

保存則による発展方程式の分類 (数式処理の利用) I - 形式的線形化可能系

広島大学工学部 渡辺芳英 (YOSHIHIDE WATANABE)

“(完全)可積分な発展方程式”の定義はまだなされていないが,少なくとも無限個の保存量が存在するという条件は必要ではないかと思われる.このような無限個の各保存量をハミルトン関数として時間発展を考えれば高次のヒエラルキーと呼ばれる一連の発展方程式が得られ,これらの発展方程式の flow は互いに可換となる.高次ヒエラルキーの概念を保存量に対応する正準ベクトル場(無限小正準変換)と見ると Lie Bäcklund symmetry と呼ばれるものになる.このような観点から無限個の Lie Bäcklund symmetry (以後単に symmetry と略記)を許容する発展方程式を数え挙げる(分類する)という研究の流れが生じた.一つの symmetry があればそれに作用して無限個の symmetry を生成する作用素は Olver([O])により提唱され, recursion operator と呼ばれた. Sokolov と Shabat ([SS])は発展方程式が任意階数の symmetry を許容すれば(形式的には)recursion operator を許容することを示し,更に発展方程式が recursion operator を許容するためには,方程式から決まるある加算無限個の量(標準密度と呼ばれる)がすべて発展方程式の保存密度とならなくてはならないことを見出した.本稿では上記の標準密度のうち有限個をいくつかのタイプの発展方程式について計算し,それらがすべて自明な保存密度となる場合を分類する.計算はほとんどすべて計算機による数式処理の助けを借りて行う.

1. 単独方程式の場合

\mathcal{R} は u_0, u_1, u_2, \dots , の実数係数の有理関数体で $Du_j = u_{j+1} (j = 0, 1, 2, \dots)$ をみたす微分 D をもつものとする(場合によっては有理関数体よりももうすこし広い関数体で考えた方がよい場合もあり,必要に応じて体を拡大して考えることにする).習慣に従って u_0 を u とかき,微分 $\partial/\partial u_i (i = 0, 1, 2, \dots)$ を ∂_i と略記する. $f \in \mathcal{R}$ に対して $\partial_k f \neq 0$ となる最大の整数を f の階数と呼び $\text{ord } f$ とかく. また $f \in \mathcal{R}$ に対してその発展微分 ∂_f を

$$(1.1) \quad \partial_f = \sum_{j \geq 0} (D^j f) \partial_j,$$

で定義する.今 \mathcal{R} にブラケット $[,]$ を

$$(1.2) \quad [f, g] = \partial_f g - \partial_g f.$$

で導入すれば,発展微分は以下の交換関係をみたす:

$$(1.3) \quad [\partial_f, \partial_g] = \partial_f \partial_g - \partial_g \partial_f = \partial_{[f, g]}.$$

さて未知関数 $u(x, t)$ についての発展方程式

$$(1.4) \quad u_t = H(u, u_1, u_2, \dots, u_m) \quad (H \in \mathcal{R}),$$

を考えよう. ここで u_t は $(\partial/\partial t)u(x, t)$ を意味し, u_j は $(\partial/\partial x)^j u(x, t)$ を意味するものとする. このような発展方程式を考えたとき ∂_H は方程式上での時間微分になる. 方程式 (1.4) の任意の解 u に対して無限小変換 $u + \epsilon f$ ($f \in \mathcal{R}$ が ϵ の高次のオーダーを無視して再び方程式の解となる為の f に対する条件を考えよう. \mathcal{R} 上に微分作用素:

$$(1.5) \quad H_* = \sum_{j \geq 0} (\partial_j H) D^j$$

を導入すれば f は微分方程式

$$(1.6) \quad (\partial_H - H_*)f = f_t - H_*f = 0$$

を満たすことが容易に判る. この微分方程式 (1.6) の (\mathcal{R} の適当な拡大体での) 解を方程式 (1.4) の Lie-Bäcklund symmetry (以後単に symmetry) とよぶことにする (1.2) で定義したブラケットを用いて定義方程式 (1.6) を書き直せば $[H, f] = 0$ となり, 方程式 (1.4) の symmetry 全体は (1.2) で定義される Lie ブラケットをもつ Lie 環となる. 方程式 (1.4) は独立変数 x, t を陽に含んでいないので x, t に関する平行移動に対して不変であり, その事実に応じて u_1, H という二つの symmetry を常にもつ (自明な symmetry とよばれる).

KdV 方程式等のいわゆるソリトン方程式においては L_H は無限次元可換 Lie 環になるかまたはそのような Lie 環を部分 Lie 環として含んでおり, 無限個の生成元は recursion operator とよばれる作用素を自明な symmetry に順次作用させることによって得られることが知られている. \mathcal{R} 上の微分積分作用素 L (一般には \mathcal{R} 係数の L に関する形式的 Laurent 級数で与えられる) が symmetry を定義する微分作用素と可換, すなわち

$$(1.7) \quad [\partial_H - H_*, L] = L_t - [H_*, L] = 0$$

であるとき L は (symmetry の) recursion operator であるとよばれる (Olver [O]). ここで L_t は L の各係数を t で微分 (∂_H を作用) したものを表す. 方程式 (1.6) と (1.7) はよく似ている. 実際方程式 (1.6) の両辺の * 微分をとることにより

