Information Processing in Japan Vol. 9, 1969

On Runge-Kutta Type Formulae with the Error Estimating Ability

MASATSUGU TANAKA*

1. *Preface*

This paper is a continuation of our preceding thesis, "On the Kutta-Merson Process and its allied Process." ([1]) Now, we set the differential equation to be the object of numerical solution as

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

Firstly, as to the case where $f(x,y)$ is a function especially of x only, we make such Runge-Kutta formulae with the error estimating ability as need to compute functional values three to five times per step. As numerical integral formulae with error estimating ability, these methods are applicable to the routine capable of modifying pitches. In addition, in these cases we show the possibility of making formulae with better efficiency if the function satisfies a certain condition. Secondly, as to the case where $f(x,y)$ is a general function of two variables, x and y, we make just the same formulae as before. And on the study of the same part, we remark that some coefficients of Kutta-Ceschino Process with a significant figure of eight units have not sufficient accuracy. In derivating formulae, R. Merson and F. Ceschino consumed the degrees of freedom of conditional equations to simplify the process, while we use them to raise the accuracy of integral formulae and their error estimate ([2], [3]). Our methods are efficacious when $f(x,y)$, the function of right hand side of (1.1), is complicated.

2. *The case where $f(x,y)$ is a function of x only*

2.1. *Preparation*

Here, the initial value problem (1.1) is

$$\frac{dy}{dx}=f(x), \quad y(x_0)=y_0 \tag{2.1}$$

In 2, we take up the Runge-Kutta formula able to estimate truncation errors whose general expression is

$$k_i=hf(x_0+\alpha_i h) \quad i=1, 2, \cdots\cdots, m \tag{2.2}$$

$$y_{n+1}=y_n+\sum_{i=1}^{s}\nu_i k_i \tag{2.3}$$

$$y_{n+1}'=y_n'+\sum_{i=1}^{m}\mu_i k_i \quad (m\geq s,\ y_0'=y_0) \tag{2.4}$$

This paper first appeared in Japanese in Joho Shori (the Journal of the Information Processing Society of Japan), Vol. 9, No. 5 (1968), pp. 261-271.

* Faculty of Engineering, Yamanashi University, Japan.

$$T = y_{n+1} - y_{n+1}' \tag{2.5}$$

where α_i, ν_i and μ_i are constants, and also where y_{n+1} is a formula to obtain numerical solutions, y_{n+1}' is a formula of higher accuracy, and their difference T stands for the estimated value of the truncation error of y_{n+1}.

Theorem 1. Concerning to the Runge-Kutta formula (2.4) that computes function m times per step, if we name the highest order attained r, $m \leq r$.

The proof is easily made. Actually the formula of $2m$th order is attained, too.

Theorem 2. Concerning to the general formula (2.2) to (2.5) with m functional computations per step, the following can be said.

(1) We can give the error-estimating ability to the $(m-1)$th order Runge-Kutta method (2.3).

(2) We can not give it to the mth order one (2.3).

In the above cases to make the error estimation possible, we need to have the difference of at least one order between y_{n+1} and y_{n+1}'. The proof is easily made. Henceforward, p and q represent the orders each of y_{n+1} and y_{n+1}', which are given respectively by (2.3) and (2.4).

2.2. *The cases where* $\alpha_1 = 0$. (See [4] for details)

(1) The cases of $m=3$.

Formula A-1: $\alpha_2 = \frac{1}{2}$, $\alpha_3 = 1$, $\nu_1 = 0$, $\nu_2 = 1$, $\mu_1 = \frac{1}{6}$, $\mu_2 = \frac{2}{3}$, $\mu_3 = \frac{1}{6}$

Formula A-2: $\alpha_2 = \frac{4}{5}$, $\alpha_3 = \frac{1}{4}$, $\nu_1 = \frac{3}{8}$, $\nu_2 = \frac{5}{8}$, $\mu_1 = \frac{11}{264}$, $\mu_2 = \frac{125}{264}$, $\mu_3 = \frac{128}{264}$

The former is the case when $p=2$ and $q=3$ and the latter is the one when $p=2$ and $q=4$.

(2) The case of $m=4$.

