Crash course on numerics for SDEs — Summer and fall 2020
David Cohen (david.cohen@chalmers.se) — Central South University, Changsha

Computer labs

September 12, 2020

A list of matlab tutorials can be found under http://www.math.chalmers.se/~cohend/teachlinks.html

Task 1: The goal of this task is to simulate a Brownian motion/Wiener process $(W(t))_{t\in[0,1]}$.

(a) Write a Matlab code to simulate one realisation of a discretised Brownian motion on $[0,1]$ for different values of Δt. Consider grid points given by $t_m = m\Delta t$, where $\Delta t = 2^{-4}, 2^{-6}$, and 2^{-8}. Use the definition of a Brownian motion to compute $W(0)$, $W(\Delta t)$, $W(2\Delta t)$, etc. Plot your numerical results.

You may use the following

```
clear all
rng(113) % set the seed for the command rand
% discretised BM for dt=2^(-4)
Tend= ... ;dt= ... ;N= ...;
W(1)= ...; % BM starts at 0 a.s.
for n=...
  dW=sqrt(dt)*randn(1,1); % increment/normal rand. var.
  W(n+1)= ... ; % iter. procedure using def. of BM
end
% plot W against time
figure(),
plot([0:dt:Tend],W,'b','LineWidth',3)
...
% repeat the above for another step size dt
...
```

(b) With the help of the above part, compute the mean of $W(t)$ over 200, 2000, 20000 trajectories of $W(t)$ on $[0,1]$ with $\Delta t = 2^{-8}$. Plot your results. You may use a matrix to generate all the needed normally distributed random variables as described below

```
...
dW=sqrt(dt)*randn(M,N+1); % gen. M samples of Wiener increments
...
Wmean=mean(...);
...
```

(c) Finally, in the same figure, display 5 sample paths of $W(t)$ and the mean over 50000 trajectories of $W(t)$ on $[0,1]$ for $\Delta t = 2^{-8}$.

Task 2: The goal of this task is to simulate a stochastic process.

(a) Write a Matlab code to simulate one realisation of the discretised stochastic process (gBM) $X(t) = X_0 \exp((\mu - \frac{1}{2}\sigma^2)t + \sigma W(t))$ for $t_m = m\Delta t$, where $\mu = 2, \sigma = 1, X_0 = 1$ on

$[0,1]$ with $\Delta t = 2^{-4}, 2^{-6}$, and 2^{-8}. Plot your numerical results. Observe that you may use the previous task to generate the Wiener process $W(t)$.

```
...
X(1)=X0; % the process X starts at X0
for n=1:N
  ...
  % iter. to compute the process X, where W is Wiener proc.
  X(n+1)=X0*exp((mu-0.5*sigma^2)*n*dt+sigma*W(n+1));
end
...
```

(b) With the help of the above part, compute the mean of $X(t)$ on $[0,1]$ over 200, 2000, 20000 trajectories of $X(t)$ with $\Delta t = 2^{-8}$. Can you guess (more or less) the value of $\mathbb{E}[X(1)]$?

(c) Finally, in the same figure, display 5 sample paths of $X(t)$ on $[0,1]$ together with the mean over 50000 trajectories of $X(t)$ for $\Delta t = 2^{-8}$.

Task 3: The main motivation for this task is to write a fast code for the simulation of a path of a Wiener process/Brownian motion $W(t)$ for $t \in [0,1]$.

To do this, we shall use the MATLAB function CUMSUM. Study this function, understand and complete the following piece of code. Compare with your result obtained in **Task 1**.

```
% Simulation of path of WP/BM
clear all
...
t = 0:dt:T; % discrete time grid
dW = ... ; % generate Wiener increments
W = [0 cumsum(dW)]; % Brownian path
% plot the Brownian path
...
```

Task 4: This task illustrates the numerical approximation by Euler–Maruyama's scheme.

Consider the SDE (for times $0 \le t \le T$)

$$\begin{aligned} \mathrm{d}X(t) &= \lambda X(t)\,\mathrm{d}t + \mu X(t)\,\mathrm{d}W(t) \\ X(0) &= X_0 \end{aligned}$$

