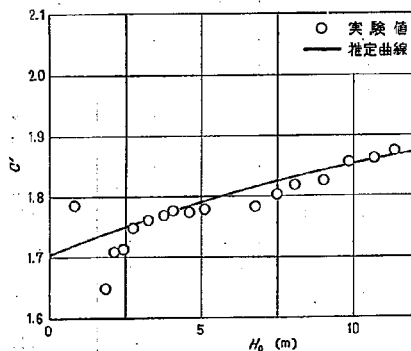


図-9-1 水沼ダムにおける実験値と推定曲線

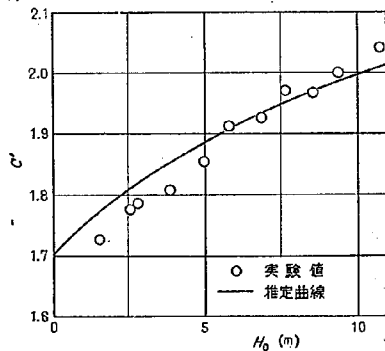


図-9-2 笠掘ダムにおける実験値と推定曲線

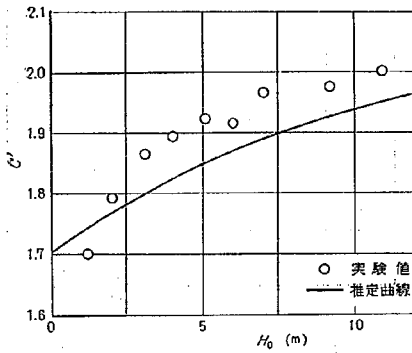


図-9-3 五郷ダムにおける実験値と推定曲線

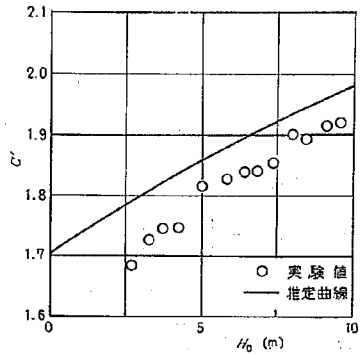


図-9-4 大野ダムにおける実験値と推定曲線

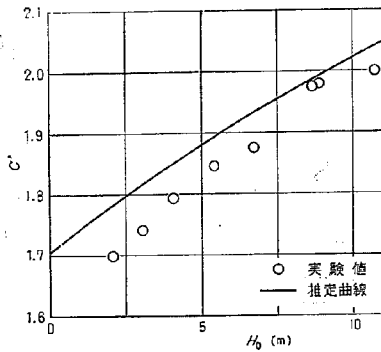


図-9-5 狹谷ダムにおける実験値と推定曲線

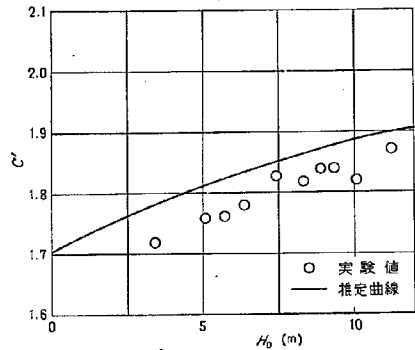


図-9-6 市房ダム(1)における実験値と推定曲線

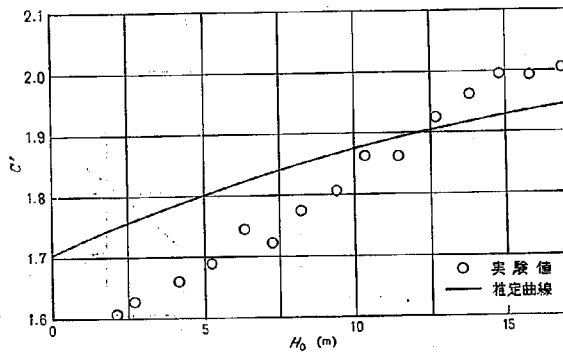


図-9-7 市房ダム(2)における実験値と推定曲線

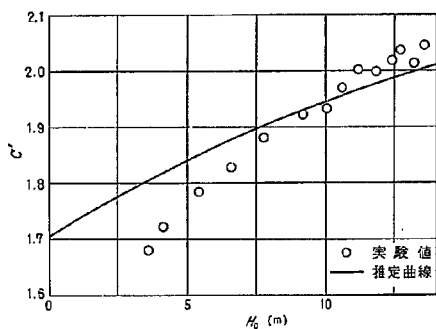


図-9-8 永瀬ダムにおける実験値と推定曲線

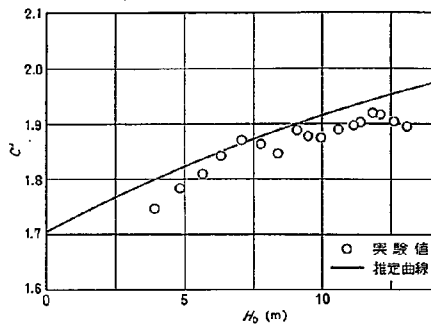


図-9-9 藤原ダムにおける実験値と推定曲線

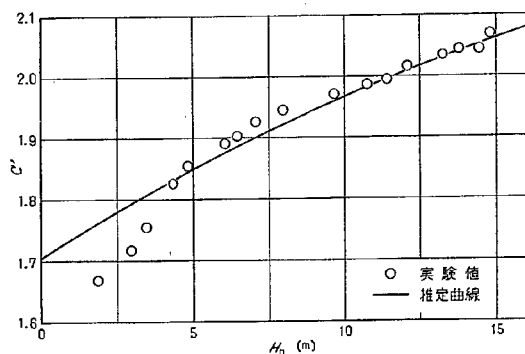


図-9-10 杉田ダムにおける実験値と推定曲線

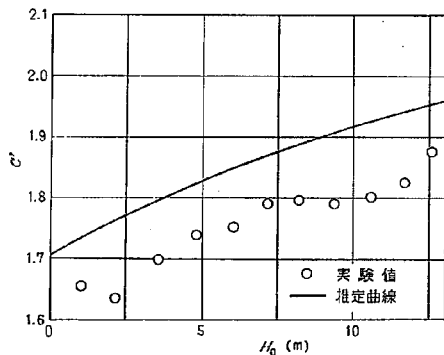


図-9-11 宮川ダムにおける実験値と推定曲線

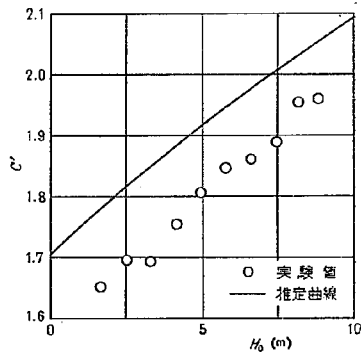


図-9-12 鹿野川ダムにおける実験値と推定曲線

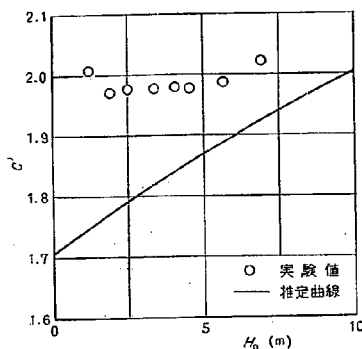


図-9-13 田瀬ダムにおける実験値と推定曲線

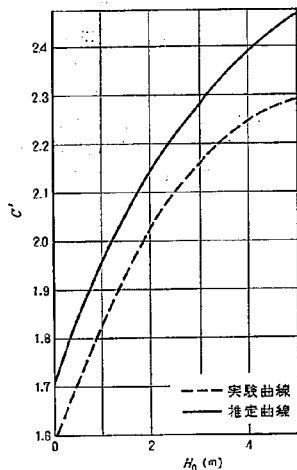


図-9-14 鳴子ダムにおける実験曲線と推定曲線

を有するものと断定できる。

しかし、実験値より計算値の方が大きい値を示す例がかなり多く、さきの円弧で構成された越流頂の場合と同様な傾向をみせたことは、理論式に基づく流量係数の基本値に対する各種の補正が十分でないものと予想され、さらに検討すべき余地があるものと考えられる。

なお、円弧あるいは放物線で構成された越流頂の流量係数におけるこのような傾向には、越流頂の底面形状そのものに起因する影響が含まれているものとも考えられ、(32)式における橋脚、橋台の流れに及ぼす影響の補正のほか、たとえば W/R あるいは E/R のような因子を導入して、標準型、非標準型越流頂の区別なく全資料に対して処理すれば、このような誤差の発生をある程

度防ぐことは可能になるものと思われる。しかし、このような越流底面すなわち理論式における境界条件そのものに関する要素を実験的に導入することは、流れが越流底面に沿って流れることを前提として導いた理論式に基づく流量係数の基本値そのものの補正を行うことになり、著者らの流量公式の基本的姿勢を不明確にするものであるので、ここではそのような補正は考えないことにした。

また、橋脚、橋台の影響についても、この非標準型越流頂の実験資料を含めて再検討すべきものと思われるが、 $H_0 \rightarrow 0$ における流量係数の収束値に根本的な問題があり、模型実験における縮尺効果および実験方法についてもさらに検討すべきであるので、ここではこのような係数の再検討は行なわないことにした。

6. 結 論

著者らは越流頂の流量係数について一連の研究を実施し、すでに標準型越流頂の流量係数についてはその推定式を提案したが、ここでは問題の対象をさらに一般的な越流頂にまで拡張し、その越流頂形状と流量係数との具体的な関係を示す越流頂の一般流量式について検討することにした。

この場合、越流頂の一般流量式はその性質上標準、非標準の区別なくすべての越流頂に適用できることが必要であることから、標準型越流頂に対する流量式もこの一般式に包含されなければならないので、ここではすでに提案した標準型越流頂の公式を基本式とし、この公式中に含まれる形状要素を越流頂に共通する形状要素に変換することによって、越流頂の一般流量式を導くことにした。

このようにして導かれた越流頂の一般流量式は、実際の堤頂における越流底面の曲率半径 R の関数として(15)式のように表わされた。

なお、この(15)式において用いられる曲率半径 R としては、研究(1)で行った Randolph 型越流頂に対する検討結果より、堤頂下流の越流面曲線を表わす数式より $X \rightarrow 0$ に対して計算される値を用いるものとした。

このように、ここでは越流頂の一般流量式を導くとともに、堤頂下流面曲線が円弧あるいは放物線で表わされる越流頂の実験資料および既往の実験公式を用いてこの式の適用性を検討した。

その結果、本公式による計算曲線は既往の実験資料および現在使用されている岩崎の実験公式とほぼ一致した値を示すことが明らかになり、これらの越流頂に対して(15)式は十分な適用性を有することが証明されたとともに、従来それぞれの形状に対して実験的にのみ論じられてきたこれらの越流頂の流量係数についても、ここに提案し

た方法を用いれば、統一的な取り扱いができることが明らかにになった。

しかし、本公式による計算曲線と既往の実験資料との間には一様な誤差の発生している恐れがあるとともに、実験資料の中には異常に小さな値を示す例も少なくないので、これらの資料の実験条件あるいは実験方法とともにさらに詳細な検討を要するものと思われる。

また、数学的には堤頂の曲率半径 R が有意な値とならない越流頂に対する計算法をはじめ、本公式の適用限界あるいは模型実験における縮尺効果等、越流底面上の流れの機構そのものに起因する問題もまだ数多く残されており、今後さらに理論、実験の両面から十分な検討を加える必要のあるものと考えられる。

なお、第3報では、第1報の越流頂の流量係数に関する理論的研究、第2報の標準型越流頂の実用流量公式に関する研究に引き続き、非標準型越流頂の流量係数について取り扱い、越流頂の一般流量公式を提案したが、本報をもって越流頂の流量係数に関する総括的な研究は一応終了したので、この第3報を越流型余水吐の水力機能に関する研究の最終報とする。

しかし、第1報において示したゲート調節時の流量係数あるいはもぐり流出の問題をはじめ、越流頂の流量係数について残された数多くの問題については、今後も研究を続け、それぞれ別個に発表する予定である。

最後に本研究を行うにあたり、ご指導くださった石井文雄赤羽支所長、伊東茂富ダム部長、川村幸司ダム水理研究室長に深甚な謝意を表します。また、計算、図面の作成あるいは資料整理等の労をわずらわしたダム水理研究室、吉村、竹田、二木、本木技官に感謝いたします。

参 考 文 献

1. Bradley, J. N : "Discharge Coefficients for Irregular Overfall Spillways." Engineering Monograph No. 9 U. S. B. R Denver Colo. March, (1952)
2. Buehler, B : Discussion on "Rating Curves for Flow over Drum Gates, by J. N Bradley" Trans. A S C E Vol. 119 (1954)
3. Kirkpatrick, K. W : "Discharge Coefficients for Spillways at TVA Dams." Proc. A S C E H Y, Vol. 81 Sep. No. 626 Feb. (1955)
4. 岩崎 敏夫 : 「越流頂余水吐きの流量係数について」土木学会論文集, 第43号 (1957)
5. 物部 長穂 : 「水理学」岩波(昭和8年) 229頁
6. Bradley, J. N : "Rating Curves for Flow over Drum Gates." Trans. A S C E Vol. 119. (1954).

建設技術に適応した電子計算システム に関する研究 (1)

— NEW ALGOL-H を中心とするシステム —

(昭和 42 年 12 月)

中 村 慶 一*

小 池 長 春**

A Study on the Electronic Computing System Adapted to the Civil Engineering Field

— A System with the NEW ALGOL-H as Central Compiler —

by Keiiti Nakamura
Nagaharu Koike

Synopsis

In the Public Works Research Institute, an electronic computing system with the NEW ALGOL-H as a central compiler has been established, and has several characteristics as follows :

- (1) calling external procedures which are written by 1500 programmers with ALGOL,
- (2) interrupting by preferential programs which might not exceed fifteen minutes for execution,
- (3) remote using from the Chiba Branch by means of electric transmission of program and error message tapes,
- (4) automatic designing using a digital plotter together.

This paper contains reasons why we selected ALGOL for compiler, comparison of NEW ALGOL-H and other compilers, and an outline of the new system which has above characteristics, and a conception of a new electronic computing system for the case of changing computer, and aimed to be usefull for same or extended scale computing system designing.

要 旨

土木研究所においては OKITAC 5090-H に対し NEW ALGOL-H を中心とする電子計算システムを確立し、(1) 1500 名の ALGOL プログラマにより製作されたプログラム中、汎用性のあるものは external procedure としてその名前を呼び出すことによる使用、(2) 15 分以下の計算の割り込み使用、(3) 千葉支所との間のプログラムやエラーメッセージの電送による遠隔地使用、(4) 図化機の併用による自動設計の実行、といった特長を持つ運営を行なっている。

本文はこれに対し使用用語を ALGOL に定めた理由、NEW ALGOL-H と他のコンパイラとの比較ならびに、土木研究所向けとして上にあげた特長を持つシステムの概要に触れ、さらに増大する要望に基づいて機種変更を行なう場合の新システムの構想等について述べ、同じ程度の計算システム設計ならびに拡張システム設計の参考となるようまとめたものである。

キーワード (電子, 計算, システム, ALGOL, 図化, 自動設計, コンパイラ)

* 技術管理室長 工助

** 技術管理室

目 次

はじめに.....	(2)
1° OKITAC 5090-H の概要.....	(2)
2° ALGOL と FORTRAN の比較.....	(5)
3° NEW ALGOL-H (H2) の構文法.....	(14)
4° NEW ALGOL-H の入出力文.....	(18)
5° 各種コンパイラの制限事項.....	(21)
6° NEW ALGOL-H のモニタシステム.....	(25)
7° 入力媒体の問題.....	(30)
8° 電子計算機の機械的改良点.....	(31)
9° 自動図化.....	(32)
10° アルゴリズムマニュアル.....	(34)
11° KDK-ALGOL の構想.....	(40)
12° 結 論.....	(42)

はじめに

土木研究所においては、昭和40年3月3日よりOKITAC 5090-H が稼動を開始した。これに先立って昭和39年3月より全所員ならびに他の建設省一般職員に対するALGOL研修を開始し、昭和43年1月現在1500名のプログラマによって効率的な稼動が行われている(文献1)。昭和38年頃からの調査により、各研究者が自分の持つ手順を計算機に直接指示するための言語としては、1960年制定1962年改訂の改訂ALGOL-60が最良であることは明らかとなっていたが、40年導入当時沖電気側より供給されたコンパイラは九大において開発されたALGOL-H(以下H1と称する)であった。このH1は処理速度が極めて早いがそのため文法上の制限が多く、OKITAC 5090-H の特質を十分表現し得ないという面があった。また長時間演算にはいってしまうと、たとえ5分で終了する計算であっても待ち時間が長くなる一方、処理プログラムの全くない時間も存在し得るのでこれらの平均化も考慮してモニタの統御下に文法制限の緩いNEW ALGOL-H(以下H2と称する)による処理中にさらにH2によるプログラム処理を割り込ませることを計画し、昭和40年1月沖電気の協力を得て概略完成されたものによる試行にはいり、その後H1の割り込み、プログラムの自動修正、プログラム印字の割り込み等の機能を追加してほぼ満足すべきものに仕上げるこ

とができた。

また各研究室における実験遂行に必要な諸手続きを始め、建設技術関係に汎用されると思われる諸手続きを完成し、external procedure として使用可能なようにシステムに格納したほか、昭和42年3月に導入した自動図化機に対応する諸手続きの作成もほぼ終了した。

一方、頻度の多い割り込み処理と増大するexternal procedure および長大化するプログラムに関連して、機種変更が計画されるに至り、新システムの計画に着手したので、建設技術に適応した電子計算システムの一例としてOKITAC 5090-H 用のNEW ALGOL-H を中心とするシステムを計画し改良して来た経過ならびにそれに関連して得た成果および新システム計画上の注意点についてのべる。

1° OKITAC 5090-H の概要

OKITAC 5090-H は沖電気製の中型電子計算機であって導入計画当時、演算速度 記憶容量 価格 ALGOL コンパイラ等の点で土木研究所の要求を満足し得るものとして選定されたものであって、増大する研究所の要求に対応して、磁気ディスク補助メモリ付きの機種への切り替えまでの3年間十分にその機能を果たし得た機種で、九州大学・関西原予・小樽商大・理研・沖センターに続く6号機であった。その構成および性能は次表のとおりである。

1.1 OKITAC 5090 H の構成

機 器 名	台数	機 器 名	台数
中央処理装置 (CPU)	1	電動タイプライタ (ET)	1
コントロール・コンソール	1	光電式紙テープ読取装置 (PTR)	3
紙テープ入出力チャネル・コントロール・ユニット	1	高速度紙テープさん孔装置 (HTP)	1
高速度印刷装置チャネル・コントロール・ユニット	1	高速度印刷装置 (LP)	1
磁気テープ装置チャネル・コントロール・ユニット	2	磁気テープ装置 (MT)	5
直流電源装置	3	自動電源電圧調整装置 (AVR)	5

1.2 各機器の性能

1.2.1 中央処理装置

演 算 方 式	回 路 形 態	スタティック
	直 並 列 の 区 分	1語並列, 同期式
	ビ ッ ト 形 式	純2進
デ ー タ 表 現	数 値 語 の 表 現	固定小数点形式: サイン 1ビット+数値 41ビット 浮動小数点形式: サイン 1ビット 仮数部 32ビット 指数部 9ビット
	文 字 の 表 現	1文字 6ビット 1 語 7文字
プ ロ グ ラ ム	内部、外部の区別 アドレス方式 命令語の形式	内部記憶方式 1 $\frac{1}{2}$ アドレス 1語1命令 多重指定: 1. アリスメティック・メモリ・アドレス 2. パーシャル・フィールド・アドレス 3. マスク・フィールド・アドレス 4. インデックス・アドレス 5. SCC 修飾 6. 間接アドレス 7. 多重修飾
	命 令 の 種 類	199 1. 演算関係 160 2. 入出力関係 39
インデックス レジスタ	語 長 インデックスの数 構 成	1語 15 1. 修飾部 14ビット 2. 増減部 14ビット 3. 基準部 14ビット
	修 飾 コ ン ト ロ ー ル	1. XR 修飾 2. XR ₁ と XR ₂ による二重修飾 3. 多重修飾 4. SCC 修飾
特 殊 レ ジ ス タ		マスク・レジスタ 16 パーシャル・レジスタ 15 リアル・タイム・クロック・カウンタ 1 インターラプション・マスク・レジスタ 2

主 記 憶 装 置	方 式	インターラプション・モード・レジスタ 2	
	容 量	ディスプレイ 1	
	容量/バンクユニット	インディケータ	
	語 の 構 成	磁気コア、一致電流方式	
演 算 速 度 (アクセス・タイム含む)		16,384 語	
		4,096 語	
		1 語	43ビット
		データ	41ビット
主 回 路 素 子		サイン	1ビット
		パリティチェック (偶数)	1ビット
	サイクル・タイム	10 μ s	
	加 減 算	30 μ s (平均)	固定 85 μ s (平均) 浮動 小数点 250 μ s (平均) 小数点 330 μ s (平均)
	乗 算	250 μ s (平均)	
	除 算	390 μ s (平均)	
	トランジスタ	23,000 個	
	ダイオード	44,000 個	

電子計算機空配置図

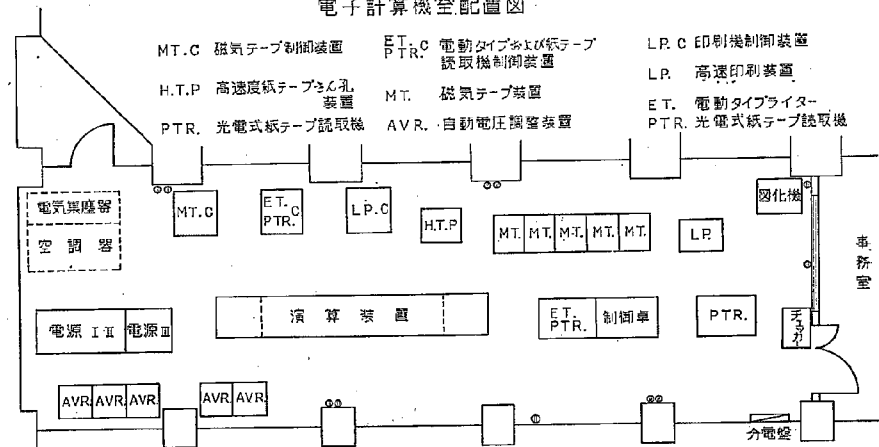


図 - 1

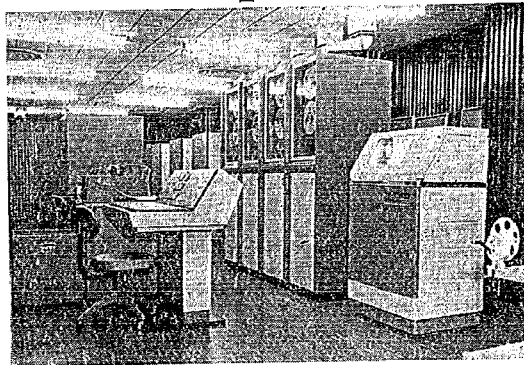


写真 - 1

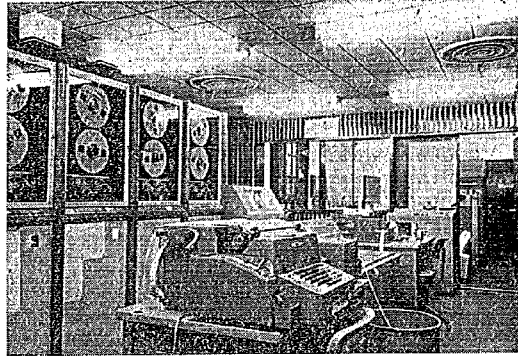


写真 - 2

1.2.2 光電式紙テープ読取装置

読取速度……………200字/秒, 400字/秒
 単 位……………8単位
 接続可能台数……………15台/チャンネル

1.2.3 高速度印刷装置

活字種類……………48種, 96種
 1行最大印字数……………120字
 印字速度……………500行/分
 印字間隔……………2.54 mm (1/10インチ)
 行間隔……………4.23 mm (1/6インチ)
 用紙幅……………7~14インチ
 接続可能台数……………15台/チャンネル

1.2.4 高速度紙テープさん孔装置

速 度……………150字/秒, 200字/秒
 接続可能台数……………15台/チャンネル

1.2.5 磁気テープ装置

テープ速度……………75インチ/秒, 150インチ/秒,
 112.5インチ/秒
 テープ幅……………1/2インチ
 伝送周波数……………41.7 kc, 62.5 kc
 接続可能台数……………15台/チャンネル

1.2.6 電動タイプライタ

印字用紙……………幅14インチのスプロケット用紙
 1行の最大印字数……………120字
 収容活字数……………96種

- i) 英大小文52, 数字10, 記号12 + - , = () ; * / 10 空白
- ii) 特殊記号 22 ! < > : ≥ ≤ := [] # ' ' † ≡ × ÷ √ ∧ ∩ ∪ 空白

さん孔……………8単位

改行間隔……………1/6インチ
 1/3インチ } 手動により切換えら
 1/2インチ } れる

印字, 読取, さん孔速度……………450字/分

接続可能台数……………15台/チャンネル

計算機室内の配置を 図-1 および 写真-1, 写真-2 に示す。

2° ALGOL と FORTRAN の比較

前述のように土木研究所において使用する電子計算用語としては ALGOL による方針のもとに選定が進められ, その後3回にわたる建設省電子計算シンポジウムにおいても建設省内における統一電子計算用語を用いることが確認されているが, その機種決定当時は, 特に現段階において普及度の大きい FORTRAN とのいずれによるべきかについて, 議論がくり返された。

たしかに既存のプログラムの数からいえば IBM の市場占有率ならびにその誕生の時点から見て FORTRAN によるものが多い。しかし その文法があまりにも機械的な約束に頼る部分が多いために, すでにできたプログラムを利用するだけでなく, これからプログラムを作る者に対する負担が ALGOL に比して多くかかる。また ALGOL は文法的に体系立っており, 機械的な制限事項が少ないのでその習得も極めて容易であるほか, その文法的特質を利用して興味ある作業を行なうプログラムに改造していくことも容易であるという理由から建設省関係においては ALGOL によって各人の考えているところを直接計算機に伝えるという方針が普及を開始し, 昭和43年1月現在で約 1500 名の ALGOL プログラマの

1) 建設省第2回電子計算シンポジウム報告 土木技術資料8巻3号(1966)

建設省第3回電子計算シンポジウム報告 土木技術資料9巻3号(1967)

養成を終わった。

以下 ALGOL と FORTRAN の特長について論じて見る。以下 FORTRAN を 'F', ALGOL を 'A' と略記する。

また、入出力関係は機種によって異なるので FORTRAN は HITAC 5020 用の HARP (機番を省略する) ALGOL に関しては OKITAC 5090-H 用の NEW ALGOL-H における入出力文を使用する。

2.1 プログラミング上の原則

2.1.1 プログラムは statement の集まりから成り立っており、特にその実行順序が変更されない限りはその

記述順に実行される。しかし 'F' では 1 つの statement (以下 $\tilde{S}$ と書く) は 1 行に、'A' では ; で区切って示される。

2.1.2 statement は 'A' においては begin end でくくって 1 つの statement とみなすことができるが、'F' にはこのような statement をひとまとめにする機能はない。

2.1.3 変数の型、配列の大きさなどを指定するためには 'F' には示様 statement, 'A' には declaration がある。

```
REAL IA, LB, MN
INTEGER AB, XY
LOGICAL A, GOOD
DOUBLE PRECISION P, Q
COMPLEX X, Y
DIMENSION A(15), B(50)
```

```
COMMON A, B, S
EQUIVALENCE (A, X)
```

```
real IA, LB, MN;
integer AB, XY;
Boolean A, GOOD;

array A[1:15], B[1:50];
integer array C, D[1:20, 1:30];
Boolean array S[1:10, 1:5, 1:100, 1:14];
```

- 注 1. 'A', 'F' 共示様 $\tilde{S}$ あるいは declaration ($\tilde{D}$) は実行可能な $\tilde{S}$ の前にこなければならない。'A' では途中に begin を置けば、その後に $\tilde{D}$ をおくことができ、この $\tilde{D}$ はその begin に対応する end までの間でのみ有効となる。(これをブロックと云い、このため数人のプログラマの共同作業が容易となる)
- 注 2. 名前は、'A', 'F' とともに、文字を先頭にしてその後に必要に応じて文字または数字が続いたものである。
- 注 3. 'F' では特に指定のないときは I, J, K, L, M, N を先頭とするものは integer 他は real と解され I~N を先頭とする real タイプの変数を必要とするときは示様を必要とする。'A' ではすべての名前は宣言を必要とする。
- 注 4. 論理変数は 'F' では LOGICAL, 'A' では Boolean として宣言するが論理定数は 'F' では TRUE, FALSE, 'A' では true false である。
- 注 5. DOUBLE PRECISION (倍精度演算) および COMPLEX (複素数演算) の示様は 'A' にはない。
- 注 6. 配列宣言は 'F' では A_1 より A_{15} までを A (15) 'A' では A [1:15] と書くが A_{20} より A_{35} のときも 'F' では A (35) として 35 個取らざるを得ない。'A' では A[20:35] とする。また A_{-15} から A_{10} までには 'F' では取れないが 'A' では A[-15:10] である。また 'A' では別段制限はないが FORTRAN II では 2 次元 FORTRAN IV では 3 次元までの配列しか認められない。したがって Boolean array S のような 4 次元の配列は、'F' では不可能である。
- 注 7. 'A' のようなブロック構造がないため 'F' では COMMON $\tilde{S}$ によって、このリストの先頭から何番目という順位のものとは異なった名前であっても共通扱いとなる。これによって COMMON 領域に場所が占有される。
- 注 8. 'A' のようなブロック構造がないため 'F' では同じ記憶場所を割当ててため EQUIVALENCE $\tilde{S}$ によって共用する。この場合 X は array であるとすれば B と X とは同じ場所を共用する。

2.1.4 I/O statement

'A' では別段規程はなく 'F' では IBM の規定に合せて format に至るまで詳細に規定してある。ここでは

これからの後の説明に必要な程度のものを 'F' は HITAC 5020 E 用 HARP, 'A' は OKITAC 5090-H 用の NEW ALGOL-H によって説明しておく。(ただし

HARP は機種指定のナンバーを含むのでこれは外しておく)

2.1.4.1 直結機器よりの読み込み

'F' では format 指定を必要とする。A, B で field 12 ケタの floating タイプの情報ならば,

```

      READ 100, A, B                      READ(A, B);
      100 FORMAT(2F12.0)

```

2.1.4.2 直結印刷機による印刷

小数点以上以下 7 ケタで field 15 ケタの情報 A, B, C を 1 行に印字したいとき,

```

      WRITE 101, A, B, C                  PRINT('3R/', A, B, C);
      101 FORMAT(1H, 3E15.7)

```

2.1.4.3 文字の印刷

N という integer タイプの情報 4 ケタを N= の右に

H という real タイプの情報 7 ケタを H= の右に 印刷するとき

```

      WRITE 104, N, H                    PRINT("N=" I, 'H=' R, N, H);
      104 FORMAT(1H, 2HN=I4, 2HH=E15.7)

```

2.2 式 (expression)

2.2.1 算術作用素

'F' では + - * / **

'A' では + - * / ↑ + (×の代りに*を用いるものが多い)

で+は整数割算であり 'F' では両辺整数のとき同じ結果となる。

'F' では ** を除いて両辺のタイプが同じでなければ誤りとなるが 'A' では両辺のタイプが異なるときは自動的に real となり / の両辺が整数のときも その結果は real となる。

2.2.2 比較作用素

'F' では .GT. .GE. .EQ. .LE. .LT. .NE.

