

Szemcsenövedés statisztikája

FIZIKA KAR

**BABES-BOLYAI
TUDOMÁNYEGYETEM**



Boda Szilárd, Babeş-Bolyai Tudományegyetem, Fizika kar, számítógépes fizika mesteri, I év

Vezető Tanár: Dr. Néda Zoltán, egyetemi tanár, Babeş-Bolyai Tudományegyetem, Számítógépes és Elméleti Fizika Tanszék

XII. Erdélyi Tudományos Diákköri Konferencia –
Kolozsvár, 2009. május 15–17.

Kivonat

Voronoi-típusú cellák nagyság-szerinti eloszlását vizsgáljuk Monte Carlo szimulációkkal. Olyan növekedési folyamatokat szimulálunk, amelyekben véletlenszerűen lerakott nukleációs központokból homogén és izotróp növekedés indul meg, egészen az összenövési pontokig. Az egyszerű Poisson típusú Voronoi cellákhoz képest a lényeges különbség az, hogy a nukleációs pontok nem feltétlenül egyszerre jelennek meg, hanem konstans rátákkal nukleálódnak. A feladatot egy- két- és háromdimenziós térben tanulmányozzuk. A kialakuló szemcsék nagyság-szerinti eloszlására az egyszerű Poisson-Voronoi eloszlástól különböző eloszlásfüggvényeket kapunk. Az eloszlás sűrűségfüggvény azonban látszólag független a nukleálási rátáktól.

Tartalomjegyzék

1. Bevezetés.....	4
2. Poisson-Voronoi cellák.....	5
3. Az egydimenziós eset egzakt tárgyalása.....	7
4. Voronoi cellák nukleációs folyamatokra.....	10
5. A kétdimenziós eset.....	11
6. A háromdimenziós eset.....	14
7. Következtetések.....	17
8. Könyvészet.....	18
9. Mellékletek.....	19

1. Bevezetés

A polikristályos strukturával rendelkező anyagokban a kristályos doméniumok növekedése során cella-szerű képződményeket kapunk. Ezeket a cella-szerű képződményeket nevezhetjük egyszerűen szemcséknek is. Ebben a dolgozatban ezen szemcséknek a nagyság szerinti eloszlását vizsgáltuk, amit először egy matematikai absztrakció, a Poisson-Voronoi cellák segítségével próbáltunk meg leírni, később pedig egy másik modellel, a folytonos nukleációval. Először lássuk mik is azok a Voronoi cellák.

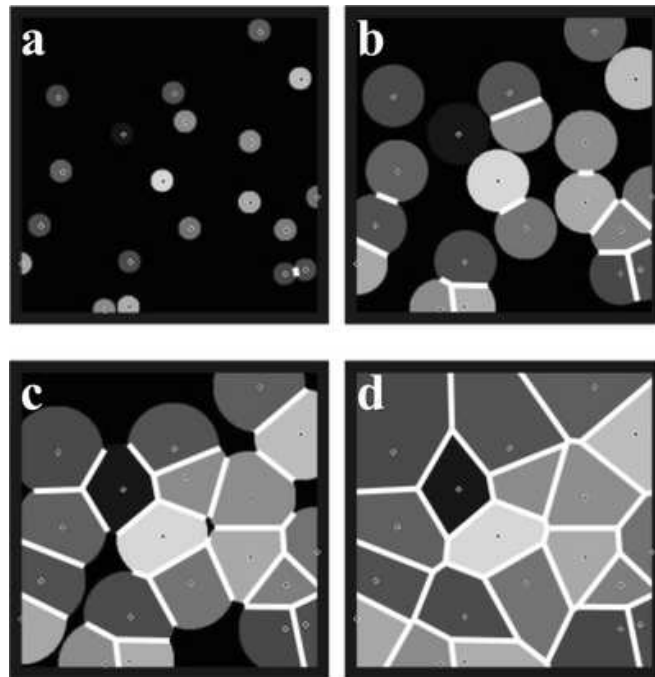
A Voronoi cellák^[2] bizonyos értelemben egy térfelosztást adnak, ahol adva van egy rögzített ponthalmaz és a teret ezeknek a befolyási övezete alapján osszuk fel. Minden Voronoi cella azokat a pontokat tartalmazza, amelyek a központjához közelebb vannak, mint bármely más központhoz. Az 1-es és 2-es ábrán láthatjuk, hogy hogyan néznek ki kétdimenzióban ezek a Voronoi cellák.

A Voronoi celláknak egy sajátosete az, mikor a kezdeti pontokat, vagyis központokat, teljesen véletlenszerűen, mindenféle korreláció nélkül tesszük le. Az így kialakuló alakzatokat, vagy szemcséket, Poisson-Voronoi celláknak nevezzük. A dolgozat keretében azt vizsgáljuk, hogy hogy néz ki ezeknek a szemcséknek a nagyság szerinti eloszlása, vagyis mennyi a valószínűsége annak, hogy egy adott szemcsenagyságot lássunk ha központok között bizonyos fokú korreláció van. Ezzel az egyszerű modellel rendkívül sok fizikai és nem csak fizikai jelenséget lehet modellezni, nem csak a polikristályos anyagok kialakulását, amit mi tűztünk ki célul, hanem többek közt: szemcsés szerkezetű anyagban való vezetés és átszivárgás (fizikában) területek élőlényekkel (növények, állatok, emberek) való benépesedése, baciluskolóniák fejlődése (biológiában) vagy települések fejlődése (szociológiában).^[1]

2. Poisson-Voronoi cellák

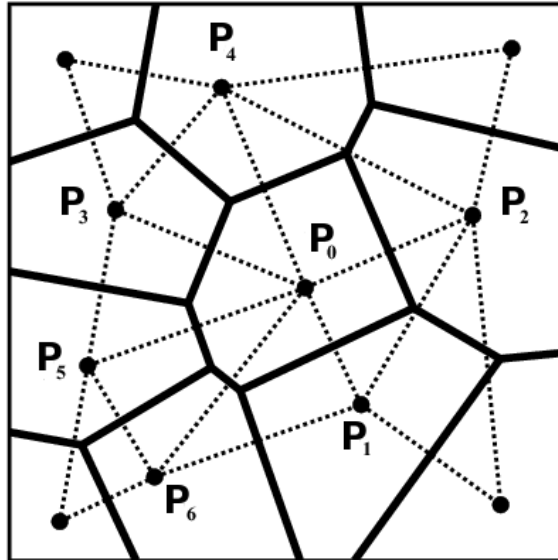
Első lépésként a Poisson-Voronoi cellákból indulunk ki.

