

Weak integration of Stochastic Differential Equations

Xalbador Otxandorena Ieregi



Bachelor thesis

Supervisors: Adrien Laurent and Hans Z. Munthe Kaas

Mathematical Institute

University of Bergen

Norway

May 2022

Contents

Introduction	3
1 Numerical introduction to SDEs	3
1.1 Brownian motion	4
1.2 Stochastic integrals	6
1.3 The Euler-Maruyama Method	7
2 Weak Convergence Analysis	9
2.1 Strong Convergence of the EM Method	9
2.2 Weak Convergence of the EM Method	12
3 High Order Integrators	17
3.1 Weak order conditions for Runge Kutta methods	17
3.2 Numerical tests for order 2 schemes	20
Conclusion	24

Introduction

The objective of this project is to give a numerical introduction to stochastic differential equations (SDEs). Then, we will present integrators to approximate SDEs and we will study their weak convergence. Approximating SDEs has great applications in fields subject to random perturbations such as finance [9] and molecular dynamics [7]. In Section 1, we will follow [4] to define SDEs and we will present the Euler-Maruyama (EM) method. In Section 2, we will study strong and weak convergence of the EM method and in Section 3 we will deduce weak order conditions for Runge-Kutta methods and present and implement a second order Runge-Kutta method.

1 Numerical introduction to SDEs

Let x_0 be a point in \mathbb{R}^d and $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be a smooth Lipschitz vector field. Then, the *ordinary differential equation (ODE)*

$$\begin{cases} \frac{dx}{dt}(t) = f(x(t)), & t > 0, \\ x(0) = x_0, \end{cases}$$

has a unique solution $x : [0, \infty) \rightarrow \mathbb{R}^d$ which has the integral formulation

$$x(t) = x_0 + \int_0^t f(x(s))ds.$$

However, experimentally measured trajectories rarely behave as predicted and they have perturbations many times. Therefore, it is convenient to include a random variable disturbing the system. It can be written this way:

$$\begin{cases} dX(t) = f(X(t))dt + g(X(t))dW(t), & t > 0, \\ X(0) = X_0, \end{cases} \quad (1.1)$$

where $g : \mathbb{R}^d \rightarrow \mathbb{M}^{d \times l}$ is a matrix function and W is a l -dimensional *Wiener process* or *Brownian motion*. Now the solution X is no longer a deterministic variable but a random variable. Also notice that we do not write $dW(t)/dt$ since the Brownian motion is nowhere differentiable almost surely. The system (1.1) is called a *stochastic differential equation*.

Remark 1. If $g \equiv 0$ and X_0 is deterministic, then the system (1.1) becomes a deterministic ordinary differential equation $dX(t)/dt = f(X(t))$, with $X(0) = X_0$.

Under some conditions, it can be proven that the system (1.1) has a unique solution. As shown in [2], if f and g are continuous and Lipschitz, there exists a unique solution of (1.1) which has the integral formulation

$$X(t) = X_0 + \int_0^t f(X(s))ds + \int_0^t g(X(s))dW, \quad t > 0. \quad (1.2)$$

Remark 2. A *linear* stochastic differential equation is of the form

$$dX(t) = \lambda X(t)dt + \mu X(t)dW(t), \quad X(0) = X_0, \quad \lambda, \mu \in \mathbb{R}, \quad (1.3)$$

and its explicit solution is $X(t) = X_0 \exp((\lambda - \frac{1}{2}\mu^2)t + \mu W(t))$. However, for solving most SDEs numerical methods such as the Euler-Maruyama method are required.

Now we will define the *Brownian motion* W and the *stochastic integral* $\int_0^t \cdots dW$.

1.1 Brownian motion

Definition 1 (Brownian motion). A real-valued *Brownian motion* or *Wiener process* is a stochastic process $W(t)$ over $[0, T]$ which satisfies these three conditions:

1. $W(0) = 0$,
2. The random variable given by the increment $W(t) - W(s)$ is $N(0, t - s)$ for $0 \leq s < t \leq T$,
3. For $0 \leq s < t < u < v \leq T$ the increments $W(t) - W(s)$ and $W(v) - W(u)$ are independent.

Let us now consider a discretized Brownian motion: Set $h = T/N$ for a positive integer N and denote W_j as $W(t_j)$ with $t_j = hj$. From condition 1 we get $W_0 = 0$ and from conditions 2 and 3 we get

$$W_j = W_{j-1} + dW_j, \quad j = 1, \dots, N,$$

where each dW_j is an independent random variable with distribution $\sqrt{h}N(0, 1)$. This allows us to simulate a Brownian path with MATLAB.

```

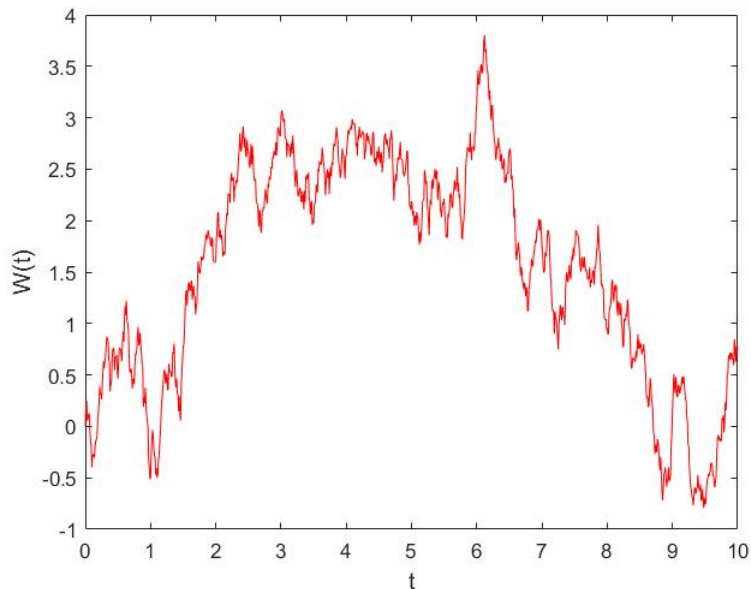
1 % Brownian Motion simulation
2
3 randn('state', 10) % seed of the random number generator
4 T = 10; N = 1000; dt = T/N;
```

```

5 dW = zeros(1,N);
6 W = zeros(1,N);
7
8 dW(1) = sqrt(dt)*randn;      % first increment
9 W(1) = dW(1);
10 for j = 2:N
11     dW(j) = sqrt(dt)*randn;  % general increment
12     W(j) = W(j-1)+dW(j);
13 end
14
15 plot([0:dt:T],[0,W], 'r-')
16 xlabel('t','FontSize',12)
17 ylabel('W(t)','FontSize',12)

```

Figure 1: Plot of one trajectory of a Brownian motion with $N = 1000$, $T = 10$



Here, the random number generator `randn` is used, each time producing an independent number from the $N(0,1)$ distribution. In order to make the simulations repeatable, we set arbitrarily the seed of the generator `randn` to be 10. Then we plot W over the interval $[0,10]$.

