

多項式のゼロ点を求める場合の計算量について

日本大学農獣医学部 五十嵐正夫 (Igarashi masao)

1. はじめに

この小論では、多項式の多重ゼロ点を色々な収束次数の Newton-Raphson 法で求めるとき、収束するまでの反復回数、解法の効率、厳密解への数値的接近の様子について考察してみる。

反復回数について得られた知見は二つである。一つは、計算桁数を固定した場合、ゼロ点の多重度がいくら増加しても、数値解が厳密解に収束するまでの反復回数はある一定の値に収束するということである。これは平野・伊理^[9]の二次収束 Newton-Raphson 法に対する結果の一般化である。

他の一つは、高次収束の Newton-Raphson 法ではゼロ点の多重度が増加するに従い、収束するまでの反復回数が減少すると言う結果である。これは 2 次収束の Newton-Raphson 法についての良く知られた常識、”ゼロ点の多重度が増加すれば収束するまでの反復回数は増加する”，と逆の結果である。

Newton-Raphson 系解法の効率を計る尺度については、Ehrmann^[3]の導入した尺度が従来良く使われてきたようである。Collatz^[2]も彼の尺度を紹介しているし、Pomentail^[6]は修正高次 Newton-Raphson 系解法の効率を計るのに彼の尺度を利用している。その尺度の基本は

$$(1) \quad \frac{\log_e(\text{収束次数})}{\text{例えば積の演算時間によって基準化された演算時間}}$$

である。我々の数値実験においては、多項式の次数が高くなると、この尺度は実際の数値結果に適合しなくなる傾向が見られた。基本的には、当時の計算機 (1957 年) に対して、数値実験的に得られた結果を、現在の計算機にそのまま適用するのは、無理な点があると

理解できる。特に分子の $\log_e(\text{収束次数})$ はその根拠が曖昧なようである。そこでここで
は次のような尺度を導入することにする。

$$\text{解法の効率} = \frac{1 \text{ 反復あたりの解への接近度合い}}{1 \text{ 反復あたりの計算手間}}$$

この尺度の意味付けは容易である。接近する度合いが大きければ解法は効率的になるし、
計算手間が少なければ少ないほど解法は効率的になるからである。上のような、尺度を導入
することにより、高次多項式 (n 次多項式) には高次収束公式 ($n/2$ 次程度の公式) が
良いと言う、実際の数値実験と良く一致する結果が得られる。

数値解の多重ゼロ点への数値的接近の様子であるが、2 次収束する Newton-Raphson
法については、1966 年の Rall^[7] の先行研究がある。彼は平均値の定理を何度も利用する
ことにより、数値解が m 重解に接近する場合の補正項の比が $1 - 1/m$ に収束することを示
している。ところで、彼の結果を k 次収束方法まで一般化すると、2 次収束の場合では、
気づきにくい数値的接近の様子が明かとなってくる。その一つは Newton-Raphson 法は
線形収束と非線形収束の組み合わせである、ということである。非常に妙な言い方である
が、数値例を見れば納得して頂けると思う。

