

Mean-square stability properties of an adaptive time-stepping SDE solver

H. Lamba and T. Seaman
Department of Mathematical Sciences,
George Mason University, MS 3F2,
4400 University Drive,
Fairfax, VA 22030, USA.

Phone: +1 (703) 993 1489

Fax: +1 (703) 993 1491

E-mail: hlamba@gmu.edu

Abstract

We consider stability properties of a class of adaptive time-stepping schemes based upon the Milstein method for stochastic differential equations with a single scalar forcing. In particular we focus upon mean-square stability for a class of linear test problems with multiplicative noise. We demonstrate that highly desirable stability properties can be induced in the numerical solution by the use of two realistic local error controls, one for the drift term and one for the diffusion.

Keywords Error control, mean-square stability, numerical integration, Milstein, Milstein-type methods, variable step-size, stochastic differential equations

1 Introduction

We investigate the numerical solution of the class of scalar linear, constant coefficient, stochastic differential equations (SDEs) with multiplicative noise, written in Itô form as

$$dX = \lambda X dt + \mu X dW, \quad X(0) = X_0 \neq 0 \quad (1.1)$$

for fixed initial data X_0 , $\lambda, \mu \in \mathbb{C}$ and with $W(t)$ a standard Wiener process. This class of test problems is a natural analogue to the linear test problems used to analyze stability properties (such as A-stability) of numerical schemes for ordinary differential equations (ODEs) and has been considered previously by several authors [14, 19, 27, 26, 23, 9, 7, 16, 15, 5] for schemes using a fixed time-step h . Linear test problems are of interest because complete analyses are often possible which, via linearization arguments, can provide insights into the behaviour of numerical schemes on more general classes of problem. Other investigations of mean-square stability with fixed time-steps include [2, 3, 22, 28, 1, 24, 29, 8, 30].

For adaptive time-stepping ODE solvers, it has long been known that adaptivity based upon local error control(s) can impart desirable stability properties for linear and nonlinear equations [12, 11], especially for 'stiff' problems, even when the underlying method is explicit and has a small stability region. This occurs because a consequence of the error control is to force the time-steps close to, but below, the linear stability limit. However, this phenomenon, which appears to be very common, is still not perfectly understood and there are very few rigorous stability results in this area. One important observation is that algorithms employing an error-per-unit step control appear to have better stability properties than their error-per-step counterparts [31].

The current state of both the theory and software for SDE solvers is at a very basic level, at least in comparison with solvers for ODEs. This is especially true for adaptive schemes, with several different approaches proposed in the literature (see for example [10, 17, 6, 20]) but little consensus as to the benefits or range of applicability of the

current options. Most attention has been paid to the issues of convergence and efficiency. In this paper we show that a realistic local error control strategy, using two error controls in conjunction with the (explicit) Milstein method, can effectively stabilize the numerical solution of (1.1) whenever the underlying exact solution is itself stable. Thus the adaptive algorithm can replicate the desirable mean-square stability properties of various implicit Euler-Maruyama and Milstein-based methods [27] when applied to (1.1) — the price to be paid is, of course, that the time-steps may be prohibitively small. Apart from the obvious theoretical interest, this property may be useful, for example, in the context of stiffness detection whereby a sophisticated algorithm uses an adaptive explicit method to advance the solution but changes to an implicit method when it is deemed more efficient to do so. In such a situation, the stability of the numerical algorithm over the entire integration time can then be assured, with only efficiency requirements determining the choice of method.

The paper is organized as follows. In Section 2 we introduce the notion of mean-square stability for the exact and numerical approximations to (1.1). We also recall the Milstein method and define the adaptive strategy to be employed. This strategy uses two local error controls, one involving only the drift term of (1.1) and the other only the diffusion term. Such dual error controls have already been proposed in the literature [20, 21] and have both theoretical and practical advantages. In Section 3 we then show that, using the local error controls introduced in Section 2, mean-square stability can be effectively achieved, even for moderately large tolerances τ , under very mild conditions on the adaptive algorithm. Finally, in Section 4 we present some numerical results that help confirm the analysis. We thus demonstrate that a feasible class of adaptive time-stepping algorithms, based upon dual error controls, can successfully mimic the mean-square stability of the test problems (1.1).

