

2階導関数を使う RUNGE-KUTTA 型公式の探索

京大 数理解 三井斌友

0. はじめに

筆者は数式処理 Symbolic and Algebraic Manipulation (SAM) システムを専門にしているものではないが、常微分方程式の初期値問題の数値解析の研究に SAM を用いた、user としての経験を報告させていただくのが、本稿の目的である。数理解では、DEC System 2020 に REDUCE-2 を導入しているが、そのすべての機能を用いたのではなく、はなはだ負しい経験があるが、幸い問題が丁度 SAM にのせやすい size であることもあって、有効に利用できている。

1. p-stage RUNGE-KUTTA 公式

微分方程式の初期値問題

$$(1.1) \quad \frac{dy}{dx} = f(x, y), \quad x \geq x_0, \quad (1.2) \quad y(x_0) = y_0.$$

を数値積分する one-step method として、まず Taylor 展開を用いることが考えられる。step-size を h として、(1.1) (1.2) の

解 $y(x)$ の $x_0 + h$ での値の近似値を

$$(1.3) \quad y_1 = y(x_0) + hf(x_0, y(x_0)) + \frac{1}{2}h^2f'(x_0, y(x_0)) + \cdots + \frac{1}{n!}h^n f^{(n-1)}(x_0, y(x_0))$$

によって定める. $y(x_0 + h)$ の Taylor 展開との比較で分るように、この方法の局所打ち切り誤差 local truncation error は $O(h^{n+1})$ であるから、方法は order n と呼ばれる.

(1.3) に現われる導関数 f' , f'' , $\dots$, $f^{(n-1)}$ は x と y の

$$\frac{d}{dx}f(x, y(x)), \left(\frac{d}{dx}\right)^2f(x, y(x)), \dots, \left(\frac{d}{dx}\right)^{n-1}f(x, y(x))$$

の意味で、微分方程式 (1.1) が具体的に与えられるば (すなわち $f(x, y)$ の関数形が与えられるば)、chain rule によって計算できる. しかし、高階導関数を実際に求めることは、 x と y 自身を手間をとるので、Taylor 法は実用性に乏しいとされている.

(注) $f(x, y)$ の関数形を与えて、 x の導関数を関数形で求めることも、SAM で可能なことである. 事実、SAM システムの発達は、Taylor 法に対する上のような主張を解消する可能性があると指摘されている [5].

そこで、解関数の 1 階導関数である $f(x, y)$ の evaluation を何度が行ない、 x と y を組み合わせて、Taylor 法と同じ order の accuracy をもつ方法が考えられた. これを総称して RUNGE-KUTTA 法と呼んでいる. one-step の間に行なう、 $f(x, y)$ の evaluation の回数を stage 数というが、 p -stage RK 公式は、次のように表わされる.

$$(1.4) \quad \begin{cases} y_1 = y_0 + h \sum_{i=1}^p a_i k_i \\ k_i = f(x_0 + c_i h, y_0 + h \sum_j b_{ij} k_j) \end{cases}$$

ここで, $b_{ij} = 0$ for $j \geq i$ なら explicit type, $b_{ij} = 0$ for $j > i$ なら semi-explicit type, そうでないとき implicit type と呼ぶ。

ふつう使われている RK 公式 (classical RK 公式) は, $p=4$ の explicit type である。

RK 公式を決定するには, k_i を変数関数として Taylor 展開し, y_1 を h のべきで整理して, $y(x_0+h)$ の Taylor 展開と比較し, 必要な order まで h のべきの係数が一致するようにして得られるような a_i, b_{ij}, c_i の関係式を解くことが必要である。これは全く代数的計算であるが, order が上っていくと大変な量になる。これを非常にみとあしよく整理したのが J. C. BUTCHER の一連の研究 ([1] ~ [4]) で, RK 公式に対して画期的展望を与えた。その結果, 当時未解決であった 5-stage 公式の order はいくつにできるかという問題に, order 4 という解答を与えたばかりでなく, 10-stage までの公式の attainable order を決定した。さらに implicit type や semi-explicit type を新しい概念として導入し, それらが公式の stability について新しい展望を与えた。

BUTCHER の研究では, 実用上は無理がある (linear multi-step method である PC 法などと比べて) と思われている implicit

type が積極的に用いられていることに注意したい。

2. 2階導関数を使う RUNGE-KUTTA 公式

RK 法が 1 階導関数 $f(x, y)$ の evaluation に限定されていたのは自然であるが、少し立場を変えて、2 階導関数

$$g(x, y) \equiv f_x(x, y) + f_y(x, y)f(x, y)$$

の evaluation を許すとする、one-step methods の新しい series ができる。

(注) $g(x, y)$ の具体的な形は、次の例をみていただきた

い。

$$f(x, y) = \begin{bmatrix} f_1(y_1, y_2) \\ f_2(y_1, y_2) \end{bmatrix} = \begin{bmatrix} (a_1 - b_1 y_2) y_1 \\ (a_2 - b_2 y_1) y_2 \end{bmatrix} \quad (a_i, b_i: \text{constants})$$

