A step size control algorithm for the weak approximation of stochastic differential equations

Dominique Küpper, Jürgen Lehn, Andreas Rößler

Darmstadt University of Technology, Fachbereich Mathematik, Schlossgartenstr.7, D-64289 Darmstadt, Germany

Abstract

A variable step size control algorithm for the weak approximation of stochastic differential equations is introduced. The algorithm is based on embedded Runge-Kutta methods which yield two approximations of different orders with a negligible additional computational effort. The difference of these two approximations is used as an estimator for the local error of the less precise approximation. We prove the convergence of the proposed method with step size control by means of rooted tree analysis. Furthermore, some numerical results are presented to demonstrate the effectiveness of the introduced step size control method.

Key words: stochastic differential equation, step size control, embedded Runge-Kutta methods, weak approximation
MSC 2000: 60H10, 65C30, 60H35, 65L06, 65L50, 34F05

1 Introduction

Ordinary differential equations (ODEs) are a well-known instrument for the modelling of time-dependent dynamic systems. However in many fields, e.g. in finance or physics, stochastic effects occur. These can be taken into account by stochastic differential equations (SDEs). Since analytical solutions for SDEs are seldomly known, numerical methods have to be applied. Most common methods use a fixed step size and thus are not able to react to the characteristics of a solution path. Therefore in the present paper we introduce a variable step size algorithm based on embedded stochastic Runge-Kutta methods to work within the scope of weak approximation. Step size control algorithms have been considered for example for strong approximation by Burrage and Burrage [1], Lamba [10], Mauthner [11] and for weak approximation

by Szepessy, Tempone and Zouraris [16].

Let $(\Omega, \mathcal{F}, P)$ be a complete probability space and $(\mathcal{F}_t)_{t\geq 0}$ a filtration which fulfills the usual conditions. We consider the d-dimensional SDE in integral form

$$X_t = X_{t_0} + \int_{t_0}^{t} a(s, X_s)ds + \int_{t_0}^{t} b(s, X_s)dW_s \tag{1.1}$$

with an initial value $X_{t_0} = x_0$ which is bounded in $\mathcal{L}^{2p}$ for $p \geq 1$ and two measurable functions $a, b : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$. $(W_t)_{t\geq 0}$ is a one-dimensional Wiener process which is adapted w.r.t. the filtration $(\mathcal{F}_t)_{t\geq 0}$. Suppose that a and b fulfill a Lipschitz condition and a linear growth condition. Then there exists a unique solution process X for equation (1.1) (see e.g. [7]).

In this paper our concern is the approximation of moments or, more general, of functionals of the solution. Therefore we consider a discretization $I_h = \{t_0, t_1, \ldots, t_N\}$ of the interval $[t_0, T]$ so that $0 \leq t_0 < t_1 < \ldots < t_N = T$. For $h_n = t_{n+1} - t_n$ let $h = \max_{0\leq n\leq N-1} h_n$ be the maximum step size of the discretization. Then an approximation process $(Y_t)_{t\in I_h}$ converges weakly with order p to X at time T as $h \to 0$ if there exists a constant δ_0 and for each $f \in \mathcal{C}_P^{2(p+1)}(\mathbb{R}^d, \mathbb{R})$ a constant C_f (independent of h) so that

$$|E(f(X_T)) - E(f(Y_T))| \leq C_f h^p.$$

holds for $h \in]0, \delta_0[$. Here $\mathcal{C}_P^{2(p+1)}$ denotes the space of $2(p+1)$-times continuously differentiable functions whose derivatives up to order $2(p+1)$ have polynomial growth [7].

2 Embedded stochastic Runge-Kutta methods

Various derivative free Runge-Kutta type methods have been developed in recent years (see e.g. [2,9,17]). In this paper we follow the approach due to Rößler [12–14] who developed stochastic Runge-Kutta (SRK) methods for the weak approximation of both, Itô and Stratonovich SDEs, which can also be applied in the case of a multidimensional Wiener process. An explicit s-stage SRK method for calculating a numerical approximation process $(Y_t)_{t\in I_h}$ with $Y_n = Y_{t_n}$ of the solution of SDE (1.1) is given by the recursive formula $Y_0 = x_0$ and

$$Y_{n+1} = Y_n + \sum_{i=1}^{s} \alpha_i \, a(t_i^{(0)}, H_i^{(0)}) \, h_n + \sum_{i=1}^{s} \left(\gamma_i^{(1)} I_{(1)} + \gamma_i^{(2)} \frac{I_{(1,1)}}{\sqrt{h_n}} \right) b(t_i^{(1)}, H_i^{(1)}) \tag{2.1}$$

with $t_i^{(0)} = t_n + c_i^{(0)} h_n$, $t_i^{(1)} = t_n + c_i^{(1)} h_n$ and

$$\begin{aligned} H_i^{(0)} &= Y_n + \sum_{j=1}^{i-1} A_{ij}^{(0)}\, a(t_j^{(0)}, H_j^{(0)})\, h_n + \sum_{j=1}^{i-1} B_{ij}^{(1)\,(0)}\, b(t_j^{(1)}, H_j^{(1)})\, I_{(1)}, \\ H_i^{(1)} &= Y_n + \sum_{j=1}^{i-1} A_{ij}^{(1)}\, a(t_j^{(0)}, H_j^{(0)})\, h_n + \sum_{j=1}^{i-1} B_{ij}^{(3)\,(1)}\, b(t_j^{(1)}, H_j^{(1)})\, \sqrt{h_n} \end{aligned} \tag{2.2}$$

for $i = 1, \ldots, s$ and $n = 0, \ldots, N-1$. Here we use the random variables $I_{(1)} = \int_{t_n}^{t_{n+1}} dW_s$ and $I_{(1,1)} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_1} dW_s dW_{s_1}$. We set $A^{(0)} = (A_{ij}^{(0)})$, $A^{(1)} = (A_{ij}^{(1)})$, $B^{(1)\,(0)} = (B_{ij}^{(1)\,(0)})$, $B^{(3)\,(1)} = (B_{ij}^{(3)\,(1)})$, $\alpha^T = (\alpha_i)$, $\gamma^{(1)T} = (\gamma_i^{(1)})$ and $\gamma^{(2)T} = (\gamma_i^{(2)})$. In the following we denote by $p = (p_D, p_S)$ the order of the SRK method (2.1), where p_D and p_S with $p_D \geq p_S$ indicate the order of convergence if the SRK method is applied to a deterministic or a stochastic differential equation, respectively. Thus, the order p_S is guaranteed in any case.