Két viszonylag egyszerű módja van annak, hogy Poisson-Voronoi cellákat hozzunk létre. Az elsónél adva van a térben nagyszámú, teljesen véletlenszerűen és minden korreláció nélkül elhelyezett pont. Ezek a pontok elkezdnek minden lehetséges irányba homogén és izotróp módon növekedni. A növekedés egy adott irányba addig tart, amíg össze nem ér egy másik növekedő ponttal. Ha egy pont már semmilyen irányba sem tud továbbnövekedni a Voronoi cellát kialakultnak tekintjük. Minden pont addig nő tovább, amíg össze nem ér a szomszédaival. Egy idő múlva minden pont leáll a növekedéssel, és a tér felosztódik Voronoi cellákra. Ezt a módszert Avrami-Johnson-Mehl módszernek^[3] nevezik, és főleg a számítógépes szimulációknál alkalmazzák, ahogy az 1. –es ábrán látható:



1. ábra Avrami-Johnson-Mehl módszer Voronoi-Poisson cellák létrehozására

Egy másik viszonylag egyszerű Voronoi cella szerkesztési módszerünk kétdimenzióban a merőleges oldalfelező módszere, ami a 2.es ábrán látható.



2. ábra a „merőleges oldalfelező” módszere Voronoi cellák létrehozására

Ennél a módszernél kiindulunk egy P_0 pontból és megkeressük a hozzá legközelebbi központot, a mi esetünkben ez P_1 , majd megszerkesszük a P_0P_1 szakaszt. A P_0P_1 szakaszra merőleges és a felezőponton áthaladó egyenes fogja a Voronoi cella egyik oldalát alkotni. Ezután megkeressük a következő legközelebbi pontot, ebben az esetben P_2 -t. Hasonlóan járunk el mint az előbb, vagyis megszerkesszük a P_0P_2 szakaszt, és a felezőpontján áthaladó, rá merőleges egyenes fogja egy másik oldalát alkotni a Voronoi cellának. Ezt addig folytatjuk, amíg egy zárt sokszöget kapunk. Ezt az algoritmust a tér összes pontjára megismételve megkapjuk a Poisson-Voronoi cellákat.^[2,3]

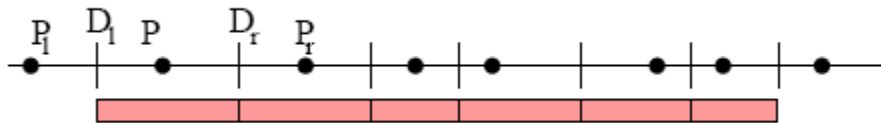
Sok más, kevésbé elterjedt módszer létezik még, amelyekkel létre lehet hozni a Poisson-Voronoi cellákat^[2-4]. Ezen egyszerű Poisson-Voronoi cellák is már nagyon hasznosak természeti és szociális jelenségek leírására és modellezésére. Hogy csak néhányat említsünk: fémcsomó növekedésének modellezése amorf felületen, véletlen rács generálása kvantumtérelméletben^[5], az aktuális galaxiseloszlás értékelésénél, vezetés és átszivárgás szemcsés szerkezetekben, kristályosodás, de használják biológiában, szociológiában, és még sok egyéb területen is.^[1,2]

3. Az egydimenziós eset egzakt tárgyalása

Annak ellenére, hogy ezek a Poisson-Voronoi cellák igen fontosak, és széles a lehetséges felhasználási területük, az ismereteink egyelőre sok szempontból még hiányosak róluk. Az egyik legfontosabb probléma, ami a magasabb ($D > 2$) dimenziók esetén még egzaktul nem ismert, a Poisson-Voronoi cellák nagyságának az eloszlásfüggvénye. Itt először az egyszerű egydimenziós esetet vizsgáljuk, ahol az egzakt megoldás is egyszerű. Ezáltal letesztelhetjük a később alkalmazott számítógépes módszert is.

Most lássuk, hogy hogyan hozhatunk létre Voronoi cellákat egydimenzióban. Adott egy L hosszúságú szakasz, amelyre egymástól függetlenül és véletlenszerűen helyezünk el N pontot. A központok sűrűsége: $n = N/L = 1/\langle d \rangle$, ahol $\langle d \rangle$ az átlagos távolság a központok között. A termodinamikai határesetet tárgyaljuk, amikor N és L tart a végtelenbe, de n véges. A 3-as ábra mutatja, hogy hogyan kapjuk meg ebben az esetben a Poisson-Voronoi cellákat.

Kiindulunk a P pontból és megkeressük a bal és jobboldali szomszédját (P_l , P_r) és feltüntetjük ezen szakaszok felezőpontját is, D_l , D_r . Az így kapott $(D_l D_r)$ szakaszok fogják a Poisson-Voronoi cellákat alkotni.