$$(1.8) \quad [\partial_H - H_*, f_*] = (f_*)_t - [H_*, f_*] = \partial_f(H_*)$$

が得られる. ここで ∂_f は H_* の各係数に作用するものとする. (1.8) の左辺は (1.7) の左辺の L を f_* に置き換えたものになっており, f の階数は f_* の D に関する次数となることに注意すると, f の階数が H の階数に比べて高ければ (1.8) の両辺の D に関する次数を比べると左辺の方が高くなり, f_* は L の第一近似と考えてよいことがわかる. この事実注目すれば, 方程式 (1.4) が任意階数の symmetry を許容するとき recursion operator が D に関する形式的 Laurent 級数として求められることが判る. 逆に recursion operator L がみつければ, 方程式 (1.4) の symmetry f に対して Lf も symmetry となる. したがってそれを自明な symmetry H に順次作用させることにより無限個の symmetry $\{L^n H; n = 1, 2, 3, \dots\}$ が得られそうだが, それは形式的な意味でしかいえない. 例えば KdV 方程式 $u_t = u_3 + 6uu_1$ の recursion operator は $D^2 + 4u + 2u_1 D^{-1}$ で与えられる. このように多くの発展方程式では recursion operator が D の負冪の項 (積分項) を含み, 従ってこのような作用素を順々に作用したものが \mathcal{R} の元として意味を持つかどうか判らない (勿論 KdV ではこのような心配はならず recursion operator の作

用は symmetry に対して well-defined になる). ソリトン方程式の一つの特徴の一つは無限個の symmetry を許容することであり, そのような方程式を数え挙げることは興味ある問題である. しかし無限個の symmetry を許容するという条件は“無限個の”という部分が漠然として取り扱いにくい. そのかわりにここでは recursion operator を許容するという条件を調べる (上記で述べた理由で recursion operator の存在は必ずしも無限個の symmetry の存在を保証しないけれども). Sokolov, Shabat[SS] によれば発展方程式が D に関する形式的 Laurent 級数としての recursion operator を許容する為には方程式から計算される加算無限個の量 $\rho_{-1}, \rho_0, \rho_1, \dots$ が方程式の保存密度となることが必要十分である.

まず保存密度について復習する. $\rho \in \mathcal{R}$ に対して $\rho_t = \partial_H \rho = D\sigma$ となる $\sigma \in \mathcal{R}$ が存在するとき ρ は発展方程式 (1.4) の保存密度であると呼ばれる. 実際, 適当な境界条件の下で ρ を積分したものは方程式の保存量となる. $f, g \in \mathcal{R}$ に対して適当な $h \in \mathcal{R}$ を用いて $f - g = Dh$ と書けるとき $f \sim g$ とかくと \sim によって \mathcal{R} に同値関係が定義される. この同値関係を用いると ρ が保存密度であることは $\rho_t = \partial_H(\rho) \sim 0$ と表される. もし $f \sim 0$ または $f = \text{定数}$ ならば t 微分 ($= \partial_H$) と D の可換性などより f は任意の発展方程式の保存密度となり, このような f は自明な保存密度と呼ばれる. 次に Sokolov と Shabat([SS]) の議論を (必要な部分だけ) 簡単に説明しよう.

\mathcal{R} 係数の D に関する形式的 Laurent 級数の全体を $\mathcal{R}((D))$ で表す. $\mathcal{R}((D))$ には通常の一変数擬微分作用素環の積が導入されて (非可換) 環となる. $\mathcal{R}((D))$ の元

$$P = p_n D^n + p_{n-1} D^{n-1} + \dots \quad p_n, p_{n-1}, \dots \in \mathcal{R}$$

に対して $p_n \neq 0$ であるなら P の次数は n であるといい $\deg P = n$ とかく. また P の D^{-1} の係数を $\text{Res } P$ とかく. よく知られていることだが $P, Q \in \mathcal{R}((D))$ に対して $\text{Res } [P, Q] \sim 0$ である. すなわち $\text{Res } [P, Q]$ は D に関して一回積分出来る. $P \in \mathcal{R}((D))$ の次数が n 以下であるとき $P = O(n)$ と表す. $L \in \mathcal{R}((D))$ が方程式 (1.7) を満たせば L の任意の整数冪, また更に $\deg L = n$ として分数冪 $L^{k/n}$ ($k = \pm 1, \pm 2, \dots$) も方程式 (1.7) を満たすことに注意すれば, L は次数 1 の元

$$(1.9) \quad L = l_1 D + l_0 + l_{-1} D^{-1} + l_{-2} D^{-2} + \dots$$

仮定してよい. そこで次の定義を行う. (1.8) 式で与えられる L が

$$(1.10) \quad \deg (L_t - [H_*, L]) \leq m + 1 - N$$

を満たすとき L は階数 N の formal symmetry であると呼ぶことにする. 明らかに任意の (次数 1 の) $\mathcal{R}((D))$ の元は階数 1 の formal symmetry であり, また $[H_*, (H_*)^{1/m}] = 0$ と $\deg L_t \leq 1$ であることに注意すれば $H^{1/m}$ (本質的には $H^{1/m}$ の最初の $m-1$ 項まで, すなわち D^{-m+3} の項までとったもの) が階数 m の formal symmetry になることも判る. 更に (1.7) を用いれば, $f \in \mathcal{R}$ が発展方程式 (1.4) の symmetry, すなわち (1.6) を満たせば f の階数を n とすれば $(f_*)^{1/n}$ は階数 n の formal symmetry となることを示すことが出来る. 従って任意に高い階数の symmetry があれば recursion operator が存在することになる. さて, これで階数 m の formal symmetry の存在は自明であることが判ったのであるから次は階数 $m+1$ 以上の formal symmetry が存在する条件を調べなくてはならない.

