Stability-Optimized High Order Methods and Stiffness Detection for Pathwise Stiff Stochastic Differential Equations

Chris Rackauckas
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, Massachusettss 02139
Email: crackauc@mit.edu

Qing Nie
Department of Mathematics
University of California, Irvine
Irvine, California, 92697
Email: qnie@uci.edu

***Abstract*—Stochastic differential equations (SDE) often exhibit large random transitions. This property, which we denote as pathwise stiffness, causes transient bursts of stiffness which limit the allowed step size for common fixed time step explicit and drift-implicit integrators. Here we present a HPC-driven method for deriving high strong order methods for stochastic differential equations. Utilizing GPU-accelerated global optimization, we numerically solve a constrained optimization problem which results stability-optimized adaptive methods of strong order 1.5 for SDEs. The resulting explicit methods are shown to exhibit substantially enlarged stability regions which allows for them to solve pathwise stiff biological models orders of magnitude more efficiently than previous methods like SRIW1 and Euler-Maruyama. These methods are benchmarked on a range of semi-stiff problems and demonstrate speedups between 6x previous adaptive algorithms while showing computationally infeasibility of fixed time step integrators on some of these test equations.**

1. Introduction

Stochastic differential equations (SDEs) are dynamic equations of the form

$$dX_t = f(t, X_t)dt + g(t, X_t)dW_t,$$

where X_t is a d-dimensional vector, $f : \mathbb{R}^d \to \mathbb{R}^d$ is the drift coefficient, and $g : \mathbb{R}^d \to \mathbb{R}^{d\times m}$ is the diffusion coefficient which describes the amount and mixtures of the noise process W_t which is a m-dimensional Brownian motion. SDEs are of interest because they can exhibit behaviors which are not found in deterministic models. For example, an ODE model of a chemical reaction network may stay at a constant steady state, but in the presence of randomness the trajectories may be switching between various steady states [1], [2], [3]. These types of models can capture crucial features such as transitions to cancerous states [4], discover rare cellular behaviors [5], and enhance personalized drug metabolism predictions for precision healthcare [6].

In many cases, these unique features of stochastic models are pathwise-dependent and are thus not a property of the evolution of the mean trajectory. However, these same effects cause random events of high numerical stiffness, which we denote as pathwise stiffness, which can cause difficulties for numerical integration methods.

A minimal example of pathwise stiffness is demonstrated in the equation

$$dX_t = \left[-1000X_t\left(1 - X_t\right)\left(2 - X_t\right)\right]dt + g(t, X_t)dW_t, \tag{1}$$

where $X_0 = 2$, $t \in [0, 5]$, and with additive noise $g(t, X_t) = 10$ where a sample trajectory is shown in Figure 1. This equation has two stable steady states, one at $X = 0$ and another at $X = 2$, which the solution switches between when the noise is sufficiently large. While near a steady state the derivative is approximately zero making the problem non-stiff, during these transitions the derivative of the drift term reaches a maximum of ≈ 400. This means that in order to be stable, explicit Stochastic Runge-Kutta (SRK) must have a small Δt. This display of large, transient, and random switching behavior in a given trajectory causes stochastic bursts of numerical stiffness, a phenomena which we will denote pathwise stiffness. The fixed time step Euler-Maruyama method would require $dt < 4 \times 10^{-3}$ to be stable for most trajectories, thus requiring greater than 2×10^4 steps to solve this 1-dimensional SDE. In many cases the switching behavior can be rare (due to smaller amounts of noise) or can happen finitely many times like in the multiplicative noise version with $g(t, X_t) = 10X_t$. Yet even if these switches are only a small portion of the total time, the stability requirement imposed by their existence determines the possible stepsizes and thus has a large contribution to the overall computational cost. While implicit methods can be used to increase the stability range, this can vastly increase the overall computational cost of each step, especially in the case large systems of SDEs like discretizations of stochastic reaction-diffusion equations. In addition, implicit solvers have in practice a smaller stability region due to requiring convergence of the quasi-Newton solvers for the implicit steps. This problem is mitigated in ODE software by high-quality stage predictors given by extrapolation algorithms for good initial conditions for the Newton steps [7]. However, there are no known algorithms for stage predictors in the presence of large noise bursts and

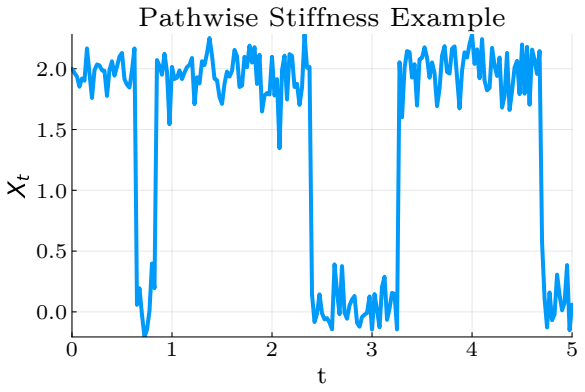

Figure 1. **Example of a Pathwise Stiff Solution**. Depicted is a sample trajectory of Equation 1 solved using the SOSRI methods developed in this manuscript with $reltol = abstol = 10^{-2}$.

thus we will demonstrate that classic implicit solvers have a form of instability. Additionally, previous literature questioned the applicability of L-stable integrators to stochastic differential equations due to high error in the slow variables [8]. Thus both fixed time step explicit and implicit solvers are inadequate for efficiently handling this common class of SDEs.

Since these features exist in the single trajectories of the random processes, methods which attempt to account for the presence of such bursts must do so on each individual trajectory in order to be efficient. In previous work, the authors have shown that by using adaptive time-stepping, a stochastic reaction network of 19 reactants is able to be solved with an average time step 100,000 times larger than the value that was found necessary for stability during the random stiff events for a high order SRK method [9]. This demonstrated that the key to solving these equations efficiently required controlling the time steps in a pathwise manner. However, the methods were still largely stability-bound, meaning the chosen tolerances to solve the model were determined by what was necessary for stability and was far below the error necessary for the application. The purpose of this investigation is to develop numerical methods with the ability to better handle pathwise stiffness and allow for efficient solving of large Monte Carlo experiments without resorting to drift-implicit methods.

We approach this problem by developing adaptive stability-optimized SRK methods with enlarged stability regions. This builds off of similar work for ODE integrators which optimize the coefficients of a Butcher tableau to give enhanced stability [10], [11], [12]. Similar to the Runge-Kutta Chebyschev methods [7] (and the S-ROCK extension to the stochastic case [13], [14], [15]), these methods are designed to be efficient for equations which display stiffness without fully committing to implicit solvers. Given the complexity of the stochastic stability equations and order conditions, we develop a novel and scalable mechanism for the derivation of "optimal" Runge-Kutta methods. We use this method to design stability-optimized methods for diagonal noise SDEs. We show through computational experiments that these adaptive stability-optimized SRK methods can adequately solve transiently stiff equations without losing efficiency in non-stiff problems. Together we test on semi-stiff equations with 2 to $6 \times 20 \times 100$ SDEs from biological literature and show a speedup around 6x over the previous adaptive SRIW1 algorithm, and demonstrate the infeasibility of common explicit and implicit methods (Euler-Maruyama, Runge-Kutta Milstein, Drift-Implicit Stochastic θ-Method, and Drift-Implicit θ Runge-Kutta Milstein) found as the basis of many SDE solver packages [16], [17], [18].

