Department of
Mathematics and Statistics

A Comparison of some Runge-Kutta Formula Pairs using DETEST

J. H. Verner

Mathematical Preprint #1990-11

August, 1990

Queen's University
at Kingston
Ontario, Canada

A Comparison of some Runge-Kutta Formula Pairs using DETEST

J.H. Verner
Department of Mathematics and Statistics,
Queen's University, Kingston, Canada, K7M 2M8

Abstract

DETEST offers a variety of options in testing methods for treating initial value problems in ordinary differential equations. Results obtained for some recently constructed explicit Runge-Kutta formula pairs of orders 5 and 6 are reported. The results indicate some directions for searching for methods that may improve, if only slightly, on formula pairs in current use. It is particularly surprising that a pair constructed using a design proposed by Butcher in 1974 is almost as efficient as some of the most efficient pairs constructed recently.

Keywords: Explicit Runge-Kutta formula pairs, DETEST.

Subject Classifications: AMS(DOS): 65L05; CR: G.1.7.

August, 1990

1. Introduction

For almost two decades it has been known that explicit Runge-Kutta formula pairs are efficient algorithms for obtaining accurate approximations to non-stiff initial value problems for ordinary differential equations. Even so, new kinds of pairs and new pairs of known types have appeared recently in the literature. It is also possible that other new formulas will be discovered.

From among these, it is desirable to select a particular pair for implementation in a general purpose software package. To motivate this selection, two types of criteria have been used. With each new pair, test results on a set of selected problems might be compared with corresponding tests on other pairs. In this respect, the DETEST program developed by Hall et al. [5] has become a standard. Alternatively, a set of *characteristic values* may be used to assess the quality of a pair. Such a set of values has been proposed by Prince and Dormand [6].

In this paper, only pairs of orders 5 and 6 are considered. Pairs of these orders have been implemented in widely distributed software, and some new pairs of different types are known. For nine pairs, both the Prince and Dormand characteristic values of and results from two applications of DETEST using the *same options* and equalized amounts of computation are reported. Such a comparison of the most efficient pairs known of each type may be expected to identify those types with the most promise. Similar comparisons for higher order pairs are also desirable. It is hoped that the comparisons included here and those for higher order pairs might assist in the development and selection of some better pairs for use in standard software.

The most efficient known pairs of orders 5 and 6 can be divided in two classes. For a pair of the first class, eight derivative evaluations or *stages* are used to obtain approximations of orders 5 and 6. For such an (8, 5:6) pair, either approximation can be propagated. In the second class, one approximation is obtained using eight stages; the second approximation uses the derivatives of the first eight stages together with the derivative evaluation of the *first* approximation. Even though this type requires nine stages per step, one stage may be reused if the first approximation is propagated. We denote these pairs as either (9, 5(6)) or (9, (5)6) depending upon whether the order 5 or order 6 approximation is the first computed. Further, an asterisk will denote that the first approximation is propagated from step to step. For example, a (9^*,5(6)) pair will propagate a fifth-order approximation with an average of eight stages per step, whereas a (9, 5(6)) pair will propagate the sixth-order approximation using nine stages in every step.

Of the nine pairs compared, four are (8, 5:6) pairs, one is a (9, 5(6)) pair, and four are (9^*,(5)6) pairs. Hence, each except the (9, 5(6)) pair will use about eight stages per step. As the (9, 5(6)) uses nine stages per step, it might be expected to be somewhat less efficient than the other pairs; the results indicate otherwise. Coefficients of five pairs may easily be obtained in the referenced literature; those for the remaining pairs are displayed in Appendix 2 in the format of Butcher tableaus [1].

In the attempt to assess the most efficient implementations of these pairs, each is implemented using local eXtrapolation; that is, the solution is propagated using the higher-order approximation. Results from DETEST are reported for two modes. First, each pair is tested using DETEST in XEPUS mode - that selected by Hall et al. [5] for presentation of the code. More specifically, the error estimate is absolute: ie. it is the difference between the approximations of orders 6 and 5 of each component respectively. The norm of the error is estimated as the magnitude of the largest component, and the Error Per Unit Step is compared to the user selected tolerance for acceptance of a step and for computing the next trial steplength. New steplengths are restricted to a 50 percent increase.

