

Egydimenziós szemcsenövekedés statisztikája

FIZIKA KAR
BABES-BOLYAI
TUDOMÁNYEGYETEM



Boda Szilárd, Babeş-Bolyai Tudományegyetem, Fizika kar, II. Fizika-Informatika

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

Kivonat

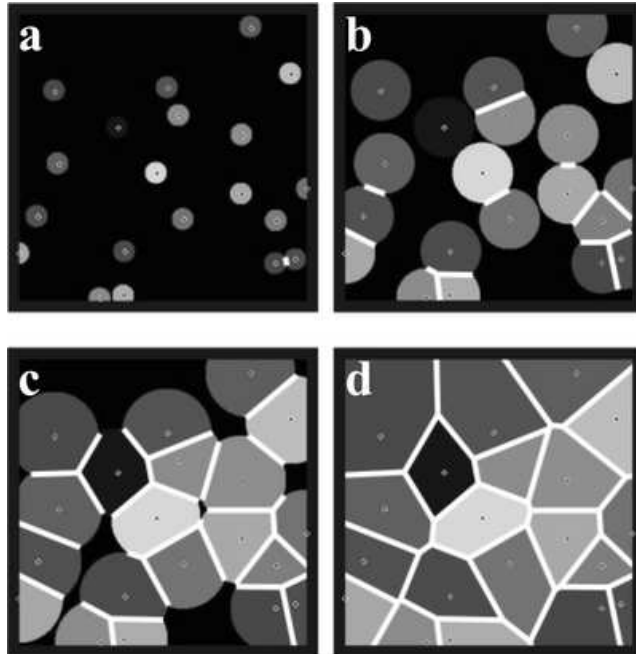
Egy egyszerű egydimenziós modellt vizsgálunk számítógépes módszerek segítségével. A nukleációs központokat térben véletlenszerűen megválasztva, de időben nem egyszerre lerakva a kialakuló szemcsék nagyság szerinti eloszlását vizsgáljuk. Az egyszerű Poisson-Voronoi eloszlástól különböző eloszlásfüggvényt kapunk. Az eloszlásfüggvény azonban látszólag független a nukleációs valószínűségtől. Ezen eredményeket úgy a Monte Carlo típusú szimulációk, mint egy analitikus közelítés is szépen igazolja.

Bevezetés

Képzeljünk el a térben sok-sok egyformán apró nukleációs központot. Most képzeljük el hogy minden ilyen szemcse elkezd egyenletesen növekedni, minden irányba. Egy adott irányba addig növekszik míg egy másik növekedő szemcsével össze nem ér. Amikor összérnek, megpróbálnak más irányba nőni, úgyhogy a rendelkezésükre álló teret teljesen betöltsék. Ha egy szemcse már nem tud tovább nőni a szomszédaitól, akkor megáll. A többi szemcse nő tovább, ha tud a szomszédaitól nőni. Ez addig megy így, még a szemcsék teljesen betöltik a rendelkezésükre álló teret. Mi azt vizsgáljuk, hogy hogy néz ki ezeknek a szemcséknek a nagyság szerinti eloszlása, vagyis hogy mennyi a valószínűsége annak, hogy egy tetszőleges szemcse nagysága az y , $y+dy$ intervallumba essen, ahol dy egységnyi kicsi. Ezzel rendkívül sok fizika és nem csak fizikai jelenséget lehet modellezni, mint például polikristályos anyagok kialakulása, folytonos anyagban való vezetés és átszivárgás (fizikában) területek élőlényekkel (növények, állatok emberek) való benépesedése, víruskolóniák fejlődése (biológiában) vagy települések fejlődése (szociológiában).^[1]

