Chapter 22
Basic Numerical Methods

Numerical methods in finance include finite difference methods, and statistical and Monte Carlo methods for computation of option prices and hedging strategies. This chapter is a basic introduction to finite difference methods for the resolution of PDEs and stochastic differential equations. We cover the explicit and implicit finite difference schemes for the heat equations and the Black–Scholes PDE, as well as the Euler and Milstein schemes for stochastic differential equations.

22.1 Euler Discretization

In order to apply the Monte Carlo method in option pricing, we need to generate a sequence $(\widehat{X}_1, \ldots, \widehat{X}_N)$ of sample values of a random variable X, such that the empirical mean

$$\mathbb{E}[\phi(X)] \simeq \frac{\phi(\widehat{X}_1) + \cdots + \phi(\widehat{X}_N)}{N}$$

can be used according to the strong law of large number for the evaluation of the expected value $\mathbb{E}[\phi(X)]$. Despite its apparent simplicity, the Monte Carlo method can converge slowly. The optimization of Monte Carlo algorithms and of random number generators have been the object of numerous studies which are outside the scope of this text, see, *e.g.*, Glasserman (2004), Korn et al. (2010).

This version: May 2, 2026
https://personal.ntu.edu.sg/nprivault/indext.html

Random samples for the solution of a stochastic differential equation of the form

$$dX_t = b(X_t)dt + a(X_t)dW_t \tag{22.1}$$

where $(W_t)_{t\in\mathbb{R}_+}$ is a standard Brownian motion, can be generated by time discretization on $\{t_0, t_1, \ldots, t_N\}$. This can be applied in particular to option pricing with local volatility, see § 9.3.

More precisely, the Euler discretization scheme for the stochastic differential equation (22.1) is given by

$$\begin{aligned}\widehat{X}^N_{t_{k+1}} &= \widehat{X}^N_{t_k} + \int_{t_k}^{t_{k+1}} b(X_s)ds + \int_{t_k}^{t_{k+1}} a(X_s)dW_s \\ &\simeq \widehat{X}^N_{t_k} + b(\widehat{X}^N_{t_k})(t_{k+1}-t_k) + a(\widehat{X}^N_{t_k})(W_{t_{k+1}} - W_{t_k}),\end{aligned}$$

where $W_{t_{k+1}} - W_{t_k} \simeq \mathcal{N}(0, t_{k+1}-t_k)$, $k = 0, 1, \ldots, N-1$.

The next R code presents a numerical solution of the stochastic differential equation

$$dS_t = \mu S_t dt + \sigma S_t dB_t \tag{22.2}$$

which defines geometric Brownian motion $(S_t)_{t\in\mathbb{R}_+}$.

```
N=2000; t <- 0:N; dt <- 1.0/N;mu=0.5; sigma=0.2; nsim <- 10; X <- matrix(0, nsim, N+1)
dB <- matrix(rnorm(nsim*N,mean=0,sd=sqrt(dt)), nsim, N+1)
for (i in 1:nsim){X[i,1]=1.0;
for (j in 1:N+1){X[i,j]=X[i,j-1]+mu*X[i,j-1]*dt+sigma*X[i,j-1]*dB[i,j]}}
plot(t*dt, rep(0, N+1), xlab = "Time", ylab = "Geometric Brownian motion", lwd=2, ylim =
  c(min(X),max(X)), type = "l", col = 0,las=1, cex.axis=1.5,cex.lab=1.5, xaxs='i', yaxs='i')
for (i in 1:nsim){lines(t*dt, X[i, ], lwd=2, type = "l", col = i)}
```

Listing 22.1: R code - Euler discretization.

22.2 Milstein Discretization

In the Milstein scheme we use (22.1) to expand $a(X_s)$ as

$$\begin{aligned}a(X_s) &\simeq a(X_{t_k}) + a'(X_{t_k})(X_s - X_{t_k}) \\ &\simeq a(X_{t_k}) + a'(X_{t_k})\big(b(X_{t_k})(s-t_k) + a(X_{t_k})(W_s - W_{t_k})\big),\end{aligned}$$