However, this mode does not make the global error proportional to the tolerance, a feature that is desirable if a user requires a solution for which a bound on the global error is specified. Shampine [7] and Stetter [8] show that tolerance proportionality of the global error is preserved by either EPUS or XEPS: Error Per Unit Step without local extrapolation, or else Error Per Step with local eXtrapolation. Since practical experience suggests that local extrapolation enhances accuracy, the latter mode is also tested. To use DETEST with XEPS, a number of changes to the code are required. The changes required are specified in Appendix 1.

In Section 2, each pair tested is specified. Section 3 tabulates the Prince and Dormand characteristic values and the summary results from DETEST. The final section makes some observations and suggests direction for further study. Differences between the two modes of DETEST used are specified in Appendix 1. Coefficients of four of the tested pairs are tabulated in Appendix 2.

2. The methods

The first pair considered forms the basis of the DVERK code. Coefficients of this pair have been tabulated recently by Enright et al. [4]. The second pair is the algorithm PD6(5)8M for which the coefficients are to be found in Prince and Dormand [6]. The third pair is closely related to PD6(5)8M: Verner [9] showed that each pair of the type developed by Prince and Dormand could be used to construct a pair of a more restricted class. Thus, from PD6(5)8M, a slightly different pair may be obtained using transformation formulas in [9]. This corresponding pair is V6(5)8M and its coefficients are contained in Tableau 1. The fourth method, V6(5)8M2, is another of the latter type. Its coefficients are to be found in [9].

The second set of pairs studied are designated as FSAL by Prince and Dormand [6] to indicate that the First function evaluation of the new step is the Same As the Last of the previous step. The first pair tested is based on a design by Butcher [1], and constructed using techniques developed in [10]. For this pair, B5(6)9, the coefficients are displayed in Tableau 2. However, for pairs of this original design by Butcher, only the approximation of order 5 is available after eight stages in a step. For the approximation of order 6, nine stages are required. Since we wish to compare results from local extrapolation, the sixth-order approximation must be propagated, and as a result derivative evaluations are required for all nine stages in each step.

The final four methods are of a modified design. In these, the first eight stages are used to obtain the approximation of order 6, and all nine stages yield the approximation of order 5. Hence, for local extrapolation, only eight derivative evaluations are required in each step. Coefficients for the first of these pairs, D6(5)9F - proposed by Dormand et. al. [3] for use in the estimation of global errors, are contained in Tableau 3. V6(5)9a is another pair of this type designed and displayed by Verner [10]. Tableau 4 gives the coefficients of V6(5)9c which is similar to the pair V6(5)9b appearing in [10]. The final pair studied is that selected by Calvo et al. [2] from a new family they derived.

While there are differences between these pairs, each has been selected at the time of construction to be optimal in some sense. In particular, the values of A_{72} of Table 1 strongly influenced most of these selections.

3. Comparison and testing of the methods

Table 1 records some characteristic values of the formula pairs which might be used to predict their reliability and efficiency. Prince and Dormand [6] propose that they be used as criteria to select particular methods from a family. These values are determined using

$$A_7 = \| T_7 \| \qquad (1)$$

$$\hat{B}_7 = \frac{\| \hat{T}_7 \|}{\| \hat{T}_6 \|} \qquad (2)$$

$$\hat{C}_7 = \frac{\| \hat{T}_7 - T_7 \|}{\| \hat{T}_6 \|} \qquad (3)$$

$$R_z = \frac{\text{number of principal error coefficients equal to zero}}{\text{number of principal error coefficients}} \qquad (4)$$

S_{6R} = Left boundary of interval of absolute stability for the method of order 6 (5)

D = Magnitude of coefficient of largest modulus (6)

