M. E. Fowler

R. M. Warten

A Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues

Abstract: An explicit nonlinear numerical integration method is presented for the solution of certain large systems of ordinary differential equations in which there is a large spread of time constants, the smallest one being real. The integration formulas are derived and some local truncation error data given for small step sizes. Some applications are discussed where the new technique saves considerable computation time over classical methods.

1. Introduction

In recent years numerous authors (see, for example, Refs. 2, 6, 8, and 11 to 15, among others) have found that the numerical solution of certain systems of ordinary differential equations by means of classical explicit methods may require excessively small integration steps to reproduce smooth and apparently perfectly well-behaved functions.

This paper describes an explicit nonlinear numerical integration method which substantially alleviates this problem in a large class of systems occurring in practice, namely those in which there is a large spread of time constants, the smallest one being real.

Section 2 contains a brief discussion of the origin of the difficulties. The new method and a truncation-error analysis are presented in Sections 3 and 4. Section 5 is concerned with applications.

2. Origin of difficulties

Although most of the practical difficulties occur in large nonlinear systems of equations, for purposes of illustration it suffices to consider linear systems. Most of the material in this section is available in scattered form in the literature (e.g., Refs. 6, 8, 10, 12, 13). We shall be brief in what follows.

In particular let A be an $n \times n$ matrix with real, distinct negative eigenvalues $\lambda_1 < \lambda_2 \cdots \lambda_n < 0$, and let I be the identity matrix.

Consider the initial value problem

$$\frac{dY}{dt} = AY, \qquad Y(0) = Y_0, \tag{1}$$

where $Y = (y_1, y_2, \cdots, y_n)^T$. Since all the eigenvalues of A are negative, the true solution $Y_T(t) = e^{At}Y_T(0)$ of (1) goes to 0 with increasing t. Let Y_c denote the computed solution and let h be the stepsize (which is assumed constant for simplicity). Then the very least that one would want from Y_c is that

$$\lim_{k\to\infty} Y_c(kh) = 0 .$$

In the next paragraph necessary and sufficient conditions are developed for this to hold.

It is easy to show that if explicit Euler, Heun, or the usual 4th order Runge-Kutta method (Ref. 9, p. 237) is used on Eq. (1), there results

$$Y_c(kh) = [M(hA)]^k Y(0) \tag{2}$$

where

$$M(x) = \begin{cases} 1 + x & \text{for Euler's method} \\ 1 + x + \dfrac{x^2}{2} & \text{for Heun's method} \\ 1 + x + \dfrac{x^2}{2} + \dfrac{x^3}{6} + \dfrac{x^4}{24} & \text{for the Runge-Kutta method.} \end{cases}$$

Since all the eigenvalues of A are distinct, there is a matrix P such that $P^{-1}AP = \lambda = \text{diag}\,(\lambda_1, \lambda_2, \cdots, \lambda_n)$, so that the difference equation (2) becomes

$$Y_c(kh) = P[M(h\lambda)]^k P^{-1} Y(0) . \tag{3}$$

Since $M(h\lambda)$ is diagonal, $[M(h\lambda)]^k$ and $Y_c(kh)$ go to 0 with increasing k if and only if

$$|M(h\lambda_i)| < 1 \quad \text{for} \quad i = 1, 2, \cdots, n . \tag{4}$$

To see what this "stability condition" means in practice consider two specially constructed initial value problems.

Let $D = \text{diag}(-1, -1)$, $\quad c = (2, 2)^T$,

$$z = (z_1, z_2)^T, \qquad y = (y_1, y_2)^T$$

and

$$A = \begin{pmatrix} -500.5 & 499.5 \\ 499.5 & -500.5 \end{pmatrix}$$

consider

$$\frac{dz}{dt} = Dz + c \qquad z(0) = 0 \tag{5a}$$

$$\frac{dy}{dt} = Ay + c \qquad y(0) = (-0.1, 1)^T . \tag{5b}$$

The solutions are given by $z_1(t) = z_2(t) = 2(1 - e^{-t})$ $y_1 = z_1 + u$, $y_2 = z_1 - u$ where $u(t) = -0.1\, e^{-1000t}$. Note that for t greater than 0.02, $u(t)$ in absolute value is less than 0.25×10^{-9}, and thus for all practical purposes y and z are identical beyond $t = 0.02$.

