解　説

数　値　計　算†

戸　田　英　雄††

1. まえがき

“機械にでも出来る計算を人間がやるのは馬鹿げている”とした数学者ライプニッツ (Leibniz) の時代から描かれていた人類の夢が実を結んだものと言える電子計算機（以後本文ではコンピュータと書く）は，1940年代に誕生のときすでに，計算にかけては天才的な能力を持っていた．草わけのコンピュータである ENIAC (Electronic Numerical Integrator and Calculator) は1946年に，微分解析機を10進法のままで強引にデジタル化して作られ，常微分方程式で記述される弾道の計算の専用機であったと伝えられている．コンピュータの計算速度も，第1世代の真空管方式，第2世代のトランジスタ方式，第3世代の集積回路 (IC) 方式，第4世代の大規模集積回路 (LSI) 方式と進化するにつれて，ミリセコンド (10^{-3} 秒) からマイクロセコンド (10^{-6} 秒)，さらにナノセコンド (10^{-9} 秒) のオーダの速さとなり，今やコンピュータは名前の示す通り超天才的な計算能力を持つに至った．

1950年前後から，1960年前半にかけては，わが国はコンピュータ草分けの時代であった．外国では，機械計算でこのような仕事が出来たというニュースを耳にして，わが国でも！と思う人が少なくなかった．

とくに計算に追われる分野の科学者にはそのような気持が強く，涙ぐましい努力が続けられた．この時代の数値計算関係の歴史を現代に視点をおいて透視図的に眺め，今から見ればどういうことをやっていたのかを歴史的に考えてみる．

なお1950年～1960年にかけての数値計算法の模様をよくまとめて記述してあるのは，

山内二郎，森口繁一，一松信共編の電子計算機のための数値計算法I 数理科学シリーズ 培風館(1965)である．また，歴史的な展望と最近の研究までをよく解説してあるのは，一松信著 数値解析 朝倉書店(1982)であろう．筆者はこの書物からいろいろのことを勉強させていただいたが，一松流の“名文句”を本歴史の記述にも引用させていただき，感謝する次第である．

2. 歴史的な考案

1950年の前半を，IBM の統計機（パンチカード分類機とか会計機等）をいろいろ工夫して利用した PC（パンチ・カード）時代とすれば，1950年の後半はようやく出はじめた国産機活用の CP（コンピュータ・プログラミング）の時代と分けて考えられる．この間の数値計算の分野の中で，主として線型計算，代数方程式，常微分方程式，関数近似等についての実例（史料）を現在の眼で選択し歴史的な流れを，私なりに，考察してみる．

2.1 線形計算

連立1次方程式の解法は古くから研究されてきたが，理論としてはクラメルの公式が有名である．しかし，この公式では n 元の場合かけ算の回数だけみても $n!$ のオーダの手間がかかり，n が大きいと実用的には無理であろう．数値計算で解を求めるときには普通 n^3 のオーダの手間の消去法(注1)が用いられ，特殊な問題では反復法が用いられていた．

1950年代のコンピュータを用いた科学技術計算では，線形計算の問題が非常に多く，ちなみに，当時わが国ではじめての依頼計算を行った有隣電機富士電算計算所 (1956年開所) のリレー・コンピュータ FACOM 128は，設計そのものが線形計算むきであり連立1次方程式で商売が成立っていたと言われている．1950年代の線型計算はどの位の規模でどのような解法で行われていたかを具体的な実例で述べてみる．

実例1　産業連関分析用の20次の行列の逆転(1954年)

今日では20次位の行列は小さい行列と思われているが，1954年 (昭29) 頃は未だ手計算（卓上計算器は用いる）が主流で，20次の行列は大規模な行列に属し，当時利用できたパンチカード式統計機 IBM 602A

† Numerical Methods by Hideo TODA (Faculty of Engineering, chiba University).

†† 千葉大学工学部

で計算時間の見積りが約10時間という規模の問題である.

ちなみに，筆者が学生のとき1949年（昭24）頃と思うが，山内二郎先生のある研究用の問題で，5行5列の密行列の逆転を消去法でするのに，タイガ計算器をまわして，検算まで含めて6～7時間かかった記憶がある.

通産省側の希望では200次位の行列を逆転したいところであったが，その当時利用できた唯一の計算機械である IBM 602A 計算穿孔機で約**10年位**の計算量となるので，規模を小さくして20次の行列の逆転を行うことにしたそうである．通産省の鴨志田さんをトップにするチームで森口先生が計算技術の相談役をされていた.（➡ 座談会 日本のソフトウェアの草創期）なお，当時の苦心談は 1)に細かに記述されているが，そのあらましを述べて参考にしていただく：

大きな行列を逆転すると有効数字が失われるが，当時のかなり用心した見積りによると（Von Neumann-Goldstine）たとえば15次の行列の逆転をすると8桁失われるとされていた．IBM 602A では8桁までの乗除算は簡単であるが，8桁を超えると急に手間が何倍もかかるが，用心のため14桁演算ルーチンを組み込んで行ったと 1)に記述されている.

今日では FORTRAN 語で4倍精度計算までは簡単にプログラムで書いて実行できるが，当時は14桁計算でも大へん苦心したわけである．さらに，もう一つの苦心は**検算**であった.

手計算の経験のある人には常識であるが，検算のない数値計算は，全く他人に信用して貰えない時代であった．IBM 602A での計算もカードをオペレートする手作業が入るので，消去を一段進めるごとに各行の計を計算して検算するサム・チェック方式が取られていた．それで計算の真中の5時間位のところで，サム・チェックにひっかかり，その原因がサム・カードを機械にパンチさせればよかったのを，つい手でパンチしたためパンチ・ミスがあったと分かった．しかし間違いの原因はそれだけであろうかという疑問に対しては，“最初，誤差が小さいと仮定してその2乗以上を無視する方針で近似的に公式を導き，その計算を実行することにより他には誤りがないと確められた”と当時の数値計算の苦心が語られている.