TABLE 1

Characteristic values of some methods

Pair	A_{72}	$\hat{B}_{72}$	$\hat{C}_{72}$	R_z	$A_{7\infty}$	$\hat{B}_{7\infty}$	$\hat{C}_{7\infty}$	S_{6R}	D	$\hat{A}_{6\infty}$
Conventional methods										
DVERK	2.07(-3)	3.75	1.48	0/20	1.93(-3)	4.61	1.40	-4.0	9.2	4.63(-4)
Parameters are (c_2=1/6, c_3=4/15, c_5=5/6, c_6=1/15, b_7=0)										
PD6(5)8M	2.33(-4)	2.20	1.51	0/20	8.82(-5)	1.60	1.05	-3.9	4.5	1.03(-4)
Parameters are (c_2=1/10, c_3=2/9, c_5=3/5, c_6=4/5, b_7=1/10, b_8=3/50)										
V6(5)8M	2.33(-4)	1.47	1.50	0/20	8.82(-5)	1.01	1.05	-3.9	3.9	7.28(-4)
Parameters are (c_2=1/10, c_3=2/9, c_5=3/5, c_6=4/5, b_7=0)										
V6(5)8M2	2.45(-4)	1.64	1.69	0/20	7.27(-5)	1.54	1.54	-3.9	6.6	3.68(-4)
Parameters are (c_2=1/6, c_3=(10-2$\sqrt{10}$)/15, c_5=4/5, c_6=9/10, b_7=0)										
FSAL methods										
B5(6)9	9.04(-5)	1.93	1.96	0/20	2.95(-5)	1.61	1.66	-3.4	14.5	3.90(-4)
Parameters are (c_2=1/9, c_3=1/6, c_5=1/3, c_6=1/2, c_7=3/4, b_9=1/16)										
D6(5)9F	4.37(-5)	1.79	1.77	0/20	2.92(-5)	1.88	1.88	-4.2	12.5	9.08(-5)
Parameters are (c_2=1/9, c_3=1/6, c_5=5/9, c_6=1/2, c_7=48/49, b_8=-1/2)										
V6(5)9a	4.93(-5)	1.44	1.32	0/20	1.87(-5)	1.35	1.32	-4.2	29.6	3.25(-4)
Parameters are (c_2=1/8, c_3=20-4$\sqrt{10}$)/45, c_5=9/16, c_6=1/[illegible], c_7=9/10, b_7=0)										
V6(5)9c	1.03(-4)	1.87	1.88	0/20	3.87(-5)	1.24	1.24	-4.2	3.0	3.06(-4)
Parameters are (c_2=1/9, c_3=1/6, c_5=1/2, c_6=3/5, c_7=4/5, $\hat{b}_7$=0)										
C6(5)9	6.00(-5)	2.10	2.12	0/20	3.83(-5)	1.87	1.83	-4.4	16.8	1.78(-4)
Parameters are (c_3=1/5, c_4=3/10, c_5=14/25, c_6=19/25, c_7=35226607/35688279, b_8=-79074570/210557597, $\hat{b}_9$=1/20)										

Each entry in Table 1 specifies the selection of arbitrary parameters made for the pair selected from its family. Each selection was made to optimize some corresponding characteristic values, and in most cases the emphasis was on minimizing A_{72} or $A_{7\infty}$ while ensuring that norms of B_7 and C_7 are not large. Prince and Dormand [6] justify this emphasis. Algorithms for evaluation of the coefficients of each pair from the arbitrary parameters may be found in the articles referenced in Section 2.

Table 2 reports the results of applying DETEST in the XEPUS mode. Each test used tolerances 10^{-k}, $k=3,\ldots,9$, and only summary results are reported. To contrast the results more sharply, additional tests were made with an error estimator scaled so that the total number of function evaluations was about the same for each pair tested. This was achieved by changing values of the parameter $\hat{b}_7$ for each pair, or more directly by taking a multiple of the error estimate. Values of the appropriate factors are stated in Tables 2 and 3. Then, the better pairs are indicated by smaller maximum errors and fewer deceptions.

