

# Fifth-order Runge-Kutta with higher order derivative approximations \*

David Goeken & Olin Johnson

## Abstract

Given  $y' = f(y)$ , standard Runge-Kutta methods perform multiple evaluations of  $f(y)$  in each integration sub-interval as required for a given accuracy. Evaluations of  $y'' = f_y f$  or higher derivatives are not considered due to the assumption that the calculations involved in these functions exceed those of  $f$ . However,  $y''$  can be approximated to sufficient accuracy from past and current evaluations of  $f$  to achieve a higher order of accuracy than is available through current functional evaluations alone. In July of 1998 at the ANODE (Auckland Numerical Ordinary Differential Equations) Workshop, we introduced a new class of Runge-Kutta methods based on this observation (Goeken 1999). We presented a third-order method which requires only two evaluations of  $f$  and a fourth-order method which requires three. This paper reviews these two methods and gives the general solution to the equations generated by the fifth-order methods of this new class. Interestingly, these fifth-order methods require only four functional evaluations per step whereas standard Runge-Kutta methods require six.

## 1 Third-order method

We consider initial value problems expressed in autonomous form. Starting with the non-autonomous form, we assume that  $f(x, y)$  is a continuous function with domain  $D$  in  $\mathbb{R}^{n+1}$  where  $x \in \mathbb{R}$ ,  $y \in \mathbb{R}^n$  and  $(x, y) \in D$ . We assume that

$$\|f(x, y_1) - f(x, y_2)\|_2 \leq L\|y_1 - y_2\|_2$$

for all  $(x, y_1), (x, y_2) \in D$ ; thus the problem

$$\begin{aligned} y' &= f(x, y) \\ y(x_0) &= y_0 \text{ with } (x_0, y_0) \in D \end{aligned}$$

has a unique solution.

---

\* 1991 *Mathematics Subject Classifications*: 65L06.

*Key words and phrases*: multistep Runge-Kutta, third-order method, fourth-order method, fifth-order method, higher order derivatives.

©1999 Southwest Texas State University and University of North Texas.

Published November 23, 1999.

In autonomous form,  $y$  and  $f$  have  $n + 1$  components with  $y_{n+1} = x$  and  $f_{n+1}(y) = 1$ . The initial value problem is then written

$$\begin{aligned} y' &= f(y) \\ y(x_0) &= y_0 \text{ where } (y_0)_{n+1} = x_0. \end{aligned}$$

Most efforts to increase the order of the Runge-Kutta methods have been accomplished by increasing the number of Taylor's series terms used and thus the number of functional evaluations required (Butcher 1987) (Gear 1971). The use of higher order derivative terms has been proposed for stiff problems (Rosenbrock 1963) (Enright 1974). Our method adds higher order derivative terms to the Runge-Kutta  $k_i$  terms ( $i > 1$ ) to achieve a higher order of accuracy. For example, our new third-order method, GJ3, for autonomous systems, lets

$$y_{n+1} = y_n + b_1 k_1 + b_2 k_2$$

and  $k_1 = hf(y_n)$ . However, we introduce additional terms by assigning

$$k_2 = hf(y_n + a_{21}k_1 + ha_{22}f_y(y_n)k_1)$$

Using Taylor's series expansion techniques, the above is uniquely satisfied to  $O(h^3)$  as follows

$$\begin{aligned} k_1 &= hf(y_n) \\ k_2 &= hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}hf_y(y_n)k_1) \\ y_{n+1} &= y_n + \frac{1}{4}k_1 + \frac{3}{4}k_2 \end{aligned}$$

## 2 Utilizing $f_y$

The previous section developed a two-stage, third-order method; however, it introduced a term with  $f_y$ . The result is the addition of a higher derivative term to the standard Runge-Kutta method. The following describes three methods to utilize  $f_y$ .

**Method 1:** If one knows or can generate  $f_y$ , and if the evaluation of  $f_y$  is cheaper than the evaluation of  $f$ , then savings can be realized. For example, with a linear system of equations,  $y' = Ay$ ,  $f_y$  is known and constant.