$0 \leqslant t_k < s$. As a consequence, we get

$$\begin{aligned}\widehat{X}^N_{t_{k+1}} &= \widehat{X}^N_{t_k} + \int_{t_k}^{t_{k+1}} b(X_s)ds + \int_{t_k}^{t_{k+1}} a(X_s)dW_s \\ &\simeq \widehat{X}^N_{t_k} + \int_{t_k}^{t_{k+1}} b(X_s)ds + a(X_{t_k})(W_{t_{k+1}} - W_{t_k}) \\ &\quad + a'(X_{t_k})b(X_{t_k}) \int_{t_k}^{t_{k+1}} (s-t_k)dW_s \\ &\quad + a'(X_{t_k})a(X_{t_k}) \int_{t_k}^{t_{k+1}} (W_s - W_{t_k})dW_s\end{aligned}$$

$$\simeq \widehat{X}_{t_k}^N + \int_{t_k}^{t_{k+1}} b(X_s)ds + a(X_{t_k})(W_{t_{k+1}} - W_{t_k})$$
$$+a'(X_{t_k})a(X_{t_k}) \int_{t_k}^{t_{k+1}} (W_s - W_{t_k})dW_s.$$

Next, using Itô's formula we note that

$$(W_{t_{k+1}} - W_{t_k})^2 = 2\int_{t_k}^{t_{k+1}} (W_s - W_{t_k})dW_s + \int_{t_k}^{t_{k+1}} ds,$$

hence

$$\int_{t_k}^{t_{k+1}} (W_s - W_{t_k})dW_s = \frac{1}{2}((W_{t_{k+1}} - W_{t_k})^2 - (t_{k+1} - t_k)),$$

and

$$\begin{aligned}\widehat{X}_{t_{k+1}}^N &\simeq \widehat{X}_{t_k}^N + \int_{t_k}^{t_{k+1}} b(X_s)ds + a(X_{t_k})(W_{t_{k+1}} - W_{t_k}) \\ &\quad +\frac{1}{2}a'(X_{t_k})a(X_{t_k})((W_{t_{k+1}} - W_{t_k})^2 - (t_{k+1} - t_k)) \\ &\simeq \widehat{X}_{t_k}^N + b(X_{t_k})(t_{k+1} - t_k) + a(X_{t_k})(W_{t_{k+1}} - W_{t_k}) \\ &\quad +\frac{1}{2}a'(X_{t_k})a(X_{t_k})((W_{t_{k+1}} - W_{t_k})^2 - (t_{k+1} - t_k)).\end{aligned}$$

As a consequence, the Milstein scheme is written as

$$\begin{aligned}\widehat{X}_{t_{k+1}}^N &\simeq \widehat{X}_{t_k}^N + b(\widehat{X}_{t_k}^N)(t_{k+1} - t_k) + a(\widehat{X}_{t_k}^N)(W_{t_{k+1}} - W_{t_k}) \\ &\quad +\frac{1}{2}a'(\widehat{X}_{t_k}^N)a(\widehat{X}_{t_k}^N)((W_{t_{k+1}} - W_{t_k})^2 - (t_{k+1} - t_k)),\end{aligned}$$

i.e. in the Milstein scheme we take into account the "small" difference

$$(W_{t_{k+1}} - W_{t_k})^2 - (t_{k+1} - t_k)$$

existing between $(\Delta W_t)^2$ and Δt. Taking $(\Delta W_t)^2$ equal to Δt brings us back to the Euler scheme.

The next R code presents a numerical solution of (22.2) using the Milstein scheme.

```
N=2000; t <- 0:N; dt <- 1.0/N;mu=0.5; sigma=0.2; nsim <- 10; X <- matrix(0, nsim, N+1)
dB <- matrix(rnorm(nsim*N,mean=0,sd=sqrt(dt)), nsim, N+1)
for (i in 1:nsim){X[i,1]=1.0;
for (j in 1:N+1){X[i,j]=X[i,j-1] +mu*X[i,j-1]*dt +sigma*X[i,j-1]*dB[i,j]
 +0.5*sigma^2*X[i,j-1]*(dB[i,j]^2-dt)}}
plot(t*dt, rep(0, N+1), xlab = "Time", ylab = "Geometric Brownian motion", lwd=2, ylim =
 c(min(X),max(X)), type = "l", col = 0,las=1, cex.axis=1.5,cex.lab=1.5, xaxs='i', yaxs='i')
for (i in 1:nsim){lines(t*dt, X[i, ], lwd=2, type = "l", col = i)}
```

Listing 22.2: R code - Milstein discretization.

