Experiment 6

Anirrudh Ramesh (MM16B014)[a]

[a] *Indian Institute of Technology Madras*

Generation of a 2D Digital Microstructure

Objectives

1. To show the growth of equiaxed and elongated grains

Solution :

I wrote a code to generate a digital microstructure based on the algorithm that creates a 2D Representative volume element (RVE). The code takes an input the following (1) size of the computational grid (2) Number of grains (3) 2 D Velocity vector (4) Frequency with which you write the output files.
We construct a Periodic set of Voronoi vertices for implication of a periodic microstructure. A virtual blowed up RVE is obtained. Now this new virtual grid has the proportionate amount of nuclei. Assuming the growth of the grains is elliptical at any point, we use the basis RVE to calculate distances from every pixel (inherently that's also a way to construct an ellipse) and the smallest distance is calculated and the corresponding ID is stored. This is a time independent algorithm and shows us only the final step.
The following assumptions motivate the choice of Voronoi tessellations for the representation of a polycrystalline aggregate:

1. Grain growth starts at all points p_i in a finite set of nuclei S $\subset$ R (referred to as point seed) at the same time T_0. The nuclei are fixed at their spatial position during the growth process, i.e. they do not move
2. Isotropic and uniform grain growth. Particularly, the velocity of the grain growth is assumed to be equal in
   (a) all grains and
   (b) all directions.
3. Grain growth in a direction stops as two grain boundaries contact each other, i.e. there is no grain overlapping. The growth process stops, if there is not any further grain growth in any direction in any grain
4. There are no voids, i.e. the entire volume is populated by grains.

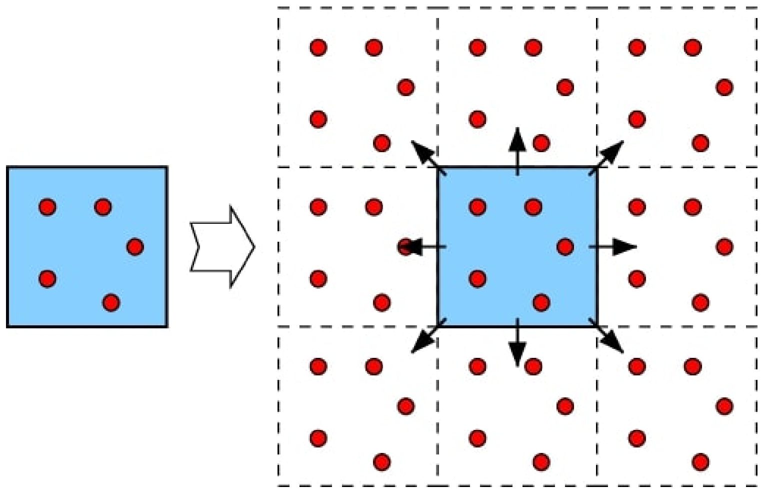


Fig 1 : Construction of a Periodic set of Voronoi vertices for implication of a periodic microstructure

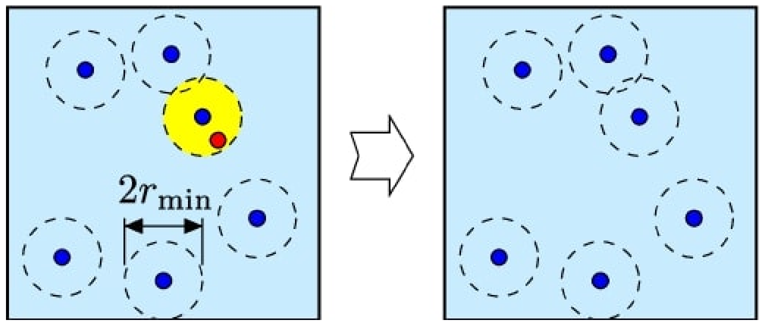


Fig 2 : Hardcore condition for the point seed

Periodic Boundary Conditions

All tesselations can be easily periodized by copying the point seed S around the unit cell as illustrated in two space dimensions in Fig. 1. In this way, we cover all the points and the virtual points thereby not having to implement explicit periodic boundary conditions.