ならば

$$g(x, y) = \begin{bmatrix} a_1^2 y_1 + (b_2 y_1 + b_1 y_2 - 2a_1 - a_2) b_1 y_1 y_2 \\ a_2^2 y_2 + (b_2 y_1 + b_1 y_2 - a_1 - 2a_2) b_2 y_1 y_2 \end{bmatrix}.$$

2 階導関数 $g(x, y)$ の evaluation を含む公式を初めて作ったのは占部 [11] のようである。その後新谷 [7, 8], CASH [6] などがみられる。占部, CASH は線形多段法に、新谷は RK 法に analogous とみることができる。

われわれは、 $g(x, y)$ を含む一般の RK 型公式, (p, g) -stage 公式を次のように定式化する。

$$(2.1) \begin{cases} y_1 = y_0 + h \sum_{i=1}^p \mu_i k_i + h^2 \sum_{i=1}^q \nu_i K_i, \\ k_i = f(x_0 + \alpha_i h, y_0 + h \sum_{j=1}^p \beta_{ij} k_j + h^2 \sum_{j=1}^q \gamma_{ij} K_j), \\ K_i = g(x_0 + \rho_i h, y_0 + h \sum_{j=1}^p \sigma_{ij} k_j + h^2 \sum_{j=1}^q \tau_{ij} K_j). \end{cases}$$

この公式の理論的可能性として，その attainable order を定めていこう．理論的研究のためには implicit type の方が都合がよいので，(2.1) は implicit としておく． $(p, 0)$ -stage formula は，ふつうの意味の RK 公式 (1.4) になるから， (p, q) -stage formula はひとつの一般化となる，という．

(注) 戸田 [10] は，RK 公式の 5 次の極限公式を扱っている (本講義録中の戸田・小野西氏の論文も同趣旨)．ここでは 5-stage の公式で，任意に決められるパラメータとある極限值に移行すると， $k_i - k_{i+1}$ のある値は 0 に近くなることが示されている．もし，それをそのまま採用すれば 2 階導関数を含むゆれゆれの公式に近いものになると思われる．[10] では RK 法の通常の約束で， $f(x, y)$ の evaluation だけを行なうことにしているので， $g(x, y)$ は現れない．

ゆれゆれの RK 公式の研究のためには，まず $y(x_0 + h)$ の h のべき級数への展開を， g と s の偏導関数を用いて表わしておくことが都合である．そのために，次の定理を示すことができる．

Theorem 1. $y(x_0+h)$ が m 次, h のべき級数として

$$(2.2) \quad y(x_0+h) = y_0 + hf_0 + \sum_{r=2}^m \frac{\kappa_{r-2}}{r!} h^r + O(h^{m+1})$$

と展開されるように. すると, κ_l は

$$(2.3) \quad \begin{cases} \kappa_0 = g_0 \\ \kappa_l = [D_0^l g]_0 + \sum_{s=1}^{l-1} \left\{ \sum_{t=1}^{m(l,s)} \frac{l!}{s!(l-s-t)!} B_{s,t} \left(\frac{\kappa_0}{2}, \dots, \frac{\kappa_{s-1}}{s+1} \right) [D_0^{l-s-t} \left(\frac{\partial}{\partial y} \right)^t g]_0 \right\}, \\ l=1, 2, \dots, m-2 \end{cases}$$

と与えられる. ここで suffix 0 は $x=x_0$ における evaluation を表わし, $D_0 = \frac{\partial}{\partial x} + f_0 \frac{\partial}{\partial y}$, $m(l,s) = \min(s, l-s)$, $B_{s,t}$ は (s,t) 次の BELL 多項式である.

(注) 微分方程式 (1.1) が連立である場合, f_y は Jacobian 行列であって, g は $f_x + f_y \cdot f$ の順序で書かれるのが正しい. しかし公式の order の研究のためには, 単独方程式で考えれば十分であるので, y に関する偏導関数と f や g との積は, 順序を問わない.

(2.3) 式において, κ_l を決めるには, 右辺で必要なのは κ_{l-2} までであることに注意すると, これは SAM システム にかける algorithm になっている. ただし, BELL 多項式が求められている必要があるが, BELL 多項式もそれを生成する algorithm が知られている. すなわち

$$\begin{aligned} \sum_{t=1}^s x_t B_{s,t}(x_1, \dots, x_s) a^{t+1} + \sum_{t=1}^s \left(\sum_{k=1}^s x_{k+1} \frac{\partial B_{s,t}}{\partial x_k} \right) a^t \\ = \sum_{t=1}^{s+1} B_{s+1,t}(x_1, \dots, x_{s+1}) a^t. \end{aligned}$$

ここで両辺は a の多項式として恒等式とする.

また (2.3) 式から, κ_l は, $[D_0^s (\frac{\partial}{\partial y})^t g]_0$ (s, t は非負整数) の形の項のいくつかの積の 1 次結合 (係数は実は正整数になる) である. そこで $\prod_{s,t} [D_0^s (\frac{\partial}{\partial y})^t g]_0$ のことを elementary differentials と呼ぶことにする.