2 Basic definitions and the adaptive algorithm

The equation (1.1) with fixed λ, μ is defined to be (*asymptotically*) *mean-square stable* if

$$\lim_{t \rightarrow \infty} E(|X(t)|^2) = 0$$

where $E(\cdot)$ denotes the expected value over $W(t)$. The exact solution of (1.1) is

$$X(t) = X_0 \exp \left(\left(\lambda - \frac{1}{2} \mu^2 \right) t + \mu W(t) \right),$$

and mean-square stability occurs when [26]

$$\operatorname{Re}(\lambda) + \frac{1}{2} |\mu^2| < 0. \tag{2.1}$$

For alternative concepts of stability see [13, 25]. In the interests of clarity, we shall perform the analysis for $\lambda, \mu \in \mathbb{R}, \mu \geq 0$ and then state the results for the general complex case at the end of Section 3. Thus, for real coefficients the mean-square stability criterion becomes

$$\lambda + \frac{1}{2}\mu^2 < 0 \quad (2.2)$$

and the stability boundary is a straight line with slope -2 when plotting μ^2 against λ .

Let us now interpret a numerical solution as consisting of (accepted) time-steps h_n , $n = 0, 1, 2, \dots$ and output values X_n , where X_n approximates $X(t_n)$ and $t_n = \sum_{i=0}^{n-1} h_i$. Then we define mean-square stability for the numerical scheme as

$$\lim_{n \rightarrow \infty} E(|X_n|^2) = 0$$

provided that $X(0) = X_0$ is deterministic and $t_n \rightarrow \infty$ as $n \rightarrow \infty$.

The Milstein method, for a general autonomous SDE, written in Itô form as

$$dX_t = f(X(t)) dt + g(X(t)) dW, \quad X(0) = X_0 \quad (2.3)$$

is defined by

$$X_{n+1} = X_n + h_n f(X_n) + \Delta W_n g(X_n) + \frac{1}{2} g'(X_n) g(X_n) (\Delta W_n^2 - h_n) \quad (2.4)$$

where $\Delta W_n = W(t_{n+1}) - W(t_n)$. Thus when applied to (1.1) this becomes

$$X_{n+1} = X_n + h_n \lambda X_n + \Delta W_n \mu X_n + \frac{1}{2} \mu^2 (\Delta W_n^2 - h_n) X_n. \quad (2.5)$$

In what follows, it will be convenient to work in Stratonovich rather than Itô calculus. For this reason we define $\underline{f} = f - \frac{1}{2} g' g$ and $\Lambda = \lambda - \frac{1}{2} \mu^2$ and rewrite (2.4) and (2.5) as

$$X_{n+1} = X_n + h_n \underline{f}(X_n) + \Delta W_n g(X_n) + \frac{1}{2} g'(X_n) g(X_n) \Delta W_n^2 \quad (2.6)$$

and

$$X_{n+1} = X_n + h_n \Lambda X_n + \Delta W_n \mu X_n + \frac{1}{2} \mu^2 \Delta W_n^2 X_n \quad (2.7)$$

respectively.

We now define the local error estimates that will be controlled for our adaptive Milstein method. The idea of using two error controls, one based upon the drift and another on the diffusion, has been explored previously [21, 20]. In [21] strong convergence was proved for an adaptive Euler-Maruyama scheme employing a deterministic error control, irrespective of any additional error control criteria being used. Thus, such a splitting can be beneficial from a theoretical point of view. In [20] a dual error control was used as the basis for a

class of adaptive strategies that can be used to better control the local errors committed on each step, thereby improving the efficiency of the algorithm. The error estimates that we choose to control are motivated in a very similar manner to those in [20] and a brief explanation is now given.

