

45—1
自動設計
技術管理



土木研究所資料

第 556 号

自動設計

(44年度版)

昭和 45 年 2 月

建設省土木研究所

正 誤 表

頁	行 目	誤	正				
1	上から7行目	モーメントと	モーメント				
6	左下から9行目	$(F_{5.1}, F_{4.2}, I_2, F_{5.1})$	$(F_{5.1}, F_{1.2}, I_2, F_{5.1})$				
8	FORTRAN 欄	$IF(P.EQ.0)K=3$	$IF(P.EQ.Q)K=3$				
9	左下から1行目	は2・3・の	は2・3・4の				
23	左上から9行目	$+\frac{1}{(R-1)}i$	$+\frac{1}{(R-1)}i$				
"	左下から13行目	$\sqrt{(\Delta x_1^P)^2 + (\Delta x_2^P)^2} \dots + (\Delta x_n^P)^2$	$\sqrt{(\Delta x_1^P)^2 + (\Delta x_2^P)^2 \dots + (\Delta x_n^P)^2}$				
24	左上から3行目	$+\sum_i (h_i^2 d_i^2 A_i)$	$+\sum_i (h_i^2 d_i^2 A d_i)$				
"	左上から4行目	$+h_i d_i^2 + A x^0$	$+h_i d_i^2 A x^0$				
25	左下から3行目	誰数量)	離散量)				
"	左下から1行目	土床係数	土圧係数				
27	左上から17行目	$\frac{M_1}{\ell_{s1} (l-x_1/3)}$	$\frac{M_1}{A_{s1} (l-x_1/3)}$				
"	左上から18行目	$\frac{2M_1}{\ell_{x2} (d-x_2/3)}$	$\frac{2M_2}{\ell_{x2} (d-x_2/3)}$				
"	左上から19行目	$\frac{M_2}{AS_2 (d-x_2/3)}$	$\frac{M_2}{A_{s2} (d-x_2/3)}$				
28	下から2つ目の表	<table border="1"> <tr><td>1,1</td></tr> <tr><td>1,1,1</td></tr> </table>	1,1	1,1,1	<table border="1"> <tr><td>11</td></tr> <tr><td>111</td></tr> </table>	11	111
1,1							
1,1,1							
11							
111							

自動設計

(44年度版)

建設省土木研究所
技術管理室

序

技術管理室では昭和39年発足以来、主として標準化を中心とした建設技術の近代化を目指し一路邁進してきた。その結果建設省制定土木構造物標準設計も擁壁類、暗渠類、橋脚、橋台、PC橋と数多く発刊することもでき、特に初期のものは電子計算によって設計計算を行った結果にもとづいて人間が作図を行なってきたが、昨年からはその作図過程すらすべて計算機によって行ない、その結果を複写して世界で初めての全自動工程による図集の出版を行なうことができた。

そうして発足以来6年たった今日組織変更が行なわれることになり企画部システム課という新名称のもとに、技術管理という名前は土木研究所から消え去ることになった。39年発足当時技術管理というはっきりしない名前をどう解釈しようかと迷った時、官房技術調査官と、Industrial Engineering にしようと意見が一致し、爾来この名前を使ってきた。つまりすでに組み立てられた計画に従って建設事業を遂行して行く時に、業務をスムーズに流すための方法論を考えその応用例を作ることにほかならない。当時考えたこの方向が誤りでなかったらしいことは6年目の今日全地建に技術管理課が設置されることになったことでも明らかである。

このように全地建に実施部門が出来た時点で土木研究所にシステムなる名称を冠した部門が誕生し技術管理の名前が消えたことはパイオニアとしての役割が終って標準化を中心とする建設技術近代化は実施の段階に入り、全地建の技術者がそれぞれの得意とする業務の標準化を開始し、土木研究所は次の良い計画を立てるための方法論 Systems Engineering の研究ならびにその実施例作成というパイオニア的業務を開始せよという意味であろうと考える。

もちろん各地建で作業した結果を検討し建設省制定土木構造物標準設計として発刊するまでの業務は今迄通り土木研究所において行なうが、電子計算による設計製図プログラムの作成ならびにチェックの仕事を各地建に引継ぐに当り、今迄に得た成果をひとまとめにしておく必要があると考えたのでここに集録することとした。

建設技術に適応した言語としてALGOLを選定し、特に建設省内共通入出力言語や便利な機能を付加したKDK(建設省土木規格)ALGOLの文法を提唱し、すでにこれに従うコンパイラとして沖電気のOKITAC5090H用NEW ALGOL-H, TOSBAC 3400-40/41用T-ALGOL, FACOM 230-20/30用ALGOL, CDC 3600 ALGOL用トランスレータなどがあるが、本文において使用した図化用手続きなどもすべてKDK ALGOLで書かれているので各地建においても有効に利用して頂ければ幸いである。

昭和45年2月12日

技術管理室長

中 村 慶 一

目

次

1. はじめに	1
2. 自動設計	2
2.1 マンマシンシステム	2
2.2 自動設計用語	2
2.2.1 ALGOL	3
2.2.2 KDK-ALGOL	5
2.2.3 FORTRAN	6
2.2.4 PL/I	6
2.2.5 ALGOLを採用した理由	7
2.3 図化用手続と図化例	9
2.3.1 図化機の概要	9
2.3.2 図化機の基本命令語	9
2.3.3 基本手続	9
2.3.4 ALGOLによる基本手続	9
2.3.5 図化用標準手続	10
2.3.6 漢字の生成	14
2.3.7 図化例	18
2.4 最適化手法	22
2.4.1 単一変数変化法	22
2.4.2 最大傾斜法	22
2.4.3 最適傾斜法	23
2.4.4 共役傾斜法	23
2.4.5 EVOP	24
2.4.6 最適化手法の通用例	25
2.5 自動設計例	29
2.5.1 ポストテンション方式によるPC単純T桁橋	29
2.5.2 RC単純スラブ橋	44
3. 標準設計と自動設計	51
4. おわりに	52

1. はじめに

工業化社会より情報化社会への移行の様相が工業先進国を中心に逐次表面化し始めてくると、その立役者である電子計算組織をめぐっていろいろな論議が行なわれることが多い。まず1番目が200年前の第1次産業革命による農業化社会から工業化社会への移行の時にも行なわれた「機械打ちこわし」にも似た論議である。「今までのおれ達の仕事の方が良い」から始まって、気持のうらの「ぬるま湯から出たくない」心理も手伝って行なわれた機械打ちこわしは、200年たった今も電子計算組織への反感となって再現された。次に現われてくるのは計算機をあくまで補助機械と見て給与計算や在庫計算あるいは曲げモーメントの計算など、くり返し等の筋肉的要素の多い計算を代行させたが、その印字結果を見てさらに人間が判断を加えてつぎの段階を実行させるといった使い方であった。

情報化社会においては人間と計算機の組み合ったいわばマンマシンシステムという新しい生物が主役となって仕事を進める社会であるという認識のもとでの成果はPPBSやMISあるいは自動計画自動設計という形で少しずつ世の中に現れ始めているということができよう。

建設省土木研究所において全研究者が計算機と対話を進めつつ研究上の壁を突き破る目的で電子計算組織を導入した昭和39年度は、また建設省において調査・計画・設計・施工の各段階における標準化作業を行なうために技術管理組織を拡充した年でもあり、このような経過で設置された土木研究所の技術管理室は直ちにこの計算機と取り組んで土木構造物の標準設計を作る作業に入った。まず技術者がプログラム用語に習熟する必要があるので全所員を始め、建設本省、地方建設局をふくめ約4500名のアルゴルプログラマを養成し、これらの人々と共に標準化作業を進めることにより、現在までに数多くの標準設計・基準類を作成することができた。

しかし初期の暗さよ、よう壁・橋台類の標準設計までは、図化作業に必要な一切の数値を印字させ、これによって人力による図化作業を行なってきたが、この際に生ずる不注意のミスはどのようにチェックしても後を断たなかった。また標準設計の特質としてその寸法のものだけを旨指して最適化しても普遍的に使用するためには前後の関係とか他の寸法のものとの材料を考え合せて選択する必要もあり、このような考慮も含めた完全な自動化システムを橋脚・PC橋に対して組み上げることができた。

本文においてはこのような標準化という立場から見た自動設計手続きの紹介、ならびに電子計算組織との対話の間に生れた種々の観点について触れてみたい。

2. 自動設計

2・1 マンマシンシステム

第2次産業革命ともいうべき情報革命による工業化社会より情報化社会への移行に際し、各組織ともに情報革命の担い手と目される電子計算組織を中心に経営情報システム(MIS)、基本計画・個別計画予算編成方式(PPBS)、システム分析(SA)、情報検索(IR)などのプログラム確立を急いでおり、従来の「強力なジョベル」としての使い方から「頭脳」としての使い方にその重点を移行しつつある。このことは第1次産業革命が「筋肉」の代替としてのエネルギー革命であったのに対し第2次産業革命が「神経」の代替としての情報革命であるという歴史的経過と全く同じ筋道を、今度はその情報処理の過程において筋肉的なものから神経的なものへとたどりつづけるものと受取って良いであろう。

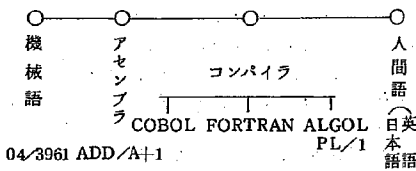
一方建設事業の分野においてもすでに電子計算組織の「筋肉的」利用はある程度進み始めているが、「神経的」利用の面では未だしの感が深い。「筋肉的」利用の場合はまだ問題を持つ者が適当に外部に発注して間に合っているが、「神経的」利用に至っては、問題を持つ者自身が、場合によってはトップマネージメント自らが自分の構想を計算機に伝える必要があるというよりは人間機械系という電子計算機と人間とが一体となって働くようないわば新しい生物が生まれたと考えるべきであることが次第に明らかとなってきた。

この機械と協同して新たな創造を行なうためにも、この人間機械系には共通の言語が必要であって、その言語はこれだけ処理能力の大きい機械が相手なのであるからできるだけ人間の言葉に近いものが望ましいわけである。

この言葉の問題については後で触れるが、いづれにせよこの言葉を用いて、問題を持つ人間が直接対話しつつ仕事を進めるべきであって、単なる付属物と考えていた時代のようにその問題の本質についてそれほど詳しくない通訳を介したりして仕事を進めたりすることはともすればその処理が停止めのかきめ方向に迷いこませたり、また問題の本質を知らぬ通訳のいいなりに処理されてしまう恐れもある上に、問題を持つ者が新たな視点を見出す機会も失わせてしまうことになる。

ジェームスワットの蒸気機関は、それまでのニューコメンの蒸気機関に対し、いわば弁の開閉を人間に代って自動的に行なわせるという点である意味の人的要素の非常に強い人間機械系のうち人間による部分を減少させたものと考えられる。この点で初期の電子計算機が機械自身の言葉で人間が命令しなければならなかったものを、機械語と1対1の対応はしているがシンボリックな(機

械によらない)命令を下せる段階を経由し、その後FORTRANさらにALGOLと、ますます人間の言葉に近い表現で命令すれば足るようになった経緯もまた、この人間機械系が人間の付属物に過ぎなかった時代から段々人間の介入する余地を減じて行なった経過とよく似ているといえよう。



2・2 電子計算用語

この電子計算機との対話のために用いられる言葉が、以前は人間の方が機械の言葉を覚えてこれを用いなければならなかったのに、だんだん人間の言葉に近いシンボリックな言葉を用いて機械自身に翻訳も行なわせるようになり、1956年にはIBM-704機に対してFORTRAN(formula translator)が生まれ、より人間の言葉に近いもので書いてやればすむようになり、さらに1958年電子計算用語ALGOL-58さらにALGOL-60および1962年改訂ALGOL-60が国際共通の電子計算用語として生まれ全世界で使用されつつある。

ALGOLはFORTRANに比して文法体系がすっきりしているだけでなく、われわれの言葉に非常に近く単なる順序とか記号のみによる約束が少ないため、われわれと電子計算機との対話に用いられるだけでなく、言葉の異なる人達や、さらにわれわれ日本人同志の間でも手順をわかりやすくかつ正確に伝達する言葉として最適であると考えられる。

電子計算機との対話には別段計算機というものの構造に関する知識はもちろん、昔プログラマなるものの唯一の武器であった機械語に関する知識、あるいはSIP、SAPなどの中間語に関する知識もいさい不必要であって、われわれの言葉に近いALGOLでプログラムを書いてやれば機械が文法チェックを行ない、エラーがなければ中間語を経由して機械語への翻訳ならびにそれによる演算を実行する。

したがって、単に電子計算機との対話をスムーズに行なう目的からはALGOL以外の知識は別段いらないわけであるが、昔からプログラマたちが自分達の相対的価値を引上げる目的で、これらの知識が入門として必須であるかのように主張し、その効あって初めての人達が電子計算機というものに親しもうとして読み始める文章

の最初はまず電気回路か機械語の話が始まるため、多くの人に電子計算機というものはむずかしいもの、したがって、それと会話を行なっているプログラマは偉い人達という印象を与えがちであった。

ALGOLの生まれた今日では、もはやプログラマという特別の職業を持った人が通訳を行なうというよりは、問題を持つ一人一人がALGOLによって直接会話をすることが普通となってきている。

建設省関係においては国土地理院 (NEAC 2206 48 ビット 10K), 建築研究所 (TOSBAC 3400-30 24 ビット 16K), 土木研究所 (TOSBAC 3400-40 24 ビット 65K および FACOM 270-30 16 ビット 65K), 岩手工事事務所 (FACOM 230-10) に引続き建設本省 (TOSBAC 3400-41 24 ビット 32K), 北陸地建 (TOSBAC 3400-41 24 ビット 32K), 近畿地建 (FACOM 230-35 16 ビット 32K) などに電子計算組織が導入され、各地建を含めて 4500 名の ALGOL プログラムの養成も終り、各技術者が自分自身の問題をもって直接計算機と対話することにより、今まで予想されていたような作業はもちろん、今まであきらめていたようなあるいは思いつくこともなかったような問題と取り組み始めており、土研における割り込み計算や、超大型機の衛星計算機としての使用、あるいは各地建における計算センターとの時間単価契約による自由使用などによってすでに相当な成果が得られている。

また、現在建設技術関係の各組織において電子計算組織の導入が計画されつつあるが、これに関連して 1) その必要性を十分納得させるだけの理由がない、2) 今でもコンサルタントを利用して間に合っているものを自ら手を下してやる必要はない、3) 計算機を入れてみても十分な仕事量がない、4) 単なるあそびで導入するのは反対である、などいろいろな反対論も多い。

各会社などにおいても、導入の当初において全く同じ議論がくり返され、1) 自分の業務遂行上の考え方の壁をつき破る、2) 人間を業務から開放する、3) その方がもつかる、4) 他の会社でうまく行っているのだからこちらでもやってみよう、などいろいろな理由で、大半の大組織は自ら導入、中小企業は共同使用という形で導入し、業績を伸ばしている所が多い。

その中でも特に FORTRAN, ALGOL 等のコンパイラを活用し、多くの社員に開放して成功している組織として東洋工業 (F), 川崎重工業 (A), トヨタ自動車 (A) などの例があり、土木研究所においても当初上あげたのと全く同じ反対があったときに、この道具を用いて研究上の壁を破る、したがって、今まで外注されていたプロ

グラムで手を加える必要のないものや、所内でテスト終了したもので本番計算に時間を費やすものは、今後も積極的に外注し、所内の計算機はいわば成功するかどうか不明であるプログラムのチェックを主目的とした外部の超大型機の衛星計算機であると割り切って OKITAC 5090-H (メモリ 16K 長語) を導入し、1) 内部はもちろん外部の計算機との接続にも使用用語は ALGOL 一本とする、2) 15 分以内の計算は割込み使用を許し見掛け上 100%稼働とする、3) 支所で作られたプログラムは電送回路により送られ、エラーは同回路によって直ちに送り返す、4) 開発された手続きはできるだけ辞書の形で格納し、その名前を呼び出すだけで自動的に使用可能にするといった特徴的な使用法により、各研究室においても相当な実績をあげることができた。

2.2.1 ALGOL

いま PC ケーブルの全長を 3 本のケーブルについて計算してそのおのおのを LEN という配列に格納するものとする。前の計算段階ですでに 1 番ケーブルの水平長の半分 $POINT$, 曲げ上げ部分の水平長 L , 垂直高 H および 1, 2, 3 番ケーブルの下辺よりの距離 E_1, E_2, E_3 を E という名の配列の 1, 2, 3 番目に格納したもの、ならびに曲げ上げ間隔 U の値を読み込みという形で受け取って、計算結果である LEN の内容を仮りに有効 5 けたの形で印刷するプログラムとするが、一連のプログラムとして使う時は、もちろんこの読み込みと印刷の部分は単なる値の受け渡しにしていればよいわけである。

放物線の水平部 (HL), 曲上部水平長 (LB), 曲上部高 (H) を知ってその実長を計算する式は、

$$HL + LB(1 + 1.5(H/LB)^2 - 0.4(H/LB)^4)$$

であるが、この部分を LB, H, HL を知って実長 IK を求める手続き $CABLE$ にまとめ、この手続きを 3 回繰返すプログラムにする。

最後の印刷は $LEN =$ の右に三つの値を印刷する形とし読み込みデータとしては、 L は 32m, H は 72cm, $POINT$ は 50cm, U は 90cm, E_1, E_2, E_3 はそれぞれ 8, 14, 20cm であったとする。でき上がったプログラムはつぎのとおりである。

```
begin                                     ①
  real  $L, H, U, POINT$ ; integer  $I$ ;      ②
  array  $E, LEN$  (1:3);                  ③
  procedure  $CABLE(LB, H, HL, IK)$ ;      ④
    real  $H, HL, LB, IK$ ;                ⑤
    begin real  $K$ ;                       ⑥
       $K := H/LB$ ;                        ⑦
```

```

      IK:=HL+LB*(1+15*K↑2-0.4*K
      ↑4); ③
end; ④
comment KEISAN START; ⑤
READ(L, H, U, POINT, E); ⑥
HS:=H-10; ⑦
for I:=1 step 1 until 3 do ⑧
  CABLE(L, HS-E(I), POINT+(if I=1
    then 0 else U), LEN(I)); ⑨
PRINTS('LEN='); ⑩
for I:=1,2,3 do ⑪
  PRINTF(1.2, LEN(I)); ⑫
CRLF(1); ⑬
end; ⑭
320, 72, 50, 90, 8, 14, 20, ⑮

```

日本語で書いたものと大差ないのでそのまま読み下すことも容易であるが、念のために逐行説明を加えてみるとつぎようになる。

①の begin は②の end と対応して文をつつむ括弧の役をしており、この間の縦線は人間の視覚にうたえるために入れたに過ぎない。

②の real とあるのは、L, HS などが小数点または 10 のべきを含む実数値の格納場所として②までの間限りで使用される変数であることを宣言している。同様に integer はそのつぎの名前が整数値の格納場所となることを宣言しているが別に区別して考えなくてもよい。

③の array は、つぎの E と LEN がそれぞれ添字 1 から 3 までを持つ 1 次元の配列の名前として使われることを宣言しており、2 行 3 列のマトリクス A なら
 A[1:2, 1:3]

とすればよい。

④の procedure は、CABLE という名前がつぎの文 (⑥から⑨までの begin と end でつづまれた外見上一つの文) を代表する手続きであることを宣言し、つぎの括弧の中の名前は、この⑥から⑨までの間で仮に使われしかも実行を命ずる手続き文の対応する実パラメタに置き換えられるべき仮パラメタである。

⑤は一見宣言と似ているが、仮パラメタに対応する実パラメタがどんな種類のものであるかを規制する部分である。

⑥～⑨は CABLE の名のもとに手続の実行を命ぜられた時行なうべき内容を示している。また⑥の K という宣言は、この名前が⑥の対応する end までの間で使われることを示し、このことを K はこのブロックに関

し局所的であるという。

⑦では、H/LB の結果を K に代入することを命じている。⑧の右辺の式の中で H/LB を直接書きこむと 2 度も同じ割算をしなければならないので、多少計算時間の損となるので K に仮りに受けてから K を使用している。

⑧の右辺の式は普通の意味と同じであるが掛算の × のかわりに * を、K⁴ のかわりに K↑4 を用いている。このようないくつずつの文が終了すると区切りとして、が入れてある。

⑩の comment は落書きで、つぎの; までの間、何を書いても計算機は無視する。

⑪は読み込み文で、最後の end の後に並んだデータテープ上から、対応する値を次々と順番に書かれた名前前の所に読み込む。この時整数実数などの型が異なっていて、宣言されたとおりの型に変換して読み込む。

⑫は配列読み込み文で、データテープ上のそのつぎの値を E[1] から E[3] まで次々と読み込む (添字はこのように表現するが 2 次元なら A(I, J) という形になる)

⑬は単なる代入文だが右辺の式の中で実数の H と整数の 10 などが混在すれば結果はすべて実数となる。

⑭～⑯は繰り返す文で、I に初期値 1 を入れて do の後の一つの文を実行し、終れば 1 おくって I を 2 としてまた実行、I が終期値 3 に達すればそれで実行して終了する。

⑮⑯は繰り返すべき一つの文で、いまの場合は CABLE という名の手続き文である。計算機は宣言をし、CABLE という手続き④～⑨について仮パラメタ L, ... などを実パラメタ L, ... などで置き換えてから⑥～⑨を⑮⑯の所におさめて実行する。第 1 回目は I が 1 であるから第 2 パラメタの値は HS-E[1] であり、第 3 パラメタは POINT+0 となり、結果は LEN[2] に格納される。第 3 パラメタの POINT+() の括弧中は条件式であって if と then の間が真ならば then の後、偽ならば else の後の値が選択される。

⑰は ' ' でつづまれた LEN= という文字そのものの印字準備を命じている。

⑱⑲によって LEN= のさらに右に LEN[1], LEN[2], LEN[3] の値を小数点以上 1 けた以下 2 けたの F タイプで印字準備することを繰り返す行なうよう命じた繰り返す文で⑱の step を用いた場合と、結果として同じ動作をする。いまの場合この計算結果 (LEN の内容) は 3.75, 4.66, 5.57 であるからそのように印字準備される。

⑳の CRLF(1) は以上の印字を実行し 1 行おくつことを命じている。

④の end は①に対応しアルゴリズムプログラム全体の終了を知らせる。

②はデータの部分で READ などに対応し、書かれた順に読み込まれる。

以上で通常出てくる ALGOL の記号などの説明の大部分を終ったが二、三補足すると、

(1) すべての文に $LA:K:=K+1$; などとコロンを介して名札をつけることができる。これがあると、go to LA; などと飛び越し文を使うことができる。

(2) if then else は式の中だけでなく、文をつつんで if A=B then SA:=R+2 else go to S; などと使用することができる。偽の時はつぎの文に移ってよければ else 以下を省略してもよい。

(3) 標準関数として sin, cos, arctan, ln, exp, abs, sign, sqrt, entier が使用できる (大文字のこともある)。 $B\sqrt{A}$ は $B * \text{sqrt}(A)$ と書く。

(4) 入出力文は機種によって多少異なっている。

もちろん、設計値決定の際に行なう最適性追求のための考慮のうち、客観的に記述可能なものはすべてプログラムに組み入れるわけで、たとえば所要鉄筋量が計算された際の鉄筋径選択に際しては単なる材料的最適性の追求だけでなく、材料置場、施工上の過誤、加工性などを考慮して行なう手続きを組み入れてある。