今日では，20次位の行列の逆転は，コンピュータで“あっ”という間に出来てしまうが，当時は本当に涙ぐましい努力がなされたわけである.

（注1） ガウス・ジョルダン法（掃き出し法）1950年代に用いられた消去法は主としてプログラミングの簡単なガウス・ジョルダン法と呼ばれる掃き出し法である．これは，算法が前進部分のみで後退部分が不要な消去法で計算の手間は乗除算で $O(n^3/2)$ である．密行列のときは，枢軸選びを行うガウスの消去法（LU分解）は，計算の手間が乗除算で $O(n^3/3)$ であり，積和計算を工夫すると精度の点からも，ガウス・ジョルダン法より優れているというのが現在では常識となっている.

これは，今では，コンピュータ・システムの進歩から非常に大きな次元の行列も扱えるようになったこととプログラム言語が進歩して枢軸選びもそんなに面倒でなくなったためであろう.

“しかし，将来 n 元連立1次方程式を，n 個の演算回路を並べた専用の並列計算機で直接1度に解くような場合には，ガウス・ジョルダン法の方が作り易いとされている．最適な算法は，常に環境とともに変るので，柔軟な頭で対応すべきであろう”．と一松[2]は，最適な算法の無常性に注意を与えている.

実例2 高橋[3]による行列式その他の exact calcultion に対する新方法

コンピュータで取り扱う数値の桁数は，たとえば10進8桁とか2進36ビットとか決まっていて，2倍桁演算や4倍桁演算でも結局は有限桁で押えられている．それより桁数の多い結果が，たとえば乗算で出てくれば下の桁は‘丸め’によって除かれてしまう．そこでこの‘丸め’を全く行わない厳密な計算が必要となることがある．実例1 で述べた産業連関行列は性質のよい行列であったが，‘たちの悪い’行列の逆転のときは，二つの大きな数のひき算で上の方の桁が大きく消えてしまういわゆる**桁落ち** が起きる.

このようなときは，通常は多倍長精度の計算で精度を上げるが，最も徹底したやり方は，最初に与えられたデータを整数と見て途中の演算は素数 p を法として計算し，最後まで‘丸め’を行わないで計算する．このことを，いくつかの素数 p について計算した結果から本当の答を再現する(注2)というユニークな算法である.

この新しい算法は高橋秀俊先生によって考案され第1回プログラミング・シンポジウム（1960）で報告された．また，これは PC1（パラメトロン式コンピュータ）でプログラムされ（石橋氏），世界でも類例のない画期的なものであった．日本のソフトウェアの草創

期の先駆的な業績として，この算法の論文は情報処理学会誌の第1巻を飾ったわけである(➡ 3))．

これに共鳴して，各計算センタ等ではこの算法のプログラムが作成された．筆者もその一人である．その当時から見ると，コンピュータは格段と進歩しているので，もう一度現代的にプログラムを作りなおしておく価値は十分あると思う．

(注2)　105 ソロバンの原理で，たとえば，連立合同式 $x\equiv 2 \bmod 3$, $x\equiv 3 \bmod 5$, $x\equiv 6 \bmod 7$ の解のうち $100\geqq x\geqq 1$ のものは次のようにして決まる．

$35y_1\equiv 1 \bmod 3$ の一つの解は $y_1=-1$

$21y_2\equiv 1 \bmod 5$ の一つの解は $y_2=1$

$15y_3\equiv 1 \bmod 7$ の一つの解は $y_3=1$

$$2\cdot 35\cdot(-1)+3\cdot 21\cdot 1+6\cdot 15\cdot 1=83$$

そこで $x\equiv 83 \bmod 105$ なので $x=83$ ときまる．

実例3　共役勾配法 (cg 法) の誕生と現状

1950年代のトピックスに，連立1次方程式の解法で，Hestenes と Stiefel が考案した method of conjugate gradients (共役勾配法) という新顔の登場がある．これは降下法型の反復計算で逐次真の解に近づいていく方法であるが，もし‘まるめ’の誤差がなければ n 元のときは n 回で終結することが証明されている．プログラムは簡単なルーチンの繰返しで，行列の要素に0が多いときはその性質が最後まで保存されるなどの利点から，発表された当初は今世紀における画期的な算法として花々しく登場した．しかし初期の熱中時代のあとは，消去法や反復法に優位を譲ってしまったかに見えた．解法の原理は次の通りである：

未知数のベクトルを $\boldsymbol{x}$，係数の行列を A，右辺のベクトルを $\boldsymbol{b}$ とし，n 元連立方程式を

$A\boldsymbol{x}=\boldsymbol{b}$　　と書く．

解ベクトルに対する方 i 近似ベクトルを $\boldsymbol{x}_i$ とし，この $\boldsymbol{x}_i$ の残差ベクトルを $\boldsymbol{r}_i$

とすると，$\boldsymbol{r}_i=\boldsymbol{b}-A\boldsymbol{x}_i$

である．第 $i+1$ 近似ベクトル $\boldsymbol{x}_{i+1}$ は $\boldsymbol{x}_i$ を修正する方向ベクトル $\boldsymbol{p}_i$ をきめてから計算される．