筆者は REDUCE-2 によって, BELL 多項式を 15 次まで, (2.3) によって κ_l を $l=8$ まで, えいぞれ求めた. 特に各 κ_l を求めて, えいに含まれている elementary differentials の数を調べておくと, 役に立つ.

なお, Th. 1 の考え方は, $y(x_0+h)$ を f とその偏導関数を用いて h のべき級数として表わすことにも応用できる.

Theorem 2. $y(x_0+h)$ が h のべき級数として

$$(2.4) \quad y(x_0+h) = y_0 + \sum_{r=1}^m \frac{\lambda_{r-1}}{r!} h^r + O(h^{m+1})$$

と展開されるとする. すると, λ_l は

$$(2.5) \quad \begin{cases} \lambda_0 = f_0, \\ \lambda_l = [D_0^l f]_0 + \sum_{s=1}^{l-1} \left\{ \sum_{t=1}^{m(l,s)} \frac{l!}{s! (l-s-t)!} B_{s,t} \left(\frac{\lambda_1}{2}, \frac{\lambda_2}{3}, \dots, \frac{\lambda_s}{s+1} \right) [D_0^{l-s-t} (\frac{\partial}{\partial y})^t f]_0 \right\}, \\ l = 1, 2, \dots, m-1 \end{cases}$$

によって与えられる.

BUTCHER は, ゆえゆゆのと は違, た意味で elementary differentials を定義して, $y(x_0+h)$ の h のべき級数展開の係数をえいによって表わしたが, Th. 2 は BUTCHER の結果の別の形の表現

となっている.

(注) (2.2) と (2.4) は $y(x_0+h)$ の h のべき級数としての
=通りの表現である. として実際, g に関する elementary
differentials を構成している $[D_0^s (\frac{\partial}{\partial y})^t g]_0$ に対しては,

$$[D_0^s g]_0 = [D_0^{s+1} f]_0 + \sum_{r=0}^{s-1} \binom{s}{r} [D_0^{s-r} f]_0 \cdot [D_0^r f_y]_0.$$

$$[D_0^s g_y]_0 = [D_0^{s+1} f_y]_0 + \sum_{r=0}^s \binom{s}{r} [D_0^{s-r} f_y]_0 [D_0^r f_y]_0 + \sum_{r=0}^{s-1} \binom{s}{r} [D_0^{s-r} f]_0 [D_0^r f_{yy}]_0.$$

⋮

の恒等式がなりたつから, K_2 は, 上の substitution を行な
うことによって, f に関する elementary differentials を使
った表現に変換できる. 筆者は, DEC 2020 の software
FILCOM (= 2 つの file の内容の比較) を用いて, λ_2 と, 変
換された K_2 とを比較し, 同一であることを確認できた.

3. (1, g)-stage formula

一般の (p, g) -stage formula を扱うことは相当困難なので,
次のような特殊の形の implicit (1, g)-stage formula をまず考察
しよう.

$$\begin{cases} y_1 = y_0 + \mu_1 h k_1 + \sum_{i=1}^g \nu_i h^2 K_i, \\ k_1 = f(x_0 + \alpha_1 h, y_0) \\ K_i = g(x_0 + \rho_i h, y_0 + \sigma_{i1} h k_1 + \sum_j \tau_{ij} h^2 K_j), \quad i=1, \dots, g \end{cases}$$

$\alpha_1 = 0$, $\mu_1 = 1$, $\sigma_{i1} = \rho_i$ であることはすぐ分るので, 実は公式

は

$$(3.1) \quad \begin{cases} y_1 = y_0 + hf_0 + \sum_{i=1}^g \nu_i h^2 K_i, \\ K_i = g(x_0 + p_i h, y_0 + p_i h f_0 + \sum_j \tau_{ij} h^2 K_j), \quad i=1, \dots, g \end{cases}$$

である。これは, explicit type の場合, 新谷[7]と一致する。

パラメータ ν_i, p_i, τ_{ij} を決定する条件を導かねばならない。

Th.1 と同じ考え方で, 次のことがいえる。

Theorem 3. 各 i について, K_i を h のべき級数として

$$(3.2) \quad K_i = \sum_{l=0}^{m-2} \frac{\kappa_{il}}{l!} h^l + O(h^{m-1})$$

と表わす。すると κ_{il} は

$$(3.3) \quad \begin{cases} \kappa_{i0} = g_0, \\ \kappa_{il} = p_i^l [D_0^l g]_0 + \sum_{s=1}^{l-1} \left\{ \sum_{t=1}^{m(l,s)} \frac{l!}{s!(l-s-t)!} p_i^{l-s-t} B_{s,t}(\sum_j \tau_{ij} \kappa_{j0}, \dots, \right. \\ \left. s \sum_j \tau_{ij} \kappa_{j,s-1}) [D_0^{l-s-t} (\frac{\partial}{\partial y})^t g]_0 \right\}, \quad l=1, 2, \dots, m-2. \end{cases}$$