議論は以下の順に従ってすすめることにする。

2. 数値解の精度桁数限界
3. 多重ゼロ点に収束するまでの反復回数
4. 解法の効率を計る尺度
5. 補正值の比について
6. おわりに

2. 数値解の精度桁数限界

与えられた多項式を

$$f(z) = c_1 z^n + c_2 z^{n-1} + \dots + c_n z + c_{n+1}$$

表1 k 次 Newton-Raphson 法のプログラム

```

SUBROUTINE LOCAL(C,N,K,Zold,Znew)
  COMPLEX*16 A(101),C(101),F(101),H(101),Zold,Znew,W
  DO 10 I=1,N+1
    F(I)=C(I)
  10 CONTINUE
  DO 20 I=2,N
    F(1)=C(1)+F(1)*Zold
  20 CONTINUE
  DO 25 J=2,N+2-1
    IF(J.LE.K) F(J)=F(J-1)+F(J)*Zold
  25 CONTINUE
  20 CONTINUE
  F(1)=C(N+1)+F(1)*Zold
  A(1)=(1.0D00,0.0D00)
  DO 30 I=2,K-1
    A(I)=F(I+1)/F(2)
  30 CONTINUE
  H(1)=-F(1)/F(2)
  DO 40 I=2,K-1
    W=A(I)
    DO 50 J=2,I
      W=W*H(J-1)+A(I-J+1)
    50 CONTINUE
    H(I)=H(1)/W
  40 CONTINUE
  Znew=Zold+H(K-1)
  RETURN

```

とする。 $f^{(k)}(z)$ は $f(z)$ の k 階の導関数を表し、 $z_l, l = 0, 1, 2, 3, \dots$ は z_0 を初期値とし反復解法によって逐次得られる近似解とする。高次収束公式を簡単に表すために

$$u(z) \equiv \frac{f(z)}{f'(z)}, \quad A_k(z) \equiv \frac{f^{(k)}(z)}{k!f'(z)} \quad k = 1, 2, 3, \dots$$

の記号を導入し、混乱がない場合は変数 z は省略する。また例えば $z' = z - f(z)/f'(z)$ において $-f(z)/f'(z)$ の項を Newton-Raphson 法の補正項とよぶことにし、 $\Phi_2(z)$ とおくことにする。すると $k+1$ 次収束する Newton-Raphson 法の補正項 $\Phi_{k+1}(z)$ は再帰的に次のように表せる。

$$\Phi_{k+1}(z) = \frac{-u}{(\dots(A_k h_1 + A_{k-1})h_2 + \dots + A_2)h_{k-1} + A_1}$$

いま、 $f(z)$ を $f(z) = (z - \alpha)g(z)$ とし、 $k+1$ 次の反復式

$$(2) \quad z_l = z_{l-1} + \Phi_{k+1} = z_{l-1} - \frac{u}{(\dots(A_k h_1 + A_{k-1})h_2 + \dots + A_2)h_{k-1} + A_1}$$

によって、数値解 z_l は厳密解 α に向かうものとする。

数値解 z_l が厳密解 α に収束を開始すると、 $f(z_l)$ の精度桁数は減少する。そこで、 L_α を $(z_l - \alpha)$ の計算での、 L_g を $g(z_l)$ の計算での桁落数とする。

もし L 桁計算において、 $f(z_l)$ の精度桁数が喪失したとすれば^[5]

$$(3) \quad L \approx L_\alpha + L_g$$

としてよい。 L_α は z_l の数値解としての精度桁数であるから、 $L_\alpha \approx L - L_g$ よりそれは $g(z_l)$ の精度桁数と一致する。

3. 多重ゼロ点に収束するまでの反復回数

初めに、2 次収束する Newton-Raphson 法に対する m 重解の数値解の減少率を考えてみる。初期値は $f(z) = 0$ のゼロ点をすべて含む凸多角形の外側にあるものとする。 $f(z)$ を $f(z) = (z - \alpha)^m g(z)$ とおくと

$$(4) \quad z_{l+1} = z_l - \frac{(z_l - \alpha)^m g(z_l)}{m(z_l - \alpha)^{m-1} g(z_l) + (z_l - \alpha)^m g'(z_l)}$$

となる。もし $|z_l - \alpha|$ が十分小さくなったとすると、

$$(5) \quad |z_{l+1} - \alpha| \approx \left(1 - \frac{1}{m}\right) |z_l - \alpha|$$

となる。Rall^[7]はこのことを平均値の定理を使って示している。同様に k 次収束公式では

$$(6) \quad |z_{l+1} - \alpha| \approx \left(1 - \frac{k-1}{m+k-2}\right) |z_l - \alpha|$$

