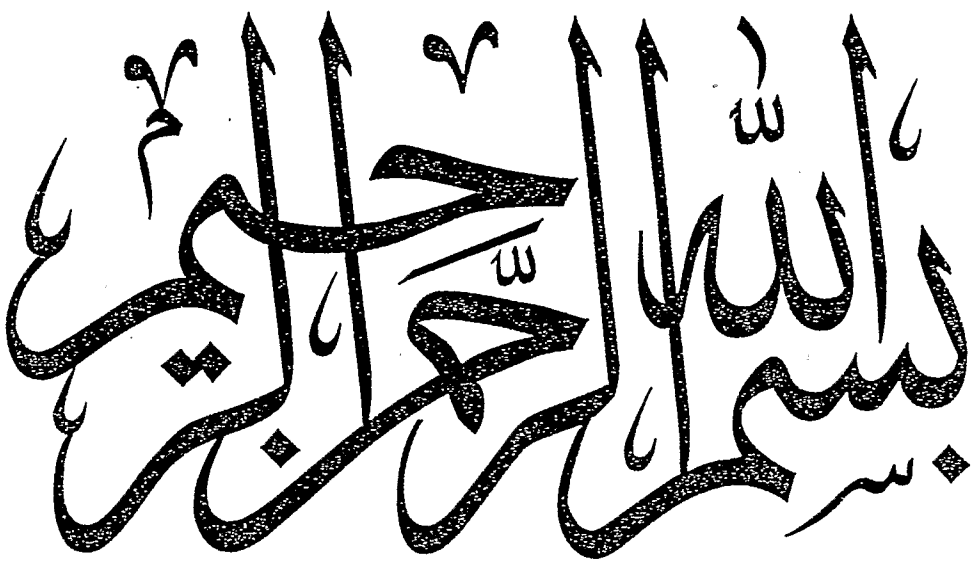

IN THE NAME OF GOD

Shahid Bahonar University of Kerman

Faculty of Mathematics and Computer

Department of Mathematics

Numerical Solution of Stochastic Differential Equations

By:

Ghasem Barid Loghmani

Supervisor:

Prof. M. Mohseni Moghadam

A Thesis Submitted in Partial Fullfilment

of the Requirement for the Degree of Doctor of Philosophy

(Ph.D) in Mathematics

June 2001

Dedicated to

my parents and my wife

Abstract

In this thesis we first give some information about stochastic calculus and stochastic differential equations (SDEs). Then we generalize the explicit Runge-Kutta methods for the numerical solutions of stochastic differential equations which was first introduced by K. Burrage and P.M. Burrage in 1996. More precisely, we will introduce implicit and semi-implicit 2-stage Runge-Kutta methods for the numerical solutions of SDEs. Based on the rooted trees theory it is shown that both methods are of strong order 1 with minimum principal error.

Also, we generalize the hybrid methods for numerical solution of SDEs. We use the idea of linear multistep formula (LMF) for SDEs which was introduced by L. Brugnano, K. Burrage and P.M. Burrage in 2000. The implementation of the methods, leading to a predictor-corrector approach.

Some numerical results of our methods are presented.

Acknowledgements:

I would like to take this opportunity to thank my supervisor professor M. Mohseni Moghadam for his support, encouragement and invalueable advices during this period of research.

I would like to express my thanks to professor M. Radjabalipour, Dr. R. Sabeti, Dr. M.Y. Rahimi Ardabili, Dr. M. Dehghan and Dr. A.R. Soheili for refereeing of the thesis and useful suggestions.

These acknowledgements would not be complete without thanks to my parents who first instilled in me the love of learning and my wife who first to learn me the love of life.

Ghasem Barid Loghmani

June 2001

Contents: Page

Chapter 1

Introduction

1.1 Overview

The area of ordinary differential equations (ODEs) is rich and well-researched with plenty of software packages and tools available for the numerical solution of such systems. ODEs arise as a description of a model of a physical system and are solved in order to provide answers to such questions as how the system is changing or developing, when change might occur, what effect a different starting point may have on the solution, and so on. Until recently, many models that have been developed to describe physical phenomena have ignored stochastic effects because of the difficulty in solution both in terms of the lack of suitable numerical methods and also the non-availability of sufficiently powerful computers. However, recently there has been much interest in developing numerical methods for the numerical solution of stochastic differential equations (SDEs) and this has meant that more realistic models are capable of being solved. Some areas where SDEs are used in modelling include investment finance, option pricing, turbulent diffusion, radio-astronomy, population dynamics (protein kinetics and genetics), experimental psychology, neuronal activity, Helicopter rotor, satellite orbit stability, biological waste treatment (hydrology and indoor air quality), seismology, structural mechanics, blood

clotting dynamics, cellular energetics and polymer dynamics. With the increase in computing power, such complex models can now be realised, and this has contributed to a burst of interest in the numerical solution of stochastic differential equations.

An ordinary differential equation such as

$$\dot{y} = \frac{dy}{dt} = f(t, y) \tag{1.1}$$

may be thought of as a degenerate form of a stochastic differential equation, as yet undefined, in the absence of randomness. It is therefore useful to review some of its basic properties. We could write (1.1) in the symbolic differential form

$$dy = f(t, y)dt \tag{1.2}$$

or more accurately as an integral equation

$$y(t) = y_0 + \int_{t_0}^{t} f(s, y(s))ds \tag{1.3}$$

where $y(t) = y(t; y_0, t_0)$ is a solution satisfying the initial condition $y(t_0) = y_0$. Some regularity assumption is usually made on f, such as lipschitz continuity, to ensure the existence of a unique solution $y(t; y_0, t_0)$ for each initial condition.

