MM3110 - Expt 8 - Monte Carlo

Vishal S, IIT Madras

1 Introduction

In this assignment we will try to create an artificial microstructure using `voronoi()` and then evolve the microstructure using Pott's model.

2 Code

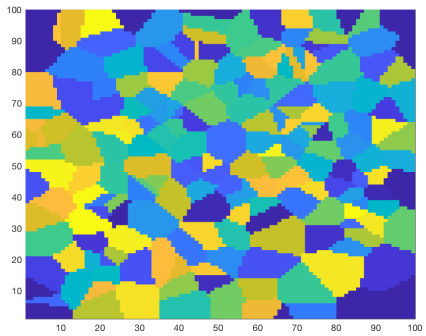
A script is written to generate an artificial microstructure using the `voronoi()` command. This gives the vertices of the polygon as the output. Then a grid is generated and the microstructure is transferred to the grid by figuring out the polygon to which each grid point belongs to using the `inpolygon()` . This is then allowed to evolve as per the Pott's model

```
clear all
clf
clc
no_of_nuclei = 200;
Nx = 100;
Ny = 100;
grain_old = zeros(Nx,Ny);
x = linspace(0,1,Nx);
y = linspace(0,1,Ny);
N = 200;
KbT = 10;
order = [-1,-1;-1,0;-1,1;0,1;1,1;1,0;1,-1;0,-1];
% Create artificial microstructure
seeds = gallery('uniformdata',[no_of_nuclei 2],0);
DT = delaunayTriangulation(seeds);
[v,c] = voronoiDiagram(DT);
[X,Y] = meshgrid(x,y);
% convert the microstructure into grids
for k = 1:length(c)
    if all(c{k}~=1)   % If at least one of the indices is 1,
        % then it is an open region and we can't
        % patch that.
        [in,on] = inpolygon(X,Y,v(c{k},1),v(c{k},2));
        for i = 1:100
            for j = 1:100
                if(in(i,j))
                    grain_old(i,j) = k;
                end
            end
        end
    end
end
% Printing the initial grain srtucture
pcolor(grain_old)
view(2);
pause(1);
shading flat;
str1 = ['initialgrain'];
print(str1,'-dpng');
% Evolve the microstructure
for t = 1:4000
    for m = 1:N
        x = randi([1 Nx],1);
        y = randi([1 Nx],1);
        H1 = calc_H(grain_old,x,y,Nx,Ny);
        temp = grain_old(x,y);
        neighbour = randi([1 8],1);
        i1 = x+order(neighbour,1);
        j1 = y+order(neighbour,2);
        % Periodic boundary condition
```

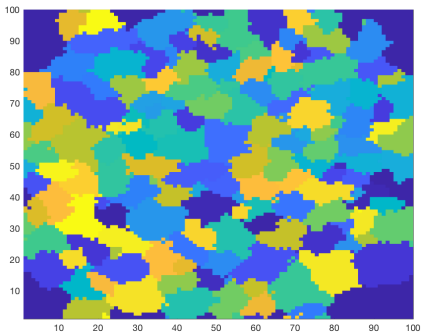
```
        if i1<1
            i1 = i1+Nx;
        end
        if j1<1
            j1 = j1+Ny;
        end
        if i1>Nx
            i1 = i1-Nx;
        end
        if j1>Ny
            j1 = j1-Ny;
        end
        grain_old(x,y) = grain_old(i1,j1);
        H2 = calc_H(grain_old,x,y,Nx,Ny);
        delH = H2-H1;
        prob = exp(-delH/(KbT));
        % check if transition is possible
        trans = rand();
        if(trans>prob)
            grain_old(x,y) = temp;
        end
    end
    % To print the microstructure every 100 time steps
    if(mod(t,200) == 0)
        t
        pcolor(grain_old);
        view(2);
        shading flat;
        str1 = ['grain',num2str(t)];
        print(str1,'-dpng');
        pause(1);
    end
end

%% Function to implement Kronecker Delta
function y = delta(a,b)
if(a==b)
    y = 1;
else
    y = 0;
end
end

%% Function to calculate the energy of the configuration
function y = calc_H(A,i,j,nx,ny)
y = 0;
k = 10;
order = [-1,-1;-1,0;-1,1;0,1;1,1;1,0;1,-1;0,-1];
for l = 1:8
    i1 = i+order(l,1);
    j1 = j+order(l,2);
    if i1<1
        i1 = i1+nx;
    end
    if j1<1
        j1 = j1+ny;
    end
    if i1>nx
        i1 = i1-nx;
    end
    if j1>ny
        j1 = j1-ny;
    end
    y = y + (k/2)*(1-delta(A(i,j),A(i1,j1)));
end
end
```

3 Results

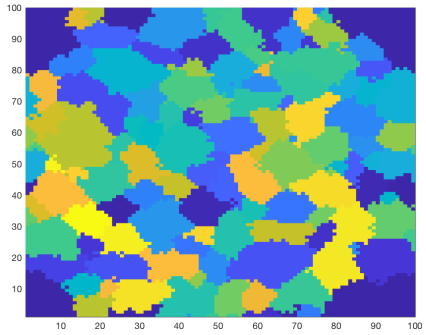
The above code is run and the results obtained are as given in Figures 1



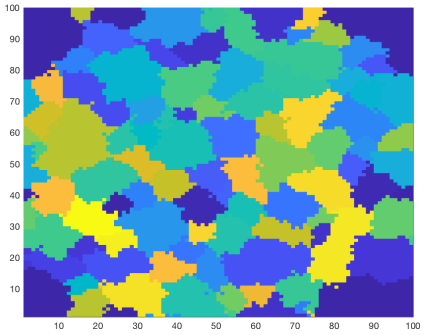
(a) Initial Microstructure



(b) Microstructure after 1000 time steps



(c) Microstructure after 3000 time steps



(d) Microstructure after 4000 time steps

Figure 1: Microstructures