The embedded SRK approximation $\hat{Y}$ only differs from Y in the coefficients α_i, $\gamma_i^{(1)}$ and $\gamma_i^{(2)}$ with $i = 1, \ldots, s$. Adding the approximation $\hat{Y}_0 = x_0$ and

$$\hat{Y}_{n+1} = Y_n + \sum_{i=1}^{s} \hat{\alpha}_i\, a(t_i^{(0)}, H_i^{(0)})\, h_n + \sum_{i=1}^{s} \left(\hat{\gamma}_i^{(1)} I_{(1)} + \hat{\gamma}_i^{(2)} \frac{I_{(1,1)}}{\sqrt{h_n}} \right) b(t_i^{(1)}, H_i^{(1)}) \tag{2.3}$$

with order $\hat{p} = (\hat{p}_D, \hat{p}_S)$ to the SRK method (2.1) of order $p = (p_D, p_S)$ such that $\hat{p}_D < p_D$ and $\hat{p}_S < p_S$ hold, yields the embedded SRK method with order $p(\hat{p}) = (p_D, p_S)((\hat{p}_D, \hat{p}_S))$. The associated Butcher tableau [3] is of the following form:

$c^{(0)}$	$A^{(0)}$	$B^{(1)\,(0)}$	
$c^{(1)}$	$A^{(1)}$	$B^{(3)\,(1)}$	
	α^T	$\gamma^{(1)T}$	$\gamma^{(2)T}$
	$\hat{\alpha}^T$	$\hat{\gamma}^{(1)T}$	$\hat{\gamma}^{(2)T}$

(2.4)

Later on we will use embedded SRK methods of order $p(\hat{p}) = (3,2)((2,1))$ for step size control. Theorem 6.4 in [12] (Theorem 2.6.2 in [14]) yields conditions for the coefficients such that the method has a prescribed order of convergence. In the following we give order conditions for weak convergence of order $p_S = 2$

for method (2.1) (for details see [13,14])

1. $\alpha^T e = 1$
2. $\gamma^{(2)^T} e = 0$
3. $(\gamma^{(1)^T} e)^2 = 1$
4. $\gamma^{(1)^T} B^{(3)^{(1)}} e = 0$
5. $\alpha^T (B^{(1)^{(0)}} e)^2 = \frac{1}{2}$
6. $\gamma^{(1)^T} (B^{(3)^{(1)}} (B^{(3)^{(1)}} (B^{(3)^{(1)}} e))) = 0$
7. $\alpha^T A^{(0)} e = \frac{1}{2}$
8. $(\gamma^{(1)^T} e)(\alpha^T B^{(1)^{(0)}} e) = \frac{1}{2}$
9. $(\gamma^{(1)^T} e)(\gamma^{(1)^T} A^{(1)} e) = \frac{1}{2}$
10. $\gamma^{(1)^T} (B^{(3)^{(1)}} (B^{(3)^{(1)}} e)) = 0$
11. $(\gamma^{(1)^T} e)(\gamma^{(1)^T} (B^{(3)^{(1)}} e)^2) = \frac{1}{2}$
12. $\gamma^{(1)^T} (B^{(3)^{(1)}} (A^{(1)} (B^{(1)^{(0)}} e))) = 0$
13. $\alpha^T ((B^{(1)^{(0)}} e)(B^{(1)^{(0)}} (B^{(3)^{(1)}} e))) = 0$
14. $\gamma^{(1)^T} ((B^{(3)^{(1)}} e)(A^{(1)} (B^{(1)^{(0)}} e))) = 0$
15. $\gamma^{(1)^T} (A^{(1)} (B^{(1)^{(0)}} (B^{(3)^{(1)}} e))) = 0$
16. $\gamma^{(1)^T} ((B^{(3)^{(1)}} e)(B^{(3)^{(1)}} (B^{(3)^{(1)}} e))) = 0$
17. $\gamma^{(1)^T} (B^{(3)^{(1)}} e)^3 = 0$
18. $\gamma^{(1)^T} (A^{(1)} (B^{(1)^{(0)}} e)) = 0$
19. $\alpha^T (B^{(1)^{(0)}} (B^{(3)^{(1)}} e)) = 0$
20. $\gamma^{(1)^T} (B^{(3)^{(1)}} (A^{(1)} e)) = 0$
21. $\gamma^{(1)^T} ((B^{(3)^{(1)}} e)(A^{(1)} e)) = 0$
22. $\gamma^{(2)^T} A^{(1)} e = 0$
23. $\gamma^{(2)^T} (B^{(3)^{(1)}} (B^{(3)^{(1)}} e)) = 0$
24. $\gamma^{(2)^T} (B^{(3)^{(1)}} e)^2 = 0$
25. $\gamma^{(2)^T} (A^{(1)} (B^{(1)^{(0)}} e)^2) = 0$
26. $\gamma^{(2)^T} B^{(3)^{(1)}} e = 1$
27. $\gamma^{(1)^T} (B^{(3)^{(1)}} (B^{(3)^{(1)}} e)^2) = 0$
28. $\gamma^{(2)^T} (A^{(1)} (B^{(1)^{(0)}} e)) = 0$

with an s-dimensional vector $e = (1, \ldots, 1)^T$ and where the product of vectors has to be calculated component-wise. Further define $c^{(0)} = A^{(0)} e$ and $c^{(1)} = A^{(1)} e$. We give two examples for coefficients of the embedded SRK method (2.1)–(2.3) resulting from these equations:

0								
1	1		$\frac{3-2\sqrt{6}}{5}$					
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{6+\sqrt{6}}{10}$	0				
0								
1	1		1					
1	1	0	-1	0				
$(3,2)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{2}$	$-\frac{1}{2}$
$(2,1)$	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	0	0	0	0

0								
1	1		$-\frac{1}{3}$					
$\frac{5}{12}$	$\frac{25}{144}$	$\frac{35}{144}$	$\frac{5}{6}$	0				
0								
$\frac{1}{4}$	$\frac{1}{4}$		$\frac{1}{2}$					
$\frac{1}{4}$	$\frac{1}{4}$	0	$-\frac{1}{2}$	0				
$(3,2)$	$\frac{1}{10}$	$\frac{3}{14}$	$\frac{24}{35}$	-1	1	1	0	1	-1
$(2,1)$	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	0	0	0	0