ところで，この方向ベクトル $\boldsymbol{p}_i$ は，解に近づく程本質的に桁落ちして0に近づく残差ベクトル $\boldsymbol{r}_i$ から求める仕組みになっている．そのため $\boldsymbol{r}_i$ の‘まるめ’の誤差が積み重なって $\{\boldsymbol{p}_i\}$ が完全に直交しなくなる．

n 元連立1次方程式のとき，n 回で正しい解に収束するという定理は，実は $\{\boldsymbol{p}_i\}$ の直交性に基づいているのに　足もとの残差ベクトル $\boldsymbol{r}_i$ の精度の方が解に近づく程落ちてしまうというのが cg 法のアキレス腱である．

現在では，解の最終の第 n 近似 $\boldsymbol{x}_n$ まで機械的に計算しないで，残差ベクトルの大きさ $|\boldsymbol{r}_m|$ が適当に小さくなったらそこで打切り，中間の $\boldsymbol{x}_m$ を新しい初期値として**高精度演算**で再計算し残差ベクトルの精度をたかめるのがよいと言われている．このように細かい配慮のもとで cg 法は大型の疎行列 (数千元) の直接解法の一つとして1970年代に再評価されてきている．

“共役勾配法が生まれてから現在に至る歴史を考えると，優れた算法は，数値計算的立場から見直されてはじめて実用的となるという一つの典型的な実例であろう”．という一松[2)]の名言は，数値解析，数値計算法と呼ばれる学問の本質を指摘したものである．

実例4　ヤコビ (Jacobi) 法についての思出

正方行列 A に対して数 λ とベクトル $\boldsymbol{x}\neq 0$ で

$$A\boldsymbol{x}=\lambda\boldsymbol{x}$$

を満たすもの求めることが行列の固有値問題である．

1950年代は，対称行列 A の固有値，固有ベクトルの計算には，2次元の回転を反復して対角行列に近づけるヤコビ法が一番コンピュータ向きの算法とされていた．第1回プログラミング・シンポジウムで，今は亡き万能計算盤でも有名だった島内武彦先生が，この算法についての報告とコメントを行い，これに対して松谷泰行さんが討論者だったと記憶している．

いまでも，小さい (10次以下) の行列には有用であるが，現在では行列 A と相似な三重対角行列に変換してから固有値を求めるハウスホルダ (Householder) 法に優位を譲ってしまった感がある．

“数学者ヤコビ (Jacobi) による提唱が1830年代であり，コンピュータの草創期時代における実用化が1950年代であった．そして1960年代にはハウスホルダ法にその王座を譲ってしまった．ここにもまた算法の栄枯盛衰の歴史が見られる” (一松[2)])

さらにまた，Francis (1961) により QR 法が一般の複素行列の固有値の算法として提唱されて，現在では実対称行列にも有用な算法として使われるに至っている．その陰には，最近のコンピュータのシステムが，1950年代とちがって複素数演算を非常に容易ならしめた“計算の道具”の進歩を見逃すわけにはいかない．これは次に述べる非線形方程式の解法にも見られることである．

2.2　代数方程式

$$P_n(z)=a_0z^n+a_1z^{n-1}+\cdots+a_{n-1}z+a_n \qquad (1)$$

の零点を求めること，すなわち代数方程式を解く問題

は昔から多くの人により研究されているが，解法を大きく分けると次の二種類となる：

A）解の第1近似を必要としない方法　たとえば，グレッフェ (Graeffe) 法，ベルヌーリ (Bernoulli) 法ステュルム (Sturm) 法，レーマ (Lehmer) 法等

B）解の第1近似を必要とする方法　たとえば，ニュートン (Newton) 法ベアストウ・ヒチコック (Bairstow-Hitchcock) 法等

1960年前後は算法言語として ALGOL 60 や FORTRANⅡ から FORTRANⅣ 等が利用できるようになっていた．代数方程式の解法のプログラムもこれらの算法言語で記述され，A)型，B)型，その組合せ型，A)やB)変形法等が研究用，実用，ライブラリ用に作られ報告されていた時代である．

現在では，代数方程式の場合には予め解の第1近似値が分らない場合でも適用できるニュートン変形法が推奨されている．コンピュータ向きの解法としては平野法 (1967) と連立法 (DKA 法ともいう) (1966) である．

実例5　平野法[4]について

平野法はニュートン法の改良版であって原理は次のようになる（6)に分かり易く解説されている)：

(1)の第0近似解 $z^{(0)}$ を修正するとき，$z^{(0)}$ が十分に(1)の根 α に近いとみて，$\alpha=z^{(0)}+\varepsilon$ とおき

$$0=P_n(\alpha)=P_n(z^{(0)}+\varepsilon)$$

を $z^{(0)}$ のまわりにテーラ展開し，

$$P_n(z^{(0)}+\varepsilon)=c_0{}^{(0)}\varepsilon^n+c_1{}^{(0)}\varepsilon^{n-1}+\cdots+c_{n-k}{}^{(0)}\varepsilon^k+\cdots+c_{n-1}{}^{(0)}\varepsilon+c_n{}^{(0)} \quad (2)$$