'A' では > ≥ = ≤ < ≠

2.2.3 論理作用素

'F' では .NOT. .AND. .OR. 'A' では ¬ ∧ ∨ ⊃ ≡

2.2.4 算術式

'F' では算術定数, 算術変数(同じ型に限る). FUNCTION, 算術作用素, 括弧で構成される。

'A' では算術定数, 算術変数(real integer いずれも可。他の定数, 関数呼び出しも同じ), 関数呼び出し, 算術作用素, 括弧の外に if then else による条件算術式を作ることもできる。

$D \leq 0$ なら $XI = S + T$ さもなければ $XI = -B/A$ としたければ

```

      IF(D .LE. 0.0) GO TO 1              XI:=if D≤0 then S+T else -B/A;
      XI=-B/A
      GO TO 2
1  XI=S+T
2

```

$P = Q$ なら $A_1 + B$ さもなければ $A_2 + B$ を S に代入したければ

```

      IF(P .EQ. Q) GO TO 2                S=A[if P=Q then 1 else 2, I]+B;
      IX=2
      GO TO 3
2  IX=1
3  S=A(IX, I)+B

```

なお添字式については 'F' では (定数) * (整数型変数) ± 定数

という形までに限るが 'A' では条件式を含めて自由に算術式が書ける。したがって配列 A の もし $P = Q$ なら $(I+J) * IR$ 番目, そうでなければ $(I-J) * IR$ 番目の要素と C との積を P に代入するには

```

      IF(P .EQ. Q) GO TO 20              [ P:=A[if P=Q then (I+J) * IR else
      J1=(I-J) * IR                      (I-J) * IR] * C;
      GO TO 30

```

```

20 J1=(I+J)*IR
30 P=A(J1)*C

```

となる。

2.2.5 論 理 式

‘F’では logical expression と云い2つの算術式を比較作用素で結合したもの、論理定数、論理変数、論理関数またはこれらを論理作用素や括弧で結合したものである。

‘A’では Boolean expression と云い、この他 if then else による条件付論理式を作ることできる。

$P > Q$ なら $A=B$ かどうかによって XI に $R+S$ または $R-S$ を代入し、さもなければ $A < 100$ かどうかによって XI に $P*S$ または R/S を代入したいときは

<pre> IF(P .LE. Q) GO TO 1 IF(A .EQ. B) GO TO 2 XI=R-S GO TO 4 2 XI=R+S GO TO 4 1 IF(A .LT. 100.0) GO TO 3 XI=R/S GO TO 4 3 XI=P*S 4 </pre>	<pre> XI:=if P>Q then (if A=B then R+S else R-S) else if A<100 then P*S else R/S; </pre>
---	--

2.2.6 行 先 式

‘F’では名札は符号なし整数に限るが‘A’では行先式があり名札(レーベル)、スイッチおよび if then else を含んだ式を認める(ALGOL 60 ではレーベルとして名前の他に符号なし整数も認める)

$P > Q$ なら $A=B$ かどうかによって $L1$ (または1)か $L2$ (または2)に分岐し、さもなければ $A < 100$ かどうかによって $L3$ (または3)か $L4$ (または4)に分岐したいときは

<pre> IF(P .GT. Q) GO TO 5 IF(A .LT. 100.0) GO TO 3 GO TO 4 5 IF(A .EQ. B) GO TO 1 GO TO 2 </pre>	<pre> go to if P>Q then (if A=B then L1 else L2) else if A<100 then L3 else L4; </pre>
---	--

2.3 代入文 (assignment S)

‘F’では $\tilde{V} = \tilde{A}\tilde{E}$ ‘A’では $\tilde{V} := \tilde{A}\tilde{E}$; $\tilde{V} := \tilde{B}\tilde{E}$; 関数名 := $\tilde{A}\tilde{E}$; 関数名 := $\tilde{B}\tilde{E}$;

でいづれも右辺の型から左辺の型への変換は自動的に行なわれ‘F’は切り捨て‘A’は右辺 real 左辺 integer のときは絶対値が $|\tilde{A}\tilde{E}|$ を越えない最も $\tilde{A}\tilde{E}$ に近い整数となる。また‘A’では多重代入文を認める。

2.4 飛び越し文 (go to S)

2.4.1 無条件飛び越し文

‘F’では $GO TO <label>$ (label は符号なし整数に限る)

‘A’では $go to <designational expression>$;

で‘A’では label の他にスイッチ、条件付行先式などが許される。

2.4.2 条件付飛び越し文

$A > B$ なら $L4$ (または4) さもなければ $L5$ (または5) に行きたいとき

<pre> IF(A .GT. B) GO TO 4 GO TO 5 </pre>	<pre> go to if A>B then L4 else L5; </pre>
---	---

$A*B < 0$ なら $L7$ (または1) $=0$ なら $L2$ (または2) >0 なら $L3$ (または3) へ行きたければ

<pre> (IF(A*B) 1, 2, 3 </pre>	<pre> go to if A*B<0 then L7 else if A*B=0 then L2 else L3; </pre>
--------------------------------	---

2.4.3 computed GO TO またはスイッチ

J が1であると共に I が1なら $L11$ (または11) 2なら $L12$ (または12) 3なら $L13$ (または13) へ行き
 J が2であると共に I が1なら $L21$ (または21), 2なら $L22$ (または22), 3なら $L23$ (23) へ行きたければ

```

GO TO (1, 2), J
1 GO TO (11, 12, 13), I
2 GO TO (21, 22, 23), I

```

```

switch S1:=L11, L12, L13;
switch S2:=L21, L22, L23;
switch S:=S1[I], S2[I];
としておいて go to S[J];

```

2.5 条件文 (conditional $\tilde{S}$)

2.5.1 一般

'F'では IF (論理式) $\tilde{S}$

によって真なら $\tilde{S}$ を実行, 偽なら次の $\tilde{S}$ へ行く

'A'では if $\tilde{B}\tilde{E}$ then $\tilde{U}\tilde{S}$; ($\tilde{U}\tilde{S}$ は無条件文)
 if $\tilde{B}\tilde{E}$ then $\tilde{F}\tilde{S}$; ($\tilde{F}\tilde{S}$ は繰り返し文)
 if $\tilde{B}\tilde{E}$ then $\tilde{U}\tilde{S}$ else $\tilde{S}$;

の3種がある 真のとき then の後が実行される。

'A'の場合は else の後にまた if がきてもよいし, then の後も begin end でつむなら if がきてもよい。

2.5.2 算術 IF $\tilde{S}$

'F'では IF ($\tilde{A}\tilde{E}$) n_1, n_2, n_3

がある。 $\tilde{A}\tilde{E} < 0$ なら n_1
 $= 0$ なら n_2
 > 0 なら n_3

にコントロールが変更される。

'A'では if $\tilde{A}\tilde{E} < 0$ then go to n_1 else
 if $\tilde{A}\tilde{E} = 0$ then go to n_2 else go to n_3 ;

か, または go to n_i の代りに実行したい $\tilde{S}$ を書けばよい。これが複数個のときは begin end で包む。

2.6 繰り返し文 (for $\tilde{S}$) ($\tilde{F}\tilde{S}$ と略)

'F'では DO $\tilde{S}$ があり,

DO <label> $I=m_1, m_2$

または DO <label> $I=m_1, m_2, m_3$

がある。 I は integer 変数で m_1 から m_3 おきに m_2 まで指定されたレーベルまでの $\tilde{S}$ が実行され, m_3 が無いときは $m_3=1$ と解される。指定されたレーベルのついた $\tilde{S}$ は実行可能な $\tilde{S}$ でなければならないため GO TO $\tilde{S}$ や算術 IF $\tilde{S}$, DO $\tilde{S}$ その他 STOP 等の $\tilde{S}$ は禁止されるので一般には CONTINUE と書いておく。

'A'では

for $I=<\text{for list}>$ do $\tilde{S}$;

の形をとり <for list> は

1. $\tilde{A}\tilde{E}_1$ step $\tilde{A}\tilde{E}_2$ until $\tilde{A}\tilde{E}_3$
2. $\tilde{A}\tilde{E}$
3. $\tilde{A}\tilde{E}$ while $\tilde{B}\tilde{E}$
4. 1~3のいくつかをカンマでつないだもの

のいずれでもよいし, また $\tilde{S}$ はどの $\tilde{S}$ であってもよいし, いくつかあるときは begin end によって1つの $\tilde{S}$ に直せばよい。また 'F' とちがって real type のコントロール変数も自由に使用できる。 I を1から1きざみで50まで J を1から1きざみで100まで変えてマトリクス和

$$c_{ij} = a_{ij} + b_{ij}$$

を行なうには

```

DO 10 I=1, 50
DO 20 J=1, 100
20 C(I, J)=A(I, J)+B(I, J)
10 CONTINUE

```

I が1から2.5きざみで20まで 5きざみで100まで $\sqrt{T}$ を計算して I と $\sqrt{T}$ をプリントするなら

```

DO 50 J=1, 8
AI=FLOATF(J)*2.5-2.5+1
SAI=SQRTF(AI)
WRITE 100, AI, SAI
100 FORMAT(1H0, 2F15.7)
50 CONTINUE
DO 60 J=25, 100, 5
SAI=SQRTF(FLOATF(J))
WRITE 110, J, SAI
110 FORMAT(1H0, I3, F15.7)
60 CONTINUE

```

2.7 コメント

‘F’では左欄に C と書けば 72 けた以下何を書いてもよいし ‘A’ では begin および ; の次は comment と書いた後何を書いてもよく ; で終わる。また end の後は end else ; 以外は何を書いてもよい。

2.8 配列の処理

ベクトル・マトリクス等は配列として扱うことができるが ‘F’ では 3 次元までに限り, ‘A’ には制限はない。3 行 5 列の行列を行なりに読んでこれを転置した後, 新しい行列を行ごとに改行して印刷するプログラムを作ってみる。ただし各要素は 4 けた以下の整数とする。

```

DIMENSION IA(3, 5)
READ10, ((IA(I, J), J=1, 5), I=1, 3)
10 FORMAT(I4)
DO 30 I=1, 3
DO 30 J=1, 5
IB=IA(I, J)
IA(I, J)=IA(J, I)
30 IA(J, I)=IB
WRITE 50, ((IA(I, J), I=1, 5),
J=1, 3)
50 FORMAT(I4/)
STOP
END

```

```

begin integer I, J;
integer array IA[1:3, 1:5]; READ(IA);
for J:=1 step 1 until 5 do
begin for I:=1 step 1 until 3 do
begin integer A; A:=IA[I, J];
IA[I, J]:=IA[J, I];
IA[J, I]:=A; PRINT(I, A);
end; PRINT('/');
end;
end;

```

N 個の浮動小数点モードの情報を N を先頭に読み, その逆数の和を印刷せよ, ただし 0 の逆数は 0 とする (N ≤ 100)

```

DIMENSION A(100), B(100)
READ 10, N
10 FORMAT(I3)
READ 20, (A(I), I=1, N)
20 FORMAT(F 15.7)
SUM=0.0
DO 30 I=1, N
IF(A(I)) 1, 2, 1
2 B(I)=0.0
GO TO 30
1 B(I)=1.0/A(I)
30 SUM=SUM+B(I)
WRITE 40, SUM
40 FORMAT(F15.7)
STOP
END

```

```

begin integer N, I;
real SUM; array A[1:100];
READ(N); READARRAY(A, 1, N);
SUM:=0;
for I:=1 step 1 until N do
SUM:=SUM+(if A[I]=0
then 0 else 1/A[I]);
PRINT('R/', SUM);
end;

```

$y=x^5+4x^3+3$ について $x=-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$
 に対して計算し各値および最大値を $MAXY=$ として印刷せよ

```

    DIMENSION IY(11)
    DO 40 I=1, 11
      IX=I-6
      IY(I)=IX**5+4*IX**3+3
      IF (I-1) 20, 10, 20
10    IMAX=IY(I)
20    IF (IMAX .LT. IY(I)) IMAX=IY(I)
40    CONTINUE
      WRITE 60, (IY(I), I=1, 11)
60    FORMAT(I10)
      WRITE 50, IMAX
50    FORMAT(1H0, 5HMAXY=I10)
      STOP
    END
  
```

```

begin integer X, Y, MAXY; array YA(-5:5);
for X:=-5 step 1 until 5 do
  begin Y:=X↑5+4*X↑3+3; YA(X):=Y;
  end;
MAX:=YA(-5);
for X:=-5 step 1 until 5 do
  begin if MAX<YA(X) then MAX:=YA(X);
  PRINT('I', YA(X));
  end;
PRINT('/ 'MAXY='I'/, MAX);
end;
  
```

2.9 スイッチ

条件によって分岐するために 'F' には computed GO TO $\checkmark$ が, 'A' にはスイッチがある。

クリスマスカードを送るのに相手がフランス語を解するなら 1, 日本語を解するなら 2, 英語を解するなら 3 を入れて, かつ受取ったほか 12 月 1 日から 25 日までなら, それぞれ

JOYEUX NOEL

クリスマス おめでとう

MERRY CHRISTMAS

12 月 26 日より 1 月 31 日はそれぞれ

BONNE ANNEE

新年おめでとう

A HAPPY NEW YEAR

として 2 月から 11 月までは白紙にして印刷するプログラムは

```

    LOGICAL I,J
    READ 10 N,D
10    FORMAT(I1,F4.2)
    J=D .LT. 2.01 .OR. D .GT. 11.30
    I=J .AND. D .GE. 12.01 .AND. D .LE. 12.25
    GO TO (20,30,40), N
20    IF (I) GO TO 101
    IF (J .AND. .NOT. I) GO TO 102
    GO TO 450
101    WRITE 201
201    FORMAT(11H JOYEUX ♪ NOEL)
    GO TO 450
102    WRITE 202
202    FORMAT(11H BONNE ♪ ANNEE)
    GO TO 450
30    IF (I) GO TO 111
    IF (J .AND. .NOT. I) GO TO 112
    GO TO 450
111    WRITE 301
301    FORMAT(18H KURISUMASU ♪ OMEDETO)
    GO TO 450
112    WRITE 302
302    FORMAT(14H SINNEN ♪ OMEDETO)
    GO TO 450
40    IF (I) GO TO 121
    IF (J .AND. .NOT. I) GO TO 122
    GO TO 450
121    WRITE 401
401    FORMAT(15H MERRY ♪ CHRISTMAS)
    GO TO 450
122    WRITE 402
402    FORMAT(16H A ♪ HAPPY ♪ NEW ♪ YEAR)
450    WRITE 500
500    FORMAT('/')
    STOP
  END
  
```

```

begin integer N,I;
  real D;
  switch SF:=FC,FN,END;
  switch SN:=NC,NN,END;
  switch SE:=EC,EN,END;
  switch S:=SF(I),SN(I),SE(I);
  procedure G; go to END;
  go to ST;
FC: PRINT("JOYEUX#NOEL"); G;
FN: PRINT("BONNE#ANNEE"); G;
NC: PRINT("KURISUMASU#OMEDETO"); G;
NN: PRINT("SINNEN#OMEDETO"); G;
EC: PRINT("MERRY#CHRISTMAS"); G;
EN: PRINT("A#HAPPY#NEW#YEAR"); G;
ST: READ(N,D);
  I:=if D<2.01 ∨ D>11.30 then (if D≥12.01
    ∧ D≤12.25 then 1 else 2) else 3;
  go to S(N);
END: PRINT('/');
end;
  
```

2.10 SUBROUTINE または procedure

まとまった手続をある名前とパラメタ (ない場合もある) で呼び出して使用するために 'F' には SUBROUTINE, 'A' には procedure がある。

'F' では dummy argument として添字なし変数が配列の名前他の FUNCTION, SUBROUTINE の名前が許されるし,

'A' では formal parameter として変数, 配列, 他の procedure, 他の function designator, label, switch, string が許される。

'F' ではこの使用は CALL statement により, 'A' では procedure statement によって行なう。

$\Sigma x + \Sigma x^2 + \Sigma x^3$ を x を読んで計算し結果をプリントさせるプログラムを作ってみる。データは個数 N を先頭に x が N 個並んでいるものとする。 ($N \leq 100$)

```

    DIMENSIONE(100)
    READ 10, N
10  FORMAT(I3)
    READ 15, (A(I), I=1, N)
    CALL SUM(N, 1, A, SX)
    CALL SUM(N, 2, A, SXX)
    CALL SUM(N, 3, A, SXXX)
15  FORMAT(F10.3)
    WRITE 20, SX+SXX+SXXX
20  FORMAT(I 8 / )
    STOP
    SUBROUTINE SUM(N, P, A, S)
    S=0
    DO 30 I=1, N
30  S=S+A(I)**P
    RETURN
    END

```

```

begin real X, SX, SXX, SXXX;
integer N; array A[1:100];
procedure SUM(N, P, A, S);
value N, P; integer N, P;
real S; array A;
begin integer I; S:=0;
for I:=1 step 1 until N do
S:=S+A(I)↑P;
end;
READ(N); READARRAY(A, 1, N);
SUM(N,1,A,SX); SUM(N,2,A,SXX);
SUM(N,3,A,SXXX);
PRINT('R/', SX+SXX+SXXX);
end;

```

2.11 FUNCTION または function designator

2.10 とほとんど同じであるが 1 価関数の場合自分自身の名前に答を代入し, 式の中で直接使うことができるものとして 'F' に FUNCTION, 'A' に function designator がある。

2.10 の例を書き直すと

```

    DIMENSION A(100)
    READ 10, N
10  FORMAT(I3)
    READ 15, (A(I), I=1, N)
15  FOR MAT(F10.3)
    WRITE 20, SUM(N,1,A)+SUM(N,2,A)+SUM(N,3,A)
20  FORMAT(I 8 / )
    STOP
    FUNCTION SUM(N, P, A)
    S=0
    DO 30 I=1, N
30  S=S+A(I)**P
    SUM=S
    RETURN
    END

```

```

begin real X; integer N; array A[1:100];
real procedure SUM(N, P, A);
value N, P; integer N, P; array A;
begin integer I; SUM:=0;
for I:=1 step 1 until N do
SUM:=SUM+A(I)↑P;
end;
READ(N); READARRAY(A, 1, N);
PRINT('R/', SUM(N, 1, A)
+SUM(N, 2, A)+SUM(N, 3, A));
end;

```

2.12 標準関数等

2.12.1 標準関数

宣言を要しないで使用できる標準関数として

'F' には

組込み関数

ABSF(real)

IABSF(integer)

ライブラリ関数

LOGF(real)

SINF(real)

COSF(real)

EXP F(real)

SQRT F(real)

ATANF(real)

TANH F(real)

MODF(real) 'A' の entier に当る

INTF(real) 切りすて

FLOAT F(integer)

IFIX F(real)

SIGNF(A₁, A₂) A₁ の絶対値の A₂ の符号

等があり

'A' には

abs, ln, exp, sin, cos, arctan, sqrt, sign,
entier

がある。

sign は負なら -1, 0 なら 0, 正なら +1 となる。

2.12.2 演算ステートメント関数

'F' には

TAN(X)=SIN(X)/COS(X)

という形で演算ステートメント関数を使用できる。

'A' では関数としての宣言を行なって

real procedure TAN(X)

value X; real X;

TAN:=sin(X)/cos(X);

としておいて本文中で TAN(X) を式の中で用いることになる。

2.12.3 EXTERNAL

'F' には

EXTERNAL I, ..., I

'A' には

external I, ..., I;

または

external <type> I, ..., I;

すでに用意してある手続きの宣言である。

結 論

以上個別に比較したものを本質的な相違点のみとり出して比較してみると次のようなものになろう。

問 題 点	'F'	'A'
1. ブロック構造	ない。したがって特に共通な名の定義のために COMMON, 部分的に番地を共有するために EQUIVALENCE を用い面倒になる代りに、無宣言で名前を使用できる。 (ただし I~K が先頭になると integer 扱いになるので注意を要する)。	ある。したがって部分的に使いすぎるものはブロックの中で宣言すればよい。 その代りすべての名前は宣言を必要とする。
2. integer real の混合演算	認めない。 (換算してから計算)	認める。
3. 配列の次元数に関する宣言	'F II' では 2 次元 'F IV' では 3 次元まで	無制限。
4. 繰返しのコントロール変数 (または index)	integer に限る。 (DO の後のステートメント中で適当に変換する必要がある)	制限なし。
5. 倍精度演算	DOUBLE 宣言でよい。 (ただし 'F V' に限る)	B 5500 の DOUBLE. 可変語長の FACOM などを除き、特別な措置を要する。
6. 複素数宣言	COMPLEX 宣言でよい。 (ただし 'F V' に限る)	procedure を作る必要がある。

問 題 点	'F'	'A'
7. 条 件 式	ない。条件命令をいちいち出して行なう要あり。 <pre> IF(C EQ. D) GO TO 10 IF(E NE. F) GO TO 20 B=A(S, P, Q, U)+A2*C2 GO TO 80 20 B=A(S, P, Q, U)+A1*C2 GO TO 80 10 IF(E NE. F) GO TO 30 B=A(R, P, Q, U)+A2*C2 GO TO 80 30 B=A(R, P, Q, U)+A1*C2 80 </pre> であるがしかも 'F' では 4 次元の配列 A(R, P, Q, V) は不可能である。	算術式、論理式、行先式に対してある。 <pre> B:=A [if C=D then R else S, P, Q, if C>10 then U else V]+(if E≠F then A1 else A2)*C2; </pre> もし 'F' の各行を逐次 'A' におきかえたいならば上から順に <pre> if C=D then go to 10; if E≠F then go to 20; B:=A(S, P, Q, U)+A2*C2; go to 80; </pre> などとすれば良い。しかし逆に 'A' をこのように 'F' に逐次訳すること は不可能である。

したがって、以上を要するに 4 次元以上の配列が必要となると、それを問題の方で避けるように工夫し、real integer の混合演算に気を付け、何人かで共同作業をするときに共通の名前に気をくばり、条件式が出てくるときは、これを多少面倒でも条件命令に分解する労をいとわないならば、プログラミングだけを専門とする者にとっては 'F' も 'A' も同じことであろう。しかし仕事の途中で自分のプログラムを書く積りの人にとっては、何が何でも理屈なしに例外的な規則を棒暗記することが苦にならない人は別として、スジの通った文法によって、自由にわれわれの考えたままの道筋にしたがってプログラムを書き下していく上に 'A' が向いているということができる。

したがって建設技術関係においては各人が自分の意図をもって計算機と直接会話するのに便利で、すでに作られたプログラムを改良しやすいという点から ALGOL による方針を立て、すでに建設省関係に 1500 名のアルゴルプログラマの養成が終わっている。

3° NEW ALGOL-H (H2) の構文法

1958 年 6 月 GAMM (Gesellschaft für angewandte Mathematik und Mechanik) ならびに ACM (Association for Computing Machinery) の関係者がチューリヒにおいて ALGOL 予備報告を發表、その後 1960 年各国の関係者がバリーに会合して ALGOL-60 の文法をまとめ 1962 年 IFIP (International Federation for Information Processing) の報告として改訂 ALGOL-60 が發表 (文獻 5) され今日に至っている。

この ALGOL は 1956 年 IBM のユーザ達によって作られた FORTRAN (formula translator) が機械語の論理とあまり差がないのに対し、あくまでも一貫した理論的文法のもとに、日常語に近い表現で記述できると

共に機械対人間のみならず、人間対人間の情報伝達ならびに共同作業にも便利であるようにブロック構造という特長が組み込まれた算法定言語 (algorithmic language) であるが、その文法は Backus notation と云われる厳密ではあるが難解な記述方法が取られたためにその普及が遅れている。しかし、ALGOL そのものの合理性と習得の容易さのために逐次その利用者が増加し、IBM および二三のものを除く世界中のほとんどの機種に対してこの ALGOL が受け入れられるようになってきている。

改訂 ALGOL-60 は前に述べた Backus notation で記述されているが、その基本となる記号は次の 3 種である。

記 号	意 味
::=	(左辺) とは (右辺) のようなものである
	または
<.....>	というもの (ALGOL 外の言葉)

たとえば

<空> ::=

は空とは何も書いてないものであることを示し、

<名前> ::= <文字> | <名前> <文字> |

<名前> <数字>

は文字および数字というものゝが定義されている状態では名前とは文字かまたは名前の後に文字がきたものか、または名前の後に数字がきたものであることを示している。したがって A も AB も A5 も AN6B もすべて名前であるが、5B や A/B は名前でないことがわかる。

以下この記法で NEW ALGOL-H (H2) の構文法を示す。

ALGOL-60 の構文法と対比し H2 の変更点を●印で示す。

3.1 基本記号 (basic symbols)

<基本記号> ::= <文字> | <数字> | <論理値> |
<区切り記号>

3.1.1 文字・数字・論理値

② <文字> ::= A | B | C | D | E | F | G | H | I | J | K |
L | M | N | O | P | Q | R | S | T | U | V |
W | X | Y | Z

ALGOL-60 には小文字もあるが H2 では大文字に限る (もちろん区切り記号としては小文字も使用できる)。

数字 (digits)

<数字> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

論理値 (logical values)

③ <論理値> ::= TRUE | FALSE

ALGOL-60 では true false である。

3.1.2 区切り記号 (delimiter)

<区切り記号> ::= <作用素> | <区切り> | <括弧類> | <宣言詞> | <特別規制詞>

<作用素> ::= <算術作用素> | <比較作用素> |
<論理作用素> | <順序作用素>

④ <算術作用素> ::= + | - | * | / | + | †

ALGOL-60 では掛算が × であるが大文字のエックスとの混乱を避けるため * を用いる。

<比較作用素> ::= < | ≤ | ≥ | > | ≠

<論理作用素> ::= ≡ | ∨ | ∧ | ¬

⑤ <順序作用素> ::= go to | if | on | then | else | for |
do | code

ALGOL-60 には on code はない。

⑥ <区切り> ::= , | . | : | ; | = | step | until |
while | comment | # | halt

ALGOL-60 ではスペースは # ではなく `␣` で示し、また halt はない。

⑦ <括弧類> ::= () | [] | { } | begin | end | attach
ALGOL-60 には attach はない。

⑧ <宣言詞> ::= Boolean | integer | real | array |
switch | procedure | external

ALGOL-60 には external はない。また own がある。

<特別規制詞> ::= string | label | value

3.1.3 名前 (identifier, I と略)

<名前: I と略> ::= <文字> | <I> | <文字> |
<I> | <数字>

先頭の 6 字だけを判別する。したがって MATR1XA と MATR1XB は区別されない。A12B は良いが、45D は不可

3.1.4 数 (numbers)

<符号のない整数: $\bar{U}I$ と略> ::= <数字> |
< $\bar{U}I$ > | <数字>

<整数> ::= < $\bar{U}I$ > | + < $\bar{U}I$ > | - < $\bar{U}I$ >

<小数> ::= . < $\bar{U}I$ >

<指数部> ::= $10^{<整数>}$

<10 進数> ::= < $\bar{U}I$ > | <小数> | < $\bar{U}I$ > | <小数>

<符号のない数> ::= <10 進数> | <指数部> |

<10 進数> | <指数部>

<数> ::= <符号のない数> | + <符号のない数> |
- <符号のない数>

したがって -5 -4 4.6 10^{-2} $3 \cdot 10^{-2}$ -4.6 10^5 などはすべて数であるが, 34.10^3 $4 \cdot 10^{2.5}$ $49.2 \cdot 10$ などは不可。また取扱う数は整数で12けたまで実数で 10^{-16} までしておく。

3.1.5 記号列 (string, $\bar{S}T$ と略)

<記号の基本列> ::= <'> を除く基本記号の任意の列> | <空>

<開いた記号列> ::= <記号の基本列> | '<開いた記号列>' | '<開いた記号列>' | <開いた記号列>

<記号列> ::= '<開いた記号列>'

3.2 式 (expressions)

<式> ::= < $\bar{A}E$ > | < $\bar{B}\bar{E}$ > | < $\bar{D}\bar{E}$ >

3.2.1 変数 (variables, $\bar{V}$ と略)

<変数名> ::= <I>

< $\bar{S}\bar{V}$ > ::= <変数名>

<添字式> ::= < $\bar{A}E$ >

<添字の並び> ::= <添字式> | <添字の並び> ,
<添字式>

<配列名> ::= <I>

<添字付き変数> ::= <配列名> [<添字の並び>]

<変数> ::= < $\bar{S}\bar{V}$ > | <添字付き変数>

したがって $AB C 45$ などは単純変数 ($\bar{S}\bar{V}$) であり, $P [3] RA [4, K] PQR [I, J, K] STU [if A=B then M else N, J]$ などは添字付き変数である。また添字式が real のときは四捨五入される。