Now, since the eigenvalues of D are -1, -1, respectively, the "stability condition" (inequality (4)) requires that for the first case (5a) the stepsize h be less than 2 when Euler's and Heun's methods are used and less than 2.78 for the Runge-Kutta method.

However, since the eigenvalues of A are -1, -1000, respectively, the "stability condition" requires that in the second case (5b), the stepsize h be 1000 times smaller than in the first case.

To summarize what has been shown so far, let $\lambda_M = \max |\lambda_i|$, $\lambda_m = \min |\lambda_i|$ $i = 1, 2, \cdots, n$. If λ_M/λ_m is large (of the order of 1000 or more) excessively small time steps are required to solve $dY/dt = AY$ by explicit Euler, Heun, and Runge-Kutta methods.

The corresponding discussion for other explicit methods such as Adams-Moulton, Milne and Hamming, and extension to complex eigenvalues can be found in articles by Chase, Crane and Klopfenstein, etc.[1,3,5,7,10].

It is outside the scope of this paper to enter into a discussion of the relative merits of explicit and implicit methods.

3. Method

To circumvent the difficulties described in Section 2, the following second-order exact method was developed. Although the new technique is a multistep method, it is *not* a linear method. This section describes the step-by-step procedure. The next section is concerned with a derivation of the formulas given here.

Given the differential equation $\dot{y} = f(t,y)$ $y(t_0) = y_0$, let y denote the approximations obtained at the previous steps, y_c the computed solution, h the stepsize from $t_n = t$ to $t_{n+1} = t + h$, and h_0 the stepsize from $t_{n-1} = t - h_0$ to $t_n = t$.

Computation proceeds by the following explicit step-by-step procedure

1) $\dot{y}_A(t) = [y(t) - y(t - h_0)]/h_0$

2) $d_1 = \dot{y}(t) - \dot{y}_A(t)$

3) $y_p(t + \delta) = y(t) + \delta\dot{y}(t)$ where $\delta \leq h/4$

4) $\dot{y}_p(t + \delta) = f[t + \delta, y_p(t + \delta)]$

5) $d_2 = [\dot{y}_p(t + \delta) - \dot{y}(t)]/\delta$

$$\lambda = \begin{cases} d_2/d_1, d_1 \neq 0 \\ 0 \quad d_1 = 0 \end{cases} \qquad c_1 = \begin{cases} (e^{\lambda h} - 1)/\lambda h & \lambda < 0, \\ 1 + \lambda h/2 & \lambda \geq 0 \end{cases}$$

6)

$$c_0 = \begin{cases} e^{\lambda h} & \lambda < 0 \\ 1 + \lambda h & \lambda \geq 0 \end{cases}$$

7) $y_c(t + h) = y(t) + h\dot{y}_A(t) + hc_1d_1$

8) $\dot{y}_c(t + h) = f[t + h, y_c(t + h)]$

9) $E = h[\dot{y}_c(t + h) - (\dot{y}_A(t) + c_0d_1)]$.

The above formulas easily generalize to systems of differential equations, in which case y, d_1, d_2, λ, c_1, c_0, etc., become vectors. Thus, for example,

$$y(t) = (y_1(t), \cdots, y_n(t))$$
$$E = (e_1, \cdots, e_n) .$$

The following comments explain some of the preceding steps.

Step 1: To make the method self-starting, h_0 and $\dot{y}_A$ are initially set to 0 as in modification A, or h_0 set to 0 and $\dot{y}_A$ set to $\dot{y}$ as in modification C and modification D.

Steps 3 & 4: These steps constitute Euler integration with step size δ which is held to less than $h/4$. (See discussion of step-size control later on in this section.)

Step 5: d_2 represents an approximation to $\ddot{y}(t)$. Throughout this paper it is assumed that f is of such a form as to make the calculation of $\ddot{y}$ inconvenient.

Step 6: The calculations of λ, c_1 and c_0 as shown here will be referred to as *modification C*. Two other methods of calculation will be referred to as modification B and D, respectively. Modification B is discussed at the end of Section 4. In modification D, positive values of λ are changed to negative values as follows:

If $\lambda = d_2/d_1$ is positive, replace λ by $-\lambda$ and d_1 by $-d_1$. Then use the new values of d_1 and λ to compute respectively

$$\dot{y}_A(t) = \dot{y} - d_1,\ c_1 = (e^{\lambda h} - 1)/\lambda h$$
and $c_0 = e^{\lambda h}$.

