Lecture 5: Stochastic Runge–Kutta Methods

Arno Solin

Aalto University, Finland

November 25, 2014

Contents

Overview of this lecture

- Runge–Kutta methods for ODEs
  - Taylor series.
  - General Runge–Kutta schemes.
  - Explicit and implicit schemes.
- Strong stochastic Runge–Kutta methods
  - Itô–Taylor series.
  - A family of strong order 1.0 schemes.
  - The iterated Itô integrals.
- Weak stochastic Runge–Kutta methods
  - A family of weak order 2.0 schemes.
  - Approximating the iterated Itô integrals.

Runge–Kutta: Basic principles

- A family of iterative methods for solving differential equations.
- Based on Taylor series (see the previous lecture), ...
- ... but are derivative-free.
- Plug-and-play methods that only requires specification of the differential equation (at least ideally).
- There are other methods as well (not considered here):
  - Multistep methods (*e.g.* Adams methods)
  - Multiderivative methods
  - Higher-order methods (*e.g.* Nyström method)
  - Tailored methods (for specific problems)

Runge–Kutta: Motivation

- Consider a first-order non-linear ODE

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), t), \qquad \mathbf{x}(t_0) = \text{given},$$

- The simplest Runge–Kutta method is the (forward) Euler scheme.
- It is based on sequential linearization of the ODE system:

$$\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) + \mathbf{f}(\hat{\mathbf{x}}(t_k), t_k)\,\Delta t.$$

- Easy to understand and implement.
- The global error of the method depends linearly on the step size Δt.

Taylor series [1/2]

- The ODE system can be integrated to give

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}(\tau), \tau)\, \mathrm{d}\tau.$$

- Recall from the previous lecture that we used a Taylor series expansion for the solution of the ODE

$$\begin{aligned}\mathbf{x}(t) = \mathbf{x}(t_0) &+ \mathbf{f}(\mathbf{x}(t_0), t_0)\,(t - t_0) \\ &+ \frac{1}{2!}\,\mathcal{L}\,\mathbf{f}(\mathbf{x}(t_0), t_0)\,(t - t_0)^2 \\ &+ \frac{1}{3!}\,\mathcal{L}^2\,\mathbf{f}(\mathbf{x}(t_0), t_0)\,(t - t_0)^3 + \ldots\end{aligned}$$

- We used the linear operator

$$\mathcal{L}(\bullet) = \frac{\partial}{\partial t}(\bullet) + \sum_i f_i \frac{\partial}{\partial x_i}(\bullet)$$

Taylor series [2/2]

- In other words, the series expansion is equal to

$$
\begin{aligned}
\mathbf{x}(t) = \mathbf{x}(t_0) &+ \mathbf{f}(\mathbf{x}(t_0), t_0)\,(t - t_0) \\
&+ \frac{1}{2!}\left\{\frac{\partial}{\partial t}\mathbf{f}(\mathbf{x}(t_0), t_0) + \sum_i f_i(\mathbf{x}(t_0), t_0)\,\frac{\partial}{\partial x_i}\mathbf{f}(\mathbf{x}(t_0), t_0)\right\}(t - t_0)^2 \\
&+ \frac{1}{3!}\left\{\frac{\partial[\mathcal{L}\,\mathbf{f}(\mathbf{x}(t_0), t_0)]}{\partial t} + \sum_i f_i(\mathbf{x}(t_0), t_0)\,\frac{\partial[\mathcal{L}\,\mathbf{f}(\mathbf{x}(t_0), t_0)]}{\partial x_i}\right\}(t - t_0)^3 \\
&+ \ldots
\end{aligned}
$$

- If we were only to consider the terms up to Δt, we would recover the Euler method.

Derivation of a higher-order method [1/4]

- However, here we wish to get hold of higher-order methods.
- For the sake of simplicity, we now stop at the term $(t - t_0)^2 = (\Delta t)^2$.
- We get

$$
\begin{aligned}
\mathbf{x}(t_0 + \Delta t) \approx \mathbf{x}(t_0) + \mathbf{f}(\mathbf{x}(t_0), t_0)\, \Delta t \\
+ \frac{1}{2} \left\{ \frac{\partial}{\partial t} \mathbf{f}(\mathbf{x}(t_0), t_0) + \sum_i f_i(\mathbf{x}(t_0), t_0)\, \frac{\partial}{\partial x_i} \mathbf{f}(\mathbf{x}(t_0), t_0) \right\} (\Delta t)^2.
\end{aligned}
$$

- We aim to get rid of the derivatives and be able to write the expression in terms of the function $\mathbf{f}(\cdot, \cdot)$ evaluated at various points.