2. Adaptive Strong Order 1.0/1.5 SRK Methods for Diagonal Noise SDEs

The class of methods we wish to study are the adaptive strong order 1.5 SRK methods for diagonal noise [9], [19]. Diagonal noise is the case where the diffusion term g is diagonal matrix $\left(\sigma_i X_t^i\right)$ and includes phenomenological noise models like multiplicative and affine noise. The diagonal noise methods utilize the same general form and order conditions as the methods for scalar noise so we use their notation for simplicity. The strong order 1.5 methods for scalar noise are of the form

$$X_{n+1} = X_n + \sum_{i=1}^{s} \alpha_i f\left(t_n + c_i^{(0)} h, H_i^{(0)}\right) + \tag{2}$$

$$\sum_{i=1}^{s} \left(\beta_i^{(1)} I_{(1)} + \beta_i^{(2)} \frac{I_{(1,1)}}{\sqrt{h}} + \beta_i^{(3)} \frac{I_{(1,0)}}{h} + \beta_i^{(4)} \frac{I_{(1,1,1)}}{h}\right) \tag{3}$$

$$\times g\left(t_n + c_i^{(1)} h\right) \tag{4}$$

with stages

$$H_i^{(0)} = X_n + \sum_{j=1}^{s} A_{ij}^{(0)} f\left(t_n + c_j^{(0)} h, H_j^{(0)}\right) h + \tag{5}$$

$$\sum_{j=1}^{s} B_{ij}^{(0)} g\left(t_n + c_j^{(1)} h, H_j^{(1)}\right) \frac{I_{(1,0)}}{h} \tag{6}$$

$$H_i^{(1)} = X_n + \sum_{j=1}^{s} A_{ij}^{(1)} f\left(t_n + c_j^{(0)} h, H_j^{(0)}\right) h + \tag{7}$$

$$\sum_{j=1}^{s} B_{ij}^{(1)} g\left(t_n + c_j^{(1)} h, H_j^{(1)}\right) \sqrt{h}$$

where the I_j are the Wiktorsson approximations to the iterated stochastic integrals [20]. The tuple of coefficients $\left(A^{(j)}, B^{(j)}, \beta^{(j)}, \alpha\right)$ thus fully determines the SRK method. These coefficients must satisfy the constraint equations described in [9], [19] in order to receive strong order 1.5. These methods are appended with error estimates

$$E_D = \left|\Delta t \sum_{i \in I_1} (-1)^{\sigma(i)} f\left(t_n + c_i^{(0)} \Delta t, H_i^{(0)}\right)\right|$$

$$E_N = \left|\sum_{i \in I_2} \left(\beta_i^{(3)} \frac{I_{(1,0)}}{\Delta t} + \beta_i^{(4)} \frac{I_{(1,1,1)}}{\Delta t}\right) g\left(t_n + c_i^{(1)} \Delta t, H_i^{(1)}\right)\right|$$

and the rejection sampling with memory (RSwM) algorithm to give it fully adaptive time-stepping [9]. Thus unlike in the theory of ordinary differential equations [21], [22], [23], [24], [25], the choice of coefficients for SRK methods does not require explicitly finding an embedded method when developing an adaptive SRK method and we will therefore take for granted that each of the derived methods is adaptive.

3. Optimized-Stability Order 1.5 SRK Methods with Diagonal Noise

3.1. The Stability Equation for Order 1.5 SRK Methods with Diagonal Noise

For diagonal noise, we use the mean-square definition of stability [26]. A method is mean-square stable if $\lim_{n\to\infty} \mathbb{E}\left(|X_n|^2\right) = 0$ on the test equation

$$dX_t = \mu X_t dt + \sigma X_t dW_t. \tag{8}$$

In matrix form we can re-write our method as given by

$$X_{n+1} = X_n + \mu h\left(\alpha \cdot H^{(0)}\right) + \sigma I_{(1)}\left(\beta^{(1)} \cdot H^{(1)}\right) \tag{9}$$

$$+ \sigma \frac{I_{(1,1)}}{\sqrt{h}}\left(\beta^{(2)} \cdot H^{(1)}\right) + \sigma \frac{I_{(1,0)}}{h}\left(\beta^{(3)} \cdot H^{(1)}\right) \tag{10}$$

$$+ \sigma \frac{I_{(1,1,1)}}{h}\left(\beta^{(4)} \cdot H^{(1)}\right) \tag{11}$$

with stages

$$\begin{aligned} H^{(0)} &= X_n + \mu\Delta t A^{(0)} H^{(0)} + \sigma \frac{I_{(1,0)}}{h} B^{(0)} H^{(1)}, \\ H^{(1)} &= X_n + \mu\Delta t A^{(1)} H^{(0)} + \sigma\sqrt{\Delta t} B^{(1)} H^{(1)} \end{aligned} \tag{12}$$

where $\hat{X_n}$ is the size s constant vector of X_n. Let $z = \mu h$ and $w = \sigma\sqrt{h}$ and isolate $S = E\left[\frac{U_{n+1}^2}{U_n^2}\right]$. In this space, z is the stability variable for the drift term and w is the stability in the diffusion term. Under this scaling $\left(h, \sqrt{h}\right)$, the equation becomes independent of h and thus becomes a function $S(z, w)$ on the coefficients of the SRK method where mean-square stability is achieved when $|S(z, w)| < 1$.

3.2. An Optimization Problem for Determination of Coefficients

We wish to determine the coefficients for the diagonal SRK methods which optimize the stability. To do so, we generate an optimization problem which we can numerically solve for the coefficients. To simplify the problem, we let $z, w \in \mathbb{R}$. Define the function

$$f(z, w; N, M) = \int_{-M}^{M}\int_{-N}^{1} \chi_{S(z,w)\leq 1}(z, w) dz dw. \tag{13}$$

Notice that for $N, M \to \infty$, f is the area of the stability region. Thus we define the stability-optimized diagonal SRK method as the set of coefficients which achieves

$$\max_{A^{(i)}, B^{(i)}, \beta^{(i)}, \alpha} f(z, w) \tag{14}$$

subject to: Order Constraints

However, we impose a few extra constraints to add robustness to the error estimator. In all cases we impose $0 < c_i^{(0)}, c_i^{(1)} < 1$. Additionally we can prescribe $c_4^{(0)} = c_4^{(1)} = 1$ which we call the End-C Constraint. Lastly, we can prescribe the ordering constraint $c_1^{(j)} < c_2^{(j)} < c_3^{(j)} < c_4^{(j)}$ which we denote as the Inequality-C Constraint.

