Economical Evaluation of Runge-Kutta Formulae

By David J. Fyfe

1. Gill [1] and Blum [2] have produced special versions of the Runge-Kutta fourth order method for the solution of N simultaneous first order differential equations which require $3N + P$ storage locations against the normal $4N + P$, where P is the storage required by the program. It is shown below that it is possible to arrange all such methods in a form which requires $3N + P$ storage locations. Gill's method for reducing round off error is also extended.

2. The Runge-Kutta fourth order methods for solving the N equations

$$y_i' = f_i(x, y_1, y_2, \cdots, y_N),$$
$$y_i(x_0) = y_{i0}, \qquad i = 1, 2, \cdots, N,$$

are usually written in the form $(y_{ij} = y_i(x_j))$

$$y_{ij+1} = y_{ij} + ak_{i0} + bk_{i1} + ck_{i2} + dk_{i3},$$

where

$$\begin{aligned} k_{i0} &= hf_i(x_j, y_{1j}, y_{2j}, \cdots, y_{Nj}), \\ k_{i1} &= hf_i(x_j + mh, y_{1j} + mk_{10}, \cdots, y_{Nj} + mk_{N0}), \\ k_{i2} &= hf_i(x_j + nh, y_{1j} + (n-r)k_{10} + rk_{11}, \cdots, y_{Nj} + (n-r)k_{N0} + rk_{N1}), \\ k_{i3} &= hf_i(x_j + ph, y_{1j} + (p-s-t)k_{10} + sk_{11} + tk_{12}, \cdots, +y_{Nj} \\ &\quad + (p-s-t)k_{N0} + sk_{N1} + tk_{N2}), \end{aligned} \tag{1}$$

$$i = 1, 2, \cdots, N.$$

The $a, b, c, d, m, n, p, r, s, t$ satisfy the following eight equations

$$\begin{aligned} a + b + c + d &= 1, \\ bm + cn + dp &= \tfrac{1}{2}, \\ bm^2 + cn^2 + dp^2 &= \tfrac{1}{3}, \\ cmr + dnt + dms &= \tfrac{1}{6}, \\ bm^3 + cn^3 + dp^3 &= \tfrac{1}{4}, \\ cmnr + dntp + dmsp &= \tfrac{1}{8}, \\ cm^2r + dn^2t + dm^2s &= \tfrac{1}{12}, \\ dmrt &= \tfrac{1}{24}. \end{aligned} \tag{2}$$

The computation with the formulae arranged in this form requires $4N + P$ storage

Received October 4, 1965.

locations. It is now shown that by splitting the computation into four stages it is possible to obtain the solution by storing $3N$ quantities at each stage.

Let

$$\begin{aligned} Z_{i0} &= y_{ij}, \\ Z_{i1} &= y_{ij} + mk_{i0}, \\ Z_{i2} &= y_{ij} + (n - r)k_{i0} + rk_{i1}, \\ Z_{i3} &= y_{ij} + (p - s - t)k_{i0} + sk_{i1} + tk_{12}, \\ y_{ij+1} \equiv Z_{i4} &= y_{ij} + ak_{i0} + bk_{i1} + ck_{i2} + dk_{i3}. \end{aligned}$$

Expressing each Z_i in terms of the previous value we obtain

$$\begin{aligned} Z_{i0} &= y_{ij}, \\ Z_{i1} &= Z_{i0} + mk_{i0}, \\ Z_{i2} &= Z_{i1} + (n - m + r)k_{i0} + rk_{i1}, \\ Z_{i3} &= Z_{i2} + (p - s - t - n + r)k_{i0} + (s - r)k_{i1} + tk_{i2}, \\ Z_{i4} &= Z_{i3} + (a - p + s + t)k_{i0} + (b - s)k_{i1} + (c - t)k_{i2} + dk_{i3}. \end{aligned}$$

Let $P_{i0} = k_{i0} = hf_i(x_j, Z_{10}, \cdots, Z_{N0})$, then $Z_{i1} = Z_{i0} + mP_{i0}$.