となることが証明される。

例題 1 ここで 2 次の Newton-Raphson 法を孤立 2 重解に適用した場合、収束が始まってから反復停止までの反復回数を計算してみる。簡単のため、計算は 2 進 24 ビット (仮数部) 計算とする。 z_l で収束が始まり z_{l+p} で反復が停止したとする。孤立 2 重解であるから (3) より、 z_{l+p} の精度桁数は計算桁数の半分の 12 ビットと見積もることができるから、

$$(7) \quad \frac{|z_l - \alpha|}{|z_{l+p} - \alpha|} \approx 2^{24/2} = 2^{12}$$

となる。一方、収束が始まってからの数値解は (5) の割合で減少するから、 p 回の反復では

$$(8) \quad |z_{l+p} - \alpha| \approx \left(1 - \frac{1}{2}\right)^p |z_l - \alpha| = \left(\frac{1}{2}\right)^p |z_l - \alpha|$$

となる。(7) と (8) より $p \approx 12$ を得る。収束が始まるまでの反復回数は、もちろん初期値に依存するが、それまでの数値解の減少率については後で述べることにする。

一般に、 m 重解、 t 進 s 桁計算での精度桁数限界は (7) と同様に考えて

$$(9) \quad \frac{|z_l - \alpha|}{|z_{l+p} - \alpha|} \approx t^{s/m}$$

で評価でき、これと $k+1$ 次反復公式の減少率 (6) を組み合わせると、収束が始まってから反復が停止するまでの反復回数 p は

$$(10) \quad p \approx \frac{s \log_e t}{\log_e \left(1 - \frac{k-1}{m+k-2}\right)^m}$$

で評価できる。 m を無限大にしたとき(10)式の右辺は $(s \log_e t)/(k-1)$ となる。この結果は、平野・伊理^[9]によって指摘された定性的な性質

「 z_k が何重解の数値解であっても、 $f(z_k)$ が誤差のみとなるまでの反復回数はほぼ一定」を定量的に表したものである。具体的に

$$(11) \quad A(k, m, t, s) = -\frac{s \log_e t}{\log_e \left(1 - \frac{k-1}{m+k-2}\right)^m}$$

とにおいて10進16桁計算の場合の反復回数 $p \approx A(k, m, 10, 16)$, $m = 2, 3, 4, 5, 10$ のグラフを図1に示す。 $k \geq 5$ では m の増加と共に p の値が減少していることがわかる。すなわち、 $k \geq 4$ 次収束公式では4重解以上、 $k \geq 5$ 次収束公式では、多重度が増すごとに、収束が始まってから反復が停止するまでの反復回数が減少することになる。

このいわば数値解析の常識に反する結果は、よくよく考えてみると、理にかなったことであると言える。と言うのは多重度を固定し収束次数を高くすると、数値解は早く収束し、一方収束次数を固定し多重度を増加させると、数値解の収束は遅くなるからである。

4. 解法の効率を計る尺度

解法の効率を計る尺度を考えてみる。 n 次多項式、 k 次収束公式に対して、Aberth^[1]のある初期値を与えたとき、収束が始まるまでの数値解の減少率は

$$(12) \quad |z_{l+1}| \approx \left(1 - \frac{k-1}{n+k-2}\right) |z_l|$$

で評価できる。もしその初期値が孤立した単解に収束するとすれば、収束が始まってから反復停止までの反復回数は容易に評価できる。例えば2次収束の公式では精度桁数は1,2,4,8,16...桁と増加していくから倍精度計算では4回程度の反復回数で反復が停止する。収束次数を上げれば、収束が始まってから反復停止までの反復回数は数回と評価できる。このことより、初期値が厳密解から遠く選ばれている場合の解法の効率は、ほとんど収束が始まるまでの計算手間に依存すると言える。すなわち、(12)式の右辺の係数は公式の