そのために、このような一般化された手続きは、すべて external procedure として翻訳の終わったものをシステムに登録しておくことが望ましく、そのために何人かで作ったプログラム中、その部分限りで使用する名前をはかの部分と関係させないような ALGOL のブロック構造が有用であるほか整数実数の混合演算を認める点、繰り返し文の制御変数は正の整数に限らずどのような式でも良い点、添字も正の整数や一定の型式にこだわらずどのような式でも良い点など、FORTRAN に対して PL/I が優越すると IBM が認めている点はすべて ALGOL では最初から可能であったし、条件式の存在なども考え合わせ全技術者が直接対話するための言語としては ALGOL や PL/I が最適であると結論することができよう。

2.2.2 KDK ALGOL

土木研究所では昭和 39 年に OKITAC 5090-H を導入し、昭和 42 年にこれを TOSBAC 3400-40 に変更すると共に、風洞実験制御等を主目的として FACOM 270-30 を導入した。また建設省内各機関についてどのような計算機が導入されるか判らないので、異なる機種間でも、今まで蓄積された procedure (手続き) がそ

のまま引継がれ得るようにすることが当初から問題になった。特に入出力文に関しては国際共通文法の改訂 ALGOL 60 には何の規程もないし、明らかに人間側にとって有利なものはでき得れば追加し易いように、当時制定を急いでいた土木試験基準の中に引用された建設省土木規格 (KDK) にならって、建設省に導入される計算機が共通に受入れられることを希望する項目をまとめて、KDK ALGOL の文法をつくり、導入時の仕様書にこれを引用することによって、上記の 3 機種にはその相当部分がそのまま通用するようにした。

CDC 3600 などについてもこの受入れ用のトランスレータができたため、内部の計算機をプログラムチェックや即時性を要する計算に振り向けることが可能になっている。

以下 KDK ALGOL の特長 (JIS ALGOL 7070 に対する追加) を挙げてみると次のようになる。

(1) 入出力文

JIS ALGOL 70 では膨大なものになり、このうちのひとつでも呼出すとメモリの相当部分を占有することになるので、もっと簡単に殆ど同じ結果を期待できるような次のものも指定する。

読み込み READ (A, B, ...); A, B, ... は単純変数名でも配列名でもよい。データがどのような形でも宣言された通りに変換する。配列の適当な領域読み込みは READARRAY (NAME, L1, U1, L2, U2, ...);

数値印字準備 PRINTR (N, E); PRINTI (N, E);
PRINTF (n₁, n₂, E);

文字印字準備 PRINTA (N, A);
PRINTS ('...'); SPACE (N);

印字実行政府 CRLF (N);

フォーマット指定の数値印字
PRINT ('フォーマット,' リスト);

(2) 宣言

alpha, double, complex を追加

(3) 式にベクトル式を追加

vec E1, E2, ... tor や vec vec E11,
E12, ... tor, vec E21, E22, ... tor tor

(4) 初期値設定

real C, A initial 0, B initial 3.14, D;
(C, A に 0, B に 3.14, D には何も入らない)
array A, B [1:10] initial vec 0 tor;

(5) 配列の断面

A [I, *] := vec E1, E2, ... tor; (I 行の宣言された列の最初から E1, E2 の順に代入され不足するときは最後のものが繰返される)

(6) 文字配列, 文字関数, 文字ベクトル式, 文字作
用素(+ -)を認める。

(7) 外部手続き宣言
external BESSEL, MULTEQ;

(8) データはカンマで区切り同じデータに対し5*
3.5などを認める。

OKITAC 用 NEW ALGOL-H では(1)(7)(8)を,
TOSBAC 用 T-ALGOL ではこの全部を認めている。
なお, CDC 3600, FACOM 270-30 用 ALGOLも(1)
は満足するが, それ以外に(2)(6)(7)(8)などを認め, トラ
ンスレータの使用により土木研究所で作成したアルゴル
プログラムテープそのままを受入れることができる (FA-
COM 230 25/35に対しても準備中)

なお全員のプログラムは極力その大部分を procedure
で構成するようにつとめ修正に便ようにするほか, 使
用頻度が多いと思われる procedure はできる限り一般化
して中間語まで翻訳させたものをシステムに格納し ex-
ternal 宣言で呼出して, プログラム中で使用するほか,
10 分以内の割込み演算やマイクロ回線によるプログラ
ム, エラーメッセージ等の帯送により各人が計算機と対
話しつつ新しい観点を発見する方向を目指している。

2.2.3 FORTRAN

今 2.2.1 の ALGOL プログラムを FORTRAN で書く
とつぎのようなものになる。

```

SUBROUTINE CABLE(LB, H, HL, IK) ①
  REAL LB, IK ②
  K=H/LB ③
  IK=HL+LB*(1.0+15*K**2-0.4*
1  K**4) ④
  RETURN ⑤
END ⑥
DIMENSION E(3) ⑦
REAL LEN(3) ⑧
C KEISAN START ⑨
  READ(5,1) L, H, U, POINT ⑩
1  FORMAT(F5.1, F4.2, I2, F5.1) ⑪
  READ(5,2) (E(I), I=1,3) ⑫
2  FORMAT(3F4.1) ⑬
  HS=H-10.0 ⑭
  DO 10 I=1,3 ⑮
  IF(I.EQ.1) GO TO 20 ⑯
  A=U ⑰
  GO TO 10 ⑱
20 A=0.0 ⑲

```

```

10 CALL CABLE(L, HS-E(1), POINT
1  +4, LEN(1)) ⑳
WRITE(6,3) (LEN(I), I=1,3) ㉑
3  FORMAT(1H1, 4H LEN=, 3F5.2/) ㉒
STOP ㉓
END ㉔

```

簡単に補足すると, ①~⑥は ALGOL の procedure 宣
言に当たる。②は名前の先頭が I~N まですと自動的に
整数タイプと判断するので, LB とか IK という名前を
そろえるために REAL という宣言を行なったもので最初
から RLB とか AIK などとして逃げておけば不用であ
る。

③の=は等号でなくて ALGOL の:=に当るものであ
る。④の式の中で 1.0 としたのは混合演算を許さないた
め止むを得ずつけたものであり, ** は ALGOL の↑に
当る。

⑦は配列宣言で下限は 1 からしか使えない。⑧は LEN
という名前なので, 止むを得ず行なったもので RLEN
とでもしてよければ⑦で E(3), RLEN(3) とすればよい。

⑨のように, カードの第 1 欄に C とあればコメントで
ある。⑩の 5 は入力機器番号を仮りに 5 としたもの, 1
はフォーマットの番号をさすものでフォーマットはカー
ドに打ち込む形を指定している。

⑫の 10.0 も混合演算ができないための処置で
ある。

⑬は繰り返し文に当たりステートナンバー 10 までを
繰り返すが I は 1 きざみの場合で, そうでなければその
後にきざみを入れる。I は正整数に限る。

⑭は条件文で, .EQ. は ALGOL の=に当たり, > の
代りには .LT. などと非英語系の人間にはわかりにくい。
真の時はその右の GO TO 20 を実行する。偽の時はず
ぎの A に U の値を代入する文を実行するがそのつぎは 10
にとび越さねばならないので面倒である。

⑮はサブルーチンの実行を命じている。

⑯の 1H1 は 1 行おくりを命じておりつぎの 4H はその
後の 4 文字そのものの印字を命じている。/ の後に (),
ブランクまたは / が来たものは 1 行おくりを命じている
部分である。

2.2.4 PL/I

ついでに IBM が FORTRAN, COBOL, ALGOL を
総合したといっている PL/I で書くようにつぎのようになる。

```

KEISAN: PROCEDURE OPTIONS(MAIN); ①
  DECLARE(L, HS, H, U, POINT, (E, LEN) ②
  (3)) FLOAT, I, FIXED; ③
  CABLE: PROCEDURE(LB, H, HL, IK); ④

```

```

DECLARE (H, HL, LB, IK) FLOAT; ⑤
BEGIN; ⑥
  DECLARE K FLOAT; ⑦
  K=H/LB; ⑧
  IK=HL+LB*(1+15*K**2
    -0.4*K**4); ⑨
END CABL; ⑩
/* KEISAN START */ ⑪
GET LIST (L, H, U, POINT, E); ⑫
HS=H-10; ⑬
DO I=1 TO 3; ⑭
  BEGIN; DCL K FLOAT; ⑮
  IF I=1 THEN K=0; ELSE K=U; ⑯
  CALL CABL (L, HSE-(I), POINT ⑰
    +K, LEN(I)); ⑱
  END; ⑲
END; ⑲
PUT EDIT('LEN=',(LEN(I)DO 1=1 TO 3),②
  SKIP(1))(3)F(7,3)); ②
END KEISAN; ②
320 72 50 90 8 14 20 ②

```

ほとんどALGOLと同じなので補足の必要もないくらいであるが、いってつけ加えると、

①PL/Iではプログラムはすべて手続きの形をとり、一番外側だけこの行のような特殊な書き方をすると実行文なしで実行する。KEISANは適当な名前。

②宣言は実数FLOAT、整数FIXEDの別必要。

④手続き宣言、⑤のALGOLという規制は⑦の本体中の宣言と一緒にでもよい。

⑥の=はALGOLの:=と=に共用する。*はALGOLの↑に当たる。

⑩PROCEDUREはENDで終る。その後の名前をついたブロックまでのENDはもし不足なら自動的に入る。

⑪はコメントで、どこに書いてもよい。

⑫DOの後のセミコロンはALGOLと逆にならず必要。またDOはENDと対応し文をくくる括弧となる。

⑬のDCLはDECLAREと同じ。条件式がないので止むを得ずKを宣言した。

⑭はELSEの前の;はALGOLと逆にならず必要。

⑮手続の実行はCALLによる。

⑯BEGINはENDと対応してブロックを作る。BEGINの後はALGOLと違ってかならずが必要。

⑰出力はPUT EDITにより最初の括弧がリスト。後の括弧がフォーマットである。

⑱データは空白で区切られる。

このようにPL/IはALGOLとほとんど同じであり、このほかに構造体を扱うことができる代りに条件式がない点がALGOLと異っている。

2.2.5 ALGOLを採用した理由

ALGOLは現在1962年の改訂ALGOL-60が基となっているが、FORTRANとの間にはその成立の経過からもわかるように、多くの相似点がある。しかも現在までのIBM社の世界市場の支配ぶりからみても多くのFORTRANプログラムが過去において作成され、現在もこれらの遺産に類して仕事をする向きには重要な言語であることはもちろんであるが、何と云ってもIBM社の機械向きに作られたという動機と、IBM機のプログラマといえば機械語および中間語の訓練を受けたプログラマということを前提としているためか、機械の側に都合良くしようとするあまり、人間側に過度の注意を強制される結果となっている。この点で始めから人間と機械はもちろん、人間と人間の間の会話を念頭においてはっきりと筋の通った文法のもとに組み立てられたALGOLとの間には大分差異が見受けられ、またALGOL、COBOL、FORTRANを総合したと称するIBMのPL/Iも構造体など今までなかった概念を取り入れているが、BEGIN ENDなどによるブロック構造や文の区切りのセミコロンなどほとんどALGOLになっているのに肝心の条件式がないなど問題も多い。筆者自身も数年前にFORTRANによる短期教育を試み、ALGOLのように2日間くらいその教育によって習得させることは困難であると感じたので建設省関係はすべてALGOL一本による方針のもとに現在まで、4,500人に近いプログラマの養成を行ってきた。

しかし、よくALGOLとFORTRANおよびPL/Iの差異についての質問を受けることが多いので、以下表一に簡単にその比較を行なってみよう。

日本語「AにBというベクトルのI番目(IはP=Qなら3、さもなければ4)と3×(もしR≥SならL1さもなければL2)の差を代入せよ」

ALGOL A:=B [if P=Q then 3 else 4];

-3*(if R≥S then L1 else L2);

したがって、日本語で思いつくままに書いた内容はALGOLではほとんどそのままの形で書き下すことができるが、条件式のないFORTRANでは1番左の欄のような表現によらざるを得ない。

すでにFORTRANで書かれているプログラムはその右のようなALGOLに直訳することが可能であるが、逆に

表-1

問 題 点	FORTRAM	ALGOL	PL/1
1. ブロック構造	ない そのかわり無宣言で変数を使用できるのが先頭の文字が I ~ K のとき integer となる。	ある そのかわりすべての名前には宣言を必要とする。	あ る
2. integer と real の混合演算	認めない。	認める。	認める
3. 配列次元数	FORTRAN II では 2 次元、IV では 3 次元	無制限	無制限
4. くり返しのコントロール変数	正の integer に限る	無制限	無制限
5. 高精度演算	IV では DOUBLE	可変語長の場合を除き特別措置宣言	あ る
6. 条件式	ない	算術式、論理式、行先式に対してある。	な い
7. 配列の添字式	$C \cdot V \pm C$ の形に限る (C:定数, V:変数)	条件式を含め任意。	任 意
8. 構造式	ない	ない	あ る

このように書かれた ALGOL のプログラムを FORTRAN に直訳することは不可能である。

また PL/1 で上の内容を書きおろすとやはり条件式がないために右欄のようにならざるを得ない。またその表現を詳細にみるとちょうど条件式のない ALGOL のようなもので、ブロック構造をつくる BEGIN END や文の切れ目の ; IF THEN ELSE などの存在は結局 IBM といえども FORTRAN の不合理さをみとめざるを得なくなったことを示している。

この問題点にてらしても過去においてすでに用意されたプログラムに頼るのでなくて、これから自分の問題に向って行かなければならない。しかも多種多様の専門分化された分野の科学・技術者達は専門外のプログラムに問題を説明してプログラムを書いて貰うというやり方でなく、自からプログラムを書く方がはるかに容易で、その目的のためには記号とか順序が単なる約束で書かれる部分の多い FORTRAN より、筋の通った文法によって、しかも step P until Q 等の視覚にうったえる要素の多い ALGOL が適していると結論することができよう。

FORTRAN	ALGOL (直訳)	PL/1
$K=4$ IF (P.EQ. 0) K=3 $L12=L2$ IF (R.GE. S) L12=L1 $A=B(K) - 3.0 * L12$	$K:=4;$ if P=Q then K:=3; $L12:=L2;$ if R $\geq$ S then L12:=L1; $A:=B(K) - 3 * L12;$	$K=4;$ IF P=Q THEN K=3; $L12=L2;$ IF R $\geq$ S THEN L12=L1; $A=B(K) - 3 * L12;$

2・3 図化用手続と図化例

現在、土木研究所で使用している図化機は *CALCφ-MP* の図化機のように各ステップ毎の座標点がすべて与えられるのではなく、直線および円運動はその区別と次に続く数値情報に基づいて、図化機側で運動をコントロールするように設計されたものである。

以下、そのような図化機のために用意した基本命令語（機械語）、その命令語向きの8進数に変換したり、ペンの座標位置を追ったりして *ALGOL* 手続の中で用いる基本手続、*ALGOL* で図化プログラムを作成する場合に一般に用いる *ALGOL* 手続そして基本的な図化作業を手続化した図化手続きとその使用例について述べる。

2・3・1 図化機の概要

型	岩崎通信機製	渡辺測器製
	ロールタイプ	フラットベッドタイプ
プロット面	X軸 ∞(ロールペンの長さ)	1100mm
	Y軸 250mm	800mm
速度	400ステップ/秒	800ステップ/秒
	ステップ巾 0.1mm	ステップ巾 0.1mm
精度	0.2mm	0.15mm
記録ペン	ボールペン	インクペン
		0.1mm S型
		0.3mm D型
		0.5mm G型の3種

2・3・2 図化機の基本命令語

	(ロールタイプ)	(フラットベッドタイプ)
ペン上げ	=U	=U
ペン下げ	=D	=S(S型を下げる)
	=S(=Dと同じ)	=D(D ")
	=G(")	=G(G ")
直線	=F	=F
右廻り円弧	=R	=R
左廻り円弧	=L	=L
停止	=H	=H
原点復帰	=0は無視する	=0

これらの各種命令語は、ハード的に決められたもので、出力されたデータは両プロッタに互換性を持たせた。

2・3・3 基本手続

基本命令語に座標点として8進数を与えるので8進数に変換を行なったり、ペンの座標をおいかけたり、各種 *ALGOL* 手続きにて共通に用いるもので一般プログラマは2・3・3の *ALGOL* による手続を用いる。

(1) PUP

命令語=Uを出力する

(2) PFP

命令語=Fを出力する

(3) PRP

命令語=Rを出力する

(4) PLP

命令語=Lを出力する

(5) PSP(I)

I: integer

命令語を I=1 の時=S, I=2 の時=D, さもないければ=Gとして出力する。

(6) PNP(X,Y)

X,Y: real

X,Y の値を10倍し8進整数に変換し出力する。

(7) XYM(N,X,Y,UD,S,AX,AY)

N,UD,S: integer

X,Y,AX,AY: real

各種 *ALGOL* 手続きにペン先の状態を受け渡す。

X,Y はペン先の座標, UDは=1の時ペン先は下げの状態で=0はペン先は上げの状態で, Sは=1の時, S型=2の時D型=3の時G型をそれぞれ使用している。

AX,AY は誤差修正用にもなる。

N=1で読み込み, N=0で書き込む。

(8) STRING(S,A)

S:string

A: integer array

長さNのstring Sをその内部コードに変換し integerarray A(1:N)に入れる。POUTSにて使用する。

2・3・4 アルゴリズムによる基本手続

(1) PREADY(X,Y)

X,Y: real

現在のペン先のある位置を(X,Y)とする座標系を設定せよ、従って現在ペン先のある位置を原点にとりたい場合はPREADY(0,0)とする。

(2) PMOVE(X,Y)

X,Y: real

現在ペン先のある位置から点(X,Y)までペンを上げて移動せよ。

(3) PSEN(X1,Y1,X2,Y2)

X1,Y1,X2,Y2: real

もし現在ペン先のある位置が $(X1, Y1)$ なら
ただちにさもなければその点までペン先を上げ移動
した後に点 $(X2, Y2)$ まで直線を引け。

(4) $PSEN(X, Y)$

X, Y : real

現在ペン先のある位置から点 (X, Y) まで直線
を引け。

(5) $PSENA(A, L)$

A, L : real

現在ペン先のある位置から A 度の方向角で長さ
 L の直線を引け。

(6) $PSENCN(N, X, Y)$

N : integer

X, Y : array [1: N]

もし現在ペン先のある位置が $(X[1], Y[1])$
ならただちにさもなければその点までペン先を上げ
て移動した後に N 個の座標点を逐次直線で結ぶ。

(7) $PSENVN(N, A, L)$

N : integer

A, L : array

N 個の線分の方向角とその長さを与え、これを
逐次直線で結ぶ。

(8) $PEN(X1, Y1, X2, Y2, R)$

$X1, Y1, X2, Y2, R$: real

もし現在のペン先のある座標が $(X1, Y1)$ なら
ただちにさもなければその点までペン先を上げ移動
した後に点 $(X2, Y2)$ まで半径 R の円を R が正
なら右廻り、負なら左廻りで面け。

(9) $PENX(X, Y, R)$

X, Y, R : real

現在ペン先のある位置から点 (X, Y) まで半径
 R の円を R が正なら右廻り負なら左廻りで面け。

(10) $POUTI(A, S, N, E)$

A : real

S, N, E : integer

式 E の値を有効 N 桁で方向角 A 、大きさ S で
整数タイプで面け。

(11) $POUTF(A, S, N, E)$

S : integer

A, N, E : real

$N = a_1 \cdot a_2$ の形である。

式 E の値を小数点以上 a_1 桁、小数点以下 a_2 桁で
方向角 A 、大きさ S で面け。

(12) $POUTS(A, S, N, ST)$

A : real

S, N : integer

ST : string

長さ N の記号列 ST を方向角 A 、大きさ S で面
け。記号列中アルファベット以外はスペースに置き
換える。

(13) $PEN1$

これ以後 S 型(0.2mm)のペン先を使用する。

(14) $PEN2$

これ以後 D 型(0.3mm)のペン先を使用する。

(15) $PEN3$

これ以後 G 型(0.5mm)のペン先を使用する。

(16) $PMARU(X, Y)$

X, Y : real

中心の座標 (X, Y) を与えると、現在ペン先
のある位置から全円を面く。

(17) $PDRAW(N, X, Y)$

N : integer

X, Y : array

N 個の座標点を逐次直線で結ぶ、ただし $X[1]$
または $Y[1]$ は10,000を加えておくとその点ま
ではペン先を上げて移動する。現在の座標系と独立
な座標系を設定することが出来る。

(18) $PEND$

ペン先を原点復帰させ $=H$ をパンチしテープ
をフィードする。

(19) その他

(i) XYM で最期値とし $X=0, Y=0, S=2$
を与えているので $PREADY(0, 0)$ と $PEN2$ を
最初に発動しなくてもよい。

(ii) $PSENA, PSENVN, POUTI, POUTF,$
 $POUTS$ で用いている方向角 A は反時計廻りを正
とし単位は度である。

(iii) 座標点はすべてmm単位とする。

2.3.5 図化用標準手続

(1) N 組の座標を与え次々に座標を結ぶ手続き。

$PSEN(X1, Y1, X2, Y2)$ の一般形である。

procedure $PSENCN(N, X, Y)$;

integer N ; array X, Y ;

(注) (X_i, Y_i) : i 点の座標値($1 \leq i \leq N$)

N : 座標点の個数

(例) 現在のペン位置が $(4, 1)$ で、7組の座標値
 $((1, 2), (2, 8), (4, 6), (6, 7), (8, 6),$
 $(6, 3), (3, 4))$ を結びなさい。

答) $X := \text{vec } 1, 2,$
 $4, 6, 8, 6,$
 $3 \text{ tor};$
 $Y := \text{vec } 2, 8,$
 $6, 7, 6, 3,$
 $4 \text{ tor};$
 $PSENCN$
 $(7, X, Y);$



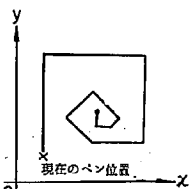
(2) N 組の“角度および長さ”を与えて、 N 本の直線を引く手続き。 $PSENA(A, L)$ の一般形。

$\text{procedure PSENVN}(N, A, L); \text{integer } N;$
 $\text{array } A, L;$

(注) A_i, L_i : N 組の“角度および長さ”
 N : 組数

(例) 現在のペン位置が $(2, 2)$ で、10組 $(90, 8; 0, 8; 270, 7; 180, 4; 135, 3; 315, 3; 225, 1; 180, 1; 90, 1;)$ の“角度および長さ”を与え10本の直線を引きなさい。

答) $A := \text{vec } 90, 0,$
 $270, 180,$
 $135, 45,$
 $315, 225,$
 $180, 90$
 $\text{tor};$
 $L := \text{vec } 8, 8,$
 $7, 4, 3, 3,$
 $3, 1, 1, 1 \text{ tor};$



$PSENVN(10, A, L);$

(3) 四角形の一辺を基に、四角形内に平行線を引く手続き。

$\text{procedure PHKF}(X, Y, P); \text{array } X, Y, P;$

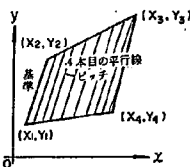
(注) (X_i, Y_i) : 四角形の交点の座標値で時計周りに入れる。

P_j : j 本目の平行間隔

平行線は (X_1, Y_1) と (X_2, Y_2) を基準に引く。

(例) 四角形 $((1, 3), (3, 9), (11, 13), (9, 4))$ 内にピッチ $(0.2, 0.2, 0.2, 0.3, 0.3, 0.2)$ で平行線を引きなさい。