Step 8: $\dot{y}_c(t + h)$ represents the derivative at the end of the current step. If the computed solution $y_c(t + h)$ is accepted (see step 9) then $\dot{y}_c(t + h)$ becomes the derivative at the start of the new integration step.

Step 9: The stepsize h can be controlled by monitoring the quantity E. Various strategies may be used, a simple and effective one being as follows:

$$\text{Let}\quad U_k = u_1 + u_2|y_{c_k}(t + h)|$$
$$L_k = l_1 + l_2|y_{c_k}(t + h)|\ ,$$

where u_1, u_2, l_1 and l_2 are constants given below.

If $|e_k| > 1.5\ U_k$ for some k, the integration step h is halved, the independent variable is restored from $t + h$ to t and the values of y and $\dot{y}$ are restored to their values at the beginning of the step.

If $|e_k| > 0.75\ U_k$ for some k and $|e_k| \leq 1.5\ U_k$ for all k, the results of the current integration step are accepted, but the stepsize is halved for succeeding steps.

If $L_k \leq |e_k| \leq 0.75\ U_k$ for all k the stepsize is unaltered.

If $|e_k| \leq 0.75\ U_k$ for all k and $|e_k| < L_k$ for at least one k, a doubling indicator is activated. Actual doubling is delayed for seven integration steps. Halving always takes precedence over doubling. Thus anytime a halving signal is received, the stepsize is halved and doubling delayed for at least seven steps. Similarly, after successful doubling another seven steps must elapse before the stepsize may be doubled again.

For absolute error control set

$$u_1 = 0.0075 \qquad u_2 = 0$$
$$l_1 = 0.00005 \qquad l_2 = 0$$

For relative error control set

$$u_1 = 0.0005 \qquad u_2 = 0.0075$$
$$l_1 = 0.00001 \qquad l_2 = 0.00005.$$

In addition to the stepsize h, the Euler step δ (Step 2) also needs to be controlled. The same general strategy as for h is followed except that

a) $E_e = \dfrac{\delta}{2}\,[\dot{y}_p(t + \delta) - \dot{y}(t)]$

b) The values of u_1, u_2, l_1, l_2, are half those used for control of h.

c) The doubling indicator is activated only if each $|e_k|$ is less than the corresponding $|L_k|$.

d) If a), b), and c) permit $\delta > h/4$, then δ is forced to assume the value $\delta = h/4$.

4. Derivation of the integration formulas

The derivation to be given in this section will show that for "small" step sizes h, the new method is locally second-order exact. Thus, for sufficiently small step sizes the new method has an accuracy comparable in practice to Heun's method.

Although classical truncation error analysis is useful in the derivation of integration formulas, in discussion of error for small step sizes, and in giving error estimates of the form $O(h^n)$, such analysis does not help in the prediction of the stability properties of the algorithm. To determine stability, methods such as those outlined for example in section 2 must be applied. In general, for linear explicit single-step or multistep methods, such an analysis serves to establish stability regions for a particular method.

The method presented in this paper, however, is a *nonlinear* explicit multistep method and the nature of the stability regions remains at present an open question. Another open question concerns the degradation of accuracy as stepsizes are increased. Much more research needs to be done in these areas.

With these limitations in mind on the following discussion, consider the differential equation $\dot{y} = f(t, y)$ with solution y_T. Since this section concerns the nature of the local error only, y_T is assumed to be known. At each step, let y_c be the computed solution. y_c is assumed to be the sum of two functions, y_A, an "asymptotic part," and y_{PE}, a "perturbation" from the asymptote. y_A is to be determined from present and past values of y_T, whereas y_{PE} is to be determined from the local behavior of y_T. In practice, of course, y_T is not available and the approximations y_c obtained at the previous steps are used in the calculations instead of y_T.

Assume y_A and y_{PE} to be of the form

$$y_A(t + \xi) = y_A(t) + \xi\dot{y}_A(t)$$
$$y_{PE}(t + \xi) = y_{PE}(t) + \frac{e^{\lambda_p \xi} - 1}{\lambda_p}\,\dot{y}_{PE}(t)$$
$$0 \leq \xi \leq h\ , \qquad \text{thus}$$
$$y_c(t + \xi) = y_A(t + \xi) + y_{PE}(t + \xi)$$
$$= y_A(t) + y_{PE}(t) + \xi\dot{y}_A(t) + \frac{e^{\lambda_p \xi} - 1}{\lambda_p}\,\dot{y}_{PE}(t)\ . \qquad (6)$$

Five conditions may now be imposed to determine the five constants $y_A(t)$, $y_{PE}(t)$, $\dot{y}_A(t)$, $\dot{y}_{PE}(t)$, λ_p.