Formula A-3: $\alpha_2 = \frac{1}{4}$, $\alpha_3 = \frac{3}{4}$, $\alpha_4 = 1$, $\nu_1 = \frac{1}{9}$, $\nu_2 = \frac{1}{3}$, $\nu_3 = \frac{5}{9}$

$\mu_1 = \frac{1}{18}$, $\mu_2 = \frac{4}{9}$, $\mu_3 = \frac{4}{9}$, $\mu_4 = \frac{1}{18}$

Formula A-4: $\alpha_2 = 0.6$, $\alpha_3 = 1.77$, $\alpha_4 = 4.277777778$, $\nu_1 = 0.1980539861$,
$\nu_2 = 0.7858499525$, $\nu_3 = 0.01609606129$, $\mu_1 = 0.2000350560$,
$\mu_2 = 0.7823640125$, $\mu_3 = 0.01782904441$, $\mu_4 = -0.0002281128525$

Formula A-5: $\alpha_2 = 0.5$, $\alpha_3 = 0.1$, $\alpha_4 = 0.8888888889$, $\nu_2 = 0.4642857143$,
$\nu_3 = 0.2640845070$, $\nu_4 = 0.2716297787$, $\mu_1 = -0.02083333333$,
$\mu_2 = 0.4523809524$, $\mu_3 = 0.2934272300$, $\mu_4 = 0.2750251509$

The first Formula is the case of $p=3$ and $q=4$, and the second and the third are those of $p=3$ and $q=5$. Especially in the latter two, the degrees of freedom have been consumed to give the full ability of error estimation to the formula y_{n+1} with as high accuracy as possible.

(3) The case of $m=5$.

Formula A-6: $\alpha_2=0.8365878726$, $\alpha_3=0.3$, $\alpha_4=-0.5$, $\alpha_5=-0.85$,
$\nu_1=0.03692328692$, $\nu_2=0.4027789988$, $\nu_3=0.5539860393$,
$\nu_4=0.006311674997$, $\mu_1=0.01652856065$, $\mu_2=0.4006292846$,
$\mu_3=0.5686749535$, $\mu_4=0.01793724026$, $\mu_5=-0.003770039033$

Formula A-7: $\alpha_2=0.8877551020$, $\alpha_3=-0.2$, $\alpha_4=0.1$, $\alpha_5=0.5$,
$\nu_2=0.2758872083$, $\nu_3=-0.004467077638$, $\nu_4=0.2752590674$,
$\nu_5=0.4533208020$, $\mu_1=-0.03256704981$, $\mu_2=0.2768673718$,
$\mu_3=0.001861282349$, $\mu_4=0.3058434082$, $\mu_5=0.4479949875$

In all of the above formulae, where $p=4$ and $q=6$, the degrees of freedom for the conditional equations have been consumed to raise the accuracy of truncation error of y_{n+1} as high as possible so long as the accuracy of T is not deteriorated.

2.3. *The case where* $\alpha_1 \neq 0$

y_{n+1}' given by (2.4) is the formula obtained by applying Gauss-Legendre quadrature formula that uses m function values to

$$\int_{x_n}^{x_{n+1}} f(x)dx \tag{2.6}$$

while concerning to y_{n+1} given by (2.3), we made a formula choosing the one with the best accuracy of truncation error out of the possible combinations. Then the order of y_{n+1}' is $2m$.

(1) The case of $m=3$.

Formula B-1: $\alpha_1=0.8872983346$, $\alpha_2=0.1127016654$, $\alpha_3=\frac{1}{2}$,

$$\nu_1=\nu_2=\frac{1}{2},\ \mu_1=\mu_2=\frac{5}{18},\ \mu_3=\frac{4}{9}$$

(2) The case of $m=4$.

Formula B-2: $\alpha_1=0.06943184420$, $\alpha_2=0.3300094782$, $\alpha_3=0.9305681558$,
$\alpha_4=0.6699905218$, $\nu_1=0.04519229241$, $\nu_2=0.6521451549$,
$\nu_3=0.3026625527$, $\mu_1=0.1739274226$, $\mu_2=0.3260725774$,
$\mu_3=0.1739274226$, $\mu_4=0.3260725774$

(3) The case of $m=5$.

