数理解析研究所講究録
第253巻 1975年 1-13

計算の手間について

京大 数理研　一松 信

0. はしがき

計算の手間 (computational complexity) の問題は，元来「計算の理論」の一分科として発生し，多くの断片的事実がえられている。それらは次第にまとまって，数値解析との境界領域に発展しつつある。この方面の研究の方向は(相互に密接な関連があるが)，大別して2つあるように見える。第1は，有限算法に対する演算回数の上限，下限および漸近式の研究であり，第2は本質的に無限回の演算を要する数値計算において，精度と手間の関係(与えられた精度での演算回数の上下限など)を考察する方面である。

数値解析としては後者の方向をおしすすめて，窮極的には自動的最適化をねらうべきであろう。しかしそのためには，特異点の情報などを，人間がはっきり把握し，計算機に教えてやる必要がありそうである。未知関数の高階導関数(の上限)など，未知で必ずしも1のオーダーとはいい難い量を定数に含むような評価式は，実用価値が乏しいという厳しい批判さえある。

ここでは数値解析からは多少それるが，主として前者の方

向での話題をいくつかとりあげたい。これらは1974年夏，講演者の出席したConference on Numerical Analysis (Dublin) とInternational Congress of Mathematicians (Vancouver) でのいくつかの話題の紹介である。([1]－[4])

1. 多項式の計算

多項式 $P(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_n$

の計算の手間は古くから研究されている。(Eve [6]，および Pan[7] の序文に当時までの歴史が詳しい)。Horner法でn回の乗法とn回の加法でできる(これが1つの上限)。以下おもな古典的結果:

$n \leqq 4$ までこれが最適: Ostrowski, 1954

$4 \leqq n \leqq 6$ 乗法 $[(n+3)/2]$，加法 $n+1$ 回で可能: Motzkin, 1955

(なお Knuth, 1962; Todd, 1955; Belaga, 1958)

$[(n+1)/2]$ 回の乗法は必要: Belaga, 1958.

複素数計算を許せば $[(n+3)/2]$ 回の乗法で十分: Belaga, 1958.

実数では $[(n+4)/2]$ 回の乗法で十分: Pan 1959 [5], Eve 1964 [6]

(ともに $n \leqq 11$ まで)

$n \leqq 12$ まで，ほとんどすべて(係数の空間の Lebesgue 測度0の集合を除いて)の多項式について，実数で $[(n+3)/2]$ 回の乗法で十分:

Lehman [4]. 1974.

(予稿には条件なしでよいように書いてあった。＿＿は講演をきき，また講演者に質問してたしかめた条件である。)

以上はいずれも2次式因子にまとめて計算する方式である。たとえば、$[(n+4)/2]$ 回の乗算でできるEveの算法は、

$$x = y + c$$

$$P_1 = \begin{cases} a_0 x^2 + a_1 x + \beta_0 & n = 2m+2 \text{ のとき} \\ a_0 x + a_1 & n = 2m+1 \text{ のとき} \end{cases}$$

$$P_{i+1} = (x^2 - \alpha_i) P_i + \beta_i \qquad (i = 1, 2, \cdots, m-1)$$

$$P(x) = (x^2 - \alpha_m) P_m$$

の形（定数は初めの多項式から定まる）である。{話定数 α_i, β_i は近似式のように定まった多項式については、あらかじめ計算しておけばよい。ただし数値解析的には、大きな桁数になったり、計算中に桁落ちを生じたりしやすいことが多いらしい。

しかし係数 $a_0, a_1, \cdots, a_n$ と x をデータとして、$P(x)$ を目標とすると、本質的にHorner法より手間の少い算法はないという([3]；定式化の詳細は不明)。ただし $P(x)$ を相異なる $n+1$ 個の点で、同時に計算するならば、$2n(n+1)$ 回ではなく、$O(n \log n)$ 回で計算できる算法がある（Fidvecia-Moench-Brodin-Cieveking, 1972）。その要点は、

$$P(x) = Q(x)(x-x_0)(x-x_1)\cdots(x-x_{n/2}) + R(x),$$

$$\deg R(x) \leqq n/2$$

と変形すれば、$P(x_i) = R(x_i)$ $(i = 0, 1, \cdots, n/2)$ であり、$R(x)$ は数式処理で求められるので、n に対する手間 γ_n は $\gamma_n \leqq 2\gamma_{n/2} +$（$n$によらない定数）という評価ができ、これ

を反復して，$\gamma_n = O(n \log n)$ もうる．

注意 1. 半分に分けて，$\gamma_n \leqq 2\gamma_{n/2} + c$ を導びくのは，$\gamma_n = O(n \log n)$ を示す常套手段である．じつさい，Paterson [2] は，もっと一般に，つぎの補題を示している．