Table 1
Embedded SRK methods RI3W1 and RI5W1 with $p(\hat{p}) = (3,2)((2,1))$

3 Step size control algorithm

We now propose an algorithm where the step size control is not based on every single approximation path, but on the mean of all paths generated for the Monte Carlo simulation. This approach is justified by the fact that in the end we are interested in the approximation of the expectation of the solution and not in a pathwise approximation. A step size control which was applied to each single approximation path seperately was also implemented, but did not yield as good results as the algorithm we now describe. Within each step of the embedded SRK method (2.1)–(2.3) described in Section 2 we obtain two types of approximations of the solution: The more precise ones Y and the less precise ones $\hat{Y}$. Starting from t_n for each $k = 1, \ldots, M$ with the approximation values $Y_{n,k}$ our algorithm calculates the corresponding approximations $Y_{n+1,k}$ and $\hat{Y}_{n+1,k}$ with a step size h, where M is the number of different generated paths for the Monte Carlo simulation. Then, after applying a general functional $f : \mathbb{R}^d \to \mathbb{R}^q$ to each approximation value, the algorithm provides the two estimates for the expectations

$$E(f_i(Y_{n+1})) = \frac{1}{M} \sum_{k=1}^{M} f_i(Y_{n+1,k}) \quad \text{and} \quad E(f_i(\hat{Y}_{n+1})) = \frac{1}{M} \sum_{k=1}^{M} f_i(\hat{Y}_{n+1,k})$$

for $i = 1, \ldots, q$. Then $\|E(f(Y_{n+1})) - E(f(\hat{Y}_{n+1}))\|$ serves as an estimate for the local error of the less precise approximation. For $i = 1, \ldots, q$ we choose tol_i as

$$tol_i = Atol_i + \max\{|E(f_i(Y_n))|, |E(f_i(Y_{n+1}))|\} Rtol_i \tag{3.1}$$

with tolerances $Atol_i$ for the absolute and $Rtol_i$ for the relative error prescribed by the user. We want the step size routine to accept only steps where

$$|E(f_i(Y_{n+1})) - E(f_i(\hat{Y}_{n+1}))| \leq tol_i, \quad i = 1, \ldots, q, \tag{3.2}$$

holds. The step size control calculates the optimal step size such that

$$err = \sqrt{\frac{1}{q} \sum_{i=1}^{q} \left(\frac{E(f_i(Y_{n+1})) - E(f_i(\hat{Y}_{n+1}))}{tol_i} \right)^2} \approx 1 \tag{3.3}$$

is fulfilled. As an estimate for the local error

$$err \approx C \cdot h^{\hat{p}_S + 1} \tag{3.4}$$

holds with some constant $C > 0$ and where $\hat{p}_S$ is the order of the less precise approximation $\hat{Y}_{n+1}$. Since we require $err \approx 1$, the optimal step size h_{opt} has to be chosen such that $1 \approx C \cdot h_{opt}^{\hat{p}_S + 1}$ is fulfilled. This implies

$$h_{opt} = h \left(\frac{1}{err} \right)^{\frac{1}{\hat{p}_S + 1}}.$$

In the deterministic setting Hairer, Nørsett and Wanner [6] propose a multiplication of h_{opt} by a safety factor $fac < 1$ (e.g. $fac = 0.8$) to prevent strong oscillations of the step size. To avoid a too fast growth or reduction of the step size we also introduce factors $facmax$ and $facmin$, so that the new step size finally is calculated as

$$h_{new} = h \cdot \min\left(facmax, \max\left(facmin, fac \cdot \left(\frac{1}{err}\right)^{\frac{1}{\hat{p}_S+1}}\right)\right). \tag{3.5}$$

If $err \leq 1$ holds, the current step with step size h is accepted for all M different paths and the next step is calculated with the proposed step size h_{new}. However if $err > 1$, the current step is repeated with the new step size h_{new} for all paths.

4 Simulation of the conditional distributions of $I_{(1)}$ and $I_{(1,1)}$

The SRK method uses the random variables $I_{(1)}$ and $I_{(1,1)}$. Since $I_{(1)} \sim \mathcal{N}(0, h)$ for step size h one gets $I_{(1)}$ and $I_{(1,1)}$ by means of the transformation

$$I_{(1)} = \sqrt{h} \cdot g \quad \text{and} \quad I_{(1,1)} = \frac{1}{2}((I_{(1)})^2 - h) \tag{4.1}$$

with a standard normal random variable g generated by some random number generator. As we want to use a step size control algorithm, we need to simulate random variables with a conditional distribution: In case that a step with step size h is rejected, the algorithm has to repeat the step with a smaller step size $h_{new} < h$. Thus the problem of simulating

$$I_{(1),t_n,\tau_2} = \int_{t_n}^{\tau_2} dW_s \quad \text{and} \quad I_{(1,1),t_n,\tau_2} = \int_{t_n}^{\tau_2}\int_{t_n}^{s_1} dW_s dW_{s_1}$$

under the condition of the already known realisations of the random variables

$$I_{(1),t_n,\tau_1} = \int_{t_n}^{\tau_1} dW_s \quad \text{and} \quad I_{(1,1),t_n,\tau_1} = \int_{t_n}^{\tau_1}\int_{t_n}^{s_1} dW_s dW_{s_1}$$

arises, where $\tau_1 = t_n + h$ and $\tau_2 = t_n + h_{new} < \tau_1$. Since $I_{(1,1)}$ can be calculated from $I_{(1)}$ the problem we are going to solve in the following reduces to the simulation of $I_{(1),t_n,\tau_2}$ given $I_{(1),t_n,\tau_1}$. Let t_n be the current starting point for the next approximation. If a step with step size h was tried, but rejected and if h_{new} is the step size to try next the integration interval $[t_n, \tau_1]$ degenerates to $[t_n, \tau_2] \cup [\tau_2, \tau_1]$. It must be pointed out that the Wiener process at the point τ_1 is already known, i.e. we have $I_{(1),t_n,\tau_1} = \sqrt{h} \cdot g$. Then we get the following

representation for $I_{(1),t_n,\tau_2}$ and $I_{(1),\tau_2,\tau_1}$ [5]