3. ábra Voronoi cellák szerkesztése 1D ben

Ahhoz, hogy analitikusan megkapjuk az eloszlásfüggvényt, kiindulunk a jól ismert Poisson eloszlásból,

$$P(N, l) = \frac{\langle N \rangle^N \exp(-\langle N \rangle)}{N!} \quad (1)$$

ahol $P(N,l)$ annak a valószínűsége, hogy egy l hosszúságú szakaszon N darab központ legyen. $\langle N \rangle_l = nl$, ahol n a pontok sűrűségét jelenti. Annak a valószínűsége, hogy egy központ se legyen az l hosszúságú szakaszon, az

$$P(0,l) = \exp(-nl) \quad (2)$$

Annak a valószínűsége, hogy balra P_l nagyobb távolságra van, mint l , az

$$P(P_l > l) = P(0,l) = \exp(-nl) \quad (3)$$

Az eloszlásfüggvény a baloldali hossza kiszámolható tehát a következőképpen:

$$g_l(d) = \frac{-\partial P_l(d_l > d)}{\partial d} = n \exp(-nd) \quad (4)$$

Szimmetriai okokból jobb oldalra is ugyanezt az elgondolást alkalmazzuk, és ugyanilyen alakú lesz a jobboldali hossza is az eloszlásfüggvény. A eloszlásfüggvény ezen szakaszok felére, $z = d_l/2$ vagy $z = d_r/2$

$$w(z) = 2n \exp(-2nz) \quad (5)$$

Mivel $d = d_r/2 + d_l/2$, ezért az eloszlásfüggvény d -re úgy számolható ki, mint az (5) képlet önmagával képezett konvolúciója, vagyis:

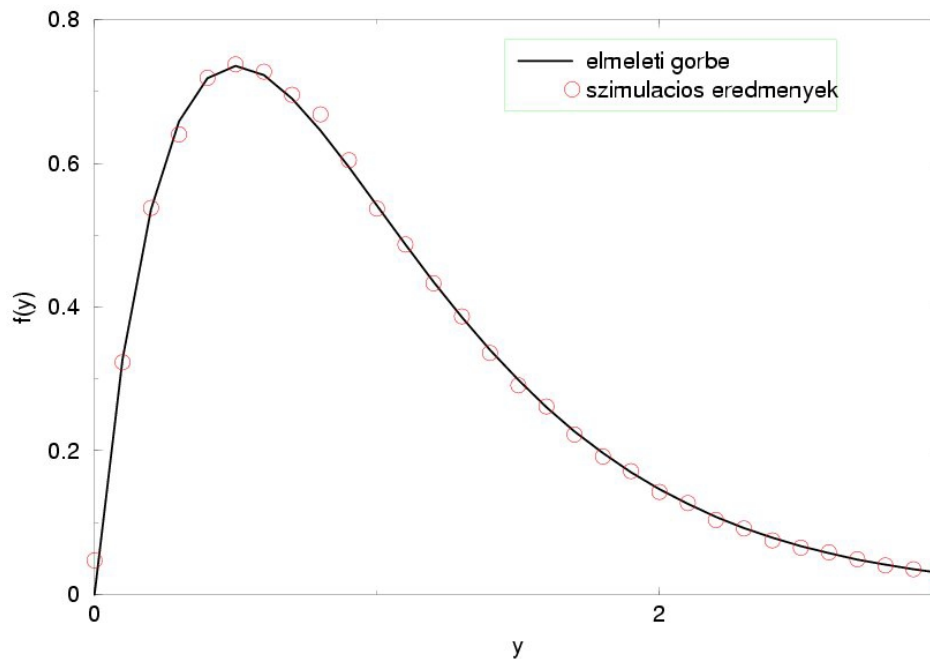
$$g(d) = \int_0^d w(z)w(d-z)dz = 4n \exp(-2nd) \quad (6)$$

Ha figyelembe vesszük azt, hogy L tart végtelenbe, és bevezetjük az $y = d/\langle d \rangle$, akkor a következő eloszlásfüggvényt kapjuk:

$$f(y) = 4y \exp(-2y), \quad (7)$$

ahol $f(y)$ annak a valószínűség-sűrűsége, hogy egy szakasz hossza az y érték körül legyen.

Ezt az elméleti képletet egy egyszerű Monte Carlo típusú szimulációval bizonyítottuk is, és mint a 4-es ábrán látható, a két görbe tökéletesen fedi egymást. Az analitikus eredményekkel való egyezés azt bizonyítja, hogy a szimulációs módszerünk jó, és használható olyan esetekben, ahol analitikus megoldás nem ismert. Szimulációs programunk a C programozási nyelvben íródott és az 1-es mellékletben megtekinthető.



4. ábra Analitikus és szimulációs eredmények a Poisson-Voronoi cellák nagyságeloszlására egydimenzióban. A fekete görbe az egzakt elméleti, a piros pöttyök a szimulációs eredményeket mutatják.

Nagyobb dimenziókban^[1,5,6] azonban már csak közelítések vannak, nincs egy ilyen egzakt és szép alakú eloszlásfüggvény, valamint a szimulációk is eléggé komplikáltak.

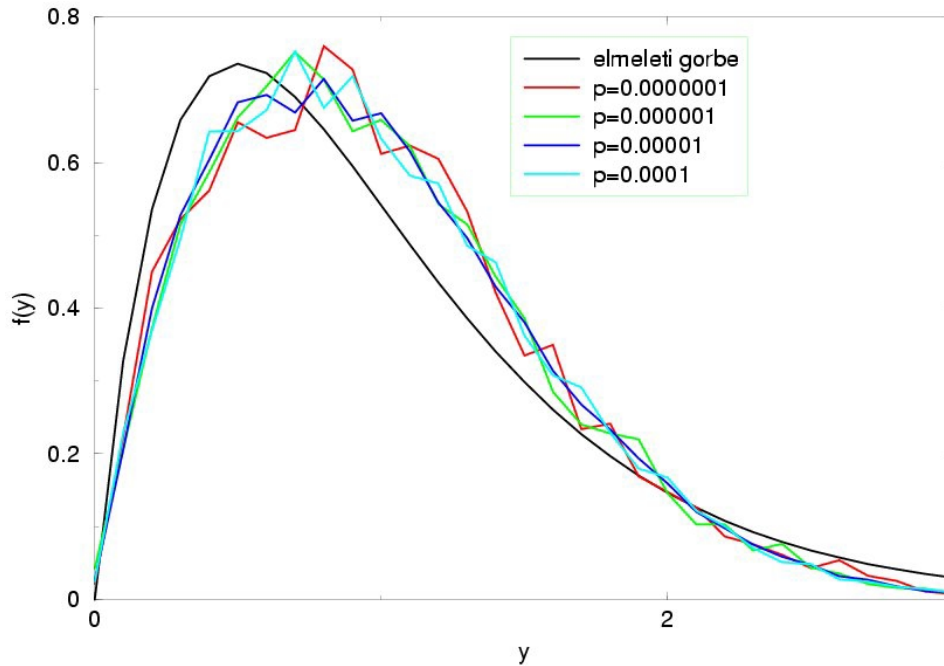
4. Voronoi cellák nukleációs folyamatokra

Arra keressük a választ, hogy mi van akkor, ha a nukleációs központok nem egyszerre alakulnak ki. Ezen esetben egyes pontok már jócskán megnövekedtek, mikor más nukleációs központok eközben jelennek meg. Az a tény, hogy nem egyszerre rakjuk le a pontokat vajon befolyásolja az eloszlásfüggvényt, és ha igen hogyan? Arra keressük a választ, hogy ha konstans nukleálási rátával tesszük le az új nukleációs pontokat a még szabadon maradt helyekre, vajon a nukleálási ráta befolyásolja-e a kapott eloszlásfüggvényt.