22.3 Discretized Heat Equation

Consider the heat equation

$$\frac{\partial \phi}{\partial t}(t,x) = \frac{\partial^2 \phi}{\partial x^2}(t,x) \tag{22.3}$$

with initial condition

$$\phi(0,x) = f(x)$$

on a compact time-space interval $[0,T] \times [0,X]$.

The intervals $[0,T]$ and $[0,X]$ are respectively discretized according to $\{t_0 = 0, t_1, \ldots, t_N = T\}$ and $\{x_0 = 0, x_1, \ldots, x_M = X\}$ with $\Delta t = T/N$ and $\Delta x = X/M$, from which we construct a grid

$$(t_i, x_j) = (i\Delta t, j\Delta x), \qquad i = 0, \ldots, N, \quad j = 0, \ldots, M,$$

on $[0,T] \times [0,X]$.

Our goal is to solve the heat equation (22.3) with *initial* condition $\phi(0,x)$, $x \in [0,X]$, and lateral boundary conditions $\phi(t,0)$, $\phi(t,X)$, $t \in [0,T]$, via a discrete approximation

$$(\phi(t_i, x_j))_{0 \leqslant i \leqslant N, \ 0 \leqslant j \leqslant M}$$

of the solution to (22.3), by evaluating derivatives using finite differences.

Explicit scheme

Using the *forward* time difference approximation

$$\frac{\partial \phi}{\partial t}(t_i, x) \simeq \frac{\phi(t_{i+1}, x_j) - \phi(t_i, x_j)}{\Delta t}$$

of the time derivative, and the related space difference approximations

$$\frac{\partial \phi}{\partial x}(t, x_j) \simeq \frac{\phi(t, x_j) - \phi(t_i, x_{j-1})}{\Delta x}, \quad \frac{\partial \phi}{\partial x}(t, x_{j+1}) \simeq \frac{\phi(t, x_{j+1}) - \phi(t_i, x_j))}{\Delta x}$$

and

$$\frac{\partial^2 \phi}{\partial x^2}(t, x_j) \simeq \frac{1}{\Delta x}\left(\frac{\partial \phi}{\partial x}(t, x_{j+1}) - \frac{\partial \phi}{\partial x}(t, x_j)\right) = \frac{\phi(t_i, x_{j+1}) + \phi(t_i, x_{j-1}) - 2\phi(t_i, x_j)}{(\Delta x)^2}$$

of the time and space derivatives, we discretize (22.3) as

$$\frac{\phi(t_{i+1}, x_j) - \phi(t_i, x_j)}{\Delta t} = \frac{\phi(t_i, x_{j+1}) + \phi(t_i, x_{j-1}) - 2\phi(t_i, x_j)}{(\Delta x)^2}. \tag{22.4}$$

Letting $\rho = (\Delta t)/(\Delta x)^2$, this yields

$$\phi(t_{i+1}, x_j) = \rho\phi(t_i, x_{j+1}) + (1-2\rho)\phi(t_i, x_j) + \rho\phi(t_i, x_{j-1}),$$

$1 \leqslant j \leqslant M-1$, $1 \leqslant i \leqslant N$, *i.e.*

$$\Phi_{i+1} = A\Phi_i + \rho \begin{bmatrix} \phi(t_i, x_0) \\ 0 \\ \vdots \\ 0 \\ \phi(t_i, x_M) \end{bmatrix}, \qquad i = 0, 1, \ldots, N-1, \tag{22.5}$$

with

$$\Phi_i = \begin{bmatrix} \phi(t_i, x_1) \\ \vdots \\ \phi(t_i, x_{M-1}) \end{bmatrix}, \qquad i = 0, 1, \ldots, N,$$

and

$$A = \begin{bmatrix} 1-2\rho & \rho & 0 & \cdots & 0 & 0 & 0 \\ \rho & 1-2\rho & \rho & \cdots & 0 & 0 & 0 \\ 0 & \rho & 1-2\rho & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-2\rho & \rho & 0 \\ 0 & 0 & 0 & \cdots & \rho & 1-2\rho & \rho \\ 0 & 0 & 0 & \cdots & 0 & \rho & 1-2\rho \end{bmatrix}.$$