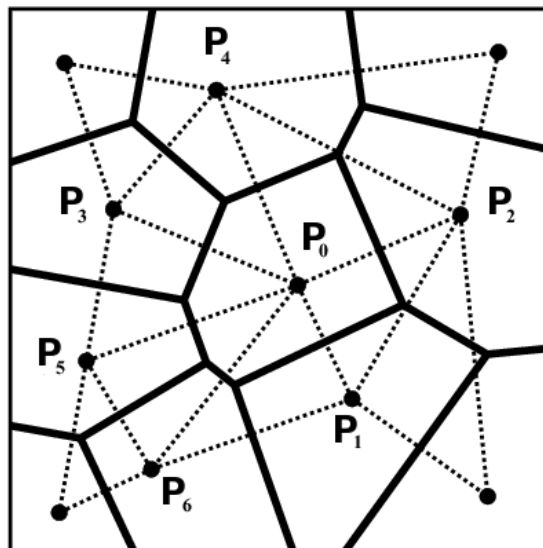
Poisson-Voronoi cellák

Első közelítésként a Poisson-Voronoi cellákból indulunk ki. Egy térben adott bizonyos számú véletlenszerűen elhelyezett pont. Ezek a pontok addig növekednek egy adott irányba, míg össze nem érnek egy másik növekedő ponttal. Ha találkoztak egy másik ponttal, más irányba növekednek. Ha egy pont már semmilyen irányba sem tud továbbnövekedni megáll, vagyis nem nő tovább. A többi pont azonban nő tovább, amíg ugyanebbe a helyzetbe nem jut. Egy idő múlva egyik pont sem nő tovább, mert az adott tér betelik. Az így keletkezett pontokat (alakzatokat) nevezzük Poisson-Voronoi celláknak, vagy Poisson-Voronoi diagramoknak, és ezt a módszert Avrami-Johnson-Mehl módszernek^[3], ez főleg a számítógépes szimulációknál alkalmazzák, ahogy az egyes ábrán látható:



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

Ez azonban csak az egyik megfogalmazásuk 2D ben az alábbi, 2 es ábra mutatja a Voronoi cellákat:



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

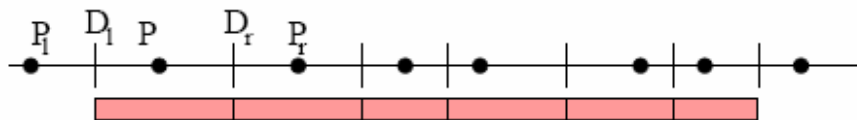
Kiindulunk a P_0 pontból és megkeressük a hozzá legközelebbi központot, a mi esetünkben P_1 , és megszerkesszük a P_0P_1 szakaszt. P_0P_1 szakaszra merőleges és a felezőponton áthaladó egyenes fogja a Voronoi cella egyik oldalát alkotni. Ezután megkeressük a következő legközelebbi pontot, a mi esetünkben P_2 . 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]

Ezek a Poisson-Voronoi cellák nagyon hasznosak egy csomó természeti és szociális jelenség leírására és modellezésére. Hogy csak néhányat említsék: fémcsomó növekedésének modellezése amorf felületen, véletlen rács generálása kvantumtérelméletben, az aktuális galaxiseloszlás értékelésénél, vezetés és átszivárgás szemcsés szerkezetekben, kristályosodás, de használják biológiában, szociológiában is, és még sok egyéb helyen.^[1,2]

Az egydimenziós eset tárgyalása^[1]

Annak ellenére hogy ezek a Poisson-Voronoi cellák ilyen fontosak, és ilyen sok területen lehet őket használni, az ismereteink egyelőre elég hiányosak róluk. Először megjegyeznénk, hogy az előbbi két módszeren kívül még sok más módszer is van, amellyel létre lehet hozni a Voronoi cellákat 1Dben^[1-4]. Adott egy L hosszúságú szakasz, amelyre egymástól függetlenül és véletlenszerűen helyezünk el N pontot. A központok sűrűségét a következő képlet adja: $n=N/L=1/\langle d \rangle$, ahol $\langle d \rangle$ az átlagos távolság a központok között. Mi azt az esetet tárgyaljuk, amikor N , L tart a végtelenbe, de n véges. A 3. ábra mutatja 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 megkapjuk az eloszlásfüggvényt, kiindulunk a jól ismert Poisson eloszlásból,

$$P(N,l)=\langle N \rangle_l^N \exp(-\langle N \rangle_l) / N!, \quad (1)$$

ahol $P(N,l)$ annak a valószínűsége, hogy egy l hosszúságú szakaszon N darab központ van, és $\langle N \rangle_l = nl$, az \exp pedig azt jelenti, hogy az e szám az illető hatványon.