TABLE 2

Results using DETEST with XEPUS

Pair	FCN calls	No. of steps	Maximum Error	Fraction Deceived	Fraction Bad Deceptions
Conventional methods					
DVERK	148,091	17,107	22.6	0.084	0.005
———	123,248	14,006	100.5	0.312	0.047
(Error estimate divided by 3.0)					
PD6(5)8M	124,475	14,176	4.3	0.017	0.000
V6(5)8M	172,833	20,261	0.8	0.000	0.000
———	124,486	14,194	4.0	0.019	0.000
(Error estimate divided by 7.0)					
V6(5)8M2	155,488	17,973	2.4	0.003	0.000
———	126,308	14,322	8.3	0.020	0.000
(Error estimate divided by 3.5)					
FSAL methods					
B5(6)9	183,991	18,807	0.7	0.000	0.000
———	127,957	12,589	11.8	0.013	0.000
(Error estimate divided by 8.92)					
D6(5)9F	125,895	14,078	22.9	0.053	0.005
V6(5)9a	151,231	17,188	2.6	0.007	0.000
———	123,935	13,727	9.1	0.055	0.001
(Error estimate divided by 3.5)					
V6(5)9c	151,975	17,205	0.6	0.000	0.000
———	123,911	13,700	3.3	0.003	0.000
(Error estimate divided by 3.5)					
C6(5)9	146,103	16,609	1.3	0.000	0.000
———	124,047	13,783	12.4	0.016	0.001
(Error estimate divided by 2.8)					

The same tests were repeated for each pair using the mode XEPS. Because this mode maintains the global error to be approximately proportional to the requested tolerance, it is more desirable than XEPUS for implementation. To use this mode, a number of changes to the DETEST code are suggested by Hall et al [5]. The particular changes made are identified in Appendix 1. The results of these tests appear in Table 3.

TABLE 3

Results using DETEST with XEPS

Pair	FCN calls	No. of steps	Maximum Error	Fraction Deceived	Fraction Bad Deceptions
Conventional methods					
DVERK	104,849	11,847	35.2	0.095	0.006
———	115,096	13,141	14.9	0.034	0.000
(Error estimate multiplied by 2.0)					
PD6(5)8M	90,076	9,982	4.1	0.024	0.000
———	115,878	13,286	1.0	0.000	0.000
(Error estimate multiplied by 7.0)					
V6(5)8M	116,118	13,302	1.0	0.000	0.000
V6(5)8M2	108,148	12,213	2.5	0.003	0.000
———	118,875	13,553	1.5	0.000	0.000
(Error estimate multiplied by 2.0)					
FSAL methods					
B5(6)9	124,865	12,393	1.4	0.000	0.000
———	113,904	11,176	1.5	0.000	0.000
(Error estimate divided by 2.0)					
D6(5)9F	90,071	9,762	31.4	0.078	0.007
———	115,543	12,986	4.1	0.003	0.000
(Error estimate multiplied by 7.0)					
V6(5)9a	103,575	11,511	2.8	0.022	0.000
———	113,335	12,742	1.2	0.000	0.000
(Error estimate multiplied by 2.0)					
V6(5)9c	105,743	11,617	1.0	0.000	0.000
———	115,743	12,877	0.6	0.000	0.000
(Error estimate multiplied by 2.0)					
C6(5)9	103,191	11,322	1.5	0.001	0.000
———	115,871	12,976	1.4	0.000	0.000
(Error estimate multiplied by 2.5					

4. Observations and conclusions

Nine formula pairs of orders 5 and 6 have been contrasted using both the characteristic values suggested by Prince and Dormand as measures of the quality of a pair, and results from two different options from DETEST. The latter tests have been adapted so that for each of the two options selected, equal amounts of computation were required by each pair for running the default problem set with the selected tolerances.

The results of Tables 2 and 3 clearly indicate that all of the newer pairs improve on the results obtained using the DVERK algorithm. It also appears that D6(5)9F is perhaps less robust than the remaining pairs. More detailed reports for three of these pairs given by Calvo et al [2] are consistent with the results in Table 3. Among the remaining pairs, the differences are only marginal, and further study of other properties of these pairs is warranted before selecting a particular pair for implementation.

