

数値解析への数式処理の2,3の応用

三井 斌友

(福井大学工学部応用理学教室)

1. はじめに

コンピュータによる数式・代数処理 (symbolic and algebraic manipulation—SAM, 最近では computer algebra と称せられることもある) ソフトウェアの発達は, TSS 環境下で相当程度研究者の助手の役割を果たせる機能を持つまでに至ってきた。

この機能は, 従来数学研究や, 物理学・天文学といった方面で役に立ってきたが, 同じくコンピュータの機能の一部である数値計算, その理論としての数値解析と関係をつけられないであろうか, またそのインターフェイスはどのようなか, このような問題意識で, 2・3の試行の結果をお話しするのが, 本講演の目的である。

このようなインターフェイスは従来全く意識せしめなかったわけではなく, 文献 [1] によれば, 若干の論文が紹介されているし, また, 数式処理特集と銘うった文献 [2] では,

(1) 流体の運動をモデル化する, 偏微分方程式系の導出

(2) それらの有限差分スキームの構成

などに, SAMが有効に用いられている事例が述べられている。

一般に非線型性をもつ大規模問題となる, このような問題とも関連させて, 筆者らによる本研究会のもう一つの講演は, 非線型代数方程式系に, 混合処理 (hybrid manipulation) なるインターフェイスを設定したが, そのようなアプローチも考えられるもののひとつであろう。

一方, 数学解析としては解けていることでも, 現実には到底実現できない故に, 近似手法をもちこむ数値解析の立場からは, 代数的に exact な計算である SAM をもちこんでくることは, 「飛び道具とは卑怯なり!」という感想もあるようである。しかし, 問題解決に可能な限り最善の手段をつくすという科学の立場から, あまりセクショナリズムにならず, 現在考えられるインターフェイスを追求してみたい。

以下述べる事例は, 規模としては, ほなほ小さいものであるが, こうした追求の端緒となれば幸いである。なお計算は, SAMは REDUCE 3.0, 数値処理は FORTRAN を用いている。京都大学数理解析研究所電子計算機室の助力に感謝申し上げる。

2. Vandermonde 行列を係数とする連立一次方程式

数値線型代数として周知のことであるが, 連立一次方程式を数値的に解くには係数行列の条件数 (condition number) が問題となる。

n 次正方行列と, n 次元ベクトル x, b による方程式

$$(1) \quad Ax = b$$

を解く際, A, b にそれぞれ $\Delta A, \Delta b$ の誤差 (観測による誤差, 丸め誤差など) が含まれているならば, 実際に解いている方程式は

$$(2) \quad (A + \Delta A)(x + \Delta x) = b + \Delta b$$

における $x + \Delta x$ についてである。 Δx が正確に知られるなら, (1) 式の解 x が分るのであるが, それほととも不可能であって, 我々が知りうるのは, ノルムではかった相対誤差 $\|\Delta x\|/\|x\|$ の上からの評価でしかない。

$$(3) \frac{\|\Delta x\|}{\|x\|} \leq \frac{\text{cond}(A)}{1 - \text{cond}(A) \cdot \|\Delta A\| / \|A\|} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right)$$

ここに $\text{cond}(A)$ は、行列 A の条件数で

$$(4) \text{cond}(A) = \|A\| \cdot \|A^{-1}\|$$

ノルムとしてユークリッドノルムをとれば、 $\text{cond}(A)$ は、絶対値最大の固有値と絶対値最小の固有値の比である。

評価式(3)は不等式であるが、殆んど等式となる b も必ず存在するので、 $\text{cond}(A)$ の大きな係数行列については、 Δx は非常に大きくなることを覚悟しなければならぬ。たとえ $\Delta A = 0$ であっても、 $\|\Delta x\| / \|x\|$ は、 $\|\Delta b\| / \|b\| \cdot (\text{データの誤差})$ の $\text{cond}(A)$ 倍程度である。

条件数の大きな行列としては、いくつか知られているが、Vandermonde 行列もその一つである。 n 変数 x_1, x_2, \dots, x_n の Vandermonde 行列 $V(x_1, \dots, x_n)$ とは、 $(n-1)$ 次正方行列で

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix}$$

で与えられるものである。これは、補間多項式を定める下手な方法をとると登場する。すなわち $(n-1)$ 次補間多項式

$$(5) I(x) = \sum_{\ell=0}^{n-1} c_\ell x^\ell$$

を、 n 個のデータ $\{(x_i, f_i)\} (i=1, 2, \dots, n)$ から決めようとするれば、未定係数 $\{c_\ell\}$ についての連立一次方程式

$$(6) \sum_{\ell=0}^{n-1} c_\ell x_i^\ell = f_i, \quad i=1, 2, \dots, n$$

がえられるからである。(6)は

$$(6') V(x_1, x_2, \dots, x_n) c = f$$

とも書く。Vandermonde 行列は条件数が大きいので、このような方法で補間多項式を決定するのは安定性を欠く。ふつうは、ニュートン補間公式とか、ネヴィユ (Neville) の算法などを用いる。しかし、Vandermonde 行列でどうしてもやるという user がいたとして (実際そういう例にぶつかったことがある)、データの相対誤差の程度に解を定める方法はなにか。