Formula B-3: $\alpha_1=0.04691007703$, $\alpha_2=0.2307653449$, $\alpha_3=0.7692346551$,
$\alpha_4=0.9530899230$, $\alpha_5=0.5$, $\nu_1=0.04083499337$, $\nu_2=0.4591650066$,
$\nu_3=0.4591650066$, $\nu_4=0.04083499337$, $\mu_1=0.1184634425$,
$\mu_2=0.2393143352$, $\mu_3=0.2393143352$, $\mu_4=0.1184634425$,
$\mu_5=0.2844444444$

Table 1 shows the true errors and the estimated truncation errors of the numerical solutions y_{n+1} obtained when the ordinary differential equation

$$\frac{dy}{dx}=e^x,\ y(0)=1 \tag{2.7}$$

is integrated one step from $x=0.0$ with the pitch of 0.1 using each of the formulae in 2, while Table 2 shows the similar ones with those of Table 1 when the ordinary differential equation

$$\frac{dy}{dx}=\frac{1}{1+x}, \quad y(0)=0 \tag{2.8}$$

is integrated one step from $x=0.0$ with the pitch of 0.1.

Table 1. The Numerical Solution of $\frac{dy}{dx}=e^x$, $y(0)=1$.

method	numerical solution y_1	$10^9\times$ (true error)	$10^9\times$ (error estimate)
Formula A-1	1.105127105	−43809	−43812
〃 A-2	1.105205441	34523	34525
〃 A-3	1.105170734	−184	−183
〃 A-4	1.105171094	176	178
〃 A-5	1.105170901	−17	−16
〃 A-6	1.105170917	−1	−1
〃 A-7	1.105170916	−2	−1
〃 B-1	1.105205964	35048	35046
〃 B-2	1.105169833	305	303
〃 B-3	1.105170914	−3	−4

Table 2. The Numerical Solution of $\frac{dy}{dx}=\frac{1}{1+x}$, $y(0)=0$.

method	numerical solution y_1	$10^9\times$ (true error)	$10^9\times$ (error estimate)
Formula A-1	0.09523809523	−72084	−72150
〃 A-2	0.09537037036	60191	60222
〃 A-3	0.09531102286	844	859
〃 A-4	0.09530973639	−443	−505
〃 A-5	0.09531025988	80	81
〃 A-6	0.09531020165	22	22
〃 A-7	0.09531017720	−2	−2.5
〃 B-1	0.09536784745	57667	57667
〃 B-2	0.09530874982	−1430	−1430
〃 B-3	0.09531014657	−33	−33

Observing Tables 1 and 2, we see each of the formulae in the cases where $\alpha_1 \neq 0$ has accuracy of remarkably high level in error estimation.

2.4. *Remarks*

When y_{n+1} of (2.3), where $s=m-1$, and y_{n+1}' of (2.4) are each the $(m-1)$th order method and the mth order one, and the estimated values of their truncation errors are each T_{n+1} and T_{n+1}', if the pitch is small enough and $f^{(m)}(x)$ does not change suddenly,

$$T_{n+1}' \approx \frac{h_{n+1}\left\{\sum_{i=1}^{m} \mu_i \alpha_i^m - \frac{1}{(m+1)}\right\}(T_{n+1}-T_n)}{mh_n\left(\sum_{i=1}^{m-1} \nu_i \alpha_i^{m-1} - \frac{1}{m}\right)} \tag{2.9}$$

where h_n is the pitch at the nth step.

3. *The Case where $f(x, y)$ is the General Function of x, y* (See [4])

3.1. *Preparation*

The general form of the formula is

$$k_i = hf(x_n + \alpha_i h,\ y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j) \quad (i=1, 2, \cdots\cdots, m) \tag{3.1}$$

$$y_{n+1} = y_n + \sum_{i=1}^{s} \nu_i k_i \tag{3.2}$$

$$y_{n+1}' = y_n + \sum_{i=1}^{m} \mu_i k_i \quad (m \geq s) \tag{3.3}$$

$$T = y_{n+1} - y_{n+1}' \tag{3.4}$$

where α_i, β_{ij}, ν_i and μ_i are constants and especially $\alpha_1 = \beta_{10} = 0$ and y_{n+1} is a formula to obtain the numerical solution while y_{n+1}' is a formula with higher accuracy than y_{n+1} and is necessary to obtain the estimated value of the truncation error of y_{n+1}. In 3, we use the criteria defined in [1] to measure the accuracy of the truncation error of the formula.

(1) The case of $m=4$.

Theorem 3. It's impossible to give the error estimating ability to a third order Runge-Kutta method by means of four functional computations per step. The proof is easily made. Consuming two degrees of freedom conditional equations have so as to make y_{n+1} a second order method have the highest accuracy possible, we obtain the following formula for which $p=2$ and $q=4$.