The resulting problem is a nonlinear programming problem with 44 variables and 42-48 constraint equations. The objective function is the two-dimensional integral of a discontinuous function which is determined by a polynomial of in z and w with approximately 3 million coefficients. To numerically approximate this function, we calculated the characteristic function on a grid with even spacing dx using a CUDA kernel and found numerical solutions to the optimization problem using the JuMP framework [27] with the NLopt backend [28]. A mixed approach using many solutions of the semi-local optimizer LN_AUGLAG_EQ [29], [30] and fewer solutions from the global optimizer GN_ISRES [31] were used to approximate the optimality of solutions. The optimization was run many times in parallel until many results produced methods with similar optimality, indicating that we likely obtained values near the true minimum.

The parameters N and M are the bounds on the stability region and also represent a trade-off between the stability in the drift and the stability in the diffusion. A method which is optimized when M is small would be highly stable in the case of small noise, but would not be guaranteed to have good stability properties in the presence of large noise. Thus these parameters are knobs for tuning the algorithms for specific situations, and thus we solved the problem for different combinations of N and M to determine different algorithms for the different cases.

3.3. Resulting Approximately-Optimal Methods

The coefficients generated for approximately-optimal methods fall into three categories. In one category we have the drift-dominated stability methods where large N and small M was optimized. On the other end we have the diffusion-dominated stability methods where large M and small N was optimized. Then we have the mixed stability methods which used some mixed size choices for N and M. As a baseline, we optimized the objective without constraints on the c_i to see what the "best possible method" would be. When this was done with large N and M, the resulting method, which we name SOSRI, has almost every value of c satisfy the constraints, but with $c_2^{(0)} \approx -0.04$ and $c_4^{(0)} \approx 3.75$. To see if we could produce methods which were

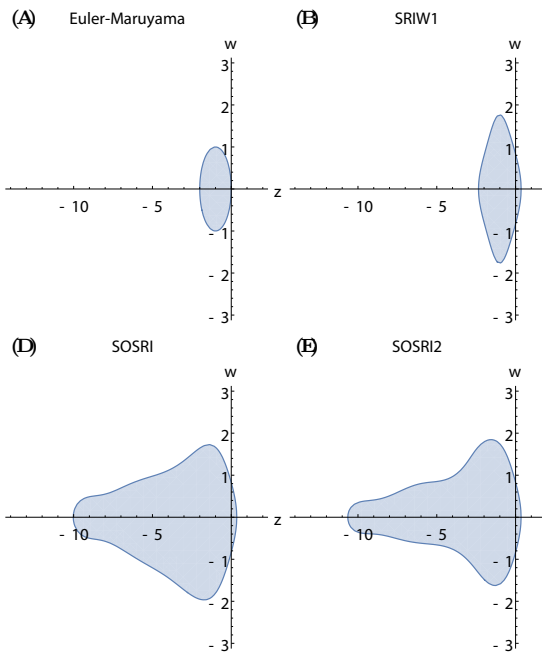

Figure 2. **SOSRI Stability Regions**. The stability regions $(S(z, w) \leq 1)$for the previous and SOSRI methods are plotted in the (z, w)-plane. **(A)** Euler-Maruyama. **(B)** SRIW1. **(C)** SRIW2. **(D)** SOSRI. **(E)** SOSRI2

more diffusion-stable, we decreased N to optimize more in w but failed to produce methods with substantially enlarged diffusion-stability over SOSRI.

Adding only the inequality constraints on the c_i and looking for methods for drift-dominated stability, we failed to produce methods whose c_i estimators adequately covered the interval. Some of the results did produce stability regions similar to SOSRI but with $c_i^{(0)} < 0.5$ which indicates the method could have problems with error estimation. When placing the equality constraints on the edge c_i, one method, which we label SOSRI2, resulted in similar stability to SOSRI but satisfy the c_i constraints. In addition, this method satisfies $c_3^{(0)} = c_4^{(0)} = 1$ and $c_3^{(1)} = c_4^{(1)} = 1$. The stability regions for these methods is shown in Figure 2.

To look for more diffusion-stable methods, we dropped to $N = 6$ to encourage the methods to expand the stability in the w-plane. However, we could not find a method whose stability region went substantially beyond $[-2, 2]$ in w. This was further decreased to $N = 1$ where methods still could not go substantially beyond $|2|$. Thus we were not able to obtain methods optimized for the diffusion-dominated case. This hard barrier was hit under many different constraint and objective setups and under thousands of optimization runs, indicating there might be a diffusion-stability barrier for explicit methods.

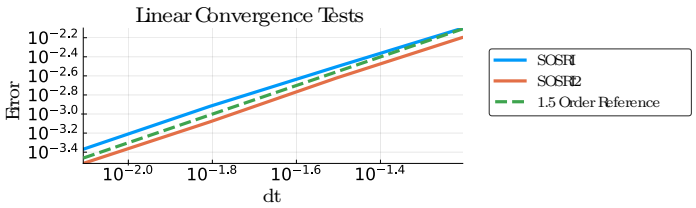

Figure 3. **Convergence results on Equation 15.** The error is averaged over 1000 trajectories. Shown are the strong l_2 error along the time series of the solution. The test used a fixed time step $h = 1/2^{-4}$ to $h = 1/2^{-7}$.

4. Numerical Results

4.1. Convergence Tests

In order to test the efficiency and correctness of the SRI algorithms, we used the linear test Appendix Equation 15. Figure 3B demonstrates that the SOSRI methods achieve the strong order 1.5 on Equation 15.

4.2. Epithelial-Mesenchymal Transition (EMT) Model (20 Pathwise Stiff SDEs)

To test the real consequences of the enhanced stability, we use the Epithelial-Mesenchymal Transition (EMT) model of 20 pathwise stiff reaction equations introduced in [3], studied as a numerical test in [9], and written in Appendix Section B. In the previous work it was noted that $t \in [0, 1]$ was a less stiff version of this model. Thus we first tested the speed that the methods could solve for 10,000 trajectories with no failures due to numerical instabilities. The tolerances were tuned for each method by factors of 2 and finding the largest values that were stable. Since SOSRI demonstrated that its stability is much higher than even SOSRI2, we show the effect of tolerance changes on SOSRI as well. The results show that at similar tolerances the SOSRI method takes nearly 5x less time than SRIW1 (Table 1). However, there is an upper bound on the tolerances before the adaptivity is no longer able to help keep the method stable. For SRIW1, this bound is much lower, causing it to run more than 15x slower than the fastest SOSRI setup. Interestingly SOSRI2 required a higher tolerance than SRIW1 but was 3x faster than SRIW1's fastest setup. We note that SOSRI's highest relative tolerance $2^{-7} \approx 7 \times 10^{-3}$ is essentially requiring 4 digits of accuracy (in strong error) when considering the conservativeness of the error estimator, which is far beyond the accuracy necessary in many cases. Lastly, we note that the SOSRI method is able to solve for 10,000 stable trajectories more than 60x faster than any of the tested fixed time step methods.