で与えられる。

Th.1 の結果とあわせると, 各 l に対して

$$(3.4) \quad (l+1)(l+2) \sum_{i=1}^g \nu_i \kappa_{il} = \kappa_l, \quad l=0, 1, \dots, m-2$$

がなりたつ。そして (2.3) と (3.3) を比較すれば, κ_l と κ_{il} の

なかに含まれる elementary differentials は同一のものとなるこ

とが分るから, パラメータに関する条件はすべて

$$(\text{正整数}) \times (\nu_i, p_i, \tau_{ij} \text{ の多項式}) = (\text{正整数})$$

となる。すなわちパラメータを決定する方程式の個数は, κ_0

から κ_{l-1} までに含まれる elementary differentials の数に等しく,

これと $(1, g)$ -stage formula で決定すべきパラメータの個数との関係を考察すれば、少なくとも到達できる order, すなわち attainable order の下限は決まる.

$m(l)$ を, x_l を含んでいる elementary differentials の数とし, $M(l)$ を

$$M(l) = \sum_{j=0}^l m(j)$$

と定義する. 表 1 では, $m(l)$, $M(l)$ を $l \leq 8$ まであげてある.

また implicit $(1, g)$ -stage formula において決定すべきパラメータの数を, g の関数として $N_I(g)$ で表わすと

$$N_I(g) = g(g+2)$$

である. 或る g に対して, $M(l) \leq N_I(g)$ をみたす最大の整数を $\bar{l}$ とすれば, 公式は少なくとも order $(\bar{l}+2)$ とすることができる. 同様に explicit $(1, g)$ -stage formula において決定すべきパラメータの数を $N_E(g)$ とすると

$$N_E(g) = g(g+3)/2$$

である. 上と同様に l^* を, $M(l) \leq N_E(g)$ から決定すると, この formula は少なくとも order (l^*+2) とするることができる. この関係は表 2 にあげてある.

$\bar{l}$ や l^* は, attainable order の下限に関係するもので, 上限の方を与えるには, (3.4) からえられる決定方程式を実際に解いてみなければならぬ.

Tab. 1

ℓ	$m(\ell)$	$M(\ell)$
0	1	1
1	1	2
2	2	4
3	3	7
4	6	13
5	9	22
6	17	39
7	26	65
8	46	111

Tab. 2

q	$N_I(q)$	$\bar{\ell}$	$N_E(q)$	ℓ^*
1	3	1	2	1
2	8	3	5	2
3	15	4	9	3
4	24	5	14	4
5	35	5	20	5
6	48	6	27	5
7	63	6	35	6
8	80	7	44	6
9	99	7	54	6
10	120	8	65	7

4. Implementation

具体的に決定方程式を書き下すためには, REDUCE-2によって (3.3) を実行させるのであるが, ための program は次のようになる.

- (1) 準備 1. BELL 多項式の input. (別の file で用意してもよい.)
- (2) 準備 2. $\kappa_{i\ell}$, $[D_0^s(\frac{\partial}{\partial y})^t g]_0$, ρ_i , ψ_i , τ_{ij} の array の用意.
 $[D_0^s(\frac{\partial}{\partial y})^t g]_0$ は, $DGY(I, J)$ という array を用意し, その各々の値を $\underbrace{DIG Y \cdots Y}_{J \text{ 個}}$ のように assign してやることにした.
- (3) 準備 3. factorial の計算 procedure.
- (4) κ_{i0} の計算 (assignment).
- (5) BELL 多項式の各変数へ, 前の結果の代入.
- (6) (3.3) による代数計算.
- (7) (5), (6) の反復.

(8) 結果を, 別の file に output.

program の実際は, 本稿の最後に添えてあるので参照していただきたい.

5. むすび

SAM システム, ここでは REDUCE-2 は, 解析にとって強力な道具である. 少なくともこれがないと, われわれの RK 型公式を扱うという気もおこらなかつたであらう. 更に, パラメータの決定方程式を解く (解けるかどうかの検討も含めて) となると, terminal を傍におきながら, trial and error を重ねるしか, 方法はみつからない. 一般的に数学研究にもっとも, SAM システムを応用すべきであると思う. 筆者もなるべく宣伝に努めているが, SAM の専門家の方々にお願いすることとして, 二・三のことをあげると

(1) SAM システムは, interactive であるべきことは当然だが, file との間で input, output ができることが是非要る.

(2) manual を familiar なものにすること. 文法書を読んでも良い文章ができるとは限らないと同様, SAM の manual も tutorial なもの, example の多いものが良さうである. 勿論, 一旦使いこんだあとでは reference manual も要る. やがては数学教科書の style もそういう方向に変わっていくように,

まず先鞭をつけられたいと思う。

(3) language を使いやすくすること。私たちのように数値解析に従事し、FORTRAN などによる programming の経験のあるものは、SAM language の理解も早いと思うが、そうでない人達には、やはりその修得には hesitating なようである。

更に、ある部分は SAM、その結果を用いた別の部分は数値処理 (FORTRAN など) という program を可能にする higher level language ができればと期待される。