(1): 階数 $m+1$ の formal symmetry を求めるには $L_t - [H_*, L]$ の D 係数を調べることにより L の D^{-m+2} の係数 l_{2-m} を求めればよい. 既に l_1, \dots, l_{3-m} は $(H_*)^{1/m}$ の係数から定まり, $L_t - [H_*, L] = O(1)$ となっているので

$$(1.11) \quad L_t - [H_*, L] = O(0)$$

となるように l_{2-m} を決めればよい. (1.11) 式の両辺に左右から L^{-1} を掛けて $(L^{-1})_t = -L^{-1}L_tL^{-1}$ に注意すると

$$(1.11a) \quad (L^{-1})_t - [H_*, L^{-1}] = O(-2)$$

よって上式の左辺の Res が消えればよく, 方程式 $\text{Res}((L^{-1})_t - [H_*, L^{-1}]) = 0$ が解ければよい. ここで $\rho_{-1} = \text{Res} L^{-1}$, $\sigma_{-1} = D^{-1} \text{Res} [H_*, L^{-1}]$ とおけば保存則

$$(1.12) \quad (\rho_{-1})_t = D\sigma_{-1}$$

が得られる. ρ_{-1} が保存密度であり, $\rho_{-1} = (\partial_m H)^{-1/m}$ で与えられる. ρ_{-1} が定数, すなわち $\partial_m H$ が定数ならば $\sigma_{-1} = 0$ とおく. その場合 $(H_*)^{1/m}$ (の最初の m 項をとったもの) は階数 $m+1$ の formal symmetry となる.

(2): 階数 $m+2$ の formal symmetry を求めるには $L_t - [H_*, L]$ の定数項を調べることにより L の D^{-m+1} の係数 l_{1-m} を求めればよい. 言い換えれば

$$(1.13) \quad L_t - [H_*, L] = O(-1)$$

となるように l_{1-m} を求める. (1.13) 式に右から L^{-1} を掛けて

$$(1.13a) \quad L_t L^{-1} - [H_* L^{-1}, L] = O(-2)$$

すなわち $\text{Res}(L_t L^{-1} - [H_* L^{-1}, L]) = 0$ を得る. $\text{Res} L_t L^{-1} = (l_0/l_1)_t$ であることに注意して $\rho_0 = l_0/l_1$, $\sigma_0 = D^{-1} \text{Res} [H_* L^{-1}, L]$ とおくと保存則

$$(1.14) \quad (\rho_0)_t = D\sigma_0$$

を得る. $m \geq 3$ ならば

$$\rho_0 = \frac{1}{m} q^m \partial_{m-1} H + \frac{1}{2} (m-1) D \log q$$

で与えられる. 但し $q = \rho_{-1}$ である.

(3): 階数 $m+3$ の formal symmetry を求めるには $L_t - [H_*, L]$ の D^{-1} 係数を調べて L の D^{-m} の係数 l_{-m} を求めればよい. すなわち

$$(1.15) \quad \text{Res}(L_t - [H_*, L]) = 0.$$

これより $\rho_1 = \text{Res} L = l_{-1}$, $\sigma_1 = D^{-1} [H_*, L]$ とおけば保存則

$$(1.16) \quad (\rho_1)_t = D\sigma_1$$

を得る.

(4): (3)と同様に $\rho_k = \text{Res } L^k$, $\sigma_k = D^{-1}[H_*, L^k]$ ($k = 2, 3, \dots$) おけば高階の formal symmetry が存在する為の条件が保存則

$$(1.16) \quad (\rho_k)_t = D\sigma_k$$

として得られる.

ρ_k ($k = -1, 0, 1, \dots$) を発展方程式 (1.4) の標準密度 (canonical densities) と呼ぶ. recursion operator を許容する発展方程式を数え上げるにはすべての ρ_k が保存密度となるように H を定めればよい. その為にはよく知られているように条件

$$(1.17) \quad \frac{\delta}{\delta u} \partial_H(\rho_k) = 0 \quad (k = -1, 0, 1, \dots)$$

を調べればよい. ここで $\delta/\delta u$ は (変分学における) Euler 微分で

$$(1.18) \quad \frac{\delta}{\delta u} = \sum_{j=0}^{\infty} (-D)^j \partial_j$$

で定義される. H だけから定まる L の係数が l_1, \dots, l_{3-m} であることに注意すれば, H だけから定まる標準密度は (最も一般的な場合には) $\rho_{-1}, \dots, \rho_{m-3}$ だけであり それ以後の密度は保存則を積分して得られる変数 $\sigma_{-1}, \sigma_0, \dots$ (ρ_k が定数なら $\sigma_k = 0$ とする) を含み, 例えば ρ_k ($k \geq m-2$) は $\sigma_{-1}, \dots, \sigma_{k-m+1}$ に依存する可能性がある. 従ってすべての k について条件 (1.17) を一斉に調べることは出来ない.

