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†

小　野　令　美††

Chebyshev の数値積分公式の分点を与える高次方程式を全根同時反復型解法の Durand-Kerner-Aberth 法を用いて1024次までのものについて解いた報告をさきに行ったが[7]，このたび CRAY 1 を使用する機会を得たので，この計算機で2048次のものを解いた．この結果 $n\to\infty$ のときこの方程式の根が並ぶと予測されている曲線に近づくようすが数値的によくわかる．この結果について追加報告をする．

さらに CRAY 1 は速さの面では勿論，指数部 15 bit 仮数部 43 bit であることが，この種の数値計算に対して非常に効果的で，これらについて述べる．

1. まえがき

Chebyshev の数値積分公式

$$\int_{-1}^{1} f(x)dx \approx \frac{2}{n}\sum_{i=1}^{n} f(x_i) \qquad (1)$$

の分点を零点に持つ多項式を $F_n(z)$ とおくと

$$F_n(z)=a_0z^n+a_2z^{n-2}+\cdots+\begin{cases} a_{n-1}z & (n:\text{奇数}) \\ a_n & (n:\text{偶数}) \end{cases} \qquad (2)$$

$$a_0=1,\ a_{2k}=-\frac{n}{2k}\sum_{j=1}^{k}\frac{1}{2j+1}a_{2(k-j)},\ k=1,2,3\cdots \qquad (3)$$

である．$n\geqq 10$ ではこの多項式の零点は実数だけとはならないことはよく知られているが，$n\to\infty$ のときこの零点は曲線

$$|(z+1)^{(z+1)/2}(z-1)^{-(z-1)/2}|=2 \qquad (4)$$

の上に密に並ぶという予想が森口・伊理によってたてられている[5]．これを Durand-Kerner-Aberth 法の2次法 Jacobi 型[3]

$$z_k^{(\nu+1)}=z_k^{(\nu)}+\delta_k^{(\nu)}, \qquad (5)$$

$$\delta_k^{(\nu)}=-P_n(z_k^{(\nu)})/\prod_{j=1,j\neq k}^{n}(z_k^{(\nu)}-z_j^{(\nu)}) \qquad (6)$$

を用いて数値実験的に確かめることを試みた，ここで $p_n(z)$ は零点を求めようとしている多項式，$z_k^{(\nu)}$ は $P_n(z)=0$ の根 $\alpha_k,\ k=1,2,\cdots,n,$ の第 ν 近似である．

n が大きくなると係数の計算，反復計算共に大きな桁落ちが起り多倍長計算が必要となり時間がかかる．そこでこの問題の特殊性を利用して $n=4m$，(m は整数）のものについて，(i) $w=z^2$ とおき $n/2$ 次の多項式とし，(ii) よい初期値を選びはじめの $n/4$ 個の中に共役な零点に収束するものがないようにしてこの $n/4$ 個だけを計算し，また桁落ちの起る (6) 式の分子だけを多倍長計算で行いあとは全部単精度で行う等の工夫により，$n=2^m, m\leqq 10$ のものについての数値解を得ることができて，これについてはすでに報告した[7]．その時使用した計算機ではメモリの面でも計算時間の面でも $n=2^{10}$ が限度であった．

ところがメモリも大きく速さも約10倍という計算機 CRAY 1 を使用する機会が与えられたので（→謝辞），この計算機で2048次のものを解くことを試みた．この結果と，CRAY 1 の使用経験について述べる．

2. $F_n(z)=0, n=2048$ の根の数値計算

低次のところで経験的に反復計算では約 $n/5$ 桁有効精度が失われ，係数の計算でも約 $n/5$ 桁の桁落ちが起ることがわかった．この法則を適用すると計算桁数は安全側で見積って，反復計算で424桁程度，この反復計算に用いる424桁の正しい係数を求めるのに840桁程度の計算が必要となる．

2.1 多倍長ルーチン

CRAY 1 では1語が 64 bit で，そのうち 15 bit が指数部，48 bit が係数部になっている．また演算の速さは配列に入れた8000個の数の乗除を DO 文で行った場合8桁の整数を倍精度実数におきかえて行えば

† On Numerical Computation of a High Degree Polynomial Equation Using the Method of Durand, Kerner and Aberth by HARUMI ONO (Tokyo Metropolitan Nogei Agricultural Upper Secondary School).

