Good.

Irzam Hardiansyah
irzam@kth.se

2. marts 2023
Friday

# FSF3581: Homework 2

# Problem 1

*Let $\overline{X}(T)$ be a forward Euler approximation of the solution to a stochastic differential equation*

$$dX(t) = a(t,X(t))dt + b(t,X(t))dW(t)$$
$$X(0) = X_0$$

*Write a computer program that compute the forward Euler approximation $\overline{X}$. Test numerically how the strong error*

$$\|X(T) - \overline{X}(T)\|_{L^2(\Omega)} = \sqrt{\mathbb{E}\left[(X(T) - \overline{X}(T))^2\right]}$$

*and the weak error*

$$\mathbb{E}\left[g(X(T))\right] - \mathbb{E}\left[g(\overline{X}(T))\right]$$

*depend on the time step $\Delta t$; i.e. what the convergence rate is. Try with functions $a$, $b$, and $g$ that satisfy the conditions in Theorems 3.1 and 5.8. Also, try a function $g$ that does not satisfy the conditions in Theorem 5.8. Can you still observe the same convergence rate?*

***Answer:*** Let our function and boundary (starting) value be:

$$dX(t) = 0.25X(t)dt - 0.5X(t)dW(t)$$
$$X(0) = 10$$

R

thus, $a(t,X(t)) = 0.25X(t)$ and $b(t,X(t)) = -0.5X(t)$, and we use the identity function as a $g_1$ satisfying conditions in Theorems 3.1 and 5.8,

$$g_1(X(T)) = X(T)$$

and the following non-differentiable function as a $g_2$ not satisfying conditions in Theorem 5.8,

$$g_2(X(T)) = |X(T) - X_0|$$

R

Then, we run $10^6$ simulations for each of 10 different step lengths $\Delta t = \frac{1}{2^n}$, $n = 1,\ldots,10$, from $t = 0$ to $t = 1 = T$ to obtain the error rates of both the strong and weak convergence for each step length, as given in the table below:

| $n$ | $\Delta t$ | $\sqrt{\Delta t}$ | Strong convergence | Weak convergence ($g_1$) | Weak convergence ($g_2$) |
|---|---|---|---|---|---|
| 1 | .500 | .707 | 1.9561 | -0.1816 | -0.2023 |
| 2 | .250 | .500 | 1.3518 | -0.0946 | -0.0705 |
| 3 | .125 | .354 | 0.9394 | -0.0486 | -0.0296 |
| 4 | .063 | .250 | 0.6533 | -0.0246 | -0.0142 |
| 5 | .031 | .177 | 0.4596 | -0.0124 | -0.0067 |
| 6 | .016 | .125 | 0.3233 | -0.0062 | -0.0033 |
| 7 | .008 | .088 | 0.2278 | -0.0032 | -0.0017 |
| 8 | .004 | .063 | 0.1607 | -0.0017 | -0.0010 |
| 9 | .002 | .044 | 0.1135 | -0.0009 | -0.0004 |
| 10 | .001 | .031 | 0.0803 | -0.0004 | -0.0002 |

Thus, it is clear that, R R OK.

$$\|X(T) - \overline{X}(T)\|_{L^2(\Omega)} = \sqrt{\mathbb{E}\left[(X(T) - \overline{X}(T))^2\right]} = \mathcal{O}(\sqrt{\Delta t})$$

and that for both $g_1$ and $g_2$,

R

$$\mathbb{E}\left[g(X(T))\right] - \mathbb{E}\left[g(\overline{X}(T)\right] = \mathcal{O}(\Delta t)$$

The MATLAB code for the simulation is given in Appendix.

# Problem 2a

*Consider the ordinary differential equation*

$$dX_t = AX_t dt$$

*where $X_t \in \mathbb{R}^2$ and the matrix $A$ has two eigenvalues $\lambda_1 = 1$ and $\lambda_2 = -10^5$. Then, the backward Euler method*

$$X(t_{n+1}) - X(t_n) = AX(t_{n+1})(t_{n+1} - t_n)$$

*is an efficient method to solve the problem. Why?*

***Answer:*** For the given matrix $A$,

$$\frac{|\operatorname{Re}(\overline{\lambda})|}{|\operatorname{Re}(\underline{\lambda})|} = \frac{|\operatorname{Re}(\lambda_2)|}{|\operatorname{Re}(\lambda_1)|} = 10^5 \gg 1$$

Thus, the problem represented by the matrix is a **stiff problem**. Because backward Euler method is both A-stable and L-stable, it allows us to take much larger step lengths without compromising the convergence of the solution; i.e. we are guaranteed to obtain the correct steady-state solution. Nonetheless, the step length will still be constrained by the accuracy requirement, as it has a global error of $\mathcal{O}(\Delta t)$.