標準密度として同値類の中で簡単な代表元をとれば十分である. 例えば

$$\rho_0 \sim \frac{1}{m} q^m \partial_{m-1} H$$

である.

一般的に言って標準密度 ρ_k が保存密度であるという条件だけから発展方程式 (1.4) の H を完全に定めるのは難しい. Sokolov Shabat の原論文でもここでは詳しく述べないがもう少し付加的な条件 (その条件は方程式のハミルトン形式と関連し, 少し雑に言うなら標準密度の一部が自明な保存密度になることに対応している) をつけて分類を行っている. その意味では “発展方程式が無数個の保存則を持てば可積分である” という標語は (形式的な意味でも) 正しくないのかも知れない. 本稿ではもう少し条件を強めてすべての標準密度が自明になる発展方程式を考える (実際は最初の有限個しか調べることが出来ない). 但し発展方程式 (1.4) にの従属変数 u を任意関数 φ を用いて $u = \varphi(v)$ と変換すると v に関する発展方程式が得られるから, このような変換の自由度は除いて考えることにする. 標準密度がすべて自明となる発展方程式は形式的線形化可能な方程式と呼ばれる. 例えば定数係数線形発展方程式では標準密度はすべて定数であり自明である.

例 1.1. 3 階の発展方程式

$$(1.19) \quad u_t = H(u, u_1, u_2, u_3) = u_3 + g(u, u_1, u_2)$$

を考える. この場合 $\rho_{-1} = 1$ であるから $\sigma_{-1} = 0$. 標準密度 $\rho_0, \rho_1, \rho_2, \rho_3$ は以下で与えられる.

$$\begin{aligned}\rho_0 &= \frac{1}{3}\partial_2 H = \frac{1}{3}\partial_2 g \\ \rho_1 &= \text{Res } H_*^{\frac{1}{3}} \sim \frac{1}{3}\partial_1 g - \frac{1}{9}(\partial_2 g)^2 \\ \rho_2 &= \frac{2}{3}\sigma_0 + \text{Res } H_*^{\frac{2}{3}} \sim \frac{2}{3}D^{-1}(\rho_0)_t + \frac{4}{9}(\rho_0)^3 - \frac{1}{9}\partial_1 g \rho_0 + \frac{2}{3}\partial_0 g \\ \rho_3 &= \sigma_1 = D^{-1}(\rho_1)_t \sim \frac{1}{9}D^{-1}[3\partial_1 g - (\partial_2 g)^2]_t\end{aligned}$$

これらがすべて自明な密度になるような発展方程式は従属変数の変換の自由度を除けば定数係数線形のもの以外では α, β, γ を定数として

$$\begin{aligned}u_t &= u_3 - \frac{3}{4}\frac{u_2^2}{u_1} + \alpha u_1 + \beta u + \gamma \\ u_t &= u_3 - \frac{3}{4}\frac{u_2^2}{u_1 + 1} + \alpha u_1 + \gamma\end{aligned}$$

に限る. 更に最初の発展方程式は $\gamma = 0$ ならば $v_t = v_3 + \alpha v_1 + \beta v$ において変換 $v = (u_1)^{1/2}$ を施せば得られる. 第二の方程式も $\gamma = 0$ なら方程式 $v_t = v_3 + \alpha v_1$ から変換 $v = (u_1 + 1)^{1/2}$ により得られる. これらの結果は [SS] の結果に含まれる. Burgers' 方程式の3階のヒエラルキ $-u_t = u_3 + 3uu_2 + 3u_1^2 + 3u^2u_1$ については ρ_1, ρ_2, ρ_3 は自明だが $\rho_0 = u$ が自明でない保存密度となって, 我々が扱うクラスには属していない. 従って必ずしも“実際に線形化出来る方程式のクラス \neq 形式的に線形化可能な方程式のクラス”である

例 1.2. 4階の発展方程式

$$(1.20) \quad u_t = H(u, u_1, u_2, u_3, u_4) = u_4 + g(u, u_1, u_2, u_3).$$

$\rho_{-1} = 1$ であり $\rho_0, \rho_1, \dots, \rho_4$ は

$$\begin{aligned}\rho_0 &= \frac{1}{4}\partial_3 H = \frac{1}{4}\partial_3 g \\ \rho_1 &= \text{Res } H_*^{\frac{1}{4}} \sim \frac{1}{4}(\partial_2 g - \frac{3}{8}(\partial_3 g)^2) \\ \rho_2 &= \text{Res } H_*^{\frac{1}{2}} \sim \frac{1}{2}\partial_1 g - \frac{1}{4}\partial_2 g \partial_3 g + \frac{1}{16}(\partial_3 g)^3 \\ \rho_3 &= \frac{3}{4}\sigma_0 + \text{Res } H_*^{\frac{3}{4}} = \frac{3}{16}D^{-1}(\partial_3 g)_t + \text{Res } H_*^{\frac{3}{4}} \\ \rho_4 &= \sigma_1 = D^{-1}(\rho_1)_t = -\frac{3}{8}(\partial_3 g)_t + \frac{1}{4}D^{-1}[\partial_2 g - \frac{3}{8}(\partial_3 g)^2]_t\end{aligned}$$

