

# 近似代数とその制御系設計への応用

筑波大学 数学系 北本 卓也 (Takuya KITAMOTO) \*

## 1 序論

近年の計算機の急速な発展とともに、かつては一部の研究者だけの道具であった数式処理システムが広く用いられるようになってきた。それと同時に数式処理システムがもつ柔軟性がネックになってきている。具体例を挙げると、記号を含む5次以上の多項式の根をべき根の形で求めることはアーベルの定理により、いかなる数式処理システムでも不可能だし、また、たまたま根が求まったとしても、多くの場合極めて複雑な式となっており、この根を用いて計算を行うのはやっかいである。数式処理システムの正確な代数演算における柔軟性を解決するために、佐々木 (筑波大) は「近似代数」を提案し ([Sas 88])、次の二種類の近似法を具体例として与えた。1つは、正確な数の浮動小数点数による近似法であり ([SN 89][ONS 91][SSKS 91][Kit 98g])、もう1つは数式の打ち切りべき級数による近似法である ([SS 92])。本稿では、近似とはこれら2つの近似法を意味することにし、特に2番目の打ち切りべき級数による近似に焦点を合わせることにする。

打ち切りべき級数は、様々な応用において重要な意味を持っており、数式処理の応用範囲を大幅に広げる可能性を持っている。もし与えられたべき級数の定数項が0でなければ(ほとんどの場合にはそうであるが) そのべき級数の扱いは非常に簡単である。例えば、 $\eta$ のべき級数  $a = \alpha_0 + \alpha_1\eta + \dots + \alpha_k\eta^k$  と  $b = \beta_0 + \beta_1\eta + \dots + \beta_k\eta^k$  が与えられた時、 $\beta_0 \neq 0$  ならば、 $a \oplus b$  ( $\oplus \in \{+, -, \times, /\}$ ) がべき級数として定義でき、

$$a \oplus b \equiv \gamma_0 + \gamma_1\eta + \dots + \gamma_k\eta^k \pmod{\eta^{k+1}}$$

と表すとき、

$$\gamma_i = \begin{cases} \alpha_i + \beta_i & (\oplus = +) \\ \alpha_i - \beta_i & (\oplus = -) \\ \sum_{j=0}^i \alpha_j \beta_{i-j} & (\oplus = \times) \\ \left( \alpha_i - \left( \sum_{j=1}^i \beta_j \alpha_{i-j} \right) \right) / \beta_0 & (\oplus = /) \end{cases} \quad (i = 0, 1, \dots, k)$$

---

\*kita@math.tsukuba.ac.jp

である。以下、与えられたべき級数の定数項は常に 0 でないとし、そうでないときの扱いは今後の課題とするが、この制限のもとでも、様々な興味深い応用が考えられる。本文では、まず近似代数の基本的応用について述べ、その後、これを用いて制御系設計を行う。今後、実数体を  $\mathbf{R}$ 、複素数体を  $\mathbf{C}$ 、体  $\mathbf{K}$  上の  $x, y, \dots, z$  の多項式を  $\mathbf{K}[x, y, \dots, z]$  で表す。また、多項式  $f \in \mathbf{K}[x, y, \dots, z]$  を要素する  $n$  次元ベクトル、 $n \times n$  行列の集合をそれぞれ  $\mathbf{K}[x, y, \dots, z]_n$ ,  $\mathbf{K}[x, y, \dots, z]_{n,n}$  と書き、 $\mathbf{K}$  上の多項式ベクトル、多項式行列と呼ぶことにする。

## 2 近似代数の基本的応用

### 2.1 近似根

多変数多項式  $f(x, \eta_1, \dots, \eta_r) \in \mathbf{C}[x, \eta_1, \dots, \eta_r]$  と  $\alpha_i \in \mathbf{C}$  ( $i = 1, \dots, r$ ) が与えられ、 $f(x, \alpha_1, \dots, \alpha_r)$  が重根を持たないとする。このとき、次式を満たす  $g^{(k)}(\eta_1, \dots, \eta_r) \in \mathbf{C}[\eta_1, \dots, \eta_r]$  を  $f(x, \eta_1, \dots, \eta_r)$  の  $(\eta_1, \dots, \eta_r) = (\alpha_1, \dots, \alpha_r)$  での  $k$  次の近似根と呼ぶ。

$$f(g^{(k)}(\eta_1, \dots, \eta_r), \eta_1, \dots, \eta_r) \equiv 0 \pmod{(\eta_1 - \alpha_1, \dots, \eta_r - \alpha_r)^{k+1}} \quad (1)$$