データ x_i は、有理数の範囲で誤差がないとするれば、SAM と NM (Numerical Manipulation) の次のようなインターフェイスが考えられる。

(i) x_i の有理数値を代入して、Vandermonde 行列 V を作る。

(ii) f_i は、変数のままで $Vc = f$ を解いて、 c の f による式 (f_i の線型結合式) を作る。

(iii) この式を NM に渡す。(以上まで SAM)

(iv) f_i の具体的な数値を代入して、各 c_ℓ を求める。

数値例として、 $n=10$ で、 $x_1=1, x_2=9/10, x_3=8/10, \dots, x_{10}=1/10$ の場合 (このとき行列の条件数は、 5.6×10^7) を扱ってみる。 $V^{-1}f$ を REDUCE で生成し、

これをFORTRAN notationで出力したのが、下におかげる表1である。但手の成分をB1, B2, ..., cの成分をY1, Y2, ... のように記してある。

```

C *** NINE-DIMENSIONAL CASE ***
C
Y1=-B1+10.*B2-45.*B3+120.*B4-210.*B5+252.*B6-210.*B7+120.
.*B8-45.*B9+10.*B10
Y2=(7129.*B1-71010.*B2+317970.*B3-842520.*B4+1461810.*B5-
.1733004.*B6+1417710.*B7-784920.*B8+275445.*B9-48610.*B10)/
.252.
Y3=(5.*(-32575.*B1+322706.*B2-1435212.*B3+3769752.*B4-
.6465046.*B5+7541100.*B6-6021876.*B7+3204632.*B8-1043307.*
.B9+159826.*B10))/504.
Y4=(25.*(180920.*B1-1779669.*B2+7846074.*B3-20381676.*B4+
.34452306.*B5-39405870.*B6+30600654.*B7-15614604.*B8+
.4765806.*B9-663941.*B10))/2268.
Y5=(125.*(-4275.*B1+41682.*B2-181796.*B3+465976.*B4-
.774402.*B5+866380.*B6-653252.*B7+320376.*B8-92771.*B9+
.12082.*B10))/72.
Y6=(625.*(3013.*B1-29061.*B2+125124.*B3-315756.*B4+
.514854.*B5-562638.*B6+412116.*B7-195084.*B8+54141.*B9-
.6709.*B10))/108.
Y7=(3125.*(-75.*B1+714.*B2-3028.*B3+7508.*B4-11994.*B5+
.12800.*B6-9124.*B7+4188.*B8-1123.*B9+134.*B10))/9.
Y8=(31250.*(145.*B1-1359.*B2+5664.*B3-13776.*B4+21546.*B5
.-22470.*B6+15624.*B7-6984.*B8+1821.*B9-211.*B10))/189.
Y9=(156250.*(-5.*B1+46.*B2-188.*B3+448.*B4-686.*B5+700.*
.B6-476.*B7+208.*B8-53.*B9+6.*B10))/63.
Y10=(1562500.*(B1-9.*B2+36.*B3-84.*B4+126.*B5-126.*B6+84.
.*B7-36.*B8+9.*B9-B10))/567.

```

表 1. REDUCE による Vandermonde 連立1次方程式の解

この結果をみれば、或る手の値を代入したとき、解cは相対誤差がほぼ計算機イプシロン(machine epsilon)のオーダーでえられることが分る。(そのことは実際、既知のc, 例えば $c_l = 1, l=1, \dots, n$ を与えてfを求め、これを上式に代入して数値解 \tilde{c} を計算し、 $\Delta c = \tilde{c} - c$ を作ることで、確認することもできる。筆者の近刊[5]を参照されたい。)

表2に与えたような、有理数値のベクトルfを用いて、FORTRANにより、LU分解による解と、表1の式による解との比較を試みたのが、表3である。やはりLU分解による数値解は精度が悪く、相対誤差がほぼ 10^{-2} 程度である。これは、machine epsilon が 3.73×10^{-9} , condition number

Right-hand side vector =
1.0000000
0.2500000
0.1111111
0.0625000
0.0400000
0.0277778
0.0204082
0.0156250
0.0123457
0.0100000

表 2.

Exact solution by REDUCE =	Solution =	Difference =	Relative error=
-0.2662698	-0.2733962	7.126E-03	-2.676E-02
7.7989588	7.9932944	-1.943E-01	-2.492E-02
-89.1762580	-91.2850760	2.109E+00	-2.365E-02
552.7049000	564.8927800	-1.219E+01	-2.205E-02
-2067.4026000	-2109.2420000	4.184E+01	-2.024E-02
4888.0423000	4977.3719000	-8.933E+01	-1.828E-02
-7357.5248000	-7477.1227000	1.196E+02	-1.626E-02
6838.6740000	6936.1921000	-9.752E+01	-1.426E-02
-3578.9960000	-3623.2103000	4.421E+01	-1.235E-02
807.1452000	815.6833300	-8.538E+00	-1.058E-02

表 3.