The Brownian motion can be easily extended to \mathbb{R}^n . We say that the stochastic process $W = (W^1, \dots, W^n)$ is an n -dimensional Brownian motion if W^k is a 1-dimensional Brownian motion.

1.2 Stochastic integrals

We refer to [5] for mathematical details in the construction of the stochastic integral.

Given a real valued function f , the integral $\int_0^T f(t)dt$ can be approximated by the Riemann sum

$$\sum_{j=0}^{N-1} f(t_j)(t_{j+1} - t_j), \quad (1.4)$$

where we use the same discretization of $[0, T]$ used in Section 1.1. The sum converges to the integral as $h \rightarrow 0$.

Similarly, we may consider the approximation of the stochastic integral $\int_0^T f(t)dW(t)$ with a sum of the form

$$\sum_{j=0}^{N-1} f(t_j)(W(t_{j+1}) - W(t_j)),$$

which is called the *Itô integral*.

We could also use another Riemann sum approximation of $\int_0^T f(t)dt$ different from (1.4) given by

$$\sum_{j=0}^{N-1} f\left(\frac{t_j + t_{j+1}}{2}\right)(t_{j+1} - t_j).$$

This sum also converges to the integral as $h \rightarrow 0$. From this we obtain a different approximation of the stochastic integral $\int_0^T f(t)dW(t)$ called the *Stratonovich integral* of the form

$$\sum_{j=0}^{N-1} f\left(\frac{t_j + t_{j+1}}{2}\right)(W(t_{j+1}) - W(t_j)).$$

We can compare both approximations using MATLAB. As an example, we will compute the Itô and Stratonovich integrals of the function $f(x) = x$ over the interval $[0, 10]$.

```
1 % Approximation of stochastic integrals
2 % Ito and Stratonovich integrals of f dW
3 % f(x)=x
4
5 randn('state',10) % set the state of randn
6 T = 10; N = 1000; dt = T/N;
7
```

```

8 dW = sqrt(dt)*randn(1,N);           % increments
9 W = cumsum(dW);                     % cumulative sum
10 f = @(x) x;
11
12 ito = sum(f([dt:dt:T]).*dW)
13 strat = sum(f(0.5*([0:dt:T-dt]+[dt:dt:T])).*dW)

```

Here instead of using a `for` loop to compute the Brownian motion we use vectorized commands for computational efficiency. The command `randn(1,N)` generates an N -vector of independent values of the distribution $N(0,1)$ and `cumsum` computes the cumulative sum of all the previous increments.

We get `ito=-7.5445` and `strat=-7.5483`. Both results are very similar.

1.3 The Euler-Maruyama Method

Solving SDEs usually requires numerical methods. For that we discretize the interval $[0, T]$ with $h = T/N$ and $t_j = jh$ for some big natural number N . From the integral form (1.2) of an SDE, notice that

$$X(t_j) = X(t_{j-1}) + \int_{t_{j-1}}^{t_j} f(X(s))ds + \int_{t_{j-1}}^{t_j} g(X(s))dW(s).$$

Considering the Itô integral and denoting the numerical approximation of $X(t_j)$ as X_j , the *Euler-Maruyama* (EM) method consists of

$$\begin{cases} X_0 = X(0), \\ X_j = X_{j-1} + f(X_{j-1})h + g(X_{j-1})(W(t_j) - W(t_{j-1})), \quad j = 1, \dots, N. \end{cases}$$

Note that in the deterministic case ($g \equiv 0$) the EM method reduces to the Euler method.

We will use MATLAB to implement the Euler-Maruyama method to the linear stochastic differential equation (1.3) with $\lambda = 2$, $\mu = 1$, $X_0 = 1$ in the interval $[0, 1]$.

```

1 % Euler-Maruyama method on linear SDE
2 % SDE is dX=lambda*X dt + mu*X dW, X(0) = Xzero
3 %
4 % Discretized Brownian path over [0,1] with dt = 2^(-6)
5
6 randn('state',10)           % set the state of randn
7 lambda = 2; mu = 1; Xzero = 1; T = 1; N = 2^6; dt = T/N;
8 dW = sqrt(dt)*randn(1,N); % increments

```

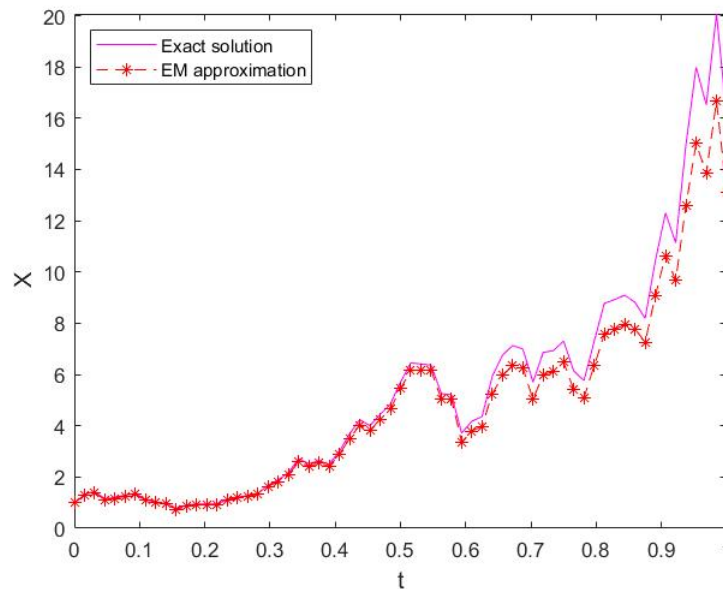
```

9 W = cumsum(dW); % discretized Brownian motion
10
11 Xtrue = Xzero*exp((lambda-0.5*mu^2)*([dt:dt:T])+mu*W);
12 plot([0:dt:T],[Xzero,Xtrue],'m-'), hold on
13
14 Xem = zeros(1,N);
15 Xtemp = Xzero;
16 for j = 1:N
17     Xtemp = Xtemp + dt*lambda*Xtemp + mu*Xtemp*dW(j);
18     Xem(j) = Xtemp;
19 end
20
21 plot([0:dt:T],[Xzero,Xem],'r--*'), hold off
22 xlabel('t','FontSize',12)
23 ylabel('X','FontSize',12,'HorizontalAlignment','right')
24 legend({'Exact solution','EM ...
approximation'},'Location','northwest')
25 emerr = abs(Xem(end)-Xtrue(end))

```

Here we compute the exact solution as `Xtrue`. Then we compare the exact solution with the EM approximation `Xem` at the endpoint $t = T$. We compute the absolute error between $X(T)$ and X_N , named as `emerr`, which was found to be 2.9858. Taking smaller steps $h = 2^{-10}$ and $h = 2^{-12}$ produce endpoint errors of 0.4900 and 0.0801 respectively.

Figure 2: Plot of the exact and the EM approximated linear SDE



2 Weak Convergence Analysis

Convergence of a numerical method is measured by comparing the exact solution $X(t_j)$ and the approximation X_j . However, since $X(t_j)$ and X_j are random variables, in order to measure their difference the expected values are used. There are two different ways:

A method is said to have *strong order of convergence* equal to p if there exists a constant C such that

$$\mathbb{E}|X_j - X(t_j)| \leq Ch^p, \quad j = 0, \dots, N, \quad (2.1)$$

for h sufficiently small.