パラメタ n に対する計算の手間の上限を $F(n)$ とするとき，

$$F(n) \leqq a\, F(\lceil n/b \rceil) + O(n^\beta)$$

（$\lceil\ \rceil$ は切り上げた整数部）

がえられたとする．$\alpha = \log_b a$ とおくとき，これから

$$F(n) \leqq \begin{cases} \alpha > \beta \text{ なら} & O(n^\alpha) \\ \alpha < \beta \text{ なら} & O(n^\beta) \\ \alpha = \beta \text{ なら} & O(n^\beta \log n) \end{cases}$$

がえられる．

注意 2. 高橋教授の FFT と，そのたたみこみや，Fresnel 変換への応用でもそうであるが，こういう手法は多数の目的データに対して，正直にやれば $O(n^2)$ の手数がかかるのを $O(n \log n)$ にへらすのが中心である．したがって 1 つの値だけを目的とする場合には，必ずしも「能率化」にはならない．

注意 3. 以上は乗法を一つの基本演算とする立場だが，マイクロプログラムの立場を借りれば，加法とシフトを基本演算として，$P(x)$ を計算する方法がえられる．たとえば，$\sqrt{x}$ の逆算法として x^2 をもとめ，$xy = [(x+y)^2 - (x-y)^2]/4$

として積を求める方法や，差分を重ねて $P(0)$ から始め，x の1ビット分ずつの増分を $P(x)$ に重ねてゆく算法などがあり，それらは実用になりうる（[10] 第4章参照）．その手間の評価なども興味ある課題であろう．

付記　講演当日高橋教授からの御注意を付記する：

実係数の多項式 $P(x)=a_0x^n+a_1x^{n-1}+\cdots+a_n$ の複素数 $x=\alpha+i\beta$ における値を計算するには，$\alpha+i\beta$ を根とする2次式 $q(x)=x^2-2\alpha x+(\alpha^2+\beta^2)$ による剰余 $r(x)=cx+d$ を求めて，$c(\alpha+i\beta)+d$ を計算すればすむ．この計算には Hitchcock-Baistow 法の計算が利用できるであろう．すなわち，$u=2\alpha$, $v=-(\alpha^2+\beta^2)$ として漸化式

$$b_0=a_0,\quad b_1=a_1+ub_0,\quad b_k=a_k+ub_{k-1}+vb_{k-2}$$

$(k=2,3,\cdots,n)$ により，剰余は $b_{n-1}(x-u)+b_n$ になり，値は $(b_n-\alpha b_{n-1})+i\beta b_{n-1}$ となる．乗法の総数は $2n+3$ 回である．—— 複素数の積を正直に求めると，毎回実数の積4回，したがってほぼ $4n$ 回の乗法がいる．

こういう「平凡」な工夫が，たいていの教科書に書いてなくて，「名人の秘伝」としてしか伝えられていないのは，残念なことである．

2. 行列算

n次正方行列A, Bの積$A \cdot B$を正直に$\sum_{j=1}^{n} a_{ij} b_{jk} = c_{ik}$として求めれば，全体で$n^3$回の乗算を要する．これはこれ以上簡単にしようもないように見えるが，Strassen [8]はこれが$O(n^{2.8})$（2.8は正確には$\log_2 7$）ですませれることを示して，大きなショックを与えた．実用上では，これは再帰的な算法を含んでいてプログラム作りが厄介であるのと，数値解析的に難があることから問題が多いが，理論上では，この分野に新生面を開き，意外な所にまで深刻な影響をもたらしたので，この解説は逡するわけにはゆかない．

$n = 2m$とし，n次正方行列A, Bおよびその積$C = AB$を半分ずつに分けて，次のようにおく．

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

このときCはつぎの計算で求められる．

$$\mathrm{I} = (A_{11} + A_{22})(B_{11} + B_{22}) \qquad \mathrm{II} = (A_{21} + A_{22}) B_{11}$$

$$\mathrm{III} = A_{11}(B_{12} - B_{22}) \qquad \mathrm{IV} = A_{22}(B_{21} - B_{11})$$

$$\mathrm{V} = (A_{11} + A_{12}) B_{22} \qquad \mathrm{VI} = (A_{21} - A_{11})(B_{11} + B_{12})$$

$$\mathrm{VII} = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$C_{11} = \mathrm{I} + \mathrm{IV} - \mathrm{V} + \mathrm{VII} \qquad C_{12} = \mathrm{III} + \mathrm{V}$$

$$C_{21} = \mathrm{II} + \mathrm{IV} \qquad C_{22} = \mathrm{I} + \mathrm{III} - \mathrm{II} + \mathrm{VI}$$

証明は式を書き下せばよい．Paterson [2] が実演してみせたように，透明原紙に色分けして重ねてみるとよくわかる．当日実演してみたが，概略を記述する．

×	B_{11}	B_{12}	B_{21}	B_{22}
A_{11}		$+_{III}$		$-_{III}\ +_{V}$
A_{12}				$+_{V}$
A_{21}	$+_{II}$			
A_{22}	$+_{II}\ -_{IV}$		$+_{IV}$	