が 5.6×10^7 であることからすれば、当然の結果といえよう。

なお、ここで用いたLU分解のプログラムは、あわせて condition number の推定を与えるようになっていて、それによれば

Singular values for VD =

4.655E+00
2.109E+00
6.506E-01
1.611E-01
3.254E-02
5.269E-03
6.568E-04
5.854E-05
3.257E-06
8.162E-08

1.15×10^8

である。一方 V に対する特異値分解を実施すると表4のようになり、最大特異値と最小特異値の比として、前記の condition number がえられるのである。
(以上数値プログラムは FORSYTHE-MALCOLM-MOLER [4] による。)

表4. V の特異値

以上は単精度演算による結果であるが、倍精度演算にすれば、確かにLU分解による数値解の相対誤差は小さくなるが、それでもこの場合の計算機イアシロン (1.08×10^{-19}) よりはずっと大きくなる。いいかえれば、数値解の有効桁数の減少がみられるはずである。これに対して、表1の数式を倍精度むきに書き直したものをいけば、machine epsilon 程度の相対誤差の解がえられ、こちらも当然のことながら優位である。結果は表5に示した。

Right-hand side vector =	Exact solution by REDUCE =
1.0000000000000000	-0.2662698412698413
0.2500000000000000	7.7989591206853112
0.1111111111111111	-89.1762589758125474
0.0625000000000000	552.7049875248453310
0.0400000000000000	-2067.4026067474314900
0.0277777777777778	4888.0431307039276700
0.0204081632653061	-7357.5236293076902200
0.0156250000000000	6838.6724545785392500
0.0123456790123457	-3578.9958987710310400
0.0100000000000000	807.1451317152375370

Solution =	Difference =	Relative error =
-0.2662698412705930	7.517E-13	-2.823E-12
7.7989591207056285	-2.032E-11	-2.605E-12
-89.1762589760323687	2.198E-10	-2.465E-12
552.7049875261194630	-1.274E-09	-2.305E-12
-2067.4026067518545900	4.423E-09	-2.139E-12
4888.0431307135643700	-9.637E-09	-1.971E-12
-7357.5236293209710100	1.328E-08	-1.805E-12
6838.6724545897722000	-1.123E-08	-1.643E-12
-3578.9958987763485100	5.317E-09	-1.486E-12
807.1451317163154110	-1.078E-09	-1.335E-12

表 5.

これらの結果をとりまとめると、(6)' の解 c に対して、係数行列 V 、データ f に誤差が入った場合の数値解

$$(V + \Delta V)(c + \Delta c) = f + \Delta f$$

について、

$$\frac{\|\Delta c\|}{\|c\|} = K \left(\frac{\|\Delta V\|}{\|V\|} + \frac{\|\Delta f\|}{\|f\|} \right)$$

とおくことにすれば、NMによればふつう $K \approx \text{cond}(V)$ であるが、SAMの結果をとり入れれば $K \approx O(1)$ の解法がえられるということである。では、このようなインターフェイスを設けることが常によいのか、あるいは全てをSAMで実行してしまうのがよいのかとなると、疑問は多い。

まず全てをSAMで実行するというのは、データには本来誤差が含まれているということからみて、殆んど無意味であろう。SAMで実行する部分 (ii)(iii) は、さほど時間をとらない。しかし、REDUCEはFORTRAN notationの出力が可能だが、やはりFORTRAN programとして実行するためには、編集(editing)が不可欠で、元の手間、つまりインターフェイスの手間が結構必要である。そういう意味から、あらゆる連立1次方程式をこのようなSAM-NM過程で解くことは、必ずしも妥当では無い。もともとこの補間問題についていえば、データ自身が固定されて、いくつかの異なるデータの組 $\{f_i\}$ に対して解くときには、有効であろう。

3. Taylor級数法による初期値問題の解法

常微分方程式(系)の初期値問題

$$(7) \quad \frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

の数値解法として、解函数 $y(x)$ が x_0 のまわりで十分滑らか(解析的)であることが分っているとき、Taylor級数法という1段階法(one-step method)がある。ステップ幅を h とすると、 $x_1 = x_0 + h$ における解 $y(x_1)$ は

$$(8) \quad y(x_0 + h) = y(x_0) + h y'(x_0) + \frac{1}{2} h^2 y''(x_0) + \dots + \frac{1}{n!} h^n y^{(n)}(x_0) + \dots$$

と展開され、そこで1階導函数値 $y'(x_0)$ は(7)より $f(x_0, y_0)$ に等しいことを用い、

$$(9) \quad y(x_0 + h) = y_0 + h f(x_0, y_0) + \frac{1}{2} h^2 f'(x_0, y_0) + \dots + \frac{1}{n!} h^n f^{(n-1)}(x_0, y_0) + \dots$$