3.2.2 関数呼び出し (function designator)

<手続き名> ::= <I>

< $\bar{A}P$ > ::= < $\bar{S}\bar{T}$ > | <式> | <配列名> |
<スイッチ名> | <手続き名>

⑨ < $\bar{A}P$ の並び> ::= < $\bar{A}P$ > | < $\bar{A}P$ の並び> ,
< $\bar{A}P$ >

< $\bar{A}P$ 部> ::= <空> | (< $\bar{A}P$ の並び>)

<関数呼び出し> ::= <手続き名> (< $\bar{A}P$ 部>)

したがって, $TAN(A) SIMPLEX(N, A, B) RAN$ などは関数呼び出しであり得る。
ALGOL-60 では $\bar{A}P$ リストに , の外にパラメタ

の区切り) <文字> : (を認める。

3.2.3 標準関数 (standard function)

ABS SIGN SIN COS SQRT ARCTAN
LN EXP

ALGOL-60 では小文字である。

●換算関数 (transfer function)

ENTIER FLOAT FIX

ALGOL-60 では entier でそれを越さぬ最大の整数をとる。したがって ENTIER (-8.6) は -9 である。
FLOAT FIX は OLD-H との接続のために残した。
FIX は切りすて。

3.2.4 算術式 (arithmetic expression, $\tilde{A}\tilde{E}$ と略)

<加減作用素> ::= + | -

●<乗除作用素> ::= * | / | +

<1次子> ::= <符号のない数> | <変数> | <関数呼び出し> | <算術式>

<因子> ::= <1次子> | <因子> ↑ <1次子>

<項> ::= <因子> | <項> <乗除作用素>

<因子>

<単純算術式> ::= <項> | <加減作用素> <項> |

<単純算術式> <加減作用素>

<項>

●<条件節> ::= if <論理式> then | on <符号のない数> then

<算術式> ::= <単純算術式> | <条件節> <単純算術式> else <算術式>

I を integer タイプ, R を real タイプとしたとき

$A \uparrow I$ は $I > 0$ なら $A * A * \dots * A$ (I回)

$I = 0$ で $A \neq 0$ なら A と同じタイプの 1

$A = 0$ なら誤り

$I < 0$ で $A \neq 0$ (A: real) なら

$1/(A * \dots * A)$

(A: integer) なら 0

$A \uparrow R$ は

$A > 0$ なら $EXP(R * LN(A))$ で real

$A \leq 0$ なら $0 \uparrow 0 = 1.0$ 以外は誤り

条件は算術式で else の左右辺は同じタイプでなければ誤りとする。またコンパイラの都合で)の直後の文字数字は無視するので $(A+B) * C + D$ を $(A+B) C + D$ と書いたときエラーチェックを行なわないで $(A+B) + C$ の結果を出すので注意する必要がある。

もちろん + の両側は integer に限るが, その他の場合もし混合しておればすべて real の結果となる。

3.2.4 論理式 (Boolean expression, $\tilde{B}\tilde{E}$ と略)

<比較作用素> ::= <| ≤ | = | ≥ | > | ≠

<比較式> ::= < $\tilde{S}\tilde{A}$ > <比較作用素> < $\tilde{S}\tilde{A}$ >

<論理1次子> ::= <論理値> | <変数> | <関数呼び出し> | <比較式> | <論理式>

<論理2次子> ::= <論理1次子> | ¬ <論理1次子>

<論理因子> ::= <論理2次子> | <論理因子> ∧ <論理2次子>

<論理項> ::= <論理因子> | <論理項> ∨ <論理因子>

<内包> ::= <論理項> | <内包> ⊃ <論理項>

<単純論理式> ::= <内包> | <単純論理式> = <内包>

<論理式> ::= <単純論理式> | <条件節> | <単純論理式> else <論理式>

if if if $A=B$ then $C1 \neq D$ else $C2=Q$
then $S>R$ else $S=45$ then $P1 \leq Q1$
else $AB=X48 \wedge (P+Q) * R < 365$

は1つの論理式である。

比較作用素の両側のタイプが異なるときは real として比較される。

3.2.5 行先式 (designational expression, $\tilde{D}\tilde{E}$ と略)

●<名札> ::= <名前>

<スイッチ名> ::= <名前>

<スイッチ> ::= <スイッチ名> [<添字式>]

<単純行先式> ::= <名札> | <スイッチ> | (<行先式>)

<行先式> ::= <単純行先式> | <条件節> <単純行先式> else <行先式>

ALGOL-60 では名札として符号のない整数を認める。

$\tilde{D}\tilde{E}$ は結果として1つの名札に帰着する表現であり

if $P=Q$ then (if $R \neq S$ then $L1$ else $L2$)

else if $P > 300$ then OPEN else

END1

は1つの $\tilde{D}\tilde{E}$ である。

3.3 文 (statement, $\tilde{S}$ と略)

3.3.1 複合文 (compound $\tilde{S}$) とブロック (block)

<名札のない基本文> ::= <代入文> | <飛び越し文> | <空文> | <手続き文>

<基本文> ::= <名札のない基本文> | <名札> : <基本文>

<無条件文> ::= <基本文> | <複合文> | <ブロック>

<文> ::= <無条件文> | <条件文> | <繰り返し文>

<複合後部> ::= <文> end | <文> : <複合後部>

<ブロックの頭> ::= begin <宣言> | <ブロック

の頭> ; <宣言>

<名札のない複合文> ::= begin <複合後部>

<名札のないブロック> ::= <ブロックの頭> ;

<複合後部>

<複合文> ::= <名札のない複合文> | <名札> :

<複合文>

<ブロック> ::= <名札のないブロック> | <L> :

<ブロック>

<プログラム> ::= <名札のないブロック> | <名札のない複合文>

● <ジョブネーム> ::= <L> ①

● <NEW ALGOL-H プログラム>

::= <ジョブネーム> <プログラム> ① |

<ジョブネーム> <アタッチ S> ①

ブロックは

begin D;; D; S;; S end;

複合文は

begin S;; S end;

であるが、プログラムミス (end の後のセミコロンを落すミス) を防止するためには end の前に常に空文をおき、また end の後も常にセミコロンを付加して

begin D;; D; S; S;; end;

begin S;; S; end;

と書く習慣をつけた方がよい。

NEW ALGOL-H のプログラムには常にジョブネームが必要であって、これによってモニタは割込んだ仕事の状態を判別する。①はテーパーマークである。

3.3.2 代入文 (assignment S)

<左辺> ::= <V> := | <手続き名> :=

<左辺の並び> ::= <左辺> | <左辺の並び>

<左辺>

<代入文> ::= <左辺の並び> <A> | <左辺の

並び>

3.3.3 飛び越し文 (go to S)

<飛び越し文> ::= go to <DE>

3.3.4 空文 (dummy S)

<空文> ::= <空>

3.3.5 条件文 (conditional S)

● <条件節> ::= if <論理式> then |

on <符号のない整数> then

<無条件文> ::= <基本文> | <複合文> |

<ブロック>

<部分条件文> ::= <条件節> <無条件文>

<条件文> ::= <部分条件文> | <部分条件文>

else <文> | <条件節> <繰り返し文> |

<名札> : <条件文>

3.3.6 繰り返し文 (for statement)

<繰り返し要素> ::= <算術式> | <算術式> step

<算術式> until <算術式> |

<算術式> while <論理式>

<繰り返し要素の並び> ::= <繰り返し要素> |

<繰り返し要素の並び>, <繰り返し要素>

<繰り返し節> ::= for <変数> := <繰り返し要素の並び> do

<繰り返し文> ::= <繰り返し節> <文> |

<名札> : <繰り返し文>

3.3.7 手続命令 (procedure S)

<AP> ::= <記号列> | <式> | <配列名> |

<スイッチ名>

● <AP の並び> ::= <AP> | <AP の並び>, <AP>

<AP 部> ::= <空> | (<AP 部>)

<手続き文> ::= <手続き名> <AP 部> | <L> :

<手続き命令>

ALGOL-60 では AP リストに、の外に) <文字> : (というパラメタデリミタを認める。

3.4 宣言 (declaration, D と略)

● <D> ::= <型 D> | <配列 D> | <スイッチ D> |

<手続き D> | <イクスターナル D>

3.4.1 型宣言

<型の並び> ::= <SV> | <SV>, <型の並び>

<型> ::= real | integer | Boolean

<型 D> ::= <型> <型の並び>

3.4.2 配列宣言

● <下限> ::= <整数>

● <上限> ::= <整数>

<上下限> ::= <下限> : <上限>

<上下限の並び> ::= <上下限> | <上下限の並び>, <上下限>

<配列片> ::= <配列名> | <上下限の並び> |

<配列名> <配列片>

<配列の並び> ::= <配列片> | <配列の並び>, <配列片>

<配列の宣言> ::= array <配列の並び> | <局所

あるいは占有型> array <配列の並び>

実数配列は array と書いても real array と書いてもよい。例を示すと

array A, B [1:10], PQ [1:5, -8:4];
Boolean array C [0:4];

下限 ≤ 上限でなければならない。

ALGOL-60 では下限および上限に算術式を認めている。

3.4.3 スイッチ宣言

<行先式の並び> ::= <行先式> | <行先式の並び>, <行先式>

<スイッチの宣言> ::= switch <スイッチ名> := <行先式の並び>

スイッチは一種の名札の配列である。例を示すと

switch Q := L1, L2, if A=B then D else LD, RS, S [I];

この場合, L1, L2, D, LD, RS は名札 S [I] はそれ以前に宣言されたスイッチリストの要素である。

3.4.4 手続き宣言

<仮パラメタ> ::= <名前>

<仮パラメタの並び> ::= <仮パラメタ> | <仮パラメタの並び>, <仮パラメタ>

<仮パラメタ部> ::= <空> | (<仮パラメタの並び>)

<名前の並び> ::= <名前> | <名前の並び>, <名前>

<値の部> ::= value <名前の並び> ; | <空>

<規制詞> ::= string | <型> | array | <型>

array | label | switch | procedure |

<型> procedure

<規制部> ::= <空> | <規制詞> <名前の並び>; | <規制部> <規制詞> <名前の並び>;

<手続きの頭> ::= <手続き名> <仮パラメタ部>;

<値の部> <規制部>

● <八進数> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

● <マシン語> ::= <八進数> <八進数> <八進数> <八進数> <八進数> <八進数> <八進数> <八進数> <八進数> <八進数>

● <マシンプログラム> ::= <マシン語> | <マシン語> <マシンプログラム>

● <手続きの本体> ::= <文> | code <マシンプログラム> | <ブロック> | <複合文>

<手続き $\tilde{D}$ > ::= procedure <手続きの頭> <手続きの本体> | <型> procedure <手続きの頭> <手続きの本体>

手続頭部には comment を書けない。

$\tilde{FP}$ リストにあげたものは、すべて規制を要する。

value 指定されたものは値と呼ばれ、その他のものは名前と呼ばれる。 $\tilde{FP}$, $\tilde{AP}$ の対応するものの型、種類は合致しなければならないが value 指定されたときは型が変わってもよい。

array は value 指定できない。

3.4.5 イクスターナル宣言

<イクスターナル $\tilde{D}$ > ::= external < $\tilde{I}$ リスト> | external <型> < $\tilde{I}$ リスト>

次の attach 命令によって機械語となった手続きが格納されているとき、そのジョブネームを頼りに引用することを宣言するもので関数呼び出しのときは external <型> の宣言詞を用いる。

3.4.6 特殊命令

<アタッチ命令> ::= attach <手続 $\tilde{D}$ > ; end

<停止命令> ::= halt ; ⑩

3.4.7 データ

<任意の記号列> ::= <下段を含む任意の字の列>

<データリスト要素> ::= <数> | <論理値> | = <任意の記号列>

<データリスト> ::= <データリスト要素> |

< $\tilde{UI}$ > * <データリスト要素>

<データリスト>, <データリスト要素> |

<データリスト>, <空>

<データ> ::= <データリスト>, ⑩ | <データリスト>; ⑩ | <データリスト> / ⑩

データテープは読込み命令に合わせて数値、論理値または = の後の文字数字を読込むために作成する。

データはすべて、で区切られ、. となったときは 0 が読込まれる。データテープの読込み停止は / ⑩ による。

アルファ読込みで 7 文字以上のときは最後の 7 文字が 7 文字以上のときは以下ブランクが読込まれる。またアルファタイプ印刷で 7 A 以上を指定したときは、最初に 7 文字をこえる分だけブランクがはいり、その後指定されたものが頭から 7 文字出てくる。

3.5 という数値を 6 回続けて読込むときは、6 * 3.5 としてよい。

4° NEW ALGOL-H の入出力文

前述のとおり ALGOL には入出力に関してはなんの規定もない。ALGOL-H には簡単な入出力文が用意されていたが、LP 上にあるパターンで印字させようと考えた時に不便が多かったので、これに変更を加え ISO の勧告を加味して建設省共通の入出力文の仕様とした。

4.1 建設省共通の入出力文

READ(A, B, C, ...);

テープ上の情報を逐次変数 A, B, C に格納する。ただしテープ上の情報と対応する変数は 0 の場合を除いては対応していなければならない。

PRINTR(N, E); $\underbrace{\pm X \cdots X}_{N} \pm X \times 10^{\pm X}$

PRINTI(N, E); $\underbrace{\pm X \cdots X}_{N}$

PRINTF (n_1, n_2, E);
式 E の結果を $\underbrace{\quad}_{n_1} \pm \underbrace{X \cdots X}_{n_2} . \underbrace{X \cdots X}_{n_2}$

の形で印刷準備

PRINTS ('X.....X');

X.....X という字を印刷準備

SPACE(N);

N 字分の空白の印刷準備

CRLF(N);

印刷実行し N 行 feed

で、簡略化されているわりには多彩な印字が可能である。

4.2 NEW ALGOL-H の入出力文

4.2.1 フォーマット スtring (format string)

i) 基本要素 (basic components)

<レプリケータ> ::= <UI> | X
<挿入部> ::= B | <レプリケータ> B | <ST>
<挿入列> ::= <空> | <挿入部> <挿入部> |
 <レプリケータ> (<挿入部>)
<Z> ::= Z | <レプリケータ> Z | <挿入列> C |
 <レプリケータ> Z <挿入列> C
<Z部> ::= <Z> | <Z部> <Z>
<D> ::= D | <レプリケータ> D | D <挿入列> C |
 <レプリケータ> D <挿入列> C
<D部> ::= <D> | <D部> <D>
<符号部> ::= <空> | <挿入部> + | <挿入列> -
<整数部> ::= <Z部> | <D部> | <Z部> <D部>
<T部> ::= <空> | T

D は 1 けたの数値を印刷する (digit)

Z は無効の 0 のとき印刷しない。(したがって・の後や D の後で用いてはならない) (zero suppression)

C はカンマの印刷 (comma), T は打ち切りで D で切れたときは四捨五入されるが, T を最後の D の次に入れたときと切りすてを行なう。+ は符号の印刷, - は負のときのみ符号を印刷する。挿入部で B は空白 (blank) でこのほか任意の文字のつながりを ' ' でくくったストリング (string ST) がみとめられるほか Z, D, B はその前のレプリケータによって繰り返しを命ずることもできる。レプリケータは原則として無符号整数であるが、場合によって REPX によって X にある値を代入しておいて X 回の繰り返しを命ずることもできる。

ii) ナンバーフォーマット

<UI フォーマット> ::= <挿入列> <整数部>
<小数部フォーマット> ::= <挿入列> <D部>
 <T部> | V <挿入列> <D部> <T部>
<指数部フォーマット> ::= 10 <符号部> <UI フォーマット>
 <10進数値フォーマット> ::= <UI フォーマット>

> <T部> | <挿入列> <小数部フォーマット> | <無符号整数フォーマット> <小数部フォーマット>

<ナンバーフォーマット> ::= <符号部> <10進数値フォーマット> | <符号部> <10進数値フォーマット> <指数部フォーマット> | <レプリケータ> (<ナンバーフォーマット>)

iii) プールフォーマット

<F> ::= F | <F> F
<プール部要素> ::= P | <F>
<プール部> ::= <プール部要素> | <レプリケータ> <プール部> | <挿入列> <プール部>

<プールフォーマット> ::= <挿入列> <プール部> | <レプリケータ> (<プール部>)

P および F の印刷様式は次のとおり

	TRUE	FALSE
P 1	0	
F T	F	
2 F TR	FA	
5 F TRUE	FALSE	
8 F TRUE	FALSE	

iv) ノンフォーマット

<ノンフォーマット要素> ::= I | R | L
<ノンフォーマット部> ::= <ノンフォーマット要素> | <レプリケータ> <ノンフォーマット要素>
<ノンフォーマット> ::= <ノンフォーマット部> | <挿入列> <ノンフォーマット部> | <ノンフォーマット> <挿入列> | <レプリケータ> (<ノンフォーマット>)

I と R はフォーマットにしたがいないタイプの異なるときは変換され real タイプに I フォーマットが指定されたときは四捨五入される。

I BB-ZZZZZZZZZZZZZZ D (13けた)
R BB-DDDDDDDDDD $10+DD$ (9けた)
L BBFFFFF

v) ストリングフォーマット

<S> ::= S | <レプリケータ> S | <挿入列>
 <S> | <S> <挿入列>
<S部> ::= <S> | <S部> S

<ストリングフォーマット> ::= <S部> | <レブリケータ> (<ストリングフォーマット>)
 ストリング1字に対応してフォーマット部にSを書く。
 もちろんフォーマット部に挿入列としてストリングを書くこともできる。

vii) アルファ・フォーマット

<A> ::= A | <レブリケータ> A | <挿入列>
 <A>
 <A部> ::= <A> | <A部> <A>
 <アルファフォーマット> ::= <A部> | <レブリケータ> <アルファフォーマット>

アルファタイプの情報は integer の変数に格納されているがこの一字に対応しフォーマット部では A を用いる。

vii) 整列マーク

<整列マーク要素> ::= / | ↑ | ;
 <整列マーク部> ::= <整列マーク要素> | <レブリケータ> <整列マーク要素> | <整列マーク要素> <整列マーク要素>
 <整列マーク> ::= <整列マーク部> | <レブリケータ> (<整列マーク部>);

	ラインプリンタ (LP)	磁気テープ (MT)	高速テープパンチ (HTP)
/	印刷し1行改行	IRG	無視
↑	新しい頁の行頭へ	TM	TM
;	改行しないで印刷	エラー	エラー

viii) フォーマットストリング

<フォーマットアイテム> ::= <ナンバーフォーマット> | <ストリングフォーマット> | <アルファフォーマット> | <ブルーフォーマット> | <ノンフォーマット> | <ST>
 <フォーマットアイテム> <整列マーク> | <整列マーク> <フォーマットアイテム> |
 <フォーマットプライマリ> ::= <フォーマットアイテム> | <挿入列> <整列マーク> | <レブリケータ> (<フォーマットセコンダリ>) | (<フォーマットセコンダリ>)
 <フォーマットセコンダリ> ::= <フォーマットプライマリ> | <フォーマットセコンダリ>
 <フォーマットストリング> ::= ' <フォーマットセコンダリ> '

4.2.2 印刷命令

<印刷リスト要素> ::= <AE> | <BE> | <ST>
 <印刷リスト> ::= <印刷リスト要素> | <印刷リスト>, <印刷リスト要素>
 <印刷リスト部> ::= <空> |, <印刷リスト>
 <印刷命令> ::= PRINT (フォーマットストリング <印刷リスト部>)

したがって real P, Q integer J をノンフォーマットで印刷するなら

PRINT(' 2R, I/, P, Q, J);

となる。

4.2.3 読み込み命令

<読み込みリスト要素> ::= <V> | <配列名>
 <読み込みリスト> ::= <読み込みリスト要素> | <読み込みリスト>, <読み込みリスト要素>
 <読み込み命令> ::= READ(<読み込みリスト>)

したがって array B [1:5] と real A とを B₁ B₂ B₃ B₄ B₅ A の順に書いてあるデータテープ上より読み込むなら

READ(B, A);

でよい。配列宣言と異なった個数の配列読み込みは次の配列読み込み命令による。

4.2.4 配列読み込み命令

<下限式> ::= <AE>
 <上限式> ::= <AE>
 <限界リスト要素> ::= <下限式>, <上限式>
 <限界リスト> ::= <限界リスト要素> | <限界リスト要素>, <限界リスト>
 <配列読み込みリスト> ::= <配列名>, <限界リスト>

<配列読み込み命令> ::= READARRAY (<配列読み込みリスト>)

したがって 20 行 30 列の配列 A に対し A₃₂ A₃₃ A₃₄ A₄₂ A₄₃ A₄₄ とある 2 行 3 列の要素に対応する番号のところに読み込むには

READARRAY(A, 3, 4, 2, 4);

とする。3 次元以上もただ限界リストを拡大すればよい。

4.2.5 パンチ命令

<パンチリスト要素> ::= <AE> | <BE>
 <パンチリスト> ::= <パンチリスト要素> | <パンチリスト>, <パンチリスト要素>
 <パンチリスト部> ::= <空> |, <パンチリスト部>

<パンチ命令> ::= PUNCH (フォーマットストリング <パンチリスト部>)

次の入力データを作るため高速テープパンチに情報を出すもので情報はすべて R, I, L および A タイプで

出される。また情報の終りにはすべてカンマが自動的に入れられる。また A タイプで指定するとテープ上ではたとえば integer K に TOKYO というアルファタイプの情報がはいっているとすれば

PUNCH('5A', K);

よって =TOKYO, とパンチアウトされる。

4.2.6 PUT GET 命令

<名札部> ::= <ŨI>

<PG リスト要素> ::= <AĒ> | <BĒ> |

<配列名>

<PG リスト> ::= <PG リスト要素> |

<PG リスト>, <PG リスト要素>

<PUT 命令> ::= PUT(<名札部>, <PG リスト>)

<GET 命令> ::= GET(<名札部>, <PG リスト>)

同じプログラムの中で一時ある情報を外部メモリに出してその番地を使用し後で持ち帰って計算を続行するときに用いる。目印としては無符号整数を用い同じ番号により大きな配列を PUT すると自動的に前の部分がいっぱいになった後は他の場所に格納する。

配列 A を名札部 3 から全部格納したいなら、

PUT(3, A);

これを同じ次元の配列 B に入れて使用したいなら、

GET(3, B);

である。

4.2.7 MT 入出力命令

<ユニット> ::= <ŨV> | <ŨI>

<MT リスト要素> ::= <AĒ> | <BĒ> |

<ŨT>

<MT リスト> ::= <MT リスト要素> | <MT リスト>, <MT リスト要素>

<MT リスト部> ::= <空> |, <MT リスト>

<MT 書き込み命令> ::= MTWRITE(<ユニット>, <フォーマットストリング>

<MT リスト部>)

<MT 読出し命令> ::= MTREAD(<ユニット>, <フォーマットストリング> <MT リスト部>)

情報を MT に出してとっておき別のプログラムで使用するときに用いる。ユニットは土研の場合 1 か 2 を使用する。これを単純変数に入れておいて指定することもできる。MT 書き込みの前には次の巻きもどし命令によって 20 単位以上消去した後 TM を記入しておく必要がある。たとえば R タイプで配列 P を 1 から N まで書き込むには

REWIND(1, 20);

MTWRITE(1, '↑');

for I:=1 step 1 until N do

MTWRITE(1, 'R', P(I));

とする。

これを別のプログラムの A に代入して使用するには

REWIND(1, 1);

TMSEARCH(1, 1);

for I:=1 step 1 until N do

MTREAD(1, 'R', A(I));

とする。

4.2.8 TMサーチ命令

<TMサーチ命令> ::= TMSEARCH

(<unit>, <整数>)

MT 上の書き込み場所はテープマークをもとにして求める。

↑ が出てくるたびに 1 つずつ TM が書き込まれるので整数のところに 5 を指定すれば 5 番目の TM の次から読み始める。整数には負を指定してよい。このときは巻き送って指定された個数を数え最後に一つだけもどる。

4.2.9 巻きもどし命令

<巻きもどしリスト> ::= <ŨI>

<巻きもどしリスト部> ::= <空> |,

<巻きもどしリスト>

<巻きもどし命令> ::= REWIND(<ユニット>

<巻きもどしリスト部>

リスト部の数の単位だけ消去を行なう。リスト部が空の場合は巻きもどしのみ。

4.2.10 繰り返し数代入命令

<繰り返し数リスト> ::= <AĒ> | <繰り返し数リスト>, <AĒ>

<繰り返し数代入命令> ::= REPX(<繰り返し数リスト>)

たとえば、N 個のブランクの右に K 個を * を打ちたかったら

REPX(N, K);

PRINT('XBK('*')/');

とする。

4.2.11 アルファ代入命令

<アルファ代入命令> ::= EQUIV

(<ストリング>)

TOKYO という字を integer P に代入するときは

P=EQUIV('TOKYO');

である。スペースは * を用い 1 語は 7 文字以下である。

5 各種コンパイラの制限事項

各計算機は共通文法 ALGOL-60 に対してそれぞれ多

少の変更を施している。以下その制限, 変更, 拡張事項を対比してみる。

5.1 OKITAC 5090-H 用 NEW ALGOL-H (H2)

A. 制限事項

- i. 名前には小文字を認めない。また有効長を 6 文字とする。
- ii. own を認めない。
- iii. 名札は名前に限る。(これは JIS ALGOL のままであるが改訂 ALGOL-60 では符号のない整数を認めている。)
- iv. 手続きの仮パラメタはすべて規制を要する。名前変えの仮パラメタの型は一致していなければならない。
- v. 配列宣言の上下限は整数の定数に限る。
- vi. 配列の value 指定を認めない。
- vii. 算術式で括弧の直後の文字数字は無視する。
- viii. 算術式の then の後と else の後の算術式の型は一致しなければならない。
- ix. 手続きの頭には comment を書けない。

B. 拡張事項

- i. アルファモードの代入・読込み・印字を integer タイプの変数に対して許す, 代入は EQUIV(' ') による。
- ii. external 宣言によってすでに中間語として格納されている手続きを呼出しすることを認める。
- iii. REPX によってフォーマット中の繰返しを変数によって制御することを認める。
- iv. on 符号のない整数 (6~10) then 文 (else 文); または then 式 (else 式) という形によりコンソールよりの情報による制御を可能にした。

C. 文法上の注意

- i. code によって機械語との接続を許す。
- ii. 名札は複合文に対しても局所的である。これは制限のように思われるが, 名札を置いたものが複合文の中かブロックの中かをいちいち気にしないでいい点ではかならずしも制限とはならない。
- iii. X の代りに * を用いる。
- iv. 論理定数, 標準関数は大文字とする。

D. 入出力文

READ(A, B, ...); 変数は real integer などであっても A, B, ... の型に合せて自動変換する。

READARRAY(A, 1, N, 1, M); N 行 M 列の array を格納, 何次元でもこのあと続けて書いていい。また宣言したおりの大きさの array A の場合 READ(A); でもよい。

PRINT('フォーマット', A, B, ...);

フォーマットの中では次の記号を許す。

R real の標準フォーマット

I integer の標準フォーマット

L Boolean の標準フォーマット

X レプリケータで 10 コマまでに限る

B ブランク

↑ 改頁

/ 改行

; 改行なしの印字

D 1 けたの情報印字

Z 1 けたの情報印字 (ゼロサプレス)

. 小数点印字

10 10 印字

- 負のときのみ印字

+ 符号を印字

V 小数点の位置指定

S 字を印字

C カンマの印字

A アルファモードの情報印字

F, P Boolean の情報印字 P は 0, 1 で示す。

T 打ち切り

PUT(N, A); GET(N, B);

マグネティックテープのレーベル N (整数型) より配列 A を格納, B に取り出す。

その他 PUNCH, MTREAD, MTWRITE 等

5.2 FACOM 230-50/60/70 用 ALGOL

A. 制限事項

- i. 名前に小文字を認めず有効長を 12 文字とする。
- ii. OWN を認めない。
- iii. 名札は名前に限る。
- iv. 名前変えの仮パラメタは規制されなければ real 扱いとする。値どりの仮パラメタはもちろん規制を要する。
- v. 名前変えの仮パラメタとは対応する実パラメタの型は一致していなければならない。

B. 拡張事項

- i. アルファモードの代入・読込み・印字を integer タイプの変数に対して許す。代入は EQUIV(' ') による。
- ii. EXTERNAL 宣言によって, すでに格納されている手続きを呼出しすることができる。
- iii. 標準関数に LOG, TANH, RANDOM などを追加する。
- iv. + の被演算数が integer でないときは変換してから整数割算を行なう。

C. 文法上の注意

- i. X の代りに *, + の代りに //, ↑ の代りに **, 10 の代りに 1, : の代りに ., ; の代りに ., = の代りに =, ' の代りに " などを用いるほか > 8 等を

'GR' 'AND, begin real' 等を 'BEGIN' 'REAL' 等とする。

- ii. 標準関数論理定数は大文字とする。

D. 入出力文

PEAD(A, B, ...);

PRINT(A);

A のタイプにより -6 D₁₀+DD -10 Z で印字
PRINTSTRING('X.....X');

X.....Xの字を印字

PRINTFORM(≡.....≡);

PRINTFORM(≡.....≡);

SPACE(N);

CRLF;

LFEEED; CRLF10回繰り返し。

EJECT; 改頁

OUTPUT(N, 200, フォーマット, リスト);

Nは偶数 200は機番 フォーマット中の記号はH1
にほぼ同じ その他 PUT, GET 等

5.3 NEAC2200用ALGOL ALEX

A. 制限事項

- i. 名前に小文字を認めずその有効長は6文字とする
- ii. 整数割算÷を認めない。
- iii. own は認めない。
- iv. 名札は名前に限る。
- v. 条件付行先式を認めない。
- vi. 配列宣言の上下限は整数の定数に限る。
- vii. 配列の次元数は4次元に限る。
- viii. スイッチリストは名札に限り、しかも9個以内とする。
- ix. 仮パラメタはすべて規制を必要とする。
- x. 仮パラメタと実パラメタの型種類は一致しなければならない。