The Milstein method has strong order 1 and so the leading order terms in the truncation error at each step are order $h^{\frac{3}{2}}$. In Stratonovich calculus these leading-order terms are [18]

$$J_{10}\underline{f}'g + J_{01}g'\underline{f} + \frac{1}{6}\Delta W_n^3 g''gg + \frac{1}{6}\Delta W_n^3 (g')^2g \quad (2.8)$$

where

$$J_{10} = \int_{t_n}^{t_n+h} \int_{t_n}^{s_1} \circ dW ds_1, \quad J_{01} = \int_{t_n}^{t_n+h} \int_{t_n}^{s_1} ds \circ dW(s_1).$$

Since g'' is not calculated and the Stratonovich integrals J_{10} and J_{01} are unknowable from samples of the Brownian motion, we use an approximate upper bound for the final term, namely, $\frac{1}{6}|\Delta W_n|^3 \|g'\|_\infty \|g'g\|_\infty$ as our first local error estimate. Removing the factor of $\frac{1}{6}$ for simplicity, this becomes $|\Delta W_n|^3 \mu^3 |X_n|$ for our test problems.

Note that this error estimate is based only upon the diffusion component of the SDE. The next highest terms in the truncation error in the Milstein method are of order h^2 . From these we choose the term $\frac{1}{2}h^2 \underline{f}'\underline{f}$ that is determined solely by the drift. This can be approximated, for example, by using the difference between the forward Euler method and the Heun method on the ODE $\frac{dX}{dt} = \underline{f}(X)$ resulting in a second error estimate for a general nonlinear SDE. For the test problems (1.1) this reduces exactly to $\frac{1}{2}h_n^2 \Lambda^2 X_n$ and so this, without the factor of $\frac{1}{2}$, will be the second quantity that we control. We could have instead chosen to control the quantity $\frac{1}{2}h^2 f'f$ (ie. using the Itô rather than Stratonovich drift) but this is of little consequence to our analysis. For further details of this point, the reader is directed to [20].

A complete algorithm is now described. A user-defined tolerance τ is specified, together with two positive functions $\sigma_i(\tau, h, X)$, $i = 1, 2$. A candidate time-step h and its corresponding ΔW will be accepted, and thus be defined as h_n and ΔW_n respectively, if and only if

$$h^2 \Lambda^2 |X_n| \leq \sigma_1(\tau, h, X_n) \quad (2.9)$$

and

$$|\Delta W|^3 \mu^3 |X_n| \leq \sigma_2(\tau, h, X_n). \quad (2.10)$$

If a candidate time-step is rejected because either (2.9) or (2.10) is violated then a smaller time-step is chosen until both conditions are satisfied. Once a suitable time-step is found an initial candidate time-step must be chosen for the next time-step and so forth.

Our choices for the functions σ_1 and σ_2 are

$$\sigma_1(\tau, h, X_n) = \tau h |X_n|$$

and

$$\sigma_2(\tau, h, X_n) = \tau^{\frac{3}{4}} h^{\frac{3}{4}} |X_n|.$$

The error control on the drift term is therefore just a standard error-per-unit-step criterion measuring the relative error. The diffusion control is similar and the reason for $h^{\frac{3}{4}}$ rather than h appearing in σ_2 will become apparent from the analysis of Section 3. The choice of exponent $\frac{3}{4}$ for the tolerance τ is purely to streamline the algebra. Our main results remain unchanged if, for example, we use different tolerances for each of the error controls and reduce them to zero under mild restrictions. For the class of problems (1.1), together with the above choices of σ_1 and σ_2 , the general error controls (2.9) and (2.10) simplify to

$$h\Lambda^2 \leq \tau \tag{2.11}$$

and

$$|\Delta W|^4 \mu^4 \leq \tau h. \tag{2.12}$$

Note that (2.12) can be regarded as the diffusion equivalent of the deterministic error-per-unit-step control (2.11).

The precise details of the time-step selection mechanisms will heavily impact the efficiency of the algorithm, but that issue is not considered here. Instead, in Section 3 we shall place a very mild restriction on the selection mechanism, and then make a definite choice for the numerical results of Section 4. The analysis relies only upon the fact that the error controls (2.11) and (2.12) are satisfied at each step and that $\lim_{n \rightarrow \infty} t_n = \infty$ with probability 1, which is then proved for the specific algorithm used in Section 4. Further examples of possible time-step selection strategies can be found in [20].

