Numerical Methods for Financial Mathematics. **Exercise Handout 7**
Lecture: Prof. Dr. Christian Fries, Exercises: Dr. Andrea Mazzon, Tutorium: Niklas Weber
Sommersemester 2023

Exercise 1

Write a class `RandomVariableFromArray` which wraps a one-dimensional array of `doubles` in an object representing a random variable: the one-dimensional array represents the simulated realizations of the random variable. For example, an object of such a class may represent the realizations of a stochastic process at a given time.

This class must have methods to:

- Return an array of `doubles` containing all the realizations of the random variable representing the object calling the method.
- Return a `double` representing a single realization of the random variable corresponding to a given `int` simulation index.
- Compute and return the average and the standard deviation of the realizations of the random variable represented by the object calling the method.
- Compute and return the sum, the product, the difference and the ratio of the random variable represented by the object calling the method with the one represented by another object of type `RandomVariableFromArray` given as an argument. The result must be given as another object of type `RandomVariableFromArray`.
- Compute and return the sum, the product, the difference and the ratio of the random variable represented by the object calling the method with a `double` argument. The result must be given as another object of type `RandomVariableFromArray`.

Hint: the array storing the realizations might conveniently be a field of the class, initialized in the constructor. Then, all the operations are performed on that array (possibly involving the array of the other random variable). For that scope, you can use the methods you find in

`com.andreamazzon.usefulmethodsmatricesandvectors.UsefulMethodsMatricesAndVectors`.

Exercise 2

Write a class `BrownianMotion` giving a first, basic implementation of the discretization and the simulation of a Brownian motion. Here you can suppose that the size of the time steps of the time discretization is constant. The Brownian motion has to be represented by a one-dimensional array of objects of type `RandomVariableFromArray`.

The class has to have fields representing the number of simulated trajectories, the initial time, the number of the time steps, the size of the time steps and the final time, and methods to return:

- The whole trajectory of the Brownian motion, as an array of `RandomVariableFromArrays`.
- The realizations of the Brownian motion at a given `int` time index, as a `RandomVariableFromArray` object.
- A single trajectory of the Brownian motion corresponding to a given `int` simulation index, as an array of `doubles`.
- A single realization of the Brownian motion at a given `int` time index and corresponding to a given `int` simulation index, as a single `double`.

Hint: the main task here is to fill the array of `RandomVariableFromArrays` representing the Brownian motion, i.e., to generate the Brownian motion. In order to do that, you can exploit the fact that the increments of the Brownian motion are random variables $\mathcal{N}(0, \Delta)$, where Δ is the time step of the time discretization, and then construct the Brownian motion as the sum of the increments.

Exercise 3

Write a class where you create an object of type `BrownianMotion` and you do some tests, calling the suitable methods of `RandomVariableFromArray` and `BrownianMotion`. For example, you can check the evolution of the average and the standard deviation of the realizations of your Brownian motion. You can also prove analytically, and then check numerically, that

$$\mathbb{E}\left[W_t W_s\right] = \min\{s, t\}$$

for $s, t > 0$.