答) $X := \text{vec } 1, 3,$
 $11, 9, 1 \text{ tor};$



$Y := \text{vec } 3, 9, 13, 4, 3 \text{ tor};$

$P := \text{vec } 0.2, 0.2, 0.2, 0.3, 0.3, 0.3,$

$0.2 \text{ tor}; \text{PHKF}(X, Y, P);$

(4) 任意の多角形の中に平行線を引く手続き

$\text{procedure PHKN}(X, Y, P, TH, N);$

$\text{array } X, Y, P \text{ real } TH; \text{integer } N;$

(注) (X_i, Y_i) : 多角形の i 番目の頂点の座標値

P_j : j 本目の平行線のピッチ

TH : 平行線が X 軸となす角度

N : 頂点の数

(例) 7角形 $((2, 4), (2, 6), (8, 6), (6, 10), (13, 5), (6, 2), (8, 4))$ の中に、ピッチ $(0.3, 0.3, 0.3, 0.3, 0.5, 1, 1, 0.5)$, X 軸となす角度 (45°) で平行線を引きなさい。

答) $X := \text{vec } 2, 2,$

$8, 6, 13,$

$6, 8, 2$

$\text{tor};$

$Y := \text{vec } 4, 6,$

$6, 10, 5,$

$2, 4, 4 \text{ tor};$

$P := \text{vec } 0.3, 0.3, 0.3, 0.3, 0.3, 0.5, 1,$

$1, 0.5 \text{ tor};$

$\text{PHKN}(X, Y, P, 45, 7);$

(5) 円の中に中心を基準に平行線を引く手続き

$\text{procedure PHKE1}(X, Y, P, TH, R);$

$\text{real } X, Y, P, TH, R;$

(注) (X, Y) : 円の中心座標

TH : 平行線が X 軸となす角度 (反時計方向を正とする)。

R : 円の半径

P : 平行線のピッチ

$P < 0$ のときは $pitch$ の中間を円の中心にもってくる。

$P > 0$ のときは中心を通る。

(例) 円の中心が

$(5, 13)$ で半径

3 である円に、

X 軸となす角

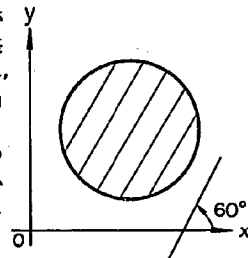
度が 60° で、

ピッチ 0.5 の

平行線を中心

が通るように

引きなさい。



答) PHKE1(5, 13, 0.5, 60, 3);

(6) 扇形の中に平行線を引く手続き

procedure PHKE2(X, Y, P, TH, R, TH1, K);
read X, Y, TH, R, TH1, K; array P;

(注) (X, Y) : 円弧の中心の座標

P_i : i 本目の平行線のピッチ

TH : 平行線が X 軸となす角度

R : 円の半径

TH1 : 扇形の一辺が X 軸となす角度

K : 扇形の二辺が作る位相角

角度はすべて、反時計方向を正とする。

(例) 位相角が

135° の扇形に、
ピッチ 0.4, 0.4,
0.4, 0.4, 0.4,
0.8, 0.8, 0.4,
0.4 で、X 軸と
なす角度 90°
の平行線を引き
なさい。

答) P := vec 0.4,

0.4, 0.4,

0.4, 0.4,

0.8, 0.8, 0.4, 0.4 tor;

PHKE2(6, 8, P, 90, 3, 22.5, 135);

(7) 引き出し線を引く手続き。(引き出し線上の記号列も同時に描く)

procedure PHIK1(X1, Y1, S, A, ST1, SM1,
ST2, SM2)

real X1, Y1, S, A, SM1, SM2;

string ST1, ST2;

(注) (X1, Y1) : 引き出し線端の座標値

S : 記号 (鉄筋の種類および数字
等) のサイズ

A : 上記の記号の傾斜角度

ST1 : 鉄筋の記号

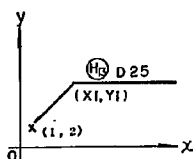
SM1 : 鉄筋番号

ST2 : 鉄筋種類

SM2 : 鉄筋径

(例) (1, 2) の

点より、引き
出し線を描き
引き出し線
上に "H₁₃ D25"
の記号列を描



くプログラム。

答) PHIK1(6, 5, 0, "H", 13, "D", 25);

(8) 2点間に矢線を描く手続き

procedure PARR0W(X, Y, D, PR, S, F, K);
array X, Y; real P, S; integer F, K;
procedure PR;

(注) (X_i, Y_i) : 矢線を引く 2 点の座標値
(i = 1, 2)

D : 2 点を結ぶ線分から矢印まで
の隔り

PR : 矢線上に描く記号列の手続き名

PRX X_(F) D(K)

PRM (F) × (K) = (F × K)

PR A 数値を描く。(F) は桁
数

S : 記号列 (上記の PR) のサイズ

F, K : 数値 (F 引き出し線を引く手
続参照)

D = 0 のときは、引き出し線は描かず矢線
のみ

(例) 2 点 (2, 7),

(1, 2) 間に矢

線を描き、上

に 150 という

数値を描くプ

ログラム。

答) X := vec 2,

8 tor;

Y := vec 1, 2 tor;

PARR0W(X, Y, 4, PRA, 5, 3, 150);

(9) N 個の点を通過する矢線を描く

procedure PDARR0(X, Y, P, TH, N);
array P; real X, Y, TH; integer N;

(注) (X, Y) : 始点の座標

P_i : i 番目の矢印のピッチ (最後の
ピッチは零を必ず入れる)

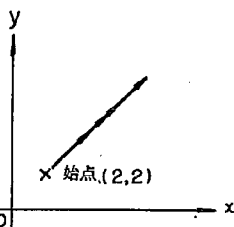
TH : 矢線が X 軸となす角度

N : 矢印の大きさ (mm)

(例) 始点 (2, 2) より、ピッチ 15, 10, 5,
10, 0 で X 軸と 45° の傾きを持つ矢線を描
きなさい。

答) $P := \text{vec}$

15,
10,
5,
10,
0 *for*;
PDARR0
(2, 2, P,
45, 2);



(10) 折り曲げ筋の表示について

procedure PBEND(A, K, H, L);

real A, K, H; *array* L;

(注) A: 折り曲げ筋(全体)の傾斜角

K: 折り曲げ角

H: 折り曲げ高さ

L: i番目の水平部分の長さ

L=0のとき、前の位置で終る

(例) X軸と15°; 折

り曲げ高さ1, 折

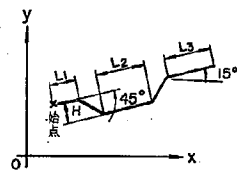
り曲げ角45°, 水

平部分の長さL

(1, 2, 2, 0)の

折り曲げ筋を描く

プログラム。



答) $L := \text{vec } 1, 2, 2, 0 \text{ for};$

PBEND(15, 45, 1, L);

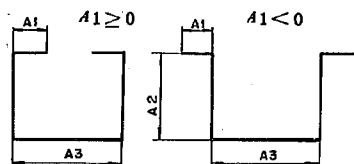
(11) スターラップを描く手続き

procedure PSTIR(K, A1, A2, A3);

integer K; *real* A1, A2, A3;

(注) K: スターラップ(全体)の傾斜角

A1, A2, A3: 下図参照



(例) 全体の傾斜角0°

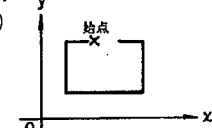
A1(2), A2(4), A3(6)

のスターラップを

描くプログラム。

答) PSTIR

(0, 2, 4, 6);



(12) ヒストグラムを描く手続き

procedure PHISTG(AY, PX, PY, SX, SY, N);

real PX, PY, SX, SY; *integer* N;

array AY;

(注) PX, PY: 軸方向のピッチ

AY: 計算値

SX, SY: 縮尺

N: 個数

(例) 計算値AY (0.5,

1, 2, 4, 3, 1)

を、軸方向1, 縮

尺1でヒストグラ

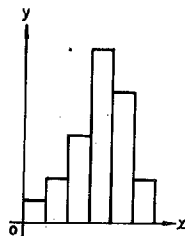
ムを描け。

答) $AY := \text{vec } 0,$

5, 1,

2, 4, 3, 1 *for*;

PHISTG(AY, 1, 1, 1, 1, 6);



(13) 座標軸を描く手続き

procedure PAXIS(X, Y, AX, AY, DX, DY, NX, NY);

real X, Y, DX, DY; *integer* NX, NY;

array AX, AY;

(注) (X, Y): 原点の座標

DX, DY: プロットするきざみ幅

NX, NY: 個数

AX, AY: 計算値

(例) $AX := \text{vec } 10,$

20, 30,

40 *for*;

$AY := \text{vec } 0,$

1, 2, 3,

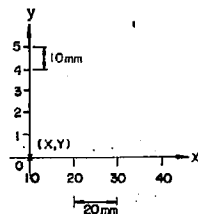
4, 5 *for*

PAXIS(X,

Y, AX, AY,

20, 10, 4,

6);



以上

例題 図の引き出し線および矢印等を作図するプログラム

答

$P := \text{vec } 0, 10, 10,$

20, 10 *for*;

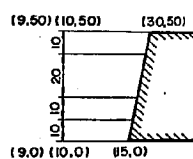
$A := \text{vec } 9, 15, 30,$

9 *for*;

$B := \text{vec } 0, 0, 50,$

50 *for*;

PHKF(A, B, P);



```

A := vec 10 tor;
B := vec 10 tor;
for I := 1 step 1 until 4 do
begin
  X := vec A(I), A(I+1) tor;
  Y := vec B(I), B(I+1) tor;
  PARROW (X,Y,0,PRA,5,2,P(I));
end;

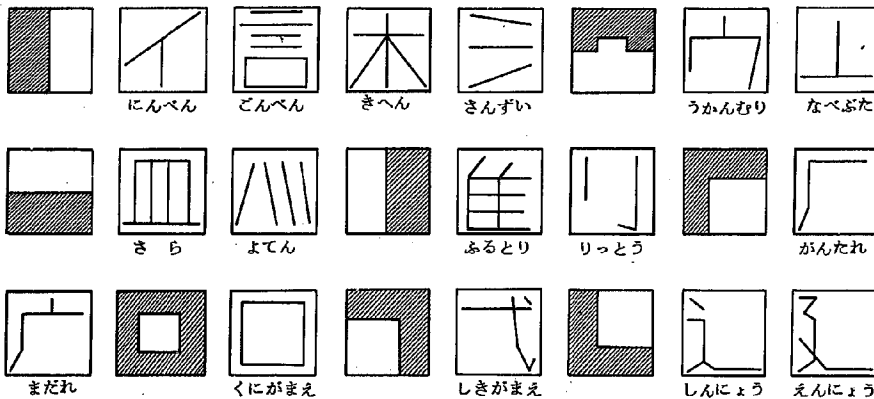
```

2.3.6 漢字の生成

(1) 生成方式

漢字の各部首（にんべん、うかんむり等）をパターン化（数値化）し、各部首を座標値で記憶させ部首以外の漢字はパターン化された部首の組合せにより生成する。

(1) 部首のパターン化



《部首パターン化例》

ALGOLによる“部首パターン”プログラム（関数手続）例

```

integer procedure K1;
begin
  real N;
  for N:=0,8,10,3,5,0,10,4,5,7,10,3,0,4,10,4,10,
    10,10,3,10,0,10,9 do PNO(N); K1:=1;
end;

```

```

integer procedure GON;
begin
  real N;
  for N:=0,8,10,3,2,10,10,4,8,10,10,3,10,8,5,
    10,4,0,8,5,10,3,2,7,10,4,8,7,10,3,8,5,5,10,4,
    2,5,5,10,3,1,4,10,4,9,4,5,0,1,0,1,4,
    10,3,10,0,10,9 do PNO(N); GON:=1;
end;

```

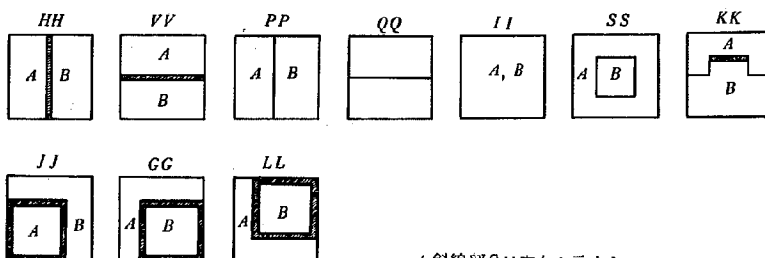


```
integer procedure ZKI;
begin real N;
  for N:=108,103,5,0,104,5,10,103,10,7,104,
    0,7,103,0,0,104,5,7,10,0,109 do
    PNO(N); ZKI:=1;
end;
```

空白

```
----- integer procedure B; 5:=1;
```

(ii) パターンの組合せ



(斜線部分は空白を示す)

<< パターン組合せ要素 >>

組合せ要素一覧表

組合せ要素名	パート	始点		縦長	横長	X軸との傾斜角度	備考
		X 軸	Y 軸				
HH	A	x_1	y_1	h	$9d/20$	k	
	B	$x_1 + 11d/20$	y_1	h	$9d/20$	k	
VV	A	x_1	$y_1 + 11d/20$	$9h/20$	d	k	
	B	x_1	y_1	$9h/20$	d	k	
PP	A	x_1	y_1	h	$d/2$	k	
	B	$x_1 + d/2$	y_1	h	$d/2$	k	
QQ	A	x_1	$y_1 + h/2$	$h/2$	d	k	
	B	x_1	y_1	$h/2$	d	k	
II	A	x_1	y_1	h	d	k	
	B	x_1	y_1	h	d	k	
SS	A	x_1	y_1	h	d	k	
	B	$x_1 + d/4$	$y_1 + h/4$	$h/2$	$d/2$	k	
KK	A	x_1	$y_1 + h/2$	$h/2$	d	k	
	B	x_1	y_1	$13h/20$	d	k	
LL	A	x_1	y_1	h	d	k	
	B	$x_1 + 7d/20$	$y_1 + 7d/20$	$11h/20$	$11d/20$	k	
JJ	A	x_1	$y_1 + 3h/4$	h	d	k	
	B	$x_1 + d/10$	$y_1 + h/10$	$11h/20$	$11d/20$	k	
GG	A	x_1	y_1	h	d	k	
	B	$x_1 + 7d/20$	$y_1 + h/10$	$11h/20$	$11d/20$	k	

(Ⅲ) ALGOLによるパターン組み合わせ関数

たとえば3番目にあげたPPは、指定領域を縦に2つに割ってその左の領域には最初のパラメタ、右の領域には2番目のパラメタによって指定されるものを記入したいわけである。ところが「松」のように右の領域にさらにQQを適用し上下2つの領域に分けて上に第1パラメタ、下に第2パラ

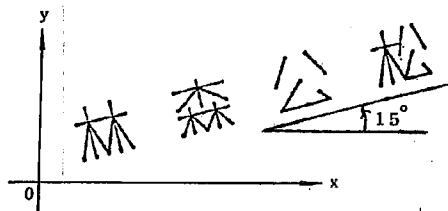
メタによって指定されるものを記入したり、さらに複雑な組合せを実現するためには、これらのPP, QQ等は関数にして置き、やりたい仕事の全部は関数の副作用として実行させ多重構造の回数、パターン組合せ関数DIVの値として得られるようにして置き、将来必要となった場合にそなえてある。

```
integer procedure DIV(N,A,B);
integer N,A,B;
begin real X,Y,H,D,XX,YY,HH,DD,KK;
  switch SW=D0,D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,
    D11,D12,D13,D14,D15,D16,D17,D18,D19;
  procedure END; go to OWARI;
  procedure OUT; go to MODORI;
  DD=D; go to SW(N+1);
  LSD(1,X,Y,H,D,KK); XX=X; YY=Y; HH=H;
  D10: DD=D/20; OUT; DO: XX=X+11*D/20; END;
  D11: YY=Y+11*H/20; OUT; D1: YY=Y; END;
  D12: DD=D/2; OUT; D2: XX=X+D/2; END;
  D13: YY=Y+H/2; HH=H/2; OUT; D3: YY=Y; END;
  D14: OUT; D4: END;
  D15: OUT;
  D5: XX=X+D/4; YY=Y+H/4; HH=H/2; DD=D/2; END;
  D16: YY=Y+H/2; HH=H/2; OUT;
  D6: YY=Y; HH=13*H/20; END;
  D17: OUT;
  D7: XX=X+7*D/20; YY=Y+7*H/20; HH=11*H/20; DD=11*D/20; END;
  D18: OUT;
  D8: XX=X+D/10; YY=Y+H/10; HH=11*H/20; DD=11*D/20; END;
  D19: OUT;
  D9: XX=X+7*D/20; YY=Y+H/10; HH=11*H/20; DD=11*D/20; END;
  MODORI: LSD(0,XX,YY,HH,DD,KK); DIV=A; go to SW(N+1);
  OWARI: LSD(0,XX,YY,HH,DD,KK); DIV=DIV+B; PMO(X+D,Y);
end;

integer procedure HH(A,B); integer A,B; HH=DIV(0,A,B);
integer procedure VV(A,B); integer A,B; VV=DIV(1,A,B);
integer procedure PP(A,B); integer A,B; PP=DIV(2,A,B);
integer procedure QQ(A,B); integer A,B; QQ=DIV(3,A,B);
integer procedure II(A,B); integer A,B; II=DIV(4,A,B);
integer procedure SS(A,B); integer A,B; SS=DIV(5,A,B);
integer procedure KK(A,B); integer A,B; KK=DIV(6,A,B);
integer procedure LL(A,B); integer A,B; LL=DIV(7,A,B);
integer procedure JJ(A,B); integer A,B; JJ=DIV(8,A,B);
integer procedure GG(A,B); integer A,B; GG=DIV(9,A,B);
```

(2) 漢字生成プログラム例

(例題) X軸と15°、大きさ10mmの「林、森、公、松」を描くプログラムを作成せよ。



```

begin
  external PREADY, PMOVE, SINE, COSINE,
    PNO, KHA, KMU, ZKI, PENH;
  real MX, MY;
  procedure KANZI(A, S, DP);
    real A, S; integer DP;
    begin real X, Y; integer PP;
      external XYM, LSD;
      XYM(1, X, Y);
      LSD(0, X, Y, S, S, A); Z:=DP;
    end;
  PREADY(0, 0);
  MX:=15*COSINE(10); MY:=15*SINE(10);
  KANZI(15, 10, PP(ZKI, ZKI));
  PMOVE(MX, MY);
  KANZI(15, 10, PP(ZKI, PP(ZKI, ZKI)));
  PMOVE(2*MX, 2*MY);
  KANZI(15, 10, QQ(KHA, KMU));
  PMOVE(3*MX, 3*MY);
  KANZI(15, 10, PP(ZKI, QQ(KHA, KMU)));
  PENH;
end;

```

<<外部関数の説明>>

SINE, COSINE : 角度(°)のSIN, COS.
 PNO : 座標変換および紙テープパンチ。
 KHA, KMU, ZKI : 部首パターン関数(ハ, ム, 木)。
 PP, QQ : 組合せ関数(左右分割, 上下分割)である。例えば左右分割(PP)の組合せ関数を示す。

```

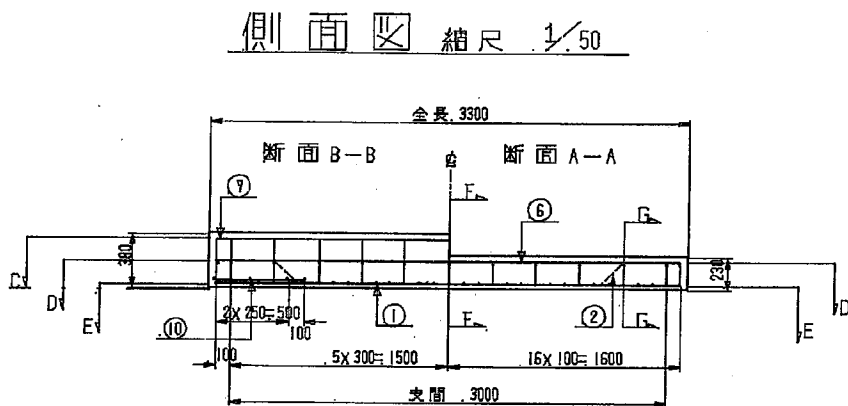
real procedure PP(P1, P2);
  real P1, P2;
  begin real X, Y, H, D, K;
    LSD(1, X, Y, H, D, K);
    D:=D/2; LSD(0, X, Y, H, D, K); PP:=P1;
    X:=X+D; LSD(0, X, Y, H, D, K); PP:=PP+P2;
  end;

```

LSD : 手続XYMと同様な内容であり、
 部首の組合せに必要な始点・高
 さ・幅・傾斜角を格納および取
 り出す。

2・3・7 図化例

以上の図化手続および漢字生成手法を使って、鉄筋コンクリート単純スラブ橋の側面図（図・2・1）を描くプログラムを紹介すると次の通りである。



図・2・1 図化機によって描いたRC単純スラブ橋の側面図

```

procedure SOKUHENZU(L,LL,U,WL,S,MAS,HHAS,HAS,YAS,D,YL,YE1,YE2,
                    YE3,YE4,HL1,HL2,HL3,HL4,HU1,HU2,HU3,HU4,N);
    real L,LL,U,WL,S,MAS,HHAS,HAS,YAS,D,YL,YE1,YE2,
        YE3,YE4,HL1,HL2,HL3,HL4,HU1,HU2,HU3,HU4,N;
begin
    procedure PM(X,Y);
        PMOVEの略
    procedure P(X,Y);
        PSENXの略
    procedure TEN(X,Y);
        鉄筋断面を描く手続き
    procedure SMZS(A,S,X,Y);
        「側面図」という字を描く手続き
    procedure SKN(A,S,X,Y);
        「支間」という字を描く手続き
    procedure ZNT(A,S,X,Y);
        「全長」という字を描く手続き
    procedure DM(A,S,X,Y);
        「断面」という字を描く手続き
    procedure SCALE(X,Y,A,N);
        縮尺を描く手続き
    procedure SNPD(X,Y,A,N);
        寸法線を描く手続き
    procedure HG(X,Y,N,A);
        数字を0°方向に描く手続き
    procedure VG(X,Y,N,A);
        数字を270°方向に描く手続き
    procedure PHSEN(N,X1,Y1,X2,Y2);
        破線を描く手続き
    procedure HIKITE(X1,Y1,X2,Y2,S,N);
        引出し線を描く手続き
    procedure ERNN(X,Y,N,A);
        円の中に数字を描く手続き

    real A,LL1,D1,H,K,K1,DD1,X,Y,YL1,PP,SS1,SS2,SS3,SS4,SS5,L1;
    array XA,YA[1:31];
    if L≤10 ∧ L>7 ∧ W≤17 ∧ W>10 then A=100;
    if L≤10 ∧ L>7 ∧ W≤10 then A=80;
    if L≤7 ∧ L>5 ∧ W≤17 ∧ W>8 then A=80;
    if L≤7 ∧ L>5 ∧ W≤8 then A=50;
    if L≤5 ∧ W≤17 ∧ W>6 then A=100;
    if L≤5 ∧ W≤6 then A=50;
    LL1=LL*1000/A;
    H=3.5*10/A; K=15*10/A;
    D1=D*10/A;
    K1=5*10/A; DD1=D1+K; X=X;
    Y=20; L1=L*1000/A; YL1=YL*1000/A;
    SS1=(L/10)*1000/A; SS2=((LL-L)-0.10)/2)*1000/A;
    SS3=(0-(3.5*2))*10/A; SS4=SS1+SS2+SS3; SS5=SS1+SS2;

