

Geometriai Brown-mozgás és pénzügyi modellezés

„A kutatómunka információs eszközei”
tárgyra készült hallgatói dokumentáció

Írta: Bóna Márton
ver.: 2018.04.17.



Kivonat

Ezen dokumentum „A kutatómunka információs eszközei” tárgyra készített programozási eszközök megismerése céljából létrehozott beadandó dokumentációja.

A dokumentum röviden összefoglalja a tárgy elvárását a beadandóval kapcsolatban, továbbá ismerteti a kiválasztott probléma alapjait. A feladatot párfős csapatoknak kell megoldaniuk, így szó fog esik a csapatmunkáról és a feladatok szétosztásáról. Továbbiakban a saját feladatrészem munkafolyamatát és a fejlesztés során adódó problémák megoldásáról fogok írni. Végezetül bemutatásra kerül a végeredmény.

Tartalomjegyzék

1. Bevezetés	2
2. A feladat	3
2.1. A kitűzött feladat	3
2.2. A feladatok felosztás	3
3. Csapatmunka	3
4. Saját feladatréssz	4
4.1. Alapgondolatok	4
4.2. A kód elkészítése	4
4.3. Automatizálás és dokumentáció	5
5. Összefoglalás	5
Köszönetnyilvánítás	6
Irodalomjegyzék	7

Ábrák jegyzéke

1.0.1.GBM adott σ és öt különböző μ értékkel	2
---	---

1. Bevezetés

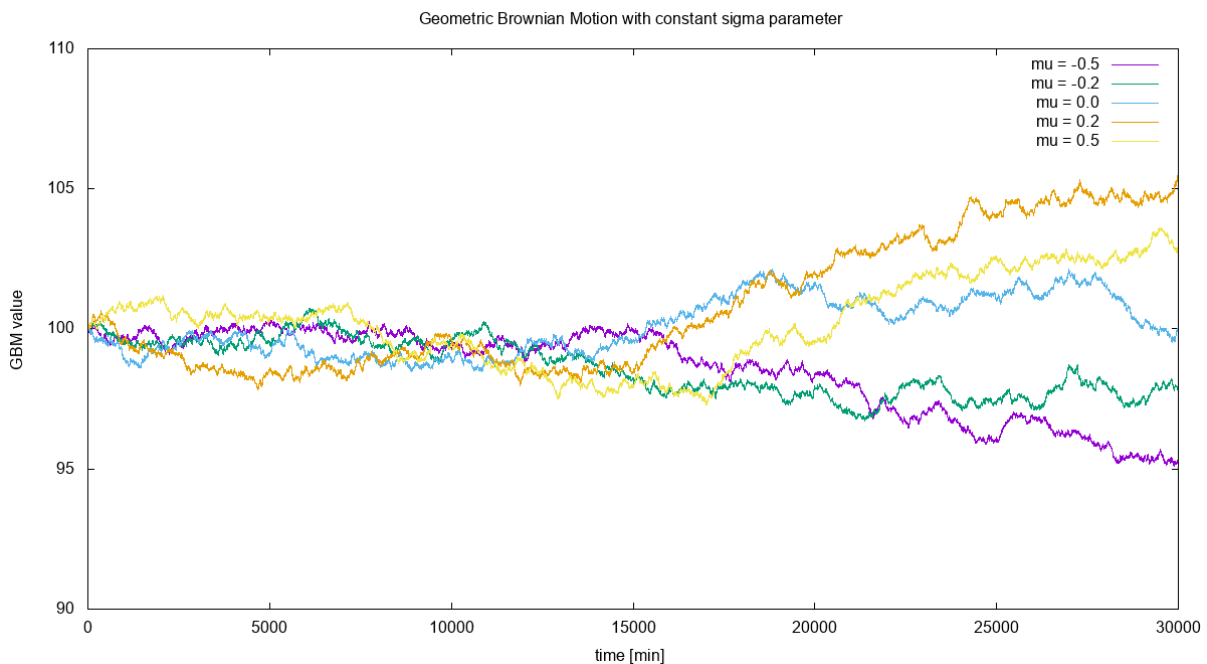
A pénzügyi modellezésben használt elméletek tesztelésére, továbbá különböző gazdasági szituációk előállítására régóta használnak sztochasztikus folyamaton alapuló árfolyammodelleket.