Derivation of a higher-order method [2/4]

- We now seek a form with an extra stage:

$$\mathbf{x}(t_0 + \Delta t) \approx \mathbf{x}(t_0) + A\,\mathbf{f}(\mathbf{x}(t_0), t_0)\,\Delta t \\ + B\,\mathbf{f}(\mathbf{x}(t_0) + C\,\mathbf{f}(\mathbf{x}(t_0), t_0)\,\Delta t, t_0 + D\,\Delta t)\,\Delta t,$$

  where A, B, D, and D are unknown.

- In the last term, we can consider the truncated Taylor expansion (linearization) around $\mathbf{f}(\mathbf{x}(t_0), t_0)$ with the chosen increments as follows:

$$\mathbf{f}(\mathbf{x}(t_0) + C\,\mathbf{f}(\mathbf{x}(t_0), t_0)\,\Delta t, t_0 + D\,\Delta t) = \mathbf{f}(\mathbf{x}(t_0), t_0) \\ + C\left(\sum_i f_i(\mathbf{x}(t_0), t_0)\,\frac{\partial}{\partial x_i}\mathbf{f}(\mathbf{x}(t_0), t_0)\right)\Delta t + D\,\frac{\partial \mathbf{f}(\mathbf{x}(t_0), t_0)}{\partial t}\,\Delta t + \cdots$$

Derivation of a higher-order method [3/4]

- Combining the previous two equations gives:

$$
\begin{aligned}
\mathbf{x}(t_0 + \Delta t) \approx \mathbf{x}(t_0) + (A + B)\, \mathbf{f}(\mathbf{x}(t_0), t_0)\, \Delta t \\
+ B \left[C \sum_i f_i(\mathbf{x}(t_0), t_0)\, \frac{\partial}{\partial x_i} \mathbf{f}(\mathbf{x}(t_0), t_0) + D\, \frac{\partial \mathbf{f}(\mathbf{x}(t_0), t_0)}{\partial t} \right] (\Delta t)^2.
\end{aligned}
$$

- If we now compare the above equation to the original truncated Taylor expansion, we get the following conditions for our coefficients:

$$
A + B = 1, \quad B = \frac{1}{2}, \quad C = 1, \quad \text{and} \quad D = 1.
$$

Derivation of a higher-order method [4/4]

- We derived here is a two-stage method (known as Heun's method):

$$\hat{\mathbf{x}}(t_0 + \Delta t) = \mathbf{x}(t_0) + \frac{\Delta t}{2} \big\{ \mathbf{f}(\tilde{\mathbf{x}}_1, t_0) + \mathbf{f}(\tilde{\mathbf{x}}_2, t_0 + \Delta t) \big\},$$

  where the supporting values are given by

$$\begin{aligned} \tilde{\mathbf{x}}_1 &= \mathbf{x}(t_0), \\ \tilde{\mathbf{x}}_2 &= \mathbf{x}(t_0) + \mathbf{f}(\tilde{\mathbf{x}}_1, t_0)\, \Delta t. \end{aligned}$$

- The method (in practice the finite differences) are determined by the choices we did in truncating the series expansion.
- This method is of order 2.

A general Runge–Kutta method

Algorithm: Runge–Kutta method

Start from $\hat{\mathbf{x}}(t_0) = \mathbf{x}(t_0)$ and divide the integration interval $[t_0, t]$ into n steps $t_0 < t_1 < t_2 < \ldots < t_n = t$ such that $\Delta t = t_{k+1} - t_k$. The integration method is defined by its Butcher tableau:

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \boldsymbol{\alpha}^{\mathsf{T}} \end{array}$$

On each step k approximate the solution as follows:

$$\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) + \sum_{i=1}^{s} \alpha_i \, \mathbf{f}(\tilde{\mathbf{x}}_i, \tilde{t}_i) \, \Delta t,$$

where $\tilde{t}_i = t_k + c_i \Delta t$ and $\tilde{\mathbf{x}}_i = \hat{\mathbf{x}}(t_k) + \sum_{j=1}^{s} A_{i,j} \, \mathbf{f}(\tilde{\mathbf{x}}_j, \tilde{t}_j) \, \Delta t$.