Similarly, a method is said to have *weak order of convergence* equal to r if there exists a constant C such that

$$|\mathbb{E}\phi(X_j) - \mathbb{E}\phi(X(t_j))| \leq Ch^r, \quad j = 0, \dots, N, \quad (2.2)$$

for any smooth test function ϕ and h sufficiently small.

In simpler words, strong convergence measures 'the expected value of the error' and weak convergence 'the error of the expected values'. The strong order always satisfies $p \leq r$, since it is a more demanding measurement of the error. Weak approximation focuses on the law of the process and does not require the knowledge of W .

We will focus on the weak convergence. We will do numerical tests to study both strong and weak convergence orders of the EM method but we will give a proper theoretical proof only for the weak convergence.

2.1 Strong Convergence of the EM Method

It can be shown that the EM method has strong order of convergence $1/2$. We will test this numerically. As in [4], we will focus on the error at the endpoint $t = T$, so we let

$$e_{\Delta t}^{\text{strong}} := \mathbb{E}|X_L - X(T)|, \quad L\Delta t = T,$$

denote the endpoint error in the strong sense.

Since we know its explicit solution, we will use the linear SDE (1.3) with $\lambda = 1$, $\mu = 0.5$ and $X_0 = 2$. We compute 4000 different Brownian paths over the interval $[0, 0.1]$ with $\delta t = 2^{-12}$. For each path, the EM method is applied with 6 different step sizes: $\Delta t = 2^{q-1}\delta t$ for $1 \leq q \leq 6$. Therefore the EM method needs

the general increment $W(j\Delta t) - W((j-1)\Delta t)$. Setting $R = 2^{q-1}$, the increment is given by

$$W(j\Delta t) - W((j-1)\Delta t) = W(jR\delta t) - W((j-1)R\delta t) = \sum_{k=(j-1)R+1}^{jR} dW_k.$$

This is computed as `Winc = sum(dW(R*(j-1)+1:R*j))` in line 21.

Then, the endpoint error in the s th sample path with the q th stepsize is stored in `Xerr(s,q)` in line 24. Then the command `mean` is used to average all sample paths by replacing each column of `Xerr` by its mean. Therefore, the q th element of `mean(Xerr)` is an approximation of $e_{\Delta t}^{\text{strong}}$ for $\Delta t = 2^{q-1}\delta t$.

Now, assuming that the inequality (2.1) holds approximate equality and that the strong order of convergence is $1/2$, taking logs we have

$$\log e_{\Delta t}^{\text{strong}} \approx \log C + \frac{1}{2} \log \Delta t.$$

So from a log-log plot of $e_{\Delta t}^{\text{strong}}$ against Δt we expect to see a line of slope $1/2$. This plot is produced with blue asterisks in line 29 with command `loglog`. We also added a dashed red line with slope $1/2$ for reference in line 30. We see in Figure 3 that the two lines appear to match, suggesting that the order of convergence of the EM method is indeed $1/2$.

Finally, we assume there exists a relation $e_{\Delta t}^{\text{strong}} = C\Delta t^p$ for some constants C and p . Then, from the equality $\log e_{\Delta t}^{\text{strong}} = \log C + p \log \Delta t$, we compute a least square fit for $\log C$ and p in line 37. We get the value `p=0.5060` with a least squares residual of `resid=0.0199`.

```

1 % Test strong convergence of EM
2 %
3 % dX=lambda*X dt + mu*X dW, X(0)=Xzero,
4 % where lambda=1, mu=0.5 and Xzero=2.
5 % Discretized Brownian path over [0,0.1] with dt=2^(-12)
6 % EM uses 6 different timesteps: 32dt, 16dt, 8dt, 4dt, 2dt, dt.
7
8 randn('state',10)
9 lambda = 1; mu = 0.5; Xzero = 2; T = 0.1; N = 2^12; dt = 1/N;
10 M = 4000; % number of paths sampled
11
12 Xerr = zeros(M,6); % we will store the endpoint errors here
13 for s =1:M

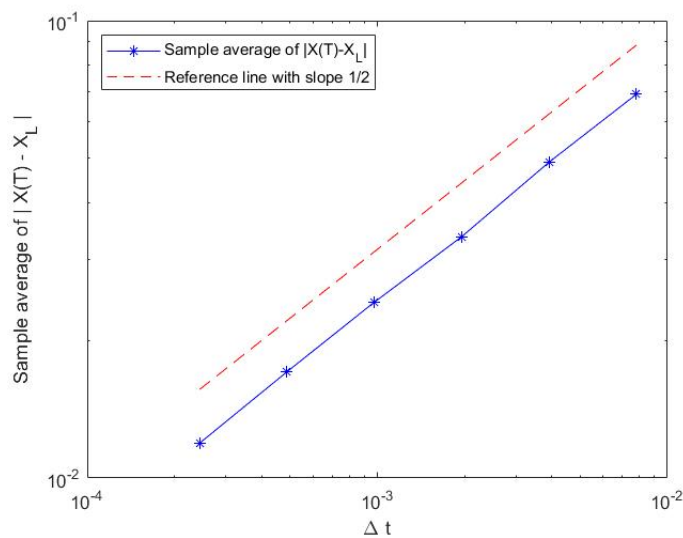
```

```

14     dW = sqrt(dt)*randn(1,N);
15     W = cumsum(dW);
16     Xtrue = Xzero*exp((lambda-0.5*mu^2)+mu*W(end));
17     for q = 1:6
18         R = 2^(q-1); Dt = R*dt; L = N/R; % L steps of size Dt=R*dt
19         Xtemp = Xzero;
20         for j = 1:L
21             Winc = sum(dW(R*(j-1)+1:R*j));
22             Xtemp = Xtemp + Dt*lambda*Xtemp + mu*Xtemp*Winc;
23         end
24         Xerr(s,q) = abs(Xtemp - Xtrue);
25     end
26 end
27
28 Dtvals = dt*(2.^([0:5]));
29 loglog(Dtvals,mean(Xerr),'b*-'), hold on
30 loglog(Dtvals,(Dtvals.^(1/2)),'r--'), hold off
31 axis([1e-4 1e-2 1e-2 1e-1])
32 xlabel('\Delta t'), ylabel('Sample average of | X(T) - X_L |')
33 legend({'Sample average of |X(T)-X_L|', 'Reference line with ...
34         slope 1/2'}, 'Location','northwest')
35
36 % Least squares fit of error = C*Dt^p to find p
37 A = [ones(6,1), log(Dtvals)']; logerr = log(mean(Xerr)');
38 sol = A\logerr; p = sol(2)
39 resid = norm(A*sol - logerr)

```

Figure 3: Test of strong convergence of EM



2.2 Weak Convergence of the EM Method

Now we will see that the EM method has weak order of convergence of 1. First, as in Section 2.1, we will do a numerical test and then we will prove it theoretically using local weak convergence orders.