The conditions for equiaxed and elongated grains are trivial. Recall that my assumption of growth of the grains is elliptical. So varying the ratio of the Velocity vectors in X and Y directions yield elongated and when the ratio is 1, we get equiaxed grains.

$$\frac{(x-x_0)^2}{v_x^2}+\frac{(y-y_0)^2}{v_y^2}=1$$

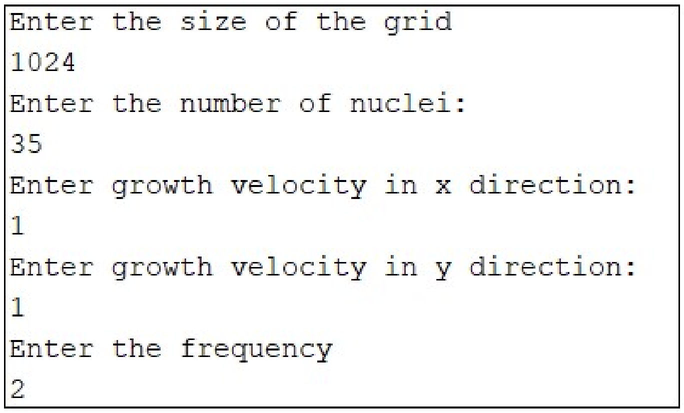
where (x_0, y_0) is the coordinate of the center of the nuclei and the remaining have their usual meanings.

The **MATLAB** code is attached overleaf.

```
% Assignment - 8 : Digital Microstructure - 25/10/2018
% Written by Anirrudh Ramesh, MM16B014, 09:34 AM
x_max = input('Enter the size of the grid\n');
no_nuclei=input('Enter the number of nuclei: \n');
v_x=input('Enter growth velocity in x direction: \n');
v_y=input('Enter growth velocity in y direction: \n');
time = input('Enter the frequency\n');
y_max = x_max;
n = x_max;
grain_id = zeros(n,n);
pos = zeros(no_nuclei,2);
pos2 = zeros(no_nuclei*9,3);
for t=1:no_nuclei
  x_rand = rand()*1e05;
  y_rand = rand()*1e05;
  x1 = mod(round(x_rand),x_max)+1;
  y1 = mod(round(y_rand),y_max)+1;
  grain_id(x1,y1) = t;
  pos(t,1)= x1;
  pos(t,2)= y1;
end
tot_no_nuclei = no_nuclei*9;
counter = 1;
for i = 1 : no_nuclei
    for j = -1 : 1
        for k = -1 : 1
            pos2(counter,1) = i;
            t1 = pos(i,1)+(n*k);
            t2 = pos(i,2)+(n*j);
            pos2(counter,2) = t1;
            pos2(counter,3) = t2;
            counter = counter + 1;
        end
    end
end
for i = 1 : n
    for j = 1 : n
        ref_dist = 8*n^2;
        for k = 1 : tot_no_nuclei
            distance=(((i - pos2(k,2))/(v_x*time)).^2) + ...
                      (((j - pos2(k,3))/(v_y*time)).^2);
            if(ref_dist > distance)
                          ref_dist = distance;
             grain_id(i,j) = pos2(k,1);
```

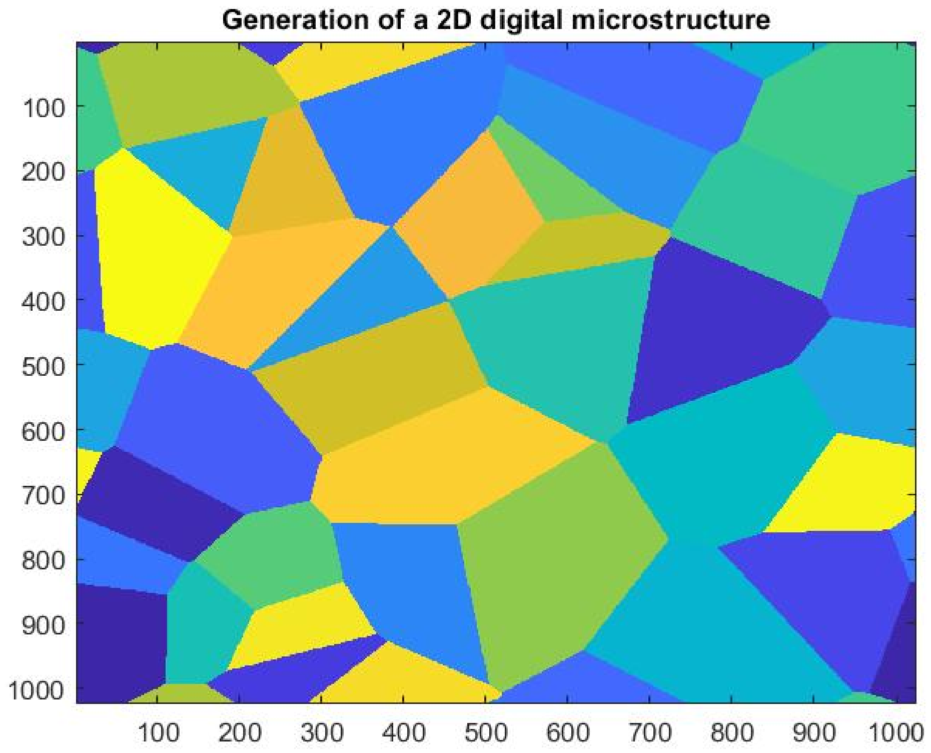
```
                end
            end
        end
end
imagesc(grain_id);
title('Generation of a 2D digital microstructure');
```