効率を評価するのに重要な因子となる。そこで

$$\text{解法の効率} = \frac{1 \text{ 反復あたりの解への接近度合い}}{1 \text{ 反復あたりの計算手間}} = \frac{\frac{1}{1 - \frac{k-1}{n+k-2}}}{1 \text{ 反復あたりの計算手間}}$$

の尺度を導入することにする。 k 次解法の実際のプログラムは表1に示すが、そのプログラムにおいて、計算手間はDOループの回数で数えることにする。すると

DO 10 に関しては n

DO 20 と 25 に関しては $(n-2) + (n-1)(n-2)/2$

DO 30 に関しては $k-3$

DO 40 と 50 に関しては $(k-3) + (k-2)(k-3)/2$

その他 4

より、合計 $(k^2 - k + n^2 + n)/2$ で評価できる。ただし、ここでの計算手間評価においては、四則演算の計算手間は無視した。いま

$$E(k, n) = \frac{\frac{1}{1 - \frac{k-1}{n+k-2}}}{(k^2 - k + n^2 + n)/2}$$

とおくと、 $E(k, n)$ は $k = (\sqrt{2} - 1)n + 2$ の近辺で最大値をとる。 $n = 18, 20, 22$ の場合のグラフを図2に示す。標語的には n 次多項式の単根を求める場合、Aberth の初期値を選べば、 $n/2$ 次の Newton-Raphson 法がほぼ最適な解法となる、と言える。この結果は、高次多項式には高次収束公式がよいと言う、実際の数値結果と良く一致する^[8]。これは従来の結果^{[3],[6]}、例えば4次以上の多項式には3次収束の反復解法がよいと言ったことを修正するものである。

5. 補正值の比について

以上の議論は、 k 次収束の Newton-Raphson 法を

$$z_{l+1} = z_l - \Phi_k(z_l)$$

とおいたとき、 $|z_l|$ が大きいときは

$$\Phi_k(z_l) = \frac{k-1}{n+k-2} z_l$$

で近似したものである。そのことより

$$(13) \quad \frac{\Phi_k(z_{l+1})}{\Phi_k(z_l)} = \frac{z_{l+2} - z_{l+1}}{z_{l+1} - z_l} = 1 - \frac{k-1}{n+k-2}$$

となる。

また z_l が m 重解 α に接近したときは

$$\Phi_k(z_l) = \frac{k-1}{m+k-2} (z_l - \alpha)$$

で近似したものであるから

$$(14) \quad \frac{\Phi_k(z_{l+1})}{\Phi_k(z_l)} = \frac{z_{l+2} - z_{l+1}}{z_{l+1} - z_l} = 1 - \frac{k-1}{m+k-2}$$

となる。

ここでは、実際そのような近似が妥当なものであるか否かを数値的に検証してみる。

(13) と (14) は、数値解が厳密解から遠く離れている場合と、厳密解に極めて近づいた場合は、補正值の比一定となることを示している。この事実は、我々の経験則と良く一致している。次の例題2で具体的に考察してみる。

例題2 $f(z)$ を2重解を1つ持つ5次多項式とする。2次のNewton-Raphson法を適用すると、数値解が厳密解から遠く離れている場合は、

$$(15) \quad \frac{\Phi_k(z_{l+1})}{\Phi_k(z_l)} = \frac{z_{l+2} - z_{l+1}}{z_{l+1} - z_l} = 1 - \frac{k-1}{n+k-2} = \frac{4}{5}$$

より、補正值の比は $4/5$ となる。また厳密解に接近すると

$$(16) \quad \frac{\Phi_k(z_{l+1})}{\Phi_k(z_l)} = \frac{z_{l+2} - z_{l+1}}{z_{l+1} - z_l} = 1 - \frac{k-1}{m+k-2} = \frac{1}{2}$$

