文　献　紹　介

A: 数値解析　B: プログラミング　C: 計算機方式
D: 回路および機器　E: オートマトン　F: 応用その他

A-62. 丸め誤差伝播に対する確率的モデルの験証

T.E. Hull and J.R. Swenson: Tests of Probabilistic Models for Propagation of Roundoff Errors〔CACM, Vol. 9, No. 2, Feb., 1966, pp. 108～113〕

現在行なわれている統計的誤差評価の基本的仮定は次の二つである.

1. 各回の誤差は，一様分布する確率変数である.
2. 引き続く誤差は，互いに独立である.

この仮定を験証するために次のようなシュミレーションを行なった. ここで対称とする演算は，7090/94 の FORTRAN II による符号別絶対値演算であり，四捨五入（または切りすて）後，標準化する浮動小数点演算のみを考えている. 四捨五入の場合のシュミレーションは

イ　倍長演算によって得られた結果に対して，単長演算の有効桁の最下位を基準として（$-1/2$, $1/2$）の区間に分布する疑似乱数を発生させ，上の結果に加えて次の段階の数値とする.

ロ　単長演算を，通常の丸めで行なう.

ハ　倍長演算の結果をそのままとる. この実験ではこの値は正確なものと仮定した.

イが，確率論的四捨五入のモデルである. ロ，ハは，比較のため行なう. 多数回の演算を繰り返えした後に，イ—ハの比較による誤差の平均を M, 標準編差を S とし, ロ—ハによって得られる誤差を E とすれば, この一つながりの実験によって $(E-M)/S$ が定まる. この実験を各種の問題について繰り返えせば，$(E-M)/S$ の値の分布によって，仮定 1, 2 を実験的に，確かめることができよう.

詳細に論じられている実験例は，差分近似

$$y_{n+1}=y_n+(h/2)\cdot(f(x_n, y_n)+f(x_n+h, y_n+hf(x_n, y_n))$$

によって，微分方程式

$$y'=x/y$$

を $h=0.0625$ で初期値 $y(0)$ を変えて，それぞれ百ステップごとに $(E-M)/S$ を求めたものである.

結論として，仮定 1, 2 は，良い近似であること，また，切り捨て方式の場合も，同様の結果を与えると報告しているが，正規分布から予想される分布とは食い違うこともあることを指摘している. この理由としてたとえば，四捨五入の場合は，イのモデルは exact な値に対しては，過評価であり，その結果 S が，実際の値より大きすぎ，$(E-M/S)$ が少さすぎると思われる. また切りすてのときは，M が大きくなりすぎるであろう. 実験では，数百回の試行中，$|E-M/S|>2.58$ となったものは，四捨五入による場合，4% 程度，切りすてによる場合，7% 程度（正規分布であれば，略 1% と期待される）であった.

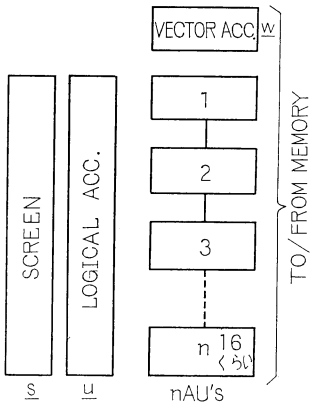
他の食い違いは，相対的にわずかである. この理由は，各回の誤差は，統計的な考察を必要とするほどには，多くのステップにまたがって影響すると考えなくて良いからであろう.　（牛島照夫）

C-63. 並列データ処理向きの計算機

D.N. Senzig and R.V. Smith: Computer Organization for Array Processing —VAMP (Vector Arithmetic Multi-Processor)〔Proc. FJCC, 1965, Part I, pp. 117～128〕

大がかりな天気予報をつくる場合など，いくつものデータのまとまりに対して同一の計算を行なうからこれを並列処理することによって効果を上げることができる. VAMP は，この目的に多重演算装置を適用した点，またそれらを1個の中央制御装置の管理下に規則正しく配列して置いた点，SOLOMON 計算機と同じである. ただし，演算はビット並列に行なわれる.

