Copyright © 1981 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 8, October 1981
Printed in U.S.A.

Numerical Integration of Stochastic Differential Equations—II

By H. S. GREENSIDE and E. HELFAND

(Manuscript received March 19, 1981)

In a previous paper, a method was presented to integrate numerically nonlinear stochastic differential equations (SDEs) with additive, Gaussian, white noise. The method, a generalization of the Runge-Kutta algorithm, extrapolates from one point to the next applying functional evaluations at stochastically determined points. This paper extends (and at one point corrects) algorithms for the simple class of equations considered in the previous paper. In addition, the method is expanded to treat vector SDEs, equations with time-dependent functions, and SDEs higher than first order. The parameters for several explicit integration schemes are displayed.

I. INTRODUCTION

There are two approaches to the study of a physical system described by a stochastic differential equation (SDE). On the one hand, one may work with an equation for the probability distribution function for the random variables such as the Fokker-Planck equation. On the other hand, one may attempt to generate representative points on a trajectory by direct solution of the SDE. With either approach it is rare that analytical solutions can be found, except for linear systems. While the deterministic equation for the probability distribution can be solved numerically with standard techniques, in practice there are great difficulties. Numerical techniques for SDEs are a less-developed subject, but quite promising since they are capable of giving direct information about the random process, such as the power spectrum, higher moments, and transition rates. Several discussions of the problem have been published.[1]

A previous paper[2] (hereafter referred to as I) describes a systematic approach to the numerical solution of SDEs. Attention was limited to the simple one-variable equation of the form

$$\frac{dx}{dt} = f(x) + A(t), \tag{1}$$

where $f(x)$ is a differentiable function through some order, and $A(t)$ is a Gaussian white noise source with

$$\langle A(t) \rangle = 0, \tag{2}$$

$$\langle A(t)A(t') \rangle = \xi\delta(t - t'). \tag{3}$$

The procedure introduced was an extension of the Runge-Kutta method for numerical solution of deterministic differential equations. In the Runge-Kutta technique, as applied for instance to $dx/dt = f(x)$, $f(x)$ is evaluated at $x(t)$ and a number of other definite points. From these evaluations an extrapolation from $x(t)$ to an estimate, $\hat{x}(t + h)$, is constructed which is accurate to a given order in the time step, h, i.e. errors are less than order h^k. To apply this procedure for SDEs, the function $f(x)$ is evaluated at stochastically selected points. The algorithm is such that all moments of $\hat{x}(t + h) - x(t)$ are correct to the kth order in the step size h.

In this paper, we continue and extend the work begun in I in two ways. First, we discuss further the algorithms given in I. The two possible second-order algorithms described earlier are generalized to two families of parameter sets. A third-order algorithm proposed in I was in error and is corrected. We go on to consider a fourth order four-stage algorithm, but report our inability to find one. Our analysis suggests that kth order k-stage algorithms do not exist for $k \geq 4$.

The second way in which we extend the discussion in I is to generalize the method to three other classes of SDEs. The first class is vector SDEs in which each component has its own independent Gaussian noise source:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}) + \mathbf{A}(t). \tag{4}$$

This generalization may then be applied to the study of the class of SDEs in which $\mathbf{f}$ is an explicit function of both $\mathbf{x}$ and t. Also we show how to handle SDEs which are higher than first order (higher order derivatives of $\mathbf{x}$ with respect to t appear).*

In Section II, we briefly review the previous work and introduce the nomenclature. Section III contains a discussion of explicit algorithms which may be used for one-variable SDEs such as eq. (1). In Section IV we discuss the integration of vector SDEs and how to solve time-dependent and higher-order systems. This is illustrated in Section V

* A discussion on how our method may be applied to SDEs with multiplicative random variables will be presented elsewhere (H. S. Greenside, to be published).

with an explicit, third-order, vector algorithm. Finally, we indicate some new directions which seem important to explore.