```

パラメタ規制部

手続き立書部

変数宣言部

縮尺および数値計算

```

PEN2:
PM((LL1/2-30)*N,(DD1+25)*N); P((LL1/2-30+55)*N,(DD1+25)*N);
SMZS(0,N,(LL1/2-28)*N,(DD1+25.3)*N);
SCALE((LL1/2-28+40)*N,(DD1+25.3)*N,A,N);
PSEN(0,0,LL1*N,0); PSEN(LL1*N,0,LL1*N,D1*N);
PSEN(LL1*N,D1*N,(LL1/2)*N,D1*N);
PSEN((LL1/2)*N,(D1+K)*N,0,(D1+K)*N); PSEN(0,(D1+K)*N,0,0);
PEN3:
PSEN((K1*N),H*N,(LL1-K1)*N,H*N);
PSEN((LL1-K1)*N,H*N,(LL1-K1)*N,(D1-H)*N);
PSEN((LL1-K1)*N,(D1-H)*N,K1*N,(D1-H)*N);
PSEN(K1*N,H*N,K1*N,(D1-H)*N);
PSEN(K1*N,(D1-H)*N,(LL1/2)*N,(D1-H)*N);
PM((LL1/2)*N,H*N);
for PP=0 step (HHAS*3*10/A) until (LL1/2-K1) do
begin
PSEN((LL1/2+PP)*N,H*N,(LL1/2+PP)*N,(D1-H)*N);
TEN((LL1/2+PP)*N,(D1-H-0.3)*N);
end;
PM((LL1/2)*N,H*N);
for PP=0 step (HHAS*3*10/A) until (LL1/2-K1) do
PSEN((LL1/2-PP)*N,H*N,(LL1/2-PP)*N,(D1-H)*N);
PM((LL1/2)*N,(H+0.2)*N);
for PP=0 step (HHAS*10/A) until (LL1/2-K1) do
TEN((LL1/2+PP1)*N,(H+0.2)*N);
PM((LL1/2)*N,(H+0.2)*N);
for PP=0 step (HHAS*10/A) until (LL1/2-K1) do
TEN((LL1/2-PP1)*N,(H+0.2)*N);
PM((LL1/2)*N,(D1-H)*N);
for PP=0 step (HHAS*3*10/A) until (LL1/2-K1) do
TEN((LL1/2-PP)*N,(D1-H-0.3)*N);
TEN((LL1-K1-0.2)*N,(H+0.2)*N);
TEN((LL1-K1-0.2)*N,(D1-H-0.3)*N);
TEN((K1+0.2)*N,(H+0.2)*N);
TEN((K1+0.2)*N,(D1-H-0.3)*N);

if S=90 then go to YY;
PSEN(K1*N,(H+0.5)*N,(K1+YL1)*N,(H+0.5)*N);
PM(K1*N,(H+0.5)*N);
for PP=0 step (YAS*10/A) until YL1 do
TEN((H+YB)*N,(H+0.7)*N);
TEN((K1+YL1)*N,(H+0.7)*N);
PEN1:
XA[1]=K1*N; YA[1]=H*N; XA[2]=(K1+(YE1*YE2)/A)*N;
YA[2]=H*N; SNPO(XA,YA,-5*N,1);
HG(K1*N,-4.2*N,N,YE1); HG((K1+3)*N,-4.2*N,N,YE2);
HG((K1+7.5)*N,-4.2*N,N,YE3);
HIKITE((YL1/2)*N,(H+0.5)*N,(YL1/2)*N,-7*N,7*N,2);
EENN(-3*N,-5.5*N,N,10);
if YE4=0 then go to YY;
XA[1]=(K1+(YE1*YE2)/A)*N; YA[1]=H*N; XA[2]=(K1+YL1)*N;
YA[2]=(H+0.5)*N; SNPO(XA,YA,-7*N,1);
YY:
XA[1]=(LL1/2-HU3/A)*N; YA[1]=H*N;
XA[2]=(LL1/2)*N; YA[2]=H*N; SNPO(XA,YA,-11*N,1);
HG((K1+15)*N,-10.2*N,N,HU1);
HG((K1+18)*N,-10.2*N,N,HU2);
HG((K1+23)*N,-10.2*N,N,HU3);
if HU4=0 then go to YY1;
XA[1]=(LL1/2-HU3/A)*N; YA[1]=H*N; XA[2]=K1*N;
YA[2]=-11*N; SNPO(XA,YA,-11*N,1);
HG((K1-1)*N,-10*N,N,HU4);

```

構造線を描く

主鉄筋、配力鉄筋、
用心鉄筋等の鉄筋線
と鉄筋断面図を描く

```

YY1:
XA[1]=(LL1/2)*N; YA[1]=0; XA[2]=(LL1/2+HL3/A)*N;
YA[2]=0; SNPO(XA,YA,-11*N,1);
HG((LL1/2+10)*N,-10.2*N,N,HL1);
HG((LL1/2+14)*N,-10.2*N,N,HL2);
HG((LL1/2+19)*N,-10.2*N,N,HL3);
if HL4=0 then go to YY2;
XA[1]=(LL1/2+HL3/A)*N; YA[1]=H*N;
XA[2]=(LL1/2+HL3/A+HL4/A)*N;
YA[2]=H*N; SNPO(XA,YA,-11*N,1);
HG((LL1/2+HL3/A)*N,-10*N,N,HL4);
YY2:
PEN3:
for P=1 step 1 until 3 do
  PHSEN(N,(K1+SS5)*N,(D1-H)*N,(K1+SS4)*N,H*N);
for P=1 step 1 until 3 do
  PHSEN(N,(LL1-K1-SS5)*N,(D1-H)*N,(LL1-K1-SS4)*N,H*N);
PEN1:
XA[1]=0; YA[1]=DD1*N; XA[2]=LL1*N; YA[2]=DD1*N;
SNPO(XA,YA,16*N,1);
ZNT(0,N,(LL1/2-5)*N,(DD1+15.8)*N);
HG((LL1/2)*N,(DD1+15.8)*N,N,LL*1000);

XA[1]=(LL1/2-L1/2)*N; YA[1]=0;
XA[2]=(LL1/2-L1/2)+L1)*N; YA[2]=0;
SNPO(XA,YA,-16*N,1);
SKN(0,N,(LL1/2-5)*N,-15.2*N);
HG((LL1/2+2)*N,-15.2*N,N,L*1000);
XA[1]=0; YA[1]=DD1*N; XA[2]=0; YA[2]=0;
SNPO(XA,YA,-11*N,1);
UG(-10.5*N,2*N,N,(D+15)*10);
XA[1]=LL1*N; YA[1]=0; XA[2]=LL1*N; YA[2]=D1*N;
SNPO(XA,YA,6*N,1);
UG((LL1+5)*N,0.5*N,N,D*10);
PM(LL1*N,H*N); P((LL1+15)*N,H*N);
P((LL1+15)*N,-7*N);
P((LL1+15.5)*N,-5.5*N); P((LL1+15)*N,-5.5*N);
HIKITE((LL1/5+10)*N,H*N,(LL1/5+10)*N,-5*N,5*N,N);
EENN((LL1/5+14)*N,-3.5*N,N,1);
HIKITE((LL1/4-15)*N,(DD1-H)*N,
      (LL1/2+10)*N,(DD1+5)*N,5*N,N);
EENN((LL1/4-11)*N,(DD1+6.5)*N,N,7);
HIKITE((LL1/2+10)*N,(D1-H)*N,
      (LL1/2+10)*N,(D1+5)*N,5*N,N);
EENN((LL1/2+14)*N,(D1+6.5)*N,N,6);
YB=D1-(35+50)/A;
HIKITE((LL1-K1-SS5-YB/2)*N,(D1/2)*N,
      (LL1-K1-SS5-YB/2)*N,-5*N,5*N,N);
EENN((LL1-K1-SS5-YB/2-1)*N,-3.5*N,N,2);
DM(0,N,(LL1/4-9)*N,(DD1+10)*N);
PRINTS('SOKUMENZU**00ARI'); GRLF(10);
end;

```

引き出し線、寸法線
数字、文字等を描く

2・4 最適化手法

土木構造物の設計では、設計条件を満足する値は、一意的に定まることなく、通常多数存在し、しかも設計計算に先立ってまづ断面等の値を仮定し、計算終了後当初仮定した値と計算値との差が許容限度以内になるまで計算を繰り返さなければならない。従ってすぐれた設計者は多年の設計経験にもとづく勘によって最適（真の最適設計ではない）に近いものを求める手法を自然に会得しており、繰り返し回数による設計時間や設計精度において未経験者との間に大きな相違が認められた。しかし電子計算機組織の発達により、極めて短時間の間に繰り返し計算が可能となったので、熟練者との差異が急速にうづめられ、また初期に行われていた人力の単なる肩替りから、人間・機械系すなわち人間と電子計算機が一体となって互いに情報の交換を行ない、当初人間が考えていた設計よりも質的に一段と向上した設計手法の創造が期待されるようになった。

従来の土木構造物設計では、実応力を許容応力に近づけ、材料を少なくするのがよい設計と考えられ、原材料の価格差（調達、在庫）、運賃、仮設費、労力、機械の調達、工程計画、管理面など工事におよぶ多くの重要な要素については、あまりにも因子が多過ぎるという理由で省略し、また設計に際しては計算の便宜から、構造物を任意の部分に分解し、（場合によっては実状とかなりかけはなれた分割をしているときもある）、各部分の最適化（現在における真の最適化ではない）を試み、それら部分の集合である構造物も亦最適になると考えられていた。

しかしシステム工学の普及により、土木設計の分野にもシステム工学の手法を導入し、設計過程をシステム¹⁾としてとらえることにより、その最適化をはからうという試みがなされてきた。つまり設計の最適化とは、 n 次元のベクトル変数 x が、ある制限領域 R^0 に属するとき、 R^0 の上で定義される関数 $f(x)$ の最小値（最大値） $f(x^0)$ を求めること²⁾と定義することができよう。

具体的には、設計条件をつぎのような制約条件式として表現し、

$$g_i(x) \leq 0 \quad (i=1, 2, \dots, m) \\ x \geq 0$$

ただし $x^1 = (x_1, x_2, \dots, x_n)$

$(x_1, x_2, \dots, x_n)$ は設計変数

このベクトル変数 x により定まる n 次元空間の制限領域内に、多数存在する設計値のうちから目的関数 $f(x)$ を最小（最大）とする解ベクトル x^0 を求めることに帰着する。

数 理 計 画 法	線 型 計 画 法
	非 線 型 計 画 法
試 行 法	動 的 計 画 法
	単 一 変 数 変 化 法
	山 登 り 法

ここでよく知られている最適化法の主なものを表示すると表-1のように、(1)目的関数と制御変数に関する制限条件式を求め、これらを解析的に解く数理計画法、(2)システムそのもの、あるいはシステムのシミュレートまたは数学的モデルについて試行錯誤的に各制御変数を変化させ、それに対して目的関数を最大または最小にする制御変数値を求める試行法とがある。

土木構造物設計に於ては、2次元モーメントを考慮する場合が多く、最適性を追求するのに線型として取り扱える場合がまれであり、ほとんどが非線型しかも市販品の形状寸法や施工性の関係上、変数は連続でなく、離散的（例えば鉄筋コンクリート構造物における鉄筋径は、13, 16, 19, …… mm であり、また鉄筋配置は現場における施工性を考え、10, 125, 15, …… cm 間隔に用いられる）、であるという特徴をもっているため、土木構造物設計の最適性を調べるのに今迄よく使われてきた試行法を中心にして、従来の設計法との比較を試みる。

2・4・1 単一変数変化法

単一変数変化法では、(i)制御変数 n 個からなる n 次元ベクトル $x^1 = (x_1, x_2, \dots, x_n)$ のうち、いづれか一つの元 x_i のみを Δx_i だけ変え、目的関数 $f(x)$ が増加または減少すれば（以下これを「成功」という）、さらに x_i を Δx_i だけ変えて、 $f(x)$ が減少または増加（以下「失敗」という）するまで続ける。もし x_i が制限範囲をこせば、その前に x_i に与えに変化 Δx_i を打消す変化 $-\Delta x_i$ を x_i にあたえた上で、以後これを一定とし、次に他の元 x_j ($j \neq i$)のみを Δx_j だけ変え、 x_j と同様の操作を続ける。(ii)どの変数 x_i を変化しても失敗が起こるようになれば、 Δx_i ($i=1, 2, \dots, n$)さらに小さくして(i)の操作をくり返すことである。²⁾

2・4・2 最大傾斜法

最大傾斜法は制御変数ベクトル $x = (x_1, x_2, \dots, x_n)$ の元 x_i を速く最適値に接近させるため、目的関数 $f(x)$ の好ましい方向の変化が最大となるように x_i を変えていく方法である。

いま目的関数 $f(x)$ を最大または最小とするため試行

をくり返し、第 p 番目の試行点における制御変数ベクトルを $\mathbf{x}^p = (x_1^p, x_2^p, \dots, x_n^p)$, ($p=0, 1, 2, \dots$, 試行の出発点では $p=0$) とし、 $\mathbf{x}^{p+1} = \mathbf{x}^p + d\mathbf{x}^p$ を Taylor 展開する。

$$\begin{aligned} f(\mathbf{x}^{p+1}) &= f(\mathbf{x}^p + d\mathbf{x}^p) = f(x_1^p + dx_1^p, x_2^p + dx_2^p, \\ &\dots, x_n^p + dx_n^p) = f(x_1^p, x_2^p, \dots, x_n^p) \\ &+ (dx_1^p \frac{\partial f}{\partial x_1^p} + dx_2^p \frac{\partial f}{\partial x_2^p} + \dots \dots \dots \\ &+ dx_n^p \frac{\partial f}{\partial x_n^p}) f(x_1^p, x_2^p, \dots, x_n^p) + \dots \dots \dots \\ &+ (R-1)! (dx_1^p \frac{\partial}{\partial x_1^p} + dx_2^p \frac{\partial}{\partial x_2^p} + \dots \dots \dots)^{R-1} \\ &f(x_1^p, x_2^p, \dots, x_n^p) + \frac{1}{R!} (dx_1^p \frac{\partial}{\partial x_1^p} \\ &+ dx_2^p \frac{\partial}{\partial x_2^p} + \dots \dots \dots + dx_n^p \frac{\partial}{\partial x_n^p})^R \\ &f(x_1^p + \theta_1 dx_1^p + x_2^p + \theta_2 dx_2^p \\ &+ \dots \dots \dots x_n^p + \theta_n dx_n^p) \end{aligned}$$

$$0 < \theta_1, \theta_2, \dots, \theta_n < 1$$

このうち第3項以下を省略し $f(\mathbf{x}^{p+1})$ と $f(\mathbf{x}^p)$ の差を求めると

$$\begin{aligned} df^p &= f(\mathbf{x}^{p+1}) - f(\mathbf{x}^p) \approx (\frac{\partial f}{\partial x_1})_p dx_1^p \\ &+ (\frac{\partial f}{\partial x_2})_p dx_2^p + \dots \dots \dots + (\frac{\partial f}{\partial x_n})_p dx_n^p \end{aligned}$$

となり、目的関数の変化率 $df^p / \|R\|$ は $\frac{df^p}{\|R\|} = (\frac{\partial f}{\partial x_1})_p \frac{dx_1^p}{\|R\|} + (\frac{\partial f}{\partial x_2})_p \frac{dx_2^p}{\|R\|} + \dots \dots \dots + (\frac{\partial f}{\partial x_n})_p \frac{dx_n^p}{\|R\|}$

である。

ここでは $\|R\|$ はノルムを示し

$$\|R\| = \sqrt{(dx_1^p)^2 + (dx_2^p)^2 + \dots \dots \dots + (dx_n^p)^2}$$

であたえられる。

この変化率を最大にするためには、 dx^p の値を $(\frac{\partial f}{\partial \mathbf{x}})$ に比例させればよい。

これを実際に適用するには

(i) 目的関数 $Z=f(\mathbf{x})$ を $\mathbf{x}$ で微分して $\frac{\partial f}{\partial \mathbf{x}}$ を求める。

(ii) $(\frac{\partial f}{\partial \mathbf{x}})$ に適当な初期値を代入し $(\frac{\partial f}{\partial \mathbf{x}})_0$ を求める。

(iii) $f(\mathbf{x}^0)$ と $f(\mathbf{x}^1) = f[\mathbf{x}^0 + h(\frac{\partial f}{\partial \mathbf{x}})_0]$ を比較して

(イ) 成功ならば

$$f(\mathbf{x}^2) = f[\mathbf{x}^1 + h(\frac{\partial f}{\partial \mathbf{x}})_1],$$

$$f(\mathbf{x}^2) = f[\mathbf{x}^2 + h(\frac{\partial f}{\partial \mathbf{x}})_2]$$

をくり返す。

(ロ) 失敗ならば (第 $i+1$ 回目), 失敗前 (第 i 回) の値にもどし,

$$f(\mathbf{x}_i) \text{ と } f[\mathbf{x}^{i+1} + h(\frac{\partial f}{\partial \mathbf{x}})_i]$$

を比較し失敗するまでくり返す。

(ハ) ただし $\mathbf{x}$ の元が制限範囲をこえれば、こえに制限範囲は以後制限範囲の手前に固定する。

を実施すればよい。

2・4・3 最適傾斜法

最大傾斜法では試行1回毎に傾斜の計算が必要であり、制御変数の多いときは計算量が非常に多くなってくるので、一つのパラメタ h を最適化して $d\mathbf{x}$ を求める方法である。

(i) $Z=f(\mathbf{x})$ を $\mathbf{x}$ で偏微分して $\mathbf{y}$ の傾斜

$\frac{\partial f}{\partial \mathbf{x}}$ を求める。

(ii) 最大傾斜の方向関数 f のこの配ベクトル ∇f^0

$$= (\frac{\partial f}{\partial x_1})_0, (\frac{\partial f}{\partial x_2})_0, \dots, (\frac{\partial f}{\partial x_n})_0$$

を含む断面上で $\mathbf{y}$ の最小値を求める。

すなわち

$$dx_1 = h(\frac{\partial f}{\partial x_1})_0$$

$$dx_2 = h(\frac{\partial f}{\partial x_2})_0$$

$$dx_n = h(\frac{\partial f}{\partial x_n})_0$$

とし $\mathbf{y} = f(\mathbf{x})$ が最小となるよう h を定める。

(iii) あらためて $\mathbf{x}^0 = \mathbf{x}^0 + d\mathbf{x}^0$ において (i) にもどる。^{3) 4)}

2・4・4 共役傾斜法

最適傾斜法は繰り返し数が非常に多くなるので、2次計画法の原理、すなわち $Z=f(\mathbf{x})$ が $\mathbf{x}$ の2次形式のときは1パラメタの最適化を n 回繰り返すと Z の最小値が得られることを応用し、目的関数 $Z=f(\mathbf{x})$ が2次形式で近似し得るときこの方法は収束が速くて有利である。いま $f(\mathbf{x})$ が2次形式と仮定すると、つぎのようにかける。

$$Z=f(\mathbf{x}) = \mathbf{x}' \cdot \mathbf{A} \cdot \mathbf{x} + \mathbf{b}' \cdot \mathbf{x} + Z_0$$

ただし $\mathbf{b}' = (b_1, b_2, \dots, b_n)$

$$\mathbf{A} = (a_{ij}) (i, j=1, 2, \dots, n)$$

$$\text{いま } \mathbf{x} = \mathbf{x}^0 + \sum_1 h_1 \mathbf{a}_1 \text{ とおくと}$$

ただし $d = (d_{11}, d_{12}, \dots, d_{1n})$
 $Z = f(x)$ は
 $Z = (x^0 + Ax^0 + b'x^0 + Z^0) + \sum_i (h_i d_i A_i + h_i d_i + Ax^0 + h_i x^0 A d_i + h_i b' d_i) + \sum_{i,j} h_i h_j d_i A d_j$
 となる。

また

$$d_i A d_j = \delta_{ij}$$

の仮定が成立すれば

$$Z = g(x^0) + \sum_i g_i(h_i d_i)$$

と書け、 $g(x^0)$ は前式の右辺第1項を、 $g_i(h_i d_i)$ は同じく第2項の()内の部分を表す。この式より Z は $h_i d_i$ ($i=1, \dots, n$) の関数であるから、 Z の最小値を求めるには $h = (h_1, h_2, \dots, h_n)'$ を求めることになる。

よって試行手順は

- (I) 適当な方向を d_1 にえらぶ(通常最大傾斜の方向ベクトルを d_1 とする)。
- (II) d_1 の方向を含む断面上で Z の最小値を求める(実際には d_1 は最大傾斜の方向にとる)。
- (III) (II)で達した点上で、 d_1 に共役な方向 d_2 をみいだす。
- (IV) d_2 を含む断面上で(II)の手順を行なう。
- (V) 以下同様の手順を繰り返す。

このようにすれば、目的関数 $f(x)$ が2次形式で近似できるとはほぼ n 回(II), (III)の手順を繰り返すことにより、最小値に達しうる。

つぎに(II)の共役な方向を見出す1方法とし PARTAN 法(Parallel tangent)がある。いま図のように、ある平面上で Z の等高線が同心の楕円になる場合を考える。

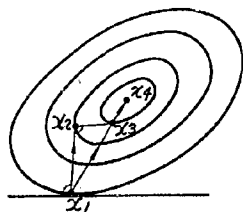


図9.11 最大傾斜と共役傾斜

任意の試行点 x_1 を選び、最速傾斜法によって最大傾斜方向における最小値 x_2 を求める。同様に x_2 から最大傾斜の方向に x_3 を選ぶ。ここで x_1, x_2 を結ぶならば楕円の性質からベクトル x_1, x_3 は最小点である頂点を通る。したがって、このベクトル上で最小点 x_4 を求めると、これが最小値を与える。このベクトルの方向が共役傾斜の方向である。

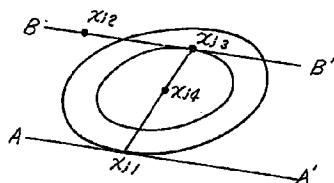


図9.12 PARTAN 法

PARTAN 法を一般的に述べると、図9.12 のように任意の1点 x_1 をとり、この点を通して等高線に接線 AA' をひく。さらに任意の1点 x_2 をとり、この点を通して AA' に平行な線をひき、これを BB' とする。 BB' 上で Z を最小にする点 x_3 をとり、さらに x_1, x_3 を結ぶ線上で Z を最小にする点 x_4 をとれば、これが求める最小値である。

一般に n 次元の場合は $2n$ 回の試行で頂点に達する。

2.4.5 EVOP

システムがある条件で運用されていて、目的関数の式が明らかでない場合に、このシステムの制御変数を最適値にし、かつ以後この条件で最適運用を続けさせるためには EVOP によるのがよい。

EVOP は次の順序で進められる。

- (I) システムが現在運用されている点(現在の制御変数の組み合わせ)の付近における $f(x)$ の式を実測および計算によって決定する。
- (II) この $f(x)$ の式を調べて、 $f(x)$ がより大きくなる制御変数の組み合わせを探し、制御変数をこの値に変化し、この点でシステムを運用する。
- (III) さらに(I)(II)の手順をくり返し、 $f(x)$ が最大になるまで続ける $f(x)$ が最大になっても非制御変数の変動により最適点が動くから以上の手順をくり返し続ける。

このような手順、ちょうど生物の進化の過程に似ていることから evolutionary operation (EVOP) とよばれるようになったのである。

この方法を応用したものに最速傾斜法があり、まず適当な初期値を定め、各因子について1ピッチづつ正負方向に動かした点の応答を計算させ、この応答に1次平面を当てはめ、ノイズを無視してよいかどうかの検定を分散分析により行ない、ノイズを無視して良ければ最急コウ配の方向に進み、極点が出たらここでまた周辺の応答を計算させ、1次平面の当てはめを行なうが、もしノ