We then timed the run time to solve 10 trajectories in the $t \in [0, 500]$ case (Table 2). This time we found the optimal tolerance in terms of powers of 10. Once again,

Algorithm	Abstol	Reltol	Run-time (seconds)	Relative Time (vs SOSRI)
SOSRI	2^{-7}	2^{-4}	2.62	1.0x
SOSRI	2^{-7}	2^{-6}	2.75	1.0x
SOSRI	2^{-12}	2^{-15}	8.78	3.3x
SOSRI	2^{-13}	2^{-7}	3.05	1.2x
SOSRI2	2^{-12}	2^{-15}	8.69	3.3x
SOSRI2	2^{-13}	2^{-11}	5.56	2.2x
SRIW1	2^{-13}	2^{-7}	15.16	5.8x
Euler-Maruyama			169.96	64.8x
Runge-Kutta Milstein			182.59	69.6x
Fixed Time-step SRIW1			424.30	161.7x
DISTM $\left(\theta = \frac{1}{2}\right)$			8912.91	3396x

TABLE 1. **SRI TIMES FOR THE THE EMT MODEL ON $t \in [0, 1]$.** THE EQUATIONS WERE SOLVED 10,000 TIMES WITH THE GIVEN TOLERANCES TO COMPLETION AND THE ELAPSED TIME WAS RECORDED. THE FIXED TIME STEP METHODS HAD THEIR Δt DETERMINED AS THE LARGEST Δt IN INCREMENTS OF POWERS OF 2 THAT PRODUCED NO UNSTABLE TRAJECTORIES, AS SHOWN IN [9]. DISTM IS THE DRIFT-IMPLICIT STOCHASTIC θ-METHOD

Algorithm	Abstol	Reltol	Run-time (seconds)	Relative Time (vs SOSRI)
SOSRI	10^{-2}	10^{-2}	22.47	1.0x
SOSRI	10^{-4}	10^{-4}	73.62	3.3x
SOSRI	10^{-5}	10^{-3}	89.19	4.0x
SOSRI2	10^{-4}	10^{-4}	76.12	3.4x
SOSRI2	10^{-5}	10^{-3}	121.75	5.4x
SRIW1	10^{-5}	10^{-3}	147.89	6.6x
DIRKM $\left(\theta = \frac{1}{2}\right)$			7378.55	328.3x
DIEM $\left(\theta = \frac{1}{2}\right)$			8796.47	391.4x

TABLE 2. **SRI TIMES FOR THE THE EMT MODEL ON $t \in [0, 500]$.** THE EQUATIONS WERE SOLVED 10 TIMES WITH THE GIVEN TOLERANCES TO COMPLETION AND THE ELAPSED TIME WAS RECORDED. THE FIXED TIMESTEP METHODS HAD THEIR Δt CHOSEN BY INCREMENTING BY 10^{-5} UNTIL 10 CONSECUTIVE TRAJECTORIES WERE STABLE. DRIFT-IMPLICIT EULER MARUYAMA (DIEM) HAD $\Delta t = \frac{1}{60000}$ AND DRIFT-IMPLICIT RUNGE-KUTTA MILSTEIN (DIRKM) HAD $\Delta t = \frac{1}{50000}$.

SRIW1 needed a lower tolerance than is necessary in order to stay stable. SOSRI is able to solve the problem only asking for around $tol = 10^{-2}$, while the others require more (especially in absolute tolerance as there is a stiff reactant whose values travel close to zero). One interesting point to note is that at similar tolerances both SOSRI and SOSRI2 receive similar timings and both over 6 times faster than the fastest SRIW1 tolerance setup. Both are nearly twice as fast as SRIW1 when matching tolerances as well. Given the conservativeness of the error estimators generally being around 2 orders of magnitude more precise than the local error estimate, the low tolerance solutions are accurate enough for many phenomenological experiments and thus present a good speedup over previous methods. The timings for Euler-Maruyama and Runge-Kutta Milstein schemes are omitted since the tests were unable to finish. From the results of [9] we note that the average dt for SRIW1 on the edge of its stability had that the smallest dt was approximately 10^{-11}. The stability region for fixed step-size Euler-Maruyama is strictly smaller than SRIW1 (Figure 2) which suggests that it would require around 5×10^{12} time steps (with Runge-Kutta Milstein being similar) to solve to $t = 500$. Thus, given it takes on our setup extrapolating the time given 170 seconds for 2^{20} steps, this projects to around 1.6×10^8 seconds, or approximately 5 years.

Algorithm	Abstol	Reltol	Run-time (seconds)	Relative Time (vs SOSRI)
SOSRI	10^{-1}	10^{-2}	700.76	1.0x
SOSRI2	10^{-3}	10^{-3}	1016.61	1.5x
Euler-Maruyama			1758.85	2.5x
SRIW1	10^{-5}	10^{-3}	4205.52	6.0x

TABLE 3. **SRI TIMES FOR THE THE RETINOIC ACID SPDE MODEL ON $t \in [0, 500]$.** THE EQUATIONS WERE SOLVED TWICE WITH THE GIVEN TOLERANCES TO COMPLETION AND THE ELAPSED TIME WAS RECORDED. THE TOLERANCES WERE CHOSEN AS THE HIGHEST PAIR OF TOLERANCES WHICH DID NOT DIVERGE (GOING UP BY POWERS OF 10). NOTE THAT NONE OF THE CASES DID THE TWO TIMINGS VARY BY MORE THAN 1% OF THE TOTAL RUN TIME. EULER-MARUYAMA USED TIME STEPS OF $\Delta t = 1/20000$ SINCE WE NOTE THAT AT $\Delta t = 1/10000$ APPROXIMATELY HALF OF THE TRAJECTORIES (SIMULATING 10) WERE UNSTABLE.

4.3. Retinoic Acid Stochastic Partial Differential Equation Model (6x20x100 Semi-Stiff SDEs)

As another test we applied the methods to a method of lines discretization of a stochastic partial differential equation (SPDE) describing the spatial regulation of the Zrafish hindbrain via retinoic acid signaling (Section C) [1]. The discretization results in a system of $6 \times 20 \times 100$ SDEs. Starting from an initial zero state, a concentration gradient emerges over $t \in [0, 500]$. Each of the methods solved the problem at the highest tolerance that was stable giving the results in Table 3. Time stepping for this problem is heavily limited by the high diffusion constant which results in a strict CFL condition for the 2nd order finite difference discretization that is used (in the PDE sense), making this problem's stepping stability-bound for explicit methods. Because of this stiffness in the real axis, we found that the previous high order adaptive method SRIW1 did not perform well on this problem in comparison to Euler-Maruyama because the drift term is expensive and the extra function calls outweighed the slightly larger timesteps. However, the enhanced stability of the SOSRI and SOSRI2 methods allowed for much larger time steps while keeping the same number of f calls per step, resulting in a more efficient solution when high accuracy is not necessary. We note that the drift-implicit stochastic θ-method and drift implicit θ Runge-Kutta Milstein methods were too inefficient to estimate since their time steps were constrained to be near that of the Euler-Maruyama equation due to divergence of the Newton iterations.