Similarly to the strong convergence, we will focus on the error at the endpoint $t = T$. Also, we will focus on the case $\phi \equiv 1$ in (2.2) for simplicity. So let

$$e_{\Delta t}^{\text{weak}} := |\mathbb{E}X_L - \mathbb{E}X(T)|, \quad L\Delta t = T,$$

denote the endpoint error in the weak sense.

We will use the linear SDE (1.3) with $\lambda = 2$, $\mu = 0.1$ and $X_0 = 1$. We compute 15000 different Brownian paths over the interval $[0, 1]$ with $\Delta t = 2^{q-10}$ for $1 \leq q \leq 5$.

The sample average of the endpoint X_L with the q th stepsize Δt is stored in `xerr(q)` in line 20. The exact solution of $\mathbb{E}[X(t)]$ is computed in line 22 following from equation (1.3). Vector `xerr` stores the weak endpoint errors.

In Figure 4 we see how the weak error varies with the different Δt on a log-log scale. Again, we added a red dashed line with slope 1 for reference. We see that the two lines appear to match, supporting our hypothesis that the weak order of convergence of the EM method is 1.

```
1 % Test weak convergence of EM
2 %
3 % dX = lambda*X dt + mu*X dW, X(0)=Xzero,
4 % where lambda=2, mu=0.1 and Xzero=1.
5 % Discretized Brownian path over [0,1] with dt=2^(-9)
6 % EM uses 5 different timesteps: dt, 2dt, 4dt, 8dt, 16dt
7
8 randn('state',10)
9 lambda = 2; mu = 0.1; Xzero = 1; T = 1; dt=2^(-9);
10 M = 15000; % number of paths sampled
11
12 Xem = zeros(5,1);
13 for q = 1:5
14     R = 2^(q-1); Dt = R*dt; L = T/Dt; % L steps of size Dt
15     Xtemp = Xzero*ones(M,1);
16     for j = 1:L
17         Winc = sqrt(Dt)*randn(M,1);
18         Xtemp = Xtemp + Dt*lambda*Xtemp + mu*Xtemp.*Winc;
19     end
20     Xem(q) = mean(Xtemp); % sample average of X_L
```

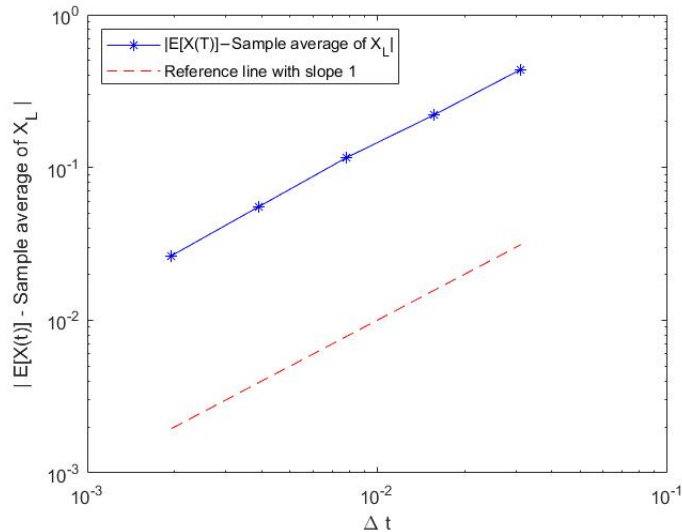
```

21 end
22 Xtrue = exp(lambda*T); % true solution for E[X(T)]
23 Xerr = abs(Xem - Xtrue);
24
25
26 Dtvals = 2.^([1:5]-10);
27 loglog(Dtvals,Xerr,'b*-' ), hold on
28 loglog(Dtvals,Dtvals,'r--' ), hold off
29 axis([1e-3 1e-1 1e-3 1])
30 xlabel('\Delta t'), ylabel('| E[X(t)] - Sample average of X_L |')
31 legend({'|E[X(T)] Sample average of X_L|', 'Reference line ...
with slope 1'}, 'Location', 'northwest')
32
33 % Least squares fit of error = C*Dt^q
34 A = [ones(q,1), log(Dtvals)']; logerr = log(Xerr);
35 sol = A\logerr; r = sol(2)
36 resid = norm(A*sol - logerr)

```

Finally, similarly to the strong convergence, we compute a least square fit for the constant $\log C$ and the weak order r in line 35. We get the value $r=1.0093$ with a least squares residual of $\text{resid}=0.0609$.

Figure 4: Test of weak convergence of EM



Now, we will present a proper proof for the weak convergence of the EM method. First, we will present the necessary notation and assumptions.

We consider the stochastic differential equation (1.1) of the form

$$dX(t) = f(X(t))dt + g(X(t))dW(t),$$

where $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is a vector field, $g : \mathbb{R}^d \rightarrow \mathbb{M}^{d \times l}$ is a matrix function and W is a l -dimensional Brownian motion.

Assumption 2.1. We assume that f and g are both smooth and globally Lipschitz.

Assumption 2.2. We assume deterministic initial condition X_0 . Also, assume that the numerical schemes $(X_j)_{j=0, \dots, N}$ considered here have bounded moments of any order, that is

$$\mathbb{E}[|X_n|^{2m}] \leq C_m < \infty,$$

for any integer m . See [8] for further details.

Definition 2. We say that a numerical scheme has *local* weak order r if the weak error after one step satisfies

$$|\mathbb{E}[\phi(X_1)|X_0 = x] - \mathbb{E}[\phi(X(h))|X(0) = x]| \leq Ch^{r+1},$$

where C is independent of h and ϕ is a test function.

Now we will present and prove a theorem linking the local and the global weak orders.

Theorem 2.1 (Global weak convergence theorem).

Let $(X_j)_{j=0, \dots, N}$ be a numerical scheme approximating X from (1.1) at time $t_j = jh$, for fixed $T > 0$, natural number N and $h = T/N$. Under Assumption 2.1 and 2.2, if the scheme has local weak order r , then for all test function ϕ there exists a positive constant C such that

$$|\mathbb{E}[\phi(X_j)|X_0 = x] - \mathbb{E}[\phi(X(t_j))|X(0) = x]| \leq Ch^r, \quad j = 0, \dots, N.$$

Proof. We use the notation X^x for a random variable X where we know that the initial condition is x . Let's begin from the last step, that is, from the time $j = N$.

$$\begin{aligned} \mathbb{E}[\phi(X^x(t_N)) - \phi(X_N^x)] &= \sum_{j=1}^N \mathbb{E}[\phi(X^{X_N^x-j}(t_j)) - \phi(X^{X_N^x-j+1}(t_{j-1}))] \\ &= \sum_{j=1}^N \mathbb{E}[\phi(X^{X_N^x-j}(h)(t_{j-1})) - \phi(X^{X_1^{X_N^x-j}}(t_{j-1}))]. \end{aligned}$$

Now consider the test function $\tilde{\phi}_i(x) = \phi \circ X^x(t_{j-1})$. Then, from the local weak convergence of order r , we know that for any integer k

$$|\mathbb{E}[\tilde{\phi}_i(X^{X_{N-j}^x}(h))] - \mathbb{E}[\tilde{\phi}_i(X_1^{X_{N-j}^x})]| \leq Ch^{r+1}(1 + \mathbb{E}[|X_{N-j}^x|^k]).$$