Először a legegyszerűbb egydimenziós esetet vizsgáljuk. Tekintsünk egy véges hosszúságú szakaszt. Ezen egyenesre minden időpillanatban egy véletlenszerűen választott koordinátában rögzített valószínűséggel (nukleálási rátával) leteszünk egy újabb pontot. Ez a pont, valamint az eddig lerakott pontok jobbra és balra is növekedni kezdenek (konstans sebességgel), egészen addig amíg a növekvő szakaszok (szemcsék) össze nem érnek. Ha egy szakasz jobb- és baloldalon is összeért a szomszédjával, akkor növekedése megáll és egy Voronoi típusú cellát hozott létre. Egy idő után ezáltal az egyenes megtelik kisebb-nagyobb szemcsékkel. Ezen esetben is a kialakuló szemcsék nagyság-szerinti eloszlásfüggvényét, illetve a nukleálási rátának az erre levő hatását fogjuk tanulmányozni. A problémát az előbbi egyszerű esetben használt Monte Carlo típusú számítógépes szimulációkkal tanulmányozzuk. Különböző P nukleációs valószínűségeket tekintünk, és aránylag nagy rendszerekre jó statisztikát próbálunk elérni. Akárcsak az előző esetben, itt is a Voronoi cellák normált méretére való eloszlás $f(y)$ sűrűségfüggvényét keressük.

A szimulációs eredmények különböző P értékre az 5. ábrán található, összehasonlítva az egydimenzióban kapott Poisson-Voronoi eloszlás sűrűségfüggvényével. A szimuláció azt mutatja, hogy ezen esetben kapott eloszlásfüggvény eltolódik az 1D Poisson Voronoi eloszláshoz képest. Érdekes eredmény azonban az, hogy a normált cellanagyságra kapott sűrűségfüggvény nem függ a P nukleációs valószínűségtől. A $P \in [10^{-3}, 10^{-7}]$ intervallumban kapott szimulációs sűrűségfüggvények gyakorlatilag egybeesnek, ha eltekintünk a statisztikus fluktuációktól.

A szimulációs programot az 1. mellékletben mutatjuk be. Ezen egydimenziós esetet megpróbáltuk analitikusan is megközelíteni, de egyelőre említésre méltó eredmény nem született.



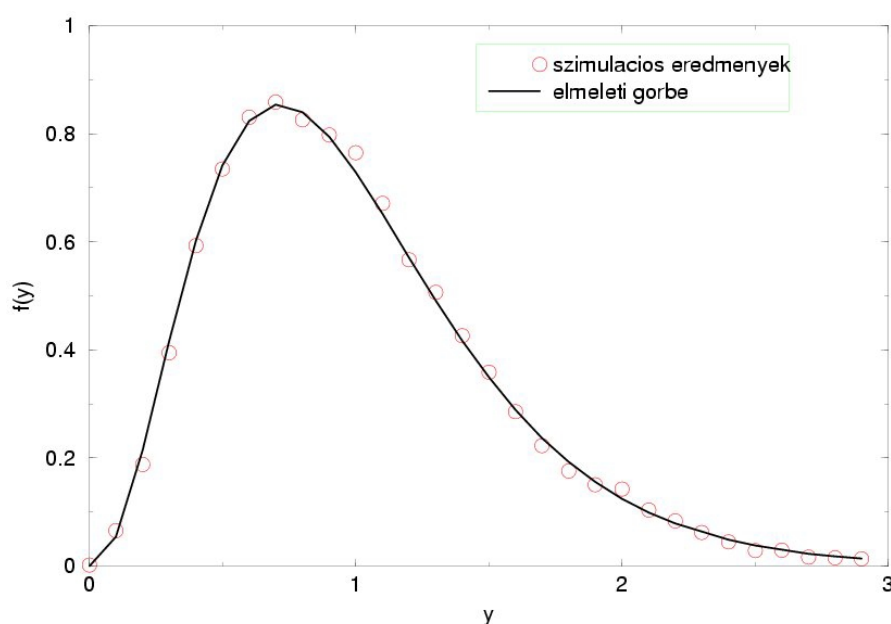
5. ábra Szimulációs eredmények a nukleációs folyamatra kapott eloszlásfüggvényre. A fekete görbe az egydimenziós Poisson-Voronoi eloszlás, a színes görbék a különböző P nukleációs valószínűségekre kapott normált eloszlásfüggvények.

5. A kétdimenziós eset

Érdekes megvizsgálni, hogy vajon magasabb dimenziókban hasonló eredményeket kapunk-e? Magasabb dimenziókban a probléma analitikus megközelítése sajnos nagyon elbonyolódik. Mindeddig nem sikerült egzaktul megkapni a Poisson-Voronoi cellák nagyság szerinti eloszlásfüggvényét két- vagy magasabb dimenziókban. Ezen esetekben a legjobb megközelítések a Monte Carlo szimulációkból adódnak. Ezekből a szimulációs eredményekből^[5-6] arra következtettek^[1], hogy kétdimenzióban az eloszlás sűrűségfüggvényére egy nagyon jó megközelítés a következő:

$$f_{2D}(y) = \frac{343}{15} \sqrt{\frac{7}{2\pi}} y^{\frac{5}{2}} \exp\left(-\frac{7}{2}y\right) \quad (8)$$

Ezen tanulmány keretében sikerült a Monte Carlo módszerünk segítségével újból igazolni a fenti közelítés helyességét. A 6-os ábrán a Poisson-Voronoi cellákra kapott szimulációs eredményünket hasonlítjuk össze a (8)-as közelítéssel, nagyon jó egyezést kapva. A kétdimenziós szimulációs program szintén megtalálható a mellékletben.

