

## 多項式のゼロ点の精度と計算方法

日大 生物資源科学部 五十嵐正夫 (Masao Igarashi)

### 1. はじめに

あるソフトウェアを作った場合、その品質保証と言うことがよく言われるが、その意味は多岐にわたるようである。例えば、ソフトウェアの信頼性、効率性、操作性、更新性はもとより、あるアルゴリズムによって導き出された数値結果に対する精度保証なども、その中に含まれるようである [1]。最近その精度保証を利用した「理論の実験的な検証」やその逆とも言える「実験的な検証から理論へ」と言った手法の確立が大きな関心を集めているようである。

ところで、数値的立場での精度保証の方法としては、昔からモデルのチェック、極端な問題を解く、計算順序や計算アルゴリズムを変える、計算桁数を変える、数の丸め方を変える等、色々な方法が提案されている。ここでは一つの事例として、計算アルゴリズムを変えるという視点から、精度保証の課題について考察してみる。

多項式のゼロ点を求める場合、その多項式の値は Horner 法によって通常は計算されるが、ごく特殊な多項式の値は漸化式計算によっても求めることができる。その場合、両者の数値解の精度桁数を比較すると、次の2つのことが観測される。

1. 漸化式計算を利用した数値解の方が断然精度がよい。
2. 漸化式を用いた場合、悪条件のゼロ点でも計算桁数と同程度の精度を持つ数値解が得られる。

例えば Chebyshev 多項式は高次になるといわゆる近接ゼロ点を持ち、計算桁数程度の精度を有する数値解が得られない例としてよく知られているが [2]、その漸化式計算値を用いて得られるゼロ点の精度桁数は、ほぼ計算桁数と同程度の精度桁数を有することが観測される。ここではその原因を考察する。

### 2. Chebyshev 多項式の計算

$n$  次の Chebyshev 多項式  $T_n(x)$  の値は、次の3項漸化式を用いて計算できる。

$$T_0(x) = 1, T_1(x) = x, T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \quad k = 1, 2, \dots, n-1 \quad (1)$$

そのゼロ点は

$$\alpha(k) = \cos \frac{2k-1}{2n} \pi, \quad k = 1, 2, \dots, n \quad (2)$$

である。Newton-Raphson 法の場合は、導関数も次の漸化式で計算するものとする。

$$T'_0(x) = 0, T'_1(x) = 1, T'_{k+1}(x) = 2T_k(x) + 2xT'_k(x) - T'_{k-1}(x), \quad k = 1, 2, \dots, n-1 \quad (3)$$

適当な初期値から出発すれば、何回かの反復後近似解  $x$  は  $|x| < 1$  を満たすとしてよい。  
 $|x| < 1$  に至るまでに反復が破綻する場合は、3項漸化式の別種の問題であり、すでにその理由  
 は明らかになっているではここでは論じない（例えば [3] 参照）。

そこで  $x = \cos(\theta)$  とおくと

$$T_n(x) = 2xT_n(x) - T_{n-1}(x) = \cos(n\theta) \quad (4)$$

となる。 $x = \cos(\theta)$  が数値解であるとは、よく知られているように計算値  $T_n(x)$  が精度桁を  
 持たないことである。従って、倍精度計算でのいわゆるマシンイプシロンを  $\epsilon = 10^{-16}$  とす  
 ると、 $T_0(x) = 1$  かつ  $|T_k(x)| < 1, k = 1, 2, \dots, n$  から

$$|\cos(n\theta)| = |T_n(x)| \approx 10^{-16} \quad (5)$$

となると  $x$  は数値解と見なすことができる。その時  $\theta$  はある  $\frac{2k-1}{2n}\pi, k = 1, 2, \dots, n$  に接近し

$$\lim_{\theta \rightarrow \frac{2k-1}{2n}\pi} \left| \frac{\cos(n\theta)}{\theta - \frac{2k-1}{2n}\pi} \right| = n \quad (6)$$

となる。(5) と (6) より

$$n\left(\theta - \frac{2k-1}{2n}\pi\right) \approx 10^{-16} \quad (7)$$

と見積もることができる。

### 3. 数値例

$n = 100$  の場合を考えてみる。

$$\theta - \frac{2k-1}{200}\pi \approx 10^{-18} \quad \text{かつ} \quad 0.0157 < \frac{2k-1}{200}\pi < 3.1415 \quad (8)$$

であるから、 $\theta - \frac{2k-1}{200}\pi$  の引き算においては、少なくとも 16 桁以上の桁落ちが起こると  
 見積もることができる。

$$x = \cos(\theta) = \cos\left(\theta - \frac{2k-1}{200}\pi + \frac{2k-1}{200}\pi\right) = \cos\left(\frac{2k-1}{200}\pi\right) + O(10^{-16})$$