From Assumption 2.2, we know that for any k there exists a positive constant C_k such that

$$\mathbb{E}[|X_{N-j}^x|^k] \leq C_k.$$

Therefore, we deduce

$$\begin{aligned} |\mathbb{E}[\phi(X^x(t_N)) - \phi(X_N^x)]| &\leq \sum_{j=1}^N |\mathbb{E}[\tilde{\phi}_i(X^{X_{N-j}^x}(h))] - \mathbb{E}[\tilde{\phi}_i(X_1^{X_{N-j}^x})]| \\ &\leq Ch^{r+1} \sum_{j=1}^N (1 + \mathbb{E}[|X_{N-j}^x|^k]) \\ &\leq \tilde{C}Nh^{r+1} = \tilde{C}(Nh)h^r = \tilde{C}Th^r. \end{aligned}$$

□

Now we are finally ready to prove the following theorem:

Theorem 2.2. The Euler-Maruyama method has global weak order of convergence of 1.

Proof. First we calculate the local error. In order to simplify notation we will use for a given natural number k

$$\phi^{(k)}(a^1, \dots, a^k) = \sum_{i_1, \dots, i_k=1}^d \frac{\partial^k \phi}{\partial x_{i_1} \dots \partial x_{i_k}}(x) a_{i_1}^1 \dots a_{i_k}^k,$$

where $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ and $a^i \in \mathbb{R}^d$, $i = 1, \dots, k$.

We will use Taylor expansions for one step as in [10]. The EM method gives for the first step the expression

$$X_1 = X_0 + hf(X_0) + \sqrt{hg}(X_0)\xi,$$

where ξ is a random vector with distribution $N(0, I_l)$.

Therefore, using the Taylor expansion of ϕ at point X_0 , the expected value of $\phi(X_1)$ knowing that $X_0 = x$ is

$$\begin{aligned}\mathbb{E}[\phi(X_1)|X_0 = x] &= \mathbb{E}[\phi(x + hf(x) + \sqrt{h}g(x)\xi)] \\ &= \mathbb{E}[\phi(x) + \phi'(hf + \sqrt{h}g\xi) + \frac{1}{2}\phi''(hf + \sqrt{h}g\xi, hf + \sqrt{h}g\xi) + \dots] \\ &= \phi(x) + h\phi'(f) + \frac{h}{2}\mathbb{E}[\phi''(g\xi, g\xi)] + O(h^2) \\ &= \phi(x) + h\phi'(f) + \frac{h}{2}\phi''(g, g) + O(h^2).\end{aligned}$$

Note that we have used the fact that $\mathbb{E}[\xi] = 0$ and $\mathbb{E}[\xi\xi] = 1$. This follows from the Isserlis' theorem.

Let $u(x, t) = \mathbb{E}[\phi(X(t))|X_0 = x] = \mathbb{E}_x[\phi(X(t))]$. Then, for $x \in \mathbb{R}^d$ and $t > 0$, $u(x, t)$ fulfils the *backward Kolmogorov equations*

$$\begin{cases} \frac{\partial u}{\partial t} = \mathcal{L}u, \\ u(x, 0) = \phi(x), \end{cases}$$

where the differential operator \mathcal{L} is defined as

$$\mathcal{L}\phi = \phi'(f) + \frac{1}{2}\phi''(g, g).$$

See [8] for further details in the Kolmogorov equations and differential operator \mathcal{L} .

Therefore, the expected value of $\phi(X(h))$ knowing that the initial condition is x is

$$\begin{aligned}\mathbb{E}_x[\phi(X(h))] &= u(x, h) = u(x, 0) + h\frac{\partial u}{\partial t} + \frac{h^2}{2}\frac{\partial^2 u}{\partial t^2}u + \dots \\ &= \phi(x) + h\mathcal{L}\phi + \frac{h^2}{2}\mathcal{L}^2\phi + \dots\end{aligned}$$

Now, we see that the difference between both expected values is

$$|\mathbb{E}_x[\phi(X_1)] - \mathbb{E}_x[\phi(X(h))]| = O(h^2).$$

So there exists a positive constant C independent of h such that

$$|\mathbb{E}_x[\phi(X_1)] - \mathbb{E}_x[\phi(X(h))]| \leq Ch^2.$$

In other words, the EM scheme has local weak order of 1. From Theorem 2.1 it follows that the EM scheme has global weak order of 1. \square

3 High Order Integrators

Now we will only focus on the stochastic differential equations of the form

$$dX(t) = f(X(t))dt + \sigma dW(t), \quad (3.1)$$

where $X(t) \in \mathbb{R}^d$ is the solution with deterministic initial condition X_0 , the vector field $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is smooth and globally Lipschitz, σ is a positive constant and $W(t)$ is a d -dimensional Brownian motion.

In this section we will study the order conditions of *Runge-Kutta* (RK) type schemes of the form

$$\begin{aligned} Y_i &= X_n + h \sum_{j=1}^s a_{ij} f(Y_j) + d_i \sigma \sqrt{h} \xi_n, \quad i = 1, \dots, s, \\ X_{n+1} &= X_n + h \sum_{i=1}^s b_i f(Y_i) + \sigma \sqrt{h} \xi_n, \end{aligned} \quad (3.2)$$

where a_{ij} , b_i , d_i are the real coefficients defining the RK scheme, and ξ_n are independent random vector with distribution $N(0, I_d)$.

3.1 Weak order conditions for Runge Kutta methods

We will use exotic aromatic forests in order to simplify notation and perform expected values more easily. We refer to [6] for further details on this methodology.

Let us perform a Taylor expansion for the first step as in Theorem 2.2. The RK method gives for the first step the expression

$$\begin{aligned} X_1 &= X_0 + h \sum_{i=1}^s b_i f(Y_i) + \sigma \sqrt{h} \xi \\ &= X_0 + h \sum_{i=1}^s b_i f \left(X_0 + h \sum_{j=1}^s a_{ij} f(Y_j) + d_i \sigma \sqrt{h} \xi \right) + \sigma \sqrt{h} \xi, \end{aligned}$$

where ξ is a random variable with distribution $N(0, I_d)$.