$f(x, \alpha_1, \dots, \alpha_r)$  が重根を持たない場合、 $f(x, \eta_1, \dots, \eta_r)$  の  $x$  に関する根  $g(\eta_1, \dots, \eta_r)$  は  $(\eta_1, \dots, \eta_r) = (\alpha_1, \dots, \alpha_r)$  でテイラー展開できる。定義からわかるように、 $k$  次の近似根は  $g(\eta_1, \dots, \eta_r)$  の全次数  $k$  次までのテイラー展開であり、記号的ニュートン法 ([KT 78]) で効率的に計算できる。 $f(x, \alpha_1, \dots, \alpha_r)$  が重根を持つ場合は、 $g^{(k)}(\eta_1, \dots, \eta_r)$  は一般には分数べきでの展開 (Puiseux 級数) となる。Puiseux 級数解の計算法については [SK 98][Kit 98f] を参照。

### 2.2 近似固有値、固有ベクトル

多項式行列  $M \in \mathbf{C}[\eta_1, \dots, \eta_r]_{n,n}$  と  $\alpha_i \in \mathbf{C}$  ( $i = 1, \dots, r$ ) が与えられ、 $M$  において  $(\eta_1, \dots, \eta_r) \leftarrow (\alpha_1, \dots, \alpha_r)$  の代入を行った定数行列の固有値が全て相異なるとする。この時、次式を満たす  $\lambda^{(k)} \in \mathbf{C}[\eta_1, \dots, \eta_r]$  とベクトル  $v^{(k)} \in \mathbf{C}[\eta_1, \dots, \eta_r]_n$  をそれぞれ  $(\eta_1, \dots, \eta_r) = (\alpha_1, \dots, \alpha_r)$  での  $k$  次の近似固有値、固有ベクトルと呼ぶ。

$$Mv^{(k)} \equiv \lambda^{(k)}v^{(k)} \pmod{(\eta_1 - \alpha_1, \dots, \eta_r - \alpha_r)^{k+1}} \quad (2)$$

定義の条件を満たすとき、 $M$  の固有値  $\lambda$  と固有ベクトル  $v$  は  $(\eta_1, \dots, \eta_r) = (\alpha_1, \dots, \alpha_r)$  でテイラー展開できる。定義からわかるように、 $k$  次の近似固有値、固有ベクトルは  $\lambda, v$  の全次数  $k$  次までのテイラー展開であり、次の算法で効率的に計算できる ([Kit 94b])。以

下では  $\lambda^{(l)}$ ,  $v^{(l)}$  でそれぞれ  $l$  次の近似固有値と固有ベクトルを表すとし、 $M$  において  $(\eta_1, \dots, \eta_r) \leftarrow (\alpha_1, \dots, \alpha_r)$  の代入を行った定数行列を  $M^{(0)}$  とする。

- (i)  $M^{(0)}$  の固有値  $\lambda^{(0)}$ 、固有ベクトル  $v^{(0)}$  を計算する。  $v^{(0)}$  をその要素の絶対値の最大が 1 となるように正規化し、最大を取る要素のインデックスを  $l$  とする。定数行列  $H$  を次のように定義する ( $H$  は正則であることが保証される)。

$$H \text{ の第 } m \text{ 列} = \begin{cases} (-M + \lambda^{(0)}E) \text{ の第 } m \text{ 列} & (m \neq l) \\ v^{(0)} & (m = l) \end{cases} \quad (3)$$

- (ii)  $j = 0, 1, 2, \dots$  に対し、次式で近似の次数が充分高くなるまで、繰り返し計算を行う。

$$\begin{bmatrix} \Delta v_1^{(j+1)} & \dots & \Delta v_{l-1}^{(j+1)} & \Delta \lambda^{(j+1)} & \Delta v_{l+1}^{(j+1)} & \dots & \Delta v_n^{(j+1)} \end{bmatrix}^T \\ \leftarrow H^{-1}(Mv^{(j)} - \lambda^{(j)}E) \pmod{(\eta_1 - \alpha_1, \dots, \eta_r - \alpha_r)^{j+1}} \quad (4)$$

$$v^{(j+1)} \leftarrow v^{(j)} + \begin{bmatrix} \Delta v_1^{(j+1)} & \dots & \Delta v_{l-1}^{(j+1)} & 0 & \Delta v_{l+1}^{(j+1)} & \dots & \Delta v_n^{(j+1)} \end{bmatrix}^T \quad (5)$$

$$\lambda^{(j+1)} \leftarrow \lambda^{(j)} + \Delta \lambda^{(j+1)} \quad (6)$$

(4)-(6) 式よりわかるように、上の計算式は 1 次収束するが 2 次収束するように計算式を拡張することも可能である ([Kit 94b])。  $M^{(0)}$  が重複固有値を持つ場合は、  $\lambda^{(k)}$ ,  $v^{(k)}$  は一般には分数べきでの展開 (Puiseux 級数) となり、上の算法では計算できない。この場合の計算法については [SK 98][Kit 98f] を参照。