Ezen árfolyammodellek lényege, hogy minden lépésben egy valószínűségi változótól függ, hogy milyen irányba és mekkorát változik az árfolyam, ezáltal lehetőség nyílik arra, hogy sok adaton tudjuk tesztelni elméleteinket. Az egyik használható sztochasztikus folyamat a geometriai Brown-mozgás. A geometriai Brown-mozgást a következő sztochasztikus differenciálegyenlet írja le.

$$dS(t) = \mu S(t)dt + \sigma S(t)dB(t) \quad (1.1)$$

Ahol $S(t)$ jelöli például egy részvény árát t időpillanatban, μ az úgynevezett *drift* paraméter, σ pedig a *volatilitás* paraméter. A drift alatt azt kell érteni, hogy hosszú távon az árfolyam milyen irányba fog változni, volatilitás alatt pedig az árfolyam rövidtávú változékonyságát kell érteni, nagy volatilitás esetén az árfolyam rövid idő alatt nagy kilengéseket produkálhat.

$B(t)$ -ről még nem beszéltünk, ez szintén egy sztochasztikus folyamat értéke adott időpillanatban, ez a folyamat pedig nem más mint a Wiener-folyamat, más néven standard Brown-mozgás. A Wiener-folyamat ismerve, hogy értékének két időpillanat közötti változásai függetlenek egymástól és normális eloszlást követnek nulla várható értékkel és dt szórásnégyzettel.



1.0.1. ábra. GBM adott σ és öt különböző μ értékkel

Az 1.0.1 ábrán jól látható, hogy a drift paraméter tényleg befolyással van a hosszútávú árfolyam alakulásra. Az említett sztochasztikus folyamatokról bővebben [1] és [2] Wikipédia oldalakon lehet olvasni.

2. A feladat

2.1. A kitűzött feladat

A feladat az volt, hogy keressünk egy témát, amelyhez kapcsolhatók programozási feladatok, majd osszuk fel egymás között a választott feladatokat. Mindenkinek implementálni kell egy programot, ami végrehajtja a kitűzött elvárást és ezen programokat egy megfelelő struktúrába rendezve létre kell hozni egy automatikus fordítási környezetet. Azaz a programhoz kapcsolható bemeneti és kimeneti fájlokat, generált képeket és a dokumentáció függőségeit rögzíteni kell, hogy egyszerű parancs kiadásával minden legyártódjon. Majd végül a feladatot és a tapasztalatokat rögzíteni kell a jelen dokumentációban.

Megoldandó feladatnak egy pénzügyi programcsomag összeállítását választottuk. Létre kellett hozni egy programot ami képes árfolyamot generálni, majd ezeken az adatokon alapvető gazdasági indikátorok segítségével pénzügyi kereskedést valósít meg. Végül a kereskedés eredményét és a legyártott árfolyamokat ábrázolni kell.

2.2. A feladatok felosztás

A csapat négy tagú volt és különböző programozási nyelvhez értettünk. Ezért úgy döntöttünk, hogy két nyelvet használunk. Ketten közülünk python nyelven dolgoztak, az egyikük a feladat kereskedés részét valósította meg, a másikuk pedig az eredmények és az árfolyamok ábrázolását. A másik kétfős egység, amiben én voltam, c++ nyelvet használt és két sztochasztikus folyamaton alapuló árfolyammodellt hozott létre, melyből az adatokat a másik kétfős csapat használta fel. A két árfolyammodell a Wiener-folyamaton és a geometriai Brown-mozgáson alapul.

Az én feladatom volt a geometriai Brown-mozgás implementálása c++ nyelven. A továbbiakban erről írok majd, de előbb még szót ejtek hogyan valósult meg a csapatmunka.

3. Csapatmunka

A csapat vezetését én vállaltam el. Kezdetben több személyes találkozón megbeszéltük, hogy ki milyen nyelven tud programozni, és ennek függvényében választottunk egy feladatot, amely felosztható megfelelő alegységekre. Ezután tisztáztuk, hogy pontosan kinek mi a feladata és a feladatrészek hogyan függenek egymástól. Megbeszéltük, hogy milyen kommunikációs csatornán folytatjuk a további egyeztetéseket. A feladat verziókövetését és a csapatmunkát a git (github) környezet segítségével oldottuk meg. A projekt github elérhetősége [3].

A csoportvezetés során a github környezet felállítását, az alap könyvtárszerkezetet és a munkafolyamtok követését én végeztem. A CMake fájlok inicializálását szintén én végeztem.

Továbbá a csapat számár egy github bevezető jegyzetet és gyakorlófeladatot is készítettem, hogy a szintén kezdő csapatársak könnyebben el tudják sajátítani a munkához szükséges ismereteket.

4. Saját feladatrész

4.1. Alapgondolatok

Miután közösen megbeszéltük kinek mi a feladata és megértettük a git alapjait, nekiálltam a saját alfeladatomnak.