で与えられる. これらがすべて自明な保存密度となるような発展方程式は従属変数の変換の自由度を除けば

$$u_t = u_4 + \alpha u_3 + \beta u_2 + \gamma u_1 + \delta u + \epsilon + k u_1^2$$

に限ることが判る. 更に

$$\rho_5 = \frac{5}{8}\sigma_2 + \text{Res } H_*^{5/4}$$

が自明になる条件という条件を付加すると $k=0$ が得られる. よって発展方程式 (1.20) で形式的線形化可能なものは (従属変数の変換の自由度を除いて) 定数係数線形の発展方程式しかないことになる.

2. 連立方程式系の場合

簡単の為未知関数 $u(x, t)$ と $v(x, t)$ に関する 2 元の連立発展方程式

$$(2.1) \quad \begin{aligned} u_t &= H_1(u, u_1, \dots, u_m, v, v_1, \dots, v_m), \\ v_t &= H_2(u, u_1, \dots, u_m, v, v_1, \dots, v_m) \end{aligned}$$

を考える. ここで $u_j = (\partial/\partial x)^j u(x, t)$, $v_j = (\partial/\partial x)^j v(x, t)$ とする. 一般の連立方程式系については [MSY] を参照せよ. この場合 \mathcal{R} として $u = u_0, v = v_0, u_1, v_1, \dots$ の有理関数体で $Du_i = u_{i+1}$, $Dv_i = v_{i+1}$ を満たす微分 D を持つものを考え, $H_1, H_2 \in \mathcal{R}$ とする. 時間発展微分は $f = (f_1, f_2) \in \mathcal{R} \times \mathcal{R}$ に対して

$$(2.2) \quad \partial_f = \sum_{j \geq 0} (D^j f_1) \frac{\partial}{\partial u_j} + \sum_{j \geq 0} (D^j f_2) \frac{\partial}{\partial v_j}$$

で定義される. 特に $H = (H_1, H_2)$ とすれば ∂_H は方程式 (2.1) 上での時間微分である. 更に 2×2 行列に値を持つ微分作用素 H_* は各成分が

$$(2.3) \quad \begin{aligned} (H_*)_{11} &= \sum_{j \geq 0} \frac{\partial H_1}{\partial u_j} D^j, & (H_*)_{12} &= \sum_{j \geq 0} \frac{\partial H_1}{\partial v_j} D^j \\ (H_*)_{21} &= \sum_{j \geq 0} \frac{\partial H_2}{\partial u_j} D^j, & (H_*)_{22} &= \sum_{j \geq 0} \frac{\partial H_2}{\partial v_j} D^j \end{aligned}$$

で与えられるものとする. $\mathcal{R} \times \mathcal{R}$ の元を縦ベクトルと思えば H_* は $\mathcal{R} \times \mathcal{R}$ に自然に左から作用する. そのとき方程式 (2.1) の symmetry は $f = (f_1, f_2) \in \mathcal{R} \times \mathcal{R}$ であって方程式

$$(2.4) \quad \partial_H(f) - H_*f = f_t - H_*f = 0$$

を満たすものとして定義される. ここで時間発展微分は成分毎に作用する. recursion operator L は 2×2 行列に値を持つ \mathcal{R} 係数の微分積分作用素 (各成分が \mathcal{R} 係数の D に関する形式的 Laurent 級数で与えられる) で方程式

$$(2.5) \quad \partial_H(L) - [H_*, L] = L_t - [H_*, L] = 0$$

を満たすものである. ここで時間発展微分は行列の成分毎に作用する. Sokolov, Shabat の方法は Mikhailov, Shabat, Yamilov ([MSY]) によりある制限条件のもとで連立方程式系にも適用できることが示された.

まず T を行列に値を持つ可逆な微分積分作用素として $\hat{L} = TLT^{-1}$ とおくと方程式 (2.5) が

$$(2.6) \quad \partial_H(\hat{L}_t) - [\hat{H}_*, \hat{L}] = \hat{L}_t - [\hat{H}_*, \hat{L}] = 0$$

と書き換えられることに注意する. ここで

$$(2.7) \quad \hat{H}_* = T_t T^{-1} + T H_* T^{-1}$$

である. そこで H_* の主項 (D に関する次数が最も高い項) の係数行列

$$(2.8) \quad H_*(m) = \begin{pmatrix} \frac{\partial H_1}{\partial u_m} & \frac{\partial H_1}{\partial v_m} \\ \frac{\partial H_2}{\partial u_m} & \frac{\partial H_2}{\partial v_m} \end{pmatrix}$$

が相異なる固有値 λ_1, λ_2 を持ち

$$(2.9) \quad \Lambda = \text{diag}(\lambda_1, \lambda_2) = T_0 H_*(m) T_0^{-1}$$

と対角化出来ると仮定する. その時以下の命題が成り立つ.