を組立除去で求める．

ここで，ニュートン法では，ε^2 以上の項を無視して

$$0=c_{n-1}{}^{(0)}\varepsilon+c_n{}^{(0)}$$

とおいて ε の近似値 $\varepsilon^{(0)}$ を

$$\varepsilon^{(0)}=-c_n{}^{(0)}/c_{n-1}{}^{(0)}$$

と求める．そこで根 α の次の近似値 $z^{(1)}$ を

$$z^{(1)}=z^{(0)}+\varepsilon^{(0)}$$

とおくわけであるが，平野法では(2)の右辺で最も**影響力の強い項**に注目して ε の近似値 $\varepsilon^{(0)}$ 値を求めるのに

$$c_{n-k}{}^{(0)}\varepsilon^k+c_n{}^{(0)}=0$$

の解から ε の近似値として ε_k を

$$\varepsilon_k=(-c_n{}^{(0)}/c_{n-k}{}^{(0)})^{1/k} \quad (3)$$

で計算する．このとき減次のことを考えて

$$|z^{(0)}+\varepsilon_k{}^{(0)}|$$

が小さくなるように選ぶため

$$|\varepsilon_m|=\min_{k=1,\cdots,n}|\varepsilon_k| \quad (4)$$

なる m を定めて次の近似値 $z^{(1)}$ を

$$z^{(1)}=z^{(0)}+\varepsilon_m \quad (5)$$

とする．この考え方は，**最も影響力が強い項は最も小さい修正量で満足される**ということに基づいている．さらに“減速”の考え方を導入して一度に定数項 $c_n{}^{(0)}$ を消すことはしないで $\mu=1, \frac{1}{2}, \frac{1}{4}, \cdots$ と変化させて，(3)の代りに

$$\varepsilon_k(\mu)=(-\mu c_n{}^{(0)}/c_{n-k}{}^{(0)})^{1/k} \quad (3)'$$

(4)の代りに

$$\varepsilon_m(\mu)=\min_{k=1,\cdots,n}|\varepsilon_k(\mu)| \quad (4)'$$

とする．

$$|P_n(z^{(0)}+\varepsilon_m(\mu)|\leq\left(1-\frac{1}{4}\mu\right)|P_n(z^{(0)})|$$

$$\mu=1, \frac{1}{2}, \frac{1}{4}, \cdots$$

となったときこの $\varepsilon_m(\mu)$ を用いて

$$z^{(1)}=z^{(0)}+\varepsilon_m(\mu) \quad (5)'$$

で $z^{(1)}$ を定めるというのが平野流の考え方である．

ニュートン法は局所的な収束性は優れているが大域的な収束性は保証されていない．そこでA）型の解法で出発させる工夫もなされたが，平野法では**減速法**を導入しているので大域的に収束が保証されることになる．実際，氏は平野法に基づくプログラムを FORTRAN で記述し，このプログラムで大域的に収束しないような問題を提起した方には賞金を進呈するという程の自信があったわけである．

われわれの仲間では，このプログラムを**平野の大薙刀**といい代数方程式の数値解法用として当時の決定版であった．

なお，この算法が大域的に収束することの証明が与えられたのは，ごく最近 (1980年) のことで，伊理研の室田[5] による．まさに平野・室田による**画龍点睛**といえよう！

実例6　連立法 (DKA 法)

フランスのデュラン (Durand) が古く提唱し後にドイツのケルナ (Kerner) が別の立場から解釈を与えた (1966年)もので，さらにイギリスのアバース(Aberth) が1970年に初期値の与え方を示したもので3人の名前の頭文字を取って DKA 法とも呼ばれる．解の事後評価法にスミス (Smith) の定理 (1970) を用いるとすれば，DKAS 法と呼ぶべきかも知れない．

デュランの最初の考えはニュートン法の代用品であった．すなわち複素係数 a_i をもつ n 次の代数方程式

$$P_n(z)=z^n+a_1z^{n-1}+a_2z^{n-2}+\cdots+a_{n-1}z+a_n=0$$

の根を $z_1, z_2, \cdots, z_n$ と書くと

$$P_n(z)=(z-z_1)(z-z_2)\cdots(z-z_n),$$

$$P'_n(z_i)=(z_i-z_1)(z_i-z_2)\cdots(z_i-z_{i-1})(z_i-z_{i+1})\cdots(z_i-z_n)$$

$$=\prod_{j\neq i}^{n}(z_i-z_j)$$

となる．

根 z_i の第 k 近似を $z_i^{(k)}$ と書き

$$z_i^{(k)}\fallingdotseq z_i$$

とみなせば，

$$P'_n(z_i^{(k)})=\prod_{j\neq i}^{n}(z_i^{(k)}-z_j^{(k)})$$

で代用できるであろう．

根 z_i の第 k 近似 $z_i^{(k)}$ から方 $k+1$ 近似を求めるニュートン法は

$$z_i^{(k+1)}=z_i^{(k)}-P_n(z_i^{(k)})/P_n'(z_i^{(k)})$$

なので分母をさきほどの積和で代用して

$$z_i^{(k+1)}=z_i^{(k)}-P_n(z_i^{(k)})\Big/\prod_{j\neq i}^{n}(z_i^{(k)}-z_j^{(k)})$$

$$i=1, 2, \cdots, n$$

と，したのがデュランの考え方であった．

デュランはこの公式の収束性を示し最終的には２乗の収束となることを証明している．

この算法が連立法と呼ばれる理由は，n 個の根が**同時**に求められるからである．従来の代数方程式の数値解法では，たとえば一根 z_1 を求めると $P_n(z)$ を $(z-z_1)$ でわり算して $n-1$ 次の多項式を作りこれから次の根 z_2 を求めるという工合に，次数を低減する方法が多かった．連立法は**並列演算**むきで，各根の収束の速さはほぼ一様である．一松[2]の言を借りれば，“仲のよい団体旅行者のように互い助け合いながら目的とする所に全員が収まる”という利点があり，現在では理論的にも実用的にも最もよい算法とされている．その陰に，最近のコンピュータ・システムでの複素演算の使い易さというシステム自身の発達を見逃してはならないと思う．

1960年代に森口繁一先生の研究室で，チェビシェフの数値積分公式に関連して現われる一連の多項式の零点の配置についての**森口・伊理**の予想が立てられ，これを実験的に確めてみようということになった．