Annak a valószínűsége, hogy egy ilyen központ sincs az l hosszúságú szakaszon, az

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

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

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

Az eloszlásfüggvény a bal oldalra tehát kiszámolható a következőképpen:

$$g_l(d) = -\partial P_l(d_l > d) / \partial d = n \exp(-nd). \quad (4)$$

Szimmetriai okokból jobb oldalra is ugyanezt az elgondolást alkalmazzuk, és ugyanilyen alakú lesz a jobboldali eloszlásfüggvény is. Az 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 konvolúciója, vagyis:

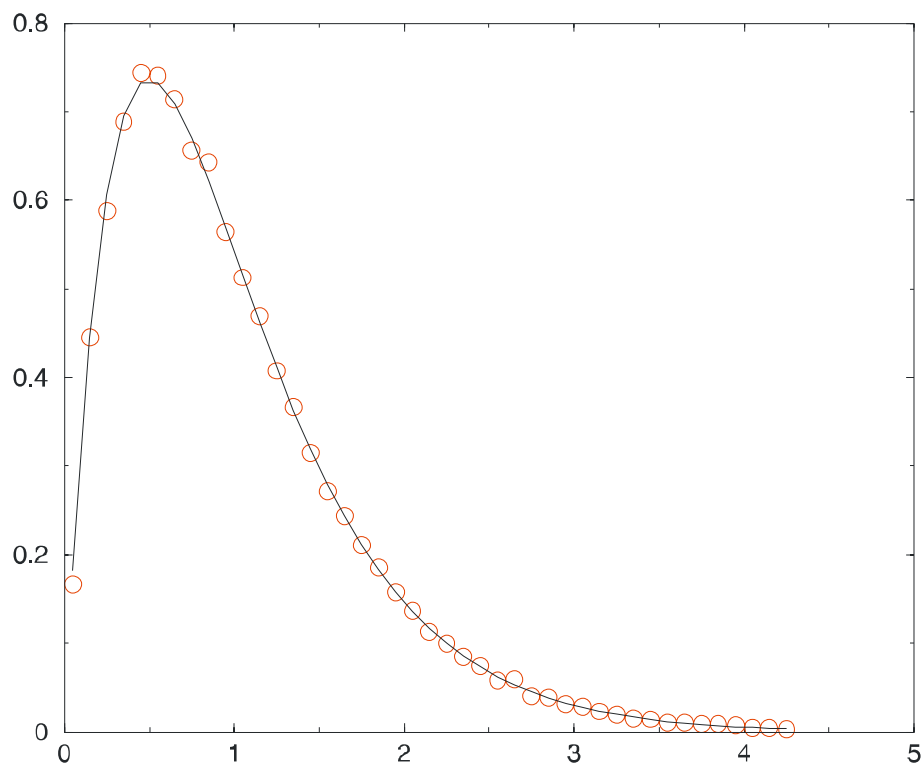
$$g(D) = \int_0^D w(z)w(D-z)dz = 4n \exp(-2nD). \quad (6)$$

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 valószínűsége, hogy egy szakasz az $(y, y+dy)$ intervallumba essen,

Ezt az elméleti képletet egy egyszerű szimulációval bizonyítottuk is, és mint a 4-es ábrán látható, a két görbe tökéletesen fedi egymást, tehát ebben az esetben ezt a problémát sikeresen megoldottuk. Szimulációs programunk az 1-es mellékletben látható.