Let $Q_{i1} = P_{i0}$, and $P_{i1} = k_{i1} = hf_i(x_j + mh, Z_{11}, \cdots, Z_{N1})$, then $Z_{i2} = Z_{i1} + (n - m - r)Q_{i1} + rP_{i1}$.

The P_{i1} and Z_{i2} are stored in the locations occupied by P_{i0} and Z_{i1} as the latter are no longer required.

Let $Q_{i2} = Q_{i1} + AP_{i1}$, and $P_{i2} = k_{i2} + BP_{i1}$. If A and B are chosen so that

$$(3) \qquad (p - s - t - n + r)A + tB = (s - r),$$

then $Z_{i3} = Z_{i2} + (p - s - t - n + r)Q_{i2} + tP_{i2}$. Again the new P_i, Q_i, Z_i replace the previous triplet.

Let $Q_{i3} = Q_{i2}$, and $P_{i3} = k_{i3} + CP_{i2}$. If A, B, C are chosen so that

$$(4) \qquad (a - p + s + t)A + dBC = (b - s)$$

and

$$(5) \qquad dC = (c - t)$$

then $Z_{i4} = Z_{i3} + (a - p + s + t)Q_{i3} + dP_{i3}$. In the above equations each triplet Z, P, Q is expressed in terms of the previous triplet only and so only $3N$ storage locations are required.

Solving (3), (4) and (5) we obtain

$$\begin{aligned} A &= \frac{(c - t)(s - r) - t(b - s)}{(c - t)(p - s - t - n + r) - t(a - p + s + t)}, \\ B &= \frac{s - r}{t} - \frac{(p - s - t - n + r)}{t} A, \\ C &= \frac{c - t}{d}, \end{aligned}$$

($d = 0$ or $t = 0$ is impossible from (2)).

Thus if the equations are arranged as follows only $3N + P$ storage locations are required:

$$
(6) \qquad
\begin{aligned}
Z_{i0} &= y_{ij}, \\
P_{i0} &= hf_i(x_j, Z_{10}, \cdots, Z_{N0}); \\
Z_{i1} &= Z_{i0} + mP_{i0}, \\
Q_{i1} &= P_{i0}, \\
P_{i1} &= hf_i(x_j + mh, Z_{11}, \cdots, Z_{N1}); \\
Z_{i2} &= Z_{i1} + (n - m - r)Q_{i1} + rP_{i1}, \\
Q_{i2} &= Q_{i1} + AP_{i1}, \\
P_{i2} &= hf_i(x_j + nh, Z_{12}, \cdots, Z_{N2}) + BP_{i1}; \\
Z_{i3} &= Z_{i2} + (p - s - t - n + r)Q_{i2} + tP_{i2}, \\
Q_{i3} &= Q_{i2}, \\
P_{i3} &= hf_i(x_j + ph, Z_{13}, \cdots, Z_{N3}) + CP_{i2}; \\
y_{ij+1} \equiv Z_{i4} &= Z_{i3} + (a - p + s + t)Q_{i3} + dP_{i3}.
\end{aligned}
$$

3. Gill's method of reducing round-off error is now applied to Equations (6). In this method artificial round-off errors are introduced to minimize the actual round-off error. This is possible because the quantities P_i and Q_i are of order h and so in general can be stored to a higher degree of accuracy than the Z_i. This is done automatically if floating point arithmetic is used. In the first equation of each triplet in Equations (6) (i.e. the calculation of Z_i) we are concerned with adding quantities of order h to quantities of order 1. We will show below that, by introducing appropriate modifications in the terms of order $h(Q_i)$, it is possible to compensate almost exactly for the errors in Z_i. If we let the round-off error in the calculation of $P_{i0}, Z_{i1}, \cdots$, be $e(P_{i0}), e(Z_{i1}), \cdots$, which we suppose are easily available when these quantities are computed, the total round-off error accumulated in one step is

$$
(7) \qquad
\begin{aligned}
E_i = {} & e(Z_{i1}) + e(Z_{i2}) + e(Z_{i3}) + e(Z_{i4}) \\
& + ae(P_{i0}) + be(P_{i1}) + ce(P_{i2}) + de(P_{i3}) \\
& + (a - m)e(Q_{i1}) + (a - n + r)e(Q_{i2}) + (a - p + s + t)e(Q_{i3}).
\end{aligned}
$$