と書き直される。ここで $f', f'', \dots, f^{(m)}$ は、次のように順次決められる函数であって、 $f(x, y)$ の高階導函数と称している。

$$(10) \quad \begin{cases} f'(x, y) = \frac{d}{dx} f(x, y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f \\ \vdots \\ f^{(m)}(x, y) = \frac{d}{dx} f^{(m-1)}(x, y) = \frac{\partial f^{(m-1)}}{\partial x} + \frac{\partial f^{(m-1)}}{\partial y} \cdot f \end{cases}$$

この関係を用いて、 $y(x_0 + h)$ の近似値 y_1 を

$$(11) \quad y_1 = y_0 + h f(x_0, y_0) + \frac{1}{2} h^2 f'(x_0, y_0) + \dots + \frac{1}{p!} h^p f^{(p-1)}(x_0, y_0)$$

と決めるものを、 p 次Taylor法という。その次数(order)は当然 p である。

p 次Taylor法の適用にあたっては、難点がある。それは $f(x, y)$ が具体的に与えられたとき、(10)に従って逐次高階導函数を解析的形式で与えねばならないが、それは爆発的な手間を要するからである。そこで、たとえば[3]では、次のように述べられている。

In view of the recent development of computer programs which formally differentiate arbitrary symbolic expressions, this argument against the direct Taylor's

expansion, with high-order terms included, may well vanish at some future time.

そこで、今やその時に至ったのか検討してみよう。

この場合のインターフェイスも、前節と同様である。

(i) $f(x, y)$ の函数形をうけとる。

(ii) (i)式に従って、 $f(x, y)$ の高階導函数をつくる。(これは REDUCE であろうと、expression f が先に与えられているとして、LET $DF(y, x) = f$; という statement を書き、 x のあと $f^{(m)} := DF(f^{(m-1)}, x)$; とすれば実現できる。DF は REDUCE において数式微分を行なう operator である。)

(iii) 結果を NM に渡す。(以上まで SAM)

(iv) $f(x, y)$ の高階導函数を、function subprogram として利用しながら、(ii)式で step-by-step integration を行なう。

まずやさしい例から始めよう。

例 3.1 $y' = 2xy$, $y(0) = 1$ (exact solution は $y(x) = e^{x^2}$)

この方程式の右辺は、 y について線型であるので、高階導函数 $f^{(m)}(x, y)$ は $2y \times$ (x の $(m+1)$ 次多項式) となることは容易に示すことができ、実際隣り合う $f^{(m)}$ と $f^{(m-1)}$ の間で、 x の多項式の係数を決定する漸化式もつくることができる。従って、あえて SAM の力を借りなくても数式微分は可能だが、

$f' = 2y(2x^2 + 1)$, $f'' = 4y(2x^3 + 3x)$, $f''' = 4y(4x^4 + 12x^2 + 3)$, $f^{(4)} = 8y(4x^5 + 20x^3 + 15x)$, $f^{(5)} = 8y(8x^6 + 60x^4 + 90x^2 + 15)$, ...

となる。classical Runge-Kutta 法(4次)と対比させた結果が、表 6 である ($h = 1/64$)

Taylor 法は、かなり高階 ($p=20$) まで用意してあるが、局所打切誤差のおおよその基準を与えて、各ステップ毎にとるべき次数を調節しながら進むようにしてある。しかし最低 $p=5$ には固定している。結果をみると、次数は 5 び一貫した Taylor 法がとられ、数値解の誤差の比較では当然 RK より良い。特に、このように発展型の問題では、高い次数がとられるほど、 x が大きくなるにつれ RK との差が歴然としてくる。

計算時間の比較はまだしていない。ひとつには、Taylor 法の subprogram が不必要に“重く”なっているのが、このままでは正確な比較(たとえ $p=5$ の RK 法を対象としたとしても)にならないからである。しかし、どうも RK より余計にかかることは確実なので、簡便化をほかってみよう。 $f^{(m)}$ の x の多項式部分は、明白な因子以外は因数分解ができないので、 x という意味で演算回数は減らせない。そこで高階導函数の間の関係式

$$(12) \quad f^{(m)} = 2(x f^{(m-1)} + m f^{(m-2)}) \quad , \quad m = 2, 3, \dots$$

を使うと、かなり時間の節約になる。しかし、これはあくまでも、この例題のように、 y について線型の場合に可能なことである。

例 3.2 $y' = y(2-y)$, $y(0) = 0.1$ (exact solution は $y(x) = 2 / \{1 + 19e^{-2x}\}$.)

いわゆる logistic law を記述する方程式である。 $\lim_{x \rightarrow \infty} y(x) = 2$ び成り立つことはよく知られている。このように安定な定常解の存在する問題では、定常解に近くなるにつれて、Taylor 法の精度は良くなることを期待される。事実そのとおりで、表 7 の結果をみれば、定常解に漸近するにつれて、相対誤差の減少がおこっている。