The first three are imposed to make the method *second-order exact*. To this end expand $(e^x - 1)/x$ in a Taylor series and rewrite (6) as follows:

$$y_c(t+\xi) = [y_A(t) + y_{PE}(t)] + \xi[\dot{y}_A(t) + \dot{y}_{PE}(t)] + \frac{\xi^2}{2}\lambda_p\dot{y}_{PE}(t) + \frac{\xi^3}{6}\lambda_p^2\dot{y}_{PE}(t) + O(\xi^4) .$$

Then the first three conditions are:

1) $y_A(t) + y_{PE}(t) = y_T(t)$

2) $\dot{y}_A(t) + \dot{y}_{PE}(t) = \dot{y}_T(t)$

3) $\lambda_p\dot{y}_{PE}(t) = \ddot{y}_T(t)$.

The next two conditions serve to determine the constants uniquely.

4) $y_A(t) = y_T(t)$

5) $\dot{y}_A(t) = [y_T(t) - y_T(t - h_0)]/h_0$,

where $t - h_0$ is the last point computed.

In case $y_T(t)$ is not available explicitly, the following computations yield $\ddot{y}_T(t)$ to order h.

Specifically, let $0 < \delta \leq h$. Then set

$$y_p(t+\delta) = y_T(t) + \delta\dot{y}_T(t)$$
$$\dot{y}_p(t+\delta) = f(t+\delta, y_p(t+\delta))$$
$$\ddot{y}_p(t) = \frac{\dot{y}_p(t+\delta) - \dot{y}(t)}{\delta} .$$

Condition 3) is then replaced by the condition

3′) $\lambda_p\dot{y}_{PE}(t) = \ddot{y}_p(t)$.

Note that

$$\ddot{y}_p(t) = \ddot{y}_T(t) + \frac{\delta}{2}\left[\dddot{y}_T(t) - \ddot{y}_T(t)\frac{\partial f}{\partial y}\right] + O(\delta^2) .$$

Then the truncation error $y_T(t+\xi) - y_c(t+\xi)$ equals

$$\frac{\xi^3}{6}[\dddot{y}_T(t) - \lambda_p^2\dot{y}_{PE}] - \frac{\xi^2}{4}\delta\left[\dddot{y}_T(t) - \ddot{y}_T(t)\frac{\partial f}{\partial y}\right] + O(\xi^4) .$$

Hence the method is locally second-order exact.

From the preceding analysis it follows that the method remains locally second-order exact if $(e^{\lambda_p\xi} - 1)/\lambda_p\xi$ is replaced by $1 + (\lambda_p\xi/2)$. In practice, this is done (modification *C*) whenever $\lambda_p \geq 0$.

We turn next to stepsize control. Ideally, step control should be based on the expression

$$E_0 = \int_0^h [\dot{y}_T(t+\xi) - \dot{y}_e(t+\xi)]d\xi$$

and E_0 should be compared with $y_T(t)$. In this expression $\dot{y}_e$ is obtained from $y_c(t+\xi)$ by *differentiation*. (This differs from $\dot{y}_c(t+\xi)$, which is obtained by *substitution* in the differential equation). However, $\dot{y}_T(t+\xi)$ is not available except at $\xi = 0$. Hence we make use of the fact that

$$\dot{y}_c(t+\xi) = f\{t+\xi, y_T(t+\xi) + [y_c(t+\xi) - y_T(t+\xi)]\} = \dot{y}_T(t+\xi) + O(\xi^3)$$

to obtain

$$E_0 = \int_0^h [\dot{y}_c(t+\xi) - \dot{y}_e(t+\xi)]d\xi + \int_0^h [\dot{y}_T(t+\xi) - \dot{y}_c(t+\xi)]d\xi = \int_0^h [\dot{y}_c(t+\xi) - \dot{y}_e(t+\xi)]d\xi + O(h^4) .$$

Hence if $\ddot{y}_T(t+\xi) - \ddot{y}_e(t+\xi)$ does not change sign for $0 \leq \xi \leq h$,

$$|E_0| \leq h|\dot{y}_T(t+h) - \dot{y}_e(t+h)| \leq h|\dot{y}_c(t+h) - \dot{y}_e(t+h)| + |O(h^4)|$$

Hence, for h small enough,

$$E = h[\dot{y}_c(t+h) - \dot{y}_e(t+h)]$$

yields an estimate of the truncation error.