5. Discussion

In this work we derived stability-optimized SRK methods for diagonal noise equations. Importantly, our derivation methods utilized heavy computational tools in order to approximately optimize otherwise intractable equations. This same method of derivation can easily be scaled up to higher orders, and by incorporating the coefficients for higher conditions, efficiency can be optimized as well by adding the norm of the principle error coefficients to the optimization function. The majority of the search was performed using global optimizers in massive parallel using a

hand-optimized CUDA kernel for the numerical integral of the characteristic function, replacing man-hours with core-hours and effectively optimizing the method. The clear next steps are to find SRI methods with minimal error estimates and sensible stability regions for the cases in which lower strong error matters, and similar optimizations on SRK methods developed for small noise problems. We note that high strong order methods were investigated because of their better trajectory-wise convergence, allowing for a more robust solution and error estimation since our application to transiently pathwise stiff equations requires such properties.

The main caveat for our methods is the restrictions on the form of noise. Further research should focus on the expansion of this these techniques to high order non-diagonal noise integrators. In addition, when g is non-zero a "diagonal noise" problem over the complex plane does not have diagonal noise (due to the mixing of real and complex parts from complex multiplication, and reinterpretation as a $2n$ real system). Thus these methods are not applicable to problems defined in the complex plane with complex Wiener processes. Development of similar integrators for commutative noise problems could allow for similar performance benefits on such problems and is a topic for future research.

Our timings show that the current high order SRK methods are stability-bound and that when scientific studies are only looking for small amounts of accuracy in stochastic simulations, most of the computational effort is lost to generating more accurate than necessary solutions in order to satisfy stability constraints. For diagonal noise approximately 6x faster than the current adaptive methods (SRIW1), while common methods like Euler-Maruyama and Drift-Implicit θ Runge-Kutta Milstein were in many cases hundreds of times slower or in many cases could not even finish. We have also shown that these methods are very robust even at high tolerances and have a tendency to produce the correct qualitative results on semi-stiff equations (via plots) even when the user chosen accuracy is low. Given that the required user input is minimal and work over a large range of stiffness, we see these as very strong candidates for default general purpose solvers for problem-solving environments such as MATLAB and Julia since they can easily and efficiently produce results which are sufficiently correct.

Appendix

1. Diagonal Noise Test Equation

$$dX_t = \alpha X_t dt + \beta X_t dW_t \;\; X_0 = \frac{1}{2}, \tag{15}$$

where $\alpha = \frac{1}{10}$ and $\beta = \frac{1}{20}$ with true solution

$$X_t = X_0 e^{\left(\beta - \frac{\alpha^2}{2}\right)t + \alpha W_t}. \tag{16}$$

2. Epithelial-Mesenchymal Transition Model

The Epithelial-Mesenchymal Transition (EMT) model is given by the following system of SDEs which correspond to a chemical reaction network modeled via mass-action kinetics with Hill functions for the feedbacks. This model was introduced in [3].

$$\begin{aligned}
A &= \left(\left([TGF] + [TGF0]\left(t\right)\right)/J0_S\right)^{n0_S} + \left([OVOL2]/J1_S\right)^{n1_S} \\
\frac{d\left[S1\right]_t}{dt} &= k0_S + k_S \frac{\left(\left([TGF] + [TGF0]\left(t\right)\right)/J0_S\right)^{n0_S}}{\left(1+A\right)\left(1+\left[S\right]/J2_S\right)} \\
&- kd_S\left(\left[S1\right] - \left[SR\right]\right) - kd_{SR}\left[SR\right] \\
\frac{d\left[S\right]}{dt} &= k_S\left(\left[S1\right] - \left[SR\right]\right) - kd_S\left[S\right] \\
\frac{d\left[miR34\right]}{dt} &= kO_{34} + \frac{k_{34}}{1 + \left(\left[S\right]/J1_{34}\right)^{n1_{34}} + \left(\left[Z\right]/J2_{34}\right)^{n2_{34}}} \\
&- kd_{34}\left(\left[miR34\right] - \left[SR\right]\right) - \left(1-\lambda_{SR}\right)kd_{SR}\left[SR\right] \\
\frac{d\left[SR\right]}{dt} &= Tk\left(K_{SR}\left(\left[S1\right] - \left[SR\right]\right)\left(\left[miR34\right] - \left[SR\right]\right) - \left[SR\right]\right) \\
\frac{d\left[Z\right]}{dt} &= k0_Z + k_Z \frac{\left(\left[S\right]/J1_Z\right)^{n1_Z}}{1 + \left(\left[S\right]/J1_Z\right)^{n1_Z} + \left(\left[OVOL2\right]/J2_Z\right)^{n2_Z}} \\
&- kd_Z\left(\left[Z\right] - \sum_{i=1}^{5} C_5^i\left[ZR\right]\right) - \sum_{i=1}^{5} kd_{ZR_i} C_5^i\left[ZR_i\right] \\
\frac{d\left[Z\right]}{dt} &= k_Z\left(\left[Z\right] - \sum_{i=1}^{5} C_5^i\left[ZR_i\right]\right) - kd_Z\left[Z\right] \\
\frac{d\left[mir\right]}{dt} &= k0_{200} + \frac{k_{200}}{1 + \left(\left[S\right]/J1_{200}\right)^{n1_{200}} + \left(\left[Z\right]/J2_{200}\right)^{n2_{200}}} \\
&- kd_{200}\left(\left[mir\right] - \sum_{i=1}^{5} iC_5^i\left[ZR_i\right] - \left[TR\right]\right) \\
&- \sum_{i=1}^{5}\left(1-\lambda_i\right)kd_{ZR_i}C_5^i i\left[ZR_i\right] - \left(1-\lambda_{TR}\right)kd_{TR}\left[TR\right] \\
\frac{d\left[ZR_1\right]}{dt} &= Tk\left(K_1\left(\left[mir\right] - \sum_{i=1}^{5} iC_5^i\left[ZR_i\right] - \left[TR\right]\right)\right. \\
&\quad \left.\left(\left[Z\right] - \sum_{i=1}^{5} C_5^i\left[ZR_i\right]\right) - \left[ZR_1\right]\right) \\
\frac{d\left[ZR_2\right]}{dt} &= Tk\left(K_2\left(\left[mir\right] - \sum_{i=1}^{5} iC_5^i\left[ZR_i\right] - \left[TR\right]\right)\left[ZR_1\right] - \left[ZR_2\right]\right) \\
\frac{d\left[ZR_3\right]}{dt} &= Tk\left(K_3\left(\left[mir\right] - \sum_{i=1}^{5} iC_5^i\left[ZR_i\right] - \left[TR\right]\right)\left[ZR_1\right] - \left[ZR_3\right]\right) \\
\frac{d\left[ZR_4\right]}{dt} &= Tk\left(K_4\left(\left[mir\right] - \sum_{i=1}^{5} iC_5^i\left[ZR_i\right] - \left[TR\right]\right)\left[ZR_1\right] - \left[ZR_4\right]\right) \\
\frac{d\left[ZR_5\right]}{dt} &= Tk\left(K_5\left(\left[mir\right] - \sum_{i=1}^{5} iC_5^i\left[ZR_i\right] - \left[TR\right]\right)\left[ZR_1\right] - \left[ZR_5\right]\right) \\
\frac{d\left[tgf\right]}{dt} &= k_{tgf} - kd_{tgf}\left(\left[tgf\right] - \left[TR\right]\right) - kd_{TR}\left[TR\right] \\
\frac{d\left[TGF\right]}{dt} &= k0_{TGF} + k_{TGF}\left(\left[tgf\right] - \left[TR\right]\right) - kd_{TGF}\left[TGF\right] \\
\frac{d\left[TR\right]}{dt} &= Tk\left(K_{TR}\left(\left[mir\right] - \sum_{i=1}^{5} iC_5^i\left[ZR_i\right] - \left[TR\right]\right)\left(\left[tgf\right] - \left[TR\right]\right) - \left[TR\right]\right) \\
\frac{d\left[Ecad\right]}{dt} &= k0_E + \frac{k_{E1}}{1 + \left(\left[S\right]/J1_E\right)^{n1_E}} + \frac{k_{E2}}{1 + \left(\left[Z\right]/J2_E\right)^{n2_E}} - kd_E\left[Ecad\right] \\
B &= k_{V1}\frac{\left(\left[S\right]/J1_V\right)^{n1_V}}{1 + \left(\left[S\right]/J1_V\right)^{n1_V}} + k_{V2}\frac{\left(\left[Z\right]/J2_V\right)^{n2_V}}{1 + \left(\left[Z\right]/J2_V\right)^{n2_V}} \\
\frac{d\left[Vim\right]}{dt} &= k0_V + \frac{B}{\left(1 + \left[OVOL2\right]/J3_V\right)} - kd_V\left[Vim\right] \\
\frac{d\left[OVOL2\right]}{dt} &= k0_0 + k_0\frac{1}{1 + \left(\left[Z\right]/J_0\right)^{n0}} - kd_O\left[OVOL2\right] \\
\frac{d\left[OVOL2\right]_p}{dt} &= k_{Op}\left[OVOL2\right] - kd_{Op}\left[OVOL2\right]_p
\end{aligned}$$