B. 拡張事項

なし。

C. 文法上の注意

- i. デリミタはすべて 'BEGIN' のようにするほか、
<A等は 'LT' 'AND' とし 10 は % [(/) は /), ↑ は **, × は * と書く。
- ii. code による ALGOL 以外の言葉との接続を認めない。
- iii. external宣言により、あらかじめ翻訳格納された
手続を呼び出すことを認めない。
- iv. 標準関数・論理定数は大文字とする。

D. 入出力文

JISによる

READ (LAYOUT, LIST);

ただし LAYOUT および LIST は procedure とし

て規定されていなければならない。

READO(フォーマット, A, B, ...);

フォーマット中の記号は H1 のものとほぼ同じ。

PRINT(LAYOUT, LIST);

PRINTO(フォーマット, A, B, ...);

その他 PUT, GET がある。

5.4 TOSBAC 3400-30 用 KT-ALGOL

A. 制限事項

- i. 名前は大文字に限り有効長は8文字とする。
- ii. 数値は real のときは、がかならず必要で 213.
*34 などとしなければならない。(これは FORTRAN
にならったもので 213. という書き方は改訂
ALGOL-60 では誤りである。また 10 のかわりにキ
を用いている。)
- iii. 名札は名前に限る。
- iv. 手続きの仮パラメタはすべて規制を要する。
- v. 'OWN' 'ARRAY' 宣言の上下限は正整数に限る
- vi. 配列の次元数は7次元までに限る。

B. 拡張事項

なし

C. 文法上の注意

- i. × は *, ÷ は %, ↑ は ≡, <V 等は 'LSS' 'OR'
10はキ, [] は '(')', " " は @ ' ' @, begin
integer 等は 'BEGIN' 'INTEGER' と表記する。
- ii. 論理定数は 'TRUE' 'FALSE' とし標準関数は
大文字とする。
- iii. 手続きの本体に KT-TAP 語を書くことができ、
これは 'ABEGIN' 'AEND' でつむ。

なおモニタ TOPS-IV に予定している ALGOL は次
のようなものに拡張される予定である。

- i. own を除く
- ii. 名前は大文字に限り有効長120字
- iii. 仮パラメタはすべて規制必要
- iv. 名札は名前に限る。

5.5 HITAC 5020 用 ALGOL

A. 制限事項

- i. 名前に小文字を認めない。また有効長は9文字
とする。
- ii. 名札は名前に限る。
- iii. 手続きの仮パラメタはすべて規制を要する。
- iv. own array 宣言の上下限は整数の定数に限る。
- v. <type> procedure の宣言の中に手続きの名前
に対する代入が必要である。

B. 拡張事項

- i. 標準関数に SINH, COSH, TANH, LOG,
CBRT (立方根) を追加する。

C. 文法上の注意

- i. 標準関数・論理定数は大文字となる。
- ii. デリミタは"BEGIN" "REAL" などとなるが続く場合は "INTEGER ARRAY" としてもよい。

D. 入出力文

INPUT n (チャンネルナンバ, リスト)

n はリストに書かれた変数の個数でなくてもよい。リストには宣言されたとおりのデータに対応するなら配列の名前だけ書いてもよい。

OUTPUT n (チャンネルナンバ, 'フォーマット', リスト)

n は省略してもよい。フォーマット中の記号は H1 とほぼ同じである。

磁気ドラム入出力として PUTARRAY (レーベル, A); GETARRAY (レーベル, A); がある。

5.6 CDC 3600 用 ALGOL

A. 制限事項

- i. 名前は小文字を認めず有効長は64文字。
- ii. 手続きの再帰的呼出しを認めない。
- iii. array の value 指定を認めない。
- iv. 名前変えの仮パラメタで規制されないものはすべて real とする。
- v. 一番外のブロックで宣言された配列の上下限は定数に限る。

B. 拡張事項

i. OUTPUT のリスト部に
for I=1 step 1 until 100 do A

WRITE のリスト部に
if D=0 then 'PQ' else 'RS'

という表現を認める。

C. 文法上の注意

- i. external 宣言によりソースプログラムの後に入れた手続 (COMPASS で書かれたもの) を使用できる。
- iii. レーベルに符号のない整数を使用できる。

D. 入出力文

READ(A, B, ...);

WRITE('×××.....×');

PRINT(リスト); 6 個までの変数に対し, そのタイプにしたがい印字

INPUT(60, 'フォーマット', リスト);

OUTPUT(61, 'フォーマット', リスト);

5.7 OKITAC 5090 H 用 ALGOL-H (H1)

A. 制限事項

- i. 名前には小文字を認めない。また有効長を 6 文字とする。
- ii. own を認めない。

iii. 名札は名前に限る。

iv. 手続きの仮パラメタはすべて規制を要する。

v. 配列宣言の上下限は整数の定数に限る。

vi. array の value 指定を認めない。

vii. 手続きのパラメタの区切りは , に限る。

viii. 論理作用素で ≡ を認めない。

ix. 繰り返し文 (旧訳 FOR 命令) で条件要素を認めない。つまり while というデリミタを認めない。

x. 特別規制詞の string label を認めない。

xi. 関数呼び出し (旧訳関数指示子) を認めない。

xii. 算術作用素の両側のものは real integer いずれか同じ型でないと誤りとなる。

xiii. 比較作用素の両側のものは同じ型でなければならない。

xiv. 代入文の左辺の型と右辺の型は同じものでなければならない。

xv. 条件付算術式の then の後と else の後は同じ型でなければならない。

xvi. 繰り返し文の段階要素で step の前後 until の後は ↑ を除く算術作用素を含む算術式に限る。

xvii. 繰り返し文の制御変数は単純変数に限る。

xviii. 手続きのパラメタとしての手続きはパラメタなしのものに限る。

xix. 名前変えのパラメタに対応する実パラメタには式を書けない。

xx. スイッチの要素は名札に限る。

xxi. 1 行の印刷に使用する印刷文 (旧訳命令) は 17 個に限る。

B. 拡張事項

なし

C. 文法上の注意

i. 標準関数は大文字 (論理定数は小文字)。

ii. 換算関数は FIX (切り捨て) FLOAT とする。

iii. × の代りに * を用いる。

D. 入出力文

READ(A, B, ...); ただしテープ上の数値の型と対応する変数の型は同じものでなければならない。

PRINT(n, A); A の型に合わせて有効 n けたで印字する。n のところが 3.4 なら小数点以上 3 けた以下 4 けたで印字。

PRINT('×.....×'); ×.....× を印字

CRLF(n);

SPACE(n);

5.8 H1 と H2 との文法比較

文法制限事項中 vii までは H1, H2 共通であるが viii から xx までは H2 においてはすべて取り除かれ, 手続

きの頭に comment を書けない、算術式の括弧直後の文字・数字は無視する、の2項がつけ加わり算術式の then の後と else の後の算術式の型が一致しなければならないという点は H1 の制限の xv の方がもっと強い制限なのであるから、事実上プリント関係を除いては、H2 は H1 を完全に包含していると言っても差つかえない。

したがって H2 でかけることを意識して H1 の文法にしたがって書くことは、次の点に注意すれば完全に可能であるということになる。

1. 論理定数

小文字 (H1) → 大文字 (H2)

2. 印刷文

PRINT (n, A); → PRINTR (n, A); 実数

PRINTI (n, A); 整数

PRINT(n₁, n₂, A); → PRINTF(n₁, n₂, A);

PRINT('×……×'); → PRINTS("×……×");

SPACE, CRLF は同じ

3. ただし、H2 は先頭の begin の前にジョブネームテープマークを入れ、その begin の後に external PRINTR, PRINTI, PRINTF, SPACE, PRINTS, CRLF; を入れると同時に最後の end の後のテープマークを1つにする。

この方法では、たとえば PRINTF (5, B); とか PRINTS("ABC"); などとアンダラインした所だけをキャンセルすれば簡単に H2 から H1 に変換することが可能であるが、修正時にプログラムテープ上の位置をつきとめるのが大変である。

これに対しては印刷に関する部分はすべて procedure にまとめ、H2 から H1 に切り換えるときにテープの頭の方をジョブネーム begin procedure…… ごとと取り換える方が簡単である。

6° NEW ALGOL-II のモニタシステム

6.1 第1次改訂

前述のように土木研究所の電子計算機はいわゆる超大型計算機の衛星計算機としての役割を主とし、研究上の問題点の検討のために常に待機する状態に置き、しかも計算機そのものは相当な稼働率をもって運転するように計画された。そのために当初は光電式紙テープリーダーを3台使用して、予定された計算時間15分以上のプログラムを第1番目の紙テープリーダーより読み込んで計算(今後この計算を第1レベル計算と云う)中に第2番目の紙テープリーダーより計算結果をテストする等の短時間ですむ計算(今後この計算を第2レベル計算と云う)を割り込んで計算し、さらに第3番目の紙テープリーダーより、プログラムのチェックをするためプログラム(今

後この計算を第3レベル計算と云う)を割り込んで処理できるように製作された。(モニタシステム付録No.)

しかし、このモニタシステムは計算結果およびエラーメッセージを1台の高速度印刷装置で印字するためプリント用紙がむだになる上に、後で計算結果をプログラムごとに編集するのに人手を要し利用しにくいものであった。

そこで、(1) 第一回目のモニタシステム改良として第1レベルの計算結果は No. 4 の磁気テープ装置(以下、MT-4 と云う4番目の磁気テープの意味である)に書き込み適当なときに印字する。(2) エラーメッセージは電動タイプライクに印字するという形に直した。これで大分使いやすいモニタシステムとなり約半年間使用していた。

6.2 第2次改訂

その後、プログラム開発が進むにしたがって、計算は長大化し、PUT, GET または、MTWRITE, MTREAD 等を利用する人が増えると5台の磁気テープ装置のNo. 1 (以下 MT-1 と云う)をシステム用に、MT-2, MT-3 をコンパイル用に、MT-4 を第1レベル計算の結果を入れるとあと1台のみとなり、割り込まれたプログラムと割り込んだプログラムのどちらにも磁気テープを使用する命令が書かれてあると、第1レベル計算が犠牲になるか、または磁気テープをかけ直す等、オペレータに負担がかかると同時に事故発生も多く計算機運営上不利であるため、第2回目のモニタシステム改良となった。

磁気テープ装置の利用度を上げるため5台の磁気テープ装置を次に述べる方法で使用した。

(1) MT-1

システムおよびモニタ用で図に示す形式で書き込まれており、改良はしなかった。

B O T	イレース	T M	モニタ プログラム	T M	コンパイル
-------------	------	--------	--------------	--------	-------

(2) MT-2

割り込みの際の待避用ならびにコンパイルの際、中間語のプログラムが書かれ第2計算、第1計算を行なうときはコンパイル用の部分は逐次 BOT 方向につめて書かれる。

B O T	第1 計算待避	T M	第2 計算待避	T M	コンパ イル用
-------------	------------	--------	------------	--------	------------

(3) MT-3

MT-4 を使用していた第1レベル計算の結果と以前から MT-3 を使用していたコンパイルを行なう際

の機械語になったプログラムを同時に書き込むことにした。

先頭に印字のためのプログラムを書いて置きその次から第1レベルの結果を書き込み、割り込みが行なわれると *TM* を書きその後をコンパイル用で使用し第2レベル以上の計算が終わると *TM* を消して第1レベルの結果を書きはじめる。

$\begin{matrix} B \\ O \\ T \end{matrix}$	印字用の プログラム	$\begin{matrix} T \\ M \end{matrix}$	第1レベル 計算の結果	$\begin{matrix} T \\ M \end{matrix}$	コンパ イル用
---	---------------	--------------------------------------	----------------	--------------------------------------	------------

(4) MT-4

MTWRITE 用使用するテープフォーマットはプログラムで指定する。

(5) MT-5

第2レベル計算の中で使用する *PUT*, *GET* 用使用する。

$\begin{matrix} B \\ O \\ T \end{matrix}$	$\begin{matrix} S \\ M \end{matrix}$	レベル	データ	$\begin{matrix} S \\ M \end{matrix}$	レベル	データ	$\begin{matrix} S \\ M \end{matrix}$
---	--------------------------------------	-----	-----	--------------------------------------	-----	-----	--------------------------------------

(6) MT-6

第1レベル計算中の *PUT*, *GET* 用使用するフォーマットは *MT-5* に同じ。

6.3 第3次改訂

この改良で土木研究所用 *ALGOL-H2* モニタシステムは完成されたかと思われたが、割り込みの利用者が増えるにしたがって *MTREAD*, *WRITE* エラーにより、第1レベル計算が計算途中で実行できなくなり、また初めからやり直す等の事故が頻発した。これはピンチローラ方式の *MT* ではやむを得ないことではあるが、ときには朝から始めた数時間の計算が夕方になっても終了しないこともあり、第1レベル計算に対する不信感が広がるに至った。

これに対処する方法として、いま使用しているピンチローラ型磁気テープを安定度の高いシングルキャブスタ型磁気テープに換え、また、増大する高速補助記憶容量に対する要望もあり他の補助記憶装置（ディスクまたはドラム）を付けることを検討したが、*OKITAC 5090-H* システムとしては良い手の打ちようがなかった。

結局、(1) *MTWRITE* のチェックレベルを上げて書けたものは絶対に読めるようにする。(2) モニタシステムを改良して *BOT* の次にイレーズ部分を増す。(3) *MTWRITE* の際エラーが発生したときには無理をして書かず第1レベル計算にもどす。(4) 第3レベル計算の利用者が少なく、第2レベル計算を10分間に規制

すれば第3レベル計算は必要ないのでやめる。

(5) 第2レベル計算のプログラムのミスのためモニタプログラムまでこわしてしまい *MTREAD* が行なわれないときにはシステムロードを *PTR2* にセットし、ブートストラップを 10300 番地にセットし 140 番地を 42300250000003 におし、モニタを読みなおした後 747 番地よりスタートすれば第1レベル計算にもどる。

(付録1-1モニターシステム)

以上の方法により現在大きなトラブルもなく、安定した運転が続いているが、この磁気テープ装置の不安定さならびに増大する高速補助記憶装置に対する要求から *TOSBAC 3400-40* への機種変更となった。

計算機およびモニタの安定度が高まり一日中連続運転を行なうようになると *ALGOL-H1* による計算は行なえなくなったことに対する不満を解決するため、*H1* のシステムおよび *H2* のモニタを改良して第2レベル計算用に *H1 ALGOL* という external procedure を作った。

```
begin
  external H1 ALGOL ;
  H1 ALGOL ;
end ;
```

このプログラムにより第1レベル計算および *H2* モニタを *MT2* に待避し、*H1* のシステムが計算機に読み込まれ、*H1* のプログラムを行ない、計算が終われば自動的にもとの *H2* の計算にもどるようにし、現在その安定性の実験を行なっている。

1. *H1* システムの改良点

- (1) システムテープを *MT-3* から *MT-1* にし *MT-3* の仕事を *MT-1* に変更する。
- (2) 1 の変更によるエラー処理プログラム改良。
- (3) 計算中にエラーが発生したときのジャンプ先を次頁に掲げる表のようにした。

2. *H2* モニタの改良点

- (1) *H2* のシステムプログラムは external procedure により長さが一定で無いため下図のような方法で *MT-1* に *H1*, *H2* 共通システムテープを作った。

$\begin{matrix} B \\ O \\ T \end{matrix}$	$\begin{matrix} T \\ M \end{matrix}$	$\begin{matrix} H1 \\ システム \end{matrix}$	$\begin{matrix} T \\ M \end{matrix}$	$\begin{matrix} T \\ M \end{matrix}$	$\begin{matrix} H2 \\ モニタ \end{matrix}$	$\begin{matrix} T \\ M \end{matrix}$	$\begin{matrix} H2 \\ システム \end{matrix}$
---	--------------------------------------	--	--------------------------------------	--------------------------------------	---	--------------------------------------	--

- (2) 第1レベル計算に第2レベル計算が割り込む際 *MT-2* にプログラムが待避されるが *H1* の計算を行なう際はその後モニタ部分も待避し *H1* システムを主記憶装置に出して *H1* プログラムの処理をはじめる。

ERROR No.	STOP したときの SCC	ERROR の 内 容	REFERENCE
1	37661	READ で DATA の TYPE が異なる	
2	37662	$e^x > 38$ ($x > 88$)	$AR_{00} = x$
3	37663	$LN(x)$ で $x \leq 0$	$AR_{00} = x$
4	37664	$SQRT(x)$ で $x < 0$	$AR_{00} = x$
5	37665	$R_1 \uparrow R_2$ で ($R_1 = 0, R_2 < 0$) ($R_1, R_2 = \text{real}$)	$AR_{00} = R_1$ $AR_{01} = R_2$
6	37666	$R_1 \uparrow R_2$ で ($R_1 < 0, R_2 \neq 0$) ($R_1, R_2 = \text{real}$)	$AR_{00} = R_1$ $AR_{01} = R_2$
7	37667	$I_1 \uparrow I_2$ で ($I_2 < 0$) ($I_1, I_2 = \text{integer}$)	$AR_{00} = I_1$ $AR_{01} = I_2$
8	37670	SWITCH の添字 ≤ 0	$AR_{00} = \text{添字}$
9	37671	SWITCH の添字 > 要素数	$AR_{00} = \text{添字}$
10	37672	MT UNIT-2 READ ERROR (DATA READ のとき)	

B O T	イレース	第1レベル プログラム	T M	モニタ プログラム	T M	コンパイル用
-------------	------	----------------	--------	--------------	--------	--------

(3) MT-1 および 2 の読み書きを行ったり H1 計算の最終処理などを行なうための補助プログラムを 37600 番地以後に書いておくプログラムの開発。

6.4 新システムの計画

以上が H2 モニタシステムの開発改良についてのあらましであるが、機種変更後もこのような割り込みを主体とする方式で行なう予定である。

なお次の新機種に対応したシステムとしてはアルゴルの中間語および FORTRAN をアルゴルプログラム中に挿入できるようにした、いっそう利用しやすいモニターシステムとする予定である。

最後に現在のモニターシステムに付随した補助プログラム、external procedure および各研究室で開発された実用プログラムは次のとおりである。(付録 2 に簡単な機械語の説明を示す)

1. 補助プログラムおよび external procedure

1-1 SYSTEM COPY

H1 用, H2 用どちらも磁気テープに書かれているシステムテープを複製するプログラム。

1-2 END DISPLAY

MT-3 の計算結果の最後に最終符号を書いてその内容を印字する。(付録 1-2)

1-3 MT DISPLAY READY

MT-3 の先頭に磁気テープの内容を印字するためのプログラムを書き込む。(付録 1-3)

1-4 ACOS(X)

$\cos^{-1} X$ を求める。

1-5 ARCTAN(A)

$\arctan A$ を求める。

1-6 HCOS(X)

$\cosh X$ を求める。

1-7 HSIN(X)

$\sinh X$ を求める。

1-8 ASIN(X)

$\sin^{-1} X$ を求める。

1-9 COVER

PTR-1 よりのプログラムに表紙を作る。

題名は 119 字までをデータテープの先頭にパンチし、終わりに TM をパンチしておく。

1-10 PRINTF(N, M)

式 M の内容を小数点以上 n_1 けた、以下 n_2 けたで印刷

$$\underbrace{\square \pm X \cdots X}_{n_1} \cdot \underbrace{X \cdots X}_{n_2} \square \quad (\text{付録 1-4})$$

1-11 PRINTI(N, M)

整数タイプの印刷を行なう。

$$\square \pm X \cdots X \square \quad (N+3 \text{ けた})$$

N けた

1-12 PRINTR(N, M)

実数タイプの印刷を行なう。

$$\underbrace{\square \pm \cdot \times \times \cdots \times \times}_{N \text{ けた}} \pm \times \times \square \quad (N+7)$$

1-13 PRINTS(S)

ストリングの内容をそのまま印刷。

1-14 SPACE(N)

印字するときブランクを入れる。

1-15 PRINTA(M)

アルファモードを7文字印刷。(スペースなし)

1-16 PRISIGN(M)

LP に $M < 0$ のとき ' ' を印字する。

1-17 PRISA(M)

アルファモードの印刷を行なう。(スペースなし)

1-18 PRISF(N, M)

小数点型実数を印字する。(スペースなし)

$$\underbrace{\pm \times \times \cdots \times \times}_{n1} \cdot \underbrace{\times \cdots \times \times}_{n2}$$

1-19 PRISI(N, M)

整数タイプの印字 (スペースなし)

$$\underbrace{\pm \times \cdots \times \times}_N$$

1-20 PRISR(N, M)

指数型実数の印字 (スペースなし)

$$\pm \cdot \underbrace{\times \cdots \times \times}_{N} \times \pm \times \times$$

1-21 RENRITSU(A, C, X, N, CASE)

連立方程式を解き、逆行列を求める。

1-22 CRLF(N)

N 回復帰改行を行なう。

1-23 NEW PAGE

LP 用紙を一枚おくる。

1-24 UN POWR (PN, PEPS, EV, RAM, WV, TV, ITV, RAT, TEV)

n 次の (正方) 対称行列 A の固有値および固有ベクトルを求める。

1-25 ZERO STOP (L1)

on 6 で計算を一時停止。

on 7 で L1 に飛び越す。

1-26 IA1PUT(NAME, N, P)

ベクトル NAME の1から N までに P を代入する。(整数型)

1-27 IA2PUT (NAM, M, N, P)

マトリクス NAME の M 行 N 列のすべての要素に P を代入する。(整数型)

1-28 RA1PUT(NAME, N, P)

ベクトル NAME の1から N までに P を代

入する。(実数型)

1-29 RA2PUT(NAM, M, N, P)

マトリクス NAME の M 行 N 列のすべての要素に P を代入する。(実数型)

1-30 PRIA2(S, M, N, P, A)

2次元 array の印字 (指数型, 実数)

1-31 PRIA1(N, A)

1次元 integer array の印字を行なう。

1-32 PRIA2 (M, N, A)

2次元 integer array の印字を行なう。

1-33 PRIA3(L, M, N, A)

3次元 integer array の印字を行なう。

($L \times M \times N$ の array)

1-34 PRIRA1(N, A)

1次元 array の印字を行なう。

1-35 PRIRA2(M, N, A)

2次元 array の印字を行なう。

1-36 TRAN21(P, L, M, N, M2, V1)

MATRIX のある行かまたは列をベクトルに格納する。

1-37 TRANA1(L, M, N, A, B)

ベクトル A の L 要素より N 個をベクトル B の M 要素より N 個の部分に移す。

1-38 DISPLAY

PTR1 よりの計算が終わると自動的に LP に印字を始める。

1-39 GSWPOUT(PN, PA, PB, PX, ZER)

連立一次方程式の根を Gauss Sweepout 法で求める。

1-40 INNER PRODUCT(N, X, Y, XY)

ベクトルの内積を求める。

1-41 INVERSE MATRIX(M, A)

(m, m) 型行列 A の逆行列 A^{-1} を求める。

1-42 DETERM(A, N)

行列式 A の値を求める。

1-43 VEC SUB(N, A, B, C)

ベクトル A, B の差を C に代入。

1-44 VEC SUM(N, A, B, C)

ベクトル A, B の和を C に求む。

1-45 TRANSPSED MATRIX (M, N, A, B)

マトリクス A の転置行列を B に格納する。

1-46 MAT SUB (M, N, A, B, C)

マトリクス C に $A-B$ を格納する。

1-47 MAT SUM (M, N, A, B, C)

マトリクス C に $A+B$ を格納する。

1-48 MAMIRA (P, N, A, MAMI, M)

1次元 array の要素の最大 (最小) を求める。

1-49 *MAT PRODUCT* (L, M, N, A, B, C)
行列 A (l, m) と B (m, n) の積を C (l, n) に格納する。

1-50 *JACOBI* (A, S, EVE, N, RHO)
 n 次実対称行列 A の固有値および固有ベクトルを求める (*JACOBI* 法)

1-51 *LEGAB* ($PN, PA, PB, X1, ZER$)
連立 1 次方程式の根を Gauss の基本法で求める。
1-52 *PLOT* ($X, Y, RITU, PO, S, C$)

データをグラフ形式に plot する。(付録 1-5)

1-53 *POWER* ($PN, PEPS, A, EV, RAM, WV, TV, ITV$)
 n 次の (正方) 対称行列 A の固有値および固有ベクトルを求める。

1-54 *END DISPLAY*
PTR 1 より計算した結果を印字するとき使用する。

1-55 *EQUAT* 3 (AA, BB, CC, XA)
3 次方程式の 3 つの根を求める標準手続きである。

1-56 *MREG* 1
2 次仮説による有意因子の選定 (付録 1-6)

1-57 *MAMIIA* ($P, N, A, MAMI, h$)
1 次元 integer array の最大 (最小) 値を求める。

1-58 *PNP* (M)
図化機制御用テープを作成
 M を 10 倍し四捨五入を行ない、8 進数にして *HTP* より次の形にパンチする。

$\pm \square \square \square \square \square$, (付録 1-7)

1-59 *PUP, PDP, PEP, PRP, PLP*
図化機制御用テープに $=U, =D, =F, =R, =L$ を *HTP* よりパンチする。

1-60 *HTP SET*
HTP のモータを運転させ紙テープを繰り出す。

1-61 *HTP STOP*
HTP の紙テープを繰り出し、*TM* をパンチした後繰り出しを行なう。

1-62 *CPUTI* (N, I) *CPUTR* (N, R)
コアの最後から integer 形の数を格納 *CGETI* で取り出し、次のプログラムでも使用できる。

1-63 *CGETI* (N, I) *CGETR* (N, R)
CPUTI で格納した数の読み出しに使用する。

1-64 *OCTAL* (I)
10 進数 10 倍し四捨五入後 8 進数に変換する。

1-65 *PREADY* (X, Y)
図化機に座標系を設定する。

いまペン先のある位置の座標を (X, Y) にする。

1-66 *PSENA* (A, L)

図化機で直線を書くテープをさん孔する。
ペン先のある位置から方向角 A (度) 長さ L の所まで直線を書く。

1-67 *PMOVE* (X, Y)

図化機のペン先を上げ、座標 (X, Y) まで移動させる。

1-68 *PSEN* ($X1, Y1, X2, Y2$)

図化機で直線を書くテープをさん孔する。
始点 ($X1, Y1$) から終点 ($X2, Y2$) へ直線を書く。

ペン先のある位置が ($X1, Y1$) でない時は移動して書き始める。(付録 1-8)

1-69 *PSENX* ($X2, Y2$)

図化機で直線を書くテープをさん孔する。
ペン先のある位置から ($X2, Y2$) へ直線を書く。

1-70 *PEN* ($X1, Y1, X2, Y2, R$)

図化機で円弧を書くテープをさん孔する。
始点 ($X1, Y1$) から終点 ($X2, Y2$) へ半径 R で $R > 0$ のとき右廻り, $R < 0$ のとき左廻りで円弧を書く。

ペン先のある位置が ($X1, Y1$) でないときは移動して書き始める。(付録 1-9)

1-71 *PENX* ($X2, Y2, R$)

図化機で円弧を書くテープをさん孔する。
いま、ペン先のある位置を始点とし、($X2, Y2$) へ半径 R で $R > 0$ のとき右廻り, $R < 0$ のとき左廻りで円弧を書く。

1-72 *PEND*

図化機で終了時にペン先を $Y=0$ にもどし X を 100 ミリおくる。

1-73 *POUTI* (A, S, N, I)

図化機で $\pm$ の値を整数タイプで書かせる。

1-74 *POUTF* (A, S, N, F)

図化機で F の値を小数点, 実数タイプで書かせる。

1-75 *POUTR* (A, S, N, R)

図化機で R の値を指数型実数で書かせる。

(付録 1-10)

1-76 *H1 ALGOL*

$H2$ モニタのもとで $H1$ プログラムで計算を行なうときに使用する。3760 番地 (8 進に $H1, H2$ 共通 *SYSTEM* 用モニタを読み込んでから使用する。

1-77 TAPE MONITOR

紙テープの内容を印字する。

1-78 HTP REPRODUCER & TAPE

COMPARE

紙テープの複製を作り、つき合わせを行なう。

7° 入力媒体の問題

7.1 紙テープとカードの対比

土木研究所において使用言語を ALGOL にするか、FORTRAN にするかを論議した際、ALGOL は紙テープベースであり FORTRAN がカードベースであることも論議の対象となった。

1. 処理速度

カードは900枚/分 1枚80字として72000字/分
テープは1000字/秒 したがって 60000字/分
ただしカードの場合80欄の最後まで情報が読めることは稀れて FORTRAN の1つの文で1枚、1つのデータで1枚であれば1/3 かせいぜい1/2 詰まっているとすればテープでむだな送りのはいっている割合を20%と見てもカード 36000字/分 テープ 48000字/分 である。

2. 費用

カードは1枚60銭で80字に対し1字当り0.0075円、テープは1巻300円で1巻270m、1cm当り4字の情報がいっているとして1字当り0.0028円、前の0.5と0.8の効率を考えるとカードは0.015円 テープは0.0035円でカードが約4倍、保管費はカードが少なくとも3倍以上の容積と、さらに恒温恒湿の条件を必要とする問題がある。オンライン入力機器で考えるとカードはレンタルで月25万円程度、紙テープは2〜3万円でカードの方が約10倍につく。

3. 修正の容易さ

カードはその部分を取り替え、テープはテープスプライザによって修正テープをつなぎ込み得るので同じことである。最近の IBM カードは上部にその内容が印字されるが、紙テープも10字位ずれた所で中央の送り孔の部分に印字できる。

4. 決裁書類兼用の場合

カードはもし書類形式を変更し得れば決裁書類として兼用できる。しかし会計法規に縛られた官庁においては実行不可能であろう。

5. 既存データの利用

すでにカードに作ってあるデータの場合はカード入力があり利となる。しかし何処かで磁気テープ (MT) への変換を行なってくれば、単にこのようなデータ利用の目的だけにカード入力装置を各所に備える必要はない。また MT に移してあれば、後の処理は極めて早い。(ただし、このためには MT の互換性が保持されている必