これを確める有力な算法としてこの連立法が用いられ倍精度計算で60次式の60個を根を十分な精度で求め，４倍精度で180次〜200次位まで解いたのは1970年代になってからである．さらにこの試みを進めて小野令美[7]は多倍長計算用の FORTRAN サブルーチンを作り256次，512次，1024次の根を精密に求めることに成功している．

さらに，超高速コンピュータ CRAY-1 で実働２時間位で2048次の高次方程式を解いて世界的な記録の更新をしている（1981年）．(➡ 8))

ここにもまた，コンピュータの発展と算法の進歩による数値計算の歴史における輝かしい一潮流をみる思いがする．

2.3　常微分方程式

常微分方程式 $y'=f(x, y)$ は解析的に厳密解が得られることは少なく，実用的には近似解で我慢する場合が多いと思われる．近似解法には二つの大きな方向がある：

A）適当な有限個の多項式や三角多項式等の結合で近似解を求める方法で，たとえば有限要素法はこれに属する．

B）連続的な変数 x の値を有限個の分点 $x_0, x_1, \cdots, x_n$ で代表させ，そこでの解関数の近似値 $y_0, y_1, \cdots, y_n$ で解の近似とする離散近似で差分法がこれに属する．

1950年から1960年代前半にかけて，常微分方程式の初期値問題

$$y'=f(x, y),\ \ y(x_0)=y_0$$

の差分法による数値解法に焦点を合せてみよう．

実例７　ルンゲ・クッタ・ギル（Runge-Kutta-Gill）法

ルンゲ・クッタ型の公式は，歴史的には，Runge (1895) が数値積分公式のうち有名なシンプソン則を拡張するという直観的な着想に基づく４段３次の公式（刻み幅 h で $x_1=x_0+h$ における数値解 y_1 を求めるのに，右辺の関数 $f(x, y)$ の計算が４回必要で，公式のもつ打切り誤差が $0(h^4)$ である，すなわち h^3 のオーダまでは正しい公式）を与えたことから幕があくことになる．

ついで Kutta (1901) が，条件方程式をたて，これを代数的に解く方法で公式を整理し，４段公式では，公式のもつパラメータのうち２つの自由パラメタを利用して６通りの４次の公式を与えた．このうち一番簡単なのが現在でも有名で，ルンゲ・クッタ（Runge-

Kutta) の公式と呼ばれる公式である.

1940 年代の末, ギル (Gill) が初期のコンピュータ EDSAC (Electronic Delay Storage Automatic Calculator) で記憶量を減ずる工夫と, 丸めの誤差のくりこみなどの技巧を取り入れた公式を発表した. 1960 年代は専らこの公式が研究され, また使用された(➡ 10)).

ちなみに, 1960 年 4 月に微分方程式の数値計算というテーマで数理科学総合研究第 2, 3, 4 班合同のシンポジウムが開催されたが, そのとき高田勝先生は "従来, シンプソン型のルンゲ・クッタ法とミルン法が有名でよく使われた. しかしルンゲ・クッタ法よりもルンゲ・クッタ・ギル法がいろいろな点で優れており, ミルン法は不安定要素を含んでいて, (たとえば伊理[9]の平滑子による) ある種の対策が必要となる. ミルン法よりハミング法の方がよい" と述べている.

その当時のコンピュータの環境条件のもとでは, ギルの改良が役に立ったと思われるが, 現在では古典的のルンゲ・クッタ法と比べて, 精度や安定性の面でどの位有利なのか疑問視されている. これに関して, 一松[2]が述べているように, "ギルの公式の人気は, 算法の性能自体よりも発表された当時の時代環境などの副次的な条件で決定された" のであろう. 算法は, その時代のコンピュータ環境により, その人気が決まるという一つの実例のようにも思われる.

実例 8 清水[11]の難問と TRAM について

1966年1月の第7回プログラミング・シンポジウムで, 常微分方程式の数値解法における各種の困難さが清水辰次郎[11]により指摘され, つぎに示す「難問」が提案された:

(S1) $dx/dt=x^3/2, \ x(0)=1$

(S2) $dx/dt=x-t, \ x(0)=1$

(S3) $dx/dt=-1/(2x), \ x(0)=1$

(S4) $dx/dt=x^6, \ x(0)=-2$

(S5) $dx/dt=-tx, \ x(0)=1$

(S6) $d^2x/dt^2+\mu(x^2-1)dx/dt+x=0,$
$x(0)=-2, \ x'(0)=0$

で, これらは次の宿題研究とされた.

1967年第8回プログラミング・シンポジウムで清水 11)の問題の結果がいろいろと報告された. その一つとして, 森口・伊理・小林[12]は TRAM (Trapezoidal Rule with Automatic step Modification) という名のプログラムを開発し, それによる実行結果を発表している. たとえば, 動く特異点のある方程式 (S1) や, 漸近する解をもつ方程式 (S4), van der pol の方程式 (S6) で $\mu=10$ の場合等でも TRAM でうまく解けることが報告された.

TRAM の算法は, 単純で "たくましい" 台形則に刻み幅の自動調節の機能をつけたもので, 粗い精度で確実に解きたいときの一つの標準的なやり方である. これは今日でもライブラリ・プログラムとして推奨されている.

実例 9 硬い方程式をめぐって

たとえば, $dy/dx=-Ay(ReA>0)$ で A が非常に大きいときは, いわゆる硬い方程式と呼ばれる. この解関数 $y(x)$ は, x が増大すると共に減衰していかねばならないが, 数値解法の刻み幅 h が大きすぎると, ルンゲ・クッタ型の公式や予測子型の多段公式による解は発散してしまうことは知られている.