†† 東京都立農芸高等学校

表 1 多倍長計算の比較

Table 1 Comparison of multiple precision arithmetic.

	n		6 桁多倍長 (CRAY 1)	8 桁多倍長 CRAY 1	8 桁多倍長 M 180
係数の計算	512	計算桁数	228 桁, 配例 38	224 桁, 配列 28	
		C.P.U. time	11.7 sec	36 sec	61 sec
	2048	計算桁数	840 桁, 配列 140	840 桁, 配列 105	
		C.P.U. time	457 sec	—	2436 sec
		計算に必要なメモリ	256 kW	—	224 kW
反復計算	2048	計算桁数	114 桁, 配列 19	112 桁, 配列 14	
		C.P.U. time	1121 sec	—	7942 sec
		計算に必要なメモリ	122 kW	—	128 kW
		その他	—	—	倍精度複素数, スケーリング使用

それぞれ 33 msec, 100 msec で FACOM M 180 の 52 msec, 94 msec, IBM の 80 msec, 90 msec と三機種とも大して変わらないが, 積の整数が 48 bit 内に収まるような 6 桁の整数を実数におきかえた乗除ならば CRAY 1 ではベクトル化演算でそれぞれ 0.53 msec, 0.89 msec と非常に速い. そこで多倍長の数を一つの配列要素に 10 進表現で 6 桁ずつの整数で入れることにすれば, これまでに用いていた一つの配列要素に 8 桁ずつ入れる方式[6] を 6 桁ずつに変えたために配列の数は増すが, それでもなおずっと速くなると期待できる.

さらに, 係数を求める多倍長の乗除は, 多倍長数×2048 の乗法と多倍長数÷(6〜4196352 の整数) の除法なのでこの計算専用のルーチンを作った. これにより計算時間をほぼ 1/2 に減らすことができた ($n=512$ のとき FACOM M 180 で 116 秒が 61 秒になった). M 180, CRAY 1 で 8 桁多倍長を用いた場合, CRAY 1 で 6 桁多倍長を用いた場合の比較を**表 1** に示す.

2.2 $F_{2048}(z)$ の零点と係数のようす

予想の曲線上に

$$(z+1)^{(z+1)/2}(z-1)^{-(z-1)/2}=2e^{i\theta}$$

$$\theta=\frac{2\pi}{n}k,\ k=0, \cdots, n-1 \qquad (7)$$

をみたす点 z_k, $k=0, \cdots, n-1$, を求め, この点とすでに求められている $n/4$, $n/2$ の数値解を用いて $\log n/n$ の 2 次式で軸上の零点の予測値を求め, 残りはこれらの値と $n/2$ の数値解と (7) 式の値を用いて線形補間

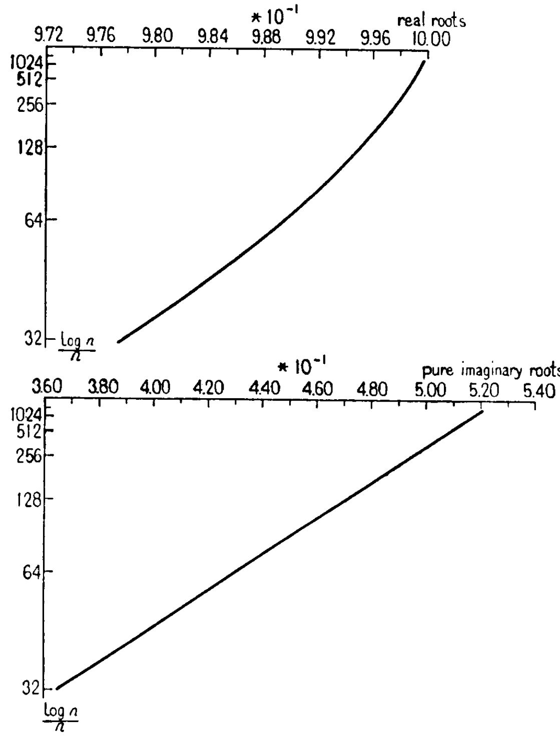

図 1 実根および純虚根と $\log n/n$ との関係

Fig. 1 Relation between real roots and $\log n/n$, and between pure imaginary roots and $\log n/n$.