$$\begin{pmatrix} I_{(1),t_n,\tau_2} \\ I_{(1),\tau_2,\tau_1} \end{pmatrix} = A \begin{pmatrix} N \\ g \end{pmatrix} \tag{4.2}$$

with a 2×2-matrix $A = (a_{ij})$ and a standard normal random variable N independent from g. For A

$$\begin{aligned} &Var((I_{(1),t_n,\tau_2}, I_{(1),\tau_2,\tau_1})^T) \\ &\qquad = Cov((I_{(1),t_n,\tau_2}, I_{(1),\tau_2,\tau_1})^T, (I_{(1),t_n,\tau_2}, I_{(1),\tau_2,\tau_1})^T) = AA^T \end{aligned} \tag{4.3}$$

holds. The object is now to determine the matrix A. The covariance matrix is given by

$$Cov((I_{(1),t_n,\tau_2}, I_{(1),\tau_2,\tau_1})^T, (I_{(1),t_n,\tau_2}, I_{(1),\tau_2,\tau_1})^T) = \begin{pmatrix} h_{new} & 0 \\ 0 & h - h_{new} \end{pmatrix}. \tag{4.4}$$

Thus considering the equation

$$I_{(1),t_n,\tau_1} = \sqrt{h} \cdot g = I_{(1),t_n,\tau_2} + I_{(1),\tau_2,\tau_1} \tag{4.5}$$

yields the following system of equations for the elements of A:

$$\begin{aligned} a_{11} + a_{21} &= 0 \\ a_{12} + a_{22} &= \sqrt{h} \\ a_{11}^2 + a_{12}^2 &= h_{new} \\ a_{21}^2 + a_{22}^2 &= h - h_{new} \\ a_{11}a_{21} + a_{12}a_{22} &= 0 \end{aligned} \tag{4.6}$$

The first two equations can be derived from (4.2) and (4.5), the last three result from (4.3) and (4.4). As a solution of (4.6) we obtain

$$\begin{aligned} a_{11} &= \sqrt{h - h_{new} - \frac{(h - h_{new})^2}{h}} & a_{21} &= -a_{11} \\ a_{12} &= \sqrt{h} - \frac{h - h_{new}}{\sqrt{h}} & a_{22} &= \frac{h - h_{new}}{\sqrt{h}}. \end{aligned}$$

Therefore, the increments finally result from (4.2) and can be simulated as follows:

$$I_{(1),t_n,\tau_{n2}} = \sqrt{h - h_{new} - \frac{(h-h_{new})^2}{h}} \cdot N + (\sqrt{h} - \frac{h-h_{new}}{\sqrt{h}}) \cdot g \qquad (4.7)$$

$$I_{(1),\tau_{n2},\tau_{n1}} = -\sqrt{h - h_{new} - \frac{(h-h_{new})^2}{h}} \cdot N + \frac{h-h_{new}}{\sqrt{h}} \cdot g \qquad (4.8)$$

5 Convergence

In order to prove the convergence of the SRK method with step size control we need a representation of the exact solution of SDE (1.1) and of the numerical solution as well. Since Taylor expansions are much more complex in the stochastic than in the deterministic setting we use the rooted tree theory instead to handle this task in an easier way (for details see [12,14,15]). In the following we briefly sketch the rooted tree theory for the autonomous version of the Itô SDE (1.1). Let $\mathbf{t}$ be an S-tree (stochastic tree) with $l = l(\mathbf{t})$ nodes. We choose the set of colors for the nodes as $\mathcal{A} = \{\tau, \sigma\}$, where $\tau =$ ● is called a deterministic and $\sigma =$ ○ a stochastic node. The set of all such trees is denoted by TS. Figure 1 shows some examples for S-trees.

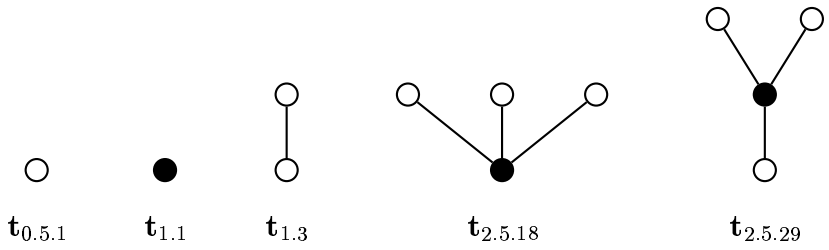


Fig. 1. Examples for S-trees

For an S-tree $\mathbf{t}$ let $d(\mathbf{t})$ denote the number of deterministic and $s(\mathbf{t})$ the number of stochastic nodes. Then the order $\rho(\mathbf{t})$ of the tree $\mathbf{t}$ is defined by $\rho(\mathbf{t}) = d(\mathbf{t}) + \frac{1}{2}s(\mathbf{t})$. For some of the examples in Figure 1 the order is calculated as $\rho(\mathbf{t}_{0.5.1}) = 0.5$, $\rho(\mathbf{t}_{1.3}) = 1.0$ and $\rho(\mathbf{t}_{2.5.18}) = 2.5$. We get a representation of $\mathbf{t}$ through $[\mathbf{t}_1, \ldots, \mathbf{t}_\lambda]$ or $\{\mathbf{t}_1, \ldots, \mathbf{t}_\lambda\}$ if the tree is formed by joining the subtrees $\mathbf{t}_1, \ldots, \mathbf{t}_\lambda$ each with a single branch to a common node of type τ or σ, respectively. Recursive application of the described technique leads to $\mathbf{t}_{1.1} = \tau$, $\mathbf{t}_{1.3} = \{\sigma\}$ and $\mathbf{t}_{2.5.29} = \{[\sigma, \sigma]\}$. We define $\gamma(\mathbf{t}) = 1$ if $l(\mathbf{t}) = 1$ and $\gamma(\mathbf{t}) = l(\mathbf{t}) \prod_{i=1}^{\lambda} \gamma(\mathbf{t}_i)$ if $l(\mathbf{t}) > 1$. Thus we obtain $\gamma(\mathbf{t}_{1.1}) = 1$, $\gamma(\mathbf{t}_{2.5.18}) = 4$ and $\gamma(\mathbf{t}_{2.5.29}) = 12$. The cardinality of $\mathbf{t}$ is denoted by $\alpha(\mathbf{t})$ and is equal to 1 for each S-tree in Figure 1 (see [12,14] for details).