**Method 2:** Since  $y'' = f' = f_y f$  for autonomous equations, and since  $k_1 = hf$ ,  $k_2$  can be replaced with

$$\begin{aligned} k_2 &= hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}hf_y k_1) \\ &= hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}hf_y hf) \end{aligned}$$

$$\begin{aligned}
&= hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}h^2f_yf) \\
&= hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}h^2f')
\end{aligned}$$

or

$$k_2 = hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}h^2y'').$$

Again, savings can be realized if one can formulate  $y''$  (or  $f'$ ) and if it is cheaper to evaluate than  $f$ .

**Method 3:** Building onto Method 2, one can approximate  $y''$  (or  $f'$ ) by using the current and previous evaluations of  $f$ . For our third-order method, this approximation must be of  $O(h)$ . Since an  $O(h)$  approximation of  $f'$  is given by  $f' = (f_n - f_{n-1})/h$ , one can approximate  $k_2$  as follows

$$\begin{aligned}
k_2 &= hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}h^2f') \\
&= hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}h^2(f_n - f_{n-1})/h) \\
&= hf(y_n + \frac{2}{3}k_1 + \frac{2}{9}h(f_n - f_{n-1}))
\end{aligned}$$

Since  $f_n$  is calculated in the current step in the evaluation of  $k_1$ , one only has to store the previous value,  $f_{n-1}$ . In effect, the use of previous values for the approximation has created a multistep Runge-Kutta method.

### 3 Fourth-order method

Similarly, our fourth-order method, GJ4, for autonomous systems, lets

$$y_{n+1} = y_n + b_1k_1 + b_2k_2 + b_3k_3$$

and

$$\begin{aligned}
k_1 &= hf(y_n) \\
k_2 &= hf(y_n + a_{21}k_1 + ha_{22}f_y(y_n)k_1) \\
k_3 &= hf(y_n + a_{31}k_1 + a_{32}k_2 + ha_{33}f_y(y_n)k_1 + ha_{34}f_y(y_n)k_2)
\end{aligned}$$

The Taylor's series expansion of these higher order methods is tedious and error prone. We used modern symbolic math packages to expand and then to solve the resulting systems of nonlinear equations that were generated. In this work, we used the symbolic math packages Reduce (Reduce 1999), PARI/GP (PARI/GP 1999), and Octave (Eaton 1997).

PARI/GP was used to generate the Taylor's series expansion of the above, resulting in the following system of equations

$$b_1 + b_2 + b_3 = 1$$

$$\begin{aligned}
b_2 a_{21} + b_3 [a_{31} + a_{32}] &= 1/2 \\
b_2 a_{21}^2 + b_3 [a_{31} + a_{32}]^2 &= 1/3 \\
b_2 a_{21}^3 + b_3 [a_{31} + a_{32}]^3 &= 1/4 \\
b_2 a_{22} + b_3 [a_{21} a_{32} + a_{33} + a_{34}] &= 1/6 \\
b_3 [a_{21} a_{34} + a_{22} a_{32}] &= 1/24 \\
b_2 a_{21} a_{22} + b_3 [a_{21} a_{32} (\frac{1}{2} a_{21} + a_{31} + a_{32}) + (a_{31} + a_{32})(a_{33} + a_{34})] &= 1/6
\end{aligned}$$

However, in order to utilize Methods 2 and 3 of Section 2, we must restrict the solution with  $a_{34} = 0$ . The general solution to the above system of equations (with  $a_{34} = 0$ ) has been found with Reduce and example solutions are shown in Table 1.

Table 1: Example of fourth-order autonomous solutions

$b_1$	$b_2$	$b_3$	$a_{21}$	$a_{22}$	$a_{31}$	$a_{32}$	$a_{33}$
1/6	1/6	2/3	1	1/2	3/8	1/8	0
1/6	2/3	1/6	1/2	1/8	-1	2	-1/2
1/6	2/3	1/6	1/2	-1/8	3	-2	5/2
1/10	1/2	2/5	1/3	1/18	-25/24	15/8	-5/18
1/10	1/2	2/5	1/3	-1/6	35/24	-5/8	5/6

## 4 Fifth-order method

In July of 1998, the authors presented (Goeken 1999) this numerical integration technique at a meeting attended by Dr. John Butcher. Using his tree-based approach (Butcher 1987), Dr. Butcher suggested a fifth-order method. Since the meeting, his technique has been verified using Taylor's series expansion techniques to determine the general solution for our fifth-order methods.