We now introduce modifications $e'(Q_i)$ in the Q_i to compensate almost exactly for the errors in Z_i i.e. put

$$
(8) \qquad
\begin{aligned}
(a - m)e'(Q_{i1}) &= -e(Z_{i1}), \\
(a - n + r)e'(Q_{i2}) &= -e(Z_{i2}), \\
(a - p + s + t)e'(Q_{i3}) &= -e(Z_{i3}).
\end{aligned}
$$

In order to introduce these modifications we define quantities R_{i1}, R_{i2}, R_{i3} as follows:

$$
\begin{aligned}
R_{i1} &= Z_{i1} - Z_{i0} = mP_{i0} + e(R_{i1}), \\
R_{i2} &= Z_{i2} - Z_{i1} = (n - m - r)Q_{i1} + rP_{i1} + e(R_{i2}), \qquad (9)\\
R_{i3} &= Z_{i3} - Z_{i2} = (p - s - t - n + r)Q_{i2} + tP_{i2} + e(R_{i3}),
\end{aligned}
$$

where $e(R_{ij}) = e(Z_{ij}), j = 1, 2, 3$.

We thus introduce $e'(Q_{i1}), e'(Q_{i2}), e'(Q_{i3})$ such that

$$
\begin{aligned}
(a - m)e'(Q_{i1}) &= -e(R_{i1}) = -(R_{i1} - mP_{i0}), \\
(a - n + r)e'(Q_{i2}) &= -e(R_{i2}) = -(R_{i2} - (n - m - r)Q_{i1} - rP_{i1}), \\
(a - p + s + t)e'(Q_{i3}) &= -e(R_{i3}) = -(R_{i3} - (p - s - t - n + r)Q_{i2} - tP_{i2}),
\end{aligned}
$$

which almost exactly compensate for the errors in Z_{i1}, Z_{i2}, Z_{i3}. Therefore, redefine Q_{i1}, Q_{i2}, Q_{i3} as follows: $Q'_{ij} = Q_{ij} + e'(Q_{ij}), j = 1, 2, 3$.

$$
\begin{aligned}
Q'_{i1} &= \left(\frac{a}{a - m}\right) P_{i0} - \frac{1}{a - m} R_{i1}, \\
Q'_{i2} &= \left(\frac{a - m}{a - n + r}\right) Q'_{i1} + \left(A + \frac{r}{a - n + r}\right) P_{i1} - \frac{1}{a - n + r} R_{i2}, \\
Q'_{i3} &= \left(\frac{a - n + r}{a - p + s + t}\right) Q'_{i2} + \frac{t}{a - p + s + t} P_{i2} - \frac{1}{a - p + s + t} R_{i3}.
\end{aligned}
$$

Making the above change requires only one additional storage location as the R_i's are used temporarily in the formation of the Z_i and Q_i'. The primes on Q_i will now be dropped for convenience. The only large error term remaining is $e(Z_{i4})$. In order to eliminate this introduce Q_{i0}, R_{i4} and Q_{i4}, where Q_{i4} at one step becomes Q_{i0} at the next.

Define:

$$
\begin{aligned}
R_{i4} &= Z_{i4} - Z_{i3} = (a - p + s + t)Q_{i3} + dP_{i3} + e(R_{i4}), \\
Q_{i4} &= e(R_{i4}) = R_{i4} - (a - p + s + t)Q_{i3} - dP_{i3}, \\
Q_{i0} &= [Q_{i4}]_{x=x_j}.
\end{aligned}
$$

Q_{i4} is the round-off error in Z_{i4}.