II. REVIEW AND NOTATION

A convenient way to solve an SDE such as eq. (1) is to rewrite it as an integral equation

$$x(h) = x(0) + \int_0^h ds\, f[x(s)] + w^{[0]}(h), \tag{5}$$

where

$$w^{[0]}(t) = \int_0^t ds\, A(s) \tag{6}$$

is the Wiener process. We later need the iterates of $w^{[0]}$ defined by

$$w^{[n]}(t) = \int_0^t ds\, w^{[n-1]}(s). \tag{7}$$

The $w^{[n]}$ are Gaussian random variables with zero mean and covariances given by eqs. (18) and (19) of I. One can expand the right-hand size of eq. (5) in a series in $h^{1/2}$, where the order of the stochastic terms is determined in probability. The result is

$$x(h) = x_0 + hf + \tfrac{1}{2}h^2 ff' + (\tfrac{1}{6})h^3(ff'^2 + f^2 f'') + \cdots + S(h), \tag{8}$$

where the stochastic part is given by

$$\begin{aligned} S(h) = {} & \{w^{[0]}(h)\}_{1/2} + \{f' w^{[1]}(h)\}_{3/2} \\ & + \left\{\frac{1}{2} f'' \int_0^h ds\, [w^{[0]}(s)]^2\right\}_2 \\ & + \Big\{f'^2 w^{[2]}(h) + ff''[h w^{[1]}(h) - w^{[2]}(h)] \\ & + \left(\frac{1}{6}\right) f''' \int_0^h ds\, [w^{[0]}(s)]^3\Big\}_{5/2} \\ & + \left\{\frac{1}{2} f' f'' \left(\int_0^h ds\, (h-s)[w^{[0]}(s)]^2 + [w^{[1]}(h)]^2\right)\right. \\ & + \frac{1}{2} ff''' \int_0^h ds\, s[w^{[0]}(s)]^2 \\ & + \left.\left(\frac{1}{24}\right) f^{(iv)} \int_0^h ds\, [w^{[0]}(s)]^4\right\}_3 + \cdots . \end{aligned} \tag{9}$$

[Note that in the equation for $S(h)$ in I, eq. (I14), the third-order term involving $f'f''$ was incorrectly given.] We have written x_0 for $x(0)$, and $f^{(n)}$ indicates the nth derivative of f evaluated at $x = x_0$. In $S(h)$ terms of order h^j in probability have been gathered together in braces and the subscript j placed after the braces. The moments of the stochastic variable $S(h)$ are, to third order in h,

$$\langle S \rangle = \tfrac{1}{4}h^2\xi f'' + h^3(\tfrac{1}{4}\xi f'f'' + \tfrac{1}{6}\xi ff''' + \tfrac{1}{24}\xi^2 f^{(iv)}) + \cdots, \quad (10)$$

$$\langle S^2 \rangle = h\xi + h^2\xi f' + h^3(\tfrac{2}{3}\xi f'^2 + \tfrac{2}{3}\xi ff'' + \tfrac{1}{3}\xi^2 f''') + \cdots, \quad (11)$$

$$\langle S^3 \rangle = (7/4)h^3\xi^2 f'' + \cdots. \quad (12)$$

(Note that the coefficient of $f'f''$ in $\langle S \rangle$ is in error in I and is corrected here.) For the expansion through order h^k the terms of S nonlinear in the w's, hence non-Gaussian, do not contribute to moments higher than $2k - 1$. It follows that if errors in $\langle S^2 \rangle$ are reduced to $O(h^{k+1})$ then errors in $\langle S^n \rangle$, $n \geq 2k - 2$, will be that order or higher order in h.

In I it was proposed to integrate the SDE, eq. (1), by an extension of the Runge-Kutta scheme.[3] The algorithm for an l stage procedure is as follows:

$$g_1 = f(x_0 + h^{1/2}\xi^{1/2}Y_1), \quad (13)$$

$$g_2 = f(x_0 + h\beta_{21}g_1 + h^{1/2}\xi^{1/2}Y_2), \quad (14)$$

$$\vdots$$

$$g_l = f(x_0 + h\beta_{l1}g_1 + \cdots + h\beta_{l,l-1}g_{l-1} + h^{1/2}\xi^{1/2}Y_l), \quad (15)$$

$$x = x_0 + h(A_1g_1 + \cdots + A_lg_1) + h^{1/2}\xi^{1/2}Y_0. \quad (16)$$

The $l + 1$ stochastic variables $Y_0, \cdots, Y_l$ are Gaussianly distributed with mean zero and covariance

$$\langle Y_iY_j \rangle = L_{ij}. \quad (17)$$

The matrix $\mathbf{L}$, being symmetric, has $\frac{1}{2}(l + 1)(l + 2)$ independent parameters. Numerically, it is convenient to generate the $\mathbf{Y}$ set by writing

$$Y_j = \sum_{n=1}^{j+1} \lambda_{jp}Z_p. \quad (18)$$

where the $\mathbf{Z}$'s are a set of independent Gaussian random variables with mean zero and variance unity. Note particularly that in eq. (18) only $j + 1$ variables Z_p need be used to define Y_j. The λ_{jn} form $l + 1$ vectors of $l + 1$ components

$$\boldsymbol{\lambda}_0 = \{\lambda_{01}, 0, 0, \cdots, 0\}, \quad (19)$$

$$\boldsymbol{\lambda}_1 = \{\lambda_{11}, \lambda_{12}, 0, \cdots, 0\}, \quad (20)$$

$$\cdots$$

$$\boldsymbol{\lambda}_l = \{\lambda_{l1}, \lambda_{l2}, \lambda_{l3}, \cdots, \lambda_{l,l+1}\}. \quad (21)$$