Our fifth-order method, GJ5, for autonomous systems, lets

$$y_{n+1} = y_n + b_1 k_1 + b_2 k_2 + b_3 k_3 + b_4 k_4$$

and

$$\begin{aligned}
k_1 &= hf(y_n) \\
k_2 &= hf(y_n + a_{21} k_1 + ha_{22} f_y(y_n) k_1) \\
k_3 &= hf(y_n + a_{31} k_1 + a_{32} k_2 + ha_{33} f_y(y_n) k_1) \\
k_4 &= hf(y_n + a_{41} k_1 + a_{42} k_2 + a_{43} k_3 + ha_{44} f_y(y_n) k_1)
\end{aligned}$$

PARI/GP was used to generate the Taylor's series expansion of the above, resulting in the following system of equations

$$b_1 + b_2 + b_3 + b_4 = 1$$

$$\begin{aligned}
b_2 a_{21} + b_3 [a_{31} + a_{32}] + b_4 [a_{41} + a_{42} + a_{43}] &= 1/2 \\
b_2 a_{21}^2 + b_3 [a_{31} + a_{32}]^2 + b_4 [a_{41} + a_{42} + a_{43}]^2 &= 1/3 \\
b_2 a_{22} + b_3 [a_{21} a_{32} + a_{33}] + b_4 [a_{21} a_{42} + a_{43} (a_{31} + a_{32}) + a_{44}] &= 1/6 \\
b_2 a_{21}^3 + b_3 [a_{31} + a_{32}]^3 + b_4 [a_{41} + a_{42} + a_{43}]^3 &= 1/4 \\
b_2 a_{21} a_{22} + b_3 [\frac{1}{2} a_{21}^2 a_{32} + (a_{31} + a_{32})(a_{21} a_{32} + a_{33})] + \frac{1}{2} b_4 [a_{21}^2 a_{42} \\
+ a_{43} (a_{31} + a_{32})^2 + 2(a_{41} + a_{42} + a_{43})(a_{21} a_{42} + (a_{31} + a_{32}) a_{43} + a_{44})] &= 1/6 \\
b_3 a_{22} a_{32} + b_4 [a_{21} a_{32} a_{43} + a_{22} a_{42} + a_{33} a_{43}] &= 1/24 \\
b_2 a_{21}^4 + b_3 [a_{31} + a_{32}]^4 + b_4 [a_{41} + a_{42} + a_{43}]^4 &= 1/5 \\
3b_2 a_{21}^2 a_{22} + b_3 [a_{21}^3 a_{32} + 3(a_{31} + a_{32})^2 (a_{21} a_{32} + a_{33})] + b_4 [a_{21}^3 a_{42} \\
+ (a_{31} + a_{32})^3 a_{43} + 3(a_{41} + a_{42} + a_{43})^2 (a_{21} a_{42} + (a_{31} + a_{32}) a_{43} + a_{44})] &= 7/20 \\
b_3 a_{21}^2 a_{32} (a_{31} + a_{32}) + b_4 [(a_{41} + a_{42} + a_{43})(a_{21}^2 a_{42} + (a_{31} + a_{32})^2 a_{43})] &= 1/15 \\
\frac{1}{2} b_2 a_{22}^2 + b_3 [a_{21} a_{32} (\frac{1}{2} a_{21} a_{32} + a_{22} + a_{33}) + a_{22} a_{32} (a_{31} + a_{32}) + \frac{1}{2} a_{33}^2] \\
+ b_4 [\frac{1}{2} a_{21}^2 (a_{32} a_{43} + a_{42}^2) + (a_{31} + a_{32})(a_{21} (a_{32} a_{43} + a_{42} a_{43}) + a_{43} (a_{33} + a_{44}) \\
+ \frac{1}{2} (a_{31} + a_{32}) a_{43}^2) + a_{21} a_{42} (a_{22} + a_{44}) + (a_{21} a_{32} a_{43} + a_{22} a_{42} \\
+ a_{33} a_{43})(a_{41} + a_{42} + a_{43}) + \frac{1}{2} a_{44}^2] &= 11/120 \\
b_4 a_{22} a_{32} a_{43} &= 1/120
\end{aligned}$$