Butcher tableau

- Ordinary Runge–Kutta methods are commonly expressed in terms of a table called the Butcher tableau:

$$
\begin{array}{c|cccc}
c_1 & A_{1,1} & & & \\
c_2 & A_{2,1} & A_{2,2} & & \\
\vdots & \vdots & & \ddots & \\
c_s & A_{s,1} & A_{s,2} & \ldots & A_{s,s} \\
\hline
 & \alpha_1 & \alpha_2 & \ldots & \alpha_s
\end{array}
$$

Example: Forward Euler

Example (Forward Euler)

The forward Euler scheme has the Butcher tableau:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

which gives the recursion $\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) + \mathbf{f}(\hat{\mathbf{x}}(t_k), t_k)\,\Delta t$.

Example: RK4

Example (The fourth-order Runge–Kutta method)

The well-known RK4 method in Ch. 1 has the following Butcher tableau:

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array}
$$

Implicit schemes [1/2]

- We have considered this far so-called explicit schemes.
- Numerical instability, when the solution includes rapidly varying terms (stiff problems).
- Explicit schemes use very small step sizes in order to not diverge from a solution path (computationally demanding).
- In implicit Runge–Kutta methods, the Buther tableau is no longer lower-triangular.
- On every step, a system of algebraic equations has to be solved (computationally demanding, but more stabile).

Implicit schemes [2/2]

- The simplest implicit method is the backward Euler scheme.

Example (Backward Euler)

The implicit backward Euler scheme has the Butcher tableau:

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

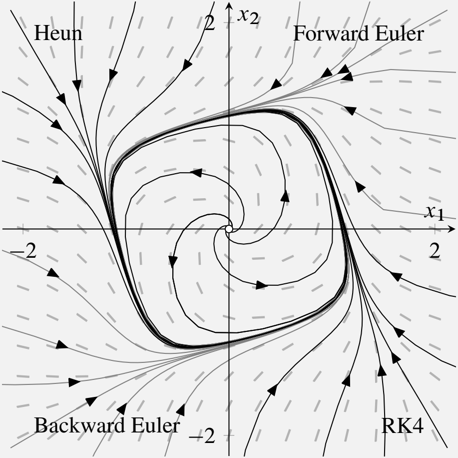
which gives the recursion $\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) + \mathbf{f}(\hat{\mathbf{x}}(t_{k+1}), t_k + \Delta t)\,\Delta t$.

Example [1/2]

- We study the two-dimensional non-linear ordinary differential equation system

$$
\begin{aligned}
\dot{x}_1 &= x_1 - x_2 - x_1^3, \\
\dot{x}_2 &= x_1 + x_2 - x_2^3,
\end{aligned}
$$

Example [2/2]



Strong stochastic Runge–Kutta: Basic principles

- A family of iterative methods for solving stochastic differential equations.
- Based on Itô–Taylor series (see the previous lecture), ...
- ... but are derivative-free.
- Plug-and-play methods that only requires specification of the drift and diffusion function of the SDE (at least ideally).
- We divide the methods into strong and weak methods (as we did for the Itô–Taylor series approximations).
- A word of warning: Stochastic Runge–Kutta methods are not as easy to grasp as the ordinary ones.

Itô–Taylor series [1/1]

- Recall the following multi-dimensional SDE formulation

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}(t), t)\,\mathrm{d}t + \mathbf{L}(\mathbf{x}(t), t)\,\mathrm{d}\boldsymbol{\beta}, \qquad \mathbf{x}(t_0) \sim p(\mathbf{x}(t_0)),$$

  where the drift is defined by $\mathbf{f} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ and the diffusion coefficients by $\mathbf{L} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d \times \mathbb{R}^m$.
- The driving noise process

$$\boldsymbol{\beta}(t) = (\beta^{(1)}(t), \beta^{(2)}(t), \ldots, \beta^{(m)}(t))$$

  is an m-dimensional standard Brownian motion.
- In integral form the equation can be expressed as

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}(\tau), \tau)\,\mathrm{d}\tau + \int_{t_0}^{t} \mathbf{L}(\mathbf{x}(\tau), \tau)\,\mathrm{d}\boldsymbol{\beta}(\tau).$$

Itô–Taylor series [1/2]