Since $Z_{i0} = y_{ij}$, the best available estimate would appear to be $[Z_{i4} - Q_{i4}]_{x=x_j}$ which gives

$$R_{i1} = Z_{i1} - y_{ij} = Z_{i1} - Z_{i0} + Q_{i0}.$$

Thus we redefine R_{i1} and consequently Q_{i1} as follows:

$$R_{i1} = mP_{i0} - Q_{i0} + e(R_{i1})$$

and

$$Q_{i1} = \left(\frac{a}{a - m}\right) P_{i0} - \frac{1}{a - m} Q_{i0} - \frac{1}{a - m} R_{i1}.$$

However, it will now be shown, following Gill [1], that it is slightly better to let

TABLE 1

Quantity	Coefficients of errors $e(R_{i0})$	$e(R_{i1})$	$e(R_{i2})$	$e(R_{i3})$
R_{i0}	1	0	0	0
Z_{i0}	1	0	0	0
Q_{i0}	1	0	0	0
R_{i1}	$-w$	1	0	0
Z_{i1}	$(1-w)$	1	0	0
Q_{i1}	$-\frac{1}{a-m}(1-w)$	$-\frac{1}{a-m}$	0	0
R_{i2}	$-\left(\frac{n-m-r}{a-m}\right)(1-w)$	$-\left(\frac{n-m-r}{a-m}\right)$	1	0
Z_{i2}	$\left(\frac{a-n+r}{a-m}\right)(1-w)$	$\left(\frac{a-n+r}{a-m}\right)$	1	0
Q_{i2}	$-\frac{1}{a-m}(1-w)$	$-\frac{1}{a-m}$	$-\frac{1}{a-n+r}$	0
R_{i3}	$-\left(\frac{p-s-t-n+r}{a-m}\right)(1-w)$	$-\left(\frac{p-s-t-n+r}{a-m}\right)$	$-\left(\frac{p-s-t-n+r}{a-n+r}\right)$	1
Z_{i3}	$\left(\frac{a-p+s+t}{a-m}\right)(1-w)$	$\left(\frac{a-p+s+t}{a-m}\right)$	$\left(\frac{a-p+s+t}{a-n+r}\right)$	1
Q_{i3}	$-\frac{1}{a-m}(1-w)$	$-\frac{1}{a-m}$	$-\frac{1}{a-n+r}$	$-\frac{1}{a-p+s+t}$
R_{i4}	$-\left(\frac{a-p+s+t}{a-m}\right)(1-w)$	$-\left(\frac{a-p+s+t}{a-m}\right)$	$-\left(\frac{a-p+s+t}{a-n+r}\right)$	-1
Z_{i4}	0	0	0	0
Q_{i4}	0	0	0	0

$$R_{i1} = mP_{i0} - wQ_{i0} + e(R_{i1}),$$

where

$$w = 1 + \frac{2a(a - m)}{2a(1 - a) - 1 + 2(cr + ds + dt)}.$$

Table 1 shows the errors in each quantity R_i, Z_i, Q_i within one step of errors $e(R_{i0})$, $e(R_{i1})$, $e(R_{i2})$, $e(R_{i3})$, where $R_{i0} = [R_{i4}]_{x=x_j}$. The error in y_{ij+1} due to errors in the P_i caused by errors in Z_i is given by

$$E(y_{ij+1}) = aE(P_{i0}) + bE(P_{i1}) + cE(P_{i2}) + dE(P_{i3}),$$

where $E(P_{i0})$ represents the total error in P_{i0} etc. If we assume that the partial derivatives $\partial f_i/\partial y_k$, $k = 1, 2, \cdots, N$, are constant over one step

$$E(y_{ij+1}) = h \sum_{k=1}^{N} \frac{\partial f_i}{\partial y_k} \{aE(Z_{k0}) + bE(Z_{k1}) + cE(Z_{k2}) + dE(Z_{k3})\},$$