The solution presented by Dr. Butcher and verified using the above system of equations is

$$\begin{aligned}
k_1 &= hf(y_n) \\
k_2 &= hf(y_n + \frac{1}{3}k_1 + \frac{1}{18}hf_y k_1) \\
k_3 &= hf(y_n - \frac{152}{125}k_1 + \frac{252}{125}k_2 - \frac{44}{125}hf_y k_1) \\
k_4 &= hf(y_n + \frac{19}{2}k_1 - \frac{72}{7}k_2 + \frac{25}{14}k_3 + \frac{5}{2}hf_y k_1) \\
y_{n+1} &= y_n + \frac{5}{48}k_1 + \frac{27}{56}k_2 + \frac{125}{336}k_3 + \frac{1}{24}k_4
\end{aligned}$$

Additional solutions to the above nonlinear system of equations have been found using Octave. Three additional solutions are shown in Table 2.

## 5 Numerical results

To demonstrate that the new methods are of the order claimed, several equations have been solved using the new third-, fourth-, and fifth-order method

Table 2: Example of fifth-order autonomous solutions

$b_1$	1/24	5/54	1/14
$b_2$	125/336	250/567	32/81
$b_3$	27/56	32/81	250/567
$b_4$	5/48	1/14	5/54
$a_{21}$	1/5	3/10	1/4
$a_{22}$	1/50	9/200	1/32
$a_{31}$	-52/27	-9/8	-329/250
$a_{32}$	70/27	15/8	252/125
$a_{33}$	-8/27	-9/32	-259/1000
$a_{41}$	43/5	17/3	209/35
$a_{42}$	-64/7	-490/81	-32/5
$a_{43}$	54/35	112/81	10/7
$a_{44}$	13/10	23/18	11/10

Table 3: Test problems

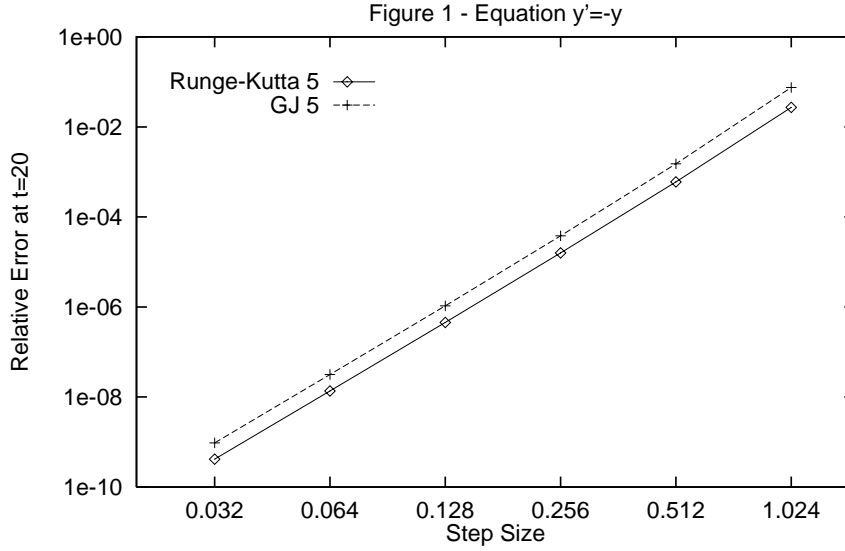
Function	Solution	$y(0)$
$y' = -y$	$y = e^{-t}$	1
$y' = \frac{y}{4} - \frac{y^2}{80}$	$y = \frac{20}{1+19e^{-\frac{t}{4}}}$	1

on scalar autonomous equations, systems of autonomous equations, and scalar non-autonomous equations.

Our previous paper (Goeken 1999) demonstrated the third- and fourth-order methods utilizing  $f_y$  and the approximation to  $f'$ . Here we will concentrate on our new fifth-order method. For scalar autonomous examples, we use the equations shown in Table 3, with initial condition  $y(0) = 1$ . These equations were solved using a standard fifth-order Runge-Kutta method along with our fifth-order method, GJ5, using  $f_y$  directly. Relative error was plotted against the step size and is shown in Figures 1 and 2. Results are comparable to standard fifth-order Runge-Kutta solution, thus demonstrating our claim. The new method requires four functional evaluations of  $f$  and one of  $f_y$  per step or four functional evaluations of  $f$  and three historical values of  $f$ ; whereas, the standard fifth-order Runge-Kutta method requires six functional evaluations of  $f$ .