表 2 各反復での Gerschgorin circle 半径 γ と C. P. U. time

Table 2 The radius of Gerschgorin circle and C. P. U. time in each iteration.

反復	γ_{max}	大部分の γ_k	時間
第 1 回 (512* 個)	.2568	$.3_{10}-1$〜$.8_{10}-1$	1121 sec
第 2 回 (32** 個 $k=2$〜33)	$.2611_{10}-1$	—	67 sec
第 3 回 (512 個)	$.8578_{10}-2$	$.2_{10}-2$〜$.5_{10}-2$	1509*** sec
第 4 回 (512 個)	$.1086_{10}-3$	$.4_{10}-5$〜$.1_{10}-4$	1121 sec

* $n/4=512$
** 実軸に近い 32 個の根についての部分反復
*** 3.4 で述べる

して初期値とする. 実根の場合の予測値は 0.99972, 数値解は 0.999731373, 純虚根の予測値 0.52045, 数値解 0.520567877 で非常によい近似値となっている. このことから特に実根の場合 $n\to\infty$ で $\log n/n$ の 2 次式で近似できるのではなかろうかと思われる. $\log n/n$ と実根および純虚根の関係を**図 1** に, 実軸の近くのようすを**図 2** に示す. 最も近接している根の距離は (z^2 のままで) $1.49_{10}-3$ である. したがって Gerschgorin circle 半径[3] γ がこの距離の 1/2 より小さけれ

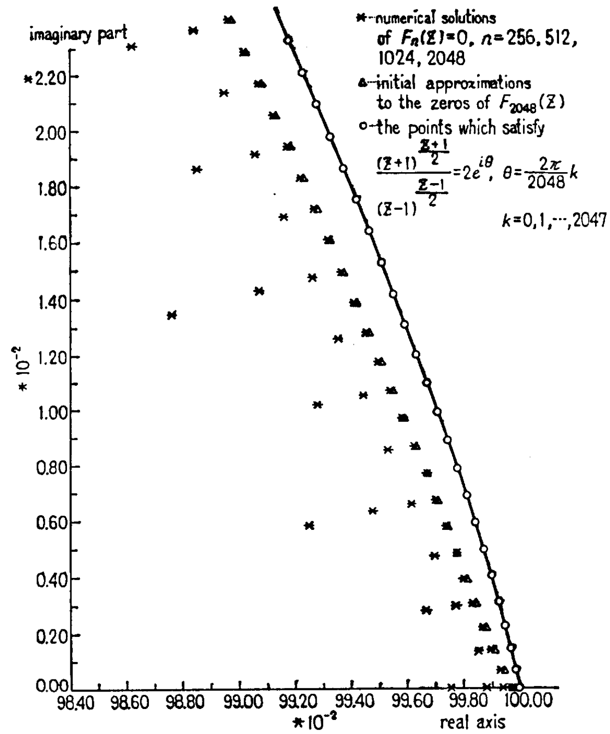

図 2 実軸に近いところ

Fig. 2 Zeros of $F_n(z)$ near the real axis.