where

$$\begin{aligned}
\sum_{i=1}^{5} iC_5^i\left[ZR_i\right] &= 5\left[ZR_1\right] + 20\left[ZR_2\right] + +30\left[ZR_3\right] + 20\left[ZR_4\right] + 5\left[ZR_5\right], \\
\sum_{i=1}^{5} C_5^i\left[ZR_i\right] &= 5\left[ZR_1\right] + 10\left[ZR_2\right] + 10\left[ZR_3\right] + 5\left[ZR_4\right] + \left[ZR_5\right], \\
\left[TGF0\right]\left(t\right) &= \begin{cases} \frac{1}{2} & t > 100 \\ 0 & o.w. \end{cases}
\end{aligned}$$

The parameter values are given in Table 4.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$J1_{200}$	3	$J1_E$	0.1	K_2	1	$k0_O$	0.35
$J2_{200}$	0.2	$J2_E$	0.3	K_3	1	kO_{200}	0.0002
$J1_{34}$	0.15	$J1_V$	0.4	K_4	1	kO_{34}	0.001
$J2_{34}$	0.35	$J2_V$	0.4	K_5	1	kd_S	0.09
J_O	0.9	$J3_V$	2	K_{TR}	20	kd_{tgf}	0.1
$J0_S$	0.6	$J1_Z$	3.5	K_{SR}	100	kd_Z	0.1
$J1_S$	0.5	$J2_Z$	0.9	$TGF0$	0	kd_{TGF}	0.9
$J2_S$	1.8	K_1	1	Tk	1000	kd_Z	1.66
$k0_S$	0.0005	$k0_Z$	0.003	λ_1	0.5	$k0_{TGF}$	1.1
$n1_{200}$	3	$n1_S$	2	λ_2	0.5	$k0_E$	5
$n2_{200}$	2	$n1_E$	2	λ_3	0.5	$k0_V$	5
$n1_{34}$	2	$n2_E$	2	λ_4	0.5	k_{E1}	15
$n2_{34}$	2	$n1_V$	2	λ_5	0.5	k_{E2}	5
n_O	2	$n2_V$	2	λ_{SR}	0.5	k_{V1}	2
$n0_S$	2	$n2_Z$	6	λ_{TR}	0.5	k_{V2}	5
k_O	1.2	k_{200}	0.02	k_{34}	0.01	k_{tgf}	0.05
k_Z	0.06	k_{TGF}	1.5	k_S	16	k_Z	16
kd_{ZR_1}	0.5	kd_{ZR_2}	0.5	kd_{ZR_3}	0.5	kd_{ZR_4}	0.5
kd_{ZR_5}	0.5	kd_O	1.0	kd_{200}	0.035	kd_{34}	0.035
kd_{SR}	0.9	kd_E	0.05	kd_V	0.05	k_{Op}	10
						kd_{Op}	10

TABLE 4. TABLE OF PARAMETER VALUES FOR THE EMT MODEL.

Parameter	Value	Parameter	Value	Parameter	Value
$\sigma_{RA_{in}}$,σ_{DNA},$\sigma_{RA_{out}}$	0.1	ω	100	u	0.01
b	0.17	γ	3.0	d	0.1
α	10000	δ	0.0013	e	1
β_0	1	η	0.0001	a	1
c	0.1	r	0.0001	ζ	0.02
ν	0.85	λ	0.85	D	250.46

TABLE 5. TABLE OF PARAMETER VALUES FOR THE RA SPDE MODEL.