## 6 Conclusions

New third-, fourth-, and fifth-order numerical integration techniques inspired by the Runge-Kutta method have been presented. The new methods exploit

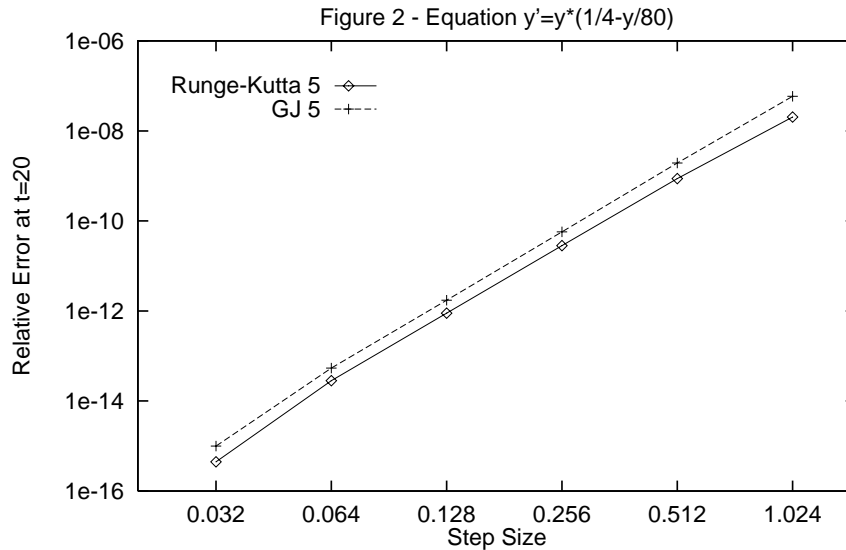


the use of higher order derivatives, specifically  $f_y$ . In particular, a technique utilizing an approximation to  $y''$  has been presented resulting in a multistep Runge-Kutta method. Table 4 compares the computational effort required for standard Runge-Kutta methods with our methods. Table 4 shows the cases where the proposed methods are more efficient than the standard Runge-Kutta methods. Specifically, the proposed methods are more efficient for cases where

- $f_y$  or  $y''$  is cheaper to evaluate than  $f$ ,
- the use of historical values of  $f$  is cheaper than evaluating  $f$ , and
- for the fifth-order case, the number of total functional evaluations can be reduced from 6 to 4 when using an approximation of  $f'$ .

Table 4: Number of evaluations comparison

Order	Standard Runge-Kutta	Proposed Method Exact $f_y$		Proposed Method Approximating $f'$	
	Num. $f$ Evals.	Num. $f$ Evals	Num. $f_y$ Evals.	Num. $f$ Evals.	Num. $f_{n-i}$ Values
3	3	2	1	2	1
4	4	3	1	3	2
5	6	4	1	4	3



## References

- [1] Butcher, J. C. (1987) *The Numerical Analysis of Ordinary Differential Equations Runge-Kutta and General Linear Methods*, John Wiley & Sons Ltd., New York
- [2] Eaton, J. W. (1997) *GNU Octave A high-level interactive language for numerical computations*,  
<http://www.we.fh-osnabrueck.de/labsim/octave/manual/octave.html>
- [3] Enright, W. H. (1974) *Second Derivative Multistep Methods for Stiff Ordinary Differential Equations*, SIAM J. Numer. Anal., **11**, 321-331
- [4] Gear, C. W. (1971) *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, New Jersey
- [5] Goeken, D. and Johnson, O. (1999) *Runge-Kutta with Higher Order Derivative Approximations*, submitted to Applied Numerical Mathematics
- [6] PARI/GP (1999)  
<http://hasse.mathematik.tu-muenchen.de/ntsw/pari/Welcome>
- [7] Reduce (1999)  
<http://www.sub.uni-goettingen.de/ssgfi/math/infodata/000838.html>
- [8] Rosenbrock, H. H. (1963) *Some General Implicit Processes for the Numerical Solution of Differential Equations*, Comp. J., **5**, 329-330



DAVID GOEKEN  
Department of Computer Science  
The University of Houston  
Houston, TX 77204-3475, USA  
e-mail: dgoeken@cs.uh.edu  
Now with the LinCom Corporation, Houston, Texas

OLIN JOHNSON  
Department of Computer Science  
The University of Houston  
Houston, TX 77204-3475, USA  
e-mail: johnson@cs.uh.edu