より、補正值の比は $1/2$ となる。図 3 に数値結果を示す。複素数の解のため補正值の比は絶対値をとってある。

比較的大きな初期値をあたえた場合、Newton-Raphson 法の収束の段階は 3 段階に分けて考えることができる。最初は数値解が厳密解から遠いため、 $z_{l+1} \approx 0.8z_l$ と線形収束となっている。次の段階では、2 重解があたかも単根のようにみなされて、Newton-Raphson 法の収束の次数の特色がよく表れている。ところがさらに厳密解に近づくと 2 重解であることが認識され、 $z_{l+1} \approx 0.5z_l$ とまた線形収束となっている。 $k = 3, 4, 5, 6$ においても同様なことが言える。この傾向は実係数、実根の場合はより顕著に表れる。

例題 2 は、特別な例をあげたわけではない。多くの数値例で実験して見たが、図 3 の傾向はすべての実験例で観測された。もちろん、厳密解に十分接近していながら、反復停止則（ここでは^[4]を用いた）等の関係から、再び厳密解から数値解がはじき飛ばされるような例は除くことにする。

次に特別な例として、5 次多項式で 5 重解の場合、つまり $n = m$ の場合を考えてみる。上の考察によれば、補正值の比は直線となるはずである。

例題 3 $f(z)$ を $1+i$ を 5 重解としてもつ 5 次多項式とする。色々な初期値をあたえて計算してみても、図 4 の傾向は変わらなかった。このことはもちろん Newton-Raphson 法を変形することによっても容易に確かめられることでもある。なお、収束の終わりの段階において、補正項の比が跳ねる現象が起きるのは、 $f(z_l)$ の値が誤差のみとなった一つの証左である。

6. おわりに

本小論では、どんな収束次数の Newton-Raphson 法においても、

- (A) 収束するまでの反復回数は、たとえどんなにゼロ点の多重度が増しても、ある一定の値に収束する、
- (B) 5 次収束以上の公式ではゼロ点の多重度が増加すると、収束するまでの反復回数が減

少する,

ことを示した。また Newton-Raphson 系解法の効率を計るのに新しい尺度を導入し,

(C) n 次多項式には $n/2$ 次収束の公式が適している,

と言う, 実際の数値実験と良く一致する結果を得た。

最後に触れた補正值の比は, Rall^[7]が 2 次収束の Newton-Raphson 法について得た結果の高次収束解法への拡張である。特に彼が考慮しなかった, Newton-Raphson 法が収束に入る前の状態をも考察の対象としたために, Newton-Raphson 法の収束のプロセスがより明確になったと思われる。Newton-Raphson 法の一つの特色は補正值の比が急激に減少する点にあると言えよう。数値結果を示す図 5, 6, 7, 8 を参照して頂きたい。

REFERENCES

1. O. Aberth, *Iteration methods for finding all zeros of a polynomial simultaneously*, Math. Comp. **27** (1977), 339-344.
2. L. Collatz, *Functional analysis and numerical mathematics*, Academic Press, New York, 1966.
3. H. Ehrmann, *Konstruktion und Durchführung von Iterationsverfahren höherer Ordnung*, Arch. Rational Mech. Anal. **4** (1957), 65-88.
4. M. Igarashi, *A Termination Criterion for Iterative Methods Used to Find the Zeros of Polynomials*, Math Comp. **42** (1984), 165-171.
5. M. Igarashi, *Practical problems arising for finding roots of nonlinear equations*, Applied Numerical Math. **1** (1985), 433-455.
6. T. Pomentale, *A class of iterative methods for holomorphic functions*, Numer. Math. **18** (1971), 193-203.
7. L.B.Rall, *Convergence of the Process to Multiple Solutions*, Numer. Math. **9** (1966), 23-37.
8. 五十嵐正夫・永坂秀子, *Newton-Raphson 系解法の収束次数と反復回数*の関係, 情報処理学会論文誌 **32** (1991), 1349-1353.
9. 伊理正夫, *ニュートン法*の実際, 数理科学 **8** (1981), 10-16.