イズが無視できないくらいに大きければ、その極点および軸方向にさらに1ピッチ、外側に進んだ点の応答を計算し、これらによって2次曲面の当てはめを試み、ノイズが無視できるくらいに小さければこの2次曲面の停留点に原点を移して各軸方向に一歩ずつふみ出して、どの場合も応答が小さくなればその点を停留点とする方法である。

2・4・6 最適化手法の適用例

- (1) 単一変数変化法による扶壁式鉄筋コンクリート擁壁前壁(単鉄筋コンクリート床版)の最適設計

制限条件式

$$\frac{2M}{bx(d-x/3)} - \sigma_{ca} \leq 0$$

$$\frac{M}{As(d-x/3)} - \sigma_{sa} \leq 0$$

$$As \geq 4.22 (\phi 13 \text{ 鉄筋を } 33.3 \text{ cm 間隔に用いる})$$

$$d \geq 20 \text{ (施工性よりきまる最小値)}$$

ここで

$$x = \frac{nAs}{b} \left[-1 + \sqrt{1 + \frac{2bd}{nAs}} \right]$$

なる拘束のもとで

目的関数

$$Z = CAs \cdot As + Cc \cdot b \cdot (d + d')$$

ここで CAs : 鉄筋工単価

Cc : コンクリート工単価

d' : 鉄筋のかぶり

Z (工費) を最小にする d と As を求めることである。(型わくは前壁のため、工費の比較には無関係であり、省略した)

ただし、 d は施工性を考え、最小を 20 cm とし、25 cm きざみに増し、かぶり (d') は 5 cm、また鉄筋量は $\phi 13 \sim 32 \text{ mm}$ 筋につき、10、12.5、15、20、25、33.3 cm

間隔に配置すると、鉄筋量については

表-2.2 鉄筋量 (100 cm 当り) (1)

鉄筋本数 (間隔)	10本 (10 cm)	8本 (12.5 cm)	6.67本 (15 cm)	5本 (20 cm)	4本 (25 cm)	3本 (33.3 cm)
$\phi 13$	12.67	10.14	8.54	6.34	5.07	4.22
16	19.86	15.89	13.24	9.93	7.94	6.22
19	28.65	22.92	19.10	14.33	11.46	9.55
22	38.71	30.97	25.81	19.36	15.48	12.95
25	50.67	40.54	33.78	25.34	20.27	16.88
29	64.24	51.39	42.83	32.12	25.70	21.42
32	79.42	63.54	52.95	39.71	31.77	26.48

であり、鉄推量の順に、差の小さいものを無視して並べ換えると

表-2.2 鉄筋量 (2)

4.22, $\phi 13, 33.3$	10.14, $\phi 13, 12.5$	20.27, $\phi 25, 25$	32.12, $\phi 29, 20$
5.06, $\phi 13, 25$	11.46, $\phi 19, 25$	21.42, $\phi 29, 33.3$	33.78, $\phi 25, 15$
6.34, $\phi 13, 20$	12.67, $\phi 13, 10$	22.92, $\phi 19, 12.5$	38.71, $\phi 22, 10$
6.62, $\phi 16, 33.3$	13.24, $\phi 16, 15$	25.34, $\phi 25, 20$	39.71, $\phi 32, 20$
7.94, $\phi 16, 25$	14.33, $\phi 19, 20$	26.48, $\phi 32, 33.3$	40.54, $\phi 32, 12.5$
8.45, $\phi 13, 15$	15.89, $\phi 16, 12.5$	28.65, $\phi 19, 10$	42.83, $\phi 29, 15$
9.55, $\phi 19, 33.3$	16.88, $\phi 25, 33.3$	30.97, $\phi 22, 12.5$	51.39, $\phi 29, 12.5$
9.93, $\phi 16, 20$	19.10, $\phi 19, 15$	31.77, $\phi 32, 25$	64.24, $\phi 29, 10$
			79.42, $\phi 32, 10$

のようになる。(d , As いずれも離散量)

入力として $\sigma_{sa} = 1600 \text{ ㏫}$, $\sigma_{ca} = 70 \text{ ㏫}$,

$n = 15$, 土床係数 0.33, 扶壁中心間隔 $\ell = 3.6 \text{ m}$,

前壁高 $H = 8 \text{ m}$, 初期値は通常用いられる鉄筋コンクリ

ート工学の公式 $d = C_1 \sqrt{M/b}$

$$C_1 = \frac{\sigma_s + n\sigma_c}{n\sigma_c} \sqrt{\frac{\sigma_n}{3\sigma_s + 2n\sigma_c}}$$

$$As = C_2 \sqrt{M \cdot b}$$

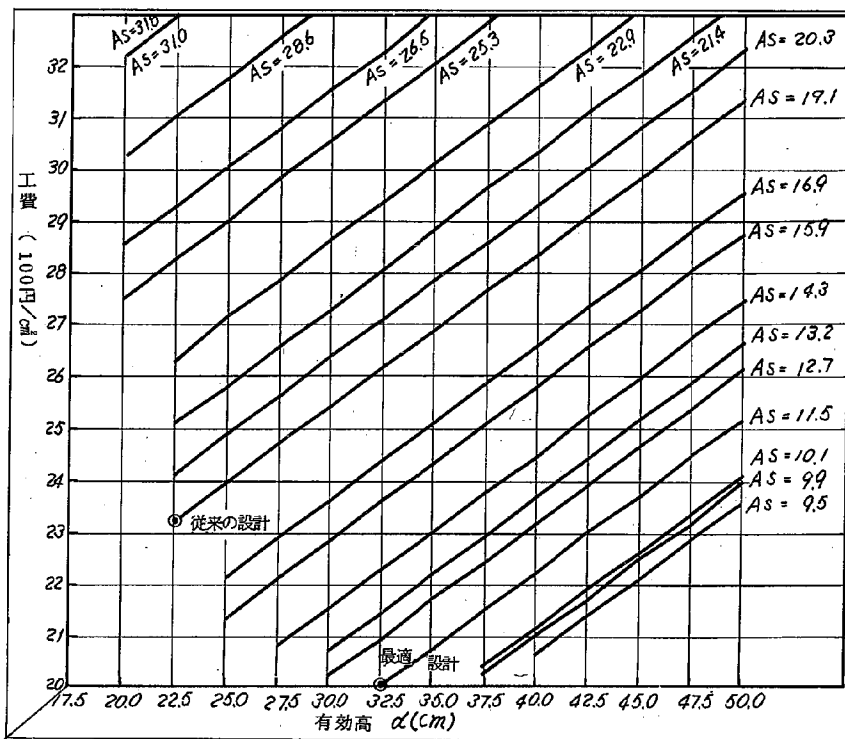
$$C_2 = \frac{\sigma_c}{2\sigma_s} \sqrt{\frac{\sigma_n}{3\sigma_s + 2n\sigma_c}}$$

より、1級さかのぼる（ d は25 cm短かく、
 As は表-2.2より一つ前の値）程度でよい。
 さらに鉄筋Ⅰ80,000 N/t 、コンクリートⅠ
 3000 N/m^2 をいれて計算すると表-2.3のよう
 になり、かなりの差を生ずる。

表-2.3 工費の比較

	有効高 d (cm)	鉄筋量 As (cm ²)	実応力 (MPa)		工 費 円/ m^2
			σ_s	σ_c	
従来の設計	22.5	19.1	1,443	68.5	2,758.76
最適設計	32.5	11.5	1,594	40.6	2,000.59

変数が d 、 As との2個につき、すべての場合について計算したものを図示する。



(2) 最大傾斜法による扶壁式鉄筋コンクリート擁壁の前壁と扶壁との最適設計例

この場合は単鉄筋T型ばりにおける最適設計の問題であり、中立軸の位置によってT形ばりが長方形ばりになる。

(i) 中立軸が像(前壁)内にある場合

この場合は長方形ばりであり、制限条件式は

前例の前壁と

同様であるが

制御変数には

A_{s1} と d のほ

かに扶壁間隔

ℓ と腹壁厚 b

と、突縁(前壁厚) t と A_{s2} が加わる。

制限条件式

$$\frac{2M_1}{\ell x_1 (t - x_1 / 3)} - \sigma_c a \leq 0$$

$$\frac{M_1}{\ell s_1 (t - x_1 / 3)} - \sigma_c a \leq 0$$

$$\frac{2M_2}{\ell x_2 (d - x_2 / 3)} - \sigma_c a \leq 0$$

$$\frac{M_2}{A_{s2} (d - x_2 / 3)} - \sigma_c a \leq 0$$

$$A_{s1} \geq 4.22$$

$$A_{s2} \geq 4.22$$

$$t \geq 20$$

$$d \geq 20$$

$$b \geq 30$$

ここで

$$x_1 = \frac{n A_{s1}}{\ell} \left\{ -1 + \sqrt{1 + \frac{2 \ell t}{n A_{s1}}} \right\}$$

$$x_2 = \frac{n A_{s2}}{\ell} \left\{ -1 + \sqrt{1 + \frac{2 \ell d}{n A_{s2}}} \right\}$$

$$M_1 = -\frac{0.8}{10} \cdot p \cdot \ell^2$$

$$M_2 = -\frac{2 p_A h + p_A}{6} \ell^2$$

$$P x = c \cdot w \cdot x$$

(ii) 中立軸が腹部内にあるときは

制限条件式

$$\frac{2M_1}{\ell x_1 (t - x_1 / 3)} - \sigma_c a \leq 0$$

$$\frac{M_1}{A_{s1} (t - x_1 / 3)} - \sigma_c a \leq 0$$

$$\frac{M_2}{A_{s2} (d - x_2 + y)} - \sigma_c a \leq 0$$

$$n \frac{M_2 \cdot x}{(d - x_2) \cdot A_{s2} (d - x_2 + y)} - \sigma_c a \leq 0$$

$$A_{s1} \geq 4.22$$

$$A_{s2} \geq 4.22$$

$$t \geq 20$$

$$d \geq 20$$

$$b \geq 30$$

ここで

$$x_1 = \frac{n A_{s1}}{\ell} - 1 - 1 + \frac{2 \ell t}{n A_{s1}}$$

$$x_2 = \frac{n \cdot d \cdot A_{s2} + \frac{\ell (t + t')^2}{2}}{n A_{s2} + \ell (t + t')}$$

$$y = x_2 - \frac{(t + t')}{2} + \frac{(t + t')}{6 (2 x_2 - (t + t'))}$$

$$M_1 = -0.8 \frac{p}{10} \ell^2$$

$$M_2 = -\frac{2 p + p}{6} \ell^2$$

$$P x = c \cdot w \cdot x$$

目的関数はいずれも

$$Z = C A_s \cdot (A_{s1} + A_{s2}) + C c \{ \ell \cdot (t + t') + (d + d' - t - t') b \} C m \cdot 2 \{ \ell + (d + d' - t - t') \}$$

ここで

($C A_s$, $C c$, $C m$ はそれぞれ鉄筋工, コンク) リート工, 型わく工の単価。

となり, Z を最小にする制御変数ベクトル, ℓ , t , A_{s1} , A_{s2} , d , b を求める

入力データは前例と同じく, $C m$ は 1,000 円/㎡ を追加する。

初期値は

$$t = C_1 \sqrt{M_1 / \ell}$$

$$x_2' = S \cdot d = S \cdot C_1 \sqrt{M_2 / \ell}$$

$$S = \frac{n \sigma_c}{n \sigma_c + \sigma_s}$$

(i) $x_2' > (t + t')$ の場合 (T 形ばり)

$$d = Z + \sqrt{Z^2 - w t^2}$$

ここで

$$m = \frac{1}{4} \left(1 + \frac{1}{S} \right)$$

$$w = \frac{1}{3 S}$$

$$Z = \frac{M_2}{2\sigma_c \ell (i + i')} + m(i + i')$$

(II) $x_2' < (i + i')$ の場合 (長方形ばり)

$$d = C_1 \sqrt{M_2 / \ell}$$

$$\ell = 0.3 \times H_1$$

$$b = 30 \text{ cm}$$

とし、これより1級さかのぼって試行を行なった。(As_1 , As_2 は i , d との関数である)

なおこの例では当初 $h = 0.5$ とし、順次 $h = 0.05$, $h = 0.005$, …… と小さい値をとった。

h	L (cm)	T (cm)	D (cm)	B (cm)	As_1 (cm ²)	As_2 (cm ²)	σ_{s1} (%)	σ_{c1} (%)
-1	250.0	25.0	77.5	27.5	9.5	192.7	1519.4	46.3
0	240.0	25.0	85.0	30.0	9.5	158.8	1400.3	42.6
1	232.0	30.0	90.0	35.0	6.6	128.5	1478.8	32.0
2	222.5	37.5	97.5	40.0	5.1	128.5	1338.3	21.5
1.1	230.0	30.0	90.0	35.0	6.3	128.5	1508.6	31.8
1.11	232.5	30.0	90.0	35.0	6.6	128.5	1478.8	32.0

σ_{s2} (%)	σ_{c2} (%)	円/㎡
1247.5	61.9	8990.6164
1307.9	55.6	8441.4700 (注1)
1476.1	52.3	7788.7657 (注2)
1376.0	49.2	8153.9114
1460.9	52.1	7818.1711
1476.1	52.3	7788.7657 (注2)

従来の設計 8,441 円/㎡より 600 円/㎡も安い
7,788 円/㎡に相当する断面が得られる。

注1) 従来の設計

注2) 最適設計

しかし制限条件に、擁壁の安定計算からきまってくる T 形ばりの有効高さ $d \geq 320 \text{ cm}$ を

加え、腹部厚さ 50 cm にとると

h	L (cm)	T (cm)	D (cm)	B (cm)	As_1 (cm ²)	As_2 (cm ²)	σ_{s1} (%)	σ_{c1} (%)
0	360.0	27.5	325.0	50.0	19.1	51.4	1443.0	62.3
1	355.0	37.5	325.0	52.5	12.7	64.2	1408.0	38.1
1.1	360.0	27.5	325.0	52.5	19.1	51.4	1443.0	62.3
1.11	360.0	27.5	325.0	50.0	19.1	51.4	1443.0	62.3

σ_{s2} (%)	σ_{c2} (%)	円/㎡	摘要
1448.6	12.1	8011.9222	従来の設計
1150.6	10.6	8152.6892	
1448.6	12.1	8074.9431	
1448.6	12.1	8011.9222	

従来の設計がそのまま最適設計になった。

以上多くの最適化手法のうち、2つの方法についてのみ試み、いずれも制御変数をしぼり、モデルを極度に単純化した。

ただしここに適用した2つの方法では極値が2つ以上あったり、関数に分水嶺 (ridge) やあん部 (saddle) などのある場合は最適値を誤ることがあるから注意する必要がある。

2・5 自動設計例

以上述べてきた手続きおよび手法の一部を使って完全自動設計になっている1・2の例について次に述べる。

2・5・1 ポストテンション方式によるPC単純T橋

(1) プログラムの概要

自動設計プログラムの入力データは、(2)に示すとおりであり、そのパラメタの指示で自動製図するために必要な数値は磁気テープに書き込まれる。また、同じパラメタの指示で断面諸定数、実応力度、曲げ上げケーブルの座標、各ケーブルの緊張応力度および支間当りの所要材料等はすべての設計数値も印刷される。そして、コンクリートとPC鋼線双方の許容応力度に無駄がないような設計にするための繰り返し計算を含めて1ケース30秒弱(CDC-3600)によるコンパイル時間を除く)で計算は終了する。

ところで、無駄な計算をやらないように断面を仮定するには、従来は過去の経験にもとづいて決定していたが、本自動設計プログラムでは、一連の計算結果からなるべく繰り返し計算を少なくできる断面を仮定するようにした。また、主ケーブルの曲げ上げに関する計算では、曲げ上げ位置を5cm単位に動かして限界核に入るよう操作した。

次に、図-2、図-3は本自動設計によって描いた標準設計の一部である。これを人間が作業した場合には、どんなに熟練された技術者でも設計から図面の仕上りまで少なくとも2週間以上は要するが、自動設計によれば80分位で図面まで仕上り、線の太さ、濃さおよび字型が一定で統制のとれた図面となり、代表的な何枚かを精査しておけば後は何枚書かせようが転記ミスも脱落もない。

(図化作業時間はCALCOMP 770/763を使用した場合複雑なもので一般構造図30分、主げた構造図50分程度であった)

プログラムは、入力データが変わった場合でもその対象となる構造図が同一のプログラムで処理できる汎用性を持っていることが必要であり、現在行なわれている鉄筋コンクリート構造物の設計方式によれば、配力鉄筋や用心鉄筋のように計算で求めることなく設計者の判断によって決められるような要素が非常に多く含まれていることや、図面中のそれぞれの詳細図の縮尺や配置をうま

く考えておかないと、条件が変わった場合には重なる部分が出てくる等プログラミングにとってはやっかいなこと、また、ペン先を上げて動く距離を最少にするよう考慮しないと作図時間が長くなる等の問題点がある。

PC橋の設計図面は、最低一般構造図と主げた構造図が必要であるが、この自動設計プログラムでも改造前のシステムの都合から2分割し、図面相互間の連絡は対照番号で解るようにした。また、斜角のある場合は、右内、左内を-、+で指示することによって、どちらでも描けるようにしてある。

以下、図化に当たって考慮した点についてふれてみる。

(i) 図の大きさと縮尺

図面は、A1版を基準とし次の縮尺とした。

① 一般構造図

側面図	平面図	……支間 ≤ 30 m	1/200
"	"	……支間 > 30 m	1/250
断面図	……		1/100
その他詳細図	……		1/40

② 主げた構造図

側面図	平面図	……支間 ≤ 30 m	1/100
"	"	……支間 > 30 m	1/120
断面図	……		1/60
その他詳細図	……		1/40

なお、鉄筋加工図については、一般構造図、主げた構造図とも見やすい大きさにして縮尺は考慮しなかった。

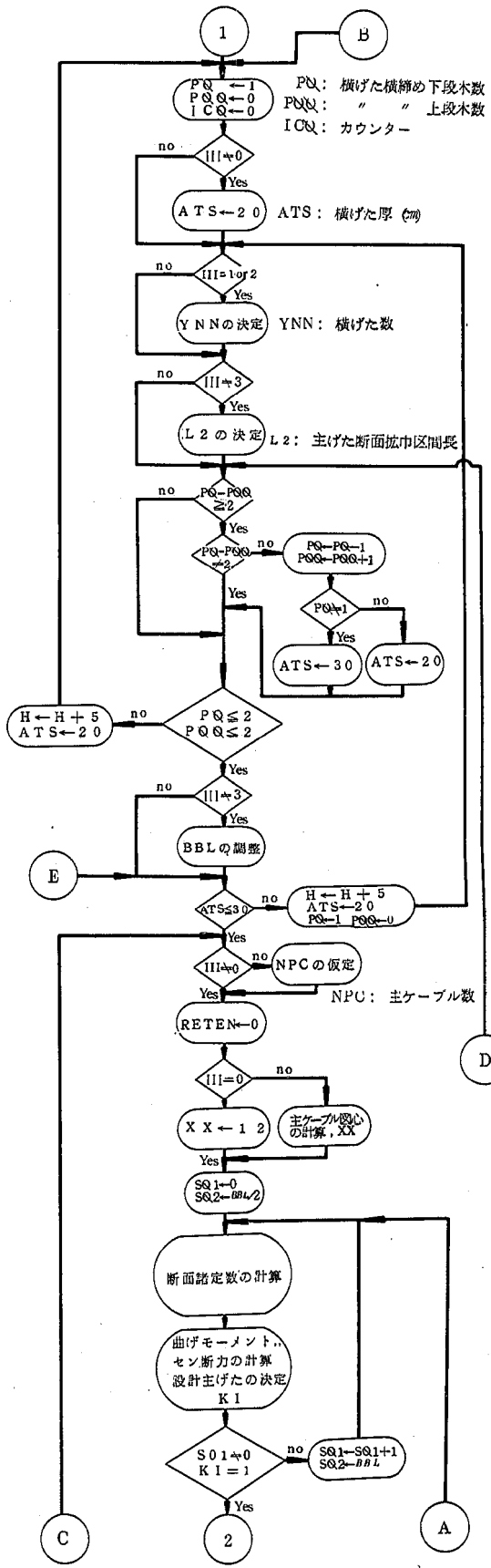
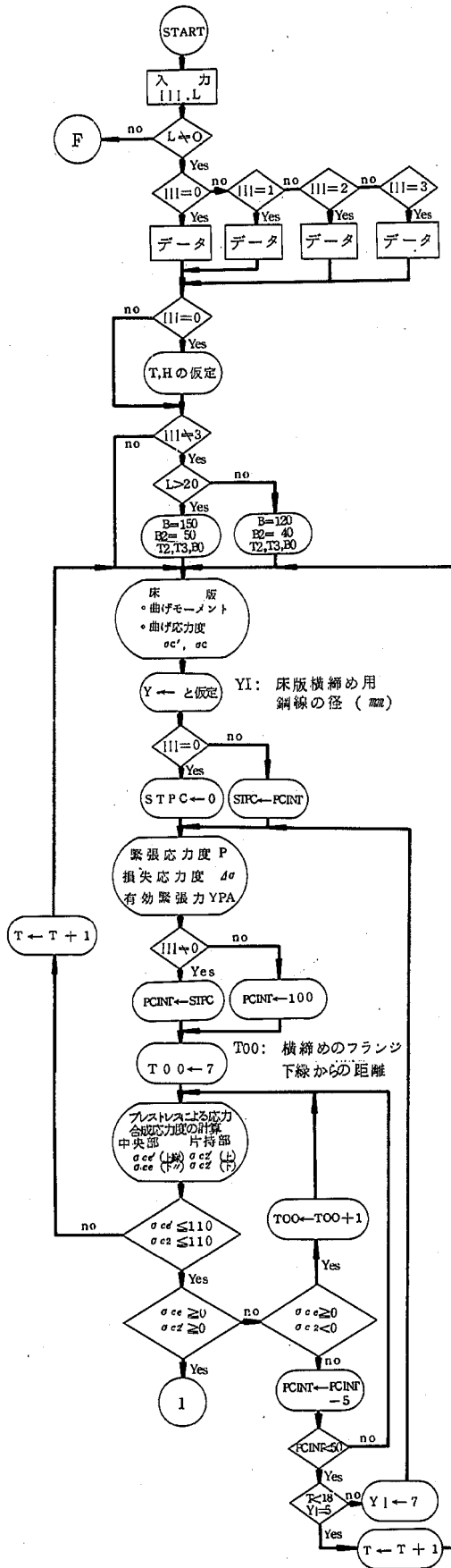
(ii) 線の太さによる区別(改造前の図化機では、ペン先の太さ一定のため下記により区別)