R e f e r e n c e s

- [1] Butcher, J.C., Coefficients for the study of Runge-Kutta integration processes, J. Austral. Math. Soc., 3(1963), 185-201.
- [2] —————, On Runge-Kutta processes of high order, J. Austral. Math. Soc., 4(1964), 179-194.
- [3] —————, Implicit Runge-Kutta processes, Math. Comput., 18(1964), 50-64.
- [4] —————, On the attainable order of Runge-Kutta methods, Math. Comput., 19(1965), 408-417.
- [5] Carnahan, B. et al., Applied Numerical Methods, John Wiley, New York, 1969. 邦訳, FORTRAN による数値計算法, 科学技術出版社, 東京, 近刊.
- [6] Cash, J.R., High order methods for the numerical integration of ordinary differential equations, Numer.Math., 30(1978), 385-409.
- [7] Shintani, H., On one-step methods utilizing the second derivative, Hiroshima Math. J., 1(1971), 349-372.

- [8] Shintani, H., On explicit one-step methods utilizing the second derivative, Hiroshima Math. J., 2(1972), 353-368.
- [9] Mitsui, T., Runge-Kutta type integration formulas including the evaluation of the second derivative, I, in preparation.
- [10] Toda, H., On the truncation error of a limiting formula of Runge-Kutta methods, Researches of Electrotech. Labor., No. 772, 1977.
- [11] Urabe, M., An implicit one-step method of high-order accuracy for the numerical integration of ordinary differential equations, Numer. Math., 15(1970), 151-164.

附録 REDUCE-2 programs

I. BELL多項式の生成.

```

00100  COMMENT  THIS IS A REDUCE-2 FILE TO CALCULATE THE BELL'S POLYNOMIAL;
00200
00300  ARRAY B(15,15),X(15),KAP(15);
00400  FACTOR A;
00500
00600  X(1):=X1; X(2):=X2; X(3):=X3; X(4):=X4; X(5):=X5; X(6):=X6; X(7):=X7;
00700  X(8):=X8; X(9):=X9; X(10):=X10; X(11):=X11; X(12):=X12; X(13):=X13;
00800  X(14):=X14; X(15):=X15;
00900
01000  B(1,1):=X1;
01100
01200  NMAX:=14;
01300  FOR N:=1:NMAX DO
01400    BEGIN K:=0; NP1:=N+1;
01500      FOR I:=1:N DO K:=K+X(1)*B(N,I)*A**((I+1)+(FOR J:=1:N
01600        SUM(X(J+1)*DF(B(N,I),X(J)))))*A**I;
01700
01800      WRITE ("N,N") ",K;
01900      COEFF(K,A,KAP);
02000      FOR I:=1:NP1 DO B(NP1,I):=KAP(I);
02100    END;
02200
02300  COMMENT OUTPUT THE RESULTS;
02400
02500  OFF ECHO; OFF ALLFAC; OUT BELLOT.RED;
02600  WRITE "COMMENT  THIS IS A REDUCE-2 FILE TO STORE THE BELL'S ";
02700  WRITE "          POLYNOMIAL AS THE RESULTS RUNNING THE FILE BELPLY.RED;";
02800  WRITE "ARRAY BELL(15,15);";
02900  OFF NAT;
03000  FOR N:=1:NMAX+1 DO
03100    FOR I:=1:N DO WRITE "BELL(",N,"",I,""):=",B(N,I);
03200  SHUT BELLOT.RED;

```

II. BELL 多項式 (前の program の結果)

```

COMMENT THIS IS A REDUCE-2 FILE TO STORE THE BELL'S
          POLYNOMIAL AS THE RESULTS RUNNING THE FILE HELPLY.RED;

ARRAY BELL(15,15);
BELL(1,1):=X1$
BELL(2,1):=X2$
BELL(2,2):=X1**2$
BELL(3,1):=X3$
BELL(3,2):=3*X1*X2$
BELL(3,3):=X1**3$
BELL(4,1):=X4$
BELL(4,2):=4*X1*X3 + 3*X2**2$
BELL(4,3):=6*X1**2*X2$
BELL(4,4):=X1**4$
BELL(5,1):=X5$
BELL(5,2):=5*X1*X4 + 10*X2*X3$
BELL(5,3):=10*X1**2*X3 + 15*X1*X2**2$
BELL(5,4):=10*X1**3*X2$
BELL(5,5):=X1**5$
BELL(6,1):=X6$
BELL(6,2):=6*X1*X5 + 15*X2*X4 + 10*X3**2$
BELL(6,3):=15*X1**2*X4 + 60*X1*X2*X3 + 15*X2**3$
BELL(6,4):=20*X1**3*X3 + 45*X1**2*X2**2$
BELL(6,5):=15*X1**4*X2$
BELL(6,6):=X1**6$
BELL(7,1):=X7$
BELL(7,2):=7*X1*X6 + 21*X2*X5 + 35*X3*X4$
BELL(7,3):=21*X1**2*X5 + 105*X1*X2*X4 + 70*X1*X3**2 + 105*X2**2*X3$
BELL(7,4):=35*X1**3*X4 + 210*X1**2*X2*X3 + 105*X1*X2**3$
BELL(7,5):=35*X1**4*X3 + 105*X1**3*X2**2$
BELL(7,6):=21*X1**5*X2$
      ⋮
(以下 続く)