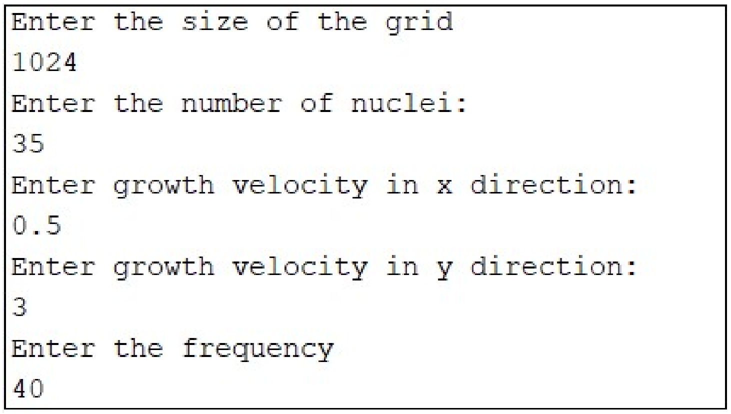
The code yields the following simulation for equiaxed and elongated grains.



```
Enter the size of the grid
1024
Enter the number of nuclei:
35
Enter growth velocity in x direction:
1
Enter growth velocity in y direction:
1
Enter the frequency
2
```

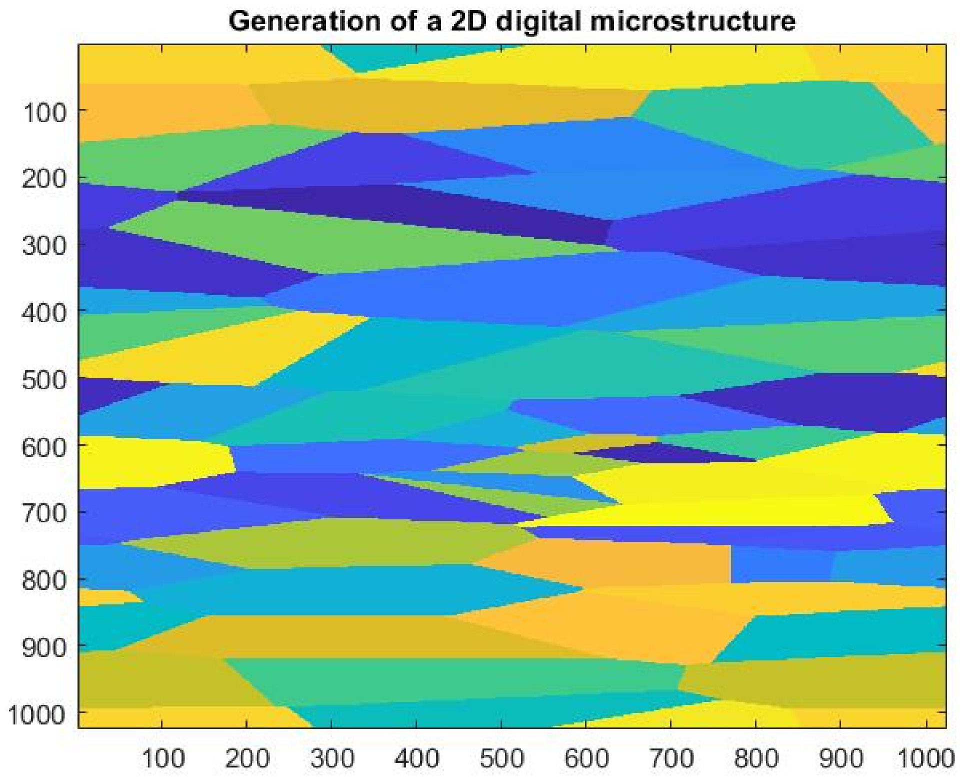
Fig 3 : The values are chosen in order to produce an equiaxed grain





```
Enter the size of the grid
1024
Enter the number of nuclei:
35
Enter growth velocity in x direction:
0.5
Enter growth velocity in y direction:
3
Enter the frequency
40
```