The vector

$$\begin{bmatrix} \phi(t_i, x_0) \\ 0 \\ \vdots \\ 0 \\ \phi(t_i, x_M) \end{bmatrix} = \begin{bmatrix} \phi(t_i, 0) \\ 0 \\ \vdots \\ 0 \\ \phi(t_i, X) \end{bmatrix}, \qquad i = 0, 1, \ldots, N,$$

in (22.5) can be given by the lateral boundary conditions $\phi(t, 0)$ and $\phi(t, X)$. From those boundary conditions and the initial data of

$$\Phi_0 = \begin{bmatrix} \phi(0, x_0) \\ \phi(0, x_1) \\ \vdots \\ \phi(0, x_{M-1}) \\ \phi(0, x_M) \end{bmatrix}$$

we can apply (22.5) in order to solve (22.4) recursively for $\Phi_1, \Phi_2, \Phi_3, \ldots$, see also Figure 22.1.

Implicit scheme

Using the *backward* time difference approximation

$$\frac{\partial \phi}{\partial t}(t_i, x) \simeq \frac{\phi(t_i, x_j) - \phi(t_{i-1}, x_j)}{\Delta t}$$

of the time derivative, we discretize (22.3) as

$$\frac{\phi(t_i, x_j) - \phi(t_{i-1}, x_j)}{\Delta t} = \frac{\phi(t_i, x_{j+1}) + \phi(t_i, x_{j-1}) - 2\phi(t_i, x_j)}{(\Delta x)^2} \tag{22.6}$$

and letting $\rho = (\Delta t)/(\Delta x)^2$ we get

$$\phi(t_{i-1}, x_j) = -\rho\phi(t_i, x_{j+1}) + (1 + 2\rho)\phi(t_i, x_j) - \rho\phi(t_i, x_{j-1}),$$

$1 \leqslant j \leqslant M - 1$, $1 \leqslant i \leqslant N$, *i.e.*

$$\mathbf{\Phi}_{i-1} = B\mathbf{\Phi}_i + \rho \begin{bmatrix} \phi(t_i, x_0) \\ 0 \\ \vdots \\ 0 \\ \phi(t_i, x_M) \end{bmatrix}, \qquad i = 1, 2, \ldots, N,$$

with

$$B = \begin{bmatrix} 1+2\rho & -\rho & 0 & \cdots & 0 & 0 & 0 \\ -\rho & 1+2\rho & -\rho & \cdots & 0 & 0 & 0 \\ 0 & -\rho & 1+2\rho & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1+2\rho & -\rho & 0 \\ 0 & 0 & 0 & \cdots & -\rho & 1+2\rho & -\rho \\ 0 & 0 & 0 & \cdots & 0 & -\rho & 1+2\rho \end{bmatrix}.$$

By inversion of the matrix B, $\mathbf{\Phi}_i$ is given in terms of $\mathbf{\Phi}_{i-1}$ as

$$\mathbf{\Phi}_i = B^{-1}\mathbf{\Phi}_{i-1} - \rho B^{-1} \begin{bmatrix} \phi(t_i, x_0) \\ 0 \\ \vdots \\ 0 \\ \phi(t_i, x_M) \end{bmatrix}, \qquad i = 1, \ldots, N,$$

which also allows for a recursive solution of (22.6), see also Figure 22.2.

This version: May 2, 2026
https://personal.ntu.edu.sg/nprivault/indext.html

22.4 Discretized Black–Scholes PDE

Consider the Black–Scholes PDE

$$r\phi(t,x) = \frac{\partial \phi}{\partial t}(t,x) + rx\frac{\partial \phi}{\partial x}(t,x) + \frac{1}{2}x^2\sigma^2\frac{\partial^2 \phi}{\partial x^2}(t,x), \qquad (22.7)$$

under the terminal condition $\phi(T,x) = (x-K)^+$, *resp.* $\phi(T,x) = (K-x)^+$, for a European call, *resp.* put, option. The constant volatility coefficient σ may also be replaced with a function $\sigma(t,x)$ of the underlying asset price, in the case local volatility models.

Note that in the solution of the Black–Scholes PDE, time is run *backwards* as we start from a terminal condition $\phi(T,x)$ at time T. Thus here the explicit scheme uses *backward* differences while the implicit scheme uses *forward* differences.