Using Taylor expansion of f at point X_0 knowing that $X_0 = x$ we get

$$\begin{aligned}
X_1^x &= x + \Delta x = x + \sigma\sqrt{h}\xi + h \sum_{i=1}^s b_i \left[f(x) + f' \left(\sqrt{h}d_i\sigma\xi + h \sum_{j=1}^s a_{ij}f(Y_j) \right) \right. \\
&\quad \left. + \frac{1}{2}f'' \left(\sqrt{h}d_i\sigma\xi + h \sum_{j=1}^s a_{ij}f(Y_j), \sqrt{h}d_i\sigma\xi + h \sum_{j=1}^s a_{ij}f(Y_j) \right) + \dots \right] \\
&= x + \sqrt{h}\sigma\xi + h \sum_{i=1}^s b_i f + h\sqrt{h}\sigma \sum_{i=1}^s b_i d_i f' \xi + h^2 f' \sum_{i=1}^s \sum_{j=1}^s b_i a_{ij} f(Y_j) \\
&\quad + h^2 \frac{1}{2} \sigma^2 \sum_{i=1}^s b_i d_i^2 f''(\xi, \xi) + \dots
\end{aligned}$$

Let us set the sum $\sum_{j=1}^s a_{ij}$ as c_i . We will then use Δx as


$$\Delta x = \sqrt{h}\sigma\xi + h \sum_{i=1}^s b_i f + h\sqrt{h}\sigma \sum_{i=1}^s b_i d_i f' \xi + h^2 \sum_{i=1}^s b_i c_i f' f + h^2 \frac{1}{2} \sigma^2 \sum_{i=1}^s b_i d_i^2 f''(\xi, \xi).$$

Now, we apply the test function ϕ to X_1 . Performing the Taylor expansion to ϕ at point x we get

$$\begin{aligned}
\phi(X_1^x) &= \phi(x) + \phi'(\Delta x) + \frac{1}{2}\phi''(\Delta x, \Delta x) + \frac{1}{6}\phi'''(\Delta x, \Delta x, \Delta x) + \frac{1}{24}\phi^{(4)}(\Delta x, \Delta x, \Delta x, \Delta x) + \dots \\
&= \phi(x) + \sqrt{h}\sigma\phi'\xi + h \sum_{i=1}^s b_i \phi' f + h\sqrt{h}\sigma \sum_{i=1}^s b_i d_i \phi' f' \xi + h^2 \sum_{i=1}^s b_i c_i \phi' f' f \\
&\quad + h^2 \frac{\sigma^2}{2} \sum_{i=1}^s b_i d_i^2 \phi' f''(\xi, \xi) + h \frac{\sigma^2}{2} \phi''(\xi, \xi) + h\sqrt{h}\sigma \sum_{i=1}^s b_i \phi''(\xi, f) \\
&\quad + h^2 \frac{1}{2} \left(\sum_{i=1}^s b_i \right)^2 \phi''(f, f) + h^2 \sigma^2 \sum_{i=1}^s b_i d_i \phi''(f' \xi, \xi) + h\sqrt{h} \frac{1}{6} \sigma^3 \phi'''(\xi, \xi, \xi) \\
&\quad + h^2 \frac{1}{2} \sigma^2 \sum_{i=1}^s b_i \phi'''(f, \xi, \xi) + h^2 \frac{1}{24} \sigma^4 \phi^{(4)}(\xi, \xi, \xi, \xi) + \dots
\end{aligned}$$

Now we compute the expected value of $\phi(X_1)$ and we get

$$\begin{aligned}
\mathbb{E}_x \phi(X_1) &= \phi(x) + h \mathbb{E} \left[\sum_{i=1}^s b_i \phi' f + \frac{\sigma^2}{2} \phi''(\xi, \xi) \right] \\
&\quad + h^2 \mathbb{E} \left[\sum_{i=1}^s b_i c_i \phi' f' f + \frac{\sigma^2}{2} \sum_{i=1}^s b_i d_i^2 \phi' f''(\xi, \xi) + \frac{1}{2} \left(\sum_{i=1}^s b_i \right)^2 \phi''(f, f) \right. \\
&\quad \left. + \sigma^2 \sum_{i=1}^s b_i d_i \phi''(f' \xi, \xi) + \frac{\sigma^2}{2} \sum_{i=1}^s b_i \phi'''(f, \xi, \xi) + \frac{\sigma^4}{24} \phi^{(4)}(\xi, \xi, \xi, \xi) \right] + O(h^3).
\end{aligned}$$

Now we will use exotic aromatic trees [3, 6] in order to compute the expected values easier. For example, the expression $\phi' f''(\xi, \xi)$ gives the tree . The node on the bottom (called *root*) represents the test function ϕ . The edge going upwards connecting the root to the other node mean that the root ϕ is derived once. The other nodes represent the function f . See that in our example this node has two edges going upwards meaning f is derived twice. Lastly the crosses represent the random variable ξ .

Therefore, using trees we get for the expected value of $\phi(X_1)$ the expression

$$\begin{aligned}
\mathbb{E}_x \phi(X_1) &= \phi(x) + h \mathbb{E} \left[\sum_{i=1}^s b_i \text{tree}_1 + \frac{\sigma^2}{2} \text{tree}_2 \right] \\
&+ h^2 \mathbb{E} \left[\sum_{i=1}^s b_i c_i \text{tree}_3 + \frac{\sigma^2}{2} \sum_{i=1}^s b_i d_i^2 \text{tree}_4 + \frac{1}{2} \left(\sum_{i=1}^s b_i \right)^2 \text{tree}_5 \right. \\
&+ \left. \sigma^2 \sum_{i=1}^s b_i d_i \text{tree}_6 + \frac{\sigma^2}{2} \sum_{i=1}^s b_i \text{tree}_7 + \frac{\sigma^4}{24} \text{tree}_8 \right] + \dots \\
&= \phi(x) + h \left(\sum_{i=1}^s b_i \text{tree}_1 + \frac{\sigma^2}{2} \text{tree}_2 \right) + h^2 \left(\sum_{i=1}^s b_i c_i \text{tree}_3 + \frac{\sigma^2}{2} \sum_{i=1}^s b_i d_i^2 \text{tree}_4 + \frac{1}{2} \left(\sum_{i=1}^s b_i \right)^2 \text{tree}_5 \right. \\
&+ \left. \sigma^2 \sum_{i=1}^s b_i d_i \text{tree}_6 + \frac{\sigma^2}{2} \sum_{i=1}^s b_i \text{tree}_7 + \frac{\sigma^4}{8} \text{tree}_8 \right) + O(h^3).
\end{aligned}$$

Now, similarly to Theorem 2.2, we compare the expected value of $\phi(X_1)$ to the expected value of $\phi(X(h))$. We know that $\phi(X(h))$ has the following expansion

$$\mathbb{E}_x[\phi(X(h))] = \phi(x) + h \mathcal{L} \phi + \frac{h^2}{2} \mathcal{L}^2 \phi + \dots$$

where the differential operator \mathcal{L} is defined as $\mathcal{L} \phi = \phi' f + \frac{\sigma^2}{2} \Delta \phi$. In order to get weak order conditions up to 2, we need to compute the expression for $\mathcal{L}^2 \phi$.