## 2.3 近似逆行列

多項式行列  $M \in \mathbf{C}[\eta_1, \dots, \eta_r]_{n,n}$  と  $\alpha_i \in \mathbf{C}$  ( $i = 1, \dots, r$ ) が与えられ、 $M$  において  $(\eta_1, \dots, \eta_r) \leftarrow (\alpha_1, \dots, \alpha_r)$  の代入を行った定数行列  $M^{(0)}$  が正則であるとする。この時、次式を満たす行列  $P^{(k)} \in \mathbf{C}[\eta_1, \dots, \eta_r]_{n,n}$  を  $M$  の  $(\eta_1, \dots, \eta_r) = (\alpha_1, \dots, \alpha_r)$  での  $k$  次の近似逆行列と呼ぶ。

$$MP^{(k)} \equiv E \pmod{(\eta_1 - \alpha_1, \dots, \eta_r - \alpha_r)^{k+1}} \quad (7)$$

定義の条件を満たすとき、  $P^{(k)}$  は次の算法で計算できる。

- (i)  $M^{(0)}$  の逆行列  $P^{(0)}$  を数値的に計算する。

- (ii)  $j = 0, 1, 2, \dots$  に対し、次式で近似の次数が充分高くなるまで、繰り返し計算を行う。

$$P^{(2^{j+1}-1)} \leftarrow P^{(2^j-1)} + P^{(2^j-1)} \left( E - MP^{(2^j-1)} \right) \pmod{(\eta_1 - \alpha_1, \dots, \eta_r - \alpha_r)^{2^{j+1}}} \quad (8)$$

(8) 式よりわかるように、上の算法は 2 次収束し、効率的に近似逆行列を計算する。

## 2.4 線形微分方程式のべき級数解

多項式行列  $A \in \mathbf{R}[\eta_1, \dots, \eta_r]_{n,n}$ 、多項式ベクトル  $B \in \mathbf{R}[\eta_1, \dots, \eta_r]_n$ 、実数  $\alpha_i$  ( $i = 1, \dots, r$ ) が与えられた時、 $A$  において  $(\eta_1, \dots, \eta_r) = (\alpha_1, \dots, \alpha_r)$  の代入を行った行列を  $A_0 \in \mathbf{R}_{n,n}$  とする。  $A_0$  の固有値  $\lambda_i^{(0)} \in \mathbf{C}$  ( $i = 1, \dots, n$ ) が  $\lambda_i^{(0)} \neq 0$  ( $i = 1, \dots, n$ ) かつ  $\lambda_i^{(0)} \neq \lambda_j^{(0)}$  ( $i \neq j$ ) を満たす時、次の線形微分方程式

$$\frac{d}{dt}x = Ax + Bu, \quad x(0) = 0, \quad u(t) = \begin{cases} 0 & (t < 0) \\ 1 & (t \geq 1) \end{cases} \quad (9)$$

の解を  $x$ 、 $x_i$  をその第  $i$  要素とする。この時、 $x_i$  の次の形での打ち切りべき級数近似（ただし、 $\alpha_{j,i} \in \mathbf{R}[\eta_1, \dots, \eta_r]$ ）

$$x_i \equiv \alpha_{0,i} + \alpha_{1,i}(t - t_0) + \dots + \alpha_{l,i}(t - t_0)^l \pmod{(t - t_0)^{l+1}, (\eta_1 - \alpha_1, \dots, \eta_r - \alpha_r)^{k+1}}, \quad (10)$$

を (9) の  $(t, \eta_1, \dots, \eta_r) = (t_0, \alpha_1, \dots, \alpha_r)$  での  $(l, k)$  次の近似解と呼ぶ。近似解は近似固有値、固有ベクトルを用いて計算できる（詳細については、本論を参照）。