4. *ábra* a fekete görbe az elméleti, a piros pöttyök a szimulációs görbét mutatják

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

Túllépés a Poisson-Voronoi féle megközelítésből

Mivel dolgozatunkban egyelőre csak egydimenzióban tanulmányozzuk a szemcsésedést ez a Poisson-Voronoi modell elégséges volna De van pár probléma ami ezzel a modellel még ebben az egyszerű egydimenziós esetben sem magyarázható. Mi van akkor ha a nukleációs központok nem egyszerre alakulnak ki? Mi van akkor ha egyes pontok már növekednek, mások meg eközben jelennek meg? Az a tény, hogy nem egyszerre rakom le a pontjaimat fogja-e befolyásolni az eloszlásfüggvényt? Mi van akkor ha minden időpillanatban bizonyos valószínűséggel tesztek csak le új nukleációs pontot, ez a valószínűség fogja befolyásolni az eloszlásfüggvényt?

Mivel a Poisson-Voronoi cellák ezekre a kérdésekre nem adnak választ, új modellre van szükségünk. Itt is a legegyszerűbb esetet vesszük, az egydimenziós esetet. Képzelnünk egy véges hosszúságú szakaszt. Erre az egyenesre minden időpillanatban bizonyos valószínűséggel letesznek egy pontot egy véletlenszerűen kiválasztott pontban. Ez a pont, valamint az eddig lerakott pontok elkezdnek jobbra és balra is növekedni, amíg a növekvő szakaszok (szemcsék) össze nem érnek. Ha egy szakasz jobb- és baloldalon is összeért a szomszédjával, akkor nem nő tovább. Egy idő után az egyenes megtelik kisebb-nagyobb szakaszokkal. Itt is természetesen az eloszlásfüggvényre vagyok kíváncsi. Arra hogy mennyi a valószínűsége annak hogy egy tetszőleges szakasz hossza y és $y+dy$ intervallumban legyen egységnyi dy intervallumban.

A problémát először Monte Carlo típusú számítógépes szimulációkkal tanulmányozzuk. Különböző P valószínűségekre futtatjuk a szimulációt. Itt P annak a valószínűsége hogy tetszőleges időpillanatban leteszünk egy újabb pontot. Az így kapott eredményeket ábrázoljuk $f(y)$, y koordináta-rendszerben. Utána analitikusan próbáljuk megközelíteni és megoldani a problémát. Az analitikus megoldáshoz a következő egyenletrendszert kell megoldani:

$$N(t+1)=N(t)+P \quad (8)$$

$$W(1,t+1)=W(1,t)(1-P_1-P_2)+W(l-1,t)P_1+W(l-2,t)P_2 \quad (9)$$

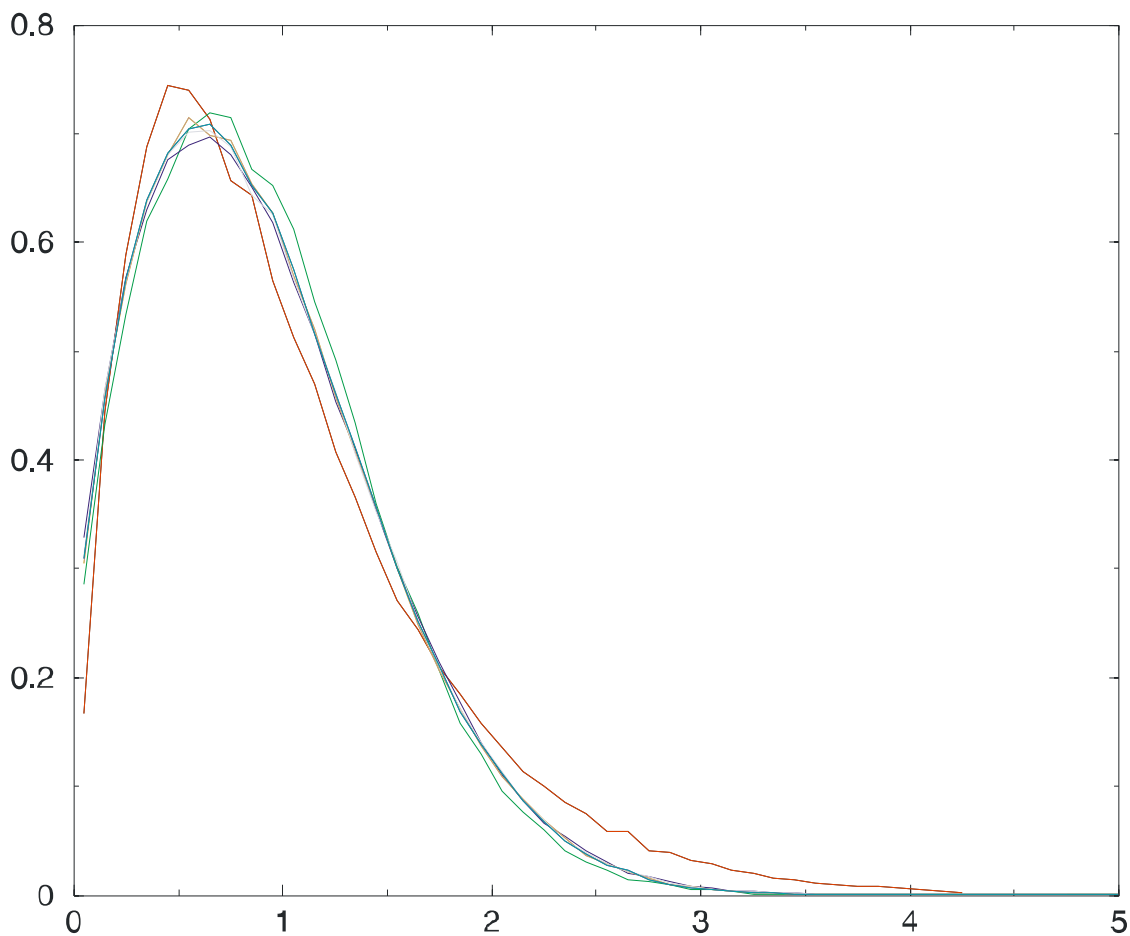
$$W(1,t+1)=W(1,t)(1-P_1-P_2)+P \quad (10)$$

$$W(2,t+1)=W(2,t)(1-P_1-P_2)+W(1,t)P_1 \quad (11)$$

Ahol, $N(t)$ a pontok száma t időpillanatban, $W(l,t)$ az l hosszúságú szakaszok száma t edik időpillanatban, P_1 -annak a valószínűsége, hogy egy tetszőleges szakasz egyik irányba nő, P_2 -annak a valószínűsége hogy egy tetszőleges szakasz mindkét irányba nő, P pedig annak a valószínűsége, hogy leteszünk egy pontot. Ezeket az egyenleteket egy egyszerű iterációs program segítségével megoldhatjuk, az iteráció keretén belül pedig áttérünk $W(l,t)$ ről $f(y)$ ra. Az itt kapott görbék az eddigi koordináta-rendszerben ábrázoljuk. A (8) es egyenlet azt mutatja, hogy hogyan növekednek a pontok az idő múlásával. A (9) as egyenlet azt fejezi ki, hogy annak a valószínűsége, hogy egy $t+1$ edik