Ex. 3.7

**** The results ****

X=	Exact value=	RK=	Rel. err.=	Taylor=	Rel. err.=
0.000000	1.000000000000	1.000000000000		1.000000000000	
0.031250	1.00097703949	1.00097703949	-1.48E-15	1.00097703949 (5)	-1.48E-15
0.062500	1.00391388934	1.00391388934	-1.26E-14	1.00391388934 (5)	-7.71E-15
0.093750	1.00882779971	1.00882779971	-4.40E-14	1.00882779971 (5)	-1.88E-14
0.125000	1.01574770859	1.01574770859	-1.10E-13	1.01574770859 (5)	-3.47E-14
0.156250	1.02471452592	1.02471452592	-2.34E-13	1.02471452592 (5)	-5.58E-14
0.187500	1.03578153702	1.03578153702	-4.53E-13	1.03578153702 (5)	-8.23E-14
0.218750	1.04901493061	1.04901493061	-1.52E-13	1.04901493061 (5)	-1.14E-13
0.250000	1.06449445892	1.06449445892	-1.44E-12	1.06449445892 (5)	-1.53E-13
0.281250	1.08231423903	1.08231423903	-2.44E-12	1.08231423903 (5)	-1.97E-13
0.312500	1.10258370681	1.10258370681	-1.0258370680	1.10258370681 (5)	-2.48E-13
0.343750	1.12542873687	1.12542873687	-6.41E-12	1.12542873687 (5)	-3.07E-13
0.375000	1.15099294469	1.15099294468	-9.99E-12	1.15099294469 (5)	-3.74E-13
0.406250	1.17943918965	1.17943918964	-1.52E-11	1.17943918965 (5)	-4.50E-13
0.437500	1.21095130113	1.21095130111	-2.27E-11	1.21095130113 (5)	-5.36E-13
0.468750	1.24573605306	1.24573605302	-3.30E-11	1.24573605306 (5)	-6.32E-13
0.500000	1.28402541669	1.28402541663	-4.73E-11	1.28402541669 (5)	-7.39E-13
0.531250	1.32607912576	1.32607912567	-6.64E-11	1.32607912576 (5)	-8.59E-13
0.562500	1.37218759362	1.37218759349	-9.19E-11	1.37218759362 (5)	-9.93E-13
0.593750	1.42267522791	1.42267522774	-1.25E-10	1.42267522791 (5)	-1.14E-12
0.625000	1.47790419541	1.47790419516	-1.68E-10	1.47790419541 (5)	-1.31E-12
0.656250	1.53827869765	1.53827869730	-2.23E-10	1.53827869764 (5)	-1.49E-12
0.687500	1.60424982730	1.60424982684	-2.92E-10	1.60424982730 (5)	-1.69E-12
0.718750	1.67632108605	1.67632108542	-3.78E-10	1.67632108605 (5)	-1.92E-12
0.750000	1.75505465696	1.75505465611	-4.85E-10	1.75505465696 (5)	-2.17E-12
0.781250	1.84107853919	1.84107853806	-6.16E-10	1.84107853919 (5)	-2.44E-12
0.812500	1.93509466932	1.93509466782	-7.75E-10	1.93509466932 (5)	-2.75E-12
0.843750	2.03788817332	2.03788817135	-9.68E-10	2.03788817331 (5)	-3.08E-12
0.875000	2.15033791595	2.15033791337	-1.20E-09	2.15033791594 (5)	-3.45E-12
0.906250	2.27342854096	2.27342853761	-1.48E-09	2.27342854096 (5)	-3.85E-12
0.937500	2.40826422640	2.40826422206	-1.80E-09	2.40826422639 (5)	-4.30E-12
0.968750	2.55608441581	2.55608441022	-2.19E-09	2.55608441580 (5)	-4.79E-12
1.000000	2.71828182846	2.71828182128	-2.64E-09	2.71828182844 (5)	-5.32E-12
1.031250	2.89642310168	2.89642309251	-3.17E-09	2.89642310167 (5)	-5.91E-12
1.062500	3.09227247680	3.09227246511	-3.78E-09	3.09227247678 (5)	-6.56E-12
1.093750	3.30781900885	3.30781899400	-4.49E-09	3.30781900883 (5)	-7.26E-12
1.125000	3.54530786122	3.54530784241	-5.31E-09	3.54530786120 (5)	-8.04E-12
1.156250	3.80727634150	3.80727631773	-6.25E-09	3.80727634147 (5)	-8.88E-12
1.187500	4.09659544711	4.09659541713	-7.32E-09	4.09659544706 (5)	-9.80E-12

表 6

これは、4次 Taylor法の epigone である4次 RK でもみられるが、これは顕著ではなく、むしろ累積誤差の効果の方が大きいようである。

この場合には、独立変数 x があらわに含まれていないから

$$\begin{aligned}
 (13) f^{(m)}(x, y) &= \frac{d^m}{dx^m} f(x, y) = \sum_{\ell=0}^m \binom{m}{\ell} y^{(m-\ell)} \cdot \frac{d^\ell}{dx^\ell} (2-y) = y^{(m)} (2-y) - \sum_{\ell=1}^m \binom{m}{\ell} y^{(m-\ell)} \cdot y^{(\ell)} \\
 &= f^{(m-1)}(2-y) - \sum_{\ell=1}^m \binom{m}{\ell} f^{(m-\ell-1)} f^{(\ell-1)}
 \end{aligned}$$