Furthermore, because the general form of solution for this differential equation is:

$$X(t) = X_0(c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t})$$

the second term of it will decay to zero very quickly - theoretically it is possible to decay this term in a single step with backward Euler because $\lambda_2 \Delta t \to -\infty$ as $\Delta t \to -\infty$ - leaving us with only the first term. Assuming the first term is the function of interest at steady-state, by taking only a few large enough steps with backward Euler we can get a reasonable approximation of this steady-state behavior - although the accuracy of this approximation is still dictated by the actual step length.

R

# Problem 2b

*Formulate and motivate a backward Euler method for approximation of the Ito SDE*

$$dX_t = aX_t dt + bX_t dW_t$$

*where $a < 0$ and $b > 0$ are constants.*

***Answer:*** We will use the following backward Euler approximation of Ito SDE:

R

$$X_{n+1} - X_n = aX_{n+1}(t_{n+1} - t_n) + bX_n(W_{n+1} - W_n)$$

Then, by rearranging the terms, we can obtain the update rule for backward Euler approximation of Ito SDE:

$$\begin{aligned}
X_{n+1} &= X_n + aX_{n+1}\Delta t_n + bX_n \Delta W_n \\
(1 - a\Delta t_n)X_{n+1} &= (1 + b\Delta W_n)X_n \\
X_{n+1} &= \frac{(1 + b\Delta W_n)}{(1 - a\Delta t_n)} X_n \\
&= kX_n
\end{aligned}$$

As $a < 0$, then the multiplication factor $k < 1$ in expectation and asymptotically, because:

$$\mathbb{E}\left[\frac{(1 + b\Delta W_n)}{(1 - a\Delta t_n)}\right] = \frac{1}{(1 - a\Delta t_n)} + \frac{\overbrace{b\,\mathbb{E}\left[\Delta W_n\right]}^{=0}}{(1 - a\Delta t_n)} < 1$$

thus, guaranteeing its stability.

# Appendix: MATLAB Code for Problem 1

```
% Stong and weak convergence for the Euler method
% a = AX = 0.25X, b = BX = -0.5X, dX = 0.25Xdt - 0.5XdW;
% thus, A = 0.25, B = -0.5
% g1(X) = X = as a function satisfying Theorems 3.1 & 5.8
% g2(X) = abs(X - X0) as one which does not

steps = [1:10]; %number of different (equidistant) step lengths
sError = []; w1Error = []; w2Error = [];
for i=steps
    N = 2^i % number of timesteps
    randn('state',0);
    T = 1; dt = T/N; t = 0:dt:T;
    A = .25; B = -.5; X0 = 10;
    M = 1E6; % number of realisations
    X = X0*ones(M,1); % X(0) for all realizations
    W = zeros(M,1); % W(0) for all realizations
    for j=1:N
        dW = sqrt(dt)*randn(M,1); % Wiener increments
        X = X + X.*(A*dt+B*dW); % processes at next time step
        W = W + dW; % Brownian paths at next step
    end
    XT = X0*exp( (A-B^2/2)*T + B*W ); % exact final value
    sError(i) = sqrt(mean((X-XT).^2)); % strong error
    g1X = X; g1XT = XT; g2X = abs(X-X0); g2XT = abs(XT-X0);
    w1Error(i) = mean(g1X-g1XT); % weak error for g1
    w2Error(i) = mean(g2X-g2XT); % weak error for g2
end

dt = T./2.^steps;
loglog(dt,abs(w1Error),'o--',dt,abs(w2Error),'*--',dt,dt,'--',dt,abs(sError),'o-',dt,sqrt
legend('Weak convergence of g1(X)','Weak convergence of
    g2(X)','O(dt)','Strong convergence of
    X','O(sqrt(dt))','Location','southeast')

figure;
histogram(g1X)
figure;
histogram(g2X)
figure;
histogram(g1X-g1XT)
figure;
histogram(g2X-g2XT)
```