- Applying the Itô formula to the terms $\mathbf{f}(\mathbf{x}(t), t)$ and $\mathbf{L}(\mathbf{x}(t), t)$ and collecting the terms gives an Itô–Taylor series expansion of the solution (see the previous lecture):

$$
\begin{aligned}
\mathbf{x}(t) = \mathbf{x}(t_0) + \mathbf{f}(\mathbf{x}(t_0), t_0)\,(t - t_0) + \mathbf{L}(\mathbf{x}(t_0), t_0)\,(\boldsymbol{\beta}(t) - \boldsymbol{\beta}(t_0)) \\
&+ \int_{t_0}^{t} \int_{t_0}^{\tau} \mathcal{L}_t \mathbf{f}(\mathbf{x}(\tau), \tau)\,\mathrm{d}\tau\,\mathrm{d}\tau \\
&+ \sum_i \int_{t_0}^{t} \int_{t_0}^{\tau} \mathcal{L}_{\beta,i} \mathbf{f}(\mathbf{x}(\tau), \tau)\;\mathrm{d}\beta^{(i)}(\tau)\,\mathrm{d}\tau \\
&+ \int_{t_0}^{t} \int_{t_0}^{\tau} \mathcal{L}_t \mathbf{L}(\mathbf{x}(\tau), \tau)\;\mathrm{d}\tau\,\mathrm{d}\boldsymbol{\beta}(\tau) \\
&+ \sum_i \int_{t_0}^{t} \int_{t_0}^{\tau} \mathcal{L}_{\beta,i} \mathbf{L}(\mathbf{x}(\tau), \tau)\;\mathrm{d}\beta^{(i)}(\tau)\,\mathrm{d}\boldsymbol{\beta}(\tau).
\end{aligned}
$$

Itô–Taylor series [1/3]

- The first row in the equation is just the Euler–Maruyama scheme.
- Similarly as we did in for the ordinary RK methods, we can consider truncated series expansions of various degrees for each of these terms.
- The extra terms involving the iterated and cross-term Itô integrals complicate the formulation.
- To present a family of actual numerical methods, we consider the following family of strong order 1.0 methods due to *Andreas Rößler*...

A class of SRK methods of strong order 1.0 [1/3]

- Start from $\hat{\mathbf{x}}(t_0) = \mathbf{x}(t_0)$ and divide the integration interval $[t_0, t]$ into n steps $t_0 < t_1 < t_2 < \ldots < t_n = t$ such that $\Delta t = t_{k+1} - t_k$. The integration method is characterized by its *extended Butcher tableau*:

$$
\begin{array}{c|c|c|c}
\mathbf{c}^{(0)} & \mathbf{A}^{(0)} & \mathbf{B}^{(0)} & \\
\cline{1-3}
\mathbf{c}^{(1)} & \mathbf{A}^{(1)} & \mathbf{B}^{(1)} & \\
\hline
 & \boldsymbol{\alpha}^{\mathsf{T}} & \left[\boldsymbol{\gamma}^{(1)}\right]^{\mathsf{T}} & \left[\boldsymbol{\gamma}^{(2)}\right]^{\mathsf{T}}
\end{array}
$$

A class of SRK methods of strong order 1.0 [2/3]

- On each step k approximate the solution trajectory as follows:

$$
\begin{aligned}
\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) &+ \sum_{i=1}^{s} \alpha_i \, \mathbf{f}(\tilde{\mathbf{x}}_i^{(0)}, t_k + c_i^{(0)} \Delta t) \, \Delta t \\
&+ \sum_{i=1}^{s} \sum_{n=1}^{m} (\gamma_i^{(1)} \Delta \beta_k^{(n)} + \gamma_i^{(2)} \sqrt{\Delta t}) \, \mathbf{L}^n(\tilde{\mathbf{x}}_i^{(n)}, t_k + c_i^{(1)} \Delta t)
\end{aligned}
$$

A class of SRK methods of strong order 1.0 [3/3]

