**Source Code List**

**'FFM.m'**

clc;

clear all;

nof=1; %number of files

NOE=10; %number of experiments

while (nof<=NOE)

n=500; %size of forest

probTree=0.5; %desity of tree

probIgnite=1; % probability of burning if a neighbouring tree is burnt (homo forest)

probGrow=1E-3; % probalility of growing tree if there is empty site

% Weather conditions:

probLightning=1E-6; %Lightening or spark cause fire although no neighbouring tree is burning

probRain=0; %probability of putting off by rain

beta=0; % wind suppression controller

loop=1; %number of step

N=200; %upper bound of loop

% forest geometry

% i0=n/2; %fire starting point

% j0=n/2; %fire starting point

% initialize forest: 0-space; 1-tree;2-burning tree;3-village

grid1 = zeros(n);

grid1(rand(n) < probTree) = 1;

% grid1(i0,j0)=2; %fire starting point

grid=zeros(n+2); %boundary of forest which can stop the fire

grid(2:n+1,2:n+1)=grid1;

state=zeros(n,n,N);%to save current state of forest

state(:,:,1)=grid(2:n+1,2:n+1);

colormap([0.5,0.5,0.5;0,1,0;1,0,0;0,0,1]); %define colar:0-grey, 1-green, 2-red,3-blue

image(grid(2:n+1,2:n+1)+ones(n)); %draw the initial state of forest

pause(0.1);

while (loop<=N)

space=grid==0; %record position of empty site

havetree=grid == 1; %record position of tree

burnt=grid==2;%record position of burning tree

nfire = circshift(grid, [1 0]) == 2; %check if there is any tree near a burning tree: N-north, S-south, E-east, W-west

sfire = circshift(grid, [-1 0]) == 2;

efire = circshift(grid, [0 -1]) == 2;

wfire = circshift(grid, [0 1]) == 2;

nefire = circshift(grid, [1 -1]) == 2;

sefire = circshift(grid, [-1 -1]) == 2;

swfire = circshift(grid, [-1 1]) == 2;

nwfire = circshift(grid, [1 1]) == 2;

susceptible\_n = rand(n+2) < probIgnite\*exp(-0\*beta)\*(1-probRain);

susceptible\_ne = rand(n+2) < probIgnite\*exp(-pi/4\*beta)\*(1-probRain);

susceptible\_nw = rand(n+2) < probIgnite\*exp(-pi/4\*beta)\*(1-probRain);

susceptible\_e = rand(n+2) < probIgnite\*exp(-pi/2\*beta)\*(1-probRain);

susceptible\_w = rand(n+2) < probIgnite\*exp(-pi/2\*beta)\*(1-probRain);

susceptible\_se = rand(n+2) < probIgnite\*exp(-3\*pi/4\*beta)\*(1-probRain);

susceptible\_sw = rand(n+2) < probIgnite\*exp(-3\*pi/4\*beta)\*(1-probRain);

susceptible\_s = rand(n+2) < probIgnite\*exp(-pi\*beta\*(1-probRain));

grid(havetree & (nfire & susceptible\_n | sfire & susceptible\_s | wfire & susceptible\_w | efire & susceptible\_e | nefire & susceptible\_ne | nwfire & susceptible\_nw | sefire & susceptible\_se | swfire & susceptible\_sw)) = 2;

grid(havetree & (rand(n+2) < probLightning))=2; %Lightening or spark cause fire although no neighbouring tree is burning

grid=grid-2\*burnt; %burnt trees fall down become empty sites

grid(space & (rand(n+2)<probGrow))=1; %trees can grow on empty sites

loop=loop+1;

state(:,:,loop)=grid(2:n+1,2:n+1);%save the current state of forest

image(grid(2:n+1,2:n+1)+ones(n)); %draw the current state of forest

pause(0.1);

end

%count\_writer(state,nof,n);

%state\_writer(state,k,nof);

nof=nof+1;

end

%videomaker(state);

**'count\_writer.m'**

function count\_writer(state,nof,n);

[b1 b2 b3]=size(state);

filename=['E:\matlab project\Data\exp\_20\count\_',num2str(nof),'.txt'];

fid=fopen(filename,'w');

Data=zeros(b3,4);

for k=1:b3

Grey=state(2:(b1-1),2:(b2-1),k)==0;

Green=state(2:(b1-1),2:(b2-1),k)==1;

Red=state(2:(b1-1),2:(b2-1),k)==2;

Data(k,1)=k;

Data(k,2)=length(find(Grey))/n/n;

Data(k,3)=length(find(Green))/n/n;

Data(k,4)=length(find(Red))/n/n;

end

for i=1:b3

for j=1:4

fprintf(fid,'%d\t',Data(i,j));

end

fprintf(fid,'\n');

end

fclose(fid);

end

**'state\_writer.m'**

function state\_writer(state,k,nof)