Finally, it is crucial that whenever a sampling of the Brownian motion is required at a new time t , all conditional probabilities induced by previously generated values must be respected. In other words, the sequence of time-steps is allowed to depend upon the Brownian forcing but the distribution of $W(t)$, for any $t > 0$, must be independent of the sequence of (attempted) time-steps. If not, then a biased stochastic forcing will result. A detailed description of these conditional probabilities is contained in [20] but the following simple example highlights the issues involved.

Let us suppose that the algorithm is on the first time-step, attempting to advance using a candidate step of length h , and that the Brownian increment over this interval is some number δ , computed from $\mathcal{N}(0, h)$ (i.e we have $W(0) = 0$, $W(h) = \delta$). If either of the error controls (2.11) and (2.12) are violated then a smaller time-step, say $h/2$, must be attempted. Therefore the value of $W(h/2)$ is required and this is drawn from the distribution $\mathcal{N}(\delta/2, h/4)$, and *not* $\mathcal{N}(0, h/2)$, since the Brownian motion is now tied down at $t = 0$ and $t = h$, forming a Brownian bridge for $0 \leq t \leq h$. Furthermore, the two Brownian increments $W(h/2)$ and $\delta - W(h/2)$ are not independent, either of each other or of the value δ . This has important consequences for the analysis of Section 3.

3 Mean-square stability analysis

We define $V_n = \Delta W_n / \sqrt{h_n}$ and so for the test problem (1.1),

$$X_{n+1} = X_n \left[1 + h_n \Lambda + \sqrt{h_n} \mu V_n + \frac{1}{2} h_n \mu^2 V_n^2 \right].$$

The distributions of the random variables V_n and h_n depend in some unspecified manner upon the time-step selection strategy employed. In particular, V_n is not necessarily a normalized Gaussian random variable, as would be the case if there were no step-size rejections. This is because the times at which the numerical solution is updated are not necessarily stopping times. Note however, that by our choice of a relative error control, V_n and h_n are independent of X_n . If, for fixed Λ and μ , we have

$$E \left(\left| \frac{X_{n+1}}{X_n} \right|^2 \right) = E \left(\left[1 + h_n \Lambda + \sqrt{h_n} \mu V_n + \frac{1}{2} h_n \mu^2 V_n^2 \right]^2 \right) \leq L < 1 \quad (3.1)$$

for some constant L and $\sum_{i=1}^{\infty} h_i = \infty$, then $\lim_{n \rightarrow \infty} (E |X_n|^2) = 0$ and numerical mean-square stability occurs.

The following lemma assumes that the algorithm is not ‘biased’ towards either positive or negative jumps in the Brownian forcing and will result in a greatly simplified stability criterion.

Lemma 3.1 *Assume that the choice of time-steps is independent of the sign of ΔW , ie. that replacing $W(t)$ by $-W(t)$ results in the same sequence of accepted time-steps. Then $E(h_n^p V_n^q) = 0$ for all $p \geq 0$ and q odd.*

Proof The error controls (2.11) and (2.12) are independent of the sign of ΔW and so the result follows immediately by symmetry. \square

After a little algebra, the condition (3.1) becomes

$$\Lambda + E(\mu^2 V_n^2) + E\left(\frac{1}{2} h_n \Lambda^2\right) + E\left(\frac{1}{2} h_n \mu^2 \Lambda V_n^2\right) + E\left(\frac{1}{8} h_n \mu^4 V_n^4\right) < 0 \quad (3.2)$$

where use has been made of Lemma 3.1 to remove the odd powers of V_n . Let us now define $E(V_n^2) = 1 + \epsilon$. The value of ϵ does in general depend upon Λ, μ and τ , and will be estimated for a specific algorithm in Section 4. The error controls (2.11) and (2.12), considered separately, imply that $E(h_n \Lambda^2) \leq \tau$ and $E(h_n \mu^4 V_n^4) \leq \tau$ respectively. Multiplying the two error controls together and then taking the square-root gives $E(h_n \mu^2 \Lambda V_n^2) \leq \tau$. Thus the last 3 terms of (3.2) are bounded in absolute value by $\tau/2, \tau/2$ and $\tau/8$ respectively and the numerical stability criterion becomes

$$\left(\lambda + \frac{1}{2} \mu^2\right) + \epsilon \mu^2 + \mathcal{O}(\tau) < 0. \quad (3.3)$$

Comparing this result to (2.2) we see that the boundary of the numerical stability region closely resembles that of the class of equations (1.1). If $\epsilon < 0$ everywhere along the boundary of (2.2) then the numerical stability region should be strictly larger than (2.2), except for some $\mathcal{O}(\tau)$ neighborhood of the origin.