The $\frac{1}{2}(l+1)(l+2)$ parameters λ_{jn} are related to the same number of independent parameters in the symmetric $\mathbf{L}$ matrix by

$$L_{ij} = \boldsymbol{\lambda}_i \cdot \boldsymbol{\lambda}_j \tag{22}$$

The algorithm eqs. (13) to (16) can be expressed as a power series in $h^{1/2}$, there being a deterministic part and a stochastic part, $\tilde{S}$. In turn, the moments of the stochastic part can be expanded in powers of h. (A two-stage, second-order illustration is given in I.) Each term of the deterministic part and of the moments takes the form of: (a power of h) $\times$ (a power of ξ) $\times$ (a product of powers of f and its derivatives) $\times$ (a coefficient which is a function of the parameters A_i, β_{ij}, and λ_{jn}). Corresponding terms occur in the expansion, eq. (8), and in the moments of $S(h)$ given by eq. (9), except that in the latter cases the coefficients have definite numerical values. Therefore, equations for the parameters are obtained by equating the two coefficients for each different term (i.e., different product of f and derivatives) through a given power, h^k. The series match independently of the explicit form of $f(x)$.

There are $(l+1)^2$ parameters: A_i $(i = 1, \cdots, l)$; β_{ij} $(i = 2, \cdots, l$, and $j = 1, \cdots, i-1)$; λ_{ij} $(i = 0, \cdots, l$, and $j = 1, \cdots, i+1)$. There may be fewer conditions to be satisfied than this. If so, it is convenient to use only m rather than $l+1$ Gaussian random variables, Z_p. This amounts to setting $\lambda_{jp} = 0$ for $p > m$.

A procedure which is correct through order h^k, which involves l stages, and which utilizes m Gaussians will be called a $k_O l_S m_G$ algorithm. Explicit examples are given in Section III.

III. THE $2_O2_S1_G$, $3_O3_S2_G$, AND OTHER ALGORITHMS

In I the parameters were displayed for a $2_O2_S1_G$ algorithm. There is one degree of freedom (6 parameters, 5 equations). The most general choice of parameters is

$$\begin{aligned} A_1 &= 1 - \tfrac{1}{2}\alpha^{-1}, \\ A_2 &= \tfrac{1}{2}\alpha^{-1}, \\ \beta_{21} &= \alpha, \\ \lambda_{01} &= 1, \\ \lambda_{11} &= \tfrac{1}{2}[1 \pm (2\alpha - 1)^{-1/2}], \\ \lambda_{21} &= \tfrac{1}{2}[1 \mp (2\alpha - 1)^{1/2}], \end{aligned} \tag{23}$$

with $\alpha > \frac{1}{2}$. In I the solutions with $\alpha = 1$ were suggested as particularly convenient.

In the Appendix of I, a discussion of the $3_O3_S2_G$ algorithm was

presented. The exposition contained an error and should be disregarded. A proper discussion follows.

The $16 - \frac{1}{2}(4 - m)(5 - m)$ parameters of a $3_O 3_S m_G$ algorithm must satisfy 14 equations obtained by matching the expansion of eqs. (13) to (16), and the expansion of eq. (8) and moments of eq. (9):

