

TASK-3:

dx=d;

dy=d;

dz=d;

if plane_2D=="xy"

if GPU==1

X1=gpuArray(-round(0.5*Max_x/dx+Lx/(2*dx)):round(0.5*Max_x/dx+Lx/(2*dx)));

X_extend=X1*dx;

Y1=gpuArray(-round(0.5*Max_y/dy+Ly/(2*dy)):round(0.5*Max_y/dy+Ly/(2*dy)));

Y_extend=Y1*dy;

Z1=gpuArray(-round(Max_z/(2*dz)):round(Max_z/(2*dz)));

Z_extend=Z1*dz;

elseif GPU==0

X1=-round(0.5*Max_x/dx+Lx/(2*dx)):round(0.5*Max_x/dx+Lx/(2*dx));

X_extend=X1*dx;

Y1=-round(0.5*Max_y/dy+Ly/(2*dy)):round(0.5*Max_y/dy+Ly/(2*dy));

Y_extend=Y1*dy;

Z1=-round(Max_z/(2*dz)):round(Max_z/(2*dz));

Z_extend=Z1*dz;

end

[Yex,Xex]=meshgrid(Y_extend,X_extend);

SIZE=size(Xex);

if GPU==1

Zex=zeros(SIZE(1,1),SIZE(1,2),'gpuArray');

else

Zex=zeros(SIZE(1,1),SIZE(1,2));

end

```

N_plane=SIZE(1,1)*SIZE(1,2); % Total number of voxels within NPs and in outside region

% Nxex=length(X_extend); % Number of voxels in the extended x-axis
% Nyex=Ny;
% Nzex=length(Z_extend); % Number of voxels in the extended z-axis
x_plane=reshape(Xex,[N_plane,1]); % X coordinate of the voxels in the extended x-axis
y_plane=reshape(Yex,[N_plane,1]); % y coordinate of the voxels in the extended y-axis
z_plane=reshape(Zex,[N_plane,1]); % z coordinate of the voxels in the extended z-axis
[y_rec,x_rec,z_rec]=meshgrid(Y_extend,X_extend,Z_extend);
%=====
elseif plane_2D=="xz"

if GPU==1
    X1=gpuArray(-round(0.5*Max_x/dx+Lx/(2*dx)):round(0.5*Max_x/dx+Lx/(2*dx)));
    X_extend=X1*dx;
    Y1=gpuArray(-round(Max_y/(2*dy)):round(Max_y/(2*dy)));
    Y_extend=Y1*dy;
    Z1=gpuArray(-round(0.5*Max_z/dz+Lz/(2*dz)):round(0.5*Max_z/dz+Lz/(2*dz)));
    Z_extend=Z1*dz;

elseif GPU==0
    X1=-round(0.5*Max_x/dx+Lx/(2*dx)):round(0.5*Max_x/dx+Lx/(2*dx));
    X_extend=X1*dx;
    Y1=-round(Max_y/(2*dy)):round(Max_y/(2*dy));
    Y_extend=Y1*dy;
    Z1=-round(0.5*Max_z/dz+Lz/(2*dz)):round(0.5*Max_z/dz+Lz/(2*dz));
    Z_extend=Z1*dz;
end

[Zex,Xex]=meshgrid(Z_extend,X_extend);
SIZE=size(Xex);

```

```

if GPU==1
    Yex=zeros(SIZE(1,1),SIZE(1,2),'gpuArray');
else
    Yex=zeros(SIZE(1,1),SIZE(1,2));
end

N_plane=SIZE(1,1)*SIZE(1,2); % Total number of voxels within NPs and in outside region

% Nxex=length(X_extend); % Number of voxels in the extended x-axis
% Nyex=Ny;
% Nzex=length(Z_extend); % Number of voxels in the extended z-axis

x_plane=reshape(Xex,[N_plane,1]); % X coordinate of the voxels in the extended x-axis
y_plane=reshape(Yex,[N_plane,1]); % y coordinate of the voxels in the extended y-axis
z_plane=reshape(Zex,[N_plane,1]); % z coordinate of the voxels in the extended z-axis
[y_rec,x_rec,z_rec]=meshgrid(Y_extend,X_extend,Z_extend);

%=====
elseif plane_2D=="yz"

if GPU==1
    X1=gpuArray(-round(Max_x/(2*dx)):round(Max_x/(2*dx)));
    X_extend=X1*dx;
    Y1=gpuArray(-round(0.5*Max_y/dy+Ly/(2*dy)):round(0.5*Max_y/dy+Ly/(2*dy)));
    Y_extend=Y1*dy;
    Z1=gpuArray(-round(0.5*Max_z/dz+Lz/(2*dz)):round(0.5*Max_z/dz+Lz/(2*dz)));
    Z_extend=Z1*dz;

elseif GPU==0
    X1=-round(Max_x/(2*dx)):round(Max_x/(2*dx));
    X_extend=X1*dx;
    Y1=-round(0.5*Max_y/dy+Ly/(2*dy)):round(0.5*Max_y/dy+Ly/(2*dy));
    Y_extend=Y1*dy;
    Z1=-round(0.5*Max_z/dz+Lz/(2*dz)):round(0.5*Max_z/dz+Lz/(2*dz));

```

```

    Z_extend=Z1*dz;
end

[Zex,Yex]=meshgrid(Z_extend,Y_extend);
SIZE=size(Yex);
if GPU==1
    Xex=zeros(SIZE(1,1),SIZE(1,2),'gpuArray');
else
    Xex=zeros(SIZE(1,1),SIZE(1,2));
end
N_plane=SIZE(1,1)*SIZE(1,2); % Total number of voxels within NPs and in outside region

%Nxex=length(X_extend); % Number of voxels in the extended x-axis
%Nyex=Ny;
%Nzex=length(Z_extend); % Number of voxels in the extended z-axis
x_plane=reshape(Xex,[N_plane,1]); % X coordinate of the voxels in the extended x-axis
y_plane=reshape(Yex,[N_plane,1]); % y coordinate of the voxels in the extended y-axis
z_plane=reshape(Zex,[N_plane,1]); % z coordinate of the voxels in the extended z-axis
[y_rec,x_rec,z_rec]=meshgrid(Y_extend,X_extend,Z_extend);

end

r_plane=[x_plane y_plane z_plane]; % Position of the each nanocubes inside and outside of
the NPs boundary

% [y_rec,x_rec,z_rec]=meshgrid(Y_extend,X_extend,Z_extend);

Nx=length(X_extend); % Number of the dipoles in the x-direction
Ny=length(Y_extend); % Number of the dipoles in the y-direction
Nz=length(Z_extend); % Number of the dipoles in the z-direction

```

```
N=Nx*Ny*Nz;
```

```
X=reshape(x_rec,[N,1]); % X-coordinates of the dipoles
```

```
Y=reshape(y_rec,[N,1]); % Y-coordinates of the dipoles
```

```
Z=reshape(z_rec,[N,1]); % Z-coordinates of the dipoles
```

```
r_block=[X Y Z]; % Position of the each dipoles inside extended rectangular block
```