For every rooted tree $\mathbf{t}$ there exists a corresponding elementary differential which is defined recursively by $F(\tau)(x) = a(x)$ and $F(\sigma)(x) = b(x)$ for single

nodes and by

$$F(\mathbf{t})(x) = \begin{cases} a^{(\lambda)}(x) \cdot (F(\mathbf{t}_1)(x), \ldots, F(\mathbf{t}_\lambda)(x)) & \text{for } \mathbf{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_\lambda] \\ b^{(\lambda)}(x) \cdot (F(\mathbf{t}_1)(x), \ldots, F(\mathbf{t}_\lambda)(x)) & \text{for } \mathbf{t} = \{\mathbf{t}_1, \ldots, \mathbf{t}_\lambda\} \end{cases} \tag{5.1}$$

for trees with more than one node. Here $a^{(\lambda)}$ and $b^{(\lambda)}$ define symmetric linear differential operators, that means the subtrees $\mathbf{t}_1, \ldots, \mathbf{t}_\lambda$ can be chosen in an arbitrary order. For ease of notation we drop the argument x and thus obtain for the Jth component of the examples $F(\mathbf{t}_{1.1})^J = a^J$, $F(\mathbf{t}_{1.3})^J = \sum_{K=1}^d \frac{\partial b^J}{\partial x^K} b^K$ and $F(\mathbf{t}_{2.5.29})^J = \sum_{K,L,M=1}^d \frac{\partial b^J}{\partial x^K} \frac{\partial^2 a^K}{\partial x^L \partial x^M} b^L b^M$ (see [12,14,15]).

Since the Taylor expansion of the numerical solution contains the coefficients of the SRK method, we define a coefficient function Φ_S. Therefore we make use of the notation

$$\begin{aligned} z^{(0)} &= \alpha h_n & Z^{(0)(0)} &= A^{(0)} h_n & Z^{(0)(1)} &= B^{(1)\,(0)} I_{(1)} \\ z^{(1)} &= \gamma^{(1)} I_{(1)} + \gamma^{(2)} \frac{I_{(1,1)}}{\sqrt{h_n}} & Z^{(1)(0)} &= A^{(1)} h_n & Z^{(1)(1)} &= B^{(3)\,(1)} \sqrt{h_n} \end{aligned}$$

Then Φ_S assigns an elementary weight to every tree $\mathbf{t}$ recursively by

$$\Phi_S(\mathbf{t}) = \begin{cases} z^{(0)^T} \prod_{i=1}^\lambda \Psi^{(0)}(\mathbf{t}_i) & \text{if } \mathbf{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_\lambda] \\ z^{(1)^T} \prod_{i=1}^\lambda \Psi^{(1)}(\mathbf{t}_i) & \text{if } \mathbf{t} = \{\mathbf{t}_1, \ldots, \mathbf{t}_\lambda\} \end{cases},$$

where $\tau = [\emptyset]$, $\sigma = \{\emptyset\}$ with $\Psi^{(\kappa)}(\emptyset) = e$, and

$$\Psi^{(\kappa)}(\mathbf{t}) = \begin{cases} Z^{(\kappa)(0)} \prod_{i=1}^\lambda \Psi^{(0)}(\mathbf{t}_i) & \text{if } \mathbf{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_\lambda] \\ Z^{(\kappa)(1)} \prod_{i=1}^\lambda \Psi^{(1)}(\mathbf{t}_i) & \text{if } \mathbf{t} = \{\mathbf{t}_1, \ldots, \mathbf{t}_\lambda\} \end{cases}$$

for $\kappa = 0, 1$ with $e = (1, \ldots, 1)^T$ and $\Phi_S(\emptyset) = 1$. The product of vectors in the definition of $\Psi^{(\kappa)}$ is defined by component-wise multiplication.

In the following proposition we give a weak convergence result for the sequence of approximations calculated by the SRK method with an arbitrary step size control. Without loss of generality we assume ${\gamma^{(1)}}^T e = 1$ for the coefficients in (2.4).

Proposition 5.1 *Let the coefficients of the SRK method (2.1) satisfy the 28 conditions for a method converging weakly with order 2.0. Assume ${\gamma^{(1)}}^T e = 1$ and let $f : \mathbb{R}^d \to \mathbb{R}$ satisfy a Lipschitz condition. Then the SRK method (2.1) yields approximations converging to the solution of SDE (1.1) in the weak sense, as long as the maximum step size converges to 0, even if the discretization points are not stopping times.*

Proof. As a first step we prove that an SRK method (2.1) of weak order 2.0 is also a method of strong order 1.0. Therefore we show at first that the method converges strongly with order 1.0: Consider the stochastic rooted trees $\mathbf{t}_{0.5.1} = \sigma$, $\mathbf{t}_{1.1} = \tau$ and $\mathbf{t}_{1.3} = \{\sigma\}$ with order $\rho(\mathbf{t}) \leq 1.0$. Then, applying Corollary 5.6 from [12] (Corollary 2.5.9 in [14]), the Taylor expansion of the Jth component of the SRK method (2.1) for $t = t_0 + h$ is given by

$$\begin{aligned}
Y_t^J =& X_{t_0}^J + \sum_{\substack{\mathbf{t}\in TS \\ \rho(\mathbf{t})\leq 1}} \frac{\alpha(\mathbf{t})\gamma(\mathbf{t})\Phi_S(\mathbf{t})F(\mathbf{t})(X_{t_0})^J}{l(\mathbf{t})!} + \mathcal{R}^J \\
=& X_{t_0}^J + \Phi_S(\mathbf{t}_{1.1})\, F(\mathbf{t}_{1.1})(X_{t_0})^J + \Phi_S(\mathbf{t}_{0.5.1})\, F(\mathbf{t}_{0.5.1})(X_{t_0})^J \\
& + \Phi_S(\mathbf{t}_{1.3})\, F(\mathbf{t}_{1.3})(X_{t_0})^J + \mathcal{R}^J \\
=& X_{t_0}^J + \alpha^T e\, a^J(X_{t_0})\, h + \left({\gamma^{(1)}}^T e\, I_{(1)} + {\gamma^{(2)}}^T e\, \frac{I_{(1,1)}}{\sqrt{h}} \right) b^J(X_{t_0}) \\
& + \left({\gamma^{(1)}}^T {B^{(3)}}^{(1)} e\, \sqrt{h}\, I_{(1)} + {\gamma^{(2)}}^T {B^{(3)}}^{(1)} e\, I_{(1,1)} \right) \sum_{K=1}^{d} \frac{\partial b^J(X_{t_0})}{\partial x^K} b^K(X_{t_0}) + \mathcal{R}^J
\end{aligned}$$