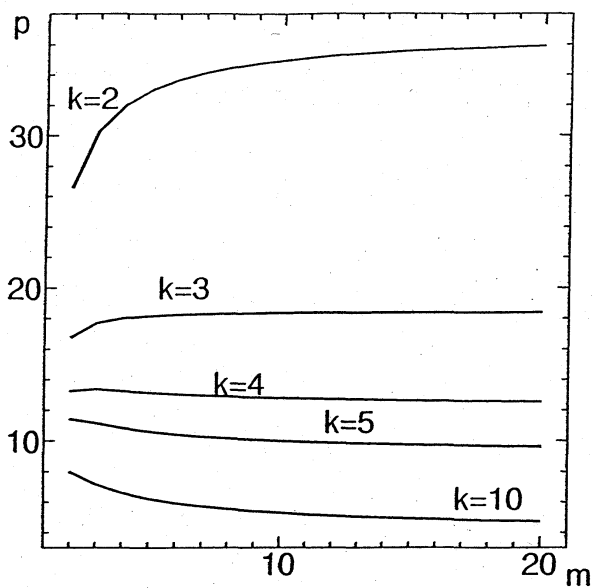


図1 多重解と反復回数

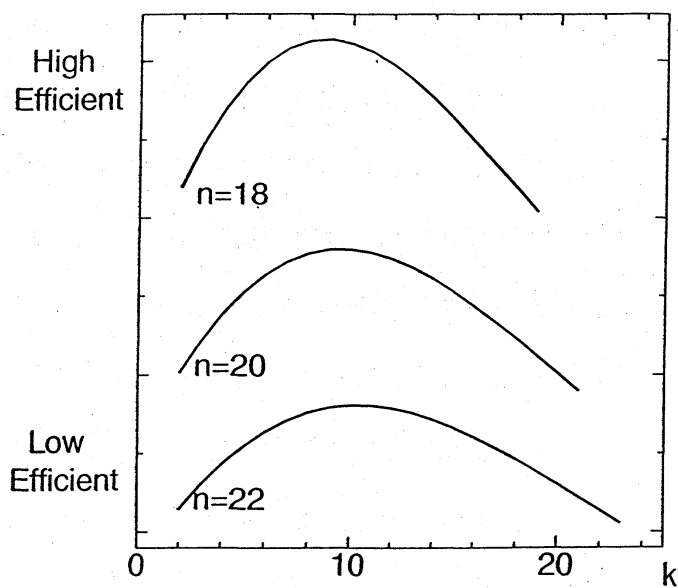


図2 収束次数と解法効率

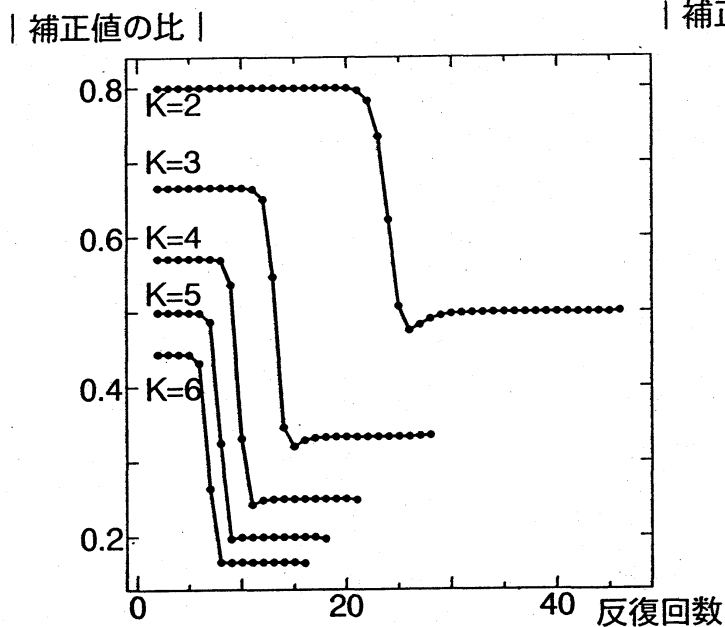


図3 5次多項式2重解の場合の補正值の比
厳密解は $100+100i, 100+100i, -5+6i, 6-5i, 2+9i$

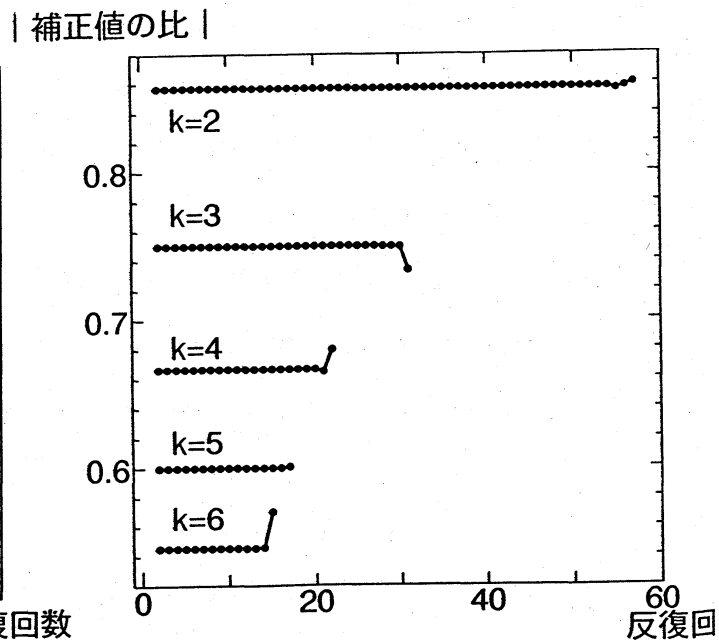