- With the supporting values

$$
\begin{aligned}
\tilde{\mathbf{x}}_i^{(0)} &= \hat{\mathbf{x}}(t_k) + \sum_{j=1}^{s} A_{i,j}^{(0)} \, \mathbf{f}(\tilde{\mathbf{x}}_j^{(0)}, t_k + c_j^{(0)} \Delta t) \, \Delta t \\
&\quad + \sum_{j=1}^{s} \sum_{l=1}^{m} B_{i,j}^{(0)} \, \mathbf{L}^l(\tilde{\mathbf{x}}_j^{(l)}, t_k + c_j^{(1)} \Delta t) \, \Delta\beta_k^{(l)}, \\
\tilde{\mathbf{x}}_i^{(n)} &= \hat{\mathbf{x}}(t_k) + \sum_{j=1}^{s} A_{i,j}^{(1)} \, \mathbf{f}(\tilde{\mathbf{x}}_j^{(0)}, t_k + c_j^{(0)} \Delta t) \, \Delta t \\
&\quad + \sum_{j=1}^{s} \sum_{l=1}^{m} B_{i,j}^{(1)} \, \mathbf{L}^l(\tilde{\mathbf{x}}_j^{(l)}, t_k + c_j^{(1)} \Delta t) \, \frac{\Delta\beta_k^{(l,n)}}{\sqrt{\Delta t}},
\end{aligned}
$$

for $i = 1, 2, \ldots, s$ and $n = 1, 2, \ldots, m$.

The increments [1/2]

- The increments in the algorithm are given by the Itô integrals:

$$\Delta\beta_k^{(i)} = \int_{t_k}^{t_{k+1}} \mathrm{d}\beta^{(i)}(\tau) \quad \text{and}$$
$$\Delta\beta_k^{(i,j)} = \int_{t_k}^{t_{k+1}} \int_{t_k}^{\tau_2} \mathrm{d}\beta^{(i)}(\tau_1)\,\mathrm{d}\beta^{(j)}(\tau_2),$$

- The increments $\Delta\beta_k^{(i)}$ are independent normally distributed random variables

$$\Delta\beta_k^{(i)} \sim \mathrm{N}(0, \Delta t).$$

The increments [2/2]

- The iterated stochastic Itô integrals $\Delta\beta_k^{(i,j)}$ are trickier.
- For these methods, when $i = j$, the multiple Itô integrals can be rewritten as
$$\Delta\beta_k^{(i,i)} = \frac{1}{2}\left(\left[\Delta\beta_k^{(i)}\right]^2 - \Delta t\right),$$
- Exact simulation from the integrals $\Delta\beta_k^{(i,j)}$, when $i \neq j$, is not possible, but can be approximated.

Example: Euler–Maruyama

Example (Euler–Maruyama Butcher tableau)

The Euler–Maruyama method has the extended Butcher tableau:

$$
\begin{array}{c|c|c|c}
0 & 0 & 0 & \\
\hline
0 & 0 & 0 & \\
\hline
 & 1 & 1 & 0
\end{array}
$$

and as we recall from the previous chapter, it is of strong order 0.5.

Example (Strong order 1.0 SRK due to Rößler)

- Consider a stochastic Runge–Kutta method with the following extended Butcher tableau:

$$
\begin{array}{c|ccc|ccc|ccc}
0 & & & & & & & & & \\
1 & 1 & & & 0 & & & & & \\
0 & 0 & 0 & & 0 & 0 & & & & \\
\cline{1-7}
0 & & & & & & & & & \\
1 & 1 & & & 1 & & & & & \\
1 & 1 & 0 & & -1 & 0 & & & & \\
\hline
 & \frac{1}{2} & \frac{1}{2} & 0 & 1 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2}
\end{array}
$$

- The (rather lengthy) algorithm is written out in the lecture notes (Alg. 6.3).

Higher-order methods

- Higher-order methods by considering more terms in the Itô–Taylor expansion.
- Not very practical in general (heavy and complicated).
- For models with some special structure this might still be feasible:
  - One-dimensional models.
  - Additive noise models.
  - Diagonal noise models.
  - Models with commutative noise.