3. Retinoic Acid SPDE Model

$$
\begin{aligned}
d\left[RA_{out}\right] &= \left(\beta(x) + D\Delta\left[RA_{out}\right] - b\left[RA_{out}\right] + c\left[RA_{in}\right]\right)dt + \sigma_{RA_{out}} dW_t^{out} \\
d\left[RA_{in}\right] &= \left(b\left[RA_{out}\right] + \delta\left[BP\right]\left[DNA\right] - \left(\gamma\left[BP\right] + \eta + \frac{\alpha\left[DNA\right]}{\omega + \left[DNA\right]} - c\right)\left[RA_{in}\right]\right)dt \\
d\left[RABP\right] &= \left(\gamma\left[BP\right]\left[RA_{in}\right] + \lambda\left[BP\right]\left[DNA\right] - \left(\delta + \nu\left[RAR\right]\right)\left[RABP\right]\right)dt \\
d\left[DNA\right] &= \left(\nu\left[RABP\right]\left[RAR\right] - \lambda\left[BP\right]\left[DNA\right]\right)dt + \sigma_{DNA}\left[DNA\right] dW_t^{DNA} \\
d\left[BP\right] &= \left(a - \lambda\left[BP\right]\left[DNA\right] - \gamma\left[BP\right]\left[RA_{in}\right] + \left(\delta + \nu\left[RAR\right]\right)\left[RABP\right]\right)dt \\
&\quad + \left(-u\left[BP\right] + \frac{d\left[DNA\right]}{e + \left[DNA\right]}\right)dt \\
d\left[RAR\right] &= \left(\zeta - \nu\left[RABP\right]\left[RAR\right] + \lambda\left[BP\right]\left[DNA\right] - r\left[RAR\right]\right)dt
\end{aligned}
$$

where $\beta(x) = \beta_0 H(x - 40)$ with H the Heaviside step function and $x = 40$ is the edge of retinoic acid production [1]. The space was chosen as $[-100, 400] \times [0, 100]$ with $\Delta x = \Delta y = 5$. The boundary conditions were no-flex on every side except the right side which had leaky boundary conditions with parameter $kA = 0.002$, though full no-flux does not noticeably change the results. The parameter values are given in Table 5.

All entries not listed are zero.

4. SOSRI

Coefficient	Value	Coefficient	Value
$A^{(0)}_{2,1}$	-0.04199224421316468	α_3	0.4736296532772559
$A^{(0)}_{3,1}$	2.842612915017106	α_4	0.026404498125060714
$A^{(0)}_{3,2}$	-2.0527723684000727	$c^{(0)}_2$	-0.04199224421316468
$A^{(0)}_{4,1}$	4.338237071435815	$c^{(0)}_3$	0.7898405466170333
$A^{(0)}_{4,2}$	-2.88959361374397 93	$c^{(0)}_4$	3.7504010171562823
$A^{(0)}_{4,3}$	2.3017575594644466	$c^{(1)}_1$	0
$A^{(1)}_{2,1}$	0.26204282091330466	$c^{(1)}_2$	0.26204282091330466
$A^{(1)}_{3,1}$	0.20903646383505375	$c^{(1)}_3$	0.05879875232001766
$A^{(1)}_{3,2}$	-0.1502377115150361	$c^{(1)}_4$	0.758661169101175
$A^{(1)}_{4,1}$	0.05836595312746999	$\beta^{(1)}_1$	-1.8453464565104432
$A^{(1)}_{4,2}$	0.6149440396332373	$\beta^{(1)}_2$	2.688764531100726
$A^{(1)}_{4,3}$	0.08535117634046772	$\beta^{(1)}_3$	-0.2523866501071323
$B^{(0)}_{2,1}$	-0.21641093549612528	$\beta^{(1)}_4$	0.40896857551684956
$B^{(0)}_{3,1}$	1.5336352863679572	$\beta^{(2)}_1$	0.4969658141589478
$B^{(0)}_{3,2}$	0.26066223492647056	$\beta^{(2)}_2$	-0.5771202869753592
$B^{(0)}_{4,1}$	-1.0536037558179159	$\beta^{(2)}_3$	-0.12919702470322217
$B^{(0)}_{4,2}$	1.7015284721089472	$\beta^{(2)}_4$	0.2093514975196336
$B^{(0)}_{4,3}$	-0.20725685784180017	$\beta^{(3)}_1$	2.8453464565104425
$B^{(1)}_{2,1}$	-0.5119011827621657	$\beta^{(3)}_2$	-2.688764531100725
$B^{(1)}_{3,1}$	2.67767339866713	$\beta^{(3)}_3$	0.2523866501071322
$B^{(1)}_{3,2}$	-4.9395031322250995	$\beta^{(3)}_4$	-0.40896857551684945
$B^{(1)}_{4,1}$	0.15580956238299215	$\beta^{(4)}_1$	0.11522663875443433
$B^{(1)}_{4,2}$	3.2361551006624674	$\beta^{(4)}_2$	-0.57877086147738
$B^{(1)}_{4,3}$	-1.4223118283355949	$\beta^{(4)}_3$	0.2857851028163886
α_1	1.140099274172029	$\beta^{(4)}_4$	0.177759119906557 04
α_2	-0.6401334255743456		

5. SOSRI2

Coefficient	Value	Coefficient	Value
$A^{(0)}_{2,1}$	0.13804532298278663	α_3	0.686995463807979
$A^{(0)}_{3,1}$	0.5818361298250374	α_4	-0.2911544680711602
$A^{(0)}_{3,2}$	0.4181638701749618	$c^{(0)}_2$	0.13804532298278663
$A^{(0)}_{4,1}$	0.4670018408674211	$c^{(0)}_3$	1
$A^{(0)}_{4,2}$	0.8046204792187386	$c^{(0)}_4$	1
$A^{(0)}_{4,3}$	-0.27162232008616016	$c^{(1)}_1$	0
$A^{(1)}_{2,1}$	0.45605532163856893	$c^{(1)}_2$	0.45605532163856893
$A^{(1)}_{3,1}$	0.7555807846451692	$c^{(1)}_3$	1
$A^{(1)}_{3,2}$	0.24441921535482677	$c^{(1)}_4$	1
$A^{(1)}_{4,1}$	0.6981181143266059	$\beta^{(1)}_1$	-0.45315689727309133
$A^{(1)}_{4,2}$	0.3453277086024727	$\beta^{(1)}_2$	0.8330937231303951
$A^{(1)}_{4,3}$	-0.04344582292908241	$\beta^{(1)}_3$	0.3792843195533544
$B^{(0)}_{2,1}$	0.08852381537667678	$\beta^{(1)}_4$	0.24077885458934192
$B^{(0)}_{3,1}$	1.0317752458971061	$\beta^{(2)}_1$	-0.4994383733810986
$B^{(0)}_{3,2}$	0.4563552922077882	$\beta^{(2)}_2$	0.9181786186154077
$B^{(0)}_{4,1}$	1.730782804441 24	$\beta^{(2)}_3$	-0.25613778661003145
$B^{(0)}_{4,2}$	-0.46089678470929774	$\beta^{(2)}_4$	-0.16260245862427797
$B^{(0)}_{4,3}$	-0.9637509618944188	$\beta^{(3)}_1$	1.4531568972730915
$B^{(1)}_{2,1}$	0.6753186815412179	$\beta^{(3)}_2$	-0.8330937231303933
$B^{(1)}_{3,1}$	-0.07452812525785148	$\beta^{(3)}_3$	-0.3792843195533583
$B^{(1)}_{3,2}$	-0.49783736486149366	$\beta^{(3)}_4$	-0.24077885458934023
$B^{(1)}_{4,1}$	-0.5591906709928903	$\beta^{(4)}_1$	-0.4976090683622265
$B^{(1)}_{4,2}$	0.022696571806569924	$\beta^{(4)}_2$	0.9148155835648892
$B^{(1)}_{4,3}$	-0.8984927888368557	$\beta^{(4)}_3$	-1.4102107084476505
α_1	-0.15036858140642623	$\beta^{(4)}_4$	0.9930041932449877
α_2	0.7545275856696072		