VAMP の演算部は n 個の演算装置の配列とその周辺のレジスタからなり，（**第1図**参照）各演算装置は累算器としてのXレジスタと，記憶部と他のレジスタのバッファとして使うZレジスタを持っている. u レジスタはXレジスタの内容について行なった論理操作の結果を貯めるもので，Sレジスタは各演算装置のスクリンである. たとえばSに（0 1 1 … 1 1）がセットされていれば1番上のものは演算を行なわない. u

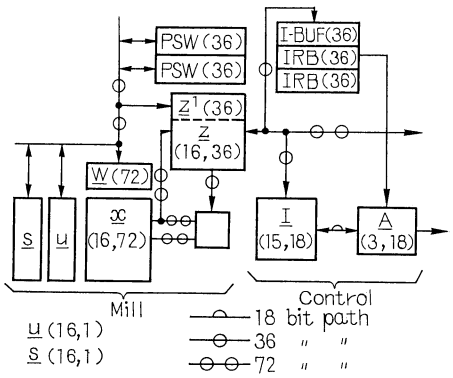


第1図　演 算 部

の内容をSに転送することによって次の計算を制御できる．

一方これらと別に高速の演算装置が用意してあり，並列操作を必要としない時に用いるようになっている．

VAMP の記憶装置は，演算装置の速度に合わせるため並列にアクセスできるが，SOLOMON とは異って，各演算装置所属の形成をとっていない．したがって全記憶装置はどの演算装置とでもデータのやりとりが可能である．番地指定には直接式と関接式とがあって直接式では命令で開始番地と増加分を与えると，n 通りの番地が計算され，n 個1ぺんに実行されるし，関接式では記憶装置の中に番地を貯めておいてそれをZレジスタに呼び出してくる．記憶装置が，普通の計算機と同様に使えるということは，動的記憶装置割付やテーブル・ルックアップに大変有利であり 7094 でシミュレーションした所によると，簡単な先回り制御



第2図　シミュレーションされた演算部

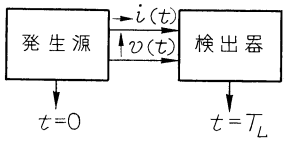
と時分割を使えば，番地列の計算はさして不利な要素にはなっていない．　（真子ユリコ）

D-64. 論理演算装置の特性を表わす測度：論理量子，論理係数，論理価

L.J. Giacoletto: On Measure of Logic Performance: Logic Quanta, Factor and Figure of Merit. [IBM J. Res. and Dev., Vol. 10, No. 1, Jan., 1966. pp. 51～64.]

電子計算機でいろいろな論理演算を行なうために，受動素子や能動素子を組み合わせて回路を構成するが，それらの回路の論理的な特性を比較する際に実際の特性をよく反映した指標があると都合がよい．

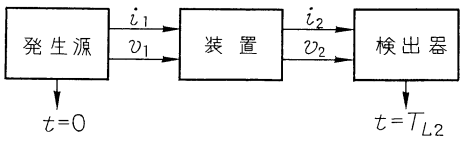
基本論理演算としてここでは feed-in と feed-out を考察する．すなわちいくつかの論理決定を組み合わせて一つにし，それをいくつかの負荷に分配することである．これは一つの論理決定をある場所から別の場所へ移すことに還元できる（第1図）．



第 1 図

$t=0$ における論理決定を伝達するために，発生源は検出器にエネルギーを送る．検出器が論理決定をするためにエネルギーの閾値を越えるのに有限の時間 T_L，エネルギー W_L が必要である．$H_L=W_L\cdot T_L$ を論理量子という．発生源と検出器を整合することによって論理量子を最少にすることができる．これを $H_L|_{op}$ とする．普通は W_L を固定して，T_L を最少にすることが行なわれる．

発生源と検出器の間にある装置を挿入した場合（第2図）すべてが整合しているときの論理量子を $H_{L_2}|_{op}$



第 2 図

として，論理係数 $F_L=\dfrac{H_L|_{op}}{H_{L_2}|_{op}}$ を定義する．$F_L>1$ は装置を挿入することによって論理増幅が行なわれた

ことを意味し T_L を増さないで分岐を行なうことができる．

発生源の電力 P_S を大きくすると T_L が小さくなり，F_L は大きくなる．そこで $F_L=1$ のときの W_L と P_S とのこれを論理価と定義する．

$$T_{FM}=\frac{W_L}{P_S}\Big|_{F_L=1}$$

以下論理係数についての具体的な計算が，線型回路，リレー回路，非線型回路について示されている．

結論として T_{FM} が，利得・帯域幅による表示よりすぐれている点は前者は非線型な動作に対しても適用できる点である． （相馬 嵩）

E-65. 多機能回路網の収束，振動および信頼性に関する新しい結果

R.C. Urbano: Some New Results on the Convergence, Oscillation, and Reliability of Polyfunctional Nets, 〔IEEE Trans. EC, EC-14, No. 6, Dec., 1965, pp. 769～781〕