Finally we remark that, for the case of complex drift and diffusion coefficients, replacing λ and μ in the error controls (2.11) and (2.12) by $|\lambda|$ and $|\mu|$ results in the stability criterion

$$(\operatorname{Re}(\lambda) + \frac{1}{2}|\mu^2|) + \epsilon(\operatorname{Re}(\mu))^2 + \mathcal{O}(\tau) < 0 \quad (3.4)$$

which should be compared to (2.1)

4 Numerical results

In this section we present numerical results using an adaptive scheme of the form described above. In particular we will estimate the quantity $E(V_n^2)$ for various choices of λ, μ and τ .

Our algorithm will employ a basic halving-and-doubling strategy. If a time-step h is rejected, the next candidate time-step is simply $h/2$. Whenever a time-step h is accepted, the solution is advanced and the first candidate time-step for the next step is $2h$. Together with the error controls (2.11) and (2.12) this completely defines an algorithm that satisfies the assumption of Lemma 3.1.

For this extremely simple algorithm we can immediately prove the following admissibility result.

Lemma 4.1 *The halving-and-doubling algorithm described above, for any fixed $\lambda \leq 0, \mu > 0$ and $\tau > 0$, will reach the end of a given finite time interval $[0, T]$ with probability 1.*

Proof Without loss of generality we suppose that $T \in \mathbb{N}$ and the set of allowable time-steps that can be used by the algorithm is $\mathcal{H} = \{2^k : k \in \mathbb{Z}\}$. The proof relies on the fact that, for the simple halving-and-doubling strategy used, a necessary condition for the integration to fail to reach time T is that the stochastic error control be violated at least once with step-size 2^{-k} , for every sufficiently large integer k . The probability that the adaptive algorithm rejects a candidate time-step, even once, with step-size h is bounded from above by the probability that the fixed step-size Milstein method, using step-size h , will fail to satisfy the diffusion error criterion at least once during the integration period. It is this probability that is proved to tend to zero as $h \rightarrow 0$.

First we note that the drift error control is satisfied for all $h < \tau/\Lambda^2$ and so we now focus upon the diffusion error control. Let the initial candidate time-step for the algorithm be $h_{\text{init}} \in \mathcal{H}$ and choose $h \in \mathcal{H}$ such that $h < \min(h_{\text{init}}, \tau/\Lambda^2)$. We now prove that, as

$h \rightarrow 0$, the probability of every step of the fixed time-step Milstein method satisfying (2.12), tends to 1.

For a fixed h , let us define $V_m = \Delta W_m / \sqrt{h}$, $m = 1, \dots, T/h$ where ΔW_m is the Brownian increment over the m^{th} interval of width h . Since we are considering the fixed time-step algorithm, the V_m are independent identically distributed standard Gaussian random variables, and the diffusion error control used over that subinterval will be satisfied if

$$|V_m| \leq \eta := \frac{1}{\mu} \sqrt[4]{\tau/h}. \quad (4.1)$$

We start from the elementary bound (see e.g. [4]) that for a standard Gaussian random variable V ,

$$P(V > \eta) < \frac{1}{\eta\sqrt{2\pi}} e^{-\eta^2/2} \quad \forall \eta > 0. \quad (4.2)$$

Now

$$\begin{aligned} P(|V_m| \leq \eta \quad \forall m = 1, \dots, T/h) &= 1 - P(\exists m : |V_m| > \eta) \\ &\geq 1 - \frac{T}{h} P(|V| > \eta) \text{ where } V \sim \mathcal{N}(0, 1) \\ &> 1 - \frac{T}{h} \sqrt{\frac{2}{\pi}} \frac{1}{\eta} e^{-\eta^2/2} \\ &= 1 - \frac{T\mu}{h} \sqrt{\frac{2}{\pi}} \sqrt[4]{\frac{h}{\tau}} e^{-\sqrt{\frac{\tau}{h}} \frac{1}{\mu^2}}. \end{aligned}$$

Since this quantity tends to 1 as $h \rightarrow 0$ we have the desired result. \square

Proving the above admissibility result for more general time-stepping strategies will depend crucially upon the precise details of the algorithm, but numerical experiments with more sophisticated time-step selection strategies strongly suggest that such results hold, at least under very mild conditions on the time-step selection process.