## 2.5 代数的 Riccati 方程式の近似解

代数的 Riccati 方程式

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (11)$$

は制御系設計において極めて重要な方程式である。その解の計算法については様々な研究がなされているが、 $A, B, Q, R$  が記号を含む場合にはこれまでの算法を直接用いることができないため、一般的には明示的な形で解を求めることができない。そこで、代数的 Riccati 方程式の解をべき級数の形で近似的に求め、これを代数的 Riccati 方程式の近似解として次のように定義する。  $A \in \mathbf{R}[\eta_1, \dots, \eta_r]_{n,n}$ 、 $B \in \mathbf{R}[\eta_1, \dots, \eta_r]_n$ 、 $Q \in \mathbf{R}[\eta_1, \dots, \eta_r]_{n,n}$ 、 $R \in \mathbf{R}[\eta_1, \dots, \eta_r]$ 、 $\alpha_i \in \mathbf{R}$  ( $i = 1, \dots, r$ ) が与えられ、 $A, B$  に  $(\eta_1, \dots, \eta_r) = (\alpha_1, \dots, \alpha_r)$  の代入を行った行列を  $A_0 \in \mathbf{R}_{n,n}$ 、 $B_0 \in \mathbf{R}_0$  とする。  $A_0, B_0$  が可制御、つまり

$$\text{rank} \left( \begin{bmatrix} B_0 & A_0 B_0 & \dots & A_0 B_0^{n-1} \end{bmatrix} \right) = n \quad (12)$$

である時、次の方程式を満たす  $P^{(k)} \in \mathbf{R}[\eta_1, \dots, \eta_r]_{n,n}$  を代数的 Riccati 方程式の  $(\eta_1, \dots, \eta_r) = (\alpha_1, \dots, \alpha_r)$  での  $k$  次の近似解と呼ぶ。

$$P^{(k)}A + A^T P^{(k)} - P^{(k)}BR^{-1}B^T P^{(k)} + Q \equiv 0 \pmod{(\eta_1 - \alpha_1, \dots, \eta_r - \alpha_r)^{k+1}} \quad (13)$$

$A, B, Q, R$  が数値であるときの代数的 Riccati 方程式の解法として、最も広く使われているのは、固有値、固有ベクトルを用いる [Lau 79][Mac 63] である。定義の条件が満たされ

るとき、近似解はこの算法を近似固有値、近似固有ベクトルを用いて拡張することにより計算できる (詳細については、本論を参照)。

### 3 制御系設計への応用

通常、制御系設計は次の手順で行われている。

1. 与えられた制御対象の数学モデル (微分方程式) を構成する。
2. 設計仕様を考慮し、設計パラメータや重み関数を決定する。
3. コントローラを設計し、制御系を構成する。
4. シミュレーションを行い、できた制御系が設計仕様を満足するかを調べる。満足するならば設計を終え、そうでなければ 2. へ行く。

しかしながら、多くの場合設計パラメータと制御系の時間応答は明確な対応関係を持っていないので、従来の設計法では時間応答の仕様に対して設計パラメータをうまく決めることは困難である。現状では、時間応答の仕様が与えられると、設計者が試行錯誤により設計パラメータを決めており、設計者に多くの負担を強いている。そこで、先に与えた近似代数の基本的応用を用い、上の 4. におけるシミュレーション結果をパラメータのべき級数の形で求めることを考える。こうすることにより、設計仕様に合うようにパラメータを直接決める事が可能であり、見通しの良い制御系設計が可能である。このアイデアに基づき、[MG 90, pp. 161–168] の制御系設計問題に対し、 $H_2$  制御系を設計すると、[MG 90] では 0.531 秒であった応答時間を 0.3902 秒に向上させることができた。

### 4 結論

数式処理は、近年広く用いられるようになってきたが、正確な代数演算をその基礎にしているため、柔軟性にかける欠点を持っている。近似代数は数式処理を柔軟化し、応用範囲を広げようとする試みであり、特に工学的な応用に対し有効であると考えられる。本稿では、多変数多項式の根や要素を多項式とする行列の固有値、固有ベクトル等を打ち切りべき級数の形で計算する算法を与え、それらを制御系設計へ応用した。従来困難であった時間応答の仕様を持つ制御系設計問題に対し、本稿の設計法が有効であることを設計例を用いて示した。今後は近似代数の制御系設計への応用をさらに推し進め、 $H_\infty$  制御系設計法や  $\mu$  設計法などへ応用を課題としたい。