Example: Duffing van der Pol oscillator [1/4]

- Consider a simplified version of a Duffing van der Pol oscillator

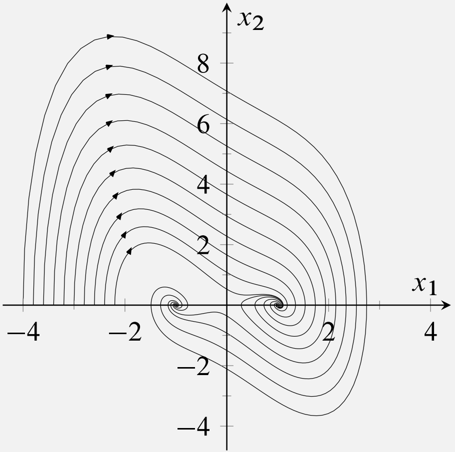
$$\ddot{x} + \dot{x} - (\alpha - x^2)\, x = x\, w(t), \quad \alpha \geq 0,$$

  driven by multiplicative white noise $w(t)$ with spectral density q.
- The corresponding two-dimensional, $\mathbf{x}(t) = (x, \dot{x})$, Itô stochastic differential equation is

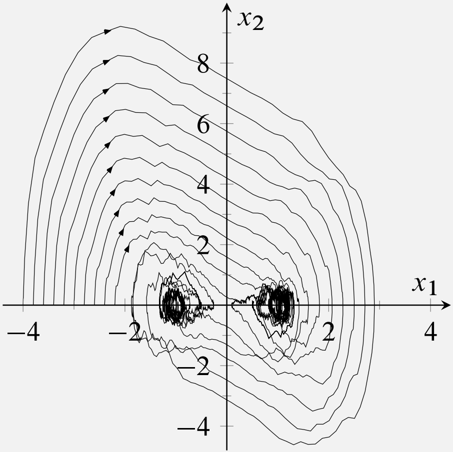
$$\begin{pmatrix} \mathrm{d}x_1 \\ \mathrm{d}x_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ (x_1(\alpha - x_1^2) - x_2 \end{pmatrix} \mathrm{d}t + \begin{pmatrix} 0 \\ x_1 \end{pmatrix} \mathrm{d}\beta,$$

  where $\beta(t)$ is a one-dimensional Brownian motion.
- Consider different initializations with $q = 0$ and $q = 0.5^2$. Use the same realizations of noise for each initialization. Use $\alpha = 1$ and $\Delta t = 2^{-5}$.

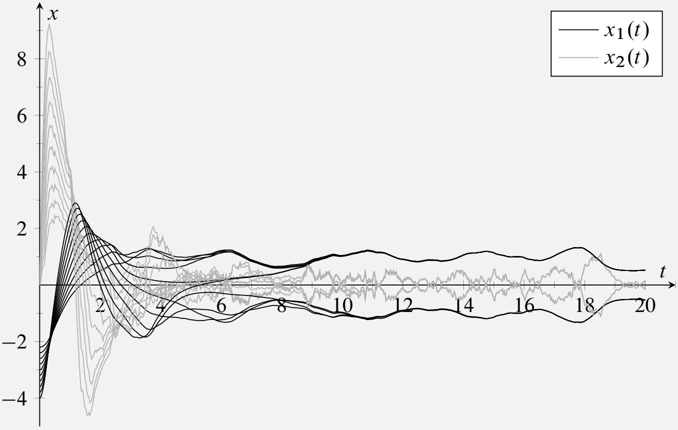
Example: Duffing van der Pol oscillator [2/4]



Example: Duffing van der Pol oscillator [3/4]



Example: Duffing van der Pol oscillator [4/4]



Weak stochastic Runge–Kutta methods

- It is possible to form weak approximations to SDEs, where the interest is not in the solution trajectories, but the distribution of them.
- We can replace weak Itô–Taylor approximations by Runge–Kutta style approximations which avoid the use of derivatives of the drift and diffusion coefficients.
- The reasoning behind the methods is very much the same as for strong SRK methods.
- Here we consider a rather general class of weak order 2.0 methods by Rößler:

A class of SRK methods of weak order 2.0 [1/3]

- Start from $\hat{\mathbf{x}}(t_0) = \mathbf{x}(t_0)$ and divide the integration interval $[t_0, t]$ into n steps $t_0 < t_1 < t_2 < \ldots < t_n = t$ such that $\Delta t = t_{k+1} - t_k$. The integration method is characterized by the following extended Butcher tableau:

$$
\begin{array}{c|c|c|c}
\mathbf{c}^{(0)} & \mathbf{A}^{(0)} & \mathbf{B}^{(0)} & \\
\hline
\mathbf{c}^{(1)} & \mathbf{A}^{(1)} & \mathbf{B}^{(1)} & \\
\hline
\mathbf{c}^{(2)} & \mathbf{A}^{(2)} & \mathbf{B}^{(2)} & \\
\hline
 & \boldsymbol{\alpha}^{\mathsf{T}} & \left[\boldsymbol{\gamma}^{(1)}\right]^{\mathsf{T}} & \left[\boldsymbol{\gamma}^{(2)}\right]^{\mathsf{T}} \\
\hline
 & & \left[\boldsymbol{\gamma}^{(3)}\right]^{\mathsf{T}} & \left[\boldsymbol{\gamma}^{(4)}\right]^{\mathsf{T}}
\end{array}
$$