Formula C-1: $\alpha_2 = -0.4$, $\alpha_3 = 0.425$, $\alpha_4 = 1$, $\beta_{21} = -0.4$,
$\beta_{31} = 0.6684895833$, $\beta_{32} = -0.2434895833$,
$\beta_{41} = -2.323685857$, $\beta_{42} = 1.125483559$,
$\beta_{43} = 2.198202298$, $\nu_2 = 0.03968253968$,
$\nu_3 = 0.7729468599$, $\nu_4 = 0.18737060041$,
$\mu_1 = 0.03431372549$, $\mu_2 = 0.02705627706$,
$\mu_3 = 0.7440130202$, $\mu_4 = 0.1946169772$

(2) The case of $m=5$.

Theorem 4. It's impossible to give the error estimating ability to a fourth order method by means of five functional computations per step.

Proof. It is because the fifth order method can't be made by five functional computations ([5]).

Formula C-2 has been made so as to make y_{n+1}' be almost a fifth order method and to make y_{n+1} have the highest accuracy attainable so long as the accuracy of T is not deteriorated.

Formula C-2: $\alpha_2=0.0005$, $\alpha_3=0.285$, $\alpha_4=0.992$, $\alpha_5=1.0$,
$\beta_{21}=0.0005$, $\beta_{31}=-80.89939470$, $\beta_{32}=81.18439470$,
$\beta_{41}=2113.327899$, $\beta_{42}=-2117.778035$, $\beta_{43}=5.442136522$,
$\beta_{51}=2249.757677$, $\beta_{52}=-2254.489040$, $\beta_{53}=5.739991965$,
$\beta_{54}=-0.008629230728$, $\nu_1=-131.2823524$, $\nu_2=131.4998223$,
$\nu_3=0.4837620276$, $\nu_4=0.2987680554$, $\mu_1=65.80784286$,
$\mu_2=-65.94767173$, $\mu_3=0.7959885276$, $\mu_4=4.715404915$,
$\mu_5=-4.371564570$

Table 3 shows the numerical solutions, true errors and estimated errors of the ordinary differential equation

$$\frac{dy}{dx}=\frac{5y}{1+x}, \quad y(0)=1 \tag{3.5}$$

which is integreted one step from $x=2.0$ with the pitch of 0.1 using each Formula C-1, C-2 and Kutta-Ceschino process.

Table 3. The Numerical Solution of $\frac{dy}{dx}=\frac{5y}{1+x}$, $y(0)=1$.

method	numerical solution y_1	$10^6\times$ (true error) TE	$10^6\times$ (error estimate) T	T/TE
Formula C-1	1.610011268	−498.7	−445.6	0.89
Formula C-2	1.610923240	415.2	422.9	1.02
Kutta-Ceschino Process	1.610932634	422.6	463.7	1.10

Both of Formula C-2 and Kutta-Ceschino Process make five computations of function, but the former needs the procedure of making a formula to obtain the numerical solution while the latter doesn't. In compensation for it, however, the former gets the higher accuracy of y_{n+1}' and accordingly of the estimated value. Table 3 proves together with the table in [4], which shows the criteria of the truncation errors of Formula C-1, C-2 and Kutta-Ceschino Process in y_{n+1} and y_{n+1}', that the above fact is correct. (See [4])

3.3. *Kutta-Ceschino Process* ([2])

Through the detailed investigations of the case where $m=5$ in Kutta-Ceschino Process, we see the method by F. Ceschino is almost the best and the improvement seems to be nealy impossible. Though his coefficients with the significant figure of eight units sometimes makes errors in the last three units and doesn't seem to have sufficient accuracy, it has little effect in general and therefore counts for nothing. (See [4] for details)

4. *Acknowledgement*

We express here our sincere gratitude to Professor Sigeiti Moriguti, University of Tokyo, for his helpful suggestions.

References

[1] Tanaka, M., Kutta-Merson Process and its allied processes, *Information Processing in Japan, 8,* (1968).

[2] Ceschino, F., Evalution de l'erreur par pas les problèmes différentiels, *Chiffres,* 5 (1962).

[3] Merson, R. H., An operational method for study of integration process, *Proceedings of Symposium on Data processing,* Weapon Research Establishment, Salisbury, South Australia (1957).

[4] Tanaka, M., On Runge-Kutta Type Formulas with Error Estimating Ability, Joho Shori (The Journal of the Information Processing Society of Japan) 9, 5 (1968), 261–271.

[5] Ceschino F. et J. Kuntzmann, Méthode Numériques Problèmes Différentiels de Conditions Initiales, Dunod, Paris (1963), 89–91.