It is easily seen that the method described in the preceding pages is not exact for the equation $\dot{y} = \lambda y + b$. The following (modification B) tries to remedy this situation.

Referring to Fig. 1, we see that if $\dot{y} = ay + b$, then $\ddot{y}_p$ and $\dot{y}_{PE} = \dot{y}_T - \dot{y}_A$ *have the same sign.* Moreover, $\ddot{y}_p/\dot{y}_T$ reproduces λ exactly. Hence when $\ddot{y}_p$ and $\dot{y}_{PE}$ have the same sign, $\dot{y}_A$ is set to 0, $\dot{y}_{PE}$ is set equal to $\dot{y}_T$ and computation proceeds from there. The other cases are handled as in modification C.

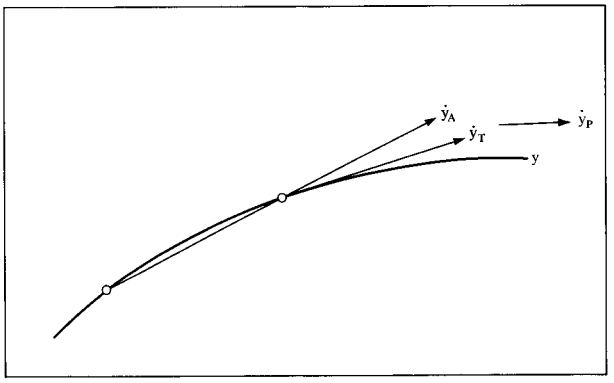

Figure 1 Basis for modification B in truncation error analysis.

5. Applications

This section is concerned with the question: Under what circumstances should the method described here be used in preference to standard classical methods?

Our experience with the method so far has led to two conclusions:

1. a) In linear systems with constant coefficients, if λ_{Max} (the eigenvalue of largest modulus) is real, then the larger the spread $|\lambda_{Max}|/|\lambda_{Min}|$, the more of an improvement can be expected.

b) If λ_{Max} is complex, the performance of the present method is approximately reduced to the performance of Heun's method. Hence, in these cases the usual 4^{th} order Runge-Kutta method is to be preferred to the present method.

2. In systems of differential equations which can be approximated by piecewise linear systems with constant coefficients, the above conclusions hold for each piece.

The rest of this section will deal with some of the experimental evidence for the preceding conclusions.

In what follows, a "pass" will mean one evaluation of the system of differential equations.

• *Example* A

The differential equations are

$$dy_1/dt = -2000\,y_1 + 1000y_2 + 1 \qquad y_1(0) = 0$$
$$dy_2/dt = y_1 - y_2 \qquad y_2(0) = 0\,,$$

where t runs from 0 to 4.

The eigenvalues are very nearly -2000.5, -0.5, so that the spread is approximately 4000. For this problem, stability requirements of the usual 4^{th}-order Runge-Kutta method limit the maximum stepsize h_{Max} to less than $2.78/2000 = 1.35 \cdot 10^{-3}$, approximately. Thus for the problem duration of 4 units of t, theory requires at least 3000 steps or 12,000 passes. In confirmation of this, a run with variable step Runge-Kutta method took 12,350 passes.

By contrast, only 480 passes were required with the new method (modification A), an improvement of 25 times.

The preceding example is linear. A two-by-two nonlinear example follows:

• *Example* B (Modified Robertson equations)

The differential equations solved are

$$dz_1/dt = 0.04(1 - z_1) - (1 - z_2)z_1 + 0.0001(1 - z_2)^2 \qquad z_1(0) = 0$$
$$dz_2/dt = -10{,}000dz_1/dt + 3000(1 - z_2)^2 \qquad z_2(0) = 1$$

These equations were derived from a system of equations investigated by H. H. Robertson (private communication from H. H. R. to H. P. Flatt).