と、比較的簡単にあらわすことができる。ここに、 $\binom{m}{\ell}$ は二項係数、 $f^{(-1)}$ は y と解釈することにする。これは、例 3.1 と同様、多少役に立つ。しかし $f^{(m)}$ を知るには、 $y, f, f', \dots, f^{(m-1)}$ のすべてが必要である。

例 3.3 $y' = x^2 \sin y / (y^2 + 1), y(0) = 0.1$

これは、変数分離形ではあるが、積分は初等函数としては表わせないから、exact solution は知られていない。では全く手がかりがないかということ、こういうタイプには、絵をかいてみるに限る。

(x, y) 平面の各点で、右辺の函数 $x^2 \sin y / (y^2 + 1)$ を計算し、その量が $\tan \theta$ に等しい角度 θ ($-\pi/2 < \theta < \pi/2$) の矢印を書いてみる。要するに、微分方程式は x の方向をもったベクトル場を生成しているとみてやるわけである。すると図1がえられる。これをみれば、 $0 \leq y \leq 4$ の間に一つの安定な定常解と、一つの不安定な定常解 ($y \equiv 0$) のあることがわかる。安定な定常解は $y \equiv \pi$ であることは、すぐに知りよう。従って

$$(13) \lim_{x \rightarrow \infty} y(x) = \pi$$

が成り立つ。

実際、RK と Taylor 法で計算すると、表 8 のような対照がえられる。RK と Taylor との計算値の差も示してあるが、 x の値の大きい方で、それほど顕著な差はない。むしろ、Taylor 法で選択されている次数の変化に注目したい。 $x=2$ の近くで、次数は 7 まで上っている。これは、図1でいえば、 x 軸の近くから出発した解曲線が“離陸”して上昇に移る段階であって、確かに次数を大きくする必要があるだろう。RK との差が大きくなっていくのもこの段階である。従って、このあたりでは、Taylor 法は、相当に良く解を近似しているとみられる。しかし、ここでの両者の差も、(13) の性質のゆえに、 x の大きい方ではもはや有意に効いてこないのである。

Ex. 3.2

**** The results **** $h = 1/32$

X=	Exact value=	RK=	Rel. err.=	Taylor=	Rel. err.=
0.000000	0.100000000000	0.100000000000		0.100000000000	
0.156250	0.13422190967	0.13422190615	-2.62E-08	0.13422190967 (5)	5.54E-12
0.312500	0.17905155625	0.17905154753	-4.87E-08	0.17905155625 (5)	9.55E-12
0.468750	0.23695259326	0.23695257747	-6.66E-08	0.23695259326 (5)	1.03E-11
0.625000	0.31038592556	0.31038590090	-7.94E-08	0.31038592556 (5)	6.62E-12
0.781250	0.40139554346	0.40139550861	-8.68E-08	0.40139554346 (5)	-1.11E-12
0.937500	0.51101957892	0.51101953351	-8.89E-08	0.51101957892 (5)	-9.90E-12
1.093750	0.63862243115	0.63862237605	-8.63E-08	0.63862243114 (5)	-1.51E-11
1.250000	0.78136751833	0.78136745572	-8.01E-08	0.78136751832 (5)	-1.35E-11
1.406250	0.93412604328	0.93412597628	-7.17E-08	0.93412604327 (5)	-6.15E-12
1.562500	1.09003602823	1.09003596023	-6.24E-08	1.09003602823 (5)	1.71E-12
1.718750	1.24165453010	1.24165446404	-5.32E-08	1.24165453011 (5)	5.47E-12
1.875000	1.38232555793	1.38232549586	-4.49E-08	1.38232555794 (5)	4.47E-12
2.031250	1.50725775725	1.50725770026	-3.78E-08	1.50725775725 (5)	1.25E-12
2.187500	1.61397742852	1.61397737701	-3.19E-08	1.61397742852 (5)	-1.46E-12
2.343750	1.70215090954	1.70215086352	-2.70E-08	1.70215090954 (5)	-2.60E-12
2.500000	1.77301664813	1.77301660747	-2.29E-08	1.77301664813 (5)	-2.45E-12
2.656250	1.82871829568	1.82871826017	-1.94E-08	1.82871829567 (5)	-1.73E-12
2.812500	1.87173960581	1.87173957521	-1.64E-08	1.87173960581 (5)	-9.77E-13
2.968750	1.90451932127	1.90451929524	-1.37E-08	1.90451932127 (5)	-4.15E-13
3.125000	1.92923818901	1.92923816716	-1.13E-08	1.92923818901 (5)	-7.78E-14
3.281250	1.94773324465	1.94773322654	-9.30E-09	1.94773324465 (5)	8.55E-14
3.437500	1.96149072897	1.96149071412	-7.57E-09	1.96149072897 (5)	1.41E-13
3.593750	1.97167966607	1.97167965402	-6.11E-09	1.97167966607 (5)	1.40E-13
3.750000	1.97920135860	1.97920134891	-4.90E-09	1.97920135860 (5)	1.15E-13
3.906250	1.98474080023	1.98474079250	-3.90E-09	1.98474080023 (5)	8.45E-14
4.062500	1.98881322419	1.98881321806	-3.08E-09	1.98881322419 (5)	5.68E-14
4.218750	1.99180327524	1.99180327041	-2.43E-09	1.99180327524 (5)	3.49E-14
4.375000	1.99399654464	1.99399654085	-1.90E-09	1.99399654464 (5)	1.89E-14
4.531250	1.99560423693	1.99560423397	-1.48E-09	1.99560423693 (5)	8.00E-15
4.687500	1.99678209287	1.99678209057	-1.15E-09	1.99678209287 (5)	1.06E-15
4.843750	1.99764471179	1.99764471001	-8.92E-10	1.99764471179 (5)	-3.02E-15
5.000000	1.99827628954	1.99827628816	-6.89E-10	1.99827628954 (5)	-5.14E-15