Certainly the characteristic values of Table 1 should be considered in making such a selection. We immediately observe for DVERK that A_{72} and $A_{7\infty}$ are larger than the corresponding values for the other pairs. However, there appears to be little correlation of these values for the remaining pairs with the corresponding performance of the pairs using DETEST. In particular, the characteristic values of D6(5)9F could be considered to be as favourable as those of each of the other pairs, yet it performed relatively poorly in both modes using DETEST. Thus, it is perhaps reasonable to conclude that making A_{72} or $A_{7\infty}$ small while maintaining norms of B_7 and C_7 are near to 1 will yield good methods. However, minimizing either of A_{72} or $A_{7\infty}$ even while ensuring B_7 or C_7 are not large might not provide optimal methods. This suggests that from among the variety of known pairs that are very efficient, it remains difficult to select one particular pair over any other.

However, this study does indicate some directions for further work. For example, the DETEST results suggest that nearly optimal pairs from different classes can be constructed which exhibit almost the same levels of reliability and efficiency. Consider the pair B5(6)9.

Because nine stages are required for each step, the constraint equalizing the amount of computation reduces the number of steps taken by 12.5 percent over that for each other pair. This results in a larger average stepsize, which might be expected to cause larger errors. Further, the larger stepsizes would cause the higher order terms in the error espansion to be more significant. Hence, we would expect the error estimate to be less accurate for this pair than that for each of the other pairs. Yet, the testing shows that the maximum error and number of deceptions are not too much larger than corresponding values for other pairs tested. This good performance of B5(6)9 suggests that it might be fruitful to construct and examine other pairs which use more than the minimum number of stages per step.

It has been expected that the best choices from among those of the newer FSAL type would be more efficient that the best of other types. The tests above support this possibility, but the evidence is only marginal at this stage - better pairs may remain to be found. Even so, for the pair V6(5)9c, the coefficients are simple rational numbers which are not large in magnitude, and its real interval of absolute stability is nearly as large as the largest found among the tested pairs. As this pair has exhibited the best results from DETEST, it is quite possible that among the best pairs that exist, we may find some for which the coefficients can be represented exactly. Such a property can be an advantage for the development of software which is to be used in several different computing environments.

Acknowledgements

This research was supported by the National Science and Engineering Research Council of Canada under Grant A8147 and a grant from the Information Technology Research Centre of Ontario. The author is grateful to P.W. Sharp for his assistance with and advice on the use of DETEST.

REFERENCES

[1] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, J. Wiley, New York, 1987.

[2] M. Calvo, J.I. Montijano and L. Rández, *A new embedded pair of Runge-Kutta formulas of orders 5 and 6*, Computers Math. Applic., 20 (1990), pp. 15-24.

[3] J.R. Dormand, M.R. Lockyer, N.E. McGorrigan and P.J. Prince, *Global error estimation with Runge-Kutta triples*, J. Comput. Appl. Math., 7 (1989), pp. 835-846.

[4] W.H. Enright, K.R. Jackson, S.P. Nørsett, and P.G. Thomsen, *Interpolants for Runge-Kutta Formulas*, ACM TOMS, 12 (1986), pp. 193-218.

[5] G. Hall, W.H. Enright, T.E. Hull, A.E. Sedgwick, *DETEST: A program for comparing numerical methods for ordinary differential equations*, Dept. of Computer Science Tech. Rep. No. 60, University of Toronto, Canada, (1973).

[6] P.J. Prince and J.R. Dormand, *High order embedded Runge-Kutta formulae*, J. Comput. Appl. Math., 7 (1981), pp. 67-76.

[7] L.F. Shampine, *Local error control in codes for ordinary differential equations*, Appl. Math. and Comput., 3 (1977), pp. 189-210.

[8] H.J. Stetter, *Tolerance proportionality in ODE-codes*, Proc. Second Conf. on Numerical Treatment of Ordinary Differential Equations, (Ed. R. März), Seminarberichte No. 32, Humboldt University, Berlin, (1980), pp. 109-123.

[9] J.H. Verner, *A contrast of some Runge-Kutta formula pairs*, SIAM J. Numer. Anal., 27 (1990), to appear.

[10] ————, *Some Runge-Kutta formula pairs*, SIAM J. Numer. Anal. (1991), to appear.