In tree notation, we have

$$\mathcal{L} \phi = \text{tree}_1 + \frac{\sigma^2}{2} \text{tree}_2.$$

Now, applying \mathcal{L} to $\mathcal{L}\phi$ we get

$$\begin{aligned}\mathcal{L}^2\phi &= \mathcal{L}\left(\bullet + \frac{\sigma^2}{2}\overset{\circ}{\bullet}\right) = \mathcal{L}\left(\bullet\right) + \frac{\sigma^2}{2}\mathcal{L}\left(\overset{\circ}{\bullet}\right) = \bullet + \text{diag} + \frac{\sigma^2}{2}\overset{\circ}{\bullet} + \frac{\sigma^2}{2}\left(\overset{\circ}{\bullet} + \overset{\circ}{\bullet} + 2\overset{\circ}{\bullet} + \frac{\sigma^2}{2}\overset{\circ}{\circ}\right) \\ &= \bullet + \text{diag} + \sigma^2\overset{\circ}{\bullet} + \frac{\sigma^2}{2}\overset{\circ}{\bullet} + \sigma^2\overset{\circ}{\bullet} + \frac{\sigma^4}{4}\overset{\circ}{\circ}.\end{aligned}$$

Now we can compare $\mathbb{E}_x\phi(X_1)$ and $\mathbb{E}_x\phi(X(h))$ to get the weak order conditions:

RK scheme (3.2) has weak order 1 if

$$\mathbb{E}_x\phi(X_1) - \mathbb{E}_x\phi(X(h)) = O(h^2) \iff \sum_{i=1}^s b_i = 1.$$

RK scheme (3.2) has weak order 2 if

$$\mathbb{E}_x\phi(X_1) - \mathbb{E}_x\phi(X(h)) = O(h^3) \iff \sum_{i=1}^s b_i = 1, \sum_{i=1}^s b_i c_i = \frac{1}{2}, \sum_{i=1}^s b_i d_i^2 = \frac{1}{2}, \sum_{i=1}^s b_i d_i = \frac{1}{2}.$$

Remark 3. Note that setting $\sigma \equiv 0$, the SDE (3.1) becomes deterministic and the values d_i disappear in the RK method. We can see that the conditions $\sum b_i = 1$ and $\sum b_i c_i = \frac{1}{2}$ are indeed the standard order 2 conditions for deterministic RK methods.

3.2 Numerical tests for order 2 schemes

Now that we know the order conditions, we can create a RK method of weak order 2. For example, for $s = 2$, the values

$$A = \begin{pmatrix} 0 & 0 \\ 1/2 & 1/2 \end{pmatrix}, \quad b = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \quad c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad d = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

fulfil the conditions for convergence of weak order 2. Using these values the RK scheme (3.2) can also be written as

$$X_{n+1} = X_n + \frac{h}{2}f(X_n) + \frac{h}{2}f(X_{n+1}) + \sigma\xi_n.$$

This method is implicit which means that it will require more computational time. In every step we will need to solve a non-linear equation or use a fixed point method.

We will test its convergence using both a linear SDE and then a *non-linear* SDE. First, we will use the linear equation $dX(t) = \lambda X(t)dt + \sigma dW(t)$ where $\lambda = 1.5$, $\sigma = 0.1$ and $X_0 = 1$. We compute 100000 different Brownian paths over the interval $[0, 1]$ with $\Delta t = 2^{q-7}$ for $1 \leq q \leq 5$. Since the RK scheme is implicit, for every step we will perform the fixed point method. The tolerance for this method is set to be $\text{tol}=10^{-6}$.

Finally, we compute a least square fit for the constant $\log C$ and the weak order r in line 42. We get the value $r=1.8293$ with a least squares residual of $\text{resid}=0.7136$.

```

1 % Test weak convergence of RK
2 %
3 % dX = lambda*X dt + sigma dW, X(0)=Xzero,
4 % where lambda=1.5, sigma=0.1 and Xzero=1.
5 % Discretized Brownian path over [0,1] with dt=2^(-6)
6 % RK uses 5 different timesteps: dt, 2dt, 4dt, 8dt, 16dt
7
8 randn('state',10)
9 lambda = 1.5; sigma = 0.1; Xzero = 1; T = 1; dt=2^(-6);
10 M = 100000; % number of paths sampled
11 tol = 10^(-6); % accepted tolerance
12
13 Xem = zeros(5,1);
14 for q =1:5
15     R = 2^(q-1); Dt = R*dt; L = T/Dt; % L steps of size Dt
16     Xtemp = Xzero*ones(M,1);
17     for j = 1:L
18         Winc = sqrt(Dt)*randn(M,1);
19         Xnew = Xtemp;
20         gnew = g(Xtemp,Xnew,lambda,sigma,Winc,Dt);
21         while norm(Xnew-gnew)>tol % fixed point method
22             Xnew = gnew;
23             gnew = g(Xtemp,Xnew,lambda,sigma,Winc,Dt);
24         end
25         Xtemp = gnew;
26     end
27     Xem(q) = mean(Xtemp); % sample average of X_L
28 end
29 Xtrue = exp(lambda*T); % true solution for E[X(T)]
30 Xerr = abs(Xem - Xtrue);
31
32
33 Dtvals = 2.^([1:5]-10);
34 loglog(Dtvals,Xerr,'b*-'), hold on
35 loglog(Dtvals,Dtvals.^2,'r--'), hold off

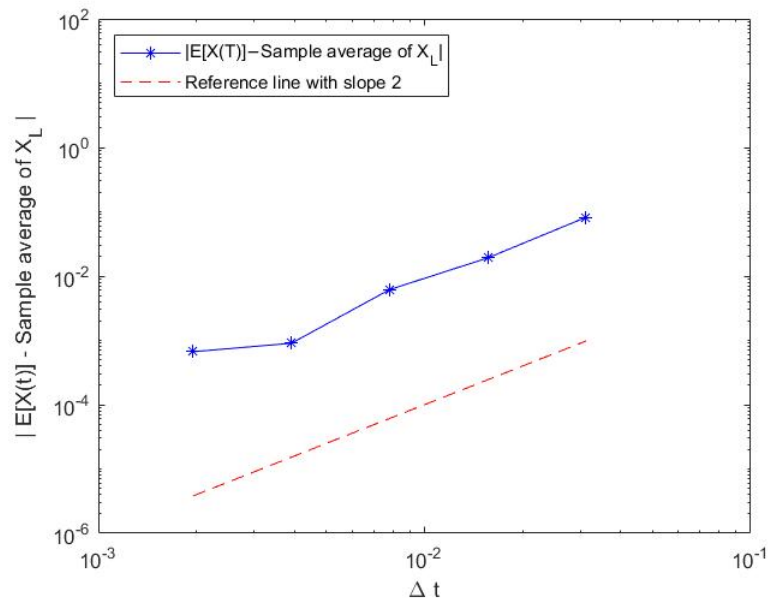
```

```

36 axis([1e-3 1e-1 1e-6 1e2])
37 xlabel('\Delta t'), ylabel('| E[X(t)] - Sample average of X_L |')
38 legend({'|E[X(T)] Sample average of X_L|', 'Reference line ...
        with slope 2'}, 'Location', 'northwest')
39
40 % Least squares fit of error = C*Dt^r
41 A = [ones(q,1), log(Dtvals)']; logerr = log(Xerr);
42 sol = A\logerr; r = sol(2)
43 resid = norm(A*sol - logerr)
44
45 function Xn = g(X0,X1,lambda,sigma,Winc,h)
46     Xn = X0 + 0.5*h*lambda*(X0+X1) + sigma*Winc; % RK method
47 end

```

Figure 5: Test of weak convergence of RK with linear SDE



Now we will do the same but with a *non-linear* SDE. We will use the equation (3.1) with $f(X(t)) = \sin(X(t))$. Since we do not know the exact solution, we will approximate the exact solution with a much smaller h . In line 31 we take the reference stepsize $dt_ref = 2^{(-6)} * dt$ and then we use our algorithm to compute the 'exact' solution.

```

1 % Test weak convergence of RK on non-linear SDE
2 %
3 % dX = sin(X) dt + sigma dW, X(0)=Xzero,
4 % where sigma=0.25 and Xzero=1.