Acknowledgments

We would like to thank the members of SciML, specifically David Widmann (@devmotion), Yingbo Ma (@Ying-

boMa) and (@dextorious) for their contributions to the ecosystem. Their efforts have helped make the development of efficient implementations possible. This work was partly supported by the NSF grant DMS1763272 and a grant from the Simons Foundation (594598, QN). This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

References

[1] C. Rackauckas and Q. Nie, "Mean-independent noise control of cell fates via intermediate states," *iScience*, vol. Accepted, 2018.

[2] B. Wang and Q. Qi, "Modeling the lake eutrophication stochastic ecosystem and the research of its stability," *Mathematical Biosciences*, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0025556418301780

[3] T. Hong, K. Watanabe, C. H. Ta, A. Villarreal-Ponce, Q. Nie, and X. Dai, "An ovol2-zeb1 mutual inhibitory circuit governs bidirectional and multi-step transition between epithelial and mesenchymal states," *PLoS Comput Biol*, vol. 11, no. 11, p. e1004569, 2015.

[4] D. K. Wells, W. L. Kath, and A. E. Motter, "Control of stochastic and induced switching in biophysical networks," *Phys. Rev. X*, vol. 5, p. 031036, Sep 2015. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.5.031036

[5] A. K. Ghosh, F. Hussain, S. Jha, C. J. Langmead, and S. K. Jha, "Discovering rare behaviours in stochastic differential equations using decision procedures: applications to a minimal cell cycle model," *International Journal of Bioinformatics Research and Applications 2*, vol. 10, no. 4-5, pp. 540–558, 2014.

[6] S. Donnet and A. Samson, "A review on estimation of stochastic differential equations for pharmacokinetic/pharmacodynamic models," *Advanced drug delivery reviews*, vol. 65, no. 7, pp. 929–939, 2013.

[7] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*. Springer, 1991.

[8] T. Li, A. Abdulle, and W. E, "Effectiveness of implicit methods for stiff stochastic differential equations," *Commun. Comput. Phys*, vol. 3, no. 2, pp. 295–307, 2008.

[9] C. Rackauckas and Q. Nie, "Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory," *Discrete and Continuous Dynamical Systems - Series B*, vol. 22, no. 7, pp. 2731–2761, 2016.

[10] J. Lawson, "An order five runge-kutta process with extended region of stability," *SIAM Journal on Numerical Analysis*, vol. 3, no. 4, pp. 593–597, 1966.

[11] D. I. K. Ahmadia and A. J., "Optimal stability polynomials for numerical integration of initial value problems," *CAMCOS*, vol. 7, no. 2, pp. 247–271, 2012.

[12] P. J. van der Houwen, "Explicit runge-kutta formulas with increased stability boundaries," *Numerische Mathematik*, vol. 20, no. 2, pp. 149–164, 1972. [Online]. Available: http://dx.doi.org/10.1007/BF01404404

[13] Y. Komori and K. Burrage, "Strong first order s-rock methods for stochastic differential equations," *Journal of Computational and Applied Mathematics*, vol. 242, no. Supplement C, pp. 261–274, 2013.

[14] A. Abdulle and S. Cirilli, "S-rock: Chebyshev methods for stiff stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 997–1014, 2008. [Online]. Available: https://doi.org/10.1137/070679375

[15] Y. Komori and K. Burrage, "Weak second order s-rock methods for stratonovich stochastic differential equations," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2895–2908, 2012.

[16] T. Schaffter, "From genes to organisms: Bioinformatics system models and software," Thesis, École polytechnique fédérale de Lausanne, 2014.

[17] H. Gilsing and T. Shardlow, "Sdelab: A package for solving stochastic differential equations in matlab," *Journal of Computational and Applied Mathematics*, vol. 205, no. 2, pp. 1002–1018, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042706004195

[18] A. Janicki, A. Izydorczyk, and P. Gradalski, *Computer Simulation of Stochastic Models with SDE-Solver Software Package*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 361–370.

[19] A. Rossler, "Runge kutta methods for the strong approximation of solutions of stochastic differential equations," *SIAM Journal on Numerical Analysis*, vol. 48, no. 3, pp. 922–952, 2010.

[20] M. Wiktorsson, "Joint characteristic function and simultaneous simulation of iterated ito integrals for multiple independent brownian motions," *The Annals of Applied Probability*, pp. 470–487, 2001. [Online]. Available: http://projecteuclid.org/euclid.aoap/1015345301

[21] F. S. Lawrence, "Some practical runge-kutta formulas," *Math. Comput.*, vol. 46, no. 173, pp. 135–150, 1986.

[22] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0771050X80900133

[23] J. C. Butcher, "Numerical methods for ordinary differential equations in the 20th century," *Journal of Computational and Applied Mathematics*, vol. 125, no. 1-2, pp. 1–29, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042700004556

[24] C. Tsitouras, "Runge-kutta pairs of order 5(4) satisfying only the first column simplifying assumption," *Computers & Mathematics with Applications*, vol. 62, no. 2, pp. 770–775, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0898122111004706

[25] W. H. Enright, D. J. Higham, B. Owren, and P. W. Sharp, "A survey of the explicit runge-kutta method," 1995.

[26] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Springer Berlin Heidelberg, 2011.

[27] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017. [Online]. Available: https://doi.org/10.1137/15M1020575

[28] S. G. Johnson, "The nlopt nonlinear-optimization package," 2018. [Online]. Available: http://ab-initio.mit.edu/nlopt

[29] A. Conn, N. Gould, and P. Toint, "A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Numerical Analysis*, vol. 28, no. 2, pp. 545–572, 1991. [Online]. Available: https://doi.org/10.1137/0728030

[30] E. G. Birgin and J. M. Martinez, "Improving ultimate convergence of an augmented lagrangian method," *Optimization Methods and Software*, vol. 23, no. 2, pp. 177–195, 2008. [Online]. Available: https://doi.org/10.1080/10556780701577730

[31] T. P. Runarsson and Y. Xin, "Search biases in constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 2, pp. 233–243, 2005.