図4 5次多項式5重解の場合の補正值の比
厳密解は $z=1+i$

| 補正項の比 |

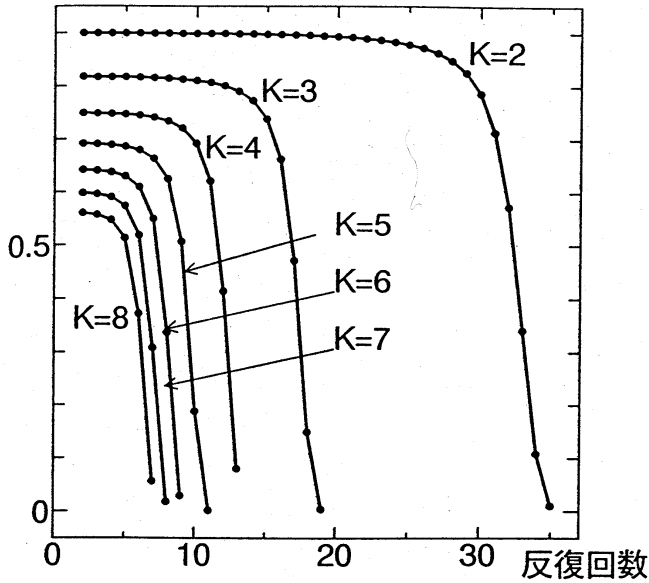


図5 悪条件の多項式の補正項の比
 $F(Z)=(Z-10)(Z-11)(Z-12)\dots(Z-19)$

| 補正項の比 |

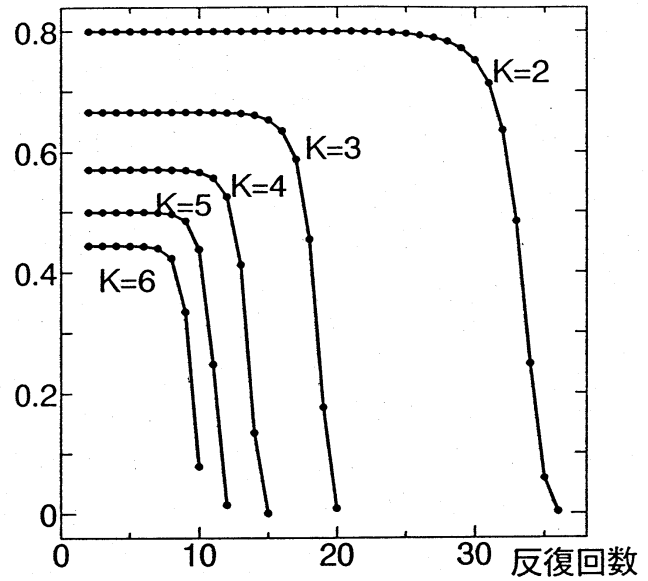


図6 近接ゼロ点を持つ多項式の補正項の比