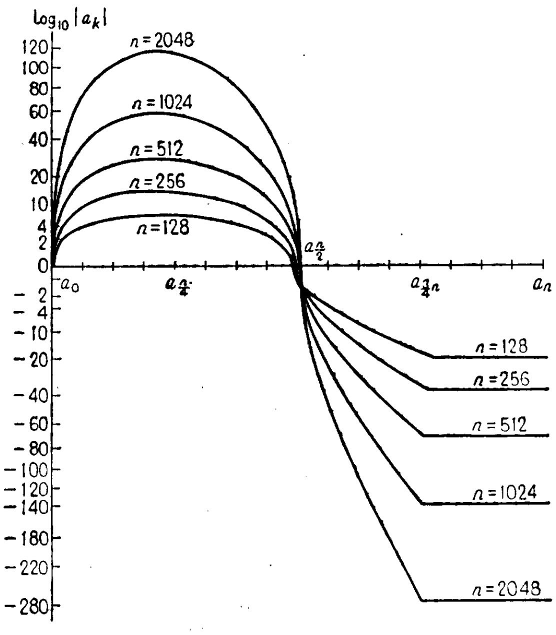

図 3 係数の変化のようす

Fig. 3 Coefficients a_k

ばよい．各反復における Gerschgorin circle 半径，計算時間などを表2に示す．

また(3)式で与えられる係数については $n \geqq 10$ では同じような傾向を示している．すなわち $a_{n/4}$ の近くで絶対値最大となり $a_{n/2}$ 近くでほぼ1のオーダ，$a_{3n/4}$ 近くまで減少して絶対値が最小に近い値を示し，そこから a_n まではオーダとしてはほとんど変化がなくわずかに減少する，そして $a_{3n/4}$ の近くまでは符号は正負交互でありそれから先はすべて負である．このようすを図3に示す．

3. 計算上の知見

3.1 計算桁数と仮数部の大きさ

2.2 で述べたように Gerschgorin circle 半径 $0.7_{10}-3$ を目標としているので修正量 $\delta_k^{(\nu)}$ は $0.7_{10}{-3}/2048 \approx 10^{-7}$ 程度である．したがって1ワード 32 bit で**仮数部 24 bit 16進6桁 (10進7.2桁) の計算機では精度が足りず**倍精度複素計算を用いなければならない．その点 CRAY1 では実部虚部共に**仮数部 48 bit 2進48桁 (10進14.4桁)** あるので単精度で十分である．

3.2 修正量のオーダと指数部の大きさ

各反復での修正量(6)の分子 $F(w_k^{(\nu)})$ は第1回目の反復で $10^{-275}\sim10^{-277}$，分母 $\prod_{j=1, j\neq k}^{n}(w_k^{(\nu)}-w_j^{(\nu)})$ は $10^{-270}\sim10^{-273}$ で修正量 $\delta_k^{(\nu)}$ は $10^{-4}\sim10^{-5}$ の数である，ここで $w=z^2$ である．したがって分子と分母は指数部 7 bit では実現できない．そのため分母の差積の部分は各因数ごとにスケーリングの定数を掛ければよいがこの因数を掛ける順にまた技巧が要る．隣りのものとの差から順に掛ければアンダフローを起し，原点に関して対称なものとの差から順に掛けるとオーバフローを起す．結局原点に関して対称なものとの差の次に隣りとの差を掛け，これを順にずらしてゆくという方法で求めることができたが，さらに高次のものになればこれよりももっと技巧が必要となろう．また分子は多倍長で計算するのでアンダフローはしないが，これを倍精度複素数におきかえるところでスケーリングの定数の $(n/2-1)$ 乗が倍精度複素数で表現できず4回に分けて掛けなければならない．ところが，CRAY 1 では指数部が 15 bit あるので**スケーリングの必要がない**．$F_n(w)$ の場合には根の予測がついているのでスケーリングでうまくいくが，全く予測のつかない問題の場合にはだめで，指数部が 15 bit あ

る CRAY 1 はこの点でもこの種の科学技術計算には非常に強力である.