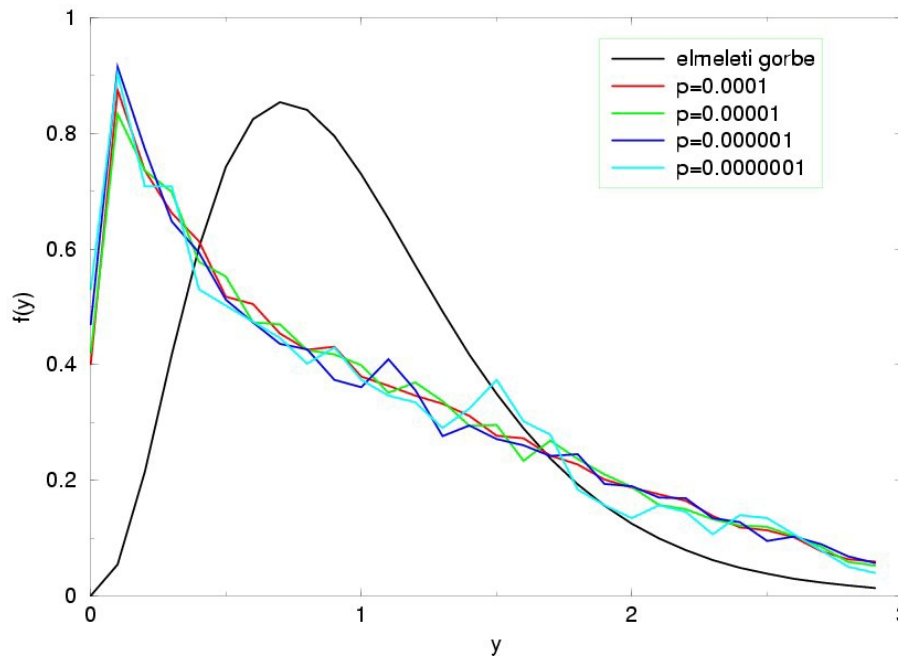


6. ábra Monte Carlo típusú szimulációs eredmények a Poisson-Voronoi cellák nagyságszerinti eloszlásfüggvényére. A fekete görbe a (8) analitikus közelítés, a piros körök a szimulációs eredmények.

Akárcsak egydimenzióban, itt is felmerül a kérdés, hogy mi történik akkor, ha a központok nukleációs csomópontok, amelyek egy adott rögzített rátával jelennek meg. Fog-e változni, s ha igen hogyan az eloszlás sűrűségfüggvénye a Poisson-Voronoi eloszláshoz képest. Vajon a nukleálási ráta befolyásolja-e a kapott sűrűségfüggvényt? A rendszert újból Monte Carlo típusú szimulációkkal tanulmányozzuk, egy kissé módosított Avrami-Johnson-Mehl módszert használva:

1. A kezdetben minden pontot P valószínűséggel aktiváltunk, vagyis P valószínűséggel változtattuk át nukleációs központtá.
2. Későbbi minden időpillanatban, a kétdimenziós tér még üres helyein P valószínűséggel új nukleációs központok jelentek meg.
3. A már meglévő nukleációs központok minden időpillanatban minden irányba növekedtek, addig, amíg össze nem értek és amíg az így kapott cellákkal a tér tökéletesen betöltődött.

Ezen esetben is a kapott Voronoi cellák (szemcsék) nagyság szerinti eloszlása érdekel minket és a sűrűségfüggvényt határoztuk meg. A szimulációs eredményeket a 7-es ábrán tekinthetjük meg.



7. ábra Szimulációs eredmények a nukleációs folyamatra kapott eloszlás sűrűségfüggvényére kétdimenziós esetben. A fekete görbe a Poisson-Voronoi eloszlás sűrűségfüggvénye kétdimenzióban, a színes görbék meg a különböző P valószínűséggel való nukleációkra kapott szimulációs eredmények.

Az ábrán feltüntetett eredmények az egydimenziós esethez hasonló eredményeket sejtetnek. A folytonos nukleációs folyamat során kapott eloszlás sűrűségfüggvénye különbözik a Poisson-Voronoi eloszlás sűrűségfüggvényétől. A normált cellanagyságra

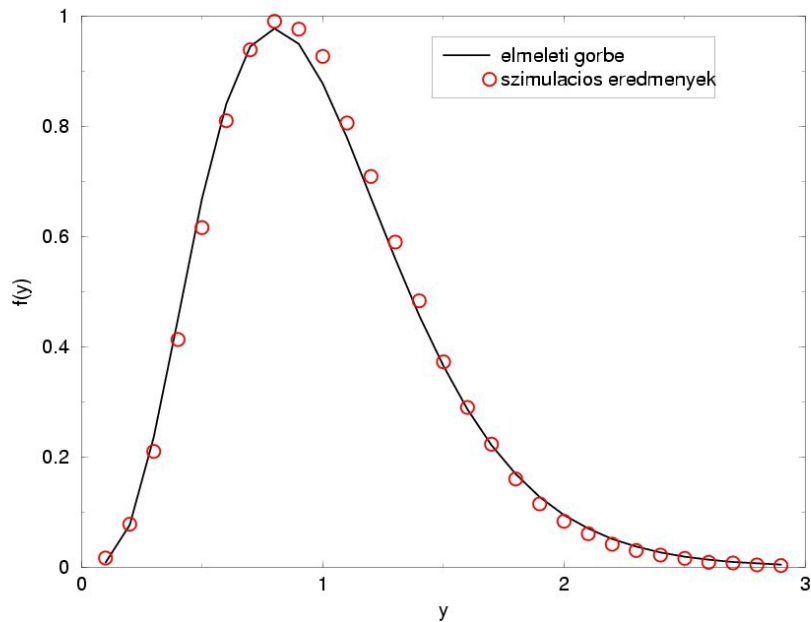
kapott sűrűségfüggvények azonban látszólag nem függnek a P nukleációs valószínűség értékétől. Az egydimenziós esettől eltérően, a folytonos nukleáció esetén kapott eloszlásfüggvény maximuma a Poisson-Voronoi eloszlás maximumához viszonyítva balra tolódik el.