APPENDIX 1

The form of the DETEST code included in Hall et. al. [5] implements a stepsize selection algorithm which is suitable for testing eXtrapolation in an Error Per Unit Step (XEPUS) mode. To test eXtrapolation in the Error Per Step (XEPS) mode, the following changes are required. First, there are three changes to METHOD:

```
 (i)  IF (EST .GT. 6.0D-1**6*TOL)
      HMAG = .9D0*(TOL/EST)**(1./6.)*DABS(H)      (twice in selecting H)
(ii)  EST = H*EST                                 (after computing EST)
```

The maximum increase in steplength allowed was 50%. These statements would need modification if a more liberal factor of 100% were selected.

There were also three changes to STATS:

```
 (i)  DATA PRECIS/0.D0/
(ii)  CALL TRUE(N,XOLD,YTRUE,X,1.D-2*ERRTOL/(X-XOLD),ISKIP)
(ii)  in determining ERRTRU, do not divide by HUSED:
      R = DABS(Y(I)-YTRUE(I))
```

The second change improves the accuracy of the TRUE solution when large stepsizes are used. (Other alternatives for a more reliable approximation of the TRUE solution are possible.) The third change is more critical in so far as it compares the Error Per Step with the error estimate. These changes, suggested by [5, p. 26], were sufficient to obtain the results reported for testing in the XEPS mode. Some alternatives were tried and found to yield the same results.

For each PAIR tested, a SUBROUTINE was written for use with DETEST. In each case, the code for METHOD in [5] for the Fehlberg formula pair of orders 5 and 6 was used as a model code.

Recent versions of DETEST provide estimates of the asymptotic formula for the global error and other statistics. Such tests were not used in this study.

APPENDIX 2

Butcher tableaus are used to display coefficients of some formula - pairs used in the tests described in this article.

TABLEAU 1

Coefficients of method V6(5)8M

0								
$\frac{1}{10}$	$\frac{1}{10}$							
$\frac{2}{9}$	$\frac{-2}{81}$	$\frac{20}{81}$						
$\frac{3}{7}$	$\frac{615}{1372}$	$\frac{-270}{343}$	$\frac{1053}{1372}$					
$\frac{3}{5}$	$\frac{3243}{5500}$	$\frac{-54}{55}$	$\frac{50949}{71500}$	$\frac{4998}{17875}$				
$\frac{4}{5}$	$\frac{-26492}{37125}$	$\frac{72}{55}$	$\frac{2808}{23375}$	$\frac{-24206}{37125}$	$\frac{338}{459}$			
1	$\frac{787}{540}$	-2	$\frac{-12369}{57460}$	$\frac{61054}{22815}$	$\frac{-770}{459}$	$\frac{385}{507}$		
1	$\frac{-2473}{1404}$	$\frac{30}{13}$	$\frac{1575}{884}$	$\frac{-1372}{351}$	$\frac{15400}{5967}$	0	0	
b^6	$\frac{61}{864}$	0	$\frac{98415}{321776}$	$\frac{16807}{146016}$	$\frac{1375}{7344}$	$\frac{1375}{5408}$	$\frac{15}{224}$	0
$\hat{b}^5$	$\frac{7}{216}$	0	$\frac{6561}{12376}$	$\frac{-2401}{5616}$	$\frac{1375}{1836}$	0	0	$\frac{13}{112}$