with $\|E(\mathcal{R})\| = \mathcal{O}(h^2)$ and $(E(\|\mathcal{R}\|^2))^{1/2} = \mathcal{O}(h^{1.5})$. From the weak order 2.0 conditions follows that $\alpha^T e = 1$, ${\gamma^{(1)}}^T e = 1$, ${\gamma^{(2)}}^T e = 0$, ${\gamma^{(1)}}^T {B^{(3)}}^{(1)} e = 0$ and ${\gamma^{(2)}}^T {B^{(3)}}^{(1)} e = 1$ holds. Since $I_{(1)}$ is an increment of the Wiener process and $I_{(1,1)} = \frac{1}{2}(I_{(1)}^2 - h)$, we have a SRK method with an expansion of type

$$Y_t^J = X_{t_0}^J + a^J(X_{t_0})\, h + b^J(X_{t_0})\, I_{(1)} + \frac{1}{2} \sum_{K=1}^{d} \frac{\partial b^J(X_{t_0})}{\partial x^K} b^K(X_{t_0})\, (I_{(1)}^2 - h) + \mathcal{R}^J.$$

Therefore, the SRK method (2.1) of weak order 2.0 is also a method of strong order 1.0. Thus Corollary 4.4 in [4] can be applied which yields the strong convergence of the SRK method even if the discretization points are not stopping times. Finally, for arbitrary $f : \mathbb{R}^d \to \mathbb{R}$ satisfying a Lipschitz condition, the weak convergence follows from strong convergence and Jensen's inequality with a Lipschitz constant $L > 0$:

$$\|E(f(X_T) - f(Y_T))\| \leq L \cdot E(\|X_T - Y_T\|) \to 0 \quad \text{as} \quad h \to 0.$$

□

6 Numerical results

In this section we compare the embedded SRK method (2.1)–(2.3) of weak order $(3, 2)((2, 1))$, that means $p_S = 2$ and $\hat{p}_S = 1$, with the step size control presented in section 3 to the same SRK method of weak order $(3, 2)$ with fixed step size. We denote these methods by M1 and M2, respectively. We proceed

as follows: We choose a starting step size h_{start} and tolerances $Atol$ and $Rtol$ for M1 and compute $M = 900.000$ approximation paths simultaneously with the same choice of step sizes given by the step size control algorithm. The safety factors are chosen as $fac = 0.8$, $facmax = 2.0$ and $facmin = 0.5$. The fixed step size for M2 is afterwards calculated by

$$h_{fix} = \frac{T - t_0}{S_{tried}} \tag{6.1}$$

with starting time t_0 and end time T of the integration interval and where S_{tried} denotes the number of steps tried by M1. Then we calculate 900.000 approximation paths with M2 and compute means at the discretization points as we did before with method M1. Consequently, M1 and M2 are charged with the same computational effort. Since the adaptive discretization times of method M1 do not necessarily coincide with the equidistant discretization times of M2, it is not possible to make a direct comparison between the results of M1 and M2 in the discretization points. In addition there do not exist analytical solutions for our test problems in general. So we calculate a linearly interpolated approximation with a very small step size as a reference solution. In the following we denote by SW_{min} the minimum and by SW_{max} the maximum step size used by the step size algorithm during the simulation. Furthermore let S_{acc} and S_{rej} be the number of accepted and rejected steps, respectively. All approximations are calculated with the SRK scheme RI3W1. However the application of RI5W1 yields assimilable results.

6.1 Test problem 1

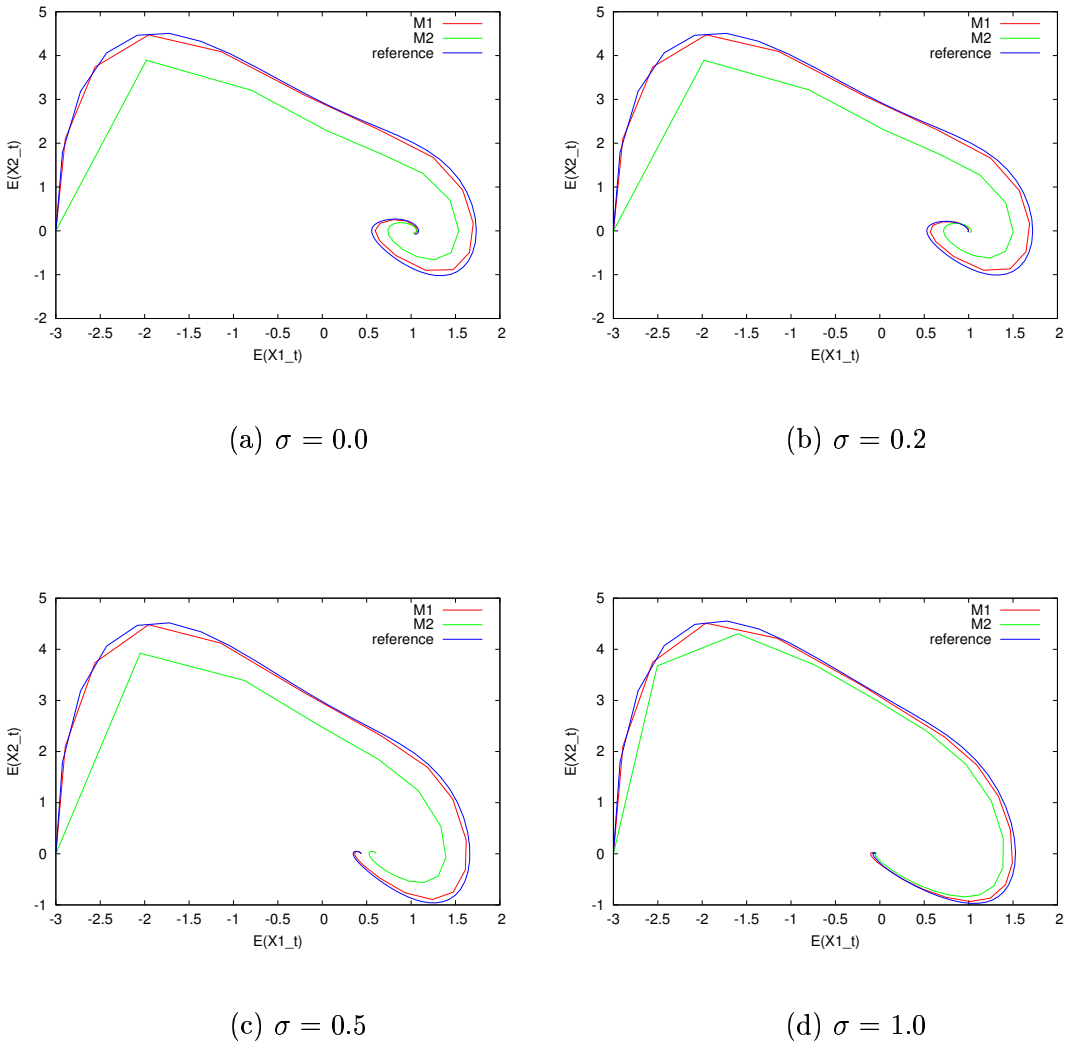
The *Duffing-Van der Pol oscillator* [7] is described by a 2-dimensional Itô stochastic differential equation