No closed-form solution seems to be known. Table 1A compares a variable-step Runge-Kutta method run with the new method (modification A). Table 1B gives additional information on the run made with the new method.

In each of the tables, the first column labeled t lists increasing values of the independent variable. For each method, two columns labeled P and ΔP are given. The P column lists the cumulative number of passes up to and as close to the corresponding t value as possible. The ΔP column lists the number of passes between the preceding t value in the table and the present one. From Table 1A it can be seen that the number of passes per 0.2 units of t stabilizes for both the Runge-Kutta method and the new method. The new method is seen to run between 40 and 50 times faster after the number of passes has stabilized.

Table 1A Comparison of variable-step Runge-Kutta method with modification A. (Example B.)

	Runge-Kutta Method		*New Method*	
t	P	ΔP	P	ΔP
.2	777	777	122	122
.4	1505	728	135	13
.6	2221	716	154	19
.8	2953	732	167	13
1.0	3669	716	179	12
1.2	4401	742	192	13
1.4	5129	728	205	13
1.6	5845	716	217	12
1.8	6577	732	230	13
2.0	7309	732	242	12
2.2	8041	732	258	16
2.4	8773	732	267	9
2.6	9509	736	280	13

Table 1B Additional data on run in Table 1A. (Example B.)

	New Method	
t	P	ΔP
10	678	678
20	1305	627
30	1853	548
40	2421	568
50	2969	548
60	3482	513
70	3982	500
80	4482	500
90	5077	595
100	5632	555

Table 2A Comparison of Runge-Kutta method with modification D. (Example C.)

	Runge-Kutta Method		*New Method*	
t	P	ΔP	P	ΔP
5	8245	8245	1491	1491
10	16349	8104	1649	158
15	24445	8096	1782	133

Table 2B Additional data on run in Table 2A. (Example C.)

	*New Method**	
t	P	ΔP
20	1959	1959
40	2945	986
60	4284	1339
80	5241	957
100	6068	827
120	7282	1114
140	7960	678
160	8426	466
180	8924	498
200	9255	341

* Step size controlled by relative error only.

• *Example* C

This example is due to Richards, Lanning and Torrey[13]. It consists of a system of sixteen nonlinear equations for which a closed form solution is available. For further information on the system, the reader is referred to the cited reference.

In Tables 2A and 2B, the new method (modification D) is compared with the Runge-Kutta method as in the preceding example.

Table 3 Summary of results for example D.

	$\lambda_r = -0.01$				$\lambda_r = -10.$				$\lambda_r = -100$				$\lambda_r = -1000$			
	Runge-Kutta Method		*New Method*		*Runge-Kutta Method*		*New Method*		*Runge-Kutta Method*		*New Method*		*Runge-Kutta Method*		*New Method*	
t	P	ΔP	P	ΔP	P	ΔP	P	ΔP	P	ΔP	P	ΔP	P	ΔP	P	ΔP
10	329	329	307	307	357	357	292	292	1717	1717	741	741	15645	15645	1352	1352
20	461	132	527	220	509	152	524	232	3341	1624	1368	627	30985	15340	2258	906
30	565	104	680	163	669	160	776	252	4973	1632	1790	422	46481	15496	2849	591
40	657	92	767	87	821	152	928	152	6613	1640	2083	293	61973	15492	3308	459
50	705	48	804	37	981	160	1024	96	8261	1648	2270	187	77317	15344	3617	309
60	745	40	845	41	1141	160	1102	78	9885	1624	2423	153	92809	15492	3848	231
70	777	32	876	31	1297	156	1158	56	11517	1632	2517	94	108301	15492	4009	161
80	797	20	903	27	1457	160	1196	38	13157	1640	2593	76	123637	15326	4163	154
90	817	20	928	25	1613	156	1238	42	14781	1624	2666	73	139125	15488	4310	147
100	837	20	956	28	1777	164	1272	34	16413	1632	2734	68				

From the tables it is seen that if the first 5 seconds are excluded, then the new method takes average steps of approximately 0.05. For the Runge-Kutta method the average step is $10/4000 = 0.0025$, so that the improvement with respect to the number of passes is approximately a factor of 40. Since the largest "eigenvalue" of this system is 997.0 (Ref. 13, p. 379), the maximum stable step for the Runge-Kutta method is less than 0.00278, which checks with the above results. As far as accuracy is concerned, at $t = 15.0$ the Runge-Kutta method and the new method differ by less than 1.5% in every component, the Runge-Kutta method being more accurate, differing from the exact solution by less than 0.5%. The accuracy of the new method as measured by relative error rapidly deteriorates, and the computed solution is much more damped than the exact solution. Thus the values computed at $t = 200$ are attained by the exact solution at approximately 280, except for one column as also observed by Richards, Lanning and Torrey. How much of this is due to round-off error is not known.