```

III. $f(x_0 + h)$ の h のべき級数への展開 (2.3)

```

00100  COMMENT  REDUCE-2 FILE TO PREPARE TO GENERATE THE TAYLOR SERIES
00200  EXPANSION OF SECOND TYPE;
00300
00400
00500  ARRAY BELL(15,15);
00600
00700  BELL(1,1):=X1$
00800
00900  BELL(2,1):=X2$
01000
01100  BELL(2,2):=X1**2$
01200
01300  BELL(3,1):=X3$
01400

```

⋮ (BELL多項式のinput)

```

07700  BELL(8,7):=28*X1**6*X2$
07800
07900  BELL(8,8):=X1**8$
08000
08100
08200  ARRAY DGY(8,7),KAPPA(8); OFF ECHO;
08300
08400  DGY(0,0):=G;
08500  DGY(0,1):=GY;
08600  DGY(0,2):=GYY;
08700  DGY(0,3):=GYYY;
08800  DGY(0,4):=GYYYYY;
08900  DGY(0,5):=GYYYYYY;
09000  DGY(0,6):=GYYYYYYY;
09100  DGY(0,7):=GYYYYYYYY;
09200
09300  DGY(1,0):=DG;
09400  DGY(1,1):=DGY;
09500  DGY(1,2):=DGYG;
09600  DGY(1,3):=DGYGY;
09700  DGY(1,4):=DGYYYY;
09800  DGY(1,5):=DGYYYYY;
09900  DGY(1,6):=DGYYYYYY;
10000  DGY(1,7):=DGYYYYYYY;
10100
10200
10300  DGY(2,0):=D2G;
10400  DGY(2,1):=D2GY;
10500  DGY(2,2):=D2GYY;
10600  DGY(2,3):=D2GYGY;
10700  DGY(2,4):=D2GYYYY;
10800  DGY(2,5):=D2GYYYYY;
10900  DGY(2,6):=D2GYYYYYY;
11000  DGY(2,7):=D2GYYYYYYY;
11100
11200  DGY(3,0):=D3G;
11300  DGY(3,1):=D3GY;
11400  DGY(3,2):=D3GYY;
11500  DGY(3,3):=D3GYGY;
11600  DGY(3,4):=D3GYYYY;
11700  DGY(3,5):=D3GYYYYY;
11800  DGY(3,6):=D3GYYYYYY;
11900  DGY(3,7):=D3GYYYYYYY;
12000

```

⋮ (DGY(I, J) の assignment)

```

16500
16600  COMMENT  INTEGER PROCEDURE FOR FACTORIAL OF AN INTEGER N;
16700  INTEGER PROCEDURE FAC(N);
16800  BEGIN INTEGER N;
16900  4:=1$
17000  L1:  IF N=0 THEN RETURN N;
17100  N:=N-1$
17200  N:=N-1$
17300  GO TO L1
17400  END;
17500

```

(続く)

```

17600 FACTOR G,DG,D2G,D3G,D4G,D5G,D6G,D7G,GY,DGY,D2GY,D3GY,D4GY,D5GY,D6GY,
17700 GYY,DGYG,D2GYG,D3GYG,D4GYG,D5GYG,GYYY,DGYYY,D2GYYY,D3GYYY,D4GYYY,
17800 GYYYY,DGYYYY,D2GYYYY,D3GYYYY;
17900
18000
18100 COMMENT GENERATING PROCEDURE;
18200
18300 KAPPA(0):=DGY(0,0);
18400
18500 X1:=KAPPA(0)/2;
18600
18700 L0:=1;
18800 INTEGER M,N,L0M1,L0*MN,MINI;
18900 KAPPA(L0):=DGY(L0,0);
19000 IF L0<=1 THEN GO TO LAB1;
19100 L0M1:=L0-1;
19200 FOR M:=1:L0M1 DO
19300 BEGIN MINI:=L0-M;
19400 IF M<=L0-M THEN MINI:=M;
19500 FOR N:=1:MINI DO BEGIN L0MN:=L0-M-N;
19600 WRITE "(",M," ",N,") ";
19700 KAPPA(L0):=KAPPA(L0)+
19800 HELL(M,N)*DGY(L0MN,N)*FAC(L0)/(FAC(M)*FAC(L0MN)) END;
19900 END;
20000 LAB1: WRITE "(",L0,") ",KAPPA(L0);
20100 X2:=KAPPA(1)/3;
20200
20300 L0:=2;
20400 KAPPA(L0):=DGY(L0,0);
20500 IF L0<=1 THEN GO TO LAB2;
20600 L0M1:=L0-1;
20700 FOR M:=1:L0M1 DO
20800 BEGIN MINI:=L0-M;
20900 IF M<=L0-M THEN MINI:=M;
21000 FOR N:=1:MINI DO BEGIN L0MN:=L0-M-N;
21100 WRITE "(",M," ",N,") ";
21200 KAPPA(L0):=KAPPA(L0)+
21300 HELL(M,N)*DGY(L0MN,N)*FAC(L0)/(FAC(M)*FAC(L0MN)) END;
21400 END;
21500 LAB2: WRITE "(",L0,") ",KAPPA(L0);
21600 X3:=KAPPA(2)/4;
21700
21800 L0:=3;
21900 KAPPA(L0):=DGY(L0,0);
22000 IF L0<=1 THEN GO TO LAB3;
22100 L0M1:=L0-1;
22200 FOR M:=1:L0M1 DO
22300 BEGIN MINI:=L0-M;
22400 IF M<=L0-M THEN MINI:=M;
22500 FOR N:=1:MINI DO BEGIN L0MN:=L0-M-N;
22600 WRITE "(",M," ",N,") ";
22700 KAPPA(L0):=KAPPA(L0)+
22800 HELL(M,N)*DGY(L0MN,N)*FAC(L0)/(FAC(M)*FAC(L0MN)) END;
22900 END;
23000 LAB3: WRITE "(",L0,") ",KAPPA(L0);
23100 X4:=KAPPA(3)/5;