左の A_{ij} と右の B_{kl} との交点は積 $A_{ij}B_{kl}$ を示す

この和 $C_{12} = A_{11}B_{12} + A_{12}B_{22}$

この和 $C_{21} = A_{21}B_{11} + A_{22}B_{21}$

×	B_{11}	B_{12}	B_{21}	B_{22}
A_{11}	$+_{I}$			$+_{I}\ -_{V}$
A_{12}			$+_{VII}$	$+_{VII}\ -_{V}$
A_{21}				
A_{22}	$+_{I}\ -_{IV}$		$+_{IV}\ -_{VII}$	$+_{I}\ -_{VII}$

これらの和 $A_{11}B_{11} + A_{12}B_{21} = C_{11}$

$C_{22} = A_{21}B_{12} + A_{22}B_{22}$ も同様である．このように同じ所をたして引くので，もしこれらの積に大きな値が現われると，情報落ちで $(\alpha+\beta)-\beta \neq \alpha$（たいていは0または α の下位が失なわれた値になる）となるため，数値解析的な困難を生ずる．しかし原理的にはこれでよいはずである．

これにより，積の手間 $P(n)$ については評価

$$P(n) \leqq 7\,P(\lceil n/2 \rceil) + O(n^2)$$

をうる．末尾の項は，和の手間などから生ずる．前の補題から，$P(n)=O(n^{\log_2 7})$ をうる．

A および A_{11} が可逆ならば，$A^{-1}=D$ を上記のように分け，$E=A_{22}-A_{21}A_{11}^{-1}A_{12}$ とおき，これも可逆と仮定すると

$$D_{11}=A_{11}^{-1}+A_{11}^{-1}A_{12}E^{-1}A_{21}A_{11}^{-1} \qquad D_{12}=-A_{11}^{-1}A_{12}E^{-1}$$

$$D_{21}=-E^{-1}A_{21}A_{11}^{-1} \qquad D_{22}=E^{-1}$$

であり，A_{11}^{-1} と E^{-1} を求めれば $A^{-1}=D$ がえられる．したがって逆行列を求める手間については，$I(n)$ とすると

$$I(n)\leqq 2I(\lceil n/2\rceil)+O(P(n))+O(n^2)$$

がえられ，前の補題から

$$I(n)=O(P(n))=O(n^{\log_2 7})$$

をうる．ただし O の係数は相当に大きいので，実際例では必ずしも掃き出し法 (=Gauss-Jordan 式消去法) による $n^3/2$ よりも有利かどうかは問題である．

一方 A,B を n 次正方行列とし，I を同じ大きさの単位行列とすると，

$$\begin{bmatrix} I & A & O \\ O & I & B \\ O & O & I \end{bmatrix}^{-1}=\begin{bmatrix} I & -A & A\cdot B \\ O & I & -B \\ O & O & I \end{bmatrix}$$

であるから，これから $P(n)\leqq I(3n)$ をうる．したがって $I(n)=O(n^\alpha)$ である算法があれば，$P(n)=O(n^\alpha)$

となり，両者はnの同じオーダーでえられる．

実際には，本来の行列計算で $A\cdot B$ をわざわざ3倍の大きさの上三角行列の逆行列として計算するのはナンセンスに近いけれども，この関係式は，次節にのべる他の変形では重要な意味をもつし，何よりも $P(n)$ と $I(n)$ とが同じnのオーダーの手間を要するという注意は本質的である．

付記　次節にのべるように $P(n)$，$I(n)$ のオーダー $O(n^\alpha)$ の α を $\log_2 7 = 2.805\cdots$ よりもさらに下げられれば，影響する所は大きい．しかしてのような算法はなかなかえられない．Aを3等分していろいろ組合せてみたが，思わしい公式（すなわち $\log_3 a < \log_2 7$ となる算法）はえられなかったという報告もある（東大工・計数工での研究）．めのこでは無理であろうが，そういう算法は「偶然に」みつけるしかないものだろうか？　これ以下にはできないという下限は，どうやって評価できるものなのだろうか？

3. 行列算の変形

行列の積および逆と似た算法で計算できる対象はいろいろある．以下にいくつかの例をあげる．

0，1 を成分とし，行列の積で×（積）を∧，＋（和）を∨でおきかえて結合するとき Boole 行列 という．それらの積は

正直にやれば $O(n^3)$ の手間がかかるはずであるが，やはり $O(n^{\log_2 7})$ の手間で可能である．それには

$A^0 = \overbrace{I}^{(\text{全部}1)}$, $A^{m+1} = A \wedge A^m$ $(m = 0, 1, 2, \cdots)$, $A^* = \bigvee_{m \geq 0} A^m$