We now numerically estimate the quantity $E(V_n^2) := 1 + \epsilon$ for the above algorithm for various values of λ, μ and τ (note that the precise values of ϵ also depend upon the time-step selection procedure employed). Of particular interest are the values of ϵ on, or close to, the mean square stability boundary for the test problems (1.1) as these will help locate the stability boundary for the numerical method. However, since the stability boundary (2.2) is equivalent to

$$\Lambda + \mu^2 = 0$$

it follows from the error controls (2.11) and (2.12), via a simple scaling argument, that $E(V_n^2)$ is independent of λ, μ and τ (to within discretization effects) along this stability boundary. The computed value of $E(V_n^2)$ is found to be approximately 0.86, giving $\epsilon = -0.14$. The fact that $\epsilon < 0$ is to be expected since the diffusion error control will

Table 1: Numerical estimates of $E(V^2)$ for various values of λ, μ .

λ	μ	$E(V_n^2)$
-2	1.5	0.87
-2	1	0.89
-2	0.5	0.99
-4	2	0.87
-8	2	0.90
-16	2	0.96

preferentially sub-divide time intervals over which the absolute Brownian increment is unusually large. We also estimate $E(V^2)$ at other points within the stability region (2.2). Table 1 shows the results of moving into the stability region starting from the boundary at $\lambda = -2, \mu = 2$. All the results are tabulated for $\tau = 0.1$. As expected, moving further into the stability region results in the drift control dominating the choice of time-steps. This reduces the number of step rejections due to large Brownian increments and $E(V^2)$ tends to 1.

From these numerical results we infer that the numerical stability boundary can be approximated by the line

$$\lambda + 0.36\mu^2 < 0 \tag{4.3}$$

and thus strictly contains the stability region (2.2) except possibly in a neighborhood of $\mathcal{O}(\tau)$ around the point $\lambda = 0, \mu = 0$. Locating the numerical stability boundary more accurately than this would appear to be a very hard problem, but this approximation suffices to show that the numerical solution is mean-square stable whenever the underlying test problem is stable. Different time-stepping strategies will induce slightly different values of ϵ , and therefore different numerical stability regions, but qualitatively similar results have been obtained for other algorithms.

Acknowledgments

H.L. gratefully acknowledges the hospitality and financial support of the Fields Institute for Research in the Mathematical Sciences, Toronto, Canada.

References

- [1] S. S. Artemiev and T. A. Averina. *Numerical analysis of systems of ordinary and stochastic differential equations*. VSP, Utrecht, 1997.
- [2] S.S. Artemiev. Mean-square stability of numerical methods for solving stochastic differential equations. *Russian Acad. Sci. Dokl. Math.*, 48:539–542, 1994.