```

: (以下 繰り返し)

IV. 前の program の結果

```

COMMENT THIS IS A REDUCE-2 FILE TO STORE THE RESULTS OF
TAYLOR SERIES EXPANSION OF SECOND TYPE BY TYEXGO.RED;

```

```

KAPPA(0):=G$
KAPPA(1):=DG$

```

(続く)

```

KAPPA(2):=G*GY + D2G$
KAPPA(3):=3*G*DG + DG*GY + D3G$
KAPPA(4):=3*G**2*GY + G*GY**2 + 6*G*D2GY + 4*DG*DG + D2G*GY + D4G$
KAPPA(5):=15*G**2*DGYY + 10*G*DG*GY + 8*G*GY*DG + 10*G*D3GY + DG*GY
**2 + 10*DG*D2GY + 5*D2G*DG + D3G*GY + D5G$
KAPPA(6):=15*G**3*GYYY + 18*G**2*GY*GY + 45*G**2*D2GY + 60*G*DG*
DGYY + 15*G*D2G*GY + G*GY**3 + 21*G*GY*D2GY + 18*G*DG**2 + 15*G*
D4GY + 10*DG**2*GY + 10*DG*GY*DG + 20*DG*D3GY + D2G*GY**2 + 15*D2G*
D2GY + 6*D3G*DG + D4G*GY + D6G$
KAPPA(7):=105*G**3*DGYYY + 105*G**2*DG*GYYY + 120*G**2*GY*DGYY + 84*G
**2*DG*GY + 105*G**2*D3GY + 66*G*DG*GY*GY + 210*G*DG*D2GY + 105*
G*D2G*DGYY + 21*G*D3G*GY + 15*G*GY**2*DG + 45*G*GY*D3GY + 105*G*DG
*D2GY + 21*G*D5GY + 70*DG**2*DGYY + 35*DG*D2G*GY + DG*GY**3 + 31*DG*
GY*D2GY + 28*DG*DG**2 + 35*DG*D4GY + 12*D2G*GY*DG + 35*D2G*D3GY +
J3G*GY**2 + 21*D3G*D2GY + 7*D4G*DG + D5G*GY + D7G$
KAPPA(8):=105*G**4*GYYYY + 225*G**3*GY*GYYY + 84*G**3*GYY**2 + 420*G
**3*D2GYYY + 840*G**2*DG*DGYYY + 210*G**2*D2G*GYYY + 81*G**2*GY**2*
GY + 465*G**2*GY*D2GY + 624*G**2*DG*DGYY + 252*G**2*D2GY*GY + 210
*G**2*D4GY + 280*G*DG**2*GYYY + 508*G*DG*GY*DGYY + 360*G*DG*DG*GY
+ 560*G*DG*D3GY + 113*G*D2G*GY*GY + 420*G*D2G*D2GY + 168*G*D3G*
DGYY + 28*G*D4G*GY + G*GY**4 + 49*G*GY**2*D2GY + 92*G*GY*DG**2 + 85
*G*GY*D4GY + 248*G*DG*D3GY + 168*G*D2GY**2 + 28*G*D6GY + 66*DG**2*GY
*GY + 280*DG**2*D2GY + 280*DG*D2G*DGYY + 56*DG*D3G*GY + 18*DG*GY**
2*DG + 75*DG*GY*D3GY + 172*DG*DG*DGYY + 56*DG*D5GY + 35*D2G**2*GY
+ D2G*GY**3 + 43*D2G*GY*D2GY + 40*D2G*DG**2 + 70*D2G*D4GY + 14*D3G*
GY*DG + 56*D3G*D3GY + D4G*GY**2 + 28*D4G*D2GY + 8*D5G*DG + D6G*GY
+ D8G$

```

V. (3.3)によつて K_{il} を求める.

```

00100  COMMENT  THIS IS A REDUCE-2 FILE TO CALCULATE IMPLICIT (1,0)-STAGE
00200          FORMULA EMPLOYING THE BELL POLYNOMIALS;
00300
00400  ARRAY BELL(15,15); OFF ECHO;
00500
00600  BELL(1,1):=X1$
00700
00800  BELL(2,1):=X2$
00900
01000  BELL(2,2):=X1**2$
01100