$$\begin{aligned} dX_t^1 &= X_t^2\, dt \\ dX_t^2 &= (X_t^1(\alpha - (X_t^1)^2) - X_t^2)\, dt + \sigma X_t^1\, dW_t \end{aligned} \tag{6.2}$$

with $\alpha \in \mathbb{R}$ and a parameter $\sigma \geq 0$ which specifies the intensity of the stochastical influence to the equation. For the simulation of the solution of equation (6.2) the functional f is chosen as the identity.

Figure 2 shows approximations with increasing stochastic influence σ. For the simulation we use $\alpha = 1.0$, the integration interval $I = [0, 8]$, the starting step size $h_{start} = 0.15$, the initial value $X_0 = (-3, 0)^T$, and the tolerances $Atol_i = 0.001$ and $Rtol_i = 0.05$, $i = 1, 2$. Considering the Figures 2(a) and 2(b) where the stochastic influence is rather small we do not realize any strong changes between both of them whereas for $\sigma = 0.5$ and $\sigma = 1.0$ we note a successive reduction of the step size. These considerations are verified by Ta-

ble 2, where we observe a declining maximum step size and an increase of the number of accepted steps S_{acc}. For any chosen σ we observe that method M1 yields better results than M2 in comparison to the reference solution.



(a) $\sigma = 0.0$ (b) $\sigma = 0.2$

(c) $\sigma = 0.5$ (d) $\sigma = 1.0$

Fig. 2. Results for SDE (6.2) with varying σ

Abb.	S_{acc}	S_{rej}	S_{acc}/S_{tried}	SW_{min}	SW_{max}	h_{fix}
(a)	21	5	0.808	0.095707	0.812528	0.307692
(b)	22	4	0.846	0.095708	0.758969	0.307692
(c)	24	3	0.889	0.095715	0.546010	0.296296
(d)	30	8	0.789	0.095741	0.517270	0.210526

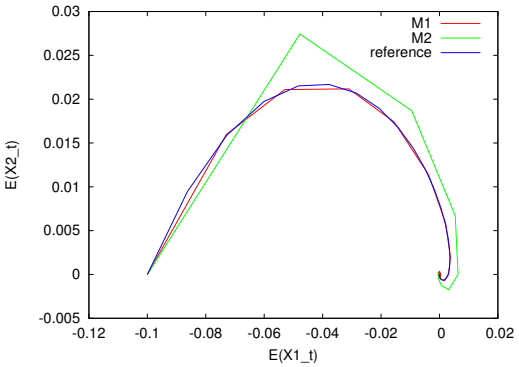
Table 2
Results of the step size control algorithm for SDE (6.2)

6.2 Test problem 2

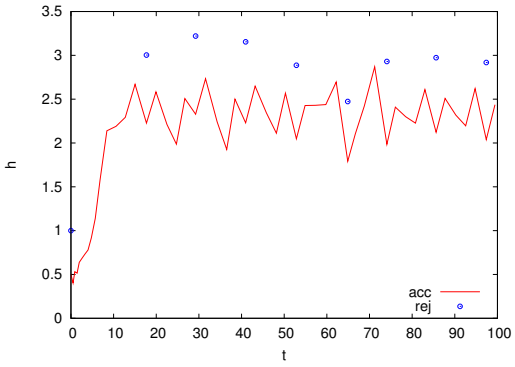
We consider the stochastic *Brusselator equation* [1,8] which is given by the following 2-dimensional system of Itô SDEs:

$$\begin{aligned} dX_t^1 &= ((\alpha-1)X_t^1 + \alpha(X_t^1)^2 + (X_t^1+1)^2 X_t^2)\ dt + \sigma X_t^1(1+X_t^1)\ dW_t \\ dX_t^2 &= (-\alpha X_t^1 - \alpha(X_t^1)^2 - (X_t^1+1)^2 X_t^2)\ dt - \sigma X_t^1(1+X_t^1)\ dW_t, \end{aligned} \tag{6.3}$$

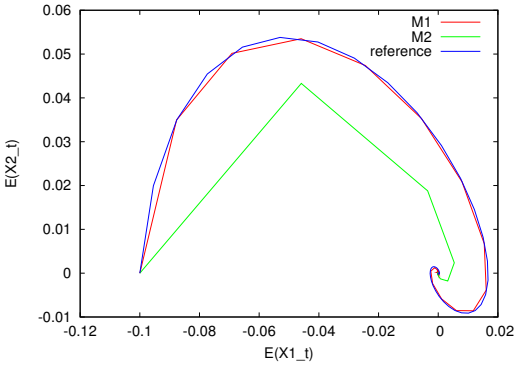
where again σ gives the intensity of the stochastic influence. In the deterministic case if $\alpha < 2$ the paths converge to the global asymptotically stable solution $(X^1, X^2) = (0,0)$ whereas for $\alpha > 2$ we obtain a boundary circle which encloses the now instable solution $(0,0)$ [1,6,8]. As there does not exist an analytical solution for equation (6.3) we again use a reference solution with very small step size for comparison. The functional f is the identity.