表 7

Ex.3.3**** The results **** $k = 1/32$

X=	RK=	Taylor=	(Order)	Difference=
0.000000	0.10000000000	0.10000000000		
0.156250	0.10012576506	0.10012576506	(5)	3.41E-12
0.312500	0.10101045625	0.10101045625	(5)	1.88E-12
0.468750	0.10345040226	0.10345040226	(5)	-4.25E-13
0.625000	0.10836785802	0.10836785803	(6)	-7.75E-12
0.781250	0.11697535687	0.11697535690	(6)	-3.48E-11
0.937500	0.13105712762	0.13105712777	(6)	-1.45E-10
1.093750	0.15345282343	0.15345282404	(6)	-6.10E-10
1.250000	0.18889228389	0.18889228642	(5)	-2.53E-09
1.406250	0.24535222789	0.24535223778	(7)	-9.89E-09
1.562500	0.33570633023	0.33570636429	(6)	-3.41E-08
1.718750	0.47728749005	0.47728757985	(7)	-8.98E-08
1.875000	0.68263740621	0.68263755767	(7)	-1.51E-07
2.031250	0.94186061780	0.94186078574	(7)	-1.68E-07
2.187500	1.22334690460	1.22334705948	(6)	-1.55E-07
2.343750	1.49741208572	1.49741222040	(5)	-1.35E-07
2.500000	1.74828902389	1.74828913791	(6)	-1.14E-07
2.656250	1.97071493843	1.97071503388	(5)	-9.54E-08
2.812500	2.16459152564	2.16459160533	(5)	-7.97E-08
2.968750	2.33189958666	2.33189965332	(5)	-6.67E-08
3.125000	2.47528188186	2.47528193783	(5)	-5.60E-08
3.281250	2.59744606805	2.59744611527	(4)	-4.72E-08
3.437500	2.70093388989	2.70093392995	(4)	-4.01E-08
3.593750	2.78804974453	2.78804977874	(4)	-3.42E-08
3.750000	2.86085653484	2.86085656425	(4)	-2.94E-08
3.906250	2.92119737610	2.92119740154	(4)	-2.54E-08
4.062500	2.97072428524	2.97072430741	(4)	-2.22E-08
4.218750	3.01092532436	3.01092534378	(4)	-1.94E-08
4.375000	3.04314660344	3.04314662054	(4)	-1.71E-08
4.531250	3.06860804874	3.06860806386	(4)	-1.51E-08
4.687500	3.08841322715	3.08841324053	(4)	-1.34E-08
4.843750	3.10355435042	3.10355436225	(4)	-1.18E-08
5.000000	3.11491407650	3.11491408692	(4)	-1.04E-08
5.156250	3.12326593028	3.12326593937	(4)	-9.09E-09
5.312500	3.12927508438	3.12927509223	(4)	-7.85E-09
5.468750	3.13350089029	3.13350089696	(4)	-6.67E-09
5.625000	3.13640199907	3.13640200463	(4)	-5.56E-09
5.781250	3.13834427509	3.13834427963	(4)	-4.54E-09
5.937500	3.13961112015	3.13961112377	(4)	-3.62E-09
6.093750	3.14041540409	3.14041540690	(4)	-2.80E-09
6.250000	3.14091200273	3.14091200485	(4)	-2.11E-09
6.406250	3.14120997258	3.14120997412	(4)	-1.54E-09
6.562500	3.14138358796	3.14138358905	(4)	-1.09E-09
6.718750	3.14148175028	3.14148175103	(4)	-7.49E-10
6.875000	3.14153557018	3.14153557068	(4)	-4.96E-10
7.031250	3.14156416511	3.14156416543	(4)	-3.17E-10
7.187500	3.14157887785	3.14157887805	(4)	-1.96E-10
7.343750	3.14158620388	3.14158620399	(4)	-1.16E-10
7.500000	3.14158973186	3.14158973193	(4)	-6.67E-11
7.656250	3.14159137389	3.14159137392	(4)	-3.68E-11
7.812500	3.14159211204	3.14159211206	(4)	-1.96E-11
7.968750	3.14159243231	3.14159243232	(4)	-9.98E-12
8.125000	3.14159256636	3.14159256636	(4)	-4.90E-12
8.281250	3.14159262043	3.14159262043	(4)	-2.31E-12
8.437500	3.14159264145	3.14159264145	(4)	-1.05E-12
8.593750	3.14159264931	3.14159264931	(4)	-4.54E-13
8.750000	3.14159265214	3.14159265214	(4)	-1.89E-13
8.906250	3.14159265312	3.14159265312	(4)	-7.54E-14
9.062500	3.14159265344	3.14159265344	(4)	-2.88E-14
9.218750	3.14159265355	3.14159265355	(4)	-1.05E-14
9.375000	3.14159265358	3.14159265358	(4)	-3.68E-15
9.531250	3.14159265359	3.14159265359	(4)	-1.23E-15
9.687500	3.14159265359	3.14159265359	(4)	-3.93E-16
9.843750	3.14159265359	3.14159265359	(4)	-1.20E-16
10.000000	3.14159265359	3.14159265359	(4)	-3.56E-17