6. A háromdimenziós eset

Először induljunk ki itt is a legegyszerűbb esetből, vagyis a Poisson-Voronoi cellákból, mikor a nukleációs pontokat egyszerre tesszük le. Ahogy az előző alfejezetben is mondtuk, ebben az esetben sem ismert egzakt elméleti képlet, ami leírná az eloszlás sűrűségfüggvényét. Monte Carlo típusú szimulációs eredményekből az jött ki, hogy az eloszlásra egy jó közelítés a következő képlet^[1]:

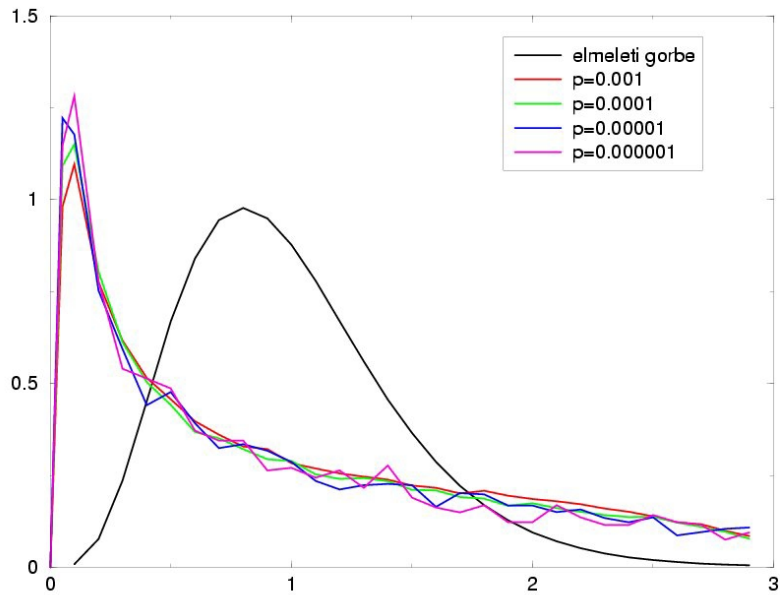
$$f_{3D}(y) = \frac{3125}{24} y^4 \exp(-5y) \quad (9)$$

Ebben az esetben is sikerült a Monte Carlo módszerünk segítségével újból igazolni a fenti közelítés helyességét. A 8-as ábrán a Poisson-Voronoi cellákra kapott szimulációs eredményünket hasonlítjuk össze a (9)-es közelítéssel, nagyon jó egyezést kapva



8.ábra Monte Carlo típusú szimulációs eredmények a Poisson-Voronoi cellák nagyságszerinti eloszlásfüggvényére. A fekete görbe a (9) analitikus közelítés, a piros körök a szimulációs eredmények.

Akárcsak az egy- és kétdimenziós esetben, itt is túlléptük ezt az egyszerű modellt és megvizsgáltuk, hogy mi történik akkor, ha folytonos nukleációval dolgozunk, konstans nukleálási rátákkal. Ugyanúgy jártunk el, mint a kétdimenziós esetben, ugyancsak módosított Avrami-Johnson-Mehl módszert használtunk, ugyanazokkal a módosításokkal, s ugyancsak a normált cellanagyságra kapott sűrűségfüggvények érdekeltek minket, az egyetlen különbség itt az volt, hogy a terünk háromdimenziós volt. Szimulációs eredményeink a 9-es ábrán tekinthetőek meg.



9. ábra Szimulációs eredmények a nukleációs folyamatra kapott eloszlás sűrűségfüggvényére háromdimenziós esetben. A fekete görbe a (9) es analitikus képlet, a színes görbék meg a különböző P valószínűséggel való nukleációkra kapott szimulációs eredmények.

Látható, hogy akárcsak eddig, itt is eltolódik a nukleációs folyamatra kapott eloszlás sűrűségfüggvénye a Poisson-Voronoi eloszlás sűrűségfüggvényétől, de a normált cellanagyságra kapott sűrűségfüggvények látszólag itt sem függenek a P nukleációs valószínűség értékétől. Az egydimenziós esettől eltérően, és a kétdimenziós esettel megegyezően a folytonos nukleáció esetén kapott eloszlásfüggvény maximuma a Poisson-Voronoi eloszlás maximumához viszonyítva balra tolódik el.

7. Következtetések

A tanulmányaink során folytonos nukleáció során kapott szemcsék nagyságának az eloszlását tanulmányoztuk Monte Carlo szimulációs módszerekkel. Egy-, két és háromdimenziós esetet tekintettünk, és a kapott eloszlás sűrűségfüggvényét a Poisson-Voronoi eloszlás sűrűségfüggvényével hasonlítottuk össze. Úgy az egy-, két-, valamint háromdimenziós esetben is a Poisson-Voronoi eloszlástól különböző eloszlást kapunk, ha nem egyszerre jelennek meg a nukleációs központok. A normált szemcsenagyság eloszlásfüggvénye látszólag nem függ azonban a nukleálódási valószínűségtől, vagyis attól, hogy milyen rátával jelennek meg az új nukleációs központok. A kapott eredmények fontosnak ígérkeznek a valós nukleációs folyamatok során kapott szemcsenagyság eloszlások megértéséhez.

Jövőbeli terveink közé tartozik az, hogy egy analitikus megközelítést kapjunk folytonos nukleációs problémára egydimenzióban. Szeretnénk ugyanakkor a feladatot kísérletileg is tanulmányozni telített KBr oldat különböző hőmérsékleteken való kikristályosításával.