要がある)。

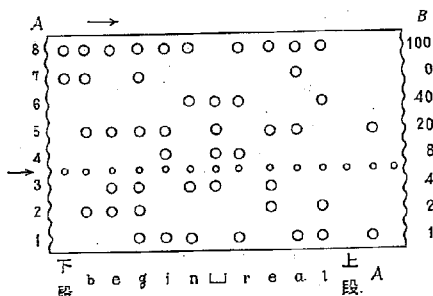
6. データパンチ外注の難易

カードならば PCS の普及によりどこでも引受けるが、特定のコードの紙テープの場合は容易でない。しかし、コード変換を計算機が行なうので、特定のコードに限る必要もないし、今後東京電力、生命保険会社等のように紙テープベースで操業する個所が増加することを考えると外注もそれほどむずかしくないものと考えられる。

7.2 紙テープ

以上のような観点と ALGOL 習得の容易さなどから、土木研究所においては紙テープベースの ALGOL によって業務処理を行なう方針とし、テープの進行方向判別の便、小文字の並びが第8ビットの存在によって容易に判別され begin end real 等の箇所がつきとめやすい、ALGOL 用の活字が豊富である等の理由により、オキタイバによる8単位テープを採用した。

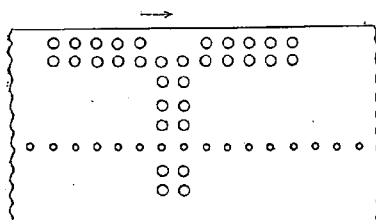
この場合、土研テープ用の先頭マーク、紙テープよりの読み取り並びにハンドパンチによるさん孔作業の便のために、次の逆引字典を作り、これを各研究室、各プログラマ席に配布した。



エラーチェック等のため孔から読み取る時は右側の B マークによって読み、たとえば一番左は 0+100=100 で表の 100 から下段コードと読み、これから後上段コード (矢印の孔のみのもの) がくるまではすべて下段であることを了解し次の 2+20+0+100=122 によって 6 以下 begin スペース real と読み上段コードの次は 1+20=21 で大文字の A と了解する。

またテープスプライザや補修紙を用いて字を訂正するとき n という字が必要ならば A コードによって 1368 と承知してこの位置にパンチすればよい。

土木研究所においては、土研のテープであることを示すために begin の前に下段 5 復帰改行下段 5 のマークを入れ、



というパターンによって土研テープの先頭であること
を容易にわからせるようにしている。

なおオキタイパに関しては下段コードによってタイプ

バーの上下を行なうため、下段コードなしに第8ビット
がきても、あるいは上段コードなしに第8ビットのあつ
た状態からない状態に移っても、エラー検出されないた
め、印字面と内容との食い違いが生じ、印字面から見た
エラーの検出が困難であった。

これに対して土木研究所においてオキタイパの改造点
を発見し(8章)、この問題を解決することができた。

8° 電子計算機の機械的改良点

8.1 電動タイプライタ

ALGOLで書かれたプログラムは大文字、小文字が混
在しているため沖電気製5091電動タイプライタを使用し

上 段	B	A
1	— 1 —	1.7
2	— 2 —	2.7
3	— 3 —	1.2
4	— 4 —	3.7
5	— 5 —	1.3
6	— 6 —	2.3
7	— 7 —	1.2,3.7
8	— 8 —	4.7
9	— 9 —	1.4
=	— 10 —	2.4
10	— 11 —	1.2,4.7
+	— 20 —	5.7
A	— 21 —	1.5
B	— 22 —	2.5
C	— 23 —	1.2,5.7
D	— 24 —	3.5
E	— 25 —	1.3,5.7
F	— 26 —	2.3,5.7
G	— 27 —	1.2,3.5
H	— 28 —	4.5
I	— 29 —	1.4,5.7
.	— 30 —	2.4,5.7
)	— 31 —	1.2,4.5
—	— 40 —	6.7
J	— 41 —	1.6
K	— 42 —	2.6
L	— 43 —	1.2,6.7
M	— 44 —	3.6
N	— 45 —	1.3,6.7
O	— 46 —	2.3,6.7
P	— 47 —	1.2,3.6
Q	— 48 —	4.6
R	— 49 —	1.4,6.7
;	— 50 —	2.4,6.7
*	— 51 —	1.2,4.6
0	— 60 —	5.6
/	— 61 —	1.5,6.7
S	— 62 —	2.5,6.7
T	— 63 —	1.2,5.6
U	— 64 —	3.5,6.7
V	— 65 —	1.3,5.6
W	— 66 —	2.3,5.6
X	— 67 —	1.2,3,5,6.7
Y	— 68 —	4.5,6.7
Z	— 69 —	1.4,5.6
(	— 70 —	2.4,5.6
(	— 71 —	1.2,4,5,6.7
● SP	— 72 —	3.4,5.6
● CR	— 74 —	2.3,4,5,6.7
Tm	— 14 —	2.3,4.7

下 段	B	A
<	— 101 —	1.8
>	— 102 —	2.8
:	— 103 —	1.2,7.8
M	— 104 —	3.8
M	— 105 —	1.3,7.8
M	— 106 —	2.3,7.8
—	— 107 —	1.2,3.8
—	— 108 —	4.8
#	— 109 —	1.4,7.8
.	— 110 —	2.4,7.8
.	— 111 —	1.2,4.8
↑	— 120 —	5.8
a	— 121 —	1.5,7.8
b	— 122 —	2.5,7.8
c	— 123 —	1.2,5.8
d	— 124 —	3.5,7.8
e	— 125 —	1.3,5.8
f	— 126 —	2.3,5.8
g	— 127 —	1.2,3,5,7.8
h	— 128 —	4.5,7.8
i	— 129 —	1.4,5.8
≡	— 130 —	2.4,5.8
×	— 131 —	1.2,4,5,7.8
÷	— 140 —	6.8
j	— 141 —	1.6,7.8
k	— 142 —	2.6,7.8
l	— 143 —	1.2,6.8
m	— 144 —	3.6,7.8
n	— 145 —	1.3,6.8
o	— 146 —	2.3,6.8
p	— 147 —	1.2,3,6,7.8
q	— 148 —	4.6,7.8
r	— 149 —	1.4,6.8
√	— 150 —	2.4,6.8
^	— 151 —	1.2,4,6,7.8
!	— 160 —	5.6,7.8
Σ	— 161 —	1.5,6.8
s	— 162 —	2.5,6.8
t	— 163 —	1.2,5,6,7.8
u	— 164 —	3.5,6.8
v	— 165 —	1.3,5,6,7.8
w	— 166 —	2.3,5,6,7.8
x	— 167 —	1.2,3,5,6.8
y	— 168 —	4.5,6.8
z	— 169 —	1.4,5,6,7.8
—	— 170 —	2.4,5,6,7.8
—	— 171 —	1.2,4,5,6.8
● =	— 175 —	1.2,3,4,5,6,7.8

●印は上下段共通

てパンチまたは紙テープの修正を行なう際、紙テープと印字面との上、下段(大文字、小文字)が異なつてシフトエラーとなることが多い。

相当熟練したパンチャであっても長いプログラムでは数回のシフトエラーが発生する。まして土木研究所のように専任のパンチャを置かず、研究員が自分のプログラムを自分でパンチする場合にはタイプライタの欠陥によるシフトエラーの発生は、いっそう多くなり、計算機の運営に障害をあたえることが多かった。

そこで、シフトエラーが発生した際に印字さん孔を停止するように電動タイプライタの改良を行なった。次に改良点を列記する。

- (1) シフトエラー表示ランプの取付け
- (2) 鍵盤部シフト指示ランプの取付け
- (3) シフトエラーの際は印字さん孔ともに行なわずキョロックされる。
- (4) 電源投入時にキョロックされており解放するには上・下段キーによって行なう。

8.2 停止警報装置

モニタシステム充実に、計算機はほとんど人手を要せず自動的に運転が行なわれるようになったが、高速度印刷装置の用紙、リボンまた高速度テープさん孔機用の紙、テープが無くなったときには人間が取りかえる以外に方法がなく、そのためにオペレータを常時コンソールの前に張りつけて置くことは勤務時間中はもちろんであるが、特に夜間長時間を要する計算を行なう際には特に不経済となる。

一方、無人運転を行なえば計算機がいつの間にか停止して計算機の利用率を著しく低下させる。以上の点を解決するには、計算機が停止したことを早く多くの関係者に知らせる装置を開発した。その仕様は次の通りである。

- (1) 計算機本体に絶対に影響のないように計算機結合部の電源は乾電池を使用し、入力インピーダンスはできるだけ高くする。
- (2) リレーおよびブザー用電源と電子計算機用電源とは完全に分離し、アースも共用しない。
- (3) 計算機が停止するとブザーが鳴りリセットボタンを押すか、または計算機がスタートするまで鳴り続ける。
- (4) ブザーおよびリセットボタンは数カ所に設置できる。

この装置を取付けたことによりオペレータは計算機室、プログラム室、事務室で他の仕事をしながら待期し、ブザーが鳴ると最寄りのリセットスイッチを押してから、その処置を行なえばよいことになり、人間、機械共に能率が倍加された。

8.3 運転時間計

計算機のレンタル会社で通電時間計はセットしてあるが土木研究所側から見て計算可能時間および実計算時間等のデータを知るため、停止警報機のリレーを利用して計算可能時間計および実計算時間計を取付けた。

朝の計算機チェックがすみ計算機を引渡された時に停止警報機のスイッチを入れると計算可能時間計に通電され、また計算機が停止している以外は実計算時間計に通電されるようにしたため、計算機の利用時間のデータを正確に知ることができ、次期計画等に役立つと思われる。

8.4 計算機自動停止装置

上記の装置およびモニタシステムの充実に、計算機はほぼ自動運転を行なえるようになったが、事故または完全に計算が終わった際に電源を切る仕事はオペレータが行なっている。

これは入出力機器、計算機本体、冷暖房器、電灯等、数多くのスイッチがある上に、その停止順序が決まっており、これを自動化することはかなりむずかしい。また完全無人運転を行なうには回転機器の加熱による火災に対する処置も行なわなくてはならない。

以上の仕事を自動的に行なう装置およびモニタシステムを昭和42年度中に開発実験し、実用の段階までにする予定である。

9° 自動化

9.1 図化機の概要

昭和42年3月より電子計算室において図化機が稼働を開始した。プロッタは岩崎通信機製、コントローラは東洋電機製で、幅250mmの用紙に400ステップ/秒、1ステップ0.1mmで精度0.2mmのプロットを行なうものである。

本機を用いてアルゴリズムプログラム中においてHTP(高速度テープパンチ機(毎秒200字))により図化機をコントロールするための紙テープを作り出すもので、図化機の基本命令としては次のものがある。

- = U ペンを上げよ
- = D ペンを下げよ
- = F 次の X および Y 座標点まで直線点に移動せよ。ただし数値は0.1ミリ単位の8進数5けたの数字とする。
- = R 次の新座標系の X および Y 座標点まで新座標系の原点を中心として右廻り円弧を描け、数字は F の場合と同じ。
- = L 次の新座標系の X および Y 座標点まで、新座標系の原点を中心として左廻り円弧を描け、数値は F の場合と同じ。
- = H 停止

これらはカンマでつながれ、たとえばペンを上げて
 $X=97.2$ ミリ $Y=67.5$ ミリに移動し、ここから、(79,
 84.7) に対して直線を引くのは

$=U, 1714, 1243, 1714, 1252, =D, =F, 0.0,$
 $1426, 1517,$

という字が紙テープ上に並んでおればよい。円弧の場合
 は単なる 8 進 10 進の変換に止どまらず、次の旧座標系
 の X, Y を指定された半径で右廻り、または左廻り円弧
 を画くための中心点を計算し、その中心点を原点とする
 新座標系への変換を行なう必要があるのでは複雑に
 なる。このために ALGOL でこれらを簡単に処理する
 システムを開発した。

9.2 ALGOL による図化手続き

前に述べたように図化機のコントロールテープ作成に
 は多少の困難が伴うので、これらをすべて計算機によっ
 て遂行させるために 3~5 月にわたって次の諸手続きを
 準備した。(数値はすべてミリ単位の 10 進数である)

PREADY($X1, Y1$); 作図準備

現在ペンのある位置が新座標系で ($X1, Y1$) と
 なるように、新座標系の原点を設定せよ。したがっ
 てもし現在ペンのある位置を原点に取りたい場合は
 (そして、それが普通の場合であるが) **PREADY**
 (0, 0); とする。

PEND; 作図終了

ペンを $Y=0$ の位置に戻し X を 100 ミリ送る。

PSEN ($X1, Y1, X2, Y2$); 直線 (1)

もし現在ペンのある位置の座標が ($X1, Y1$) な
 ら直ちに、($X1, Y1$) でなければそこまでペンを上
 げて移動した後に、その点から ($X2, Y2$) まで直
 線を引け。

PEN($X1, Y1, X2, Y2, R$); 円 (1)

もし現在のペンの位置の座標が ($X1, Y1$) なら
 ば直ちに、($X1, Y1$) でなければそこまでペンを
 上げて移動した後に、その点から ($X2, Y2$) まで
 半径 R の円を、 R が正なら右廻り (時計廻り) で、
 負なら左廻り (反時計廻り) で画け。

PMOVE($X2, Y2$); ペン移動

現在のペンの位置から ($X2, Y2$) までペンを上
 げて移動せよ。

PSENX($X2, Y2$); 直線 (2)

現在のペンの位置から ($X2, Y3$) まで直線を引
 け、

PSENA(A, L); 直線 (3)

現在のペンの位置から A 度の方向に長さ L ミ
 リの直線を引け、ただし A は X 方向を 0 度とし
 た小数点位置を度とする量で ± 180 度に限りず何
 度でもよい。

PENX($X2, Y2, R$); 円 (2)

現在のペンの位置から半径 R の円を、 R が正な
 ら右廻り (時計廻り)、負なら左廻り (反時計廻り)
 で画け。

POUTF(A, S, n_1, n_2, E);

式 E の値を角度 A の方向に大きさ S ミリの字
 で次の形に書け

$\underbrace{\text{ } \pm X \cdots X}_{n_1} \cdot \underbrace{X \cdots X}_{n_2} \text{ } \sqcap$

POUTR(A, S, N, E);

式 E の値を方向 A 度、字の大きさ S ミリで次
 の形に書け

$\underbrace{\text{ } \pm X \cdots X}_{N} \cdot \underbrace{\pm X X}_{2} \text{ } \sqcap$

POUTI(A, S, N, E);

式 E の値を方向 A 度、字の大きさ S ミリで、
 次の形に書け

$\underbrace{\text{ } \pm X \cdots X}_{N} \text{ } \sqcap$

ALPA(A, S);

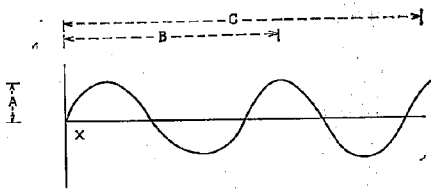
A という字を方向角 A 度大きさ S ミリで書け。

ALPB(A, S);

以下同様 **ALPZ** まで用意してある。

(これらの諸手続きについては付録に示す。)

たとえば次の正弦曲線を 0.5 ミリステップで画か
 せるプログラムは ($A \leq 125$)



begin

external **PREADY, PEND, PSENX,**

ALPX, PMOVE;

real X, A, B, C ;

READ(A, B, C);

PREADY(0, 125);

for $X:=0$ step 0.5 until C do

**PSENX($X, A * \sin(N * B / 180$
 $* 3.14159265)$);**

PMOVE(20, -10);

ALPX(0, 5);

PEND;

end;

9.3 ALGOL による自動設計の1例

ボックスカルバートの自動設計例

幅、高さ、土かぶり厚、土圧係数、答の数値を格納する array (配列) の名前をパラメータとした external procedure *BOXCULVERT*, および、その答を受けて図化する方法を示す。*DBOXCULVERT* を、あらかじめ記憶している辞書の中から呼び出して、幅 3.5m、高さ、4.0m、土かぶり厚 0m、土圧係数 0.33 として計算し、答を *ANS* という array に格納し、ただちにそれを受けて図化するプログラムはつぎのようなものである。

```
begin
  array ANS [1:38]
  external BOXCULVERT, DBOXCULVERT;
  BOXCULVERT (3.5, 4.0, 0, 0.33 ANS);
  DBOXCULVERT (ANS);
end;
```

図-2 は、上のプログラムによって図化機が画きだしたもので、同じような寸法の標準設計が図-1 である。材料表などという字の書き方は、あらかじめ辞書に教えてあるほか、図中の数字は *ANS* に格納された計算結果に基づいて計算機が自から判断して書き出したものであり、図面の大きさも、すべて機械があらかじめプログラムで示された判断基準により適当に縮尺を選定して作図

したものを(図-2)にまた図化機全体を写真-3に示す。

10° アルゴルマニュアル

土木研究所においては技術管理室において作成した *ALGOL* テキスト(文献4)による2日間研修(実習を含む)によって、各研究者の頭の中にある手順をとにかく日本語で書き下すのと同様と同じように *ALGOL* で書き下すようになるまでの教育を行ってきた。その研修プログラムは次のようなものである。

第1日午前

- 9時~11時 日本語と対比しつつ1つのケースの計算印字、個数が多い場合の処理、個数がわかっている場合の処理、配列処理の概要を修得。
- 11時~12時 その知識により、自分自身のプログラムにまとめて提出

第1日午後

- 1時~3時 計算室においてデモンストレーションならびに各自のプログラムを計算機によりチェックならびに計算
- 3時~5時 テキストにより手続きを除く文法詳論

第2日午前

- 9時~10時半 手続きの話にはいる。

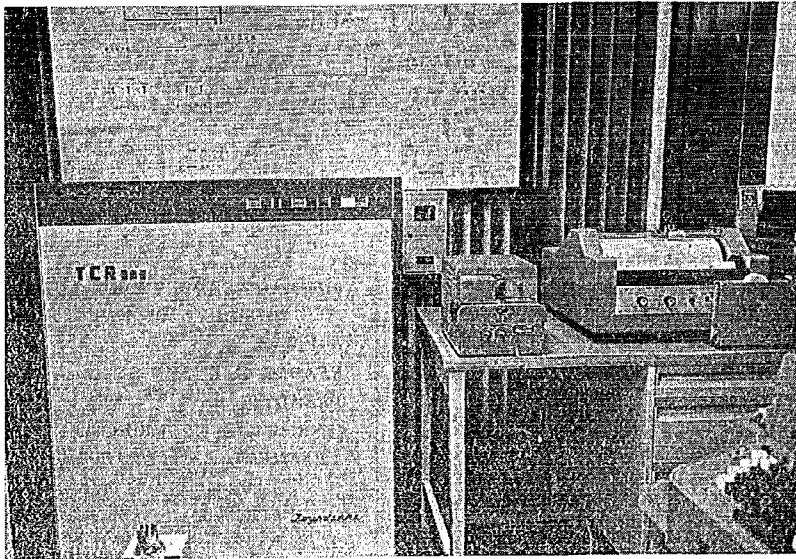
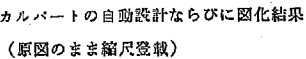


写真-3



字付変数はその所在をつきとめるのに時間がかかるので、式中で何回も現われるときはいったん他の単純変数に受けて使用するほうがよい。

変数はすべて宣言されていなければならない。

添字式は左側から計算され、1つの添字式の値の計算は算術式の計算規則に従う。

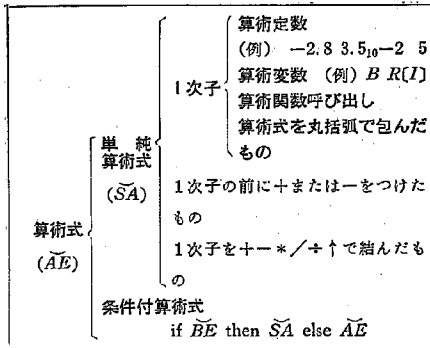
2) 標準関数, 換算関数

abs sign sin cos sqrt
arctan ln exp entier

(大文字となるものもある)

- abs は () の中と同じタイプの絶対値。
- sign は integer の結果で () の中が正のとき +1, 0 のとき 0, 負のとき -1 となるが, 他はすべて real の結果となる。
- entier はそれを越さぬ最大の整数をとる。したがって entier (8.6) は 8 であるが entier (-8.6) は -9 となる。

3) 算 術 式 ($\tilde{A}E$)

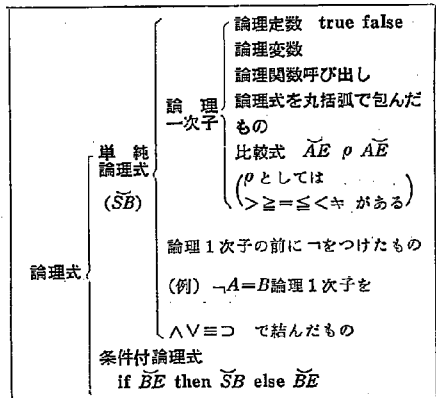


- 演算の優先順位は次のようになる。
同位のものの中では左側が優先する。

- ()
- ↑
- * / +
- + -

- 算術作用素の両側のタイプが異なるときはその結果は real となる。ただし + の両側が共に整数型の場合たとえば 5+2 は 2 となる。
- 条件式は if then else がそろっている必要があり, 条件式を作用素で結んだり then の後で用いるときは () で包んで単純式に直さなければならない。
- 算術作用素を / - 等と連続して用いてはならない。

4) 論 理 式 ($\tilde{B}E$)



- 演算の優先順位は次のようになる。

- | | |
|-----------|--------------|
| 1. () | 5. $\wedge$ |
| 2. 算術作用素 | 6. $\vee$ |
| 3. 比較作用素 | 7. $\supset$ |
| 4. $\neg$ | 8. $\equiv$ |

同位のものの間では左優先である。

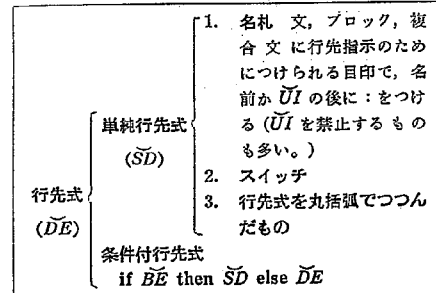
- $\neg A \vee B \supset C$ は次の結果をもたらす。

$\neg A \dots\dots A$ が true のとき false, A が false のとき true
 $A \wedge B \dots\dots A, B$ ともに true のときのみ true
 $A \vee B \dots\dots A, B$ ともに false のときのみ false
 $A \equiv B \dots\dots A, B$ ともに true またはともに false のときのみ true
 $A \supset B \dots\dots A$ true かつ B false のときのみ false

- 論理作用素は $\vee \wedge$ などと連続して用いてはならない。したがって $\neg 1 > x > 1$ は $\neg 1 > x \wedge x > 1$ としなければならない。

- 条件式を演算記号で結んで用いるときや then の後で用いるときは丸括弧に包んで単純式とする。

5) 行 先 式 ($\tilde{D}E$)



スイッチの添字式が実数型のときは4捨5入される。
スイッチとして宣言された範囲をこえるときは空文扱いとなる(エラーとするものも多い)。

10.3 節 (clause)

節 { 条件節 if $\tilde{B}\tilde{E}$ then
繰り返し節 for $\tilde{V} := \langle \text{for list} \rangle$ do

条件節は条件式や条件文の一部、繰り返し節は繰り返し文の一部として用いられる。

<for list> としては

1. $\tilde{A}\tilde{E}$
2. $\tilde{A}\tilde{E}$ step $\tilde{A}\tilde{E}$ until $\tilde{A}\tilde{E}$
3. $\tilde{A}\tilde{E}$ while $\tilde{B}\tilde{E}$
4. 1~3 をカンマでつないだもの

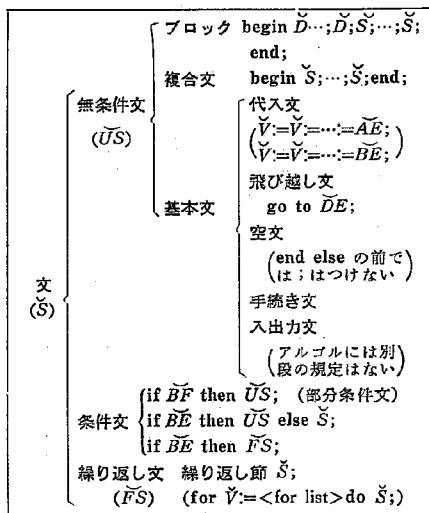
があり、たとえば

for $I=1$ step 1 until 20, 22 step 2

until 50, 70, 100, $I*2$ while $I < 10000$ do

で、次の文を I の 1, 2, 3, ..., 20, 22, 24, 26, ..., 50, 70, 100, 200, 400, 800, ..., 6400 について実行する。

10.4 文 (statement)



1) ブロックと複合文

i. ブロックと複合文の差異は宣言の有無だけであるが、名札はそれを含む最も内側のブロックの頭で宣言されたものとみなされる。

ii. 宣言はそれを含む最もせまいブロックについてしか意味をもたず、その外側において宣言された名前は内側で同じ名前が宣言されていない限り有効であ

る。同じ名前が内側でも宣言されているときは、内側では内側で宣言された意味のほうが優先する。

iii. 名札についてもブロック外に飛び出すことは可能であるが、内側には begin からはいっていかなければならない。

iv. セミコロンは宣言や文の区切りとして使用され end の前では不用であるが、つい end の後に不用であるかのように錯覚することが多いので end の前には空文があるものと解釈して常にセミコロンを入れるようにしたほうがよい。もちろん end の後は else の直前の場合を除いてセミコロンが必要である。

v. スイッチ宣言および非局所的な名札を含む手続きの宣言は、その名札を含む最も内側のブロックの頭で宣言されないと行先不明となる。

2) 代入文

左右辺のタイプが異なるときは自動的に変換され特に左辺整数型の場合は4捨5入が行なわれる。また左辺が何重かになった多重代入文が許されるが、左辺の中では同じタイプでなければならない。

演算の順序は、(1)左辺の添字式の計算、(2)右辺の式の計算、(3)左辺の変数に個別に代入の手順をとる。

3) 手続き文

宣言された手続きの内容はその名前の後に対応する実パラメタ ($\tilde{A}\tilde{P}$) を書いた手続き文によって実行される。

4) 条件文

then の後は無条件文または繰り返し文に限る。したがって条件文が then の後にくるときは begin end でくって無条件文に直す必要がある。

5) 繰り返し文

<繰り返し節> $\tilde{S};$

つまり

for $\tilde{V} := \langle \text{for list} \rangle$ do $\tilde{S};$

で do の効力は次の1つの文だけにしか及ばない。したがっていくつかの文について繰り返ししたいときは begin end でくって1つの文に直す必要がある。

until の後は2進変換の都合で実行しない場合もあるので、きざみの $1/4 \sim 1/5$ を加え(負のきざみなら引く)ておくほうがよい。

繰り返し文の途中で外に飛び出すときは、コントロール変数はその値を保持しているが、<for list> について完全に終了したときは、コントロール変数がどのような値になっているかわからない。

do の後の文の名札に飛び越すことを認めない。

6) コメント

begin および; の直後に comment と書けば次の; ま
で何を書いても機械は無視する。また end の後は次の
end else または; までの間は何を書いても機械は無視
する。

v) 入出力文

改訂アルゴリズム 60 には入出力に関する何の規定もない
ので各コンパイラの規定を用いるより仕方がない。一般
には入力は

READ(A, B, ……); Read(A, B, ……);

等が多いが

Readreal(A);

などと1つに限るものもある。出力は

PRINT('フォーマット', リスト);

PRINT(n, A); Printreal(A, n);

などまちまちである。

ただしフォーマットの中の記号としては ISO の勧告
で B (プラנק) C (カンマ) D (数字) V (印字
なしの小数点) Z (数字ただしゼロサプレス) / (行
おくり) などを共通に用いるものが多い。

10.5 宣言 (declaration)

宣 言	型 宣 言	real integer Boolean
	配列宣言	(real) array integer array Boolean array
	手 続 宣 言	手 続 き procedure (関 数) (real procedure integer procedure Boolean procedure)
	スイッチ宣言	switch

1) 宣言

宣言詞の後に単純変数をカンマで区切って記入し一
段落でセミコロン。

2) 配列宣言

宣言詞の後に名前を書き [] の中に添字の限界を行
について下限: 上限, 列について下限: 上限の順で書く。
何次元であってもこのように, でつないで書けばよい。
また同じ限界なら名前を, でつないで並べてもよい。
添字の限界には式を用いてよいが整数定数に限るもの
もある。また (下限) ≤ (上限) でなければならない。
単に array と書けば real array とみなされる。

(例) array A, B(1:10), PQ(1:5, -8:4);

Boolean array CR(0:4);

3) スイッチ宣言

一種の名札の配列であって,

switch 名前 := 行先式の並び;

スイッチの要素は行先式であるから, また他のスイ
ッチであってもよい。

使用は go to SW [I]; と飛び越し文などでその名
前のスイッチの何番目の要素であるかを指定する形で行
なう。

スイッチ宣言はその名札の宣言を含む最も内側のプロ
ックの頭において行なわなければならない。

(例) switch Q=L1, L2, if A=B then

D else LD, RS, S[I];

この場合は L1, L2 D, LD, RS は名札 S[I] は他
のスイッチリストの要素である。

4) 手続きの宣言

i. 手続き宣言は

```
procedure 手続き名 <FP 部>;
<値 の 部>
<規 制 部>
<手続き本体>;
```

の形をとる。

ii. 関数宣言は1個関数の場合に, real procedure
などとその答のタイプを前につけると同時に手続き
の本体中で自分自身の名前に対し答を代入する文が
必要である。また手続きは手続き文として使用され
るのに対して, 関数呼び出しは式の中で使用される。
iii. 宣言中仮パラメタ部は (FP, FP, ……) の形をと
るが, ない場合はこの部分を省略する。値の部は
FP 中値を外から値で貰ってくるものについて指定
を行なう部分であって,

value 名前, ……, 名前;

という形をとる。

このようにしたほうが繰り返し現われる変数につ
いては演算時間が早くなる。しかし value 指定し
なくても差し支えないわけで, このときは値の部
は省略される。value とはその実パラメタとなっ
ている変数の値を変更しないという意志表示である。