$$\sum_{i=1}^{3} A_i = 1, \tag{24}$$

$$\sum_{i=2}^{3} A_i \alpha_i = \tfrac{1}{2}, \tag{25}$$

$$\sum_{i=2}^{3} A_i \alpha_2^2 = \tfrac{1}{3}, \tag{26}$$

$$A_3 \beta_{32} \beta_{21} = \tfrac{1}{6}, \tag{27}$$

$$L_{00} = 1, \tag{28}$$

$$\sum_{i=1}^{3} A_i L_{0i} = \tfrac{1}{2}, \tag{29}$$

$$\sum_{i=1}^{3} A_i L_{ii} = \tfrac{1}{2}, \tag{30}$$

$$\sum_{i=1}^{3} A_i L_{0i}^2 = \tfrac{1}{3}, \tag{31}$$

$$\sum_{i=1}^{3} A_i L_{ii}^2 = \tfrac{1}{3}, \tag{32}$$

$$\sum_{i=1}^{3} A_i L_{0i} L_{ii} = \tfrac{1}{3}, \tag{33}$$

$$\sum_{i=2}^{3} A_i \alpha_i L_{ii} = \tfrac{1}{3}, \tag{34}$$

$$\sum_{i=2}^{3} A_i \alpha_i L_{0i} = \tfrac{1}{3}, \tag{35}$$

$$\sum_{i=2}^{3} A_i \sum_{j=1}^{i-1} \beta_{ij} (L_{ij} + \tfrac{1}{2} L_{jj}) = \tfrac{1}{4}, \tag{36}$$

$$\sum_{i=1}^{3} \sum_{j=1}^{3} A_i A_j L_{ij} + 2 \sum_{i=2}^{3} A_i \sum_{j=1}^{i-1} \beta_{ij} L_{0i} = \tfrac{2}{3}, \tag{37}$$

where, by definition,

$$\alpha_i = \sum_{j=1}^{i-1} \beta_{ij}, \qquad i = 2, 3. \tag{38}$$

A remarkable simplification occurs if we assume

$$A_1 = 0, \tag{39}$$

$$L_{0i} = L_{ii} = \alpha_i, \qquad i = 2, 3 \tag{40}$$

(it can be shown that no real solutions exist without these conditions). Then the 14 equations, eqs. (24) to (37), reduce to 7 independent equations in 8 unknowns for a $3_O3_S2_G$ algorithm. This leaves one degree of freedom which we can take as α_2. We are, of course, only interested in solutions for which all the parameters are real. This requires that

$$0 < \alpha_2 < \tfrac{1}{3} \qquad \text{or} \qquad \tfrac{2}{3} \leq \alpha_2 \leq 1. \tag{41}$$

Since some of the equations are nonlinear, there are multiple solutions in certain regions. Further details are presented in the Appendix. Table I gives an indication of the behavior of the parameters as α_2 is varied. Since λ_{12} is obtained from the solution of a quadratic equation, two choices are shown. As α_2 increases through 0.247583, a new pair of real roots of the equations appears, while at 0.2689703 the other pair becomes complex. There are four roots in the range $\frac{2}{3} \leq \alpha_2 \leq 1$ (although at $\frac{2}{3}$ and 1, roots are degenerate). The parameter set corresponding to $\alpha_2 = \frac{2}{3}$ looks particularly interesting because all

Table I—Parameters for $3_O3_S2_G$ algorithms appropriate to a one-variable SDE*

α_2†	β_{31}	A_2	λ_{11}	λ_{12}		λ_{32}
0.1	−1.82639	0.34247	0.03341	−1.14271	0.22458	0.45453
0.2	−0.82716	0.48077	0.12491	−1.15128	0.31072	0.41574
0.25	−0.72222	0.57143	0.24733	−0.90403	0.26211	0.37268
			−0.22692	0.81865	1.30789	−0.37268
0.30	−0.79630	0.67568	−0.31442	0.19962	5.71406	−0.27639
$\frac{2}{3}$	−1.0	$\frac{3}{4}$	$-\frac{1}{12}$	0.61844‡	−2.50406‡	0.0
0.7	−0.65079	0.67568	−0.14525	0.67143	−1.88108	0.27639
			0.06670	0.47809	−2.57869	−0.27639
0.8	−0.17901	0.48077	−0.14275	0.63842	−1.42839	0.41574
			0.14191	0.46368	−1.78360	−0.41574
0.9	0.01003	0.34247	−0.12381	0.63799	−1.24446	0.45453
			0.07126	0.64299	−1.21137	−0.45453
1.0	$\frac{1}{9}$	$\frac{1}{4}$	$-\frac{1}{16}$	0.76579§	−1.00149§	$2^{1/2}/3$

* The parameters not listed are given by:

$\beta_{21} = \alpha_2$
$\beta_{32} = 1/(6A_3\alpha_2)$
$A_1 = 0$
$A_3 = 1 - A_2$
$\lambda_{21} = \alpha_2$
$\lambda_{22} = +(\alpha_2 - \alpha_2^2)^{1/2}$
$\lambda_{31} = \beta_{31} + \beta_{32}$

† α_2 is varied as the one degree of freedom.
‡ $\pm 39^{1/2}/4 - 2^{3/2}/3$.
§ $-2^{1/2}/12 \pm 1799^{1/2}/48$.

parameters are ≤ 1, and $\alpha_3 = \lambda_{31} = \lambda_{32} = 0$. A second interesting parameter set is the one for $\alpha_2 = 1$.