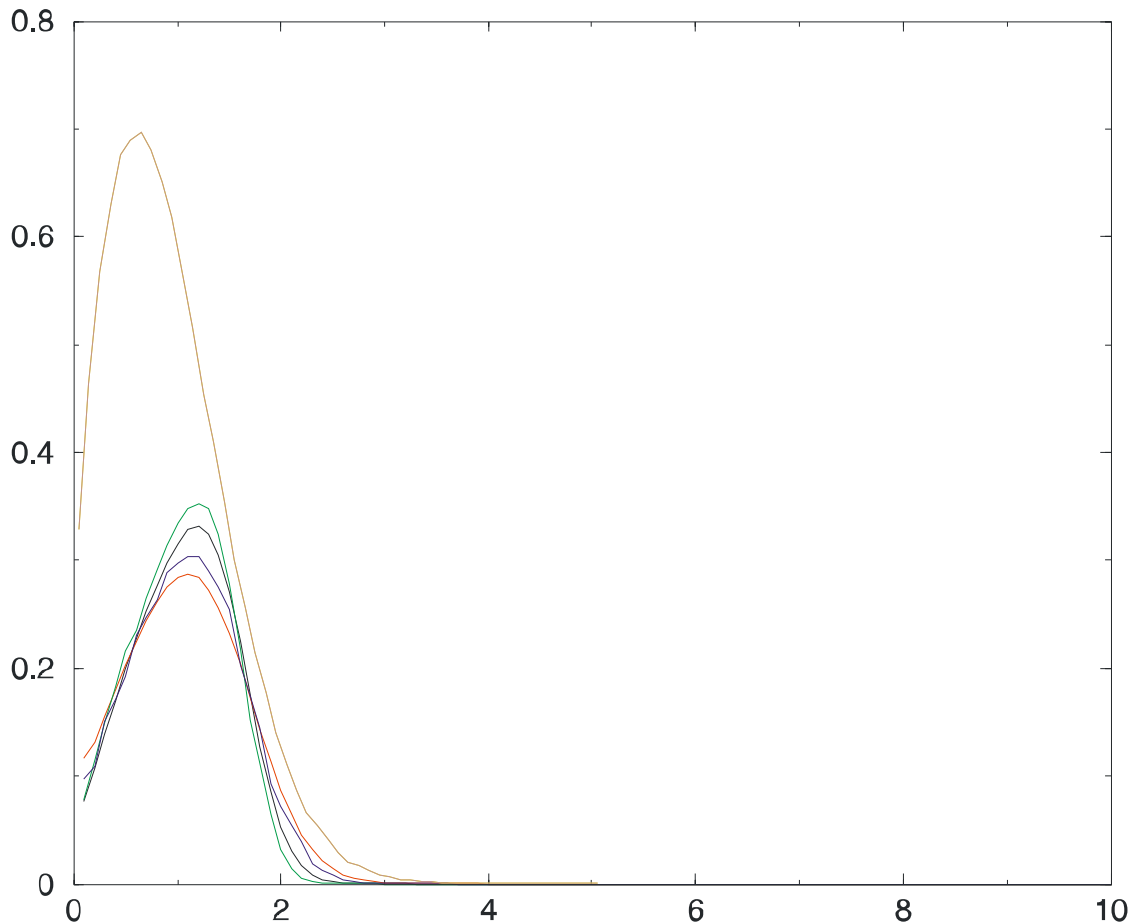
időpillanatban legyen l hosszúságú szakasz, az három valószínűség összege. Első, hogy ez létezett a t időpillanatban l hosszúságú szakasz, és nem nőtt. Második, létezett $l-1$ hosszúságú szakasz a t pillanatban, amelynek egyik oldala nőtt. Harmadik, létezett $l-2$ hosszúságú szakasz, amelyik mindkét oldalra nőtt. A (10)es és (11)es egyenletek a (9) egyenlet kevésbé általánosabb megfogalmazásai, mikor $l=1$ illetve $l=2$. Bővebben, a (10) es egyenlet azt mondja, hogy annak a valószínűsége, hogy legyen egy l hosszúságú szakasz, az annak a valószínűsége, hogy legyen egy l hosszúságú szakasz a t időpillanatban és ne nőjön, plusz annak a valószínűsége hogy megjelenjen egy új pont. A (10) es egyenlet azt jelenti, hogy annak a valószínűsége, hogy legyen egy 2 es hosszúságú szakasz a $t+1$ edik időpillanatban, az annak a valószínűsége, hogy legyen 2 es hosszúságú a t edik időpillanatban és az ne nőjön, plusz annak hogy legyen 1 hosszúságú, amelyek egyik oldala nő.

A szimulációs eredmények különböző P értékre a 4. ábrán található. A szimuláció azt mutatja, hogy az eloszlásfüggvény kicsit eltolódik abban az esetben ha nem egyszerre rakjuk le a pontokat az 1D Poisson Voronoi eloszláshoz képest. Ez az eltolódás azonban nem függ attól a P valószínűségtől, hogy hogyan rakjuk le a pontokat. A szimulációs programunk az 1. mellékletben van bemutatva.