(配列の value 指定を禁止するものもある)。

iv. 規制部は FP にあげられたパラメタがどんなも
のであるかを規制する部分である (名前で呼ばれる
ものはすべて規制不用のもの, value 指定されな
かったパラメタのうち整数型のみ規制を必要とするも
の, value 指定されたものを含んですべてのパラメ
タの規制を必要とするものなど機種によっていろ
ろある)。

規制詞としては

string, label, real, integer, Boolean, array,
real array, integer array, Boolean array,
switch, procedure, real procedure, integer

procedure, Boolean procedure

がある。

- v. 機械語等との接続を可能にするため code という宣言詞もあるがコンパイラによってはこれを禁止している。
- vi. 手続きの実行は、手続き名の後に仮パラメタのある場合はそれに対応する位置に実パラメタを書きこんだ手続き文によって行なう。

関数呼び出しも同様実パラメタを書き込み、式の中で使用する。

- vii. データは入力文に対応して必要な情報を並べて用意する。コンパイラによりカンマで区切るものなどいろいろある。

10.6 プログラムエラーのチェック

プログラムに文法的エラーがあれば機械がエラーメッセージを打ってくるが一応次の点についてチェックしてから機械にかけることが望ましい。

(1) 算術演算

- 1. 掛算の * をおとして $(A+B)C$ などとしていないか？
- 2. 割算の分母に 0 がはいってくるようになっていないか？

(代入や読込みによって値を与えてない変数を使いなり使っていないか)

- 3. $A \uparrow -5$ とか $3 / -4.8$ などと 2 つ以上の算術作用素を並べて使っていないか？ (括弧を用いる)

(2) 変 数

- 1. 未宣言の変数を使っていないか？
(特に手続き文などの実パラメタが仮パラメタと同じ名前するときなど)
- 2. 同じブロック内で変数、名札、手続き、関数呼び出し、スイッチ等の名前に同じものを使っていないか？
- 3. 代入または読込みによって値を与えないで変数を使っていないか？

(3) 区切り記号

- 1. 区切り記号の綴りに誤りはないか？
- 2. begin end, (), [], ' ' などが対応しているか？

(4) 名 前

- 1. 0 と O と W を間違えていないか？
- 2. l と 1 と I と / を間違えていないか？
- 3. r と V と v を間違えていないか？
- 4. (と C を間違えていないか？
- 5. 2 と Z を間違えていないか？
- 6. 5 と S を間違えていないか？

(5) 式

- 1. 条件付算術式、論理式、行先式は if then else の 3 つともを含んでいるか？
- 2. 条件付の式で then の次に if がくるときは、if 以下の式を () でつつんであるか？

(6) 文

- 1. 文は else の直前以外すべて ; で終わっているか？
特に begin end で包んだ複合文の場合に end の後の ; を落していないか？ (end の直前では不用であるが end の直後と間違いやすいので文はすべて ; で終わることにしておいた方が誤りが少ない。
空文があると考えれば end の直前に ; があってもさしつかえない)
- 2. 代入文の左辺の変数の型はみな同じものか？
- 3. 条件文で then の後に if がくるときは if 以下の文を begin end でつつんであるか？
- 4. 繰り返し文の do の後の繰り返すべき文は最初のセミコロン(begin がくるときは対応する end の後のセミコロン) までであるが、それでよいのか？
- 5. 繰り返し文の中で初めにもどって初期値からやり直していることはないか？ (初期値代入が do の後にはいつている場合)
- 6. 繰り返し文のコントロール変数について、指定どおりくり返し計算が終了してから次の文の中でこのコントロール変数の値を利用していないか？
- 7. 繰り返し文の do の後の繰り返すべき文の中にはいつている入力文に対応してデータテープが用意されているか？
- 8. 手続き中で繰り返し文に用いる I, J 等のコントロール変数を宣言しなかったために本体の I, J と混同を起すことはないか？
- 9. end の後の ; を忘れたために次の文がコメント扱いとなって読みとばされるおそれはないか？
- 10. comment は ; で終わっているか？
- 11. 繰り返し文の中で毎回の答を格納し忘れてないか？

11° KDK-ALGOL の構想
(建設省土木規格アルゴル)

11.1 一般

11.1.1 アルファタイプの取り扱い

ALGOL-60 によれば変数は real, integer, Boolean の 3 種に分かれ、これに対応して配列も (real) array, integer array, Boolean array 関数も real procedure, integer procedure, Boolean procedure の 3 種が用意してある。

NEW ALGOL-H においては文字の情報 (アルファタイプ) の存在を間接的に認め、整数タイプの変数に対応して

{ 読込み データテープ上 = を先頭に7文字まで
 代入 $K := EQUIV('X \cdots X')$; による
 印字 $PRINT('A', K)$; による

という処理を認めてきた。

Burroughs B 5000 において ALPHA という宣言を認め、この考え方は B 5500 にも引き継がれてきているが、このアルファタイプの変数の存在は、表などを再配列して印字する際にも極めて便利なので KDK-ALGOL においても次の形でアルファタイプの存在を認めたい。

1. alpha 宣言を認める。したがって alpha array, alpha procedure も許し、一定の規則のもとに整数タイプの情報として取り扱う。
2. alpha タイプの読込み。データテープ上にある文字数字 (=+*(')₁₀ を含むが / , ; を含まない) を6文字まで alpha タイプとして読み込む。
3. alpha タイプの代入。アルファタイプの変数 K に対し $K := 'X \cdots X'$; という形で先頭の6文字を代入。string と規制された変数に対しては $K := S$; でもよい。
4. 文字式の新設。条件付文字式の存在を認める。(四則演算も換算規則を知っているプログラマには可能のようにする。)
5. alpha タイプの印字

$PRINTA(6, K)$; などとして印字を認める。この場合 PRINTI (後述) などと違って前後にスペースを取らない。もし6文字はいつているものを5文字指定したら先頭の5文字を印字するものとする。

11.1.2 標準関数

ALGOL-60 で定められたものの他に fix, sinh, cosh, log, random, nrandom を認める。あるいはこれらは後の宣言不用の準アルゴリズムと一緒に大文字にしてもよい。fix は entier と違って負の場合も切り捨て、log は常用対数、random は (0~1) 一様乱数、nrandom は正規乱数である。

11.1.3 attach

標準手続き、あるいは関数の追加のために、括弧類に attach を追加するか、あるいはこれに代わる方法で中間語に落されたこれら手続きがその名前を索引として自由に引き出し得るものとしたい。この際照会時間がかかるならば頻度の多いものは無宣言、他は external 宣言によって引き出す形でもよい。

11.1.4 halt

プログラムの読込み中絶の目的で halt ; を認める。データの読込み中絶を / によって認める。

11.1.5 他言語との接続

機械語、あるいは中間語と FORTRAN を procedure

の本体中で接続することを認める。

11.1.6 double 宣言

倍精度演算

11.1.7 complex 宣言

複素数演算

11.2 式

ALGOL-60 には算術式、論理式、行先式の3つがある。KDK-ALGOL では前述の文字式(alpha expression) の存在を認めたい。

文法的にすっきりさせるためには switch を止めて label array ならびに label expression (designational expression), label procedure を作るとよいが、これは困難であるかも知れない。

11.2.1 算術式

型は ALGOL-60 どおり、条件節とそれに対応する else に囲まれた単純算術式とその else に続く最長の算術式の型が共に整数型なら整数型、それ以外は実数型とする。

1 次子の評価は優先順位に従い、同順位のとときは左優先で選定された時点において評価するものとする。

11.2.2 論理式

演算の順序については 11.2.1 に同じ。

11.2.3 行先式

11.2.4 文字式

11.3 複合文とブロック

名札に関して ALGOL-60 どおり複合文とブロックの区別を行なうものとする。

11.4 文

11.4.1 代入文

ALGOL-60 とは異なるが、多重代入の際左辺の real と integer の混在を認めると非常に使いやすいものになる。KDK-ALGOL では可能ならばこのようにしてみたい。

関数名と変数への多重代入はもちろん認められるが、できれば左辺の型の混在を認めたい。

代入文実行の順序は ALGOL-60 のとおりとする。

また fill A [5, *] := A1, A2, A3, ... ; などとして任意の行または列への代入を認める。

11.4.2 飛び越し文

行先式の値が定義されないスイッチであるときはエラーとする。

11.4.3 繰り返し文

もし2進10進変換の都合で階段要素の until の後を10進表現では実行させる積りのものが実行されないようならば、階段要素の until の後は符号を含めたきざみの1/5 を自動的に加えて置くものとする。

これは困難かも知れないが do の後に空文がくるとき

はエラーを検出してけると非常にありがたい。また end の後に ; がくるまでに := がきても検出してもらえると不注意のエラーが検出できる。

11.4.4 入出力文

ALGOL-60 には入出力文に関する規定はない。しかし JIS-70 の様式のほかに日常の入出力文として次のものが取り入れてあることが望ましく、KDK-ALGOL では無宣言で使用できる入出力文として次のものを規定する。

READ(A, B, ...); データテープより順に情報を読み、それぞれ A, B, ...に格納せよ。

PRINTR(N, A); 式 A の値を実数のタイプの有効 N けたの情報として次の形で印刷する準備をせよ。

$\square \pm . \underbrace{\times \times \cdots \times}_{N \text{ 個}} \times 10^{\pm \underbrace{\times \times}_{2 \text{ 個}}}$

PRINTI(N, A); 式 A の値を integer タイプの有効 N けたの情報として次の形で印刷する準備をせよ。

$\square \pm \underbrace{\times \times \cdots \times}_{N \text{ 個}} \square$

PRINTF(n_1, n_2, A); 式 A の値を実数タイプで小数点以上 n_1 けた n_2 けたの次の形で印刷する準備をせよ。

$\square \pm \underbrace{\times \times \cdots \times}_{n_1 \text{ 個}} . \underbrace{\times \cdots \times}_{n_2 \text{ 個}} \square$

PRINTS('PQR='); PQR=という字を印刷する準備をせよ。

PRINTA(N, A); 式 A の値を alpha タイプで N 文字、文前後のスペースなしに印刷する準備をせよ。

SPACE(N); N 字分の空白を印刷する準備をせよ。

CRLF(N); 今までの情報を 1 行に印刷し、N 行おくれ、CRLF は carriage return line feed の略で、この指令がくるまでは情報はつぎつぎと同じ行の右側へつめ込まれ印刷準備される。

配列読み込みは、

READARRAY(I, L1, U1, L2, U2, ...);

で配列名を I 上下限の組を L, U の順に内側から (行列なら行から) 並べてその次元数は任意となるようにしたい。また宣言された通りの大きさのデータに対しては

READ(I);

でよいこととしたい。

CRLF(0);

は改行なしの印字の実行である。

また、フォーマットを用いる場合はレプリケータをなんらかの形で変数で指定できるようにしたい。

11.4.5 手続き文

値取りのパラメタを除き規制はなくても良いこととし、ない場合は実パラメタの型に従うものとする。

再帰的呼び出しはこれを認めることとする。

本体中の非局所的な名前は同じレベルで宣言してさえあれば良いものとする。

11.5 宣 言

ALGOL-60 では 型、配列、スイッチ、手続きの宣言の 4 種がある。KDK-ALGOL では、前述の文字 (alpha) 宣言および own の他に double, complex を追加したい。

recursive switch および cross switch を認める。

11.6 そ の 他

補助記憶装置との間に PUTD(E, I); GETD(E, I); を認める。ただし、E の値は符号のない定数とする。

do の後に直ぐ ; のくること、および end の後に end else ; までの間に := がくることを禁止しエラーを表示する。(これは ALGOL-60 では許されているが、いずれも危険なので禁止したい。ただし do の後にきた procedure や S が結果として空文になることはさまたげない)

11.7 モ ニ タ

- 優先順位 2 のプログラムはすべて補助メモリより順次引出して実行され、その結果は補助メモリに出し、要求により LP に印字するものとする。
- 優先順位 1 の計算は PTR より割込み優先実行し、その結果は LP に印字する。もし順位 2 のものがコンパイル中であれば割り込みを留保し、また計算終了または 10 分間経過すれば計算を打ち切って直ちに順位 2 の計算に復帰するものとする。
- エラーメッセージはすべて ET 2 へ、プログラムごとの各段階計時記録は ET 1 へ出すものとする。
- 順位 2 の計算はすべてそのプログラム内容を 1 行 100 字でカウント (空行もカウント) した行番号を右にして LP で印字するものとする。(これを用いてテープの自動修正を行なう予定)

12° 結 論

12.1 電子計算用語の問題

問題をもつ各人がその問題に関しては素人であるいわゆるプログラマに、その問題の意味を説明する手数を考えると、人間語に近い言葉で直接計算機と対話する方式が有利であることは言うまでもないが、そのための用語

として FORTRAN をとるべきか、ALGOL をとるべきかについては導入計画の当初慎重な検討を行なった。

その普及の程度から云えば FORTRAN がまさり、その習得の容易さ、文法の一貫性と発展性から云えば ALGOL が優れている。しかし FORTRAN プログラムを ALGOL に直訳することは極めて容易であるほか (逆は不可能)、FORTRAN から中間語に落されたものを external で使用することもできるので、これから開発していこうとする建設事業分野では各人が楽に書けることの有利さの方が重要であって、この点について 2 章の結論でも触れたように、単なる機械的制限の少ない ALGOL が断然有利である。事実この言葉のおかげでわずか 3 年の間に 1500 人ものアルゴリズムプログラマを養成することができたと考えられ、今後とも建設事業各分野で使われる電子計算用語として ALGOL が適当であると結論することができる。

12.2 入力媒体の問題

使用用語 ALGOL と割り切れば入力媒体は紙テープと決まったようなものではあるが、ALGOL をカードによって使用している所もあるので一応検討を行なった。

しかし読み速度は大差のないのに対して費用には大差があり、むしろ何故カードに固執する所が多いのかが不思議に思われたが、カード分類集計機を売り込んでしまった IBM が、電子計算機の時代にはいったときに、いまだこのユーザを見ずともならず、そのまま続けたものと想像され、文句なしに紙テープ使用が有利と結論される。その中でも下段コードやテープの上下関係の判別しやすい 8 単位テープが最適で、土木研究所では、さらに 7 章で触れたように特殊な先頭マークをさん孔し、さらに矢印の印刷してあるテープを用いてテープの左右表裏を誤らないようにしている。

紙テープの修正に関してはテープスプライサを用いて自由に切り張りを行っており、孔の位置から十字ぐらいずれて送り孔の位置に文字を印字することも可能なので、全く不自由を感じていない。なおテープの保存のために、必要なテープにはかならず先端にレベルを張り、廃棄するときはこの部分を切ってからにするよう励行している。

12.3 割り込みの問題

プログラマにとっての関心事の 1 つにターンアラウンドタイムがある。いかに能力のある計算機でもプログラムができてからエラーや簡単な例題の計算結果が手にはいるまでに 1 週間もかかるようでは、せっかくの熱意やアイデアもしぼんでしまうわけで、このため土木研究所としてはこのような 15 分以内の仕事は即時処理を行なうことを考えた。しかし OKITAC 5090-H のようなメモリー 16 K 長語 浮動小数 250 μ s 程度の中型機で

は同時平行処理は不可能なので、紙テープリーダに優先順位を設け、長時間演算のものを処理中に 15 分以内の計算を割り込ませると今までの計算に関する情報を一時磁気テープに格納して新しい計算を始め、これにエラーがあるかまたは 15 分経過すると自動的に元の状態にもどし、見かけ上 100% の稼働率を可能にすることができた。詳細は 6 章に触れたが割り込まれる方の計算結果はすべて磁気テープに格納し、割り込んだ方の結果をラインプリンタに出すので、この方の混乱もまったくなく、土木研究所のシステムが満足に働いている理由の大半はこの割り込み処理を立案し実現した点にあると云うことができよう。この方式は機種変更後も採用の予定である。またルーチンの計算と新規プログラムの混在することの多い技術関係機関における電子計算システムとしても必須のものと考えられる。

12.4 NEW ALGOL-H の文法制限の問題

4 章で論じたように NEW ALGOL-H は文法制限の比較的少ない良いコンパイラであるが、製作当初における沖電気側との打ち合わせ不足から特に次の諸点については不便を感じる人が多い。

1. ALGOL-60 よりの制限事項

- 算術式の then の後 else の後の算術式の型が一致しなければならないために ENTIER, FLOAT の換算関数に注意しなければならないほか、特に他の部分で 0 と 0.0 をまったく区別しないのに、この場合だけ区別を要求せられることによるエラーが頻発した。
- 算術式で右括弧の直後の文字数字は無視する。このため $(A+B)*C$ を $(A+B)C$ と書いても $(A+B)$ と扱われるだけでエラーにならず危険である。
- 手続きの heading 部分に comment を書けないために、その procedure の説明を本体中で行なう必要を生じ不便であった。
- procedure の中で扱う非局所的な名前は既宣言のものでなければならず、一般に宣言部における宣言の順序は無関係である原則に反し不便であった。

2. 拡張事項関係

- アルファ扱いは非常に便利であったがストリングの情報との間の関連性がないことが不便であった。
- attach によってシステムに external procedure の格納ができ、非常に便利であったが、取消不能のため、一つ訂正するためにも全部の attach をやり直す必要があり不便であった。
- PRINTI, R, F, S および SPACE, CRLF は external 宣言を必要とし不便であった。

しかし全体として見るときはそれほど重大な制限事項もなく、external procedure の使用可能なこと、alpha

タイプの取り扱い可能なこと、codeにより機械語の接続が可能なこと、REPXによってフォーマット中の繰返しを認めたこと、;によって改行なしの印字を認めたこと、onによってコンソールよりの制御を認めたこと、READARRAYによって任意次元の配列読込みを認めたこと、READによってデータテープ上の型に関係なく何個でも読込みを認めたこと、PUT、GETによる配列のMTへの格納を認めたことなど、他のコンパイラにはない特長も多い。

今後の計算システムとしては、プログラムの負担をできるだけ軽くするよう、これらの欠点を正し、これらの長所を取り入れたものとする必要がある。

12.5 KDK-ALGOLの問題

建設省関係の各機関にも、今後電子計算システムの導入が予想され、すでに導入された場合にも技術の進歩、拡大する要求に応じて機種変更を必要とする場合も多いものと想像される。この場合ALGOLという用語一本にしぼってあれば、ちょうどOKITAC 5090-H用のNEWALGOL-Hで書かれたプログラムテープが、本番計算はCDC 3600によって直接読み込まれ、CDC ALGOLへの変換は計算機によって容易に実行され得るように、機械間のトラブルは割合少ない。

しかし入出力文の変換などには多少の困難さを伴う上、文法上の制限事項もある程度そろえてあることが望ましいので、11章においてOKITAC 5090-Hでの経験にかんがみ次期機種TOSBAC 3400-40への切り替え、千葉支所のFACOM 270-30との連絡、本省あるいは地方建設局の計算機との連絡を考慮して、建設省土木規格アルゴル(KDK-ALGOL)の検討を行なった。

主要点は次のとおりである。

a. 入出力文はREAD, PRINTI, PRINTR, PRINTF, SPACE, CRLF, PRINTAをかならず含むこと。

b. alphaタイプを新設しストリングとの変換を認め、文字式等を可能とすること。

c. 標準関数にSINH, COSH, RANDOM, NRANDOM, LOGを追加する。

d. external procedureをattachできるようにする。

e. fill A [4, *]:=A1, A2, ...; という配列への代入を認める。

f. complex, doubleを認める。

g. モニタにより割り込み処理を認めるものとする。
NEW ALGOL-Hは上述のとおり、問題を持つ技術者の多い組織には至極適応した良いコンパイラであり、土木研究所においても各人が自由にその発想をプログラムしてそれを試みることができ、大いに技術の進歩に貢献することができたものと信じている。このコンパイラ

第1次案の作製に当たって当時としては珍しい方式であった土木研究所側の構想に対応して作業された沖電気(株)プログラム課の滝川洋、上原一矩の諸氏の御努力、土木研究所での実施にはいつてからのOBM(株)の保守員太田靖彦氏、土研技術管理室を始め各研究室のプログラマ全員のより良いシステムの樹立に対する熱意なしにはこの成果は考えられないわけで、さらに強力な機種への変更に当り一層内容の充実したシステム樹立への御協力を切望して本報告を終わることとした。

参考文献

1. 電算速報 1～6号 土木研究所資料 38, 88, 122, 134, 175, 259号(1966～1967)
2. 昭和40年度 プログラム成果 土木研究所資料 196号(1966) 昭和41年度 プログラム成果 土木研究所資料 311号(1967)
3. 中村慶一: やさしい電子計算用語 ALGOL 1～5 土木施工 7巻4号～8号(1966)
4. 電子計算用語 ALGOL 土木研究所テキスト(1966)
5. Naur: Revised Report on the Algorithmic Language ALGOL 60. ACM. Vol. 6 No.1 (1963)
6. 建設省第2, 3, 4回電子計算シンポジウム報告 土木技術資料 8, 9, 10巻3号(1966, 67, 68)
7. 中村慶一: アルゴルプログラミング入門 森北出版(1968)

付録2 OKITAC 5090-Hの機械語について

OKITAC 5090-Hは1語が2進42ビットで機械語は8進14けたで書かれている。

命令語は1語1命令であるが、補助語を必要とするものもある。命令語の形式は本体関係および入出力、インディケータ関係の種類になる。

1. 本体関係命令語

S	f	O	a	X	i	n
1 ビット	5 ビット	9 ビット	6 ビット	6 ビット	1 ビット	1 ビット

1-1 S: SCC 修飾の指定をする。

1-2 f: マスクレジスタ、パーシャルレジスタの指定をする(MRは16個, PRは15個ある)

1-3 o: 命令を示す。

1-4 a: アリスメディクレジスタの指定をする(ARは63個ある)

1-5 X: インデックスレジスタの指定をする(XRは15個ある)

1-6 i: 間接アドレスを指定する。

1-7 n: 番地を指定する。

2. 入出力ならびにインディケータ関係命令

S	e	u	O	C	X	i	n
1 ビット	1 ビット	5 ビット	9 ビット	6 ビット	6 ビット	1 ビット	14 ビット

2-1 S: SCC 修飾の指定をする。

2-2 e: 入出力エンドインジケータを設置するかどうかを指定する。

2-3 u: ユニット番号を指定する。

2-4 O: 命令を示す。

2-5 C: チャネル番号を指定する。

2-6 X: インデックスレジスタの指定をする。

2-7 i: 間接アドレスを指定する。

2-8 n: 番地を指定する。

以上の他補助語の詳細については OKITAC 5090-H instruction manual を読んでいただきたい。
命令の一覧表はつぎのとおりである。

命 令 一 覧 表

000	STOP
001	NON EFFECT
002	MEMORY LOCK OUT
004	RETURN JUMP
005	EXECUTE
010	SET BIT
011	RESET BIT
012	CHANGE BIT
020	NORMALIZE
024	FLOAT
025	FLOAT UNNORMALIZED
026	FIX
030	ROUND OFF
031	FLOATING ROUND OFF
032	FLOATING UNNORMALIZED ROUND OFF
040	ADD FLOATING
041	ADD FLOATING DOUBLE
042	ADD ABSOLUTE FLOATING
043	ADD ABSOLUTE FLOATING DOUBLE
044	ADD UNNORMALIZED FLOATING
045	ADD UNNORMALIZED FLOATING DOUBLE
046	ADD ABSOLUTE UNNORMALIZED FLOATING
047	ADD ABSOLUTE UNNORMALIZED DOUBLE
050	SUBTRACT FLOATING
051	SUBTRACT FLOATING DOUBLE
052	SUBTRACT ABSOLUTE FLOATING
053	SUBTRACT ABSOLUTE FLOATING DOUBLE
054	SUBTRACT UNNORMALIZED FLOATING
055	SUBTRACT UNNORMALIZED FLOATING DOUBLE
056	SUBTRACT ABSOLUTE UNNORMALIZED FLOATING
057	SUBTRACT ABSOLUTE UNNORMALIZED FLOATING DOUBLE
060	MULTIPLY FLOATING
062	MULTIPLY FLOATING ROUND
064	MULTIPLY UNNORMALIZED FLOATING
066	MULTIPLY UNNORMALIZED FLOATING ROUND
070	DIVIDE FLOATING
072	DIVIDE FLOATING ROUND
074	DIVIDE UNNORMALIZED
076	DIVIDE UNNORMALIZED ROUND

100 SHIFT LEFT
101 SHIFT RIGHT
104 SHIFT LEFT LOGICAL
105 SHIFT RIGHT LOGICAL
106 SHIFT LEFT LOGICAL AROUND
110 SHIFT LEFT ZERO COUNT
116 SHIFT LEFT LOGICAL $\wedge$ BIT COUNT
124 SHIFT CONTROL
125 SHIFT CONTROL NUMERIC
126 SHIFT CONTROL ALPHA-NUMERIC
200 UNCONDITIONAL JUMP
210 INDEX ZERO JUMP OR INCREMENT
211 INDEX ZERO JUMP OR DECREMENT
212 INDEX NON ZERO JUMP $\wedge$ INCREMENT
213 INDEX NON ZERO JUMP $\wedge$ DECREMENT
214 INDEX JUMP IF $q < t$ $\wedge$ INCREMENT
215 INDEX JUMP IF $q < t$ $\wedge$ DECREMENT
220 SENSE INDICATOR JUMP
221 SENSE INDICATOR JUMP $\wedge$ RESET
222 BREAK POINT
223 BREAK POINT $\wedge$ RESET
224 JUMP $\wedge$ HALT
225 JUMP $\wedge$ HALT $\wedge$ RESET
226 SET SI $\wedge$ JUMP
227 RESET SI $\wedge$ JUMP
230 SET MI $\wedge$ JUMP
231 SET BLOCK MI $\wedge$ JUMP
232 RESET MI $\wedge$ JUMP
233 RESET BLOCK MIR $\wedge$ JUMP
300 PTR READ
302 PTR REWIND
304 PTR FORWARD
305 PTR BACKWARD
310 ET READ
311 ET TYPE
312 ET PUNCH
313 ET PUNCH $\wedge$ TYPE
317 HTP PUNCH
320 CARD READ
321 CARD READ $\wedge$ FEED
322 CARD PUNCH
323 CARD PUNCH $\wedge$ FEED
324 CARD STACKER SELECT
325 CARD FEED
330 LINE PRINT
331 LINE PRINT $\wedge$ FEED WITH TAPE CONTROL
333 LINE FEED
335 LINE FEED WITH TAPE CONTROL
340 MT READ
341 MT WRITE
342 MT BCD READ
343 MT BCD WRITE
344 MT REWIND UNLOAD
345 MT REWIND LOAD
346 MT WRITE SM
347 MT WRITE TM
350 MT FORWARD IRG

351 MT BACKWARD IRG
352 MT FORWARD SM
353 MT BACKWARD SM
354 MT FORWARD TM
355 MT BACKWARD TM
356 MT ERASE SKIP
360 DRUM READ
361 DRUM WRITE
400 IMPLICATION
401 AND
402 OR
404 NOT
405 NOT OR
406 NOT AND
407 EXCLUSIVE OR
410 ADD LITERAL
411 SUBTRACT LITERAL
412 LOAD LITERAL
430 ADD BORROW $\wedge$ CARRY
431 SUBTRACT BORROW $\wedge$ CARRY
434 EQUALIZE SIGN
435 CONVERT TO 2'S COMPLEMENT
440 ADD
441 ADD DOUBLE
442 ADD ABSOLUTE
443 ADD ABSOLUTE DOUBLE
444 ADD SPECIAL
450 SUBTRACT
451 SUBTRACT DOUBLE
452 SUBTRACT ABSOLUTE
453 SUBTRACT ABSOLUTE DOUBLE
454 SUBTRACT SPECIAL
460 MULTIPLY
462 MULTIPLY ROUND
464 MULTIPLY INTEGER
470 DIVIDE
472 DIVIDE ROUND
474 DIVIDE INTEGER
500 LOAD
501 LOAD DOUBLE
502 LOAD ABSOLUTE
503 LOAD ABSOLUTE DOUBLE
510 LOAD SPECIAL
511 LOAD REAL TIME CLOCK
516 LOAD MASK INDICATOR REGISTER
517 LOAD SET INDICATOR REGISTER
540 TRANSFER
541 TRANSFER DOUBLE
542 TRANSFER ABSOLUTE
544 TRANSFER CLEAR

545 TRANSFER CLEAR DOUBLE
546 TRANSFER ABSOLUTE CLEAR
550 TRANSFER SPECIAL
551 TRANSFER SCC
552 TRANSFER PSCC
553 TRANSFER DAY TIME CLOCK
554 TRANSFER SPECIAL CLEAR
556 TRANSFER MIR
557 TRANSFER SIR
560 TRANSFER BLOCK GATHER
561 TRANSFER BLOCK SCATTER
600 JUMP ZERO
601 JUMP NON ZERO
602 JUMP NOT NEGATIVE
603 JUMP POSITIVE
604 JUMP NOT POSITIVE
605 JUMP NEGATIVE
606 JUMP PLUS ZERO
607 JUMP SMALLER THAN +0
610 COMPARE JUMP
611 COMPARE JUMP LIMIT
640 COMPARE SKIP EQUAL
641 COMPARE SKIP NOT EQUAL
642 COMPARE SKIP SMALLER OR EQUAL
643 COMPARE SKIP SMALLER
644 COMPARE SKIP LARGER OR EQUAL
645 COMPARE SKIP LARGER
646 COMPARE SKIP INSIDE
647 COMPARE SKIP OUTSIDE
650 COMPARE LITERAL SKIP EQUAL
651 COMPARE LITERAL SKIP NOT EQUAL
652 COMPARE LITERAL SKIP SMALLER OR EQUAL
653 COMPARE LITERAL SKIP SMALLER
654 COMPARE LITERAL SKIP LARGER OR EQUAL
655 COMPARE LITERAL SKIP LARGER
660 SEARCH EQUAL
661 SEARCH NOT EQUAL
662 SEARCH SMALLER OR EQUAL
663 SEARCH SMALLER
664 SEARCH LARGER OR EQUAL
665 SEARCH LARGER
666 SEARCH INSIDE
667 SEARCH OUTSIDE
670 SEARCH MINIMUM
671 SEARCH MAXIMUM
700 EDIT
710 CONVERT TO BINARY INTEGER
711 CONVERT TO BINARY FRACTION
712 CONVERT TO DECIMAL INTEGER
713 CONVERT TO DECIMAL FRACTION