A $3_O4_S2_G$ solution will be discussed in Section IV. In this case, there are enough degrees of freedom so that the parameters can be selected to produce an algorithm which integrates the deterministic part of the equation through fourth order.

It is straightforward, but quite lengthy, to extend all of the equations in Section II to fourth order. We have done so. For a $4_O4_Sm_G$ algorithm, there are $25 - \frac{1}{2}(5 - m)(6 - m)$ parameters which must satisfy 29 equations (39 coefficients must be matched but 10 of the resulting equations are not independent). The assumption that $A_1 = 0$ and $L_{0i} = L_{ii} = \alpha_1$, $i = 2, 3, 4$, reduces the number of unknowns to $18 - \frac{1}{2}(5 - m)(6 - m)$; and, remarkably, the number of independent equations is reduced to 18. Thus, there may be solutions with 5 Gaussians (the maximum possible). Unfortunately, after a reasonably thorough search for solutions we were not able to find any real solutions.* Although we do not have a proof that no real solutions exist, it appears that there is no $4_O4_Sm_G$ algorithm.

IV. ALGORITHM FOR VECTOR SDEs

Consider next vector SDEs of the type

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}) + \mathbf{A}(t), \tag{42}$$

with

$$\langle A_\kappa(t) \rangle = 0,$$

$$\langle A_\kappa(t) A_\mu(t') \rangle = \xi_\kappa \delta_{\kappa\mu} \delta(t - t'). \tag{43}$$

If the $\mathbf{A}$ covariance matrix is not diagonal, then linear combinations of the equations can be taken to diagonalize it, or the algorithm described below can be modified.

In a fashion analogous to that used to obtain eqs. (8) and (9) one can write (using the summation convention on repeated indices)

$$x_\kappa(t) = x_{0\kappa} + h f_\kappa + \tfrac{1}{2} h^2 f_{\kappa,\mu} f_\mu + (\tfrac{1}{6}) h^3 (f_{\kappa,\mu} f_{\mu,\nu} f_\nu + f_{\kappa,\mu\nu} f_\mu f_\nu) + \cdots + S_\kappa(h), \tag{44}$$

* The algebra involved in many of these calculations is extremely lengthy. Occasionally it is susceptible to simplification by a combination of equations. For these reasons, we would be willing to provide further details of our calculations to interested parties.

$$
\begin{aligned}
S_\kappa(h) = {} & \{w_\kappa^{[0]}(h)\}_{1/2} + \{f_{\kappa,\mu} w_\mu^{[1]}(h)\}_{3/2} \\
& + \left\{ \frac{1}{2} f_{\kappa,\mu\nu} \int_0^h ds\, w_\mu^{[0]}(s) w_\nu^{[0]}(s) \right\}_2 \\
& + \left\{ f_{\kappa,\mu} f_{\mu,\nu} w_\nu^{[2]}(h) + f_{\kappa,\mu\nu} f_\mu [h w_\nu^{[1]}(h) + w_\nu^{[2]}(h)] \right. \\
& + \left. \left(\frac{1}{6}\right) f_{\kappa,\mu\nu\rho} \int_0^h ds\, w_\mu^{[0]}(s) w_\nu^{[0]}(s) w_\rho^{[0]}(s) \right\}_{5/2} \\
& + \left\{ \frac{1}{2} f_{\kappa,\mu} f_{\mu,\nu\rho} \int_0^h ds\, (h-s) w_\nu^{[0]}(s) w_\rho^{[0]}(s) \right. \\
& + f_{\kappa,\mu\nu} f_{\nu,\rho} \int_0^h ds\, w_\mu^{[0]}(s) w_\rho^{[1]}(s) \\
& + \frac{1}{2} f_{\kappa,\mu\nu\rho} f_\rho \int_0^h ds\, s w_\mu^{[0]}(s) w_\nu^{[0]}(s) \\
& + \left. \left(\frac{1}{24}\right) f_{\kappa,\mu\nu\rho\sigma} \int_0^h ds\, w_\mu^{[0]}(s) w_\nu^{[0]}(s) w_\rho^{[0]}(s) w_\sigma^{[0]}(s) \right\}_3 \\
& + \cdots, \qquad (45)
\end{aligned}
$$