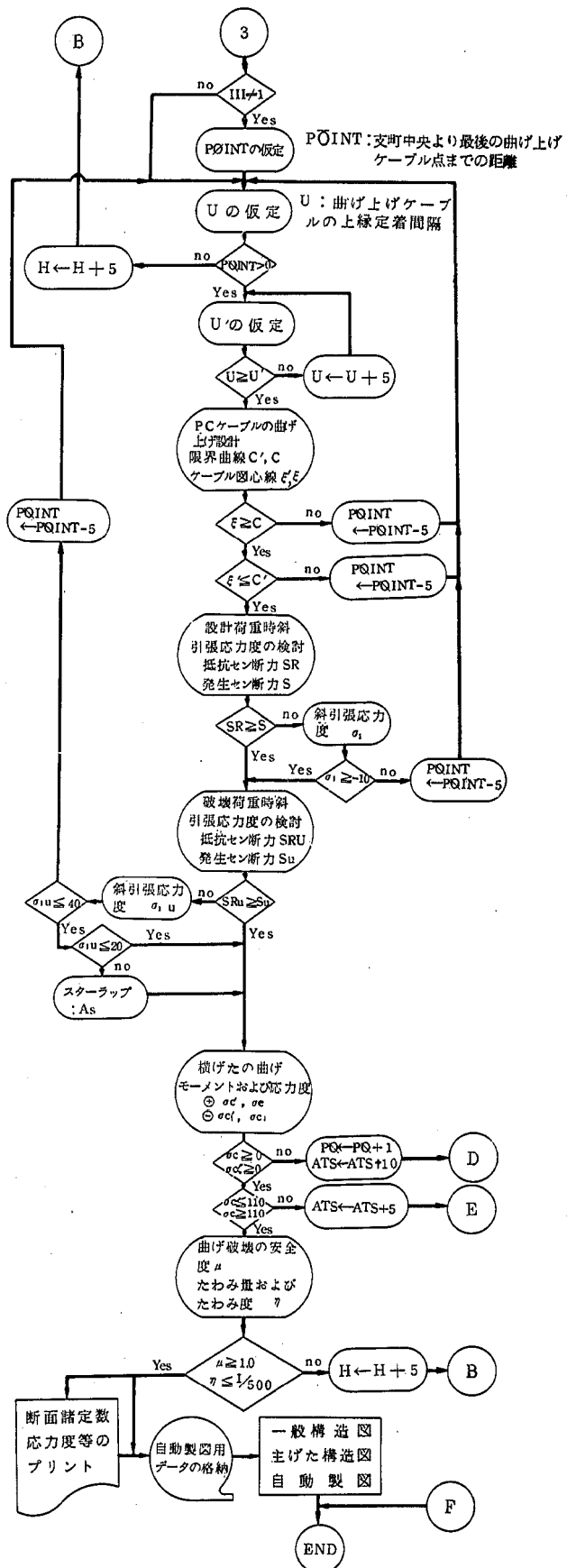
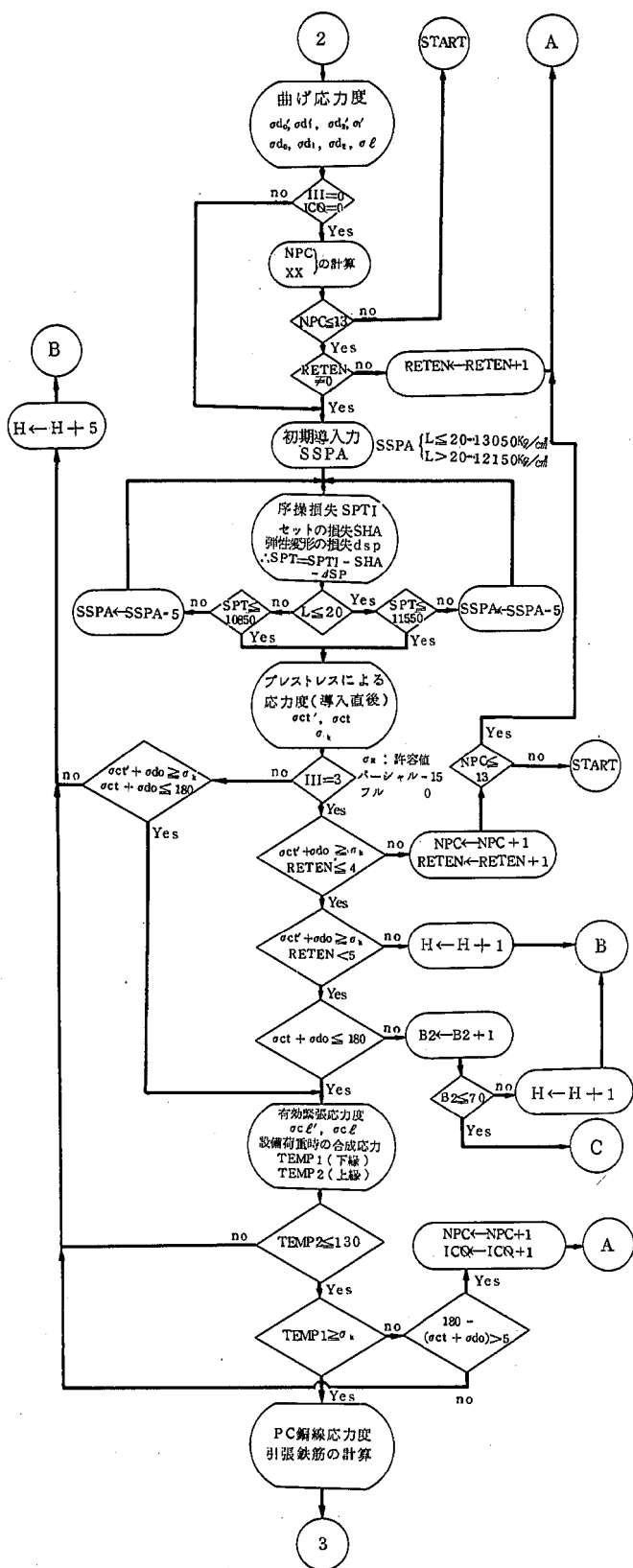
- ① 寸法線……1回書き
- ② 構造線……2回書き(重ねて書く)
- ③ 鉄筋線……1ステップ(0.1mm)づつ3本線
- (iii) 単位図(たとえば一般構造図中では、側面図、平面図)を一つの手続にまとめる。
- (iv) 単位図を描くに必要な標準手続きのうち次のようなものを用意した。

- ① 点線、鎖線、二点鎖線を描くもの
- ② 鉄筋線、寸法線、引き出し線、放物線を描くもの
- ③ 断面を表示するもの
- ④ 図中に必要な漢字、アルファベット

なお図-2・2～図-2・5は自動設計プログラムによって描いたものである。

(3)フローチャート





(4) 代表的な手続き名

自動設計プログラムに組み込んだ代表的な手続きとその流れを述べる。

```

begin
    L, H, B, T.....etc 宣言

    procedure HENSIN (パラメタ部) ;
        下縁から各々のPCケーブルまでの距離を求める

    procedure SUM (パラメタ部) ;
        手続きHENSINで求めた値を合計する

    procedure UP KAKU (パラメタ部) ;
        上縁々界核の座標を求める

    procedure LOW KAKU (パラメタ部) ;
        下縁々界核の座標を求める

    procedure KAKU (パラメタ部) ;
        任意点の上, 下限限界曲線の座標を求める

    procedure NEGIRI (パラメタ部) ;
        曲げ剛性係数 ( $\theta$ ) およびねじり剛性係数 ( $\alpha$ ) を求める

    procedure BUNPAI (パラメタ部) ;
        横げた荷重分配にける影響線縦距を求める

    procedure PCNUMBER (パラメタ部) ;
        PCケーブルの所要本数を求める

    procedure KETAMOMENT (パラメタ部) ;
        支間中央の曲げモーメントから任意点の曲げモーメントを逆算する

    procedure SF (パラメタ部) ;
        任意点の合成せん断力を求める

    procedure TEIOU (パラメタ部) ;
        任意点の抵抗せん断力を求める

    procedure EDCGEAN (パラメタ部) ;
        けた端面に定着するPCケーブルの曲げ上げ角を求める

    procedure YKOBUNPAI (パラメタ部) ;
        横げた荷重分配における影響線縦距を求める

    procedure CABLE (パラメタ部) ;
        各々のケーブルの長さを求める
    
```

自動設計

```

procedure LIST;
    数量計算を行なう

procedure AREA (パラメタ部);
    影響線の面積を求める

procedure COEF (パラメタ部);
    主げたの荷重分配係数を求める

procedure LIVE MOM (パラメタ部);
    活荷重による主げたの曲げモーメントを求める

procedure LIVE SHE (パラメタ部);
    活荷重による主げたのせん断力を求める

procedure MOMENT AND SHERE (パラメタ部);
    死荷重による主げたについての曲げモーメント, せん断力を求める

procedure HAN (パラメタ部);
    斜引張応力張の判定を行なう

procedure PC 1 (パラメタ部);
    一般構造中の側面, 平面, 断面図を描く

procedure PC 2 (パラメタ部);
    一般構造中の鉄筋加工図, 詳細図, 材料表等を描く

procedure PC 3 (パラメタ部);
    主げた構造図中の側面・平面・断面図を描く

procedure PC 4 (パラメタ部);
    主げた構造図中の鉄筋加工図, 詳細図, 材料表等を描く
    } 自動図化

START: READ(L, B, HOD, N, K, ...); ..... (データ読み込み) 支間, 本線幅, 歩道幅員, 主げた数, 橋格等
if L=0 then go to OWARI;
    断面の仮定, 断面諸定数の計算
    各手続の実行

go to START;
OWARI:
end;

```

(5) 手続例

(i) 荷重分配係数計算プログラム

自動設計プログラムに組み込んである分配係数を求めるための手続きは次のようなものである。

<手続名およびパラメタ>

procedure COEFICENT (ALPHA, THETA, N, XX, TW, BW, HOW, CW, I, COEF) :

ALPHA: ねじり剛性係数

THETA: 曲げ剛性係数

N : 主げた数

XX : 分配係数を求めようとするけたの図心から橋軸までの距離 (m)

TW : 抵抗幅 (m)

BW : 車道幅 (側帯も含む) (m)

HOW : 歩道幅 (m)

CW : 地覆幅 (m)

I : 分配係数を求めようとするけたの番号

COEF : 分配係数が格納される (2次元の配列を宣言しなければならない)。

COEF[1,1] = 活荷重による分配係数

COEF[2,1] = 群集荷重による分配係数

COEF[3,1] = 舗装による分配係数

COEF[4,1] = 歩道死荷重による分配係数

COEF[5,1] = 地覆・高欄による分配係数

```
procedure COEFICENT(ARPHA, THITA, N, XX, TW, BW, HOW, CW, I, COEF);
  value ARPHA, THITA, N, XX, TW, BW, HOW, CW, I;
  real ARPHA, THITA, XX, TW, BW, HOW, CW;
  integer N, I;
  array COEF;
  begin
    real AE, MA, A1, A2, A3, A4, A5, X, SAIKAW, XL;
    integer IN, II, J;
    array MY[1:14];
    real procedure AREA(Y1, Y2, L0);
      value Y1, Y2, L0;
      real Y1, Y2, L0;
      AREA = (Y1 + Y2) * L0 / 2;
    procedure DISX(Y1, Y2, L0, X);
      value Y1, Y2, L0;
      real Y1, Y2, L0, X;
      begin
        real L1, Y;
        Y = ABS(Y1) + ABS(Y2);
        L1 = L0 * ABS(Y2) / Y;
        X = L0 - L1;
      end;
  end;
```

```

procedure MENSEKI;
begin integer IX;
  if MY[10]>0 then SAIKAW=2*BW
  else
    begin DISX(MY[IN-1],MY[IN],BW/4,X);
      SAIKAW=BW/4*(IN-3)+X;
    end;
  if MY[10]<0 then A1=A1+MY[IN-1]*X/2; A2=0;
  if SAIKAW>5.5 then
    begin MA=MY[12];
      IX=1;
      if XX=0.0 then
        begin
          if BW>5.5 then begin II=6; go to NEX3; end
          else IX=6;
        end;
      for II=IX step 1 until 10 do
        if MA>MY[II] then go to NEX3;
      NEX3:
        XL=(II-2)*BW/4-5.5;
        IX=II;
        A2=AREA(MY[12],MY[II],XL);
        if IN-IX-1<0 then go to NEX5;
        for J=1 step 1 until IN-IX-1 do
          begin A2=A2+AREA(MY[II],MY[II+1],BW/4);
            II=II+1;
          end;
        NEX5:
          if MY[10]<0 then A2=A2+AREA(MY[IN-1],0.0,X);
        end;
      A3=A1; A1=A1-A2;
      if MY[10]<0 then
        begin if 10-IN>1 then
          begin II=IN;
            for J=1 step 1 until 10-IN do
              begin A3=A3+AREA(MY[II],MY[II+1],BW/4);
                II=II+1;
              end;
            A3=A3+AREA(MY[IN],0.0,(BW/4-X));
          end
          else A3=A3+AREA(MY[10],0.0,(BW/4-X));
        end;
      end;
    for IN=1 step 1 until 14 do MY[IN]=0;
    IN=1;
    for AE=(BW+HOW+CW/2),(BW+HOW),3*BW/4,BW/2,BW/4,
      0.0,-BW/4,-BW/2,-3*BW/4,-(BW+HOW),-(BW+HOW+CW/2) do
      begin
        BUNPAI(ARPHA,THITA,XX,AE,TW,MY[IN]);
        IN=IN+1;
      end;
    MA=if BW>5.5 then (BW-5.5) else -(5.5-BW);
    BUNPAI(ARPHA,THITA,XX,MA,TW,MY[IN]);
    IN=IN+1;
    if HOW<0 then

```

```

begin
  for AE=BW,-BW do
    begin BUNPAI(ARPHA,THITA,XX,AE,TW,MY(IN));
      IN=IN+1;
    end;
  end;
  if HOW<0 then
    begin MA=MY[10]; MY[10]=MY[14]; MY[14]=MA;
    end;
  for IN=6 step 1 until 10 do
    if MY(IN)<0 then go to NEX1;
  NEX1;
    if HOW<0 then go to NEX2;
    II=2; A1=0;
    if MY[10]>0 then IN=11;
    for J=1 step 1 until IN-3 do
      begin
        A1=A1+AREA(MY[II],MY[II+1],BW/4);
        II=II+1;
      end;
    MENSEKI;
    A4=0;
    A5=0;
    go to NEX4;
  NEX2;
    if MY[10]>0 then IN=11;
    A1=AREA(MY[13],MY[3],BW/4);
    II=3;
    for J=1 step 1 until IN-4 do
      begin
        A1=A1+AREA(MY[II],MY[II+1],BW/4);
        II=II+1;
      end;
    MENSEKI;
    if MY[10]>0 and MY[14]<0 then
      begin DISX(MY[10],MY[14],HOW,XL);
        A4=AREA(MY[2],MY[13],HOW)+AREA(MY[10],0,0,XL);
        A5=A4+AREA(MY[14],0,0,(HOW-XL));
      end
    else
      begin
        A5=AREA(MY[2],MY[13],HOW)+AREA(MY[10],MY[14],HOW);
        if MY[10]<0 then
          A4=AREA(MY[2],MY[13],HOW)
        else A4=A5;
      end;
    NEX4;
    COEF[1,II]=(A1+A2/2)/N;
    COEF[2,II]=A4/N;
    COEF[3,II]=A3/N;
    COEF[4,II]=A5/N;
    COEF[5,II]=(MY[1]+MY[11])/N;
  end;

```

(iii) 標準断面図を描くプログラム

次に示すプログラムは、図化プログラム部分のうち主桁の中央断面を書かせる *procedure* の部分を抜き出してまとめたものである。まず、図化に必要な基本手続きの呼び出しを行ない、橋軸直角方向の鉄筋を書く *procedure*、橋軸方向の鉄筋断面を書く *procedure* をして、断面全体を書く *procedure* と続いているが、ここでは最後の *procedure* *SECT2* のみをそのまま記載した。*procedure* *SECT2* はまずわく線を書き、ついで前者を使って鉄筋を書く手続きになっている。つぎに横断面を書く作業を実行させるには、まず、原

点を指示し、必要なデータを読み込み、ついで前に示した手続き *procedure* *SECT2* を呼び出す（実パラメータを代入）という手順になっている。図-1はこのプログラムによって画いたものである。

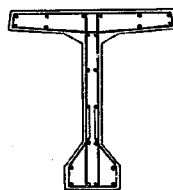


図-2・1

```
begin external PSENX, PMOVE, PREADY, PCUT1; .....基本サブルーチンの呼び出し
```

```
real B,N,T,B0,B2,T2,T3,A;  
integer II,N;  
procedure DSEN(X1,Y1,X2,Y2);
```

鉄筋線を描く手続き

```
procedure RPOINT(X,Y,N);
```

鉄筋断面を書く手続き

```
procedure SECT2(B,H,T,B0,B2,T2,T3,A,II,N); -----
```

```
real B,H,T,B0,B2,T2,T3,A; integer II,N;
```

```
begin
```

```
real G1,G2,G3,G4,G5,G6,G7,G8,G9,C,C1,C2,C3,C4,C5,  
C6,C7,C8,C9,CN1,CN,D1,D2,D3,D4,D5,D6,D,F;
```

```
integer I,K;
```

```
C:=SIN(3.141592*A/180); CN:=(H-T3)/6*N;
```

```
G1:=B/12/C*N; G2:=B2/12/C*N;
```

```
G3:=B0/12/C*N; G4:=T/6*N;
```

```
G5:=H/6*N; G6:=(B-4)/12/C*N;
```

```
G7:=(T+3)/6*N; G8:=(T+12)/6*N;
```

```
G9:=(H-T2-T3)/6*N;
```

```
for I:=1 step 1 until 2 do
```

```
for F:=1,-1 do
```

```
begin
```

```
if F=-1 and II=1 then PSENX(F*G1,0) else PSENX(F*G6,0);
```

```
PSENX(F*G1,G4); PSENX(F*G2,G7); PSENX(F*G3,G8);
```

```
PSENX(F*G3,G9); PSENX(F*G2,CN); PSENX(F*G2,G5);
```

```
PSENX(0,G5); PMOVE(0,0);
```

```
end;
```

```
C1:=(B-10)/12/C*N; C2:=(B0-6)/12/C*N; C3:=(H+4.5)/6*N;
```

```
C4:=(T+3)/6*N; C5:=(B2-6)/12/C*N; D6:=T/6*N;
```

```
C7:=(H+T3)/6*N; C8:=(H+T2+T3)/6+0.5*N;
```

```
C9:=if D=150 then 28/6/C*N else 25/6/C*N;
```

```
D2:=(B0-6)/(2+1.5)/6/C*N; D1:=C1-C9-D2; C6:=(T-3);
```

```
K:=(H-T2-T3+C6)/3; D:=H-T2-T3+C6-K; D3:=(D+5)/2;
```

```
D4:=(H-T2-T3+C6-2*D3)/6*N; D5:=-3/6*N; D:=(D-6)/12/C*N;
```

```
C6:=C6/6*N; D3:=D3/6*N;
```

```
for F:=1,1 do
```

```
begin
```

```
if F=-1 and II=1 then DSEN(0,D5,D*F,D5) else DSEN(0,D5,C1*F,D5);
```

```
DSEN(C1*F,D5,D*F,C4); DSEN(D*F,C4,0,C6); DSEN(C2*F,D5,C2*F,C3+D5);
```

```
DSEN(0,C3+D5,C5*F,C3+D5); DSEN(C5*F,C3+D5,C5*F,C7);
```

```
DSEN(C5*F,C7,(C2-0.5/C)*F,C8); DSEN((C2-0.5/C)*F,C8,(C2-0.5/C)*F,
```

```
C6-D4-D3); RPOINT(D2*F,D5-0.5,N); RPOINT((D1+D2)*F,D5-0.5,N);
```

```
if F=-1 and II=1 then RPOINT((D-0.5/C)*F,D5-0.5,N)
```

```
else RPOINT((C1-0.5/C)*F,D5-0.5,N);
```

```
RPOINT((D-0.5/C)*F,C4+0.5,N); RPOINT((D1+D2)*F,D6+0.5*N,N);
```

```
RPOINT((C2-0.4/C)*F,C6,N); RPOINT((C2-0.4/C)*F,C6-D4,N);
```

```
RPOINT((C2-0.4/C)*F,C6-D4-D3,N); RPOINT((C2-0.4/C)*F,C6-D4-2*D3,N);
```

```
RPOINT((C5-0.4/C)*F,C7,N); RPOINT((C5-0.4/C)*F,C3,N);
```

```
RPOINT((C2-0.4/C)*F,C3,N);
```

```
end;
```

```
PMOVE(0,0);
```

```
end;
```

```
PREADY(0,0); .....原点を指示する
```

```
READ(D,H,T,B0,B2,T2,T3,A); II:=90; N:=2; .....データの読み込み
```

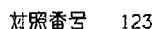
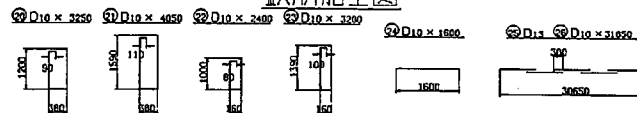
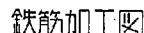
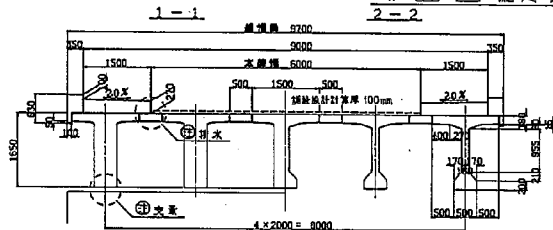
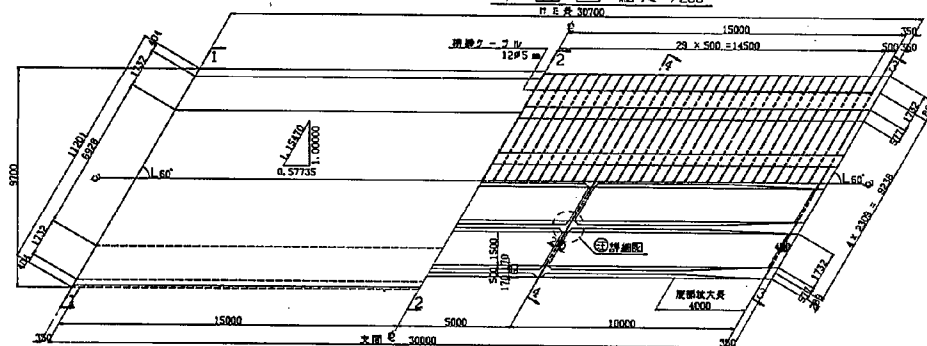
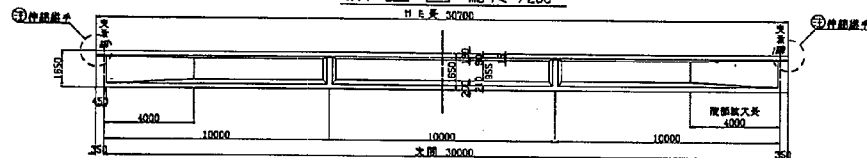
```
SECT2(D,H,T,B0,B2,T2,T3,A,II,N); .....本体呼び出し
```

```
PRINTS('SECT2....PUNCH1.OWARI....'); CRLF(1);
```

```
end;
```

(注) データの説明

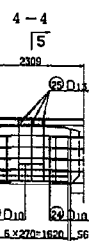
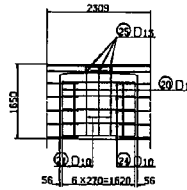
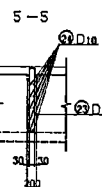
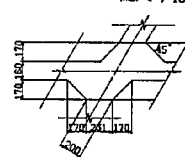
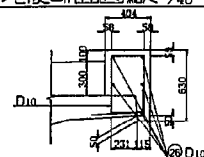
B:上フランジ幅(cm) T3:下フランジ厚(cm) B0:ウエブ幅(cm) N: A1版, A3版の指示
H:指 高(cm) A:斜 角(DEG) D2:下フランジ幅(cm) A1版.....N=2
T:上フランジ厚(cm) II:カウ ン ト 用 T2:下フランジハズレ高さ(cm) A3版.....N=1



04-PCT-1264(S_{30.0}-W_{6.0}-F_{1.50}-A_{60L}-G₁)-S43

(支間) (本線幅) (歩道幅) (斜角) (橋格)