- [3] S.S. Artemiev. The mean-square stability of numerical methods for solving stochastic differential equations. *Russian J. Numer. Anal. Math. Modeling*, 9:405–416, 1994.
- [4] H. Bauer. *Probability Theory*. Walter de Gruyter, Berlin, 1996.
- [5] A. Bryden and D.J. Higham. On the boundedness of asymptotic stability regions for the stochastic theta method. *BIT*, 43(1):1–6, 2003.
- [6] K. Burrage and P. Burrage. A variable stepsize implementation for stochastic differential equations. *SIAM J. Sci. Comp.*, 24:848–864, 2003.
- [7] K. Burrage, P.M. Burrage, and T. Mitsui. Numerical solutions of stochastic differential equations—implementation and stability issues. *J. Comput. Appl. Math.*, 125(1-2):171–182, 2000.
- [8] K. Burrage and T. Tian. A note on the stability properties of the Euler methods for solving stochastic differential equations. *New Zealand J. Math.*, 29:115–127, 2000.
- [9] P.M. Burrage. *Runga-Kutta methods for stochastic differential equations*. PhD thesis, The University of Queensland, Brisbane, Queensland, Australia, January 1999.
- [10] J.G. Gaines and T.J. Lyons. Variable step size control in the numerical solution of stochastic differential equations. *SIAM J. Appl. Math.*, 57(5):1455–1484, 1997.
- [11] D. F. Griffiths. The dynamics of some linear multistep methods with step-size control. In *Numerical analysis 1987 (Dundee, 1987)*, pages 115–134. Longman Sci. Tech., 1988.
- [12] G. Hall. Equilibrium states of Runge-Kutta schemes. *ACM Trans. Math. Software*, 11:289–301, 1985.
- [13] R.Z. Has'minskii. *Stochastic stability of differential equations*. Sijthoff and Noordhoff, 1980.
- [14] D.B. Hernandez and R. Spigler. Convergence and stability of implicit Runge-Kutta methods for systems with multiplicative noise. *BIT*, 33:654–669, 1993.
- [15] D.J. Higham. A-stability and stochastic mean-square stability. *BIT*, 40(2):404–409, 2000.
- [16] D.J. Higham. Mean-square and asymptotic stability of the stochastic theta method. *SIAM J. Numer. Anal.*, 38(3):753–769, 2000.
- [17] N. Hofmann, T. Müller-Gronbach, and K. Ritter. Optimal approximation of stochastic differential equations by adaptive step-size control. *Math. Comp.*, 69(231):1017–1034, 2000.

- [18] P.E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*. Springer-Verlag, Berlin, 1992.
- [19] Y. Komori, Y. Saito, and T. Mitsui. Some issues in discrete approximate solution for stochastic differential equations. *Comp. Math. Appl.*, 28:269–278, 1994.
- [20] H. Lamba. An adaptive timestepping algorithm for stochastic differential equations. *J. Comp. Appl. Math.*, 161:417–430, 2003.
- [21] H. Lamba, J. Mattingly, and A.M. Stuart. Strong convergence of an adaptive Euler-Maruyama scheme for stochastic differential equations. Preprint (<http://math.gmu.edu/~harbir/lms.ps>).
- [22] W. P. Petersen. Some experiments on numerical simulations of stochastic differential equations and a new algorithm. *J. Comput. Phys.*, 113(1):75–81, 1994.
- [23] W.P. Petersen. A general implicit splitting for stabilizing numerical simulations of Itô stochastic differential equations. *SIAM J. Num. Anal.*, 35:1439–1451, 1998.
- [24] L.B. Ryashko and H. Schurz. Mean square stability analysis of some linear stochastic systems. *Dynam. Systems Appl.*, 6(2):165–189, 1997.
- [25] Y. Saito and T. Mitsui. T-stability of numerical schemes for stochastic differential equations. *World Sci. Ser. Appl. Math.*, 2:333–344, 1993.
- [26] Y. Saito and T. Mitsui. Stability analysis of numerical schemes for stochastic differential equations. *SIAM J. Numer. Anal.*, 33(6):2254–2267, 1996.
- [27] H. Schurz. Asymptotical mean square stability of an equilibrium point of some linear numerical solutions with multiplicative noise. *Stoch. Anal. App.*, 14:313–353, 1996.
- [28] H. Schurz. *Stability, stationarity, and boundedness of some implicit numerical methods for stochastic differential equations and applications*. Logos Verlag Berlin, Berlin, 1997.
- [29] H. Schurz. The invariance of asymptotic laws of linear stochastic systems under discretization. *ZAMM Z. Angew. Math. Mech.*, 79(6):375–382, 1999.
- [30] H. Schurz. Numerical analysis of stochastic differential equations without tears. In *Handbook of stochastic analysis and applications*, volume 163 of *Statist. Textbooks Monogr.*, pages 237–359. Dekker, New York, 2002.
- [31] A.M. Stuart and A.R. Humphries. The essential stability of local error control for dynamical systems. *SIAM J. Num. Anal.*, 32:1940–1971, 1995.