Explicit scheme

Using here the *backward* time difference approximation

$$\frac{\partial \phi}{\partial t}(t_i,x) \simeq \frac{\phi(t_i,x_j) - \phi(t_{i-1},x_j)}{\Delta t}$$

of the time derivative, we discretize (22.7) as

$$\begin{aligned} r\phi(t_i,x_j) = {} & \frac{\phi(t_i,x_j) - \phi(t_{i-1},x_j)}{\Delta t} + rx_j\frac{\phi(t_i,x_{j+1}) - \phi(t_i,x_{j-1})}{2\Delta x} \\ & + \frac{1}{2}x_j^2\sigma^2\frac{\phi(t_i,x_{j+1}) + \phi(t_i,x_{j-1}) - 2\phi(t_i,x_j)}{(\Delta x)^2}, \end{aligned} \qquad (22.8)$$

$1 \leqslant j \leqslant M-1$, $0 \leqslant i \leqslant N-1$, *i.e.*

$$\begin{aligned} \phi(t_{i-1},x_j) = {} & \frac{1}{2}(\sigma^2 j^2 - rj)\phi(t_i,x_{j-1})\Delta t + \phi(t_i,x_j)(1 - (\sigma^2 j^2 + r)\Delta t) \\ & + \frac{1}{2}(\sigma^2 j^2 + rj)\phi(t_i,x_{j+1})\Delta t, \end{aligned}$$

$1 \leqslant j \leqslant M-1$, where the lateral boundary conditions $\phi(t_i,0)$ and $\phi(t_i,x_M)$ are (approximately) given as follows.

European call options. We take the lateral boundary conditions

$$\phi(t_i,x_0) = 0, \quad \text{and} \quad \phi(t_i,x_M) \simeq \left(x_M - K\,\mathrm{e}^{-r(T-t_i)}\right)^+ = x_M - K\,\mathrm{e}^{-r(T-t_i)},$$

$i = 0,1,\ldots,N$, provided that x_M is sufficiently large.

European put options. We take the lateral boundary conditions

$$\phi(t_i, x_0) \simeq \left(K\,\mathrm{e}^{-(T-t_i)r} - x_0\right)^+ = K\,\mathrm{e}^{-(T-t_i)r}, \quad \text{and} \quad \phi(t_i, x_M) = 0,$$

$i = 0, 1, \ldots, N$, with here $x_0 = 0$.

Given a terminal condition of the form

$$\phi(T, x_j) = (x_j - K)^+, \text{ resp. } \phi(T, x_j) = (K - x_j)^+, \quad j = 1, \ldots, M-1,$$

this allows us to solve (22.8) successively for

$$\phi(t_{N-1}, x_j),\ \phi(t_{N-2}, x_j),\ \phi(t_{N-3}, x_j), \ldots, \phi(t_1, x_j),\ \phi(t_0, x_j).$$

The explicit finite difference method is nevertheless known to have a divergent behaviour as time is run backwards, as illustrated in Figure 22.1.

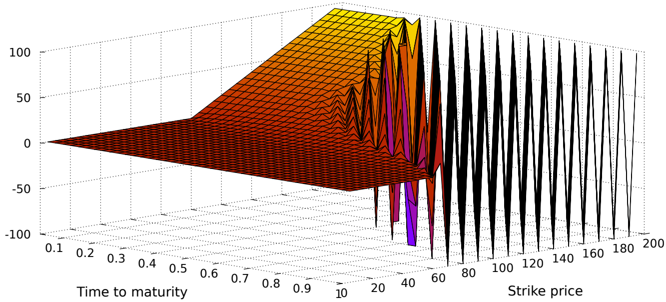

Fig. 22.1: Divergence of the explicit finite difference method.

Implicit scheme

Using the *forward* time difference approximation

$$\frac{\partial \phi}{\partial t}(t_i, x) \simeq \frac{\phi(t_{i+1}, x_j) - \phi(t_i, x_j)}{\Delta t}$$

of the time derivative, we discretize (22.7) as

$$\begin{aligned} r\phi(t_i, x_j) = {} & \frac{\phi(t_{i+1}, x_j) - \phi(t_i, x_j)}{\Delta t} + r x_j \frac{\phi(t_i, x_{j+1}) - \phi(t_i, x_{j-1})}{\Delta x} \\ & + \frac{1}{2} x_j^2 \sigma^2 \frac{\phi(t_i, x_{j+1}) + \phi(t_i, x_{j-1}) - 2\phi(t_i, x_j)}{(\Delta x)^2}, \end{aligned} \tag{22.9}$$