```

∴ (BELL多項式のinput)

```

07900
08000  ON ECHO;
08100  ARRAY KAPPA(2,8), DGY(8,7), RHO(2), NU(2), TAU(2,2),
08200          CAPTAU(2,5), CAPSIG(2,4), CAPRHO(2,3), QQ(2,8);
08300  OFF ECHO;
08400
08500  DGY(0,0):=G;
08600  DGY(0,1):=GY;
08700  DGY(0,2):=GY*GY;
08800  DGY(0,3):=GYYY;
08900  DGY(0,4):=GYYYY;

```

∴ (DGY(I,J)のassignment)

```

16400  DGY(8,6):=D8GYYYYYY;
16500  DGY(8,7):=D8GYYYYYY;
16600
16700  RHO(1):=RHO1;
16800  RHO(2):=RHO2;
16900
17000  TAU(1,1):=TAU11;
17100  TAU(1,2):=TAU12;
17200

```

(続<)

```

17300 TAU(2,1):=TAU21;
17400 TAU(2,2):=TAU22;
17500
17600 NU(1):=NU1;
17700 NU(2):=NU2;
17800
17900 CAPTAU(1,0):=CAPTAU10;
18000 CAPTAU(2,0):=CAPTAU20;

```

・ (各arrayへのassignment)
・

```

20900
21000 COMMENT INTEGER PROCEDURE FOR FACTORIAL OF AN INTEGER N;
21100 INTEGER PROCEDURE FAC(N);
21200 BEGIN INTEGER N;
21300 N:=1;
21400 L1: IF N=0 THEN RETURN N;
21500 N:=N*N;
21600 N:=N-1;
21700 GO TO L1
21800 END;
21900
22000 FACTOR G,DG,D2G,D3G,D4G,D5G,D6G,D7G,D8G,GY,DGY,D2GY,D3GY,D4GY,D5GY,D6GY,D7GY,
22100 GYY,DGY,D2GY,D3GY,D4GY,D5GY,D6GY,GYYY,DGY,D2GY,D3GY,D4GY,D5GY,
22200 GYYY,DGY,D2GY,D3GY,D4GY,D5GY,D6GY,D7GY,D8GY,D9GY,D10GY,D11GY,D12GY,
22300
22400 IMAX:=2;
22500
22600 FOR I:=1:IMAX DO KAPPA(I,0):=DGY(0,0);
22700
22800 L0:=1;
22900 INTEGER LOM1,M,N,LOMMN,MINI;
23000 LOM1:=L0-1;
23100 FOR I:=1:IMAX DO QQ(I,L0):=L0*(FOR J:=1:IMAX SUM(TAU(I,J)*KAPPA(J,LOM1)));
23200 I:=1;
23300 X1:=QQ(I,1);
23400 KAPPA(I,L0):=RHO(I)**L0*DGY(L0,0);
23500 IF L0<=1 THEN GO TO LAB11;
23600 FOR M:=1:LOM1 DO
23700 BEGIN MINI:=L0-M;
23800 IF M<=MINI THEN MINI:=M;
23900 FOR N:=1:MINI DO
24000 BEGIN LOMMN:=L0-M-N;
24100 KAPPA(I,L0):=KAPPA(I,L0)+
24200 RHO(I)**LOMMN*BELL(M,N)*DGY(LOMMN,N)*FAC(L0)/(FAC(M)*FAC(LOMMN))
24300 END;
24400 END;
24500 LAB11: WRITE ("I,",I,"",L0,"") ",KAPPA(I,L0);
24600
24700 I:=2;
24800 X1:=QQ(I,1);
24900 KAPPA(I,L0):=RHO(I)**L0*DGY(L0,0);
25000 IF L0<=1 THEN GO TO LAB12;
25100 FOR M:=1:LOM1 DO
25200 BEGIN MINI:=L0-M;
25300 IF M<=MINI THEN MINI:=M;
25400 FOR N:=1:MINI DO
25500 BEGIN LOMMN:=L0-M-N;
25600 KAPPA(I,L0):=KAPPA(I,L0)+
25700 RHO(I)**LOMMN*BELL(M,N)*DGY(LOMMN,N)*FAC(L0)/(FAC(M)*FAC(LOMMN))
25800 END;
25900 END;
26000 LAB12: WRITE ("I,",I,"",L0,"") ",KAPPA(I,L0);
26100
26200 L0:=2;
26300 INTEGER LOM1,M,N,LOMMN,MINI;
26400 LOM1:=L0-1;
26500 FOR I:=1:IMAX DO QQ(I,L0):=L0*(FOR J:=1:IMAX SUM(TAU(I,J)*KAPPA(J,LOM1)));
26600 I:=1;
26700 X1:=QQ(I,1); X2:=QQ(I,2);
26800 KAPPA(I,L0):=RHO(I)**L0*DGY(L0,0);

```

(以下繰り返す)