where

$$
f_{\kappa,\mu\nu\cdots\rho} = \frac{\partial}{\partial x_\mu} \frac{\partial}{\partial x_\nu} \cdots \frac{\partial}{\partial x_\rho} f_\kappa |_{\mathbf{x}=\mathbf{x}(0)}, \qquad (46)
$$

$$
w_\kappa^{[0]}(t) = \int_0^t ds\, A_\kappa(s), \qquad (47)
$$

with $w_\kappa^{[n]}$ being the nth iterate of $w_\kappa^{[0]}$. There is one major difference to note between eqs. (45) and (9). In the former, a distinction must be made between the term with $f_{\kappa,\mu} f_{\mu,\nu\rho}$ and that with $f_{\kappa,\mu\nu} f_{\nu,\rho}$; in the latter, both are $f'f''$. This leads to an extra equation which the parameters will have to satisfy for sets. Such differences first enter in fourth order for sets of deterministic differential equations, while for SDEs they enter at third order.

The earlier algorithm for numerical integration is easily generalized to

$$
g_{1\kappa} = f_\kappa(\{x_{0\mu} + h^{1/2} \xi_\mu^{1/2} Y_{1\mu}\}), \qquad (48)
$$

$$
g_{2\kappa} = f_\kappa(\{x_{0\mu} + h\beta_{21} g_{1\mu} + h^{1/2} \xi_\mu Y_{2\mu}\}), \qquad (49)
$$

$$
\vdots
$$

$$g_{l\kappa} = f_\kappa(\{x_{0\mu} + h\beta_{l1}g_{1\mu} + \cdots + h\beta_{l,l-1}g_{l-1\mu} + h^{1/2}\xi_\mu^{1/2}Y_{l\mu}\}), \quad (50)$$

$$x_\kappa(h) = x_{0\kappa} + h(A_1 g_{1\kappa} + \cdots + A_l g_{l\kappa}) + h^{1/2}\xi_\kappa^{1/2}Y_{0\kappa}, \quad (51)$$

where $\{x_\mu\}$ denotes the set of variables $x_1, \cdots, x_N$. It is appropriate to take the covariance of the $Y_{i\kappa}$ as

$$\langle Y_{i\kappa}Y_{j\mu}\rangle = L_{ij}\delta_{\kappa\mu}; \quad (52)$$

or, equivalently, to write

$$Y_{i\kappa} = \sum_{j=1}^{m} \lambda_{ij}Z_{j\kappa}, \quad (53)$$

where the $Z_{j\kappa}$ are Nm independent Gaussian random variables of mean zero and variance unity. In general, $m = l + 1$, but it may be possible to construct an algorithm with smaller m, i.e., a $k_O l_S m_G$ scheme for vector SDEs.

Equation (51) may be expanded to any desired order, h^k, giving a deterministic and a stochastic part, $\tilde{\mathbf{S}}$. Once again, equations for the parameters are determined by demanding equality of the deterministic part to that of the expansion, eq. (44). Further equations result from equating the moments of $\mathbf{S}$ and $\tilde{\mathbf{S}}$. In general, there are more equations to be satisfied for sets than for a single variable because of the mixed partial derivatives. Note, however, that the parameters for the algorithm, A_i, β_{ij}, λ_{ij}, are not functions of the component index, κ.

Once an algorithm is available for vector SDEs, it can be applied to two other classes of SDEs. Consider the generalization of eq. (4) where $\mathbf{f}$ is time dependent

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) + \mathbf{A}(t). \quad (54)$$

By introducing an extra variable $x_{N+1} = t$ (i.e., $dx_{N+1}/dt = 1$), one can rewrite eq. (54) in the form eq. (42) as an $N + 1$ dimensional, autonomous vector SDE:

$$\frac{d\mathbf{y}}{dt} = \mathbf{F}(\mathbf{y}) + \mathbf{A}(t), \quad (55)$$

$$\mathbf{y} \equiv (x_1, x_2, \cdots, x_N, x_{N+1}), \quad (56)$$

$$\mathbf{F}(y) \equiv (f_1, f_2, \cdots, f_N, 1), \quad (57)$$

$$\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_N, 0). \quad (58)$$

In this case, the random variables $Y_{i,N+1}$ or $Z_{j,N+1}$ need not be generated since they are always multiplied by zero.