```

```

5 % Discretized Brownian path over [0,1] with dt=2^(-4)
6 % RK uses 5 different timesteps: dt, 2dt, 4dt, 8dt, 16dt
7
8 randn('state',10)
9 sigma = 0.25; Xzero = 1; T = 1; dt=2^(-4);
10 M = 100000; % number of paths sampled
11 tol = 10^(-6); % accepted tolerance
12
13 Xem = zeros(5,1);
14 for q =1:5
15     R = 2^(q-1); Dt = R*dt; L = T/Dt; % L steps of size Dt
16     Xtemp = Xzero*ones(M,1);
17     for j = 1:L
18         Winc = sqrt(Dt)*randn(M,1);
19         Xnew = Xtemp;
20         gnew = g(Xtemp,Xnew,sigma,Winc,Dt);
21         while norm(Xnew-gnew)>tol % fixed point method
22             Xnew = gnew;
23             gnew = g(Xtemp,Xnew,sigma,Winc,Dt);
24         end
25         Xtemp = gnew;
26     end
27     Xem(q) = mean(Xtemp); % sample average of X_L
28 end
29
30 Xtemp_true = Xzero*ones(M,1); % true solution for E[X(T)]
31 dt_ref = 2^(-6)*dt; L = T/dt_ref;
32 for j = 1:L
33     Winc = sqrt(dt_ref)*randn(M,1);
34     Xnew_true = Xtemp_true;
35     gnew = g(Xtemp_true,Xnew_true,sigma,Winc,dt_ref);
36     while norm(Xnew_true-gnew)>tol
37         Xnew_true = gnew;
38         gnew = g(Xtemp_true,Xnew_true,sigma,Winc,dt_ref);
39     end
40     Xtemp_true = gnew;
41 end
42 Xtrue = mean(Xtemp_true);
43 Xerr = abs(Xem - Xtrue);
44
45
46 Dtvals = 2.^([1:5]-10);
47 loglog(Dtvals,Xerr,'b*-'), hold on
48 loglog(Dtvals,Dtvals.^2,'r--'), hold off
49 axis([1e-3 1e-1 1e-6 1e2])
50 xlabel('\Delta t'), ylabel('| E[X(t)] - Sample average of X_L |')
51 legend({'|E[X(T)] Sample average of X_L|','Reference line ...
52         with slope 2'},'Location','northwest')

```

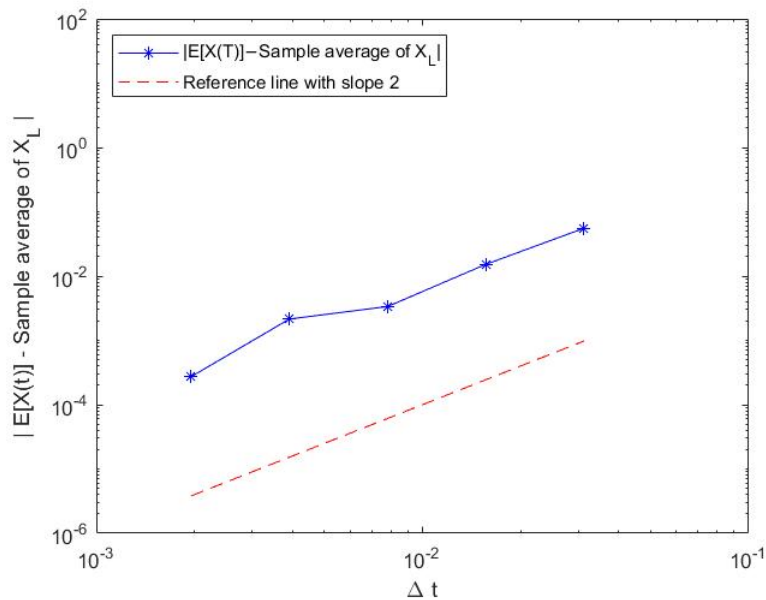
```

53 % Least squares fit of error = C*Dt^r
54 A = [ones(q,1), log(Dtvals)']; logerr = log(Xerr);
55 sol = A\logerr; r = sol(2)
56 resid = norm(A*sol - logerr)
57
58 function Xn = g(X0,X1,sigma,Winc,h)
59     Xn = X0 + 0.5*h*sin(X0)+0.5*h*sin(X1) + sigma*Winc;
60 end

```

Computing the least square fit for the constant $\log C$ and the weak order r in line 55, we get the value $r=1.8157$ with a least squares residual of $\text{resid}=0.6594$.

Figure 6: Test of weak convergence of RK with non-linear SDE



Conclusion

We have found first and second weak order conditions for RK methods for approximating SDEs of the form (3.1). We have seen that for order 1 there is 1 condition and for order 2 there are 4 conditions. However, we have not considered RK methods to approximate SDEs with multiplicative noise, that is, SDEs (1.1) with a matrix function $g : \mathbb{R}^d \rightarrow \mathbb{M}^{d \times l}$ instead of a positive constant σ . In [1] Runge-Kutta methods are presented for approximating SDEs with multiplicative noise and it was found that there are 9 conditions for order 1 schemes and 59 conditions for order 2 schemes.

References

- [1] K. Debrabant and A. Rößler. Families of efficient second order Runge-Kutta methods for the weak approximation of Itô stochastic differential equations. *Appl. Numer. Math.*, 59(3-4):582–594, 2009.
- [2] L. C. Evans. *An introduction to stochastic differential equations*. American Mathematical Society, Providence, RI, 2013.
- [3] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.
- [4] D. J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Rev.*, 43(3):525–546, 2001.
- [5] H.-H. Kuo. *Introduction to stochastic integration*. Universitext. Springer, New York, 2006.
- [6] A. Laurent and G. Vilmart. Exotic aromatic B-series for the study of long time integrators for a class of ergodic SDEs. *Math. Comp.*, 89(321):169–202, 2020.
- [7] T. Lelièvre, M. Rousset, and G. Stoltz. Computation of free energy differences through nonequilibrium stochastic dynamics: the reaction coordinate case. *J. Comput. Phys.*, 222(2):624–643, 2007.
- [8] G. N. Milstein and M. V. Tretyakov. *Stochastic numerics for mathematical physics*. Scientific Computation. Springer-Verlag, Berlin, 2004.
- [9] B. Øksendal. *Stochastic differential equations*. Universitext. Springer-Verlag, Berlin, sixth edition, 2003. An introduction with applications.
- [10] D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Anal. Appl.*, 8(4):483–509 (1991), 1990.