Fig 4 : The values chosen are in order to get a nice elongated grain



2. Using Voronoi tesselation in MATLAB

Voronoi diagram is a partitioning of a plane into regions based on distance to points in a specific subset of the plane. That set of points (called seeds, sites, or generators) is specified beforehand, and for each seed there is a corresponding region consisting of all points closer to that seed than to any other. These regions are called **Voronoi cells**.
The line segments of the Voronoi diagram are all the points in the plane that are equidistant to the two nearest sites. The Voronoi vertices (nodes) are the points equidistant to three (or more) sites. The **MATLAB** code is attached below. It takes the value of 'n' as input for number of grains.

```
% Voronoi Tesselation in MATLAB
n = input('Enter the number of points/nuclei');
x = gallery('uniformdata',[1 n],0);
y = gallery('uniformdata',[1 n],1);
voronoi(x,y)
title('Voronoi partition');
```

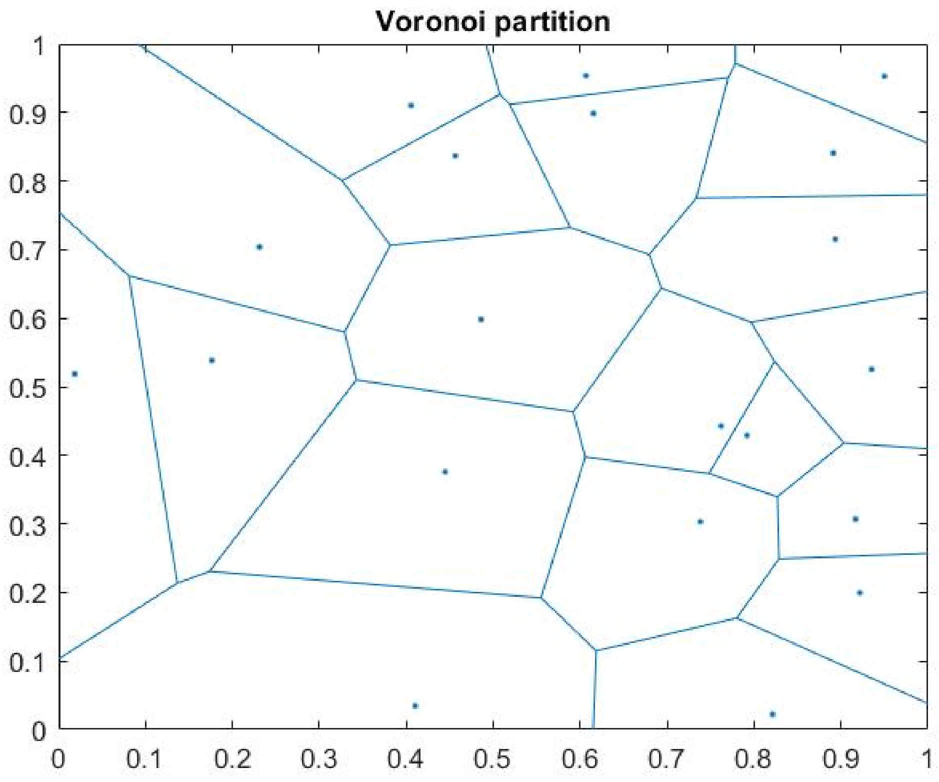


Fig 5 : Voronoi partition for n = 20

As we can see, the partitioning isn't periodic. With an inbuilt MATLAB function, we get only partitioning of space.
Thus we've generated random 2D digital microstructures.

****