A feladatom a geometriai Brown-mozgás implementálása c++ programozási nyelven, továbbá adatok (árfolyamok) gyártása a többiek számára feldolgozáshoz. Az előbbiből következik, hogy egy jól általánosított programot érdemes készíteni megfelelő beolvasással felhasználók számára, így bárki képes további adatsorokat készíteni a program segítségével. Mások számára a könnyebb használhatóság szempontjából a git projektben *help.txt* fájlok találhatóak, amelyek a legfontosabb információkat tartalmazzák.

4.2. A kód elkészítése

A program elkészítéséhez szükség volt megfelelő véletlen szám generátor létrehozására, ami képes egy adott eloszlásnak megfelelő véletlen számokat generálni, ugyanis ezen alapul az egész feladat. Itt volt egy kisebb probléma, belefutottam egy olyan implementációba, ami nem minden fordító környezet esetében működik. A probléma az volt, hogy a véletlen szám generátor inicializálásához szükséges inputot egy nem-determinisztikus eljárás hozta volna létre (*random_device*), de bizonyos fordítókörnyezetben előfordul, hogy állandóan nullával tér vissza, ami mindig ugyan azt a véletlen szám sorozatot eredményezné, ezért nem ezt az eljárást használtam.

Az előbbieket után a geometriai Brown-mozgás elméleti alapjainak és annak gyakorlati alkalmazásnak kellett utánanézniem. Miután megértettem milyen paraméterei vannak a folyamatnak és hogyan lehet implementálni egy algoritmust ami előállítja a megfelelő kimenetet nekikezdtem a kód megírásának.

A kész programot tesztelni kezdtem és különböző források eredményeihez hasonlítva azt tapasztaltam, hogy valami nem megfelelő, mert a kimenet drámaian eltért. További utánanézés után sikerült kideríteni a hiba okát. A gyakorlati alkalmazásban van egy rögzített időegység, ami az egy év. Az idő lépésközét (dt) ehhez képest kell kiszámolni, például egy hónap az év egy tizenketted része, így $dt = 1/12$. Ezen standard rögzítése azért fontos jelen esetben, hogy azonos paraméter értékek mellett hasonló viselkedést mutasson az általánosságban használt megvalósításokhoz képest. Ezt követően a program az elvárt kimenetet produkálta.

Következő lépésként az adatok szerkezetét és tárolását kellett szabványosítani. Az adatfájlok elnevezése tartalmazza a legfontosabb ismérveket, ezért könnyebb eligazodni közöttük és a későbbiekben szükséges automatizálás is megköveteli.

A fejlesztés során a c++ kód elkészítéshez *Code::Blocks* fejlesztőkörnyezetet használtam, az automatizálást *Visual Studio Code* környezetben, *CMake* segítségével, a verziókövetés *Git* használatával történt.

4.3. Automatizálás és dokumentáció

A feladat részét képezte egy automatizált összeállítás létrehozás és jelen dokumentáció elkészítése. Az automatizálást a CMake segítségével lett megvalósítva. A működéshez A különböző fájlok és forráskódok egymáson való függőségét kell rögzíteni, így ha egy módosítás történik egy adatfájlban, minden ami azt felhasználja újragenerálódik automatikusan és nem nekünk kell kézzel újrafuttatni a programot, ábrázolni az eredményeket és újraserkeszteni a dokumentációt.

A feladatrészemben mindent sikerült megfelelően automatizálni. A program bemeneti konfigurációs fájlokat vár be, ami alapján elkészíti a kimeneti adatfájlokat. Az adatfájlokból a Gnuplot ábrázolóprogram meghívásának segítségével létrejön egy kép, amely bekerül jelen dokumentációba. Majd az összeállító rendszer lefordítja a jelen dokumentáció latex fájlját és elkészül belőle a végleges pdf dokumentáció. A folyamat minden függősége rögzítve van és módosítás esetén minden szükséges lépés újra lejátsszódik és elkészülnek a megfelelő kimenetek.

5. Összefoglalás

Végeredményben született egy, a geometriai Brown-mozgáson alapuló árfolyam generáló program és a munkafolyamatot leíró dokumentáció.

A munkám során sikerült megérteni a használt szoftverek és fejlesztői környezetek alapjait, továbbá kellő gyakorlatot szereztem bennük, így későbbi feladatoknál alkalmazni tudom az itt tanultakat.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani a tárgy oktatóinak a feladat elvégzéséhez szükséges szoftverek bemutatásáért és a félév során nyújtott segítségükért.

Hivatkozások

- [1] Wiener-folyamat: <https://hu.wikipedia.org/wiki/Wiener-folyamat>
- [2] Geometriai Brown-mozgás: https://en.wikipedia.org/wiki/Geometric_Brownian_motion
- [3] Github projekt: http://github.com/bonamarton/algo_trad_rand_walk