A class of SRK methods of weak order 2.0 [2/3]

- On each step k approximate the solution by the following:

$$
\begin{aligned}
\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) &+ \sum_{i=1}^{s} \alpha_i \, \mathbf{f}(\tilde{\mathbf{x}}_i^{(0)}, t_k + c_i^{(0)} \Delta t) \, \Delta t \\
&+ \sum_{i=1}^{s} \sum_{n=1}^{m} \gamma_i^{(1)} \, \mathbf{L}^n(\tilde{\mathbf{x}}_i^{(n)}, t_k + c_i^{(1)} \Delta t) \, \Delta \hat{\beta}_k^{(n)} \\
&+ \sum_{i=1}^{s} \sum_{n=1}^{m} \gamma_i^{(2)} \, \mathbf{L}^n(\tilde{\mathbf{x}}_i^{(n)}, t_k + c_i^{(1)} \Delta t) \, \frac{\Delta \hat{\beta}_k^{(n,n)}}{\sqrt{\Delta t}} \\
&+ \sum_{i=1}^{s} \sum_{n=1}^{m} \gamma_i^{(3)} \, \mathbf{L}^n(\bar{\mathbf{x}}_i^{(n)}, t_k + c_i^{(2)} \Delta t) \, \Delta \hat{\beta}_k^{(n)} \\
&+ \sum_{i=1}^{s} \sum_{n=1}^{m} \gamma_i^{(4)} \, \mathbf{L}^n(\bar{\mathbf{x}}_i^{(n)}, t_k + c_i^{(2)} \Delta t) \, \sqrt{\Delta t},
\end{aligned}
$$

A class of SRK methods of weak order 2.0 [3/3]

- With supporting values

$$
\begin{aligned}
\tilde{\mathbf{x}}_i^{(0)} &= \hat{\mathbf{x}}(t_k) + \sum_{j=1}^{s} A_{i,j}^{(0)}\, \mathbf{f}(\tilde{\mathbf{x}}_j^{(0)}, t_k + c_j^{(0)} \Delta t)\, \Delta t \\
&\quad + \sum_{j=1}^{s} \sum_{l=1}^{m} B_{i,j}^{(0)}\, \mathbf{L}^l(\tilde{\mathbf{x}}_j^{(l)}, t_k + c_j^{(1)} \Delta t)\, \Delta\hat{\beta}_k^{(l)}, \\
\tilde{\mathbf{x}}_i^{(n)} &= \hat{\mathbf{x}}(t_k) + \sum_{j=1}^{s} A_{i,j}^{(1)}\, \mathbf{f}(\tilde{\mathbf{x}}_j^{(0)}, t_k + c_j^{(0)} \Delta t)\, \Delta t \\
&\quad + \sum_{j=1}^{s} \sum_{l=1}^{m} B_{i,j}^{(1)}\, \mathbf{L}^l(\tilde{\mathbf{x}}_j^{(l)}, t_k + c_j^{(1)} \Delta t)\, \Delta\hat{\beta}_k^{(l,n)}, \\
\bar{\mathbf{x}}_i^{(n)} &= \hat{\mathbf{x}}(t_k) + \sum_{j=1}^{s} A_{i,j}^{(2)}\, \mathbf{f}(\tilde{\mathbf{x}}_j^{(0)}, t_k + c_j^{(0)} \Delta t)\, \Delta t \\
&\quad + \sum_{j=1}^{s} \sum_{\substack{l=1 \\ l \neq n}}^{m} B_{i,j}^{(2)}\, \mathbf{L}^l(\tilde{\mathbf{x}}_j^{(l)}, t_k + c_j^{(1)} \Delta t)\, \frac{\Delta\hat{\beta}_k^{(l,n)}}{\sqrt{\Delta t}},
\end{aligned}
$$

The increments [1/2]