付 録 1-1

L	0000000000000	0000000000000	01001010000153	02001010000153
4L	03001010000153	04001010000153	05001010000153	01001020000161
10L	02001020000161	03001020000161	04001020000161	05001020000164
14L	06001020000161	07001020000161	01001030000214	02001030000222
20L	03001030000222	04001030000222	05001030000230	0000000000000
24L	0000000000000	02000000000234	03000000000241	04000000000315
30L	05000000000320	06000000000323	0000000000000	0000000000000
34L	0000000000000	0000000000000	0000000000000	0000000000000
40L	0000000000000	0000000000000	0000000000000	0000000000000
44L	0000000000000	0000000000000	0000000000000	0000000000000
50L	0000000000000	0000000000000	0000000000000	0000000000000
54L	01001060000353	02001060000345	0000000000000	0000000000000
60L	0000000000000	0000000000000	0000000000000	0000000000000
64L	0000000000000	0000000000000	0000000000000	0000000000000
70L	23341160003365	00000000003173	23341160003365	06754121143173
74L	02344170000000	00000000002260	02344170000000	00000000002260
100L	0000000000000	0000000000000	0000000000000	0000000000000
104L	21320210002063	00000000002044	21320210002063	00000000002044
110L	21323220001246	00000000001135	21323220001246	02545057741135
114L	21330230015013	00000000015031	21330230015013	00000000015031
120L	0000000000000	77777777777777	77777777777777	0000000000000
124L	01300250000143	00000000000140	01300250000143	00000000000140
130L	21311260001306	00000000001262	21311260001306	00000000001262
134L	0000000000000	77777777777777	77777777777777	0000000000000
140L	41300250000003	12226040000000	12225040015000	07000000015000
144L	0000000000000	77777777777777	77777777777777	0000000000000
150L	00000000003425	00000000000000	77777777777777	40200000000001
154L	00500620001406	04005000001045	00412610000000	00004000000361
160L	00200000001067	40200000000001	00500620001407	00200000000155
164L	40004000000001	00000000000000	00560000037735	00016000041630
170L	01005000000445	00540610001637	00500610000165	16641400137777
174L	00411610000001	00500350137777	01104350000016	00500400000212
200L	40604350000002	00500400000213	00544400000210	17510350137777
204L	17550350000210	00560000001630	00016000077735	00500610001637
210L	00104000000000	01233020040165	02104000000124	01104000000052
214L	40004000000001	00000000000745	04005000001045	00004000001027
220L	00412770000001	00200000000454	40004000000001	00000000000000

224L	41221260000002	4020000003777	21311260001567	01233040040223
230L	40200000000001	04005000001045	00412610001455	00200000001072
234L	40200000000001	00412610000002	05005000000440	00004000000403
240L	00200000000306	40004000000001	00000000001026	05005000000440
244L	00412610000003	00004000000403	00650630000016	00200000000306
250L	21500620000313	21550620000267	21550620000273	21660620002030
254L	00010000040255	00001000000000	23500620000314	00651620000313
260L	00200000000301	16550400040255	16500620000313	23660620001557
264L	00010000040255	40200000000002	00200000000306	00351000000001
270L	16500620000313	23660620001555	00004000040255	00356000000000
274L	01005000000313	01005000000311	00517000001652	02005000000450
300L	01233140040242	16500620040255	00410620000001	16550620040255
304L	00652620000024	00200000000267	00500620301372	00004000000361
310L	00200000001067	00001000000000	00000000000541	21300250001615
314L	00000000000541	40200000000001	00412610000004	00200000000236
320L	40200000000001	00412610000005	00200000000236	40004000000001
324L	00000000001026	05005000000440	40500620037776	00540620000242
330L	00412610000006	00004000000403	00650630000026	40200000000003
334L	00517000001564	00200000000342	00651630000025	00200000000306
340L	00500620301372	00004000000361	57224260000001	00004000000431
344L	00200000000274	40004000000001	00000000000000	04005000001045
350L	00004000001027	00412770000002	00200000000454	40004000000001
354L	00000000000000	04005000001045	00004000001027	00412770000003
360L	00200000000454	00200000000310	41221260000002	40200000037777
364L	00540620001412	00553000037762	02104610000036	01105620000006
370L	02105610000022	01105610000014	00444620001426	00540620001411
374L	00500620001414	00540620001413	40600610000003	00500620101417
400L	00540620001413	21311260001566	00200000040361	00200000000332
404L	00412630000016	15550610000410	17550630000410	00410630000001
410L	06220250000412	40200000037775	00411630000001	00500620037763
414L	01104620000002	22560000200000	00010000040311	17550630000423
420L	17550630000424	15550610000423	00540400200000	06221250000424
424L	44221250000003	00010400000134	00544400000311	00004000000431
430L	00200000040403	00200000000430	17550630000435	00500620001565
434L	15550620000435	00227250000436	01214620000434	00200000040431
440L	01231040000000	00560000037735	00016000041640	00560000037761
444L	00006000041647	00560000001601	00016000077735	00557000001652
450L	00560000001640	00016000077735	00560000001647	00006000077761

454L	00643770001701	00200000001131	00500610037740	00540600001675
460L	00450610002020	40600610000002	00440610001571	47220230000000
464L	01333230100005	00500750037740	00410760000002	14550770001433
470L	21330230001616	00004000001023	00601750000534	00641400001701
474L	00200000000534	00553000001665	00540400001664	00710000001664
500L	00002000001666	00500010001666	00474010001572	00464010001571
504L	00440010037702	00440010001704	00450010001667	40602010000002
510L	00440010001573	00540010001704	02345160000000	47220160000000
514L	02354160000001	47220160000000	02355160000001	47220160000000
520L	22346160000000	00004000001021	52221160000001	50221160000001
524L	22341160001610	00004000001021	00500010001614	00004000000772
530L	22347160000000	00004000001021	00500610001701	00540600101655
534L	15550770002012	00010770000105	15550770000541	00011770000105
540L	00500010037760	21300250001615	00004000001025	00500610001575
544L	00500020001664	00500030001461	02105020000006	00641030001437
550L	00200000000560	41214610037775	00500020001664	00600010000541
554L	00540400037701	00540020002024	00540021701652	00200000000541
560L	00500030001461	02104020100006	00600010000564	02005000000555
564L	00560000037740	00020000001700	00412010001146	00540010001700
570L	00540770001701	00553000001703	00540400002020	00500740037740
574L	00412610001460	00004000001002	00450760001571	00011760000152
600L	01333230042020	01333231600000	00540771701655	00517000001702
604L	00640400001706	10226160000000	07005000000475	00540010001667
610L	00640400001705	00200000000614	00651770000003	01226040000614
614L	40601740000003	06005000001051	01233770041700	06005000001060
620L	01233770041700	00540600001675	00500770001701	00004000001027
624L	03005000000456	40600610000002	00440610001571	00600770000737
630L	00540401701655	01333230000005	01333230100000	00412610001447
634L	42221040000003	00641400001707	40200000000002	00412610001452
640L	00004000001002	01333230000002	00410760000002	00412610001462
644L	00640400001705	00412610001470	23550610001620	21330230001620
650L	00004000001023	00410760000002	00500010001703	01104010000022
654L	00540010001664	00700000001441	02106063701664	21330230001621
660L	00004000001023	11005000000473	40602010000002	00440010001573
664L	00464010001574	00540010001664	00712000001664	00002000001665
670L	00700000001441	02106064141666	21330230001621	00004000001023
674L	00410760000003	21330230001622	00004000001023	00410760000001
700L	00500610001576	00500010001513	00540010001442	13550610001442

704L	00500010001461	00640400101655	00500010101652	00540010001443
710L	00500010001514	00641400101660	00500010001515	00540010001444
714L	21330230001621	00004000001023	00410760000001	01214610000703
720L	00450760001571	01333231600000	00412760000005	00412770000003
724L	00640401701655	00200000000746	00640401701660	40200000000004
730L	00411770000001	00600770000737	00200000000724	00540401701660
734L	01333230000005	40500750000000	00200000000466	00540400001701
740L	00517000037740	00540400002020	06005000001060	41233770000001
744L	00000000000000	40200000037777	01333230000005	02345160000000
750L	02354160000001	02355160000001	47220160000000	02353160000001
754L	47220160000000	02352160000001	47220160000000	22340160001610
760L	00004000001021	00500010001613	00004000000772	02353160000001
764L	22347160000000	00004000001021	00500770001701	00412610001466
770L	00412740001705	00200000000575	00200000000000	40540010000003
774L	00500610001612	23550610001611	00000000000000	00004000001021
1000L	41214610037775	00200000040772	00200000000641	00500010002024
1004L	00540010001441	00560000100000	00006000041442	00553000037702
1010L	00500010037777	01104020000022	02104010000044	00540010001664
1014L	00700000001445	02034063601664	21330230001617	00004000001023
1020L	00200000041002	01221160001123	40200000037777	01221230000472
1024L	40200000037777	01221250000543	40200000037777	00200000000624
1030L	00560000037700	00200000041710	00500610001027	00500400137775
1034L	00540400001700	00557000001702	50220160000002	01104400000052
1040L	00544400001706	00412760000005	00560000001577	00022000077733
1044L	00200000041027	00233770000000	00231640000000	01233770000000
1050L	00540600001674	00233770000000	00231760000000	01231760000000
1054L	00540400001674	00540400001675	00560000001710	00200000077700
1060L	00233770000000	00231200000000	01231770000000	00540400001674
1064L	00540400001675	00560000001710	00200000077700	00540600001707
1070L	04005000001045	00412610001447	01005000000445	23550610000633
1074L	00500630001623	00004000000431	41214630037777	00641400001675
1100L	00200000000621	00517000037740	00004000001027	00500770037740
1104L	00500610001576	00500620037740	00641400101655	40200000000003
1110L	00410620000030	23540610037777	41214610037774	02345160000000
1114L	02350160200000	47220160000000	02356160000000	02356160000000
1120L	47220160000000	22347160000000	00004000001021	00600770000621
1124L	00450760001571	01333231600000	00540400002020	00500770001701
1130L	00200000000621	00540601701660	00200000000617	00412770000000

1134L	00540770001701	00540770001705	00560000037777	00100000001630
1140L	41221260000001	21311260001570	00540600001674	00540600001675
1144L	00516000001563	00200000001070	01345160000000	01356160000000
1150L	01356160000000	01354160000002	01104200000052	00540200000206
1154L	005402000002027	00500000001537	00200000001311	12221040001270
1160L	01220050001204	00500000001531	00200000001254	00200000006200
1164L	02220040001274	00200000001303	00500000001532	00200000001254
1170L	00200000006200	02220040001274	00200000001303	01220040001070
1174L	00500000001533	00200000001254	002000000010000	02220040001274
1200L	00200000001303	42220050000602	00200000001070	03221040001231
1204L	00500000001534	00200000001254	01354160000001	00200000007000
1210L	02220040001274	00200000001303	00500000001536	01355160000001
1214L	01353160000002	00200000001254	04005000001225	44220050000002
1220L	40200000000002	00000000000000	00412000001070	37550000004213
1224L	01233770004214	00540600001705	00233770000000	00231200000000
1230L	01231770000000	00500000001535	00200000001254	01354160000002
1234L	01355160000001	002000000010600	422200400000003	01352160000002
1240L	40200000000002	01353160000001	21347160000000	43221160000003
1244L	41221160000005	402000000037776	01351160000001	01356160000000
1250L	402000000037772	02220040001274	00200000001303	00200000001070
1254L	00552000001267	00540000001527	00500010002024	00544010001525
1260L	41221260000002	402000000037777	21311260001306	00641000001535
1264L	00200000001267	00200000001311	52221040000002	00200000001163
1270L	41221260000002	402000000037777	21311260001307	02226040001070
1274L	00501000001545	41221260000002	402000000037777	00541000001541
1300L	21311260001310	02220040001670	00200000000000	00552000001302
1304L	00501000001543	00200000001275	00012000001523	00014000001547
1310L	00006000001540	00552000001400	01104200000052	00540200037740
1314L	00500570001607	00500560001624	00500370001624	00500360001625
1320L	00500770001626	00500760001626	00001000000000	21340160001401
1324L	10221160001403	43221160000003	41221160000006	40200000037775
1330L	43214770000002	12226040001400	01351160000001	00200000001323
1334L	00011770001601	006400000002040	00200000001322	16500740002041
1340L	21340160001402	43221160000003	41221160000006	40200000037776
1344L	43214770000002	12226040001400	01351160000001	00200000001340
1350L	00011770001601	00500730001627	37510130002040	17550130037773
1354L	37550740001374	17550130001374	00500100002042	43214730000002
1360L	40200000000004	00407101302041	00500100037711	43214730037776

1364L	00641100002041	40200000000005	43214760000002	12226040001400
1370L	01351160000001	00200000001340	00011760001601	00560000002042
1374L	00410000046600	00410740000400	36641200002040	00200000001340
1400L	00200000001266	00020000002040	01004000002040	01355160000002
1404L	01354160000001	00200000001322	33444740255151	33234640255151
1410L	33334463402330	00000000000000	00000000000000	00000000000000
1414L	17000000000000	33334347402330	33306347402330	33476351402330
1420L	33332563402330	33334751637001	33334751637002	33333331456263
1424L	33333321245162	33332143215144	77000033000033	77464725512163
1430L	25602551514651	17000000000000	61616143256525	43740134404146
1434L	22603145632551	51644763616161	37020000000000	17606060606060
1440L	54545441462274	60606060606060	43256525430353	60606060606060
1444L	73455037010000	17000000000000	00330000170000	34402545246123
1450L	46444743256325	60606060606060	34402545246131	45234644474325
1454L	63256060606060	34402545246151	25636451456060	60606060606060
1460L	34406263215163	60606060606060	60606060606060	45466025672523
1464L	64632537020000	17000000000000	34405125626321	51636060606060
1470L	60606060606060	25672523646331	46453702000000	17000000000000
1474L	43654340007300	00331600170000	60606060606060	62632151636063
1500L	31442560600000	33000037010000	17000000000000	60606060606060
1504L	63466321436063	31442515000000	00164431453702	17000000000000
1510L	60606023645151	25456360626321	63646237010000	43256525430053
1514L	73503701000000	73455037010000	77404040404040	41462260446443
1520L	63314743256725	51604647254540	40404040401700	37010000000000
1524L	54545441462274	21000005606060	34606060606060	47302162254001
1530L	17000000000000	47302162254001	47302162254002	47302162254003
1534L	47302162254004	21636321233060	25672523646325	44632362214663
1540L	60606025452460	00000000000000	00000000000000	60602346444743
1544L	25632517000000	31452346444743	25632517000000	77606060606060
1550L	60606060606060	62706263254460	63214725605125	21246025515146
1554L	51170000000000	00000000000340	00000000000342	00000000000352
1560L	00000000000353	00000000000354	00000000000355	00377760376000
1564L	00000003776000	00040000040001	00010000001411	00006000001427
1570L	00012000001516	00000000000074	00000000000144	00000000002640
1574L	00000000000012	00110000300000	00006000040001	77777777777777
1600L	77770077777777	60777777777777	77000777777777	77777007777777
1604L	00000000000000	60777007777777	40000007740000	77777777740000
1610L	00272000001700	02000000000000	56000040002036	22340160001611

1614L	22341160001611	04002073241664	00014000001432	00016000001440
1620L	00010000001462	00012000001441	00006000001510	000520000040016
1624L	00001777777777	77770000077777	00012000040000	00000000040001
1630L	00000000000000	00000000000000	00000000000000	00000000000000
1634L	00000000000000	00000000000000	00000000000000	00000000000000
1640L	60777777777777	77000777777777	77777007777777	00000000000000
1644L	60777007777777	40000007740000	77777777740000	00006000040003
1650L	00000000000000	00052000040025	00000000000460	21000005606060
1654L	00000000000000	00000000000000	00000000000000	00000000000000
1660L	00000000000000	00000000000000	00000000000000	00000000000000
1664L	00000000000000	20000000000000	00000000000000	00000000000000
1670L	00000000000000	00000000000000	37010000000000	17000000000000
1674L	00000000000000	00000000001045	21175371203243	07325042430416
1700L	00540600001675	00000000000001	00000000000460	00000000000000
1704L	00000000000000	00000000000000	00000000000000	00000000000000
1710L	00000000000000	00000000000000	76161245062401	02352170000001
1714L	00003570501741	54546054545454	54545454543701	00000000000000
1720L	65612161101415	65612161101415	00400000000000	00000000000012
1724L	00000000000000	00000000000000	00000000001400	44464531600261
1730L	02000170000000	77777777740000	00001777777777	75673067777752
1734L	00401000000000	77000777777777	02000033333333	00000000000000
1740L	27300110011011	25300220022002	11300000007556	77777777770077
1744L	77770077777777	60777777777777	77000777777777	77777007777777
1750L	00000000000000	60777007777777	40000007740000	77777777740000
1754L	77777774001777	76017017777777	77777774377777	77777777776017
1760L	60777777777777	77777007777777	60777007777777	00007777777777
1764L	77777777770077	00000000001000	00001777777777	77777777740000
1770L	00000000001124	00006000004003	00000000000000	00052000040025
1774L	00440000040220	11400010000000	15400010000000	00010000040000
2000L	6247000001624	03776000040000	00020000040000	00032000040000
2004L	00000000001553	00542000040000	00000000000005	00000000000001
2010L	01000160000000	00000000000000	01000250000000	01000230000000
2014L	03000160000000	02000160000000	03000160000000	01000240000000
2020L	00000000000000	00000000011420	00000000030000	00000000000000
2024L	21000005606060	00000000000000	00000000000001	00000000000000
2030L	04000160000000	05000160000000	06000160000000	07000160000000
2034L	41321210000022	01325210000002	41321210000021	01325210000001

付 録 1-2

ENDDISPLAY

attachProcedure END DISPLAY:
code

00231440000000	00640010001701	40200000000056	04347160000000
00412010000001	04346160000000	64341160000004	41221160000010
04347160000000	40200000037776	02000000027000	00000000000000
43221160000004	04356160000000	40200000037770	00001000000000
04351160000001	04345160000000	04356160000000	64340160000004
00412010000000	43221160000005	40200000037776	00310000027000
41221160000013	00000000000001	40440010037776	40641010037776
00000000000001	04351160000001	40200000037765	00004000027000
00000000000000	04352160000001	64340160000004	41221160000010
00412010000000	40200000037776	02000000027000	00000000000006
43221160000004	40641010037776	00000000000000	04345160000000
00410010000001	04350160000001	04356160000000	04356160000000
04356160000000			
00233440000000;			

end;

付 録 1-3

MT DISPLYREADY

```
begin
code
03345160000000 03356160000000 03356160000000 03356160000000
63341160000012 41221160000005 43221160000002 40200000037776 03351160000001
40200000037772 03356160000000 03356160000000 03356160000000 402000000000145
40310000000001

00200000000001 01226230000000 00011400005201 00560000027106 00020000077721 00501410027122 00560000027116
00006000077761 23342160027124 00500650027121 01221160027047 03221160027040 10221160027016 00200000027012

00601210027031 00500210037722 22653630000073 40200000000003 22454630037722 40200000037775 00500650037722
22454650037763 00004000027102 21333230500000 00200000027006 00004000027102 00553000037700 01104000000014

00700000027143 00061345737700 01311260027131 00200000027000 00500000000072 03351160000001 40544000000001
00000000000000 03214650027012 01311260027130 00200000027031 00500210037740 00500210037740 00004000027102

43214620000003 23342160027125 00200000027056 23342160027124 21660300227150 00120000077764 00000000000000
00500010437776 05640230037701 00200000027070 00500010437777 21660610027076 00001002300000 00200000027076

00500010437777 00011020005201 02105010000006 05640230037701 00200000027071 00601020027066 06411640227150
07554640127126 21330230127126 00200000027011 01221230000000 46221230000002 00200000027102 17224000067102

00000000000000 00000000000074 00000000000037 01000000000000 77777777777700 777777777740000 00001777777777
17000000000000 00002000040001 00120002400050 00000000040000 00006000040000 00777777777777 777777777740000

00120000027150 00120000027220 00000000027150 00000000027220 00010000027132 00010000027143 37010000000000
75754463754546 33037551252124 75255151465175 37010000000000 75151633000075 44637524316247 43217075254524

17000000000000
00001000000000
end;
```

付 録 1-4

```

PRINTF
attach procedure PRINTF(N,M);
  value M,N; real N,M;
  begin integer I,J;
    I:=FIX(N); J:=(N-I)*10;
    if I<1 then
      begin
        REPX(J); PRINT('B-.XDB',M); go to END;
      end;
    REPX(I,J);
    if J<1 then
      begin
        PRINT('B-XZB',M); go to END;
      end;
    PRINT('B-XZ.XDB',M);
  END;
end;
end;

```

付 録 1-5

PLOT

```

attach procedure PLOT(X,Y,RITU,PO,S,C);
  value X,RITU,PO,S; integer S,C; real X,Y,RITU,PO;
  begin integer W,Z,N,M,L,I;
    switchG=A,B; switchGG=AA,BB;
    procedure PRINTA(W);
      value W; integer W;
      begin switchH=K,R;
        go to HIW;
        K:PRINT('+++++++'); go to E;
        R:PRINT('#####');
      E: end;
    procedure PRINTN(W);
      value W; integer W;
      begin switchH=K,R;
        go to HIW;
        K:PRINT(''); go to E;
        R:PRINT('');
      E: end;
    PRINT('ZD',Y); PRINT('ZD',X);
    Y:=RITU+PO; N:=Y+0.5;
    DA:=N+10; L:=N-10;
    if N<0 then begin N:=N+PO; go to DA; end;
    if N>99 then begin N:=N-PO; go to DA; end;
    go to G(S);
    A:=C=0; W=1; go to P;
    B:=C+1; if C=10 then begin C:=0; W=1; go to P; end; W=2;
    P: if M=0 then goto D;
      for I:=1 to plunt11M do
        PRINTA(W);
    D: if L=0 then go to GGLW;
      PRINT('');
    if L-1=0 then go to LL;
      for I:=1 to plunt11L-1 do
        PRINTN(W); LL: PRINT('');
    Z:=10-(L+1);
    if Z=0 then go to E;
      for I:=1 to plunt11Z do
        PRINTN(W); go to E;
    BB:PRINT('#####'); go to E;
    AA:PRINT('+++++++');
    E: Z:=10-(M+1);
    if Z=0 then go to I;
      for I:=1 to plunt11Z do
        PRINTA(W);
    I: end;
  end;
end;

```

附錄 1-6

```

attach procedure MREG1;

begin
  comment SELECTION OF SIGNIFICANT FACTORS BY MEANS OF SECONDARY HYPOTHESES
    IN MULTIPLE REGRESSION ANALYSIS,

    INPUT DATA (in order)

      (utilizing 'PUT' and 'GET')

    S: S=1 for the case with constant, S=0 without constant term,
    P: number of input factors except Y,
    Q: number of factors including T-terms,
    N: number of data groups,
    D: number of T-terms,
    TA: values of originated points of T-terms (integer array)
      If not exist then D=0 and TA is not necessary,
    I1: for instance (D=2 case)
      1st factor X1
      2nd factor X1*X1
      3rd factor log(X1*X2)*X3
      4th factor sqrt(X2)
      5th factor ln((1+T)/(1+T1))
      6th factor ln((1+T)/(1+T2))

      then 1,0,1,1,0,1,2,37,3,0,2,38,0,39,0,39,0,

    If Q=P+D then 100, insted of 1,0,2,0,3,0,.....,39,0,39,0,39,0,
    LS: If log(Y) then 1, sqrt(Y) then 2, else 0,
    If D=0 or D1 is 1,2,3,..... then D1 is not necessary,
    DY: for the case of corrected Y 1, else 0,
    DYA: corrected Y array (size N) If DY=0 then DYA is not necessary,
    INN: starting point of T-term,
      If IN is not necessary then 0, FEB 5th 1959 is 590205,
    IN: integer array (size P+1)
      zero point for each variable,
      If INN is 0 then IN is not necessary,
    X1: data array for D1,P factors and Y
      (D1 is the date for instance JAN 31st 1961 is 610131),
      for the last of data 0,0,0,0, is necessary;

    integer INN,DY,LS,RP,C,D,I,J,K,L,M,N,P,Q,S,X1,T,U,V,I1,J1,A1,T1,T2;
    real ST,R1,R2,R3,SR, SE, FE, VE, FO, F, VR, SRR;
    array IN,W6,X1,TA,X,Y,W[1:3],SA[1:30],1:301,W7[1:200],DYA[1:300],GA[1:31];
    integer array DI,DIA[0:300],GC,NA[1:301],PA[1:6C,1:7],SSA[1:3C];
    Boolean BI,BL,BL1;

    external PRINTR,PRINT1,PRINTF,PRINTS,CRLF,SPACE,IA2PUT,RA1PUT,RA2PUT;
    procedure PRIA2(S,M,N,A,NA);
      value M,N;
      integer M,N; integer array NA; array A; string S;
      begin
        integer I,J;
        PRINT('///10B'); PRINT(S); PRINT('///'); PRINT('6B');
        for I=1 step 1 until N do
          begin
            if I=9 then PRINT('//6B');
            PRINT('2B'('ZD')'7B',NA(I));
          end;
        PRINT('///');
        for I=1 step 1 until M do
          begin
            PRINT(''('ZD')'',NA(I));
            for J=1 step 1 until N do
              begin
                if J=9 then PRINT('//4B');
                PRINT('B'-.5DN+2DB',A(I,J));
              end;
            PRINT('///');
          end;
        PRINT('/////');
      end;
end;

```

```

procedure PRIA1(S,N,A,NA):
  value N;
  integer N; integer array NA; string S; array A;
  begin
    integer I;
    PRINT('///3B'); PRINT(S); PRINT('///');
    for I=1 step 1 until N do
      begin
        PRINT('('ZD')',NA[I]);
        PRINT('B-.5DN+2D/',A[I]);
      end;
    PRINT('/////');
  end;

integer procedure DATE(DA):
  value DA;
  integer DA;
  begin
    integer U,T1,T2,T3,T4,T5;
    integer array A[1:12];
    T1:=DA+10000;
    T2:=T1-53;
    T3:=DA-T1+10000;
    T4:=T3+100;
    T5:=T3-T4+100;
    U:=if T2+4*4=T2 then 29 else 28;
    A[1]:=0; A[2]:=31; A[3]:=31+U; A[4]:=62+U;
    A[5]:=92+U; A[6]:=123+U; A[7]:=153+U; A[8]:=184+U;
    A[9]:=215+U; A[10]:=245+U; A[11]:=276+U; A[12]:=306+U;
    DATE:=T2+365+T2+4+A[T4]+T5;
  end;

procedure PRDATE(DA):
  value DA;
  integer DA;
  begin
    integer T1,T2,T3,T4;
    T1:=DA+10000;
    T2:=(DA-T1+10000);
    T3:=T2+100;
    T4:=T2-T3+100;
    PRINT('##2Z,2(B2Z)',T1,T3,T4);
  end;

procedure TRANA1(L,M,N,A,B):
  value L,M,N;
  integer L,M,N; array A,B;
  begin integer I;
    for I=1 step 1 until N do
      B[I+1]:=A[L+I-1];
    end;

procedure TRANA2(AI,AJ,BI,BJ,M,N,A,B):
  value AI,AJ,BI,BJ,M,N;
  integer AI,AJ,BI,BJ,M,N; array A,B;
  begin integer I,J;
    for I=1 step 1 until M do
      for J=1 step 1 until N do
        B[Bi+I-1,BJ+J-1]:=A[AI+I-1,AJ+J-1];
      end;
    end;

integer procedure CRSP(I):
  integer I;
  if I+8*8+1=1 ^ 1#1 then begin PRINT('/'); SPACE(12); end;

procedure PR(A):
  value A;
  integer A;
  begin PRINT('X'); PRINT('ZDB',A); end;

procedure CH(S):
  string S;
  begin
    PRINT('****CHECK#'); PRINT(S); PRINT('****'); PRINT('S/'); go to END;
  end;

```

```

RP=1;
SZ: READ(S,P,Q,N); BI=FALSE;
PRINTS('START-'); PRINT('2D',RP); CRLF(5);
If ~(S=1 V S=0) then CH('S');
M:=Q+S+1;
A1:=Q;
If P=0 then go to END;
READ(D);
If P<0 V P>40 then CH('P');
If D<0 V D>40 then CH('D');
If D=0 then READARRAY(TA,A1-D+1,Q);
begin
PRINT('###NO.#####FACTOR'); CRLF(2); PRINT('17(-#)');
IA2PUT(PA,60,7,0);
If S=1 then
begin PRINTS('###1'); SPACE(9); PRINT('CONST.');
```

CRLF(2); end;

R1=0;

for I=1 step 1 until Q do

begin

Integer J;

procedure G; begin PA(I,J)=I; J=J+1; R1=I; go to LLL; end;

J:=1;

PRINTI(2, If S=1 then (I+1) else I); SPACE(8);

LLL: READ(I);

If R1<30 A11>30 A11<36 then begin SPACE(5); end;

If I1>30 A11<36 then

begin PRINT('###GROUP-2Z//',I1);

GC(I)=PA(I,J)=I; go to L1;

end;

If I1=100 then go to LA1;

If I1<0 V I1>39 then CH('I1');

If I1=37 then begin PRINT(')=LOG'); G; end

else If I1=38 then begin PRINT(')=SQRT'); G; end

else If I1=39 then

begin

PRINT('T='); PRINT('42',TA(I)); PRINT(')##');

SSA(I)=TA(I); G;

end

else If I1=30 then

begin

If R1=0 then

begin PRINT('('); R1=I1; end else PRINT(')');

PR(I); R1=I1; G;

end;

L1: R1=0;

end;

OUT:

go to LA2;

LA1:

for K=1 step 1 until P do

begin

If K>1 then begin PRINTI(2, If S=1 then (K+1) else K); SPACE(8); end;

PRINT('('); PA(K,1)=K; PR(K); PRINT(')');