$1 \leqslant j \leqslant M-1$, $0 \leqslant i \leqslant N-1$, *i.e.*

This version: May 2, 2026
https://personal.ntu.edu.sg/nprivault/indext.html

$$\phi(t_{i+1}, x_j) = -\frac{1}{2}(\sigma^2 j^2 - rj)\phi(t_i, x_{j-1})\Delta t + \phi(t_i, x_j)(1 + (\sigma^2 j^2 + r)\Delta t)$$
$$-\frac{1}{2}(\sigma^2 j^2 + rj)\phi(t_i, x_{j+1})\Delta t,$$

$1 \leqslant j \leqslant M-1$, *i.e.*

$$\Phi_{i+1} = B\Phi_i + \begin{bmatrix} \frac{1}{2}\left(r - \sigma^2\right)\phi(t_i, x_0)\Delta t \\ 0 \\ \vdots \\ 0 \\ -\frac{1}{2}\left(r(M-1) + (M-1)^2\sigma^2\right)\phi(t_i, x_M)\Delta t \end{bmatrix}, \qquad (22.10)$$

$i = 0, 1, \ldots, N-1$, with

$$B_{j,j-1} = \frac{1}{2}\left(rj - \sigma^2 j^2\right)\Delta t, \quad B_{j,j} = 1 + \sigma^2 j^2 \Delta t + r\Delta t,$$

and

$$B_{j,j+1} = -\frac{1}{2}\left(rj + \sigma^2 j^2\right)\Delta t,$$

for $j = 1, 2, \ldots, M-1$, and $B(i, j) = 0$ otherwise.

By inversion of the matrix B, Φ_i is given in terms of Φ_{i+1} as

$$\Phi_i = B^{-1}\Phi_{i+1} - B^{-1} \begin{bmatrix} \frac{1}{2}\left(r - \sigma^2\right)\phi(t_i, x_0)\Delta t \\ 0 \\ \vdots \\ 0 \\ -\frac{1}{2}\left(r(M-1) + (M-1)^2\sigma^2\right)\phi(t_i, x_M)\Delta t \end{bmatrix},$$

$i = 0, 1, \ldots, N-1$, where the lateral boundary conditions $\phi(t_i, x_0)$ and $\phi(t_i, x_M)$ can be provided as in the case of the explicit scheme, allowing us to solve (22.9) recursively for $\phi(t_{N-1}, x_j), \phi(t_{N-2}, x_j), \phi(t_{N-3}, x_j), \ldots$

The implicit finite difference method is known to be more stable than the explicit scheme, as illustrated in Figure 22.2, in which the discretization parameters have been taken to be the same as in Figure 22.1.

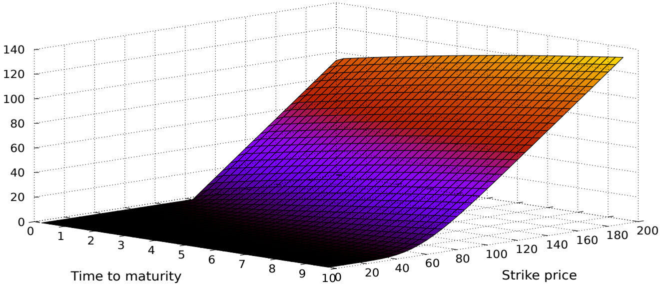

Fig. 22.2: Stability of the implicit finite difference method.

Exercises

Exercise 22.1 Show that when the terminal condition is a constant $\phi(T,x) = c > 0$ the implicit scheme (22.10) recovers the known solution $\phi(s,x) = c\,\mathrm{e}^{-r(T-s)}$, $s \in [0,T]$.

Exercise 22.2 Let S_t be the geometric Brownian motion given by the stochastic differential equation

$$dS_t = rS_t dt + \sigma S_t dW_t.$$

a) Compute the Euler discretization $\big(\widehat{S}^N_{t_k}\big)_{k=0,1,\ldots,N}$ of $(S_t)_{t\in\mathbb{R}_+}$.
b) Compute the Milstein discretization $\big(\widehat{S}^N_{t_k}\big)_{k=0,1,\ldots,N}$ of $(S_t)_{t\in\mathbb{R}_+}$.