Accuracy will be discussed in more detail in connection with example E.

• *Example* D

This example is designed to test conclusion 1, i.e., the larger the spread, the greater the improvement obtained by using the new method. The system chosen consists of three equations with one real and two complex conjugate roots. The complex roots are $-0.1 \pm 1.0\,i$. The real root λ_r takes on values of -0.01, -10, -100, -1000 in succession, depending on the values of a, b, and c. Letting $\dot{y} = dy/dt$, the differential equations chosen are:

$$\dot{y}_1 = -ay_1 - 1000y_3 + t \qquad y_1(0) = 1$$

$$\dot{y}_2 = by_3 \qquad y_2(0) = 0$$

$$\dot{y}_3 = c(y_1 - y_2) \qquad y_3(0) = .1\,,$$

Table 4 Summary of results for example E.

	Runge-Kutta Method		*New Method*	
t	P	ΔP	P	ΔP
50	41369	41369	2317	2317
100	62789	21420	3720	1403
150	83849	21112	4907	1187
200	116661	32812	6639	1732
250	161577	44916	8794	2155
300	210201	48724	11130	2336
350	262505	52304	13378	2248
400	316865	54360	15953	2575
450	372285	55420	18594	2641
500	429045	56760	21274	2680

where a, b, and c are determined by equating the coefficients in the equations

$$\lambda^3 + a\lambda^2 + (b + 1000)c\lambda + abc = 0$$

and

$$\lambda^3 + (|\lambda_r| + 0.2)\lambda^2 + (0.2|\lambda_r| + 1.01)\lambda + 1.01|\lambda_r| = 0\,.$$

The results are summarized in Table 3. The case $\lambda_r = -0.01$ confirms conclusion 1.b). Of particular interest for this case is that as $|\lambda_r|$ is increased in turn from 10 to 100 to 1000, the number of Runge-Kutta passes taken increases as predicted by theory by approximately a factor of 10 each time. On the other hand, the number of passes taken by the new method (modification B) after steady state has been reached (essentially after $t = 50$) increases by approximately a factor of 2 each time $|\lambda_r|$ is multiplied by a factor of 10. This confirms conclusion 1.a).

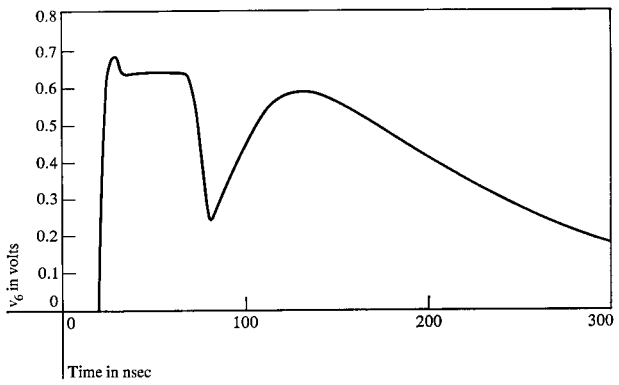

Figure 2 Nonlinear system studied by Brayton, Gustavson and Liniger (from Ref. 2, Fig. 3). See Example E in the text.

Table 5 Seven sample values of t and v_6 compared for two methods. (Example E.)

	Runge-Kutta Method		*New Method*	
	t	v_6	t	v_6
1)	31.2	.6857	31.3	.6862
2)	53.8	.6338	53.9	.6338
3)	82.1	.2516	82.5	.2514
4)	132.5	.5890	133.8	.5915
	205.	.3999	205.	.4165
	300.	.1792	300.	.1878
	500.	$.2663.10^{-2}$	500.	$.2769.10^{-2}$

• *Example* E

This is an example of a nonlinear system which was investigated by Brayton, Gustavson and Liniger[2]. A full discussion is given in the cited reference, to which the reader is referred for further information.

For our purposes it suffices to say that this system of six equations can be approximated by a piecewise linear system so that we are testing conclusion 2.