- Again, the increments are given by the double Itô integrals.
- In the weak schemes we can use the following approximations:

$$\Delta\hat{\beta}_k^{(i,j)} = \begin{cases} \frac{1}{2}\left(\Delta\hat{\beta}_k^{(i)}\,\Delta\hat{\beta}_k^{(j)} - \sqrt{\Delta t}\,\hat{\zeta}_k^{(i)}\right), & \text{if } i < j, \\ \frac{1}{2}\left(\Delta\hat{\beta}_k^{(i)}\,\Delta\hat{\beta}_k^{(j)} + \sqrt{\Delta t}\,\hat{\zeta}_k^{(j)}\right), & \text{if } i > j, \\ \frac{1}{2}\left([\Delta\hat{\beta}_k^{(i)}]^2 - \Delta t\right), & \text{if } i = j. \end{cases}$$

- Here only $2m - 1$ independent random variables are needed.
- No problems with the cross-term integrals any more.

The increments [2/2]

- For example, we can choose $\Delta\hat{\beta}_k^{(i)}$ such that they are independent three-point distributed random variables:

$$\mathrm{P}\big(\Delta\hat{\beta}_k^{(i)} = \pm\sqrt{3\,\Delta t}\big) = \frac{1}{6} \quad \text{and} \quad \mathrm{P}\big(\Delta\hat{\beta}_k^{(i)} = 0\big) = \frac{2}{3},$$

- The supporting variables $\hat{\zeta}_k^{(i)}$ such that they are independent two-point distributed random variables.

$$\mathrm{P}\big(\hat{\zeta}_k^{(i)} = \pm\sqrt{\Delta t}\big) = \frac{1}{2}.$$

Example: A weak order 2.0 method

Example (Weak order 2.0 SRK due to Rößler)

- Consider a stochastic Runge–Kutta method with the following extended Butcher tableau:

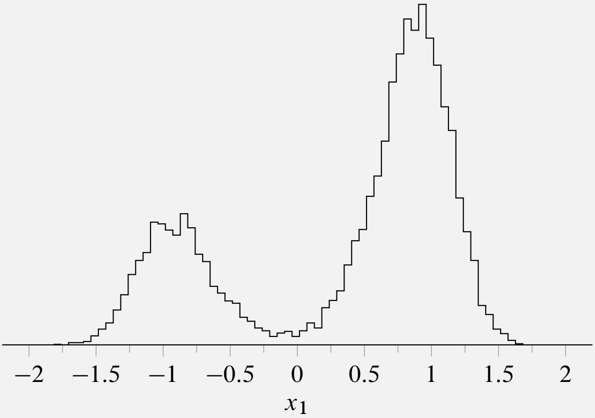
$$
\begin{array}{c|ccc|ccc|ccc}
0 & & & & & & & & & \\
1 & 1 & & & 1 & & & & & \\
0 & 0 & 0 & & 0 & 0 & & & & \\
\cline{1-7}
0 & & & & & & & & & \\
1 & 1 & & & 1 & & & & & \\
1 & 1 & 0 & & -1 & 0 & & & & \\
\cline{1-7}
0 & & & & & & & & & \\
1 & 1 & & & 1 & & & & & \\
1 & 1 & 0 & & -1 & 0 & & & & \\
\hline
 & \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{2} & -\frac{1}{2} \\
\cline{2-10}
 & & & & -\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{2} & -\frac{1}{2}
\end{array}
$$

- The (rather lengthy) algorithm is written out in the lecture notes (Alg. 6.5).

Example: Weak SRK for Duffing van der Pol [1/2]

- We are interested in characterizing the solution at $t = 20$ for the initial condition of $\mathbf{x}(0) = (-3, 0)$.
- We use the stochastic Runge–Kutta method of weak order 2.0.
- Discretization interval: $\Delta t = 2^{-4}$.
- We show the results as a histogram of $x_1(20)$ with 10,000 samples.
- With a Δt this large, the Euler–Maruyama method does not provide plausible results.

Example: Weak SRK for Duffing van der Pol [2/2]



Summary

- Stochastic Runge–Kutta methods are derivative-free methods for solving SDEs.
- They cannot be derived as simple extensions to ordinary Runge–Kutta methods.
- You cannot get rid of the iterated Itô integral.
- The complexity of the methods grows with the approximation order.
- Higher order schemes can be practical for models with some special structure (scalar, additive, commutative, *etc.*).
- The choice between a weak and strong scheme depends on your application.