Dahlquist (1957) は上記の方程式を深く研究し A 安定 (Absolutely stable) すなわち絶対安定という概念を導入した. これは一定の刻み幅で解いたとき $ReA<0$ であればいくら $|ReA|$ が大きくても収束する解が得られる差分公式は絶対安定であるという. 前進型のルンゲ・クッタ型公式や予測子型の多段型公式は A 安定ではないこと, また修正子型の公式でも局所打切り誤差が陰的台形則よりよいものはすべて A 安定ではないことが示された.

1960 年代になって, A 安定より弱い意味での硬い方程式専用の S 安定 (Stiffly stability) で実用的には十分であるという研究が行われ, ギア (Gear) の公式 (1969) 等が典型的である. 1970 年代の終りになって汎用の陰的ルンゲ・クッタ型公式の中から A 安定なものを探す研究が行われるようになり, これは今後の課題と思われるが, このような公式を導くための**道具**として**数式処理システム**の普久が望まれる.

これを示唆する例として Runge-Kutta 型公式の発達の小史がある:

Runge-Kutta 型公式は, 当初における研究の主眼は公式の係数を手計算むきに簡単にすることであった. コンピュータの出現は, この公式の発展に大きな影響を与え, 公式の係数の簡単化から合理化への道をとらせた. すなわち打切り誤差やまるめ誤差の観点から係数を最適化する方向に研究が進んだ. すなわち, Gill (1951), Merson (1957), Ralston (1962), Ceschino (1962), 田中正次 (1968), …等は, その流れをくむもので, このように公式の再検討が 1960 年代から活発になっていった. Kutta 型の公式は研究しつくされ

たように思われていたが，近年になって，また興味あるものが出てきた．たとえばいくつかの節点が一致した極限の公式で，達成不可能な公式として知られていた５段５位，６段６位の公式等である．これは数式処理システムの進歩で，Kutta の条件式を数式的に扱うことが可能になったお陰で導きだされた公式である．14)はその方向を示したものであり，今後“硬い方程式に対しても強力で，普通の方程式にも使える汎用で手間の少ない公式”を探す道具に数式処理システムがきっと役立つと思っている．

2.4 関数近似

実例 10 函数表から関数近似式へ

1940 年代にコンピュータが出現するまでは，卓上計算器や数表等を用いた手計算の時代であった．数表自身の作成も手計算で大へんな労力を必要とした．たとえば，林桂一著，ベルリン Julius Springer 発行の Fünfstellige Tafeln der Kreis-und Hyperbelfunktionen (1921) の苦心談は，同書の日本語版“円及雙曲線函数表 (1939) の諸言に記せられている．この時代は数表には誤まりのあるのが普通で，誤まりのないので有名なのが**バローの数表**であると学生のとき教えられたものである．

コンピュータの出現後は数表作りは簡単となり1940年から 1950 年にかけておびただしい数表のシリーズが刊行された．たとえばアメリカでは New Deal 政策にもとづく失業救済事業の一環として取り上げられたとも言われている (➡13))．しかし簡単に誰でもできるものは，価値も少ない，結局コンピュータの出現により数表の必要性が激減してしまったのはまことに皮肉な運命であった．ある関数の値を表でひいて（表にないときは補間して）求めるよりは，必要のときにコンピュータで直接計算する方が速いからである．

したがって，関数値の計算には，関数表よりそのもとになる近似公式の方が重要視され，1950 年代から 1960 年前半にかけてはコンピュータ用の近似公式の研究や，その公式集の作成が行われた．

1960年代におけるこの分野の参考書であり教科書でもあった，1963 年初版の一松信著 近似式 竹内書店の序文で近似式の効用について次のような**夢**が語られている：“現在函数計算として広く常用されているのは，適当な計算しやすい**近似式**を作っておいて必要の都度コンピュータ内部で計算する方法である．一見まだるっこいようであるが，コンピュータの高速度を思うとこれが最も早く最も能率がよい．大部の数表を作成するより各種の近似式とプログラムを完備させ，函数値が必要ならば，**函数**の**種類**と**引数**の値をボタンなりダイヤルで指示すれば，**コンピュータ**で**自動的**に**計算**して値をだしてくれるシステムを完備させる方がいい．”

この**夢**は，いまでは TSS 端末を電卓なみに用いて大型機で実現するのも簡単であるが，1970 年代に別のやり方で実用化された．すなわち：

実例 11 初等関数の統一的計算（数表の復活）

その**夢**が**初等関数**について，関数キーをもつ卓上計算機（電卓）で実現したのは今から 10 年位前になる．これは‘近似式’を用いたプログラムによる方法ではなく，CORDIC 等に見られるような“座標変換を反復して初等関数を統一的に計算する”という算法を実用化して達成されたわけである．

この方法について，1972 年の京大数理解析研の研究集会で一松先生が Walther (1971) の論文の紹介をされたとき，“この算法をみて大へんなショックを受け，近似式の研究はもう止めたくなってしまった”と前おきされたのを今でもよく覚えている．

この新しい算法は，ある程度の量の定数表をコンピュータの内部に記憶させておき，加法定理を用いて関数値を計算するときに，**この表を利用する**という方式と言える．関数計算に数表が追われて近似式が用いられ，また簡潔な形の数表が復活して利用される，……という歴史の流れが興味深い．ともかく，“驕る平家久しからず”で初等関数の近似式の価値は，1950 年代よりずっと落ちてしまったと言えるであろう．