論理回路網を構成する素子の出力関数として特定の一つの関数が与えられるのではなく，関数のある集合が与えられる場合を考える．

入力数 k_i, 出力数 l_i の2値論理関数全体の集合は，$2^{2^{k_i}\cdot l_i}$ 個の要素からなるが，この部分集合 S_i に含まれる任意の関数を出力関数としてとりうるような N 個の論理素子 $E_i(i=1, 2, \cdots, N)$ から構成されている m 入力，n 出力の論理回路網を多機能回路網（polyfunctional net）と呼ぶ．もちろん回路網自体の出力関数の集合で与えられる．

多機能回路網の各素子の入力数と出力数がそれぞれ回路網自体の入力数と出力数に等しいとき（すなわち，$k_i=m$, $l_i=n$; $i=1, 2, \cdots, N$），この回路網は斉次（homogeneous）であるという．斉次回路網の各素子を回路網自身でおきかえる操作を“完全な繰り返えし”（complete iteration）という．

最初の回路網の出力関数の集合を F_1，一度完全な繰え返りしを行なった回路網の出力関数の集合を F_2, …とすれば，関数の集合の系列 $F_1, F_2, \cdots$ が得られるが，m 入力 n 出力の論理関数全体の集合は有限個の要素からなり，したがってその部分集合の数も有限であることから，ある正整数 p, k が存在して，任意の $i\geq k$ に対して，$F_{i+p}=F_i$ となる．$p=1$ のとき，この回路網は収束するという．また $p\geq 2$ のときには振動するといい，p をその周期という．

つぎのようなことが問題となる．

（a） 斉次回路網と F_1 が与えられたとき，その振動周期はいくらか．またどのような条件のもとで収束するか．

（b） 斉次回路網と $F_1=\{f_1, f_2, \cdots, f_k\}$ および f_i の確率 $p_i(\sum_{i=1}^{k} p_i=1)$ が与えられたとき，何度かの完全な繰り返えしを行なった後のある関数 g の確率はいくらか，また繰り返えしの数を増加することにより，この確率が1に近づくのはどのような条件のもとでか．

これらの問題は，回路網の入力数 m，出力数 n，素子数 N，回路網の構造 G，最初の関数の集合 F_1 などに依存しているが，Urbano はすでに発表した論文で，種々の場合に分けて考察しているが，この論文では，特に $m=1$ の場合について考察し，新しく得られた結果を述べている． （橋本昭洋）

F-66. 計算機の配線面における配線についての関数を最小にするアルゴリズム

Tomaso Pomentale: An Algorithm for Minimizing Backboard Wiring Functions 〔CACM, Vol. 8, No. 11, Nov., 1965, pp. 699～703〕

部品の位置と接続すべき箇所とを与えて，計算機の配線面における配線の全長を最短にするような配線方法や，交点の数を最小にするような方法を求めることはすでに行なわれている．

この論文は，二点間を水平または垂直の配線の合成によって結ぶと仮定し，二点間の接続が水平と垂直の両者の配線を必要とする鉤状配線となる場合の数を最小にする方法を論じている．

配線面を $a\times b$ の単位に分け，部品の配置をマトリックス $P=[P_{ij}]$ で表わし，また N 種の部品があるものとしてそれら相互間の配線を $N\times N$ の対称マトリックス $C=[C_{ij}]$ で表わすと，ある一つの部品について水平または垂直方向に接続される配線の本数は P と C より求めることができる．

これをマトリックスで表現すると

$$[A, B]=\begin{bmatrix} A_{11} & A_{12}\cdots\cdots & A_{1a} & B_{11} & B_{12}\cdots\cdots & B_{1b} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{N1} & A_{N2}\cdots\cdots & A_{Na} & B_{N1} & B_{N2}\cdots\cdots & B_{Nb} \end{bmatrix}$$

となる．ただし，A_{ik} は素子 E_i と k 番目の水平線に属する素子とが接続される配線数，B_{ik} は E_i が k 番目の垂直線に属する素子と接続される配線数である．これは基本マトリックスであり，この論文の付録に示