上部工-PC単純Tげた橋(一般図)-その 1264



設計条件

付 長	30700
支 間	30000
本 幅	6000
角 度	左 60°
荷 重	L-20 T-20
型 式	T型ボステンション

反力

	機台 (t)	機脚 (t)
死荷重	235	470
活荷重	85	136
合計	320	606

材料強度

コンクリート	Kg/cm ²	P.C 鋼線 6 mm	Kg/mm ²	
主げは設計基準強度	60a = 400	引 張 強 度	155	
許容引張応力度	部材圧縮時	60a = 130	部材圧縮力	135
	部材引張時	60a = 180	引張力	93
許容引張応力度	部材圧縮時	60a = -15	応力変位係数	108.5
	部材引張時	60a = -15	緊張作業時	121.5
許容引張応力度	設計基準時	61a = 10	P.C 鋼線 5 mm	Kg/mm ²
	換算値時	61a = 20	引 張 強 度	165
許容圧縮力	60a = 110	部材圧縮力	145	
プレレストール主げ	61a = 340	部材圧縮力	99	
圧縮強度	61a = 255	部材圧縮力	115.5	
場所別床版設計基準強度	60k = 300	緊張作業時	130.0	

材料表

主	別	仕	単	1 通りの数量	摘	要
理	げ	た	次	S		
床 版 積 げ た	コンクリート	50k $\times$ 300 kg/m ²	m ²	25.5		
	P C 鋼 筋	12 Φ 5 mm	t	1.296		
	シ ー ス	Φ 55	m	700.7		
	グラウト		m ³	0.674		
	廻 転		m ³	146.7		
	鉄 筋	SD30, D10 D15	t	1.102		
構 造 部	部 材	部 材	m ²	168.9		
	歩 道	歩 道	m	92.1		
	側 溝	側 溝	m	15.4		
	コンクリート		m ³	38.8		
	地 盤	地 盤	m ³	97.3		
	型 材	型 材	t	0.176		
工	伸 縮 継 手					
	天 水					

鉄筋表

記号	径	長さ (mm)	本数	取付重さ (g)	1本入り重量 (g)	重量 (%)	形状	備 考
㊦	D10	3250	32	0.559	1.817	58.1	□	㊦は 他数種と異なり 合致には 含まず
㊦	φ	4050	24	φ	2.264	54.3	□	
㊦	φ	2400	32	φ	1.342	42.9	□	
㊦	φ	3200	24	φ	1.789	42.9	□	
㊦	φ	1600	160	φ	0.894	143.1	—	
㊦	D13	31850	24	0.995	31.691	760.6	〰	
㊦	D10	31850	10	0.559	17.804	178.0	〰	
合 計							1.102 t	

横縄プレストレッシング(フルプレストレス)

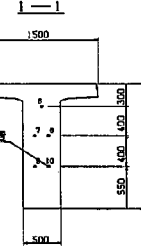
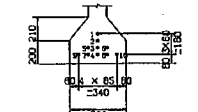
種別	緊張係数 α の算出		伸び(mm)	ケーブル長(mm)	摘 要
	($\tan \theta \times 2$)	(kg/mm^2)			
横ばた	30.7	130.5	47	9700	入=0.004, $\mu=0.3$ 斤引き。余長含まず。 横折4本,床版61本
床版	30.7	130.5	52	10850	
合計ケーブル長				700650	

側面図 縮尺 1/100

04-PCT-123(S 30.0-C 10-A 60L-G)-S43

(支間)(ケーブル数)(斜角)(橋格)

上部エーPC単純Tけた橋-その 123

PC鋼線配置図
縮尺 1/40

材料表

名称	種別	仕様	単位	数量	備考
コンクリート		$f_{ck}=40.0 \text{ N/mm}^2$	m^3	21.7	※増けた上
PC鋼線		12φ7mm 10本	t	0.974	※鉄筋部が
スチール		φ45	t	268.7	より増けた
グラウト			m^3	0.427	にのみ入れ
型枠		上鉄筋部および付いた鉄筋	m^2	13.4	
		増けた部分		3.3	
		その他		126.9	
定置鉄筋		主ケーブル用 12φ7mm	個	20	
		渡り用 12φ5mm		65	
鉄筋		S10 中けた	t	2.012	
		D10, D13 中けた		2.063	

鉄筋表

番号	径	長さ(mm)	本数	単位重量(kg)	体積(kg)	重量(kg)	形状	単位
①	D13	3500	74	0.995	3.482	257.7	U	
②	φ	3700	40	φ	3.682	147.3	U	
③	φ	3700	8	φ	3.682	28.9	U	
④	D10	2050	86	0.559	1.146	98.6	△	
⑤	D13	2950	114	0.995	2.935	334.6	□	
⑥	D10	1000	86	0.559	0.559	48.1	▽	
⑦	D13	31850	24	0.995	31.681	780.6	▽	
⑧	φ	4300	4	φ	4.279	17.1	—	
⑨	D10	900	40	0.559	0.503	20.1	—	
⑩	D13	750	228	0.995	0.746	170.1	—	
⑪	D10	700	104	0.559	0.381	40.7	—	
⑫	φ	1500	104	φ	0.838	87.2	□	
⑬	φ	1300	40	φ	0.727	28.1	—	
⑭	D13	750	458	0.995	0.746	340.3	—	
増けた							2.012 t	
中けた							2.063 t	

プレストレスング(パーシャルプレストレス)

番号	径	長さ(mm)	本数	単位重量(kg)	体積(kg)	重量(kg)	形状	単位
①	10	55.0	119.1	97	17780		スチールとPC鋼線との間の摩擦係数	
②	9	55.3	119.7	111	20420		入=0.004/m	
③	8	55.5	120.2	125	23040		μ=0.3/rad	
④	7	55.8	120.7	139	25680		鉄筋作業時耐力はジャッキの降下損失は見込み	
⑤	6	56.0	121.3	153	28240		ケーブル長は長さを含み	
⑥	1	55.7	120.7	167	30860		す	
⑦	2	54.5	117.9	165	30720		両引き	
⑧	3	54.5	117.9	165	30720			
⑨	4	52.9	114.5	163	30600			
⑩	5	52.9	114.5	163	30600			
合計ケーブル長							266660	

※ けた製作上の注意事項

1. 適用に当たっては、解説を熟読すること。
2. 図中の④のうち、左側孔(径10mm)程度の傾けた施工用孔については解説で示されており、その他については、設計の仕様に従って施工すること。
3. PC鋼線の断面形状は、図面に示すように(支間中央から両端に1/10)において、けた高の1/200とし、その傾きが5mmに満たないものは5mmとする。また両端に1/10以内とする。
4. 打設時のコンクリート表面は、けたより5mm以内とし、また傾きに5mm以内とする。
5. けたの傾きは、図面に示す傾き、たわみ等と記し、必要に応じて図に示すようにしてある。

参照番号 137

04-PCT-1296(S35.0-W8.0-F1.50-A75L-G1)-S43

(支間)(本線幅)(歩道幅)(斜角)(橋格)

上部E-PC単純Tげた橋(一般図)-その1296

設計条件

けり長	35700
支間	35000
本線幅	8000
角 度	左 75°
荷 重	L-20 T-20
型 式	T型ポスティング

反力

	橋台(t)	橋脚(t)
死荷重	333	666
活荷重	106	173
合 計	439	839

材料強度

	Kg/cm ²	P.C鋼線φ7mm	Kg/mm ²
コンクリート			
主げり設計基準強度	$f_{ck} = 400$	引 張 強 度	155
許容主げり応力	$f_{ca} = 130$	許容引張設計荷重	93
許容主げり圧縮力	$f_{ca} = 180$	許容引張 繰返し荷重	108.5
許容主げり引張力	$f_{ca} = 15$	許容引張 繰返し荷重	121.5
設計荷重時	$f_{ca} = 10$	P.C鋼線φ5mm	Kg/mm ²
破壊荷重時	$f_{ca} = 20$	引 張 強 度	165
許容圧縮力	$f_{ca} = 110$	許容圧縮力	14.5
プレストレス入時	$f_{cl} = 340$	許容引張設計荷重	99
圧縮強度	$f_{cl} = 255$	応力変入荷重	115.5
設計引張設計基準強度	$f_{ck} = 300$	設計作業時	130.5

材料表

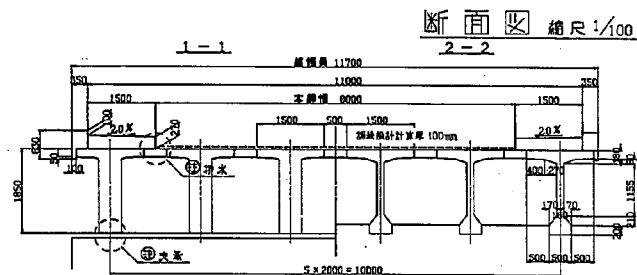
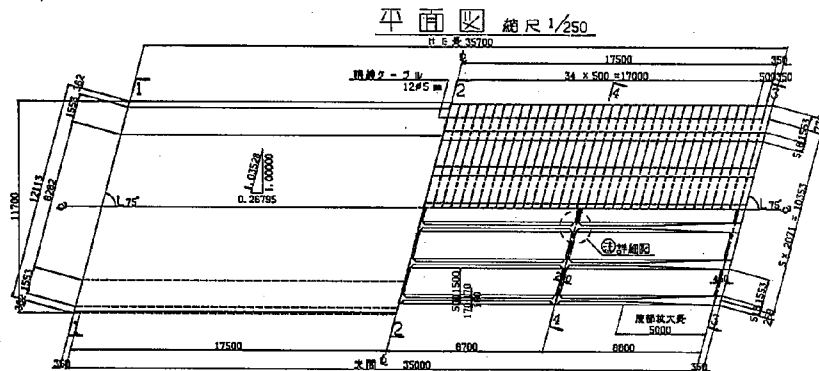
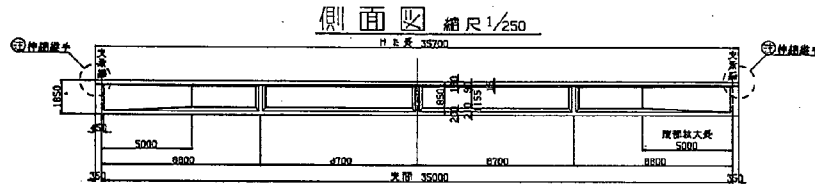
種 別	仕 様	単 位	1通ちり数量	備 考
主 げり			6	
コンクリート	$f_{ck} = 300 \text{ Kg/cm}^2$	m ³	35.2	
P.C鋼線	12φ5mm	t	1.649	
シ ー ス	φ35	m ²	891.7	
グラウト		m ³	0.858	
型 枠		m ²	221.8	
鉄 筋	SD30, D10, D15	t	1.689	
橋 脚				
歩 道		m ²	232.1	
歩 道		m ²	107.1	
側 溝		m ²	53.5	
面 積		m ²	42.8	
地 盤		m ²	112.7	
鉄 筋		t	0.208	
天 承				

鉄筋表

記号	径 (mm)	長さ (mm)	本数	単位重量 (kg/m)	1本重量 (kg)	重量 (kg)	形状	備 考
①	D10	5850	40	0.559	2.152	86.1	□	②は
②	シ	4450	30	シ	2.408	74.6	□	位置鉄筋
③	シ	2850	60	シ	1.593	95.6	□	あり
④	シ	3550	45	シ	1.984	89.3	□	合計には
⑤	シ	1400	300	シ	0.783	234.8	—	含まず
⑥	D13	37150	30	0.995	35.964	1108.9	—	
⑦	D10	37150	10	0.559	20.767	207.7	—	
合 計								1.689 t

横断プレストレスング(フルプレストレス)

種 別	有効断面積 (mm ²)	有効断面 (mm)	ケーブル長 (mm)	備 考
横断	30.7	130.5	52	入=0.004, ユ=0.3
床版	30.7	130.5	57	片引3, 全長含まず
合計ケーブル長				891650
				橋脚 5本, 床版 71本

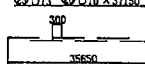
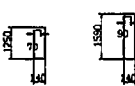
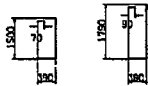


鉄筋加工図

① D10 x 3850 ② D10 x 4450 ③ D10 x 2850 ④ D10 x 3550

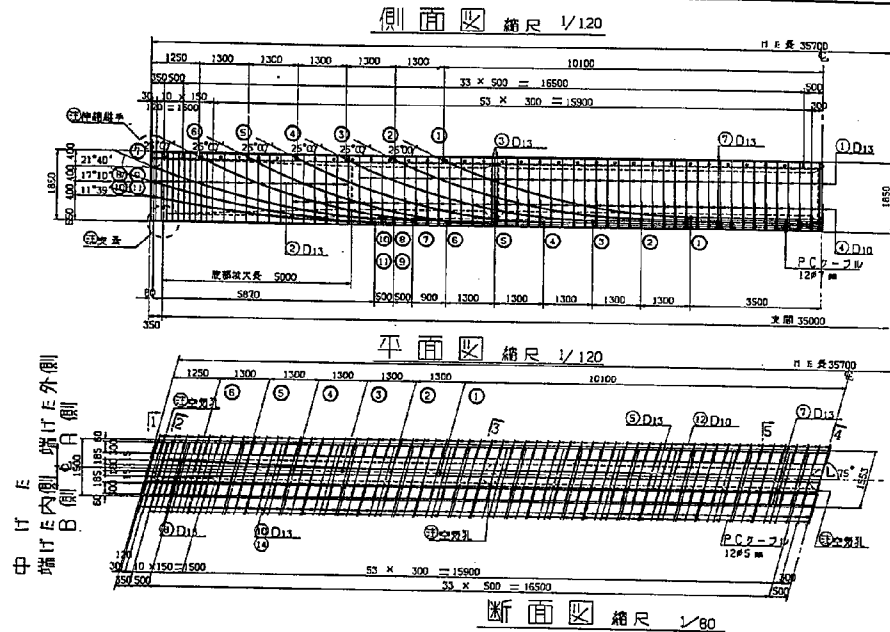
⑤ D10 x 1400

⑥ D13 ⑦ D10 x 37150

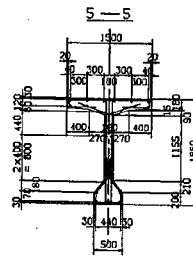
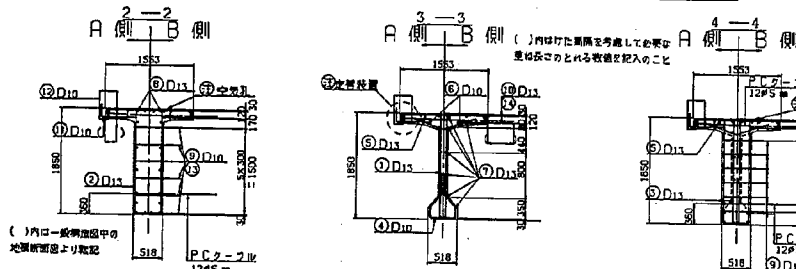
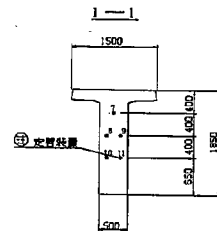
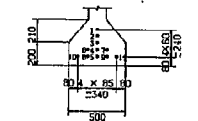


施工上の注意事項

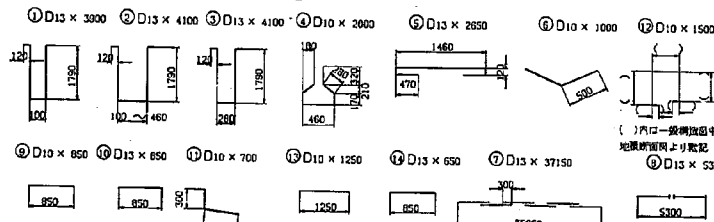
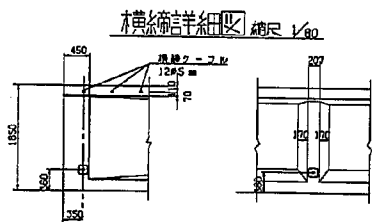
- 適用に当たっては設計と照合すること。
- 天承、特設橋脚、歩道設置、それ以外の詳細設計が既成のジャッキアップに必要な荷重については事前に充分検討すること。
- 橋脚ごう配が2%以下の場合にはこれを横けて設置してよい。また2%を超えた場合には上管線と横断ごう配に合わせて設置する。



PC 鋼線配置図
縮尺 1/40



鉄筋加工要



04-PCT-137 (S_{35.0} - C₁₁ - A_{75L} - G₁) - S43

(支間) (ケーブル数) (斜角) (橋格)

上部工・PC単純Tげた橋—その 137

材料表

名称	規格	仕 様	単位	数 量	備 考
主 骨	コ ン กรี	6kg = 4.00 Kg / m ²	m ²	26.9	☆増付した
	P C 鋼 筋	12φ 7mm	1 本	1.239	☆従前部比
	シ ー ス	φ 45	m	341.6	☆増付した
	ク ラ ウ ト		m ²	0.543	☆にのみ入る
	型 枠	上鉄筋設置および下鉄筋 との間に取付可能	m ²	15.2	
た		その他	-	5.6	
			-	161.9	
	足 場 具	スチール制網 12φ 7mm 縦 横 材 用 φ 12 × 5 mm	個	22	
鉄 筋			-	76	
		S030 10φ、013	個 び た	2.379	
			-	2.465	

鉄筋表

[illegible]

プレストレッシング (バーチャルプレストレス)

[illegible]

※ けた製作上の注意事項

1. 適用に当たっては、解題を念慮のこととする。
2. 図中の①と②は、塩粒(径10mm程度の球形の塩工団の孔)については解料でふれてあり、その他については解料でふれない。
3. PC材の平均密度は塩粒は、塩粒より2.5%分厚きとする。
4. PC材の平均密度は塩粒は、塩粒より2.5%分厚きとする。また塩粒については10/200とし、その径が5mmに満たないものは5mmとする。また塩粒については5mm以内とする。
5. 打撃のコンクリート部300mm厚若しくはよりするた組にしないがけはせぬ。
6. 打撃の鉄筋は塩粒は、けたより2.5mm以内とする。
7. 解料は、解料は塩粒、灰分、たまりを記し、実量にのびるようになっている。

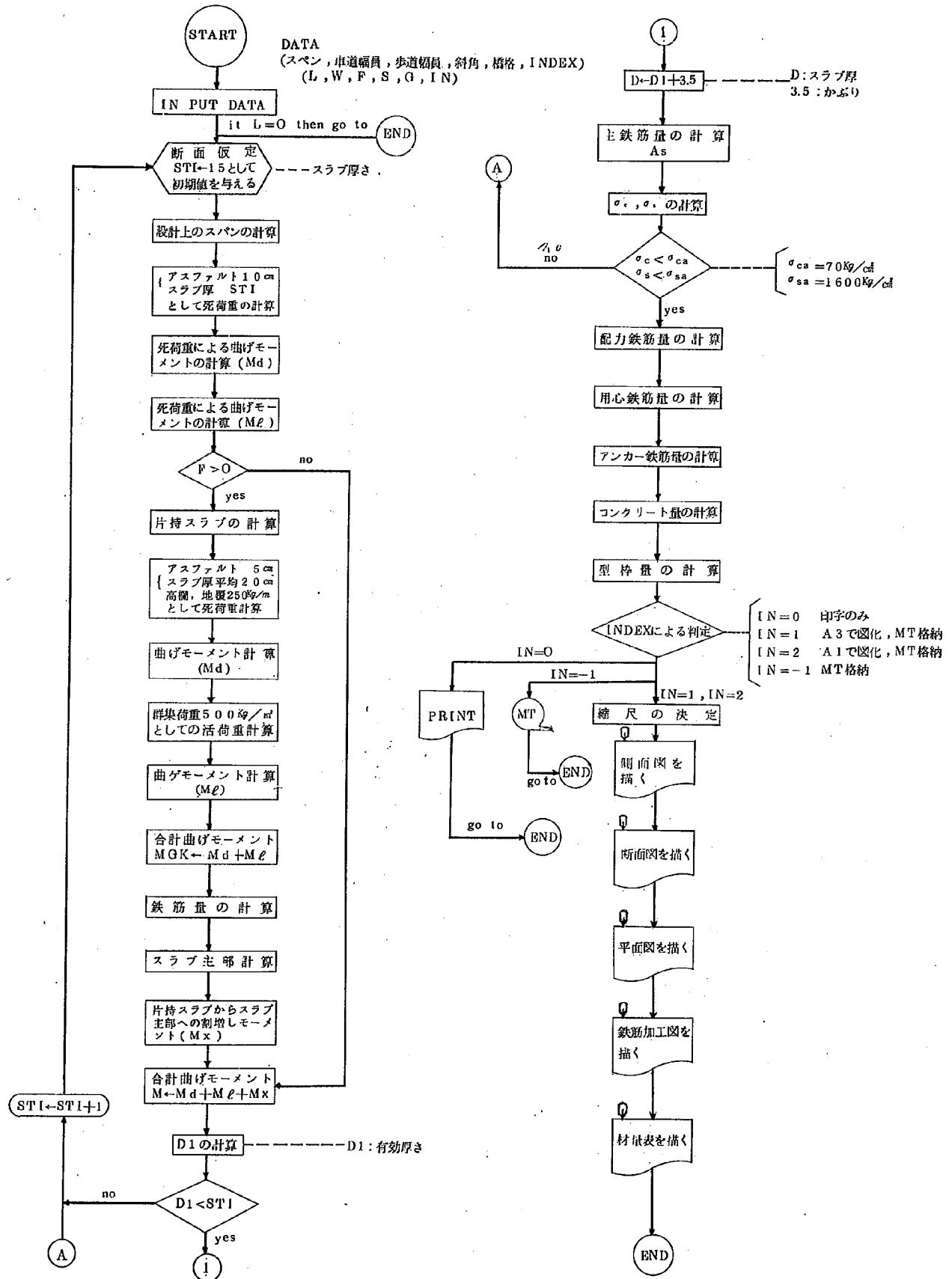
PC橋と同じく完全自動設計プログラムの完成している。構造物の1つであるRC単純スラブ橋について次に

(1) 代表的な手続名と入力データ

自動設計用

自動図化用

(2)フローチャート

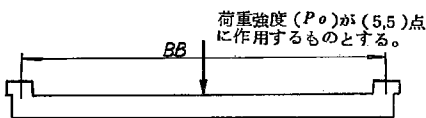


(3) 手続例

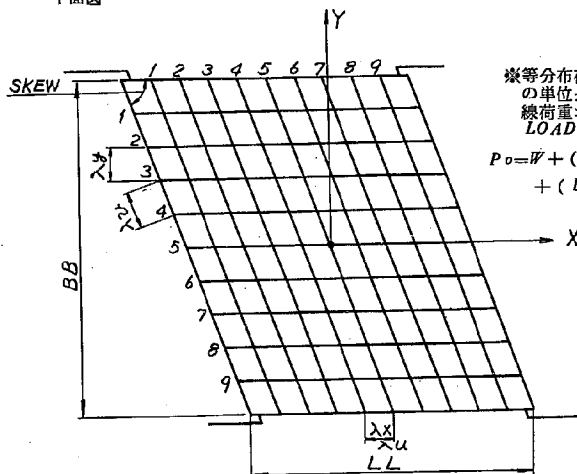
(i) 直交等方性版の階差式による曲げモーメントの計算手続，計算部分より直交等方性版としての階差式による曲げモーメントを計算する手続の1つを紹介する。

直交等方性版の階差式による曲げモーメントを求める手続は次のようである。

断面図（幅員方向）



平面図



※等分布荷重 $\bar{W}$, X 方向の単位長さあたり P の線荷重および集中荷重 $LOAD$ よりなるものとする。

$$P_0 = \bar{W} + (P / \lambda_y) + (LOAD / \lambda_x \cdot \lambda_y)$$

<手続名およびパラメタ>

procedure KAI SAI ($PO, E, H, LL, BB, SKEW, XN, YN, LOAD, P, \bar{W}, M$)