An nth order differential equation may also be integrated in a straightforward manner. Consider as an example the simple case

$$\frac{d^n x}{dt^n} + c_1(x)\frac{d^{n-1}x}{dt} + \cdots + c_n(x) = A(t). \tag{59}$$

This equation is equivalent to the n dimensional vector SDE

$$\frac{dy_1}{dt} = y_2, \tag{60}$$

$$\vdots$$

$$\frac{dy_{N-1}}{dt} = y_N, \tag{61}$$

$$\frac{dy_N}{dt} = -c_1(y_1)y_{N-1} - c_2(y_1)y_{N-2} - \cdots - c_1(y_1) + A(t). \tag{62}$$

Note that

$$\xi = (0, 0, \cdots, 0, \xi) \tag{63}$$

so that only one Gaussian variable is needed per time step. More complicated equations than eq. (59) are just as easily handled; e.g., equations nonlinear in the derivatives and equations with an explicit time dependence on the left.

V. PARAMETERS FOR VECTOR SDE ALGORITHMS

To second order, the equations for the vector algorithm parameters are identical with those of a single equation. Thus, the parameters given earlier in eq. (23) may be used for a vector $2_O2_S1_G$ scheme.

To third order, one new equation enters. All the eqs. (24) to (38) hold except that eq. (36) splits into two (because of the difference of mixed derivatives):

$$\sum_{i=2}^{3} A_i \sum_{j=1}^{i-1} \beta_{ij} L_{ij} = \frac{1}{6}, \tag{64}$$

$$\sum_{i=2}^{3} A_i \sum_{j=1}^{i-1} \beta_{ij} L_{jj} = \frac{1}{6}. \tag{65}$$

The one degree of freedom of the one-variable $3_O3_S2_G$ algorithm is now removed, but a solution might still exist. Unfortunately, no real solution can be found regardless of whether there are 2, 3, or 4 Gaussians.

In order to find a third-order algorithm, it was necessary to consider a $3_O4_S2_G$ procedure, that is, to add a stage. This leaves many degrees of freedom—in fact enough so that the deterministic part of the equation could be satisfied to fourth order with one degree of freedom left. Actually, these extra degrees of freedom only exist if one assumes [in the pattern of eqs. (39) to (40)] that

$$A_1 = 0, \tag{66}$$

$$L_{0i} = L_{ii} = \alpha_i, \qquad i = 2, 3, 4. \tag{67}$$

Since the parameter equations are nonlinear there are multiple families of solutions. We have not explored all the branches, but have looked particularly at a branch for which $\alpha_4 = 1$. This implies that $\lambda_4 = (1, 0, \cdots)$. In Table II we present two parameter sets which can be used for the $3_O4_S2_G$ algorithm with 4_O deterministic-part accuracy. We have the parameters for three other families of solutions but have not presented them because some parameters are large, i.e., $\gtrsim 5$. The degree of freedom is in the relation between λ_{11} and λ_{12} which is

$$\left(\sum_{i=2}^{4} A_i\alpha_i\beta_{i1}\right)\lambda_{11} + \left(\sum_{i=2}^{4} A_i\lambda_{i2}\beta_{i1}\right)\lambda_{12} = \left(\frac{1}{6}\right) - \sum_{i=3}^{4} A_i \sum_{j=2}^{i-1} \beta_{ij}L_{ij}. \tag{68}$$

All the parameters in this equation except λ_{11} and λ_{12} are determined by other equations. In Table II we present two solutions, for which $\lambda_{12} = 0$ and $\lambda_{11} = 0$, respectively.

From the point of view of computer time, a $3_O4_S2_G$ algorithm might be faster than a $3_O3_S4_G$ algorithm (if the latter existed); i.e., an extra functional evaluation may be faster than generating more Gaussians. For problems in which the effect of noise is small (small ξ) the $3_O4_S2_G$ algorithm, being fourth order in the deterministic part, would be more accurate.

VI. CONCLUSION

In Section IV, we considered various $k_Ok_Sm_G$ algorithms and found that for the one variable problem there were 5, 14, and 29 equations to be satisfied for $k = 2$, 3, and 4, respectively. On the other hand, the numbers of parameters available to satisfy these equations are maximally $(k + 1)^2 = 9$, 16, and 25, respectively. It appears that the number of equations is increasing more rapidly than the number of parameters. [The situation is complicated for a number of reasons: (*i*) assumptions like eqs. (39) and (40) seem capable of reducing the number of equa-