されている手段によれば，水平と垂直の部分を同時に必要とする鉤状の配線の数をこれから容易に求めることができる．

あとは，鉤状配線の数が減るように基本マトリックスを Dounhill Method に似た繰り返えし計算によって変形していけばよいのであるが，最初に与えるマトリックスが適当でないと，計算時間が長くなったり，得られた結果が必ずしも最小でない場合がある．

この論文では，配線面を 9×9 単位に分け，素子の種類を 81 とした場合の計算例が述べられている．

（石立　喬）

F-67. タイム・シェアリングの二つの側面

M. Greenberger: The Two Side of Time-Sharing〔Datamation, Vol. 11, No. 11, Nov., 1965, pp. 33～36〕

タイム・シェアリング計算システムを考察するとき，二つの観点，すなわち，道具（システム）の性質からみる場合と，使用者からみる場合がある．本論文は，その二つの観点からタイム・シェアリング・システムとそのスーパーバイザについて，MIT で行なわれている MAC 計画の経験にもとづいて，考察をしている．さらに，使用者側の観点に立って，設計製作された，OPS と呼ばれるシステムについて述べ，将来の展望を行なっている．

オンライン使用者にとって，エコー検査，計算，中間結果の表示，ガイド，キー変数の追跡などが便利である．また，通常の数値計算だけでなく，シミュレーションや統計の機能も有用である．

OPS はこれらの機能を備えた標準オペレータ（群）を持ち，使用者はターミナルからこれらのオペレータを記号的に参照したり，それらを自由に複合したりすることができる．

標準オペレータのそれぞれについて簡単な解説が行なわれている．その中で特徴的なのは，たとえば，行列やベクトルを記号的に指定，演算のできる SET オペレータ，多項式を扱う POLY オペレータなど，統計解析のための FIT オペレータなど，待行列（キュー）を扱う STACKQ などのオペレータ，DELAY，CANCEL などのシミュレーション用オペレータなどであり，またフレキシブルな入出力や報告，編集，監視のためのオペレータが備えられている．

将来の方向としては，システム側としては，多プロセッサ（多モジュール）方式や記憶のリロケーションのハードウェアの出現とともに，それに適合した能率的なスーパーバイザが開発されるであろうこと，一方使用形態から見れば，自然（に近い）言語の使用や，ターミナルにおける図形，音声入出力装置の使用が開発されるであろうことを指摘し，コンピュータ・ユティリティの普及についての確信を述べている．

（渕　一博）

F-68. コンパクトな計算機の比較

E.O. Boutwell: Comparing the Compacts〔Datamation, Vol. 11, No. 12, Dec., 1965, pp. 61～73〕

計算機の語長は，36～48 ビットが普通であったが，1961 年から 18～30 のものが増え，1963 年からは 12～18 のものも増えてきている．これは，精度を要する数値計算のほかの分野に計算機が使用されるからだろう．アメリカの代表的な小形機 12 種の特性を 31 項目にわたって表にし，傾向を検討した．

命令語の形式は，基本的に三つの部分からなり，16 ビット以下の機種では，倍長命令語がおかれている．相対アドレス方式が一般的で，命令場所カウンタで変更するものや，カウンタが上位桁だけを提供するものではデータが命令の近くにおかれる．これらではいづれも，命令やデータを先取りするスクラチパッドが有効で，30～500 語程度のものがおかれている．

インデックスによるアドレスの変更は大幅には採用されていないが，記憶装置や累算器で変更するものもある．間接アドレスの変更に先立ってインデックスによる変更が行なわれる．マイクロプログラミングを採用している機種もある．

記憶容量は，16～31 K 語であるが，現実に設置されたものは 4 K や 8 K も多い．アクセスはプログラムの進行と入出力のデータ転送とが並列に行なわれる．記憶保護の鍵をスーパーバイザが扱うものもある．

入出力は，データチャネルを介して実行され，字単位に外部と支持する．記憶装置の使用頻度はたかだか 1 Mc である．入出力制御語は，チャネルにおく機種が多く，記憶装置におくものもある．データの連鎖を許すものは二つしかないが，指示の連鎖とともに今後はとり入れられていくだろう．

