数式処理 *Bulletin of JSSAC(2022)*
Vol. 28, No. 1, pp. 1 - 2

巻頭言

桁落ち誤差からはじまる研究

讃岐 勝 *

筑波大学医学医療系

筆者は浮動小数を係数に持つ多項式の数式処理の研究を行っており，この分野は数値数式融合計算 (symbolic-numeric computation) と呼ばれる．この世界では数式処理の基本とされるユークリッドの互除法やグレブナー基底計算のアルゴリズムを実行すると誤差のためすぐに計算が破綻し，まったく期待しない出力が返ってくる．計算が破綻する大きな原因としては丸め誤差の積み重ねによる精度不足，**突然**発生する桁落ち誤差による劇的な精度消失などがあるが，計算が急に破綻する原因のほとんどは桁落ち誤差に起因する．

桁落ち誤差が発生する例としてよくあげられるのは二次方程式の解法である．$ax^2 + bx + c = 0$ の 2 つの解は $x = -b \pm \sqrt{b^2 - 4ac}$ であるが，この計算法が有限精度の計算において不安定なのは既知であろう．$\pm\sqrt{b^2 - 4ac}$ のどちらかと b が近い値であれば解の 1 つは有効桁精度を失うからであり，これに伴う誤差は桁落ち誤差と呼ぶ [1)]．このため，桁落ち誤差の入らない解 x_1 をこの公式から計算し，もう一つは $x_2 = c/(ax_1)$ と引き算をしない形で計算をする．

計算の過程においては係数が 0 になることは頻回であり，計算の停止条件が 0 である場合が多い．後者は，不要な桁落ち誤差を回避しながらすべての係数で同時に大きな桁落ち誤差が発生するとき停止条件とする，という言い方もできる．何が不要な桁落ち誤差であるか見極める・避けることは困難で，数式処理のアルゴリズムは数値数式融合計算においては実は相性が悪い．ただし，見極めることができる場合もある．多項式 F_1, F_2 の GCD をユークリッドの互助法で計算してみる．除算を繰り返すアルゴリズムなので，$F_{i+2} = F_i - L_i^{d_i+1}\tilde{Q}_{i+1}F_{i+1}$ と一般的にかける ($L_i = \mathrm{LC}(F_i)/\mathrm{LC}(F_{i+1})$．$\mathrm{LC}(F_i)$ は F_i の主係数，$\tilde{Q}_{i+1}$ は擬商)．$|L_i| \gg 1$ のとき，$F_{i+2} \approx F_{i+1}$ となるため F_{i+3} の計算で桁落ち誤差が発生し，F_{i+4} 以降の計算が破綻する．これを回避して研究へつなげるのが学位論文の主テーマであった．

桁落ち誤差が発生する計算法について記載された専門書は数少ない．桁落ち誤差の発生は避けることであり失敗だからである．失敗は次へのステップであるが，通常の数学の理論では失敗をできるだけ見せない．桁落ち誤差の発生は理論構築に必要な種であり失敗ではない．そう思うと失敗を恐れることなく研究ができるのである．

---

*e-mail sanuki@md.tsukuba.ac.jp

1) $x^2 + 100.01x + 1 = 0$ を解の公式で計算してみよう．