TABLEAU 2

Coefficients of method B5(6)9

0									
$\frac{1}{9}$	$\frac{1}{9}$								
$\frac{1}{6}$	$\frac{1}{24}$	$\frac{1}{8}$							
$\frac{1}{4}$	$\frac{1}{16}$	0	$\frac{3}{16}$						
$\frac{1}{3}$	$\frac{2}{27}$	0	$\frac{1}{9}$	$\frac{4}{27}$					
$\frac{1}{2}$	$\frac{13}{232}$	0	$\frac{12}{29}$	$\frac{-16}{29}$	$\frac{135}{232}$				
$\frac{3}{4}$	$\frac{4083}{14848}$	0	$\frac{-25299}{7424}$	$\frac{15567}{1856}$	$\frac{-89505}{14848}$	$\frac{195}{128}$			
$\frac{205}{239}$	$\frac{-7279672392380}{22614526690771}$	0	$\frac{6781148987955}{1040835956479}$	$\frac{-3610900789495300}{248759793598481}$	$\frac{2652649393744854}{248759793598481}$	$\frac{-1462673240280}{779811265199}$	$\frac{3365424781824}{8577923917189}$		
$\hat{b}^5$	$\frac{2809}{18450}$	0	0	$\frac{-2208}{2905}$	$\frac{1917}{1175}$	$\frac{-1454}{2565}$	$\frac{7808}{23175}$	$\frac{610145215867}{2957892874650}$	
b^6	$\frac{2683}{36900}$	0	0	$\frac{4096}{8715}$	$\frac{-15147}{75200}$	$\frac{716}{2565}$	$\frac{5248}{23175}$	$\frac{8577923917189}{94652571988800}$	$\frac{1}{16}$

Tableau 3

Coefficients of method D6(5)9F

$\frac{1}{9}$	$\frac{1}{9}$								
$\frac{1}{6}$	$\frac{1}{24}$	$\frac{1}{8}$							
$\frac{1}{4}$	$\frac{1}{16}$	0	$\frac{3}{16}$						
$\frac{5}{9}$	$\frac{280}{729}$	0	$\frac{-325}{243}$	$\frac{1100}{729}$					
$\frac{1}{2}$	$\frac{6127}{14680}$	0	$\frac{-1077}{734}$	$\frac{6494}{4037}$	$\frac{-9477}{161480}$				
$\frac{48}{49}$	$\frac{-13426273320}{14809773769}$	0	$\frac{4192558704}{2115681967}$	$\frac{14334750144}{14809773769}$	$\frac{117092732328}{14809773769}$	$\frac{-361966176}{40353607}$			
1	$\frac{-2340689}{1901060}$	0	$\frac{31647}{13579}$	$\frac{253549596}{149518369}$	$\frac{10559024082}{977620105}$	$\frac{-152952}{12173}$	$\frac{-5764801}{186010396}$		
b^6	$\frac{203}{2880}$	0	0	$\frac{30208}{70785}$	$\frac{177147}{164560}$	$\frac{-536}{705}$	$\frac{1977326743}{3619661760}$	$\frac{-259}{720}$	
$\hat{b}^5$	$\frac{36567}{458800}$	0	0	$\frac{9925984}{27063465}$	$\frac{85382667}{117968950}$	$\frac{-310378}{808635}$	$\frac{262119736669}{345979336560}$	$\frac{-1}{2}$	$\frac{-101}{2294}$

Tableau 4

Coefficients of method V6(5)9c

0									
$\frac{1}{9}$	$\frac{1}{9}$								
$\frac{1}{6}$	$\frac{1}{24}$	$\frac{1}{8}$							
$\frac{1}{4}$	$\frac{1}{16}$	0	$\frac{3}{16}$						
$\frac{1}{2}$	$\frac{1}{4}$	0	$\frac{-3}{4}$	1					
$\frac{3}{5}$	$\frac{134}{625}$	0	$\frac{-333}{625}$	$\frac{476}{625}$	$\frac{98}{625}$				
$\frac{4}{5}$	$\frac{-98}{1875}$	0	$\frac{12}{625}$	$\frac{10736}{13125}$	$\frac{-1936}{1875}$	$\frac{22}{21}$			
1	$\frac{9}{50}$	0	$\frac{21}{25}$	$\frac{-2924}{1925}$	$\frac{74}{25}$	$\frac{-15}{7}$	$\frac{15}{22}$		
b^6	$\frac{11}{144}$	0	0	$\frac{256}{693}$	0	$\frac{125}{504}$	$\frac{125}{528}$	$\frac{5}{72}$	
$\hat{b}^5$	$\frac{1}{18}$	0	0	$\frac{32}{63}$	$\frac{-2}{3}$	$\frac{125}{126}$	0	$\frac{-5}{63}$	$\frac{4}{21}$