4. ábra szimulációs eredmények. A piros görbe a Poisson-Voronoi eloszlás, a színes görbék a különböző valószínűséggel letett pontok eloszlásfüggvényei

Az analitikus megközelítés is ezt az eredményt sugallja, habár ez csak egy pontatlan első megközelítésnek jó. Az analitikus megközelítéssel kapott eloszlásfüggvények hasonló alakúak lesznek de, mint látható a 5. ábrán látható, nagy különbség van köztük és a szimulációkkal kapottakkal. Az analitikus megközelítést számoló programunk a 2. mellékletben található.



5. ábra Az analitikus megközelítések különböző valószínűségekre, összehasonlítva egy szimulációs eredménnyel

Következésképp levonhatjuk, hogy a szemcsenövekedés eloszlásfüggvénye egydimenzióban csak attól függ, hogy egyszerre vagy nem egyszerre jelennek meg a nukleációs központok. Érdekes, hogy eredményeink azt sugallják, hogy nem függ azonban attól, hogy a szemcsék milyen valószínűséggel jelennek meg.....

A jövőben szeretnénk 2Dben és 3Dben is tanulmányozni és megoldani a feladatot .

Hivatkozások:

[1] Járai-Szabó Ferenc, Zoltán Néda: On the size-distribution of Poisson Voronoi cells

[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)

Mellékletek.

1. Melléklet 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, "összesen %ld db pont ", K);
close(fp);
return(0);
}

```

2. Melléklet Iterációs program

```

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

int main()
{
int l,t,k,n,i,x;
float r;
double N,P,P1,P2,S,Pn,C,X,Y,L,m,la,l1;
FILE* fp;
FILE* fa;
FILE* fb;
double *A;
A = (double *)calloc( 10000, sizeof( double ) );
double *B;

```

```

B = (double *)calloc( 10000, sizeof( double ) );
double *W;
W=calloc(10000*10000, sizeof(double));

printf("L=");
scanf("%lf",&L);
*(W+1*10000+0)=1;
N=1;
S=1;
X=1-(1.0/(L-S+1));
P=0.2;
C=(L-S)*P/L;
Y=2*(N-1);
P2=pow(X,Y)*2*P/L;
Y=Y/2;
P1=pow(X,Y)*sqrt(2*P/L);
t=0;
do
{
*(W+1*10000+(t+1))=*(W+1*10000+t)*(1-P1-P2)+C;
*(W+2*10000+(t+1))=*(W+2*10000+t)*(1-P1-P2)+*(W+1*10000+t)*P1;
N=N+C;
n=(int)(N);
for(l=3; l<=n; l++)
{
*(W+1*10000+(t+1))=
*(W+1*10000+t)*(1-P1-P2)+*(W+(l-1)*10000+t)*P1+*(W+(l-2)*10000+t)*P2;
}
m=S;
S=0;
for(i=1; i<=n; i++)
{
S=S+*(W+i*10000+t)*i;
}
S);
X=1-(1.0/(L-S+1));
Y=(N-1)*2;
P2=(2*P*pow(X,Y))/L;
Y=Y/2;
P1=pow(X,Y)*sqrt(2*P/L);
C=((L-S)*P)/L;
t=t+1;
k=(long)(S+0.01);
x=(long)(L);
} while (x!=k);
printf("\n");
fp=fopen("i02","w");
S=0;
for (i=1; i<=1000; i++)
{
S=S+*(W+i*10000+t);
}
l1=0;
for (i=1; i<=1000; i++)
{
A[i]=*(W+i*10000+t)/S;
}

```

```
l1=l1+A[i]*i;
}
la=L/N;
printf("%lf %lf", la, l1);
for (l=1; l<=1000; l++)
{
x=(int) (l*10/l1);
B[x]=A[l]*l1/2;
}
for (i=1; i<=100; i++)
{
r=(float) (i);
r=r/10;
fprintf(fp, "%f %lf \n", r, B[i]);
fflush(stdout);
}
close(fp);
return(0);
}
```