[b1 b2 b3]=size(state);

filename=['E:\matlab\project\Data\exp\_21\state\_',num2str(nof),'\_',num2str(k),'.txt'];

fid=fopen(filename,'w');

for m=1:b1

for n=1:b2

fprintf(fid,'%d\t',state(m,n,k));

end

fprintf(fid,'\n');

end

fclose(fid);

end

**'plotcount.m'**

clear all

clc

M=load(['E:\matlab project\Data\exp\_20\count\_1.txt']);

figure;

subplot(2,1,1);

plot(M(:,1),M(:,2),'b',M(:,1),M(:,3),'g');

axis([0,1000,0,1]);

legend('empty','tree','fire');

subplot(2,1,2);

plot(M(:,1),M(:,4),'r');

legend('fire');

axis([0,1000,0,0.01]);

**'scan3d.m'**

clc

clear all

for i=1:101

t=0+(i-1)\*0.0000001;

filename=['E:\matlab project\Data\exp\_18\count\_',num2str(t),'.txt'];

data{i}=load(filename);

end

x=0.00:0.0000001:0.00001;

y=data{1}(:,1);

k=length(y);

for i=1:101

for j=1:k

z(j,i)=(data{i}(j,2));

end

end

surf(x,y,z);

shading flat;

axis([0.0000001,0.00001,1,3000]);

xlabel('probLightening');

ylabel('Step');

zlabel('Fire');

**'videomaker.m'**

function videomaker(state)

[b1 b2 b3]=size(state);

aviobj=avifile('E:\matlab\project\video\test.avi','Compression','None','fps',5);

hax=axes;

colormap([0.5,0.5,0.5;0,1,0;1,0,0;0,0,1]);

for k=1:b3

image(state(2:(b1-1),2:(b2-1),k)+ones(b1-2),'parent',hax);

frame = getframe(gcf);

aviobj=addframe(aviobj,frame);

end

aviobj=close(aviobj);

end

**'sizedistribution\_plot.m'**

clc

clear all

M1=load(['E:\matlab project\Data\exp\_21\state\_10\_1.txt']);

M2=load(['E:\matlab project\Data\exp\_21\state\_10\_55.txt']);

initial=M1==0;

final=M2==0;

S=final-initial;

TREE=S==-1;

BURNT=S+TREE;

spy(BURNT);

distribution\_writer(BURNT);

**'distribution\_writer.m'**

function distribution\_writer(Y)

filename=['E:\matlab project\Data\exp\_21\size\_dis\_10.txt'];

fid=fopen(filename,'w');

[b1 b2]=size(Y);

for m=1:b1

for n=1:b2

fprintf(fid,'%f\t',Y(m,n));

end

fprintf(fid,'\n');

end

fclose(fid);

end

**'sizedistribution\_calculator'**

clear all;

clc;

M=load(['E:\matlab project\Data\exp\_21\size\_dis\_10.txt']);

global r;

global c;

set(0,'RecursionLimit',2000);

r=zeros(500^2,1);

c=zeros(500^2,1);

[r,c,v]=find(M);

SIZE=zeros(length(M));

Length\_max=length(r);

for i=1:length(M)

for j=1:length(M)

countsize(M,i,j);

SIZE(i,j)=Length\_max-nnz(r);

Length\_max=nnz(r);

end

end

[r1,c1,v1]=find(SIZE);

filename=['E:\matlab project\Data\exp\_21\histogram\_10.txt'];

fid=fopen(filename,'w');

for i=1:length(v1);

fprintf(fid,'%d\n',v1(i:1));

end

fclose(fid);

**'countsize.m'**

function countsize(M,i,j)

global r;

global c;

marker=checklist(i,j);

if marker==1

if M(i,j)==1

countsize(M,i,j+1);

countsize(M,i,j-1);

countsize(M,i+1,j);

countsize(M,i-1,j);

countsize(M,i+1,j+1);

countsize(M,i+1,j-1);

countsize(M,i-1,j+1);

countsize(M,i-1,j-1);

end

end

end

**'checklist.m'**

function marker=checklist(i,j)

global r;

global c;

marker=0;

m=1;

while marker==0 & m<=length(r)

if i==r(m) & j==c(m)

marker=1;

r(m)=0;

c(m)=0;

else marker=0;

end

m=m+1;

end

end

**'histogram.m'**

clc

clear all

A=load(['E:\matlab project\Data\exp\_21\histogram.txt']);

bin=20;

hist(A,bin);

Y=hist(A,bin)./length(A);

incre=(max(A)-min(A))/(bin-1);

X=(min(A)+incre/2):incre:(max(A)+incre/2);

loglog(X,Y,'o');