命題 ([MSY]). T_0 を主項とする (次数 0 の) 可逆な作用素

$$(2.10) \quad T = T_0(E + T_{-1}D^{-1} + T_{-2}D^{-2} + \dots)$$

で $\hat{H}_* = T_t T^{-1} + T H_* T^{-1}$ の D に関する係数行列がすべて対角行列となるものが存在する. ここで E は 2×2 単位行列, T_k ($k = -1, -2, \dots$) は対角成分をもたない行列である.

上記で得られた変換行列作用素 T を用いて \hat{H}_* を対角化して \hat{L} に対する方程式 (2.6) を解くと \hat{L} もすべての係数行列が対角行列となる.

変換作用素 T と \hat{H}_* の各係数行列は方程式

$$(2.11) \quad \hat{H}_* T = T H_* + T_t$$

を解くことにより同時に計算される. この命題により recursion operator に対する方程式 (2.5) は (2.6) に帰着し, それは対角成分に関する二つのスカラー方程式

$$(2.12) \quad (L^a)_t - [H^a, L^a] = 0, \quad (L^b)_t - [H^b, L^b] = 0$$

と同値である. ここで L^a, L^b は次数 1 の (スカラー) 作用素, H^a, H^b は次数 m (以下) の作用素である. また t 微分は ∂_H で計算されることに注意する. (2.12) は単独方程式の場合の recursion operator に対する方程式 (1.7) と同じ形で異なる点は, (1.7) においては H_* は微分作用素であるのに対して (2.12) では T で変換した為 H^a, H^b は一般的には任意階数の積分作用素 (D の負冪の項) を含むことである. このような相違は (計算は面倒になるけれども) 本質的ではなく, 単独方程式の場合と全く同様にして (2.12) の二つの方程式からそれぞれ標準密度 ρ_k^a ($k = -1, 0, 1, \dots$) と ρ_k^b ($k = -1, 0, 1, \dots$) が定義され,

方程式 (1.1) が recursion operator を許容する為の条件はこれらの標準密度がすべて方程式 (2.1) の保存密度となることである. 標準密度 ρ_k^a, ρ_k^b がすべて自明な保存密度になる発展方程式を形式的線形化可能と呼ぶことにしてそのような方程式を数え上げる問題を考える. その為の条件は標準密度の u と v に関する Euler 微分が共に 0 となることである (v に関する Euler 微分は (1.17) と同様に定義される).

例 2.1. 連立 3 階の発展方程式

$$(2.13) \quad \begin{aligned} u_t &= H_1 = u_3 + f(u, u_1, v, v_1) \\ v_t &= H_2 = -v_3 - g(u, u_1, v, v_1) \end{aligned}$$

を考える. ここで f, g が u, u_1, v, v_1 にのみ依存するのは計算上の便宜である. また v_3 の係数が -1 であるのはこの場合全く本質的ではなく任意の定数でよい. 但し形式的線形化可能性よりも少し弱い条件 ([SS], [MSY] ではそれを完全可積分性の条件としている) の下で, 偶数階の方程式 (この場合は奇数階なので関係ない) を考えると主項の係数行列 $H_*(m)$ の固有値の和は 0 でなければならず, その場合とあとで対比する為 v_3 の係数を -1 と採った. 標準密度を計算するには \hat{H}_* を D に関する Laurent 級数として求める必要がある. 実際 $\rho_k^a = \text{Res} (L^a)^k, \rho_k^b = \text{Res} (L^b)^k$ を計算するには \hat{H}_* に関しては D^{2-k} まで T に関しては D^{-k-1} までの係数行列が必要となりそれらは方程式 (2.11) から同時に求められる.

$$\begin{aligned} H^a &= D^3 + \varphi_2 D^2 + \varphi_1 D + \varphi_0 + \varphi_{-1} D^{-1} + \dots \\ H^b &= -D^3 - \psi_2 D^2 - \psi_1 D - \psi_0 - \psi_{-1} D^{-1} - \dots \end{aligned}$$

とおけば

$$\begin{aligned} \varphi_2 &= 0 (= \psi_2), \quad \varphi_1 = \partial_{u_1} f, \quad \varphi_0 = \partial_u f, \quad \varphi_{-1} = -\frac{1}{2} \partial_{v_1} f \partial_{u_1} g (= \psi_{-1}), \\ \varphi_{-2} &= \partial_{v_1} f D \partial_{u_1} g + \frac{3}{4} D \partial_{v_1} f \partial_{u_1} g - \frac{1}{2} (\partial_{v_0} f \partial_{u_1} g + \partial_{v_1} f \partial_{u_0} g), \dots \end{aligned}$$

であり, ψ_k は φ_k において置き換え $f \leftrightarrow g, u \leftrightarrow v$ を行うことにより得られる. これらより標準密度 ρ_k^a は

$$\begin{aligned} \rho_0^a &= \frac{1}{3} \varphi_2 = 0 (= \rho_0^b), \quad \rho_1^a = \text{Res} (H^a)^{\frac{1}{3}} \sim \frac{1}{3} \varphi_1 = \frac{1}{3} \partial_{u_1} f \\ \rho_2^a &= \frac{2}{3} \sigma_0^a + \text{Res} (H^a)^{\frac{2}{3}} \sim \frac{2}{3} \varphi_0 = \frac{2}{3} \partial_u f \\ \rho_3^a &= \sigma_1^a + \varphi_{-1} = \frac{1}{3} D^{-1} [\partial_{u_1} f]_t - \frac{1}{2} \partial_{v_1} f \partial_{u_1} g \\ \rho_4^a &= \frac{2}{3} \sigma_2^a + \text{Res} (H^a)^{\frac{4}{3}} = \frac{4}{9} D^{-1} [\partial_u f]_t + \text{Res} (H^a)^{\frac{4}{3}}, \dots \end{aligned}$$