であるから、 $x$  の精度桁数は 16 桁程度以上と見積もることができる。

数値例は、減次の影響を避けるために Durand-Kerner 法\*

$$x_{k+1}(l) = x_k(l) - \frac{T(x_k(l))}{\prod_{j=1, j \neq l}^n 2(x_k(l) - x_k(j))}, \quad l = 1, \dots, 100, k = 0, 1, 2, \dots \quad (9)$$

によって次のような環境で計算した。

\*分母の計算のあふれを防止するために通常の計算式をやや変更した。

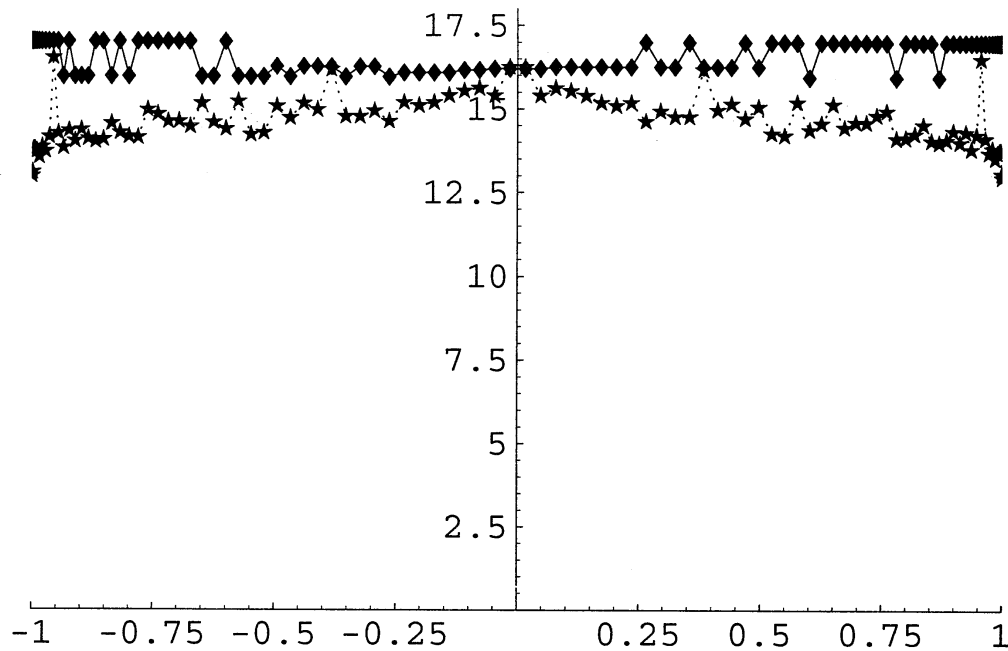


図 1:  $T_{100}(x) = 0$  の数値解の精度と関数値の大きさ

1. 厳密解が実数解と分かっているの、初期値は Aberth 型の円を変形し、 $x$  軸が長軸となるような楕円上に配置した。
2. 反復は  $T_n(x)$  の値の |虚部| がマシンイプシロンよりも小となったとき停止した。
3. MS-FORTRAN の複素倍精度計算を用いた。

数値解の精度桁数とそのときの関数値の大きさを図 1 に示す。 $x$  軸は数値解の位置、 $y$  軸は対応する数値解の精度桁数 (菱形印) と数値解を代入したときの 1 から見た関数値の桁落ち数 (星印) を表している。ほぼ予測通りの結果、すなわちすべての数値解が計算桁数と同程度の精度を有することが示されている。 $n = 2000$  程度 (倍精度計算での Durand-Kerner 法計算の計算限界) まで同様な結果が得られる。

#### 4. おわりに

ここでは精度保証の課題を、計算アルゴリズムの点から考えるための一つの事例、「 $x = \pm 1$  の近辺の数値解の精度桁数は計算桁数よりも少ないと言う従来の定説と異なる事例」、を挙げその原因を考察してみた。

$n$  が大きくなると  $T_n(x)$  の計算において丸め誤差は増加する。しかし、(7) を見ても分かるように、 $n$  の増加が丸め誤差を相殺する方向に働いているため、数億次 (Newton-Raphson 法) までのゼロ点が、ほぼ正確に計算できるものと思われる。もちろん、Chebyshev 多項式が正確に計算できるからと言って、一般の多項式をその 1 次結合として表し、精度の良いゼロ点を求めようとしても、1 次結合の係数の丸め誤差が影響し、Chebyshev 多項式の例の

ようにはいかない。

漸化式計算できる多項式の例としては、Legendre の多項式、Laguerre の多項式、Hermit の多項式などが良く知られている。それらの例についても数値的には Chebyshev の多項式の場合と同様なことが観測される。それらについては、別の機会に考察する予定である。

## 参考文献

- [1] Deutch M.S. Software Verification and Validation, Prentice-Hall, 1982(邦訳：島崎恭一,1984)
- [2] Wilkinson, J.H., Rounding Errors in Algebraic Processes, Prentice-Hall, Englewood Cliffs, N.J., 1963.
- [3] 森正武, 数値解析法, 朝倉書店, 東京, 1984