PO : ボアソン比

E : 弾性係数比

H : 仮定したスラブ厚 (cm)

LL : スパン X 方向 (m)

BB : 幅員 Y 方向 (m)

$SKEW$: 斜 角 (度)

XN : X 方向分割数

YN : Y 方向分割数

$LOAD$: 集中荷重 (中央に作用) (kg)

P : 線荷重 (中央に作用) (kg)

$\bar{W}$: 等分布荷重 (kg/m^2)

M : 各点曲げモーメント (二次元 array) ($t \cdot m$)

(ii) 側面図を描く手続

図化プログラムの中からは先に 2・3・7 で、

側面図を描く手続を紹介したのでここでは省略する。

```

procedure KAISA1(P0,E,H,LL,BB,SKEW,XN,YN,LOAD,P,W,M);
real P0,E,H,LL,BB,SKEW,XN,YN,LOAD,P,W;
array M;
begin
  real BX,BY,KB,RY,RX,RU,RV,P2,A,B,ALF,MX,MY,
    A1,BBX,G1,P1,PP,PP1,MM,DETERM;
  integer I,I1,J,J1,IX1,J1,J2,I1,I2,K,MN,N,ID;
  array AR,BR[1:50,0:50],KON,X[1:100],
    XX,YY[1:81,1:81],AR1[1:171];
  procedure MATINV(A,N,B,M,DETERM,ID);
  real DETERM; integer N,M,ID; array A,B;
  begin integer I,J,K,L,IROW,JROW,ICOLUM,JCOLUM;
    array INDEX[1:N,1:3]; real PIVOT,AMAX;
    DETERM:=1;
    for I:=1 step 1 until N do
      INDEX[I,3]:=0;
      for I1:=1 step 1 until N do
        begin
          AMAX:=0;
          for J:=I1 step 1 until N do
            if INDEX[J,3]≠1 then
              for K:=1 step 1 until N do
                begin
                  if INDEX[K,3]>1 then
                    begin ID:=2; go to END; end;
                  if AMAX<ABS(A[J,K])∧INDEX[K,3]<1 then
                    begin
                      IROW:=J; ICOLUM:=K; AMAX:=ABS(A[J,K]);
                    end;
                end;
              INDEX[ICOLUM,3]:=INDEX[ICOLUM,3]+1;
              INDEX[I,1]:=IROW; INDEX[I,2]:=ICOLUM;
              if IROW=ICOLUM then go to L1;
              DETERM:=-DETERM;
              for L:=1 step 1 until N do
                begin real SWAP;
                  SWAP:=A[IROW,L]; A[IROW,L]:=A[ICOLUM,L];
                  A[ICOLUM,L]:=SWAP;
                end;
              for L:=1 step 1 until M do
                begin real SWAP;
                  SWAP:=B[IROW,L]; B[IROW,L]:=B[ICOLUM,L];
                  B[ICOLUM,L]:=SWAP;
                end;
              L1:
              PIVOT:=A[ICOLUM,ICOLUM];
              DETERM:=DETERM*PIVOT;
              A[ICOLUM,ICOLUM]:=1;
              for J:=1 step 1 until N do
                A[ICOLUM,J]:=A[ICOLUM,J]/PIVOT;
              for L:=1 step 1 until M do
                B[ICOLUM,L]:=B[ICOLUM,L]/PIVOT;
              for J:=1 step 1 until N do
                if J≠ICOLUM then
                  begin real T;
                    T:=A[J,ICOLUM];
                    A[J,ICOLUM]:=0;
                    for L:=1 step 1 until N do
                      A[J,L]:=A[J,L]-A[ICOLUM,L]*T;
                    for L:=1 step 1 until M do

```

```

      B[J,L]:=B[J,L]-B[ICOLUM,L]*T;
    end;
  end;
  for I:=1 step 1 until N do
  begin
    J:=N+1-I;
    if INDEX[J,1]≠INDEX[J,2] then
    begin
      JROW:=INDEX[J,1];
      JCOLUM:=INDEX[J,2];
      for K:=1 step 1 until N do
      begin
        real SWAP;
        SWAP:=A[K,JROW];
        A[K,JROW]:=A[K,JCOLUM];
        A[K,JCOLUM]:=SWAP;
      end;
    end;
    for K:=1 step 1 until N do
    if INDEX[K,3]≠1 then
    begin
      ID:=2; go to END; end;
    end;
  end;
END;
end;
real procedure SIND(S);
value S; real S;
begin
  real PAI;
  PAI:=3.14159265358; SIND:=SIN(S*PAI/180);
end;
real procedure COSND(S);
value S; real S;
begin
  real PAI;
  PAI:=3.14159265358; COSND:=COS(S*PAI/180);
end;
real procedure TAND(S);
value S; real S;
begin
  real PAI;
  PAI:=3.14159265358;
  TAND:=SIN(PAI*S/180)/COS(PAI*S/180);
end;
RY:=BB/YN; RV:=RY/SIND(SKEW);
RX:=LL/XN; RU:=RX;
BX:=(E*H13)/12*(1-P012)/XN*RX;
BY:=(E*H13)/12*(1-P012)/YN*RV;
ALF:=SQRT(BY/BX);
KK:=RY/RX;
A:=KK12*(1+ALF*(TAND(90-SKEW)12));
B:=ALF*KK*TAND(90-SKEW);
A1:=1-(BB/100-5.5)/50;
Q1:=P*A1; P1:=W*A1;
PP=P1+(Q1/RY)+(LOAD/(RX*RY));
PP1=P1+(Q1/RY)+(KK*LOAD/RY12);
BBX:=(PP*RY14)/BX;
MM:=1.8*((LL/100)*SIND(SKEW))+0.5;
X[13]:=6*ALF12+8*ALF*A+6*A12+B12;
X[14]:=-4*(ALF*A+A12);
X[15]:=A12-B12/2; X[12]:=X[14];
X[11]:=X[15]; X[8]:=-4*(ALF12+ALF*A);
X[9]:=2*(ALF*A-ALF*B-A*B);
X[10]:=A*B; X[7]:=2*(ALF*A+ALF*B+A*B);

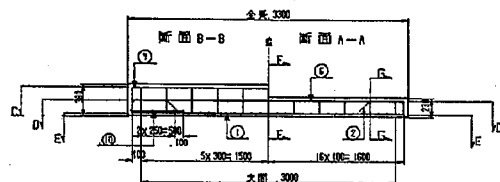
```

```

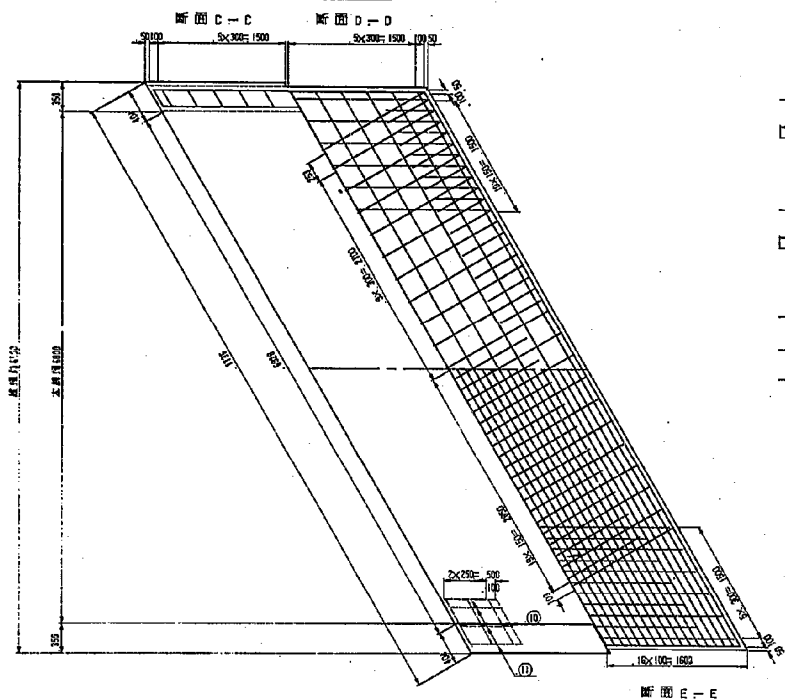
X[6]:=A*B; X[3]:=ALF*2-B*2/2;
X[4]:=ALF*B; X[5]:=B*2/4;
X[2]:=ALF*B; X[1]:=X[5];
X[18]:=X[8]; X[19]:=X[7];
X[20]:=A*B; X[17]:=X[9];
X[16]:=X[10]; X[23]:=X[3];
X[24]:=X[2]; X[25]:=X[1];
X[22]:=X[4]; X[21]:=X[5];
IX1=1;
AR1:=vec 0,0,0,0,0,0,1,6,11,16,21,0 tor;
for I1=1 step 1 until 9 do
  begin integer IX2; IX2=1;
    for I2=1 step 1 until 9 do
      begin integer K1;
        K1=82-((I1-1)*9+I2);
        for J1=1 step 1 until 9 do
          begin array AR2[1:81];
            real C1,C2,C3,C4,C5,D;
            D=AR1[IX1+J1-1];
            if D#0 then
              begin
                C1=X[D]; C2=X[D+1]; C3=X[D+2];
                C4=X[D+3]; C5=X[D+4];
              end else
                C1:=C2:=C3:=C4:=C5:=0;
            AR2:=vec 0,0,0,0,0,0,C1,C2,C3,C4,C5,0 tor;
            for J2=1 step 1 until 9 do
              begin integer K2;
                K2=(J1-1)*9+J2;
                XX[K1,K2]=AR2[IX2+J2-1];
              end;
            end;
            IX2=IX2+1;
          end;
        IX1=IX1+1;
      end;
    N:=XN*YN; MN=1;
    for I=1 step 1 until N do
      for J=1 step 1 until MN do YY[I,J]=BBX;
      MATINV(XX,N,YY,MN,DETERM,ID);
      if ID=2 then go to CHECK;
    for I=-10 step 1 until 1 do XX[I]=0;
    for I=N step 1 until N+10 do XX[I]=0;
    for I=1 step 1 until N do XX[I]=YY[I,1];
    for I=1 step 1 until N do
      begin
        MX:=-BX*((XX[I if I-I+9*9=1 v I=1 then 0 else
          I-1]-2*XX[I]+XX[I if I-I+9*9=0 then 0 else
          I+1]))/(RU*2)+PO*(XX[I-9]-
          2*XX[I]+XX[I+9])/((RV*2));
        MX1=MX/(RU/100)/100000;
        MX2=ABS(MX1);
        if I#9 then begin II=1; JJ=I; end;
        if I>9*II then II=II+1;
        if I>9 then JJ=I-II*9;
        M[II,JJ]=MX2;
      end;
    CHECK:
      CRLF(2); SPACE(20);
      PRINTS('THE MATRIX IS SINGULAR'); CRLF(2);
end;

```

側面圖 縮尺 1/50



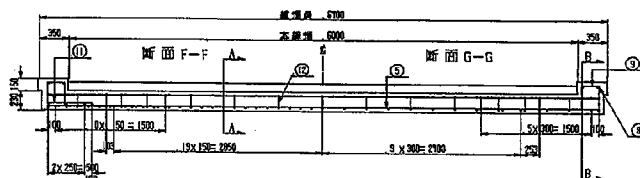
平面圖 縮尺 1/50



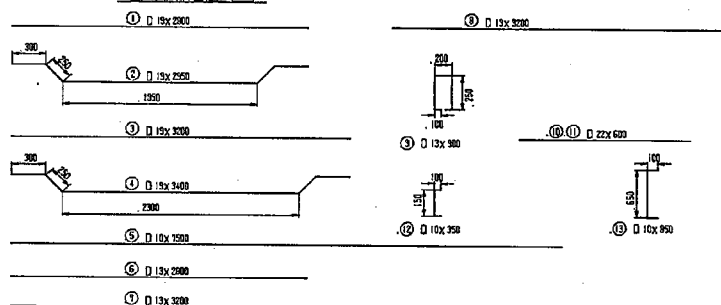
04-RCS-256 (S. 3.00-W. 6.00-A. 03-G. 2)-S. 44

256

斷面圖 縮尺 1/50



鉄筋加工図



鉄筋表

鉄筋番号	規格	長さ	本数	重量	合計重量
① D 19	2800	21	2.249	6.272	131.71
② D 19	2950	20	2.240	6.608	132.16
③ D 19	3200	14	2.240	7.168	100.35
④ D 19	3400	10	2.240	7.616	76.16
⑤ D 10	7500	45	0.559	4.192	192.65
⑥ D 13	2800	21	0.994	2.783	59.45
⑦ D 13	3200	14	0.994	3.181	44.53
⑧ D 10	3200	4	0.559	1.989	7.16
⑨ D 10	900	26	0.559	503	13.08
⑩ D 22	600	9	3.050	1.830	14.64
⑪ D 22	600	9	3.050	1.830	14.64
⑫ D 10	350	195	0.559	195	38.15
⑬ D 10	850	7	0.559	4.95	3.33

設計条件

屋根勾配	3.00
屋根厚	6.00
屋根材	9.1
屋根下地	2
屋根上地	0.1
屋根下地	0.2
屋根上地	1600
屋根下地	300
屋根上地	70

3. 標準設計と自動設計

自動設計プログラムは一組の設計条件に対して設計、施工管理等を総合した場合に対する最適化を考えればよいが、これを標準設計とする場合にさらに不特定な環境条件のもとで施工される事なども考慮しなければならない。

以下ポストテンション方式PC単純T桁橋およびよう壁を例に自動設計し標準設計の相違点の2、3例についてふれる。

(1) 桁高について

標準設計は、計画当初同一支間、同一主桁を目標にし

ていたのがあったが、23種類の幅員について計算してみたところ一等橋のみでも1本の主桁に統一することは不経済になり、また、支間方向からもながめてみることも必要であった。

表3.1は支間25mにおける主桁を決定するための資料であり、横に並んだ計算値が夫々自動設計の結果で桁高のみについて言えば105cmから140cmまでであるものを標準設計では1等橋は130cmと135cmの2種類、2等橋は120cmの1種類にまとめた。

表3.1 主桁決定資料(支間25m)

総幅員-車道-歩道	主桁数	桁高		主ケーブル数		中間横桁厚		床版厚		横筋間隔		下縁応力度
		計算値	採用値	計算値	採用値	計算値	採用値	計算値	採用値	計算値	採用値	
13.0-9.0-2.0	⑦	125		8		25		18		55		-12.3%
12.5-7.5-2.5	⑦	125		8		25		18		55		-9.8
11.5-7.5-2.0	⑥	130		8		20		18		50		-9.5
11.0-8.0-1.5	⑥	130		8		25		18		50		1.6
11.0-7.0-2.0	⑥	130		8		25		18		50		0.9
10.5-7.5-1.5	⑥	120	130	8	8	20	25	18	18	55	50	-14.2
10.5-7.5-1.5	5	135		8		20		19		50		-14.2
10.5-6.5-2.0	⑧	130		8		25		18		50		6.2
10.0-7.0-1.5	④	130		8		25		18		50		12.0
10.0-6.0-2.0	⑥	130		8		25		18		50		11.5
9.5-6.5-1.5	⑤	130		8		25		18		50		-6.2
9.5-8.0-0.75	⑤	135		8		20		18		50		-0.9
9.0-6.0-1.5	⑤	135		8		20		18		50		7.6
9.0-7.5-0.75	⑤	135		8		20		18		50		5.6
8.0-8.0-0	⑤	135		8		20		18		50		16.2
8.0-6.5-0.75	⑤	120	135	8	8	20	20	17	18	50	50	-13.3
8.0-6.5-0.75	4	140		8		20		19		55		-12.6
7.5-6.0-0.75	④	135		8		20		18		50		-10.0
7.0-7.0-0	④	130		8		20		18		55		-11.2
6.5-6.5-0	④	125		8		20		17		50		-8.8
(2等橋)												
8.0-6.5-0.75	⑤	110		8		20		17		50		-7.2
8.0-6.5-0.75	4	125		9		20		19		55		-0.5
7.5-6.0-0.75	④	120	120	8	8	20	20	18	18	50	50	-14.0
7.0-7.0-0	④	110		8		20		18		55		-13.3
6.5-6.5-0	④	105		8		20		17		50		-12.8

(2) 上フランジ幅と下フランジ幅について

標準設計では、上フランジおよび下フランジの幅は、支間20m以下の場合120cmと40cm、21m以上の場合は150cmと50cmに統一してある。これらも標準設計なるがゆえに統一した値であるが、上フランジ幅を限定したことによってある幅員では場所打ち床版幅がうまくとれない場合があり、主桁本数を一本増加しなければならなくなったことや許容値ぎりぎりの設計となってい

る場合がある。

また、下フランジ幅を限定したことはプレストレス導入時において桁高の増加につながった感もあり、もう少し広くすれば現在の標準設計は多少桁高が低くなるのではないかと思われるが、余裕があることは適用範囲が広くなるので一概にはいえない。

次に、自動設計プログラムで桁高を制限して設計を行なう場合は、下フランジ幅を1cmづつ調整し桁高へのフ

ードバックをしないようにしてあり、また、主桁の断面形状で妥当な数値を決め得るようなものについては入力データとして与え、より最適な設計となるよう考慮した。

(3) 扶壁式鉄筋コンクリート擁壁の扶壁中心間隔
自動設計の扶壁中心間隔は擁壁の高さと土質で決定されるが、標準設計では 3.60 m に統一した。

その理由は ①市販の鋼製型わくの長さが 2.70 m で前壁背面の長さをこれに合せたこと。②擁壁は高さや土質が一定であることがめずらしいため、現場のはん雑をさけるため等である。

4. おわりに

今でこそわれわれは人間を主役、機械を脇役とする工業化社会から人間機械系という新しい生物を主役とする情報化社会に移行しつつあること、そしてどうも日本がその先頭を切るらしいことは常識化しているが、昭和 39 年頃にその準備としてこの人間機械系の通信用語のうち当時一番人間の ALGOL を各人が習得して、問題をもつ者が直接計算機と対話する方針を打出し ALGOL 研修(当初 5 日後不用かつ有害な部分を切りすて 2 日)を開始したときは「そんなことをしなくても今のままでよい」という消極派から「計算機は敵だ」という積極派に至るまでいろいろな抵抗に行き当たった。それは 200 年前のイギリスに起った第一次産業革命のときの、農業化社会から工業化社会への移行に当たってみられた「機械打

ちこわし」の様に似たものであった。しかもそれも実際に 1 日目の午後に自分自身の問題のプログラムをつくらせて計算機と会話し、2 日目の午後には *procedure* を含む卒業論文を通じて他のコンパイラ言語との比較などを知って翌日から日常の業務のプログラム化、そうして *external procedure* の集積によるある意味での計算機の教育に参加する人が増加するにつれて、これらの抵抗も逐次減少し、本年 2 月現在で 4500 人を越えるプログラムの集団を形成することができた。

これらの人達の手によって今後各自の業務を標準化することが始められることになる。

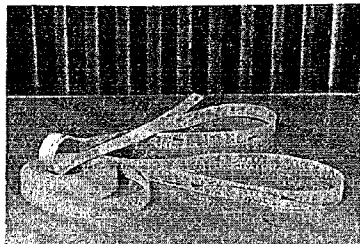
今まで土木研究所において作業を行なった我々が標準化作業によって得た最大の成果は、決してそのプログラムや設計図ではない。この過程で得た最大の成果は真の最適設計とは何であるかを考えさせられ、今までにげなしにやり過ぎていた誤りや問題点を発見し、既存の知識によって解決し得たものおよび机上実験や模型実験によって解決し得たものは組み込み、実施の際の埋設計器その他によって将来解決し得るものおよび問題点としての認識はなし得たのが当分の間解決の見込のないものをたしかめることの出来たことである。

ここにかけたものは自動設計と称するにはまだ十分なものであるが、これを足がかりにして各地建においても実施例を重ね、その間に得た共通の方法論の討論結果により、近い将来にこの改訂がなされることを期待して止まない次第である。

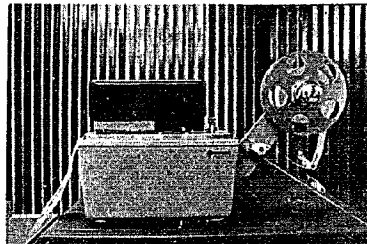
参 考 文 献

- 1) 建設省制定土木構造物標準設計 1, 2, 6, 13, 14, 15, 16, 17 全日本建設技術協会
- 2) 土木研究所 編: 建設省制定土木構造物標準設計 (1), (2) および (13)~(17) の手引き 全日本建設技術協会
- 3) 中村・富田: 擁壁の自動設計方式に関する研究 土木研究所報第 138 号 (1970)
- 4) 中村・岩松・工藤: PC 単純 T 桁橋の自動設計方式に関する研究 土木研究所報第 139 号 (1970)
- 5) 中村・岩松・工藤: 標準化資料第 4 号 土木研究所資料 480 号 (1969)
- 6) 沖野数郎: 自動設計 (8), (9) 機械研究第 21 巻第 12 号, 第 22 巻第 1 号 (1969~1970)
- 7) 丸安・中村・村井・若林: 土木設計における最適化 東大生産技術研究所報告 第 19 巻第 4 号 (1969)
- 8) 中村慶一: 山登りの方法による最適条件追跡プログラム 土木技術資料 第 5 巻第 8 号 (1963)
- 9) 中村慶一: アルゴリズムプログラミング入門 森北出版 (1968)
- 10) 小岩・伊藤: 新しいプログラミング言語 PL/I 日刊工業新聞 (1969)
- 11) 中村慶一: 電子計算による自動設計 土木施工 9 巻 2 号 (1968)
- 12) 中村・岩松・工藤: PC 橋の標準化と自動設計 プレストレストコンクリート 11 巻 2 号 (1969)
- 13) 中村慶一: 土木技術と電子計算組織 土木施工 11 巻 1 号 (1970)
- 14) 中村慶一: 土木構造物の標準化と自動設計 カラム 34 (1970)

自動設計計算から自動図化までの流れ



アルゴル入力テープ



光電式紙テープ読込装置 (PTR)

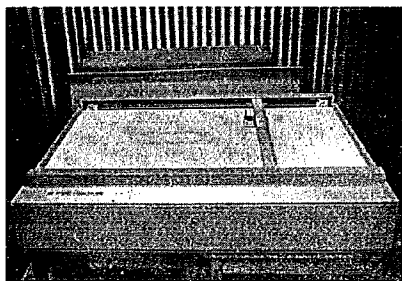
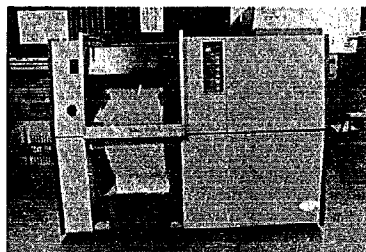
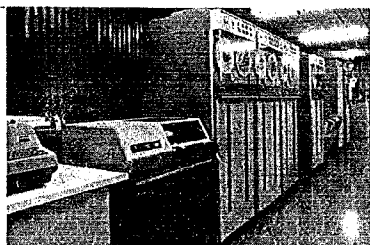
中央演算処理装置
TOSBAC-3400-40



磁気ディスクおよび
磁気テープ (M.T)
図化に必要な計算結果を格
納するもので、短期処運用
にはディスク、永久保存用
には、M.Tを使用する。

印刷装置 (LP)

プログラムのチェックや
計算結果を印刷するた
めに使用する。



図化装置

〔ディスク、M.T、HTP等からの
情報により図化を実行する。〕



高速テープ
穿孔装置 (HTP)

〔デモンストレーション等の
ため計算結果の紙テープを
作る。〕