Table II—Parameters for a $3_O4_S2_G$ algorithm appropriate to vector SDEs

A_1	0.0	A_2	0.644468
A_3	0.194450	A_4	0.161082
β_{21}	0.516719	β_{31}	−0.397300
β_{32}	0.427690	β_{41}	−1.587731
β_{42}	1.417263	β_{43}	1.170469
λ_{01}	1.0	λ_{02}	0.0
λ_{11}	0.0	λ_{12}	0.271608
or			
λ_{11}	−0.567253	λ_{12}	0.0
λ_{21}	0.516719	λ_{22}	0.499720
λ_{31}	0.030390	λ_{32}	−0.171658
λ_{41}	1.0	λ_{42}	0.0

tions considerably; (*ii*) even when there are sufficient parameters real solutions do not always exist; and (*iii*) vector SDEs produce more equations with the same number of parameters.] To achieve a third-order algorithm for sets it was necessary to use four stages, and our failure to find a 4_O4_S algorithm in Section III probably means that at least a fifth stage is necessary. A similar situation occurs for deterministic equations, in which case more than k stages are needed when the order of accuracy is $k \geq 5$.[4]

Higher order methods can be achieved in other ways than by increasing the number of stages. One possibility, suggested in I, would be to adapt iterative multistep methods, e.g., of the Adams-Moulton type.[3] Another approach would be to use implicit Runge-Kutta methods in which later stage g_j's are used in earlier stage g_i's.[4] An l-stage implicit method would then require the self-consistent solution of l nonlinear equations for the l g_i's at each time step. For mildly nonlinear SDEs and small fluctuations of the stochastic parts, this could be more efficient than larger stage methods. It is known that deterministic implicit methods are capable of achieving kth order accuracy with fewer than k stages, so this could well be the case for SDEs also.

It is our hope that the work presented here and in I, besides providing some practical schemes for integrating SDEs, will stimulate further research on this interesting and important topic.

VII. ACKNOWLEDGMENT

H. S. Greenside would like to thank P. C. Hohenberg for bringing the subject matter of this paper to his attention.

APPENDIX

We briefly present further details of the solution of the equations for the parameters of the $3_O3_S2_G$ algorithm to illustrate the procedures which one follows in cases of higher order or more stages.

The seven independent equations to be solved, after assuming eqs. (39) and (40), are eqs. (24) to (28), (36), and (37). The eight unknowns may be taken as α_2, α_3, A_2, A_3, β_{32}, λ_{01}, λ_{11}, and λ_{12}. Other λ parameters are given by

$$\lambda_{21} = \alpha_2, \tag{69}$$

$$\lambda_{31} = \alpha_3, \tag{70}$$

$$\lambda_{22} = +(\alpha_2 - \alpha_2^2)^{1/2}, \tag{71}$$

$$\lambda_{32} = \pm(\alpha_3 - \alpha_3^2)^{1/2}. \tag{72}$$

(The use of the negative sign for λ_{22} changes all signs for the λ_{i2}'s, and is a trivial modification equivalent to changing the sign of Z_2.)

Equations (24) to (26) may be solved for A_1, A_2, and A_3 in terms of α_2 and α_3. Setting $A_1 = 0$ leads to

$$\alpha_3 = \frac{3\alpha_2 - 2}{6\alpha_2 - 3}. \tag{73}$$

By eqs. (71) and (72) we see that both α_2 and α_3 must be between zero and unit, for a real solution, which, coupled with eq. (73), implies inequality eq. (41).

The remaining parameters are solved for as follows: eq. (27) yields β_{32} in terms of parameters now dependent only on α_2; eq. (28) dictates $\lambda_{01} = 1$, which is true for every algorithm; eq. (36) is a linear equation for λ_{11}; and eq. (37) is a binomial equation for λ_{12}.

REFERENCES

1. (a) D. J. Wright, "The Digital Simulation of Stochastic Differential Equations," IEEE Trans. Auto. Control, *19* (February 1974), p. 75.
   (b) N. J. Rao, J. D. Borwankar, and D. Ramkrishna, "Numerical Solution of Itô Integral Equations," SIAM J. Control, *12* (February 1974), p. 124.
   (c) T. Yamada, "On the Approximation of Solutions of Stochastic Differential Equations," Z. Wahrscheinlichkeitstheorie Verw Gebiete, *36* (1976), p. 153.
   (d) W. Rümelin, unpublished work.
2. E. Helfand, "Numerical Integration of Stochastic Differential Equations," B.S.T.J., *58*, No. 10 (December 1979), pp. 2289–99.
3. A. H. Stroud, *Numerical Quadrature and Solution of Ordinary Differential Equations*, New York: Springer-Verlag, 1974.
4. J. C. Butcher, "Coefficients for the Study of Runge-Kutta Integration Processes," J. Australian Math. Soc., *3* (1963), p. 185.