8. Könyvészet

- [1] Járai-Szabó Ferenc, Zoltán Néda: On the size-distribution of Poisson Voronoi cells
Physica A 385: 518-526 (2007)
- [2] A. Okabe, B. Boots, K. Sugihara and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams* (Wiley, Chichester, 2000)
- [3] Chapter 7 (Voronoi diagrams) in "Computational Geometry: Algorithms and Applications, Second Edition" by Mark de Berg, Marc van Kreveld, Mark Overmars, Utrecht (the Netherlands) and Otfried Schwarzkopf, Hong Kong (China)
- [4] J.L. Meijering, *Philips Research Reports*, 8, 270 (1953)
- [5] M. Tanemura, *Forma* 18, 221 (2003)
- [6] M. Tanemura in *Proceedings of Intersections of Art and Science*, July 8-14, 2001, Sydney eds. G. Lugosi and D. Nagy
(<http://www.mi.sanu.ac.yu/vismath/proceedings/>)
- [7] J. M. Drouffe and C. Itzykson, *Nucl. Phys. B* 235, 45 (1984)
- [8] G.R. Jerauld, J. C. Hatfield and H. T. Davis, *J. Phys. C* 17(9), 1519 (1984);
G.R. Jerauld, L. E. Scriven and H. T. Davis, *J. Phys. C* 17(19), 3429 (1984)
- [9] S. B. DiCenzo and G.K. Wertheim, *Phys. Rev. B* 39(10), 6792 (1989)
- [10] P. H. Winterfield, L. E. Scriven and M. T. Davis, *J. Phys. C* 14, 2361 (1981)
- [11] Y. Talmon and S. Prager, *Nature* 267, 333 (1977); Y. Talmon and S. Prager, *J. Chem. Phys.* 69(7), 2984 (1978)
- [12] H. Brumberger and J. Goodisman, *J. App. Cryst.* 16, 83 (1983)
- [13] S. Yoshioka and S. Ikeuchi, *Ap. J.* 341, 16 (1989)
- [14] I. K. Crain, *Random Processes in Geology* (New York: Springer-Verlag, 1976)

9. Mellékletek

1. Melléklet 1D Szimulációs program

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#define MBIG 1000000000
#define MSEED 161803398
#define MZ 0
#define FAC (1.0/MBIG)

float ran3(long *idum)
{
    static int inext,inextp;
    static long ma[56];
    static int iff=0;
    long mj,mk;
    int i,ii,k;
    if (*idum < 0 || iff ==0)
    {
        iff=1;
        mj=labs(MSEED-labs(*idum));
        mj %=MBIG;
        ma[55]=mj;
        mk=1;
        for (i=1; i<=54; i++)
        {
            ii=(21*i) % 55;
            ma[ii]=mk;
            mk=mj-mk;
            if (mk < MZ) mk += MBIG;
            mj=ma[ii];
        }
        for (k=1; k<=4; k++)
            for (i=1; i<=55; i++)
                {
```

```

        ma[i] -= ma[1+(i+30) % 55];
        if (ma[i] < MZ) ma[i] += MBIG;
    }
    inext=0;
    inextp=31;
    *idum=1;
}
if (++inext == 56) inext=1;
if (++inextp == 56) inextp=1;
mj=ma[inext]-ma[inextp];
if (mj < MZ) mj += MBIG;
ma[inext]=mj;
return mj*FAC;
}

```

```

main ()
{
    FILE *fp;
    long n,i,j,k,l,y,f,ln,s,q,X,K,F;
    long t;
    float d,v,x;
    long a[100000];
    long b[100000];
    long m[30000];
    long E[30000];

    printf("n=");
    scanf("%d",&k);
    printf("\n");
    t=-100000;
    n=k*10;
    K=0;
    for (i=1; i<=n; i++)
    {
        E[i]=0;
    }
    F=0;

```

```

    f=0;
printf("X=");
scanf("%d",&X);
for (i=1; i<=20000; i++)
    {
    m[i]=0;
    }
for (q=1; q<=X; q++)
{

k=(int) (n/10);
ln=1;
do
{
for (i=1; i<=n; i++)
{
    d=(float) (rand()/(RAND_MAX+1.0));
    if (d<0.1)
    {
        v=ran3(&t);
        i=(int) (v*k*10);
        if (a[i]==0)
        {
            a[i]=i;
        }
    }
}
i=1;
do
{
    if (a[i]!=0)
    {
        if (a[i-1]==0)
        {
            a[i-1]=a[i];
        }
        if (a[i+1]==0)

```

```

        {
            a[i+1]=a[i];
            i++;
        }
    }
    i=i++;
} while(i<=n);
l=1;
for (i=1; i<=n; i++)
{
    if(a[i]==0)
    {
        l=0;
    }
}
} while(l!=1);
i=2;j=1;
b[1]=1;
do
{
    if (a[i]==a[i-1])
    {
        b[j]=b[j]+1;
    }
    else
    {
        j++;
        b[j]=1;
    }
    i++;
} while (i<=n);

k=j;
K=K+k;
s=(int) (n/k);
for(i=1; i<=k; i++)
{
b[i]=b[i]*10;

```

```

b[i]=(int) (b[i]/s);
}
for (i=1; i<=n; i++)
{
if(f<b[i])
    f=b[i];
}
if (F<f)
{
F=f;
}
for (j=1; j<=f; j++)
{
    for (i=1; i<=k; i++)
    {
        y=j-1;
        if ( (b[i]>y)  && (b[i]<=j) )
        {
            m[j]=m[j]+1;
        }
    }
}
l=0;
v=0;

for (i=1; i<=f; i++)
{
E[i]=E[i]+m[i];
m[i]=0;
}
for (i=1; i<=n; i++)
{
a[i]=b[i]=0;
}
}
fp=fopen("akt1","w");
for (j=1; j<=F; j++)
{

```

```

x=j;
if (E[j]!=0)

{
    v=(float)((x-1)/10);
    d=(float)(x/10);
    fprintf(fp," %f ",((v+d)/2.0));
    fprintf(fp,"%f \n", (float)((E[j]*10)/(float)(K)));
}
}
fprintf(fp,"osszesen %ld db pont ", K);
close(fp);
return(0);
}

```

2. Melléklet 2D szimulációs program