等となる. 同様に ρ_k^b を計算することが出来る. 計算機 (による数式処理) の助けを借りた単純だが非常に長い計算によって $\rho_1^a, \rho_1^b, \dots, \rho_6^a, \rho_6^b$ が自明な保存密度になるような発展方程式 (2.13) を定めることが出来る. 但し, 入れ替え $f \leftrightarrow g, u \leftrightarrow v$ の不定性を除い

て考えることにする. 結果だけを述べると, そのような方程式は定数係数線形の発展方程式以外では本質的に以下の二つのタイプに限る.

$$(I) \quad \begin{cases} u_t = u_3 + a_{22}v_1^2 + \alpha_1 u_1 + \beta_1 v_1 + \gamma_1 u + \delta_1 v + k_2 \\ v_t = -v_3 - \alpha_2 u_1 - \beta_2 v_1 - k_2 \end{cases}$$

$$(II) \quad \begin{cases} u_t = u_3 + \alpha_1 u_1 + \beta_1 v_1 + \bar{f}(v, v_1) \\ v_t = -v_3 - \beta_2 v_1 - \delta_2 v - k_2 \end{cases}$$

ここで $a_{22}, \alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2, \delta_1, \delta_2, k_1, k_2$ は任意定数, $\bar{f}(v, v_1)$ は v, v_1 に関する任意関数である. (II) の発展方程式は H_1 に任意関数を含んでいるが, H_2 は u 及びその微分を含まず, 2 番目の方程式は v に関して単独の発展方程式である. また標準密度 $\rho_1^a, \dots, \rho_6^b$ はすべて定数となっている.

3. 数式処理 (REDUCE) による計算

分類を実行する為にしたパッケージについて簡単に説明しよう.

数式処理の利用においてまず問題になるのは必要な計算に現れる基本的な表現式を (使用しようとする) 数式処理システムでどのように表すかということである. まず単独方程式の場合を考える. \mathcal{R} の生成元 u, u_1, u_2, \dots を $u(0), u(1), u(2), \dots$ で表す (REDUCE で operator u ; と宣言をしておけばよい, 但し $u(k)$ を x で微分すると $u(k+1)$ になると定義しておく). 次に $f \in \mathcal{R}$ が u, u_1, \dots, u_m に依存するとき f を $f(0, 0, \dots, 0)$ (0 の数は $m+2$ 個) で表す. 更に

$$(3.1) \quad D^k \partial_0^{k_0} \dots \partial_m^{k_m} f$$

を $f(k, k_0, \dots, k_m)$ で表す. (3.1) を u_j で微分する (∂_j を作用する) には帰納的な関係式

$$(3.2) \quad \partial_j D = D \partial_j + \partial_{j-1}$$

を用いればよい. 微分則 (3.2) を定義するには REDUCE の微分のパッケージを部分的に書き換える. 微分則 (3.2) をもつ表現 (3.1) を定義するには

doperator f;

と宣言するだけでよい. そのとき $f(k, k_0, \dots, k_m)$ は (任意の正の整数 m に対して) (3.1) を表すことになる. 例えば

df(f(3,0,0,0,0),u(3));

と評価すると

f(3,0,0,0,1) + 3*f(2,0,0,1,0) + 3*f(1,0,1,0,0) + f(0,1,0,0,0)

が返る.

次に代入の規則を定義する. 以後簡単のため例によって説明しよう. 例えば上記の f について $\partial_3 f = u^3$ という関係式を代入したければ

dlet(f(0,0,0,0,1),u(0)3);**

とすればよい. この **dlet** 文は普通の **let** 文とは異なり **f(1,1,0,0,1)** を評価すると $D\partial_0 u^3$ すなわち $6*u(1)*u(0)$ が返る. **doperator** を含む代入も可能である. 例えば $\partial_3 f$ が u, u_1 だけの関数であるなら **doperator p;** と宣言しておいて

```
dlet(f(0,0,0,0,1),p(0,0,0));
```

とすればよい. 最後に上記の代入文を解除するときは

```
dclear(f(0,0,0,0,1));
```

とすればよい.