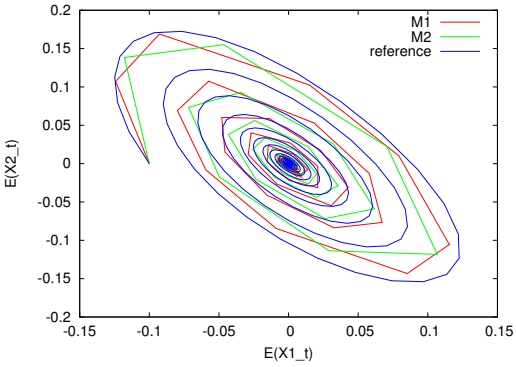
(a) $\alpha = 0.5$, $Atol_i = 0.001$, $Rtol_i = 0.01$, $i = 1,2$



(b) $\alpha = 0.5$, accepted and rejected steps



(c) $\alpha = 1.0$, $Atol_i = 0.001$, $Rtol_i = 0.01$, $i = 1,2$



(d) $\alpha = 1.9$, $Atol_i = 0.01$, $Rtol_i = 0.05$, $i = 1,2$

Fig. 3. Results for SDE (6.3) with varying α

Abb.	S_{acc}	S_{rej}	S_{acc}/S_{tried}	SW_{min}	SW_{max}	h_{fix}
(a)	51	9	0.850	0.402337	2.868974	1.666667
(c)	51	15	0.773	0.430555	3.806200	1.515152
(d)	74	26	0.740	0.573800	2.824429	1.000000

Table 3
Results of the step size control algorithm for SDE (6.3)

In Figure 3 we present the results of three simulations where we choose $\alpha = 0.5$, 1.0 and 1.9, respectively. Further parameters are $\sigma = 0.1$, the integration interval $I = [0, 100]$, the starting step size $h_{start} = 1.0$ and the initial value $X_0 = (-0.1, 0)^T$. We observe that method M1 yields a better approximation than M2 for every choice of α. In Figures 3(a) and 3(c) the approximation obtained by M2 is particularly imprecise at the beginning of the interval. This can be explained by analyzing the accepted and rejected step sizes, exemplarily pictured for $\alpha = 0.5$ in Figure 3(b). For times $t < 5$ one observes step sizes which are rather small compared to the fixed step size 1.666667 used by method M2. For $\alpha = 1.9$ we notice that the approximation path obtained by M2 reaches the origin much earlier than the path resulting from M1 and the path of the reference solution.

7 Conclusion

The object of the present paper was the development of a method with automatic step size control for the weak approximation of the solution of SDEs. We propose an algorithm which works with an embedded SRK method and thus causes a low additional computational effort for the step size control. Furthermore the error criterion of our step size algorithm is based on the expectation of the functional f applied to the approximation processes - a proceeding which is suitable, given that we are rather interested in expectations of functions of the solution process $E(f(X_t))$ than in the pathwise solution X_t itself. Numerical simulation of some test problems demonstrated the improvement in accuracy of the approximation yielded by our proposed method when compared to a method with fixed step size and an assimilable computational effort.

References

[1] Burrage, K. and Burrage, P.M.: A Variable Stepsize Implementation for Stochastic Differential Equations. SIAM, J. Sci. Comput., 24(3):848-864, 2002.

[2] Burrage, K. and Burrage, P.M.: Order conditions of stochastic Runge-Kutta methods by B-series. SIAM, J. Nume. Anal., 38(5):1626-1646, 2000.

[3] Butcher, J. C.: Numerical Methods for Ordinary Differential Equations. Chichester (England). John Wiley & Sons 2003.

[4] J. G. Gaines and T. J. Lyons: Variable step size control in the numerical solution of stochastic differential equations. SIAM J. Appl. Math., Vol. 57, No. 5:1455–1484, 1997.

[5] Gänssler, P. and Stute, W.: Wahrscheinlichkeitstheorie. Berlin, Heidelberg, New York. Springer Verlag 1977.

[6] Hairer, E. and Nørsett, S.P. and Wanner, G.: Solving Ordinary Differential Equations I. Berlin, Heidelberg. Springer Verlag 1993.

[7] Kloeden, P.E. and Platen, E.: Numerical Solution of Stochastic Differential Equations. Berlin, Heidelberg. Springer-Verlag 1999.

[8] Kloeden, P.E. and Platen, E. and Schurz, H.: Numerical Solution of SDE Through Computer Experiments. Berlin, Heidelberg, New York. Springer Verlag 1994.

[9] Komori, Y. and Mitsui, T. and Sugiura, H.: Rooted tree analysis of the order conditions of ROW-type scheme for stochastic differential equations. BIT, 37(1):43-66, 1997.

[10] Lamba, H.: An adaptive timestepping algorithm for stochastic differential equations. J. Comput. Appl. Math. 161(2):417-430, 2003.

[11] Mauthner, S.: Schrittweitensteuerung bei der numerischen Lösung stochastischer Differentialgleichungen. Düsseldorf. VDI Verlag GmbH, Reihe 10, Bd. 578, 1999.

[12] Rößler, A.: Rooted tree analysis for order conditions of stochastic Runge-Kutta methods for the weak approximation of stochastic differential equations. Preprint Nr. 2368, Darmstadt University of Technology, 2004 (submitted).

[13] Rößler, A.: Runge-Kutta methods for Itô stochastic differential equations with scalar noise. Preprint Nr. 2354, Darmstadt University of Technology, 2004 (submitted).

[14] Rößler, A.: Runge-Kutta Methods for the Numerical Solution of Stochastic Differential Equations. Ph.D. Thesis, Darmstadt University of Technology. Aachen. Shaker Verlag 2003.

[15] Rößler, A.: Stochastic Taylor Expansions for the Expectation of Functionals of Diffusion Processes. Stochastic Anal. Appl. 22(6):1553-1576, 2004.

[16] Szepessy, A. and Tempone, R. and Zouraris, G.: Adaptive weak approximation of stochastic differential equations. Commun. Pure Appl. Math. 54(10):1169-1214, 2001.

[17] Tocino, A. and Vigo-Aguiar, J.: Weak second order conditions for stochastic Runge-Kutta methods. SIAM J. Sci. Comput. 24(2):507-523, 2002.