CRLF(2);

end;

PA(I,1)=100; BI=TRUE;

LA2:

READ(LS);

If ~(LS=0 V LS=1 V LS=2) then CH('LS');

else

begin

PRINT('###0'); SPACE(9);

If LS=1 then PRINT('LOG(Y)/////');

else If LS=2 then PRINT('SQRT(Y)/////');

else PRINT('Y/////');

end;

end;

for I=1 step 1 until N do

O(I)=1;

READ(DY); If ~(DY=1 V DY=0) then CH('DY');

If DY=1 then READ ARRAY (DYA,1,N);

READ(INN); If INN=0 then READARRAY(1N,1,P+1);

BI=1; DY=0 then FALSE else TRUE;

RA2PUT(SA,31,31,0); RA1PUT(Y,31,0);

SPACE(15); PRINTS('*****DATA*****'); CRLF(2);

SPACE(12);

for I=1 step 1 until Q+S do

begin CRSP(I); PRINTS('##('); PRINTI(3,I); PRINTS(')##');

end;

If (Q+S)+8*8=Q+S then SPACE(12);

```

PRINTS('####(#0)'); CRLF(2);
for L=1 step 1 until N do
begin
array ANS(1:31,1:31),WK(1:30);
Boolean Z; Z=PA(1,1)=100;
if D≠0 then
begin READ(DI(L));
if DI(L) < 50000 then begin PRINT('CHECK-DI-'*Z/'*,L);
goto END; end;
if L=1 then VE=DATE(1; INN≠0 then INN else DI(1))-1;
DIA(L)=DATE(DI(L))-VE;
end;
READARRAY(X1,1,P+1);
if INN≠0 then
for I=1 step 1 until P+1 do
X1(I)=X1(I)-IN(I);
if BL1 then X1(P+1)=DIA(L);
if Z then go to E2;
for K=Q-D+1 step 1 until Q do
begin
real Q;
Q:=(1+DIA(L))/(1+TA(K));
W6(K)=if Q≤1 then 0.0 else LN(Q);
end;
for I=1 step 1 until Q do
begin
real A;
Boolean L1;
A=PA(I,1); L1=A=39;
if L1 then go to E1;
X(I)=1;
for J=1 step 1 until 7 do
begin
Boolean L2,L3;
Integer S;
S=PA(I,J);L2=S>30&S<36;
if L2 then begin G(I)=S; go to EX; end;
X(I)=if S=37 then 0.43429448*LN(X(I))
else if S=38 then SQRT(X(I)) else X(I)*X(I);
end;
E1: X(I)=W6(I);
EX: WK(I)=X(I);
end;
go to E3;
E2:
if Z then
begin
for I=1 step 1 until P do WK(I)=X(I);
for I=Q-D+1 step 1 until Q do WK(I)=W6(I);
end;
E3:
if S=1 then X(1)=1-0;
TRANA(1,1+S,Q+1,WK,X);
W7(L)=X(Q+S+1)=if LS=1 then 0.43429448*LN(X1(P+1))
else if LS=2 then SQRT(X1(P+1)) else X1(P+1);
U=Q+S+1;
PUT(L,X);
PRINT('422B',L); PRINT('('');
PRINT('4Z',if D=0 then L else DIA(L)); PRINT(')');
for I=1 step 1 until Q+S+1 do
begin CRSP(I); PRINTR(S,X(I)); end;
CRLF(2);
for I=1 step 1 until U do
begin
Y(I)=Y(I)+X(I);
for J=1 step 1 until U do
ANS(I,J)=X(I)*X(J);
end;
for I=1 step 1 until U do
for J=1 step 1 until U do
SA(I,J)=SA(I,J)+ANS(I,J);
end;
CRLF(2);
SPACE(4); PRINTS('(#SUM#)');
for I=1 step 1 until U do
begin CRSP(I); PRINTR(S,Y(I)); end;
CRLF(1);

```

```

SPACE(4); PRINTS('*(#MEAN)*');
for I=1 step 1 until U do
begin CRSP(I); PRINTR(5,Y[I]/N); end;
CRLF(5);
end;

begin
array SA2(1:31,1:31);
TRANA2(1,1,1,1,U,U,SA,SA2);
for I=1 step 1 until M-1 do NA[I]=1;NA[M]=0;
PRIA2('SUM-OF-PRODUCT#MATRIX',U,U,SA2,NA);
Q=Q+S+1;
for I=1 step 1 until U do
begin
for J=1 step 1 until U do
begin
real P,Q,R;
P=SA2[I,J]; Q=Y[I]; R=Y[J]; SA2[I,J]=P-Q*R/N;
end;
end;
PRIA2('COVARIANCE#MATRIX',Q,Q,SA2,NA);
SPACE(20); PRINT('CORRELATION#MATRIX'); CRLF(2);
SPACE(9);
for I=1 step 1 until Q do
begin
if I=13 then begin CRLF(1); SPACE(9); end;
PRINT('');
if I=Q then PRINT('##Y##') else PRINT(2,I); PRINT('')##';
end;
CRLF(2);
for I=1 step 1 until Q do
begin
real R,S;
R=SA2[I,I]; PRINT('');
if I=Q then PRINT('##Y##') else PRINT(2,I); PRINT('');
for J=1 step 1 until Q do
begin
S=SA2[I,J];
if R=QVS=0 then begin if J=13 then SPACE(7); PRINTF(1,3,0) end
else
begin
if J=13 then PRINT('/7B');
PRINTF(1-3,SA2[I,J]/SQRT(R*S));
end;
end;
END; CRLF(2);
end;
end;
CRLF(5);
begin array ANS(1:31,1:31,WK4,WK5(1:31)); real S3,S2;
NA[M]=0; S2=SA(M,M); T=M-1;C=M-1;
SPACE(7); PRINTS('SR#####F#####SV##(FACTOR)'); CRLF(2);
K1=0; D=M-1; PRINTR(5,S2); PRINT('#####TOTAL'//');
START;
for I=1 step 1 until T do
WK5[I]=SA[I,M];
L=0; M=T+1; TRANA2(1,1,1,1,M,M,SA,ANS);
begin array WK1(1:450);
for K=1 step 1 until M-1 do
begin
real S;
array WK2,WK3(1:30);
S=ANS(K,K);
for I=K+1 step 1 until M do
WK2[I]=ANS(K,I)/S;
for I=1 step 1 until M-1 do
WK3[I]=ANS[I,K];
WK3[K]=1;
for I=1 step 1 until M-1 do
for J=1+K step 1 until M do
ANS[I,J]=if I=K then WK2[J] else (ANS[I,J]-WK2[J]*WK3[I]);
for J=1 step 1 until K do
WK1[L+J]=ANS[J,M];
L=L+K;
end;
I=(M-1)*M/2; ST=0;
for I=1 step 1 until M-1 do

```

```

begin S3:=WK4(1)=WK1(I1-M+I+1); ST:=ST+S3*WK5(1); end;
if T=1VD=-1 then
begin
  CRLF(5); SPACE(17); PRINT('ADV#LAST'////);
  SPACE(7); PRINT('SR#####DF###F#####SV'////);
  PRINTR(5,ST); PRINTI(3,M-1); VE:=(S2-ST)/(N-M+1);
  PRINTF(3.3,ST/(M-1)/VE); PRINT('###');
  for I=1 step 1 until M-1 do
    PRINT('2D,'#,NA(I));
    PRINT(')'); CRLF(1);
  PRINTR(5,S2-ST); PRINTI(3,N-M+1); SPACE(13); PRINT('E'); CRLF(1);
  PRINTR(5,S2); PRINTI(3,N); SPACE(10); PRINT('###TOTAL'); CRLF(2);
  PRINT('RVE='); PRINTR(5,SQRT(VE)); CRLF(2);
  go to OUT;
end;

L:=0; SE:=S2-ST;
for K=M-1 step -1 until 1 do
begin
  real SR;
  SR:=0;
  for I=1 step 1 until K do
    SR:=SR+WK1(I1-L-K+I)*WK5(1);
  VR:=if K=M-1 then ST else (ST-SR);
  VE:=SE/(N-M+1);
  ST:=SR; FO:=VR/VE;
  F:=(1.4+1.02/(N-K))*4;
  PRINTR(5,SR); PRINTF(3.3,FO); SPACE(3);
  if K=M-1 then for I=1 step 1 until K do PRINTI(2,NA(I))
    else PRINTI(2,NA(K+1));
  if K=M-1 then
  begin
    PRINT('*( ');
    for I=1 step 1 until K do
      PRINTI(2,NA(I));
    PRINT(')');
  end;
  if FO=F ^ K*M-1 then
  begin
    PRINT('***'); T=K+1; D=D-1;
    go to DIM;
  end;
  CRLF(1); L=L+K; T=K+1; if K=M-1 then D=D-1;
  if K=1 then begin T=1; go to DIM; end;
end;
end;
CRLF(1);
DIM:
for I=1 step 1 until M do
begin
  SA[T+1,I]:=SA[M,I];
  SA[I,T+1]:=SA[I,M];
end;
NA[T+1]:=0;
if D=-1VT=1 then go to START;
CRLF(2); M:=T+1;
for I=1 step 1 until M do
  X[I]:=SA[M-1,I];
for I=M-2 step -1 until 1 do
  for J=1 step 1 until M do
    SA[I+1,J]:=SA[I,J];
  for I=1 step 1 until M do
    SA[I,I]:=X[I];
  for I=1 step 1 until M do
    W[I]:=SA[I,M-1];
  for I=1 step 1 until M do
    for J=M-2 step -1 until 1 do
      SA[I,J+1]:=SA[I,J];
    for I=1 step 1 until M do
      SA[I,1]:=W[I];
  NA[M]:=NA[M-1]; SSA[M]:=SSA[M-1];
  for I=M-2 step -1 until 1 do
    begin NA[I+1]:=NA[I]; SSA[I+1]:=SSA[I]; end;
  SSA[1]:=SSA[M];
  NA[1]:=NA[M]; NA[M]:=0;
  go to START;
OUT: for I=1 step 1 until D do
  if NA[A1-D+1]=0 then TA[A1-D+1]:=0;

```

```

OUT2:
  if T=0 then
    begin CRLF(5);PRINTS('NONE'); CRLF(5); go to E1; end;
    CRLF(5);
    PRIA2('SA*LAST',M-1,M,SA,NA);
    if D=-1VT=1 then go to OUT3;
    K1:=1;T:=M-1; go to START;
  OUT3:
    PRIA1('SOLUTION',M-1,WK4,NA);
    PRINTS('###NO.###');
    if ~B1 then
      begin
        PRINT('DATE####');
        for I:=31 step 1 until 35 do
          PRINT('GROUP-2D####',I);
        end;
      PRINT('OBS.##Y)##(CUL.##Y)');
      PRINTS('##(CY-OY)'); CRLF(2);
      for I:=1 step 1 until N do
        begin
          array R1A[31:35];
          real P; integer K;
          for K=31 step 1 until 35 do R1A[K]:=0;
          GET(I,X);
          P:=W7[I]; PRINT('('); PRINTI(3,I); PRINTS(')');
          if ~B1 then PRDATE(DI[I]);
          R1:=0;
          for J:=1 step 1 until M-1 do
            begin
              integer L; real EL;
              L:=NA[J];
              EL:=WK4[J]*X[L]; R1:=R1+EL;
              if ~B1 then
                for K=31 step 1 until 35 do
                  R1A[K]:=R1A[K]+(if GC[L]=K then EL else 0.0);
            end;
          L5:
          if ~B1 then
            for K=31 step 1 until 35 do
              PRINTR(5,R1A[K]);
            PRINTR(4,P); PRINTR(4,R1); PRINTR(4,R1-P); CRLF(1);
          end;
          CRLF(5);
          PRINT('8B*ESTIMATOR'//');
          PRINT('Y=');
          if S=1ANA[1]=1 then
            begin PRINTR(5,WK4[1]);
              U:=1;
            end else U:=0;
          for I:=1+U step 1 until M-1 do
            begin
              integer R,Q; R:=NA[I]-S;
              Q:=0; PRINT('#+#');
              for J:=1 step 1 until 7 do
                begin
                  integer Q,P; P:=PA[R,J]; if P=0 then go to EN;
                  if P>30 ^ P<36 then
                    begin PRINT(''); go to EN; end;
                  if J=1AP=30 then PRINT('');
                  if P=30 v P=100 then begin PRINT('(-.5Dn+2D)',WK4[I]); end;
                  if J=1 then PRINT('');
                  if P=37 then PRINT(')=LOG#'); else
                    if P=38 then PRINT(')=SQRT#'); else
                      if P=39 then PRINT(')=LN((1+T)/(1+.4Z'))',SSA(R))
                        else
                          begin
                            PRINT('X'2Z',R);
                            if INN=0 then
                              begin
                                if IN[R]<0 then PRINT('++') else PRINT('--');
                                PRINT('-.5Dn+2D',IN[R]);
                              end;
                            PRINT('');
                          end;
                end;
              Q:=P;
            end;
          EN:

```

```

      if I+3*3=1 then begin CRLF(1); SPACE(5); end;
    end;
    CRLF(2);
  end;
E1:
  CRLF(5);
  PRINTS('END-'); PRINTI(1,RP); PRINT(' '); RP:=RP+1;
  go to SZ;
END:
  if -(S=0^P=0^Q=0^N=0) then PRINTS('(DATA#CHECK)');
  PRINTS('(END)');
end;
end;
end;

```

付 録 1-7

```

procedure PNP(M);
  value M; real M;
  code
    40500050037776      40064050000031      00542050037705      40040050000030
    00026050000451      40712000037771      00002000037701      02104000000006

    00500010037705      00412020000073      01104020000044      00412610020002
    01104610000001      02105010000003      01105020000003      41213610037776

    00500010037702      02105000000006      40540010000007      61317240000005
    41221240000010      46221240000002      40200000037776      57224040037774

    40002000000001      00000000000000      20224000000000      20020000000000;

```

付 録 1-8

PSEN

```

attach
  procedure PSEN(X1,Y1,X2,Y2);
    value X1,X2,Y1,Y2; real X1,X2,Y1,Y2;
    begin real PXXX,PYYY;
      external integer OCTAL; external CGETR,CPUTR;
      CGETR(-1,PXXX); CGETR(-2,PYYY); CPUTR(-1,X2); CPUTR(-2,Y2);
      if PXXX+X1VPPYY+Y1 then
        PUNCH('A',41,A',EQUIV('U'),EQUIV('F'),
          OCTAL(PXXX),OCTAL(PYYY),OCTAL(X1),OCTAL(Y1),EQUIV('D'));
        PUNCH('A',41',EQUIV('F'),OCTAL(X1),OCTAL(Y1),OCTAL(X2),OCTAL(Y2));
      end;
    end;
end;

```

付録 1-9

PEN

```

attach
procedure PEN(X1,Y1,X2,Y2,R);
value X1,X2,Y1,Y2,R; real X1,X2,Y1,Y2,R;
begin
  real XC1,XC2,YC1,YC2,PXXX,PYYY,A,A1,A2,B,B1,B2,C1,X,Y,T,T1,XX1,XX2,YY1,YY2;
  procedure G1; begin X=XX1; Y=YY1; end;
  procedure G2; begin X=XX2; Y=YY2; end;
  procedure CH; PRINT('CHECK for DATA-R');
  external integer OCTAL; external CGETR,CPUTR;
  if X1=X2 then
    begin A:=(Y1+Y2)/2; YY1=YY2=A;
      B1:=X1+2+A+2+Y1+2-2*Y1*A-R+2;
      if X1+2-B1<0 then CH; B2:=SQRT(X1+2-B1);
      XX1=X1+B2; XX2=X1-B2; T=Y2-Y1; T1=XX1-X1;
      if R*T+T1>0 then G1 else G2;
    end else
    if Y1=Y2 then
      begin A:=(X1+X2)/2; XX1=XX2=A;
        B1:=X1+2+Y1+2+A+2-R+2-2*A*X1;
        if Y1+2-B1<0 then CH; B2:=SQRT(Y1+2-B1);
        YY1=Y1+B2; YY2=Y1-B2; T=X2-X1; T1=YY1-Y1;
        if R*T+T1<0 then G1 else G2;
      end else
      begin A:=(X2+2+Y2+2-X1+2-Y1+2)/2/(X2-X1);
        B:=(Y1-Y2)/(X2-X1); A1:=B+2+1; B1:=2*(A*B-X1*B-Y1);
        C1:=A+2+Y1+2+X1+2-R+2-2*A*X1; A2:=-B1/2/A1;
        if A2+2-C1/A1<0 then CH; B2:=SQRT(A2+2-C1/A1);
        YY1=A2+B2; YY2=A2-B2;
        XX1=A+B*YY1; XX2=A+B*YY2;
        T=X2-X1; T1=YY1-Y1+B*(XX1-X1);
        if T*T1+R<0 then G1 else G2;
      end;
  XC1=X1-X; YC1=Y1-Y; XC2=X2-X; YC2=Y2-Y;
  CGETR(-1,PXXX); CGETR(-2,PYYY); CPUTR(-1,X2); CPUTR(-2,Y2);
  if PXXX+X1+PYYY+Y1 then
    PUNCH('A,A,41,A',EQUIV('U'),EQUIV('F'),
      OCTAL(PXXX),OCTAL(PYYY),OCTAL(X1),OCTAL(Y1),EQUIV('D'));
    PUNCH('A,41',if R>0 then EQUIV('R') else EQUIV('L'),
      OCTAL(XC1),OCTAL(YC1),OCTAL(XC2),OCTAL(YC2));
  end;
end;

```

付 録 1-10

POUTR

```

attach
procedure POUTR(A,S,N,VV):
  value A,S,N,VV: real A,VV; integer N,S;
begin
  procedure CGETR(N,R):
    value N: integer N; real R;
    code 4050000003777 00004000003564 00412020037560
        40450020037771 00500040077702 00540040077700;
  procedure CPUTR(N,R):
    value N,R: integer N; real R;
    code 00412000037560 40450000037772 40500020037774
        00540020077700;
  procedure UF: PUNCH('A,A',EQUIV('J'),EQUIV('F'));
  procedure DF: PUNCH('A,A',EQUIV('D'),EQUIV('F'));
  procedure DR: PUNCH('A,A',EQUIV('D'),EQUIV('R'));
  real procedure RAD(A):
    value A: real A;
    RAD:=A*3.1415926/180;
  procedure PUC(X,Y):
    value X,Y: real X,Y;
    begin
      integer procedure OCTAL(X):
        value X: real X;
        begin integer J,N,M;
          N:=0; M:=FIX(10*X+0.5);
          for J:=0 step 1 until 8 do
            begin N:=(M-M*8*8)*10!J+N; M:=M*8; end;
          OCTAL:=N;
        end;
      PUNCH('2I',OCTAL(S*(X*COS(RAD(A))+Y*SIN(RAD(A))));
        OCTAL(S*(-X*SIN(RAD(A))+Y*COS(RAD(A))));
    end;
  procedure MINUS:
    begin UF: PU(0,0); PU(0.1,0.5); DF: PU(0.1,0.5); PU(0.4,0.5); UF:
      PU(0.4,0.5); PU(0.5,0);
    end;
  procedure POINT:
    if S=6 then
      begin UF: PU(0,0); PU(0.1,0.2); DF: PU(0.1,0.2); PU(0.2,0.3); PU(0.3,0.2);
        PU(0.2,0.1); PU(0.1,0.2); PU(0.3,0.2); PU(0.2,0.3); PU(0.2,0.1);
        UF: PU(0.2,0.1); PU(0.5,0);
      end else
      begin UF: PU(0,0); PU(0.1,0.2); DR: PU(0,-0.1); PU(0,-0.1);
        UF: PU(0.1,0.2); PU(0.5,0);
      end;
  procedure CH:
    begin PRINT('CHECK#DATA-N#POUTR'5/'); go to OVARI; end;
  integer I,B,I1: real V,PXXX,PYYY;
  integer array D1[0:9],D2[0:2];
  V:=VV; if V=0 then D1[0]=1 else begin D1[0]=0; V:=-V; end;
  if V=10=1 then
    begin
      for I:=0,I+1 while 10.0!I=V do B:=if V=1 then I+1 else 0;
        D2[0]=1; D2[1]=FIX(B/10); D2[2]=8-D2[1]*10;
        I:=FIX(V*(if B<N then 10.0!(N-B) else 1/10!(B-N)));
      end else
      begin
        for I:=1,I+1 while 1/10.0!I>V do B:=1;
          D2[0]=0; D2[1]=FIX(B/10); D2[2]=8-D2[1]*10;
          I:=FIX(V*10.0!(B+N));
        end;
      if N=10 then CH;
      for I:=1 step 1 until N do
        begin D1[I]=FIX(I/10!(N-I)); I1:=I-D1[I]*10!(N-I); end;
      if D1[0]=0 then MINUS else begin UF: PU(0,0); PU(0.5,0); end; POINT;
      for I:=1 step 1 until N+2 do

```

```

begin
  procedure TEN;
  begin UF: PU(0,0); PU(0.1,0.1); DF: PU(0.1,0.1); PU(0.1,0.5);
        UF: PU(0.1,0.5); PU(0.25,0.5); DF: PU(0.25,0.5); PU(0.35,0.5);
        PU(0.4,0.4); PU(0.4,0.2); PU(0.35,0.1); PU(0.25,0.1); PU(0.2,0.2);
        PU(0.2,0.4); PU(0.25,0.5); UF: PU(0.25,0.5); PU(0.5,0);
        end;
  procedure PLUS;
  begin UF: PU(0,0); PU(0.1,0.5); DF: PU(0.1,0.5); PU(0.4,0.5); UF:
        PU(0.4,0.5); PU(0.25,0.3); DF: PU(0.25,0.3); PU(0.25,0.7);
        UF: PU(0.25,0.7); PU(0.5,0);
        end;
  procedure G; go to END;
  switch SW=S0,S1,S2,S3,S4,S5,S6,S7,S8,S9;
  if I=N+1 then
    begin TEN; if D2(I)=0 then MINUS else PLUS; end;
    go to if ISN then SW[D1(I)+1] else SW[D2(I)-N1+1];
  S0: UF: PU(0,0); PU(0.2,0.1); DF: PU(0.2,0.1); PU(0.3,0.1); PU(0.4,0.2);
        PU(0.4,0.8); PU(0.3,0.9); PU(0.2,0.9); PU(0.1,0.8); PU(0.1,0.2);
        PU(0.2,0.1); UF: PU(0.2,0.1); PU(0.5,0); G;
  S1: UF: PU(0,0); PU(0.25,0.1); DF: PU(0.25,0.1); PU(0.25,0.9);
        UF: PU(0.25,0.9); PU(0.5,0); G;
  S2: UF: PU(0,0); PU(0.1,0.7); DF: PU(0.1,0.7); PU(0.1,0.8);
        PU(0.2,0.9); PU(0.3,0.9); PU(0.4,0.8); PU(0.4,0.7); PU(0.1,0.2);
        PU(0.1,0.1); PU(0.4,0.1); UF: PU(0.4,0.1); PU(0.5,0); G;
  S3: UF: PU(0,0); PU(0.1,0.8); DF: PU(0.1,0.8);
        PU(0.2,0.9); PU(0.3,0.9); PU(0.4,0.8); PU(0.4,0.6); PU(0.2,0.5);
        PU(0.4,0.4); PU(0.4,0.2); PU(0.3,0.1); PU(0.2,0.1); PU(0.1,0.2); UF:
        PU(0.1,0.2); PU(0.5,0); G;
  S4: UF: PU(0,0); PU(0.3,0.1); DF: PU(0.3,0.1); PU(0.3,0.9);
        PU(0.1,0.3); PU(0.4,0.3); UF: PU(0.4,0.3); PU(0.5,0); G;
  S5: UF: PU(0,0); PU(0.1,0.2); DF: PU(0.1,0.2); PU(0.2,0.1); PU(0.3,0.1);
        PU(0.4,0.2); PU(0.4,0.5); PU(0.3,0.6); PU(0.2,0.6); PU(0.1,0.5);
        PU(0.1,0.9); PU(0.4,0.9); UF: PU(0.4,0.9); PU(0.5,0); G;
  S6: UF: PU(0,0); PU(0.1,0.4); DF: PU(0.1,0.4); PU(0.2,0.5); PU(0.3,0.5);
        PU(0.4,0.4); PU(0.4,0.2); PU(0.3,0.1); PU(0.2,0.1); PU(0.1,0.2);
        PU(0.1,0.8); PU(0.2,0.9); PU(0.3,0.9); PU(0.4,0.8); UF: PU(0.4,0.8);
        PU(0.5,0); G;
  S7: UF: PU(0,0); PU(0.2,0.6); DF: PU(0.2,0.6); PU(0.1,0.9);
        PU(0.4,0.9); PU(0.3,0.1); UF: PU(0.3,0.1); PU(0.5,0); G;
  S8: UF: PU(0,0); PU(0.1,0.2); DF: PU(0.1,0.2); PU(0.1,0.4); PU(0.2,0.5);
        PU(0.3,0.5); PU(0.4,0.6); PU(0.4,0.8); PU(0.3,0.9); PU(0.2,0.9);
        PU(0.1,0.8); PU(0.1,0.6); PU(0.2,0.5); UF: PU(0.2,0.5); PU(0.3,0.5);
        DF: PU(0.3,0.5); PU(0.4,0.4); PU(0.4,0.2); PU(0.3,0.1); PU(0.2,0.1);
        PU(0.1,0.2); UF: PU(0.1,0.2); PU(0.5,0); G;
  S9: UF: PU(0,0); PU(0.1,0.2); DF: PU(0.1,0.2); PU(0.2,0.1); PU(0.3,0.1);
        PU(0.4,0.2); PU(0.4,0.8); PU(0.3,0.9); PU(0.2,0.9); PU(0.1,0.8);
        PU(0.1,0.6); PU(0.2,0.5); PU(0.3,0.5); PU(0.4,0.6); UF: PU(0.4,0.6);
        PU(0.5,0);
  END;
end;
CGETR(-1,PXXX); CPUTR(-1,(N+6)*S/2*COS(-RAD(A))+PXXX);
CGETR(-2,PYYY); CPUTR(-2,(N+6)*S/2*SIN(-RAD(A))+PYYY);
QWARI:
end;

```

土木研究所報告132-1

昭. 43. 5 (1968)

越流型余水吐の水理機能に関する研究 (3)

——非標準型越流頂の流量係数について——

藤本 成

要 旨

本文は、越流頂の一般流量公式について考えたものであって、さきに研究(2)で導いた越流頂の流量係数の基本式および標準型越流頂に対する流量公式を考慮し、これに基づく一般公式を提案するとともに、非標準型越流頂に対するこの公式の適用性に検討を加えたものである。

その結果、ここに提案した越流頂の一般流量公式は、円弧および放物線で構成された越流頂に対して十分な適用性を有することが明らかになり、従来それぞれの形状に対して模型実験を行なうことが必要であったこれらの越流頂の流量係数についても、一般公式により取扱うことが可能となった。

土木研究所報告132-2

昭. 43. 5 (1968)

建設技術に適應した電子計算システム に関する研究 (1)

——NEW ALGOL-H を中心とするシステム——

中村慶一、小池長春

要 旨

土木研究所においては OKITAC 5090-H に対し NEW ALGOL-H を中心とする電子計算システムを確立し、(1) 1500 名の ALGOL プログラムにより製作されたプログラム中、汎用性のあるものは external procedure としてその名前を呼び出すことによる使用、(2) 15分以下の計算の割り込み使用、(3) 千葉支所との間のプログラムやエラーメッセージの電送による遠隔地使用、(4) 国化機の併用による自動設計の実行、といった特長を持つ運営を行なっている。

本文はこれに対し使用用語を ALGOL に定めた理由、NEW ALGOL-H と他のコンパイラとの比較ならびに、土木研究所向けとして上にあげた特長を持つシステムの概要に触れ、さらに増大する要望に基づいて機種変更を行なう場合の新システムの構想等について述べ、同じ程度の計算システム設計ならびに拡張システム設計の参考となるようまとめたものである。

キーワード (電子, 計算, システム, ALGOL, 国化, 自動設計, コンパイラ)

Report of The Public Works Research Institute Ministry
of Construction Vol. 132, May 1968

Studies on Hydraulic Performances of Overflow Spillways (3)

—On the Discharge Coefficients for Irregular Overflow Crests—

Sei Fujimoto

Synopsis:

This paper reports a new discharge formula of irregular overflow crests, introduced from the basic formula of the discharge coefficients of overflow crests and the discharge formula of the standard crest, which were studied in the former paper.

As a result of experimental verifications, it is found that the discharge coefficients of parabolic crests and overflow spillways with a finite value in the radius of curvature on the crest are expressed by this formula with sufficient accuracy.

Report of The Public Works Research Institute Ministry
of Construction Vol. 132, May 1968

A Study on the Electronic Computing System Adapted
to the Civil Engineering Field (1)

—A System with the *NEW ALGOL-H* as Central Compiler—

Keiiti Nakamura
Nagaharu Koike

Synopsis:

In the Public Works Research Institute, an electronic computing system with the *NEW ALGOL-H* as a central compiler has been established, and has several characteristics as follows:

- (1) calling external procedures which are written by 1500 programmers with *ALGOL*,
- (2) interrupting by preferential programs which might not exceed fifteen minutes for execution,
- (3) remote using from the Chiba Branch by means of electric transmission of program and error message tapes,
- (4) automatic designing using a digital plotter together.

This paper contains reasons why we selected *ALGOL* for compiler, comparison of *NEW ALGOL-H* and other compilers, and an outline of the new system which has above characteristics, and a conception of a new electronic computing system for the case of changing computer, and aimed to be usefull for same or extended scale computing system designing.

Report
of
The Public Works Research Institute
Ministry of Construction

Vol. 132, May 1968

CONTENTS

1. Sei Fujimoto

Studies on Hydraulic Performances of Overflow Spillways (3)

—On the Discharge Coefficients for Irregular Overflow Crests—(1)

2. Keiiti Nakamura and Nagaharu Koike

A Study on the Electronic Computing System Adapted to the
Civil Engineering Field

—A System with the *NEW ALGOL-H* as Central Compiler—(25)