次に連立発展方程式の場合を説明する. $f \in \mathcal{R}$ が u, v, \dots, u_m, v_m に依存するとき f を $1 + 2(m + 1)$ 個の 0 をもつ $f(0, \dots, 0)$ で表す. 更に

$$(3.3) \quad D^n \partial_{v_0}^{l_0} \dots \partial_{v_m}^{l_m} \partial_{u_0}^{k_0} \dots \partial_{u_m}^{k_m} f$$

を $f(n, k_0, \dots, k_m, l_0, \dots, l_m)$ で表す. (3.2) に対応する帰納的な微分則は

$$(3.4) \quad \partial_{u_j} D = D \partial_{u_j} + \partial_{u_{j-1}}, \quad \partial_{v_j} D = D \partial_{v_j} + \partial_{v_{j-1}}$$

である. 微分則 (3.4) をもつ表現 (3.3) を定義するには

```
doperator2 f
```

と宣言すればよい. 代入とその解除は単独の場合と同様に **dlet** 文と **dclear** 文によって行われる. 発展方程式が 3 元連立になった場合は従属変数として w をとり, $f \in \mathcal{R}$ が $u, v, w, \dots, u_m, v_m, w_m$ に依存するとき f を $1 + 3(m + 1)$ 個の 0 をもつ $f(0, \dots, 0)$ で表す. このような表現式で (3.4) と同様な微分則をもつものを定義するには

```
doperator3;
```

と宣言すればよい. 以上のプログラムは **ddiff.rl** を呼べば作動する. 現在扱えるのは 3 元連立の発展方程式までであるが, これらを一般化して任意個数の連立発展方程式を扱えるようにすることはそれほど困難ではないであろう. しかしその場合二重の添字をもった $u(i, j)$ の導入が必要となるなど, 式が繁雑になり, 特に必要性も感じていないのでそのような一般化は行っていない.

発展方程式の分類を行う為には, 今までに述べた表示 (**doperator** 等) に対する様々な **procedure** を作る必要がある.

まず **diffop.rl** には Euler 微分 ((1.17) 式), Frechet Jacobian H_* ((1.5) 及び (2.3) 式で定義) 等を計算するプログラムが含まれている. f の v に関する Euler 微分を計算するときは **eul(f, v)**; とすればよい. 単に **eul f**; とすると u に関する Euler 微分が計算される (勿論 **eul(f, u)** でも同じ). 一般的にいて Euler 微分の計算は結果が非常に長い式になることが多く, その一部分だけを取り出すなどの工夫が必要である. Frechet Jacobian を計算する **procedure fj(h)** は h がスカラーなら (1.5) を, $h = \text{list}(h_1, h_2)$ なら行列に値をもつ微分作用素 (2.3) を返す.

もう一つのパッケージ **seki.rl** はスカラーまたは行列に値をもつ微分積分作用素 (D に関する形式的 Laurent 級数) の作用素としての (漸近的な) 積や (スカラー) 作用素の分数冪を計算するプログラムからなる. 例えば

```
seki(f, g, m);
```

は作用素 f, g の積の最初の $m + 1$ 項をとったものを返す. 但し, f, g は \mathcal{R} 係数の D に関する有限 Laurent 級数である.

以上のようなパッケージを用いることにより, 標準密度が自明な保存密度となるような発展方程式を分類するのだが, まだ計算のすべてが自動化されている訳ではない. 例えば f の Euler 微分が 0 となれば f は定数でなければ $f = Dg$ と書ける筈だがその g を計算

する有効な一般的アルゴリズムがまだよく判らない(この計算は方程式の形がある程度決まったとき, 先の方の標準密度を求める際等に必要になる). 例えば f が u, v, u_1, v_1, \dots の多項式の場合には Gel'fand-Dickey によるものがよく知られている(例えば [GZ]). しかし f が多項式ではなく有理式になるとそのアルゴリズムは有効ではないし, 更に f のなかに **doperator** で表される表現が含まれているときは役に立たない.

計算はワークステーション Sun3 上に加古富志雄氏(現在奈良女子大学理学部情報科学科)によりインプリメントされた REDUCE(Ver. 3.3)によって行った. プログラム作成に際しても加古氏から教わったことは多い. この場を借りてお礼を述べたい.

REFERENCES

- [O] P.J. Olver, *Evolution equations possessing infinitely many symmetries*, J. Math. Phys. **18**(6) (1977), 1212-1215.
- [SS] V.V. Sokolov and A.B. Shabat, *Classification of integrable evolution equations*, Math. Phys. Review **4** (1984), 221-280.
- [MSY] A.V. Mikhailov, A.B. Shabat and R.I. Yamilov, *The symmetry approach to the classification of non-linear evolution equations. Complete list of integrable systems*, Russian Math. Surveys **42**(4) (1987), 1-63.
- [FW1] A. Fujimoto and Y. Watanabe, *Polynomial evolution equations of not normal type admitting nontrivial symmetries*, Phys. Lett. **136A**(6) (1989), 294-299.
- [FW2] A. Fujimoto and Y. Watanabe, *Classification of the third-order polynomial evolution equations of not normal type admitting nontrivial symmetries*, Memoirs Fac. Eng. Hiroshima Univ. **10**(2) (1989), 23-33.
- [FW3] A. Fujimoto and Y. Watanabe, *Classification of the fifth-order polynomial evolution equations of not normal type admitting nontrivial symmetries*, Memoirs Fac. Eng. Hiroshima Univ. **11**(1) (1991), 1-12.
- [GZ] V.P. Gert and A.Yu. Zharkov, *Computer generation of necessary integrability conditions for polynomial-nonlinear evolution systems*, Proceedings of the international symposium on symbolic and algebraic computation (1990), 251-254.