```

#include <stdlib.h>
#include <stdio.h>

#define N 5000
#define k 40000
#define p 0.0001

unsigned long inext(unsigned long i)
{
    if (i<N-1) {return i+1;}
    else return 0;
}

unsigned long iprev(unsigned long i)
{
    if (i>0) {return i-1;}
    else return N-1;
}

```



```

int main()
{
unsigned long *a;
a=calloc(N*N, sizeof(unsigned long));
unsigned long *c;
c=calloc(N*N, sizeof(unsigned long));
//int a[N] [N],c[N] [N];
unsigned long b[100];
unsigned long j,i;
unsigned long m,x;

double w;
int q;
FILE *fo;

for(i=0; i<N; i++)
{
for(j=0; j<N; j++)
{
*(a+i*N+j)=0;
//a[i] [j]=0;
*(c+i*N+j)=0;
//c[i] [j]=0;
}

}
m=0;
for(i=0; i<100; i++)
{
b[i]=0;
}

do
{

```

```

i=0;
do
{
j=0;
do
{
if ((* (a+i*N+j)==0) && (m<k)) {
if ((rand()/(RAND_MAX+1.0)) < p) {
//printf("k ");

m++;
*(a+i*N+j)=m;
//printf("%ld ",m);fflush(stdout);
}
}

j=j+1;
//printf("\n%ld\n",j);
}while(j<N);
i=i+1;
}while(i<N);
}while(m!=k);
printf("tele \n");fflush(stdout);

do
{
m=0;
for (i=0; i<N; i++)
{
for(j=0; j<N; j++)
{
*(c+i*N+j)=*(a+i*N+j);
}
}
}

i=0;
do
{

```

```

j=0;
do
{
if ( *(a+i*N+j) != 0 ) {
        if (*(a+iprev(i)*N+iprev(j))==0) {if
(* (c+iprev(i)*N+iprev(j))==0)
{* (c+iprev(i)*N+iprev(j))=*(a+i*N+j);m=1;}
                else if
((rand()/(RAND_MAX+1.0)) >0.5)
{* (c+iprev(i)*N+iprev(j))=*(a+i*N+j);}}
        if (*(a+iprev(i)*N+j)==0)          {if (*(c+iprev(i)*N+j)==0)
{* (c+iprev(i)*N+j)=*(a+i*N+j);m=1;}
                else if
((rand()/(RAND_MAX+1.0)) >0.5) {* (c+iprev(i)*N+j)=*(a+i*N+j);}}
        if (*(a+iprev(i)*N+inext(j))==0) {if
(* (c+iprev(i)*N+inext(j))==0)
{* (c+iprev(i)*N+inext(j))=*(a+i*N+j);m=1;}
                else if
((rand()/(RAND_MAX+1.0)) > 0.5)
{* (c+iprev(i)*N+inext(j))=*(a+i*N+j);}}
        if (*(a+i*N+iprev(j))==0)          {if
(* (c+i*N+iprev(j))==0) {* (c+i*N+iprev(j))=*(a+i*N+j);m=1;}
                else if
((rand()/(RAND_MAX+1.0)) > 0.5) {* (c+i*N+iprev(j))=*(a+i*N+j);}}
        if (*(a+i*N+inext(j))==0)          {if (*(c+i*N+inext(j))==0)
{* (c+i*N+inext(j))=*(a+i*N+j);m=1;}
                else if
((rand()/(RAND_MAX+1.0)) > 0.5) {* (c+i*N+inext(j))=*(a+i*N+j);}}
        if (*(a+inext(i)*N+iprev(j))==0) {if
(* (c+inext(i)*N+iprev(j))==0)
{* (c+inext(i)*N+iprev(j))=*(a+i*N+j);m=1;}
                else if
((rand()/(RAND_MAX+1.0)) > 0.5)
{* (c+inext(i)*N+iprev(j))=*(a+i*N+j);}}
        if (*(a+inext(i)*N+j)==0)          {if (*(c+inext(i)*N+j)==0)
{* (c+inext(i)*N+j)=*(a+i*N+j);m=1;}
                else if
((rand()/(RAND_MAX+1.0)) > 0.5) {* (c+inext(i)*N+j)=*(a+i*N+j);}}

```

```

        if (*(a+inext(i)*N+inext(j))==0) {if
(* (c+inext(i)*N+inext(j))==0)
{* (c+inext(i)*N+inext(j))=*(a+i*N+j);m=1;}
                                else if
((rand()/(RAND_MAX+1.0)) > 0.5)
{* (c+inext(i)*N+inext(j))=*(a+i*N+j);}}
        }
j++;
}while(j<N);
i++;
}while(i<N);

for (i=0; i<N; i++)
{
for(j=0; j<N; j++)
{
*(a+i*N+j)=*(c+i*N+j);
}
}

}while(m!=0);

printf("kiiras");fflush(stdout);
fo=fopen("v2d510.dat","w");
do
{
m=0;
w=0.0;
for(i=0; i<N; i++)
{
for(j=0; j<N; j++)
{

if ((*(a+i*N+j) != 0 ) && (m==0 )) {m=*(a+i*N+j);*(a+i*N+j)=0;w=1.0;}

}
}
}

```

```

//printf("%ld",m);fflush(stdout);
q=0;
for(i=0; i<N; i++)
{
for(j=0; j<N; j++)
{
if (*(a+i*N+j) != 0 ){q=1;
if (*(a+i*N+j)==m) {w=w+1.0;*(a+i*N+j)=0;}}
}
}
if (w!=0.0)
//printf("%ld ",x);fflush(stdout);
{
w=((w*(double)(k))*10.0/(double)(N*N));x=(long)(w);if ((w-x) < (x+1-
w)) {b[x]++;}
else if ((w-x) == (x+1-w))
&& ((rand()/(RAND_MAX+1.0)) > 0.5)) {b[x]++;}
else {b[x+1]++;}}
//printf("m1");fflush(stdout);
}while(q!=0);
printf("szamolva");fflush(stdout);
for(i=0; i<30; i++)
{
if (b[i]!=0) {fprintf(fo,"%f %f \n",((float)(i))/10.0,
(float)(b[i])*10.0/(float)(k));}
}

printf("vege"); fflush(stdout);
return 0;
}

```