## 参 考 文 献

- [CC 82] G. Corliss and Y.F. Chang, "Solving Ordinary Differential Equations Using Taylor Series," ACM Trans. Math. Soft., Vol. 8, pp. 114–144, 1982.
- [HW 91] E. Hairer and G. Wanner, "Solving Ordinary Differential Equations II," Springer-Verlag, 1991.
- [Kit 94a] T. Kitamoto, "Approximate Eigenvalues, Eigenvectors and Inverse of a Matrix with Polynomial Entries," Japan J. Indus. Appl. Math., Vol. 11, No. 1, pp. 73–85, 1994.
- [Kit 94b] T. Kitamoto, "Approximate Matrix Operations on a Matrix," Winter Workshop on Computer Algebra, Proc. of Extended Abstracts, pp. 16–20, Tokyo, 1994.
- [Kit 95a] 北本 "近似固有値、固有ベクトルとその最適制御への応用," 電子情報通信学会論文誌, Vol. J78-A, No. 4, pp. 531–534, 1995.
- [Kit 96] T. Kitamoto, "Hensel Construction with Arbitrary Degree of Convergence," Japan J. Indus. Appl. Math., Vol. 13, No. 2, pp. 203–215, 1996.
- [Kit 98a] 北本 "近似代数を用いた  $H_2$  最適制御系の解析と設計," 電子情報通信学会論文誌, Vol. J81-A, No. 2, pp. 289–292, 1998.
- [Kit 98b] 北本 "代数的 Riccati 方程式のべき級数解の計算法について," 電子情報通信学会論文誌, Vol. J81-A, No. 3, pp. 445–447, 1998.
- [Kit 98c] 北本 "近似固有値の計算法について," 電子情報通信学会論文誌, Vol. J81-A, No. 4, pp. 803–806, 1998.
- [Kit 98d] T. Kitamoto, "Approximate Singular-value Decomposition of a Matrix with Polynomial Entries," Proc. of ATCM'98, Tsukuba-shi, pp. 342–351, 1998.
- [Kit 98e] 北本 "線形微分方程式のべき級数解とその応用" 第 21 回 D.S.T. シンポジウム予稿集, pp. 283–286, 1998.
- [Kit 98f] T. Kitamoto, "On Puiseux Expansion of Approximate Eigenvalues and Eigenvectors," IEICE Trans. Fundamentals, Vol. E81-A, No. 6, pp. 1242–1251, 1998.
- [Kit 98h] 北本 "近似代数とその制御系設計への応用" システム/制御/情報 12 月号 pp. 668–676, 1998.
- [Lau 79] A.J. Laub, "A Schur method for solving algebraic Riccati equations," IEEE Transactions on Automatic Control, Vol. 24, pp.913–921, 1979.

- [Mac 63] A.G.J. MacFarlane, "An eigenvector solution of the optimal linear regulator," *Journal of Electronics and Control*, Vol. 14, pp.643–654, 1963.
- [MG 90] D. C. McFarlane and K. Glover "Robust Controller Design Using Normalized Coprime Factor Plant Descriptions," *Lecture Notes in Control and Information Sciences*, Springer-Verlag, 1990.
- [NT 91] M-T. Noda and T. Sasaki, "Approximate GCD and its application to ill-conditioned algebraic equations," *J. Comp. Appl. Math.* Vol. 38, pp. 335–351, 1991.
- [ONS 91] M. Ochi, M-T. Noda and T. Sasaki, "Approximate greatest common divisor of multivariate polynomials and its application to ill-conditioned system of algebraic equations," *J. Inf. Proces.*, Vol. 14, pp. 292–300, 1991.
- [Sas 88] T.Sasaki "Approximate Algebraic Computation (in Japanese)," *Suriken-Kokyuroku* (Collection of Research Reports, Research Institute of Mathematical Study, Kyoto University) Vol 676, pp.307–319, 1988.
- [SSKS 91] T. Sasaki, M. Suzuki, M. Kolar and M. Sasaki, "Approximate factorization of multivariate polynomials and absolute irreducibility testing," *Japan J. Indus. Appl. Math.* Vol. 8, pp. 357–375, 1991.
- [SS 93] T. Sasaki and M. Sasaki, "A unified method for multivariate polynomial factorizations," *Japan J. Indus. Appl. Math.*, Vol. 10, pp. 21–39, 1993.
- [SS 95] T. Sasaki and M. Sasaki, "A Study of Approximate Polynomials I - Representation and Arithmetic-," *Japan J. Indus. Appl. Math.*, Vol. 12, pp. 137–161, 1995.
- [SK 98] T. Sasaki and F. Kako, "Solving multivariate algebraic equation by Hensel construction," *Japan J. Indus. Appl. Math.*, (to appear).