with parameters $X_0 = 1, \lambda = 2, \mu = 1, T = 1$. For this particular problem, we know that the exact solution reads $X(t) = X_0 \exp\big((\lambda - \frac{1}{2}\mu^2)t + \mu W(t)\big)$. Compute a discretised Brownian motion over $[0,1]$ with a very small discretisation parameter $\delta t = 2^{-8}$. This is used to compute our "exact solution" **Xtrue**, also called a reference solution. Apply Euler–Maruyama's method with 3 different time steps $\Delta t = 2^4\delta t, 2^2\delta t$, and δt. Here, it is important to be on the same discretised Brownian path as the one used for generating the exact solution. In three different figures, display one realisation of the numerical solution together with the exact solution.

```
close all, clear all
rng(100)
% parameters
lambda=2; mu=1; X0=1;
Tend=1; N=2^8; dt=Tend/N;
dW=sqrt(dt)*randn(1,N);     % Brownian increments
% discretised Brownian path
W=...
% exact solution
Xtrue=X0*exp((lambda-0.5*mu^2)*([dt:dt:Tend])+mu*W);

%% First EM approx.
R=2^4; Dt=R*dt; L=N/R;    % L EM steps of size Dt = R*dt
Xem=zeros(1,L);       % preallocate for efficiency
for j=1:L
  Winc=sum(dW(R*(j-1)+1:R*j)); % Winc for EM on the same BM as above
  ... ; % iter. for EM scheme with step Dt
  Xem(j)= ... ; % store the result of EM
end
% first plot on first figure
...
figure()
plot([0:dt:Tend],[X0,Xtrue],'k-'), hold on
plot([0:Dt:Tend],[X0,Xem],'r--*'), hold off
...
```

The expression **Winc** in the code above is computed as follows

$$\text{Winc} = W(j\Delta t) - W((j-1)\Delta t) = W(jR\delta t) - W((j-1)R\delta t) = \sum_{k=jR-R+1}^{jR} \mathrm{d}W_k,$$

with $\mathrm{d}W_k = W(k\delta t) - W((k-1)\delta t)$ the original Wiener increments.

Task 5: This task asks you to confirm numerically the weak order of convergence of Euler–Maruyama's scheme.

Consider the SDE from the previous exercise with parameters $\lambda = 2, \mu = 0.1, X_0 = 1, T = T_N = 1$. Verify numerically that

$$e_{\Delta t}^{\text{Weak}} := \left|\mathbb{E}[X_N] - \mathbb{E}[X(t_N)]\right| \leq C\Delta t,$$

where, for the exact solution, one has $\mathbb{E}[X(t_N)] = \mathbb{E}[X(1)] = \mathrm{e}^{\lambda\cdot 1}$ (how do you get this?). In order to approximate expectations, you may use a Monte-Carlo approximation with $M_s = 50000$ realisations of the numerical solution.

In order to display the results in a loglog plot, one computes (for example) 5 different approximations with Euler–Maruyama's method with step sizes $\Delta t = 2^p\delta t$, where $\delta t = 2^{-10}$ and $p = 1,\ldots,5$.

```
...
Ms=50000; % number of paths sampled
Xem=zeros(5,1); % preallocate arrays containing num. sol.
for p=1:5 % take various Euler timesteps
  Dt=2^(p-10);L=T/Dt; % L Euler steps of size Dt
  Xtemp=X0*ones(Ms,1); % initial values
  for j=1:L
    Winc= ... ;
    Xtemp= ... ; % Ms EM approximation samples
  end
  Xem(p)= ... ; % mean over Ms samples of EM at time Tend
end
Xerr=abs( ... ); % weak errors
% loglog plots
Dtvals=2.^( ... ); % array of time steps
figure()
...
loglog(Dtvals,Xerr,'ks-',Dtvals,Dtvals.^1,'r--','LineWidth',2),
...
```

Task 6: This task asks you to confirm numerically the strong order of convergence of Euler–Maruyama's scheme.

Consider the SDE (geometric Brownian motion on $0 \leq t \leq T$)

$$\begin{aligned} \mathrm{d}X(t) &= \lambda X(t)\,\mathrm{d}t + \mu X(t)\,\mathrm{d}W(t) \\ X(0) &= X_0 \end{aligned}$$

with parameters $\lambda = 2, \mu = 1, X_0 = 1, T = T_N = 1$. Verify numerically that

$$e_{\Delta t}^{\text{strong}} := \mathbb{E}[|X_N - X(T_N)|] \leq C\Delta t^{1/2},$$

where $X(T_N)$ denotes the exact solution at time $T_N = 1$ and X_N the last step of Euler–Maruyama's method. In order to approximate the expectations, you may use $M_s = 1000$ samples.

In order to display the results in a loglog plot, one computes (for example) 5 different approximations with Euler–Maruyama's method with step sizes $\Delta t = 2^p \delta t$, where $\delta t = 2^{-10}$ and $p = 1, \ldots, 5$.

```
...
Xerr=zeros(Ms,5);  % preallocate array error
for s=1:Ms,  % sample over discrete BM
  ...
  W= ...;     % discrete BM
  Xtrue= ...; % exact sol.
  for p=1:5
    R=2^(p);Dt=R*dt;L=N/R; % L Euler steps of size Dt=R*dt
```

```
    XEM=Xzero;
    for j=1:L
      Winc= ...; % Wiener increm.
      XEM= ...; % EM scheme
    end
    Xerr(s,p)= ... ; % error at T=1
  end
end
% compute strong errors+plots
...
```

Some of the exercises are inspired by materials from D. Higham, A. Szepessy, .