The Runge-Kutta method vs new method (modification D) comparison is contained in Table 4, and it is seen that the improvement runs to around a factor of 20.

For purposes of accuracy comparisons, Fig. 3 of the cited reference is reproduced here (see Fig. 2).

Table 5 contains seven sample values of t and v_6 for the Runge-Kutta method and the new method. Explanations of the entries are as follows:

1) Values for t and v_6 for the first peak
2) Representative values of v_6 during the flat portion
3) Values for t and v_6 for the major dip
4) Values for t and v_6 for the second maximum.

The last three entries are representative values for the remainder of the run. One other point in connection with the example may be of interest. Brayton et al, (Ref. 2, p. 296) observe that the time-constant problems experienced by the Runge-Kutta method in this case are due to three small resistors. When these are removed, the response is virtually unchanged; however, the time-constant problems disappear and the Runge-Kutta method at a fixed step of 0.25 or 0.50 gave excellent results.

One thousand integration steps or 4000 passes are required at a step of 0.50 and 8000 passes are required at a step of 0.25 to run the three-dimensional system to 500. Contrasting this with 21,274 passes for the new method on the six-dimensional system, it is seen that the new method has made up a large part of the difference.

References

1. Abbas I. Abdel Karim, "The Stability of the Fourth Order Runge-Kutta Method for the Solution of Systems of Differential Equations," *Communications of the ACM* **9,** No. 2, 113–116 (Feb. 1966).
2. R. K. Brayton, F. G. Gustavson, W. Liniger, "A Numerical Analysis of the Transient Behavior of a Transistor Circuit," *IBM Journal* **10,** No. 4, 292–299 (July 1966).
3. R. R. Brown, J. D. Riley, M. M. Bennett, "Stability Properties of Adams-Moulton Type Methods," *Math. of Computation*, **19,** No. 89, 90–96 (January 1965).
4. J. Certaine, "The Solution of ODE's with Large Time Constants," *Math. Methods for Digital Computers*, Ralston & Wilf, Wiley, 1960, pp. 128–132.
5. P. E. Chase, "Stability Properties of Predictor-Corrector Methods for Ordinary Differential Equations," *J. of the ACM*, **9,** No. 4, 457–468 (October 1962).
6. E. R. Cohen and H. P. Flatt, "Numerical Solution of Quasi-Linear Equations," published in *Codes for Reactor Comps.*, Proc. of the Seminar on Codes for Reactor Comps., Intl. Atomic Energy Agency, Vienna, 1961, p. 461.
7. R. L. Crane and R. W. Klopfenstein,"A Predictor-Corrector Algorithm with an Increased Range of Absolute Stability," *J. ACM* **12,** No. 2, 227–241 (April 1965).
8. G. Emanuel, "Numerical Analysis of Stiff Equations," Report No. TDR-269(4230-20)-3, Aerospace Corp., El Segundo, Calif.
9. F. B. Hildebrand, *Introduction to Numerical Analysis*, McGraw-Hill, New York, 1956.
10. W. Liniger, "Zur Stabilität der numerischen Integrationsmethoden für Differentialgleichungen," Thesis, University of Lausanne, Switzerland, 1957.
11. W. W. Little, Jr., K. F. Hansen, E. A. Mason, B. V. Koen, "A Stable Numerical Solution of the Reactor Kinetics Equations," *Trans. Amer. Nuclear Society* (Philadelphia meeting, June 1964), **7,** No. 1, pp. 3.4.
12. G. Moretti, "The Chemical Kinetics Problem in the Numerical Analysis of Nonequilibrium Flows," *Proc. IBM Sci. Comp. Symp., Large-Scale Problems in Physics*, Dec. 9–11, 1963, pp. 167–182.
13. P. I. Richards, W. D. Lanning, M. D. Torrey, "Numerical Integration of Large, Highly Damped, Nonlinear Systems," *SIAM Review* **7,** No. 3, 376–380 (July 1965).
14. R. W. Stineman, "Digital Time-Domain Analysis of Systems with Widely Separated Poles," *J. of the ACM*, **12,** No. 2, 286–293 (April 1965).
15. C. E. Treanor, "A Method for the Numerical Integration of Coupled First-Order Differential Equations with Greatly Different Time Constants," *Math. of Computation*, **20,** No. 93, 39–45 (January 1966).

Received March 6, 1967.