この間の事情をコンピュータ関係史と並べて年表にしたのが**表-1** である．これについて少し補足する：

（1） 1955 年発行された C. Hastings 編 Approximations for digital Computers はコンピュータ用の関数近似のための min-max 近似公式集で，当時国産機で関数のサブルーチンの作成のためこの公式集がよく用いられた．これや，IBM の Journal 等に刺激されて自分自身でも公式を作り，アセンブラでサブルチンを作りその精度や速度を競い合った思い出がある．このときは多項式による近似が主であった．

（2） 1958 年頃の IBM 7090 FORTRANⅣ の関数副プログラムでは，すでに計算の手間や精度の点で多項式近似より優れている有理式による min-max 近似式が用いられていた．H. J. Maehly (1960) の UNESCO の論文で有理式近似をみたが，論文の形では，メーカの製品よりずっとおくれて公表されていた．この頃，この Maehly は C. Hastings の多項式

表-1 コンピュータ関連史と関数近似

	コンピュータ関連		関数近似関係の論文
	1952(昭27)	東大で TAC 委員会発足 ETL-MKI 完成（リレー式）	National Bureau of Standords, Applied Mathematics Series (AMS) 25, Tables of Bessel Functions $Y_0(x)$, $Y_1(x)$ $K_0(x)$, $K_1(x)$, $0 \leq x \leq 1$ (1952).
	1953(昭28)	IBM 650	(AMS) 26, Table of Arctan x (1953).
	1954(昭29)	パラメトロンの発明 PC 委員会発足 IBM 602A による20次の行列の逆転に成功	(AMS) 34, Table of Gamma Function for Complex Arguments (1954).
第1世代	1955(昭30)	FACOM 100 ETL・MKII 完成(リレー式) IBM 704, UNIVAC 1103	C. Hastings 編; Approximations for digital computers, Princeton, 1955.
	1956(昭31)	FUJIC 完成 ETL-MKIII 完成(トランジスタ式) 有隣電機 FACOM128（リレー式）で計算サービス開始	
	1957(昭32)	通研 M1, HIPAC 1 完成 IBM 704 用 FORTRAN 1 PC 委員会より CP 委員会となる	E.G. Kogbetliantz; Computation of e^N for $-\infty < N < +\infty$ using electronic computer, IBM Journ. 1, No. 2 (1957), p. 110-115. Y.L. Luke; On the computation of log x and arctan x, MTAC **11** (1957), pp. 16-18.
	1958(昭33)	PC1 運転開始（パラメトロン式） ETL-MKIV完成（トランジスタ式） IBM 650 入荷 IBM 7090 用 FORTRANIV	
	1959(昭34)	TAC 完成，(真空管式) 第1回国際情報処理会議に，NEAC 2201 と HIPAC 101 出品	E.G. Kogbetliantz; Computation of arctan N (arcsin N, sin N, cos N, $m\sqrt{N}$) using an electronic computer, IBM Journ. 2 (1958), pp. 43-53, 218-222, **3** (1959), pp. 147-152. 通研電子応用研究室「計算機を使った関数の多項式近似」経過資料第 818 号 (1959)，電電公社，通研．
	1960(昭35)	IBM 7090 輸入始まる 第1回プログラミング・シンポジウム（数理科学総合研究第IV班）	山内二郎「大型計算機用関数近似についての一つの試み」第1回プログラミング・シンポジウム報告集 (1960).
	1961(昭36)	PC2 完成(パラメトロン式) コンパイラ言語の成長と多様化モニタ OS の始まり	宇野利雄「関数近似の基礎理論」第2回プログラミング・シンポジウム報告集 (1961)， 山下真一郎「多項式による最良近似法の定式化」同上 (1961). 坪井定一「$\tan\frac{\pi}{4}x$ の分数式近似」同上 (1961).
第2世代			H.J. Maehly; Rational approximations for transcendental functions, UNESCO/NS/ICIP/A.I. 12 (1960), pp. 48-61. A.J.W. Duijvestijn-A.J. Dekkers; Chebyshev approximations of some transcendental functions for use in digital computing, Philips research reports **16** pp. 145-174(1961).
	1962(昭37)	電子計算機研究組合結成	宇野利雄「関数の近似公式」情報処理 3，2号 (1962)，pp. 90-94. 一松 信「関数近似」第4回電子計算機活用セミナー・テキスト (1962)，日科技連． 山内二郎「有理関数近似の一様最良化について」第3回プログラミング・シンポジウム報告集 (1962).
	1963(昭38)	UNICON (University Contribution.) の試用開始，本使用は翌年 IBM 7090 等の大型機の大学・研究機関計算機時間無料提供	宇野利雄・永坂秀子「ベッセル関数の計算法；漸化式における誤差の伝播」日本数学会講演 (1961年5月，10月)，および数学 **15**，1号 (1963). 一松 信「多変数関数の多項式近似」第4回プログラミング・シンポジウム報告集 (1963). 宇野利雄「ベッセル関数の近似公式」同上 (1963). 山内二郎「奇関数の一様最良近似多項式の折りたたみ計算法」同上 (1963)，および数学 **15**，1号 (1963).
			宇野利雄編『計算機のための函数近似公式集』数理科学総合研究第IV班第5分科会，1 (1961)，2 (1962)，3 (1963). Simauti, T (島内剛一); Approximation formulas for some elementary functions. Comm. Math. Univ. St. Pauli **12** pp. 23-35(1963). 一松 信，近似式，竹内書店 (1963).
	1964(昭39)	HITAC 5020 完成（トランジスタ式） IBM システム 360（第3世代）を発表	一松 信「関数近似について—特に近似式の選択について—第5回プログラミング・シンポジウム報告集 (1964). 二宮市三「一般の次数の Bessel 関数の一計算法」同上 (1964). 山内二郎「正規分布に関する近似関数」同上 (1964).
	1965(昭40)	東京大学大型計算センタ開所，翌年 HITAC 5020E で発足	桂 重俊・井上雄次「Legendre 函数の積の積分表」第6回プログラミング・シンポジウム報告集 (1965). 山内二郎「正規分布の百分率点の一次有理関数近似」同上 (1965). 戸田英雄「折りたたみ計算法を利用して函数近似公式を作るプログラムについて」同上 (1965).
	1966(昭41)	OS，TSS 等の本格化	山内二郎「折りたたみ計算法による一様最良化有理関数近似の例」第7回プログラミング・シンポジウム報告集 (1966). 戸田英雄「$\Gamma(2+x)$ の minmax 多項式近似」同上 (1966).
第3世代	1967(昭42)	FORTRAN，ALGOL の JIS 制定 大阪大学 NEAC 2200-500 で TSS 開始	二宮市三「平方根の有理関数近似」第8回プログラミング・シンポジウム報告集 (1967). 山内二郎・戸田英雄「級数の逆転の一般式とプログラム」同上 (1967).
	1968(昭43)	京都大学 FACOM230-60 で大型センタ開始	J.F. Hart 編，Computer Approximations, John Wiley & Sons (1968).
第4世代	1970(昭45)	IBM 370（第4世代）発表 マイクロ・プログラム方式による初等関数の計算（卓上計算機）	J.S. Walther, A unified algorithm for elementary functions, S.J.C.C. (1971). 一松 信著，初等関数の数値計算，教育出版社 (1974).
	1980(昭55)		