$$= h \sum_{k=1}^{N} \frac{\partial f_i}{\partial y_k} \{Se(R_{k0}) + Te(R_{k1}) + Ue(R_{k2}) + Ve(R_{k3})\},$$

where

$$S = a + (1 - w)\left[b + c\left(\frac{a - n + r}{a - m}\right) + d\left(\frac{a - p + s + t}{a - m}\right)\right],$$

$$T = b + c\left(\frac{a - n + r}{a - m}\right) + d\left(\frac{a - p + s + t}{a - m}\right),$$

$$U = c + d\left(\frac{a - p + s + t}{a - n + r}\right),$$

$$V = d.$$

Assuming the $e(R_i)$ are randomly distributed between $-\frac{1}{2}$ unit and $+\frac{1}{2}$ unit, the standard deviation in y_{ij+1} from this source is a minimum if $S = 0$, which leads to the optimum value

$$w = 1 + \frac{a(a - m)}{b(a - m) + c(a - n + r) + d(a - p + s + t)},$$

$$= 1 + \frac{2a(a - m)}{2a(1 - a) - 1 + 2(cr + ds + dt)} \qquad \text{(from Equation (2))}.$$

The final formulae are therefore:

$$Z_{i0} = y_{ij}, \quad Q_{i0} = [Q_{i4}]_{x=x_j},$$

$$P_{i0} = hf_i(x_j, Z_{10}, \cdots, Z_{N0});$$

$$R_{i1} = mP_{i0} - wQ_{i0} + e(R_{i1}),$$

$$Z_{i1} = Z_{i0} + R_{i1},$$

$$Q_{i1} = \left(\frac{a}{a - m}\right) P_{i0} - \frac{1}{a - m} Q_{i0} - \frac{1}{a - m} R_{i1},$$

$$P_{i1} = hf_i(x_j + mh, Z_{11}, \cdots, Z_{N1});$$

$$
\begin{aligned}
R_{i2} &= (n - m - r)Q_{i1} + rP_{i1} + e(R_{i2}),\\
Z_{i2} &= Z_{i1} + R_{i2},\\
Q_{i2} &= \left(\frac{a - m}{a - n + r}\right) Q_{i1} + \left(A + \frac{r}{a - n + r}\right) P_{i1} - \left(\frac{1}{a - n + r}\right) R_{i2},\\
P_{i2} &= hf_i(x_j + nh, Z_{12}, \cdots, Z_{N2}) + BP_{i1};\\
R_{i3} &= (p - s - t - n + r)Q_{i2} + tP_{i2} + e(R_{i3}),\\
Z_{i3} &= Z_{i2} + R_{i3},\\
Q_{i3} &= \left(\frac{a - n + r}{a - p + s + t}\right) Q_{i2} + \left(\frac{t}{a - p + s + t}\right) P_{i2} - \frac{1}{a - p + s + t} R_{i3},\\
P_{i3} &= hf_i(x_j + ph, Z_{13}, \cdots, Z_{N3}) + \frac{c - t}{d} P_{i2};\\
R_{i4} &= (a - p + s + t)Q_{i3} + dP_{i3} + e(R_{i4}),\\
y_{ij+1} \equiv Z_{i4} &= Z_{i3} + R_{i4},\\
Q_{i4} &= R_{i4} - (a - p + s + t)Q_{i3} - dP_{i3}.
\end{aligned}
$$

Mathematics Department
Woolwich Polytechnic
London, England

1. S. GILL, "A process for the step-by-step integration of differential equations in an automatic digital computing machine," *Proc. Cambridge Philos. Soc.*, v. 47, 1951, pp. 95–108. MR **12**, 538.
2. E. K. BLUM, "A modification of the Runge-Kutta fourth order method," *Math. of Comp.*, v. 16, 1962, pp. 176–187. MR **26** #3190.