表 8.

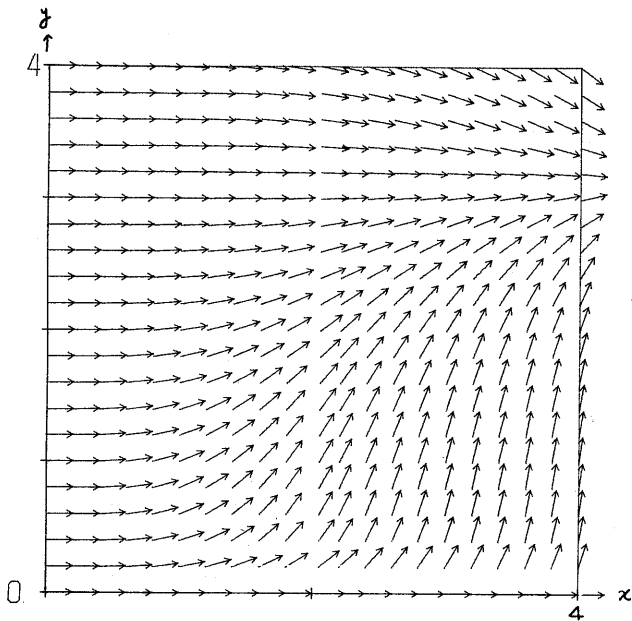


図 1.

以上をまとめると, Taylor法は期待されたおりの精度を得ることができるとは言えない。調べた程度の次数の段階では, RKと有意な差があるには至っていない。しかし, 可変次数(variable order)の公式にすることは, 比較的たやすい。

そこで, 従来の公式でうまくいかない場合にTaylor法はどうか, 検討してみなければならぬ。ひとつはstiff problemであるが, これにとりかかるとは線型安定性を考えなければならぬ。test equation $y' = \lambda y$ にTaylor法を適用すれば, λ の安定性因子 $\pi_p(\lambda)$ は

$$(14) \quad \pi_p(\lambda) = 1 + \lambda + \frac{1}{2}\lambda^2 + \dots + \frac{1}{p!}\lambda^p, \quad (\lambda = \lambda h)$$

となる。RKの場合と異なり, 安定性領域 $\mathcal{R} = \{\lambda \in \mathbb{C}; |\pi_p(\lambda)| < 1, \text{Re } \lambda < 0\}$ は p が大きくなるほど大きい。これは $\pi_p(\lambda)$ が, e^λ のTaylor展開を p 次で打ち切ったものであり, $|e^\lambda| < 1$ for $\text{Re } \lambda < 0$ であることから, 明らかなることであるが, 安定性を良くしようとするには, order p を格段に大きくしなければならぬというdilemmaを意味する。これを打開するものとしては e^λ のPadé近似, たとえば

$$e^\lambda \sim (1 + \frac{1}{2}\lambda + \frac{1}{6}\lambda^2) / (1 - \frac{1}{2}\lambda + \frac{1}{6}\lambda^2) \quad O(\lambda^4)$$

を用いるimplicit typeが考えられる。これは既にObrenchkoff formulaとして知られたものである。

その他Taylor法が活躍できる可能性のありえる場合として

(a) 解の孤立特異点のdetection

(b) implicitに与えられた微分方程式 $f(x, y, y') = 0$ の解法

などが考えられ, 引き続き検討してみたい。

参考文献

- [1] Buchberger, B., Collins, G.E. & Loos, R.(ed.), Computer Algebra, Computing Supplement 4, Springer-V., 1982.
- [2] 数理科学, 1983年8月号, 特集数式処理, サイエンス社
- [3] Carnahan, B., Luther, H.A. & Wilkes, J.O., Applied Numerical Methods, John Wiley, 1969. (邦訳, 計算機による数値計算法, 日本コンピュータ協会, 1982)
- [4] Forsythe, G.E., Malcolm, M.A. & Moler, C.B., Computer Methods for Mathematical Computations, Prentice-Hall, 1977. (邦訳, 計算機のための数値計算法, 科学技術出版社, 1978)
- [5] 三井, 京都大学大型計算機センター-広報, 近刊.