 $F(Z)=(Z-1.23)(Z-1.24)(Z-1.25)(Z-1.26)(Z-1.27)$

| 補正項の比 |

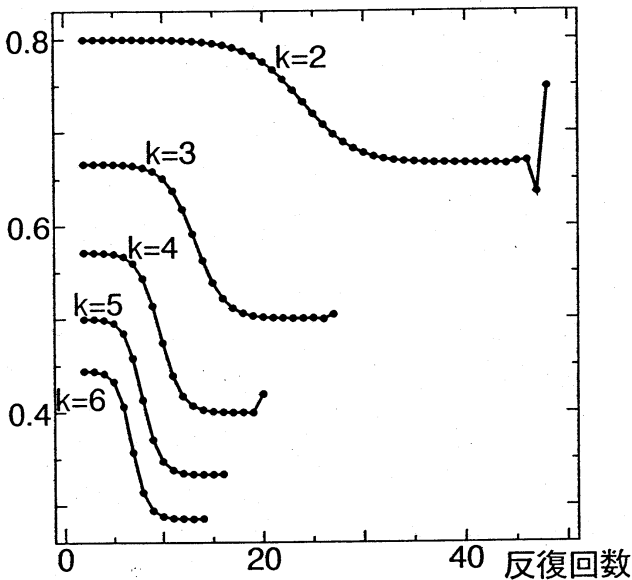


図7 Rallの例題の補正項の比
 $f(z)=(z-1)(z-1)(z-1)(z-2)(z-3)$

| 補正項の比 |

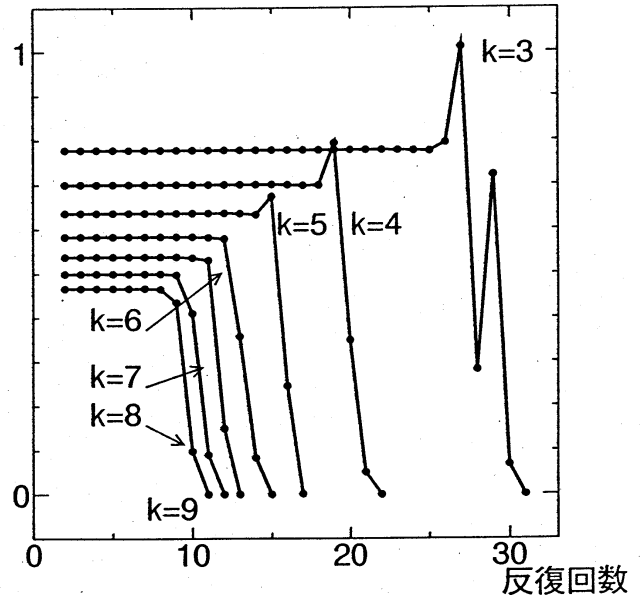


図8 ゼロ点の近傍で数値解が激しく変化する例
 $z=1+i, -1+i, 1, -1, -i, i, -1+i, -1-i$