とおくと，$A^* = D$ を二分した行列は，ちょうど $*$ を -1，$\wedge$ を $\vee$ におきかえた上記の逆行列の公式 D_{ij} と同じ式をみたすことが示される．したがって上記の逆行列の算法で A^* を求め，$P(n) \leqq I(3n)$ とした注意を活用すればよい．

なお $\times$ のかわりに $\equiv$（同値；二進和の否定），$+$ のかわりに $\wedge$ をとったとき，および $\times$ のかわりに $\wedge$，$+$ のかわりに $\oplus$（二進 Exclusive or，同値の否定）をとったときも，同様の議論が成りたつ（Paterson [2]）．

他の変形として，$R_+ \cup \{\infty\}$ を要素とする行列で，$\times$ のかわりに $+$，$+$ のかわりに $\min$ をとった積は，a_{ij} を i 番目の頂点から j 番目の頂点への距離とした有向グラフの最短路問題に現われる．この演算も，上と同様の注意により，$O(n^3)$ でなく，$O(n^{2.8})$ の手間で可能である．

もっとかわった応用として，生成規則が $A_i \longrightarrow A_j A_k$ という形のみの（他に端末記号への おきかえを含む）文脈自由言語において，要素を語の集合とし，$+$ を集合の合併に，積を

$$S_1 \otimes S_2 = \{ A_i \mid A_j \in S_1,\ A_k \in S_2;\ A_i \longrightarrow A_j A_k \}$$

とした場合が考えられる．この積 $\otimes$ は一般に結合法則をみ

たさない（反例はすぐに作られる）。それにもかかわらず、このような成分の上三角行列（対角線以下の成分はすべて空集合）に対して、一種の「逆」行列が作られ、その計算および「行列」の積が、同じく $O(n^{2.8})$ の手間で作られる。

最後の例については、かなり独自の修正を要するし、またこれが直接 Parsing の手間の評価とどう結びつくのかはっきりしない。しかし、この講演（Paterson [2]）をきいて強く感じたのは、行列算のアナロジーはいたる所にあり、ある一つの個所での新算法は、思いもかけない所にまで波及するという点であった。Strassen の算法 [8] ただ1つが、これほど多くの影響を与えることは予想外であり、何が何と結びつくかわからない所に、無類のおもしろさを感じた次第である。

4. 解析的計算の手間

最後に Traub [1] の話の要点をメモしておく。Traub は反復法 ϕ（たとえば Newton 法）により、$x_i \to \alpha$ となる列を考え、$x_{i+1} - \alpha = O((x_i - \alpha)^p)$ $(p > 1)$ であるとき、$(\log_2 p)/C$ （C はある定数）をもって反復 ϕ の能率 $e(\phi)$ と定義する。これは「能率の公理」（誤差に反比例し、自己合成 $\phi \circ \phi$ について不変）をみたす。これをもとに

して，たとえば f の $n-1$ 次導関数 $f, f', \cdots, f^{(n-1)}$ まで必要とする反復について，その能率を上から評価し，最適の n を求める．—— 多くの例では n は 2 か 3 であり，4 以上のことは珍らしいらしい．

これはおもしろい着眼であり，今後の発展が期待できる．しかしこの「能率」の定義は，多くの反復法がそうであるのだが，線型収束 ($p=1$) の場合にはまったく使えない．(始めから 1 乗以上の収束を念頭においている) という強い批判がある。1 乗収束の場合をも含むように理論を修正することもされているが (古部教授の研究がある)，なお今後にまつ所が大きいと思う．

参考文献

[1] J.F.Traub, Recent results and open problems in analytic computational complexity, Conf. on Num. Anal. lecture, Dublin, 1974 Aug. 2.

[2] M.Paterson, Complexity of matrix production, ICM Vancouver, invited lecture, 1974 Aug. 23.

[3] V.Strassen, The computational complexity of representations of polynomials and rational functions, ICM Vancouver, invited lecture, 1974 Aug. 24.

[4] R.S.Lehman, Evaluation of all polynomials, ICM Vancouver, short comm., Aug. 28. 1974

[5] V.Ja. Pan, Schemes for the calculation of polynomials with real coefficients, Dok. Akad. Nauk SSSR 127 (1959), 266.

[6] J.Eve, The evaluation of polynomials, Num.Math. 6(1964) 17-21.

[7] V.Ja. Pan, Methods of computing values of polynomials (英訳), Russian Math. Surveys 21 (1966), 105-136.

[8] V.Strassen, Gaussian elimination is not optimal, Num. Math., 13 (1969), 354-356.

[9] 伊理正夫, 数値計算の手順について, 数理科学, 1970年9月号, 21－24.

[10] 一松 信, 初等関数の数値計算, 教育出版(HITシリーズ) 1974年11月