The inclusion of random effects in differential equations leads to two distinct classes of equations, for which the solution processes have differentiable

and nondifferentiable sample paths, respectively. They require fundamentally different methods of analysis. The first and simpler class arises when an ordinary differential equation has random coefficients, a random initial value or is forced by a fairly regular stochastic process, or when some combination of these holds. The equations are called *random differential equations* and are solved sample path by sample path as ordinary differential equations. The sample paths of the solution processes are then at least differentiable functions. As an example consider the linear random differential equation

$$\dot{y} = \frac{dy}{dt} = f(w)y + g(t,w) \tag{1.4}$$

where the forcing process g is continuous in t for each w. For an initial value $y_0(w)$ at $t = 0$, the solution is given by

$$y(t,w) = e^{f(w)t}(y_0(w) + \int_0^t e^{-f(w)s} g(s,w)ds). \tag{1.5}$$

Its sample paths are obviously differentiable functions of t.

The second class occurs when the forcing is an irregular stochastic process such as a Gaussian white noise. The equations are then written symbolically as stochastic differentials, but are interpreted as integral equations with Itô or Stratonovich stochastic integrals. They are called *stochastic differential equations*, which we shall abbreviate by SDEs, and in general their solutions

inherit the nondifferentiability of sample paths from the Wiener processes in the stochastic integrals. In many applications such equations result from the incorporation of either internally or externally originating random fluctuations in the dynamical description of a system. An example of the former is the molecular bombardment of a speck of dust on a water surface, which results in Brownian motion. The intensity of this bombardment does not depend on the state variables, for instance the position and velocity of the speck.

Einstein's explanation of observed Brownian motion during the first decade of the last century, attempts were made by Langevin and others to formulate the dynamics of such motions in terms of differential equations. The resulting equations were written in the form of

$$dy(t) = f(t, y(t))dt + g(t, y(t))\xi(t)dt \tag{1.6}$$

with a deterministic or averaged drift term (1.1) perturbed by a noisy, diffusive term $g(t, y(t))\xi(t)$, where $\xi(t)$ is the standard Gaussian random variables for each t and $g(t, y)$ a (generally) space-time dependent intensity factor. This symbolic differential was interpreted as an integral equation

$$y(t, w) = y(t_0, w) + \int_{t_0}^{t} f(s, y(s, w))ds + \int_{t_0}^{t} g(s, y(s, w))\xi(s, w)ds \tag{1.7}$$

for each smaple path. When extrapolated to a limit, the observations of Brow-

nian motion seemed to suggest that the covariance $C(t) = E(\xi(s)\xi(s+t))$ of the process $\xi(t)$ had a constant spectral density, that is with all time frequencies equally weighted in any Fourier transform of $C(t)$. Such a process became known as the Gaussian white noise, particular in the engineering literature. For the special case of (1.7) with $f \equiv 0, g \equiv 1$ we see that $\xi(t)$ should be the derivative of pure Brownian motion, that is the derivative of a Wiener process $W(t)$, thus suggesting that we could write (1.7) alternatively as

$$y(t,w) = y(t_0,w) + \int_{t_0}^{t} f(s,y(s,w))ds + \int_{t_0}^{t} g(s,y(s,w))dW(s,w). \qquad (1.8)$$

The problem with this is, as we will see in future, that a Wiener process $W(t)$ is nowhere differentiable, so strictly speaking the white noise process $\xi(t)$ does not exist as a conventional function of t, indeed, a flat spectral density implies that its covariance function $c(t)$ is a constant multiple of the Dirac delta function $\delta(t)$. Thus the second integral in (1.8) cannot be an ordinary Reimann or Lebesgue integral. Worse still, the continuous sample paths of a Wiener process are not of bounded variation on any bounded time interval, so the second integral in (1.8) cannot even be interpreted as a Reimann-Stieltjes integral for each sample path.

1.2 Numerical Methods for SDEs

Consider the general Itô SDE

$$dy(t) = f(t, y(t))dt + g(t, y(t))dW(t), \tag{1.9}$$

where f is the drift coefficient and g is the diffusion coefficient. Two particular cases of (1.9) are those in which the noise is multiplicative or additive. (If $g(t, y(t))$ depends linearly on $y(t)$ then the noise term is called multiplicative, while if g is constant the SDE has additive noise.)

There are two main classes of methods for solving such an equation numerically, namely one-step methods and multistep methods. A one-step method needs one starting value for commencing the computation and then proceeds by updating the numerical solution based only on information from the previous step and intermediate values within a step. On the other hand, multistep methods require several starting values (which need to be calculated by, for example, a low order one-step method), as the update values rely on past information from a number of steps.

The implementation of numerical methods for SDEs requires the sampling of Wiener increments to approximate the white noise in the SDE, and this is acheived by computer generation of pseudo-random numbers. For example the

simplest stochastic numerical approximation is the Euler-Maruyama method

$$y_{n+1} = y_n + h_n f(y_n) + g(y_n)\Delta W_n \tag{1.10}$$

where $h_n = t_{n+1} - t_n$ and $\Delta W_n = W(t_{n+1}) - W(t_n) \sim N(0, h_n)$. The noise increments ΔW_n are $N(0, h_n)$-distributed random variables, and can be generated numerically by using a pseudo-random number generator. Two efficient ways of generating these increments are by the Box-Muller method or by the Polar-Marsaglia method, each involves sampling from a Uniform distribution (over $[0, 1]$), and then applying a transformation with subsequent scaling by the factor $\sqrt{h_n}$. The Polar-Marsaglia method returns two samples each time it is used, and is the scheme used in the numerical implementations carried out in this thesis.