による近似式より優れた有理関数による近似公式集をIBM の大型機で作成中であると爆弾宣言したのでわが国のこの関係の研究者はおりてしまう者も少なくなかった．しかし公式集が実際に発刊されたのはずっとおくれて，1968年に Hart 編で Computer Approximations として出版された．アメリカのように厳しい競争社会では，おどして相手をおろすのも一つの戦略と思われる．おどしに乗らなかった人の 1968年までの研究の主なものを，プログラミング・シンポジウム報告集からリストして表にのせてみた．

（3） 1970年代マイクロ・プログラム方式で「初等関数の統一的計算法」による電卓が登場した．この辺のくわしいことは一松信[15]に解説されている．

3. 今後の数値計算

1940年代にコンピュータが生まれその後の進歩により従来の数値計算法は革命的な影響をうけたのは当然である．わが国のコンピュータの草分け時代の 1950年代から 1960年前半位までに焦点を合せながら，コンピュータのパイオニヤたちの主として数値計算の分野における努力の跡を見てきた．そこには，ハードウェア及びソフトウェアの発展に伴い，算法のいくたびかの栄枯盛衰の歴史が繰返されている．

この歴史の流れから，われわれは将来の数値計算への指針が与えられる筈である．その一つは，歴史の中で，パイオニヤたちの活動の過程と成果を観察すれば得られるものである．もう一つは，今後のコンピュータ・システムの動向から予測されるものである．このことは，歴史の流れから分かるように数値計算とコンピュータ・システムは強い正の相関があることから当然であろう．

その指針は，歴史を繙く人々の主観によって違う面もあるかも知れないが，筆者の主観的立場から述べると，「今後何年間，数値計算の分野では，並列計算が可能な算法の開発に迫られ，また数値計算の道具として，ますます便利となる図形処理システムと数式処理システムが活用されるようになる」と予想される．

参 考 文 献

1) 内藤 勝，増山元三郎，森口繁一共編：統計学へのいざない，東京大学出版会 (1959)．
2) 一松 信著：数値解析，朝倉書店 (1982)．
3) 高橋秀俊，石橋善弘：電子計算機によるExactな計算の新方法 (Modp 演算の応用)，情報処理，Vol. 1，No. 2，pp. 78-86 (1960)．
4) 平野菅保：代数方程式の解法および誤差，第8回プログラミング・シンポジウム報告集，情報処理学会 (1967)．
5) 室田一雄：平野の変形 Newton 法の大域的収束性，情報処理学会論文誌，Vol. 21，No. 6，pp. 469-474 (1980)．
6) 伊理正夫著：数値計算，朝倉書店 (1981)．
7) 小野令美：Durand-Kerner 法と Aberth 法を用いた超高次方程式の数値計算，情報処理学会論文誌，Vol. 20，No. 5，pp. 399-404 (1979)．
8) 小野令美：Durand-Kerner-Aberth 法を用いたある種の超高次方程式の解の数値計算，情報処理学会論文誌，Vol. 22，No. 2，pp. 165-168 (1981)．
9) 伊理正夫：常微分方程式の数値解法における不安定現象に対する対策一平滑子の理論，設計公式及び使用例一，情報処理，Vol. 4，pp. 249-260 (1963)．
10) 伊理正夫，松谷泰行：Runge-Kutta-Gill 法について，情報処理，Vol. 8，pp. 103-107 (1967)．
11) 清水辰次郎：常微分方程式の数値解法における数学的諸問題，第7回プログラミング・シンポジウム報告集，情報処理学会 (1966)．
12) 森口繁一，伊理正夫，小林光夫：常微分方程式の数値解法における難問対策へのある試み，第8回プログラミング・シンポジウム報告集，情報処理学会 (1967)．
13) 林 桂一著，森口繁一増補：高等函数表第2版，岩波書店 (1967)．
14) 戸田英雄：Runge-Kutta 系のある極限公式の打切り誤差について，電総研研究報告 (1977)．
15) 一松 信著：初等関数の数値計算，教育出版社 (1974)．

(昭和58年2月3日受付)