割り込み処理は，原因別の記憶場所へ飛び越しを行なう形式である．集積回路のものが 5 機種，ハイブリッドが 1 機種ある．

値段は，20 K～50 K $ なので，大きな問題を扱うのに小型機をたくさん使う方がよいということになるだろうと思われる．（川合英俊）

F-69. 推理問答プログラムに関する実験

James R. Slagle: Experiments With a Deductive Question-Answering Program〔CACM, Vol. 8, No. 12, Dec., 1965, pp. 792～798〕

人工知能の研究は知的機械行動を引き起こすことを目標とする．計算機プログラムに知的機械行動を起させる多くの試みが heuristic プログラムを用いて行なわれた．これらのプログラムのあるものは知的にむずかしい問題を解くことができたが，広い範囲の問題に対する適応性および学習機能を持たないという意味で知的であるとはいえなかった．

著者は広い範囲の問題に対する適応性および学習機能を持った heuristic プログラムを得るために DEDUCOM と呼ばれる推理問答（deductive question-answering）プログラムを LISP 言語で書いた．DEDUCOM は LISP インタープリータの拡張で約 2 K 語の大きさであり，IBM 7094 計算機で実験された．この論文は DEDUCOM の方法論および DEDUCOM が行なった問答について記述している．人間の実験者はまず一般的な問題に答える方法を DEDUCOM に教える．次にいくつかの事柄およびそれに関連した問を与える．実際の問は，たとえば，次の形式で DEDUCOM に与えられる．

（1） howmany〔FINGER; HAND〕

この意味は“How many fingers on a hand?”である．事柄は，たとえば，次の形式で与えられる．

（2） ansupset〔howmany〔FINGER; HAND〕; 5〕

この意味は“There are fine fingers on a hand.”である．

事実（2）が与えられたあとで問（1）が与えられると DEDUCOM は形式 howmany〔FINGER; HAND〕を付値して値5を得る．そして5と答える．実験の結果，DEDUCOM は6種類の問に答えることができた．それらの問題は yes, no で答えるもの，数値で答えるもの，または簡単なプログラムを生みだすものなどであった．このことは DEDUCOM が広範囲の問題に対する適応性および学習能力のしるしである self-program の機能を持つことを実証するものであると著者は主張している．（二村良彦）

F-70. FORMAC における自動単純化

R.G. Tobey, R.J. Bobrow and S.N. Zilles: Automatic Simplification in FORMAC〔Proc. FJCC, 1965, pp. 37～53〕

数式を扱かうシステムにおいては数式を単純化する機能が必要である．しかも単純化はできるだけ自動的に行なわれることが望ましい．この論文では FORMAC（FORmula MAnipulation Compiler）プログラミングシステムにおける単純化の役割および単純化のアルゴリズムを作成するための一般的方法が記述されている．FORMAC は数式を数学的に解析するプログラムを書くためのシステムである．それはフル FORTAN Ⅳ 言語の拡張である FORMAC 言語，FORMAC ステートメントを FORTRAN Ⅳ ステートメントに変換する preprocessor および preprocessor で呼びだされる object time ルーチンの集合，の三つの部分から構成される．

FORMAC においては，全ての式に適用できるような単純化変換（主として作用素同志の間の変換，たとえば，$\log 5bc \rightarrow \log 5+\log b+\log c$）は自動的に行なわれ，それらは全て AUTSIM（automatic simplification）と呼ばれるルーチン（object time ルーチン）に含まれる．そうでないような変換，たとえば数式の展開および因数分解，および常数引数を持つ関数の付値（たとえば，X+SIN(1.4)→X+0.98545），はプログラマの選択にまかせられる．自動単純化変換（AUTSIM）は数式を擬似標準形式に還元する．このようにすると扱う数式の形式が決まるので自動単純化のアルゴリズムが簡単になる．

AUTSIM ルーチンによって変換された数式は LEXICO と呼ばれるルーチンによって整頓される．

LEXICO ルーチンは式を走査して，各項を辞書式順序（lexicographic order）でソートし同類項または同類因子をまとめて係数を簡単にする．

このように自動単純化は AUTSIM ルーチンおよび LEXICO ルーチンを数式に回帰的に適用することによって行なわれる．

FORMAC の能力を示す一例として，人間が行なうと 60 年はかかると推定される天文学上の数式を得るために，IBM 7094 は 18.67 分しかかからなかった例をあげている．（二村良彦）