3.3 計算の速さ

表1からわかるように 32 bit 用の多倍長のルーチンをそのまま用いた係数の計算でも CRAY 1 は M 180 の約2倍速く，CRAY 1 用に6桁多倍長に直したもので約5.3倍，反復計算では倍精度複素数を用いなくてもよいこともあって約7倍の速さである．計算のすべてがベクトル化できる配列要素での演算ばかりではないから，**2.1** に述べた 100 倍程の差とはならないのは当然である．そのような最適化がほとんど速さに影響しないと思われる初期値の計算では(7)式を用いるので M 180 の 1.7 秒に対し 0.43 秒と約4倍であった．この計算は複素数の指数関数や対数関数など基本外部関数の計算の手間が相当に含まれているものである．

3.4 割り算について

高精度計算を行うための多倍長ルーチンでは整数の乗除で正確な積と正確な整数の商と剰余を求めなければならない[6]．CRAY 1 では整数の除法を実数におきかえて実行すると速くなるが答が正確に得られない場合がある．たとえば 5.0/5.0 は内部表現で最後の 1 bit がちがって正確に1とならず，整数に直すと0になる．そこで何らかの方法でこれを防がなければならない．M 180 では実数でも倍精度実数でもこのようなことは起らず16進表現で正確な値となっている．

表2で第1,2,4回目の反復では，この割り算の不正確さを防ぐため FORTRAN 文で余りをチェックする方法をとった．*** 印の第3回目の反復では FORTRAN 文で余りをチェックするよりは整数のままで割る方が速いであろうと，ある一つのサブルーチンの一箇所を直しただけであるが，結果は表の通りで，実数の割り算におきかえた方が，FORTRAN 文で余りのチェックを入れてもなお整数の割り算より約1.4倍速かった．

4. むすび

すでに求められている低次の n についての数値解から軸上の零点の予測値を求めたが，$\log n/n$ の2次式が特に実軸の方でかなりよい近似を与えている．ほかの点の初期値は $n=1024$ の数値解と(7)式との線形補間で求めたが，これはやはり $n/2$, $n/4$ の数値解と(7)式の極限の曲線上の対応する点を用いた $\log n/n$ の2次式で補間すべきであったと思う．そうすれば第2回目の部分的な反復をしないでも済んだのではなかったかと思われる．

今回の数値計算に CRAY 1 を用いて特に感じたことは，指数部の大きいことがこのような数値計算では非常に有効なことである．整数の割り算を実数計算にしたため起る最後の 1 bit の不正確さを防ぐには適当な小さな数を加える方法にするほうが速かったであろうと思われる．しかし CRAY 1 にオプションででも割り切れる整数の割り算を実数型の数で行ったとき商が正しく求められる機能があればさらに速いであろう．

謝辞 本研究は三菱総合研究所の CRAY 1 を使用して行ったもので，東京大学教授伊理正夫博士よりその機会を与えられたものである．さらに博士には，この研究に関し種々ご懇切なるご教示を賜わった．また機械の使用に当り三菱総合研究所反町洋一部長，細谷葉子，山本諭の諸氏にはいろいろと便宜をおはかりいただいた．また電子技術総合研究所戸田英雄博士には絶大なるご援助を賜わった．ここに深く感謝の意を表する．

参考文献

1) Kerner, I. O.: Ein Gesamtschrittenverfahren zur Berechnung der Nullstellen von Polynomen, Numer. Math., Bd. 8 pp. 290-294 (1966).
2) Smith, B. T.: Error Bounds for Zeros of a Polynomial based upon Gerschgorin's Theorems, J. ACM., Vol. 17, pp. 661-674 (1970).
3) 山本哲郎：ある代数方程式解法と解の事後評価法, 数理科学, Vol. 14, No. 7, pp. 52-57 (1976).
4) 山本哲郎，古金卯太郎，野倉久美：代数方程式を解く Durand-Kerner 法と Aberth 法，情報処理，Vol. 18, No. 6, pp. 566-571 (1977).
5) 伊理正夫，山下 浩，寺野隆雄，小野令美：大域的収束性をもつ代数方程式の解法，京都大学数理解析研究所講究録 339, pp. 43-69 (1978).
6) 戸田英雄，小野令美：高精度計算用の除算のアルゴリズムに関して，電子技術総合研究所彙報，Vol. 42, No. 4, pp. 66-71 (1978).
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).

(昭和55年9月12日受付)
(昭和55年10月23日採録)

訂　正

Vol. 22, No. 2, pp. 165～168 掲載の小野令美の論文「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」の梗概に次の訂正があります.

p. 165 上から5行目

誤：仮数部 43 bit　　正：仮数部 48 bit