

6. Spreading Phenomena

From Chapter 4 of "Computer Simulations with Mathematica", Gaylord & Wellin, Springer 1995 [DCU library ref 510.2855GAY]

■ 6.1 Examples of Spreading

- 1) Flow of a drink that has been knocked over
- 2) Tumour growth
- 3) Epidemic spread

■ 6.2 Basic model (due to Eden)

Operates on a two-dimensional square lattice system

Have a **cluster** list and a **perimeter** list

At start:

Cluster list consists of a single *seed* site located at the origin

Perimeter list consists of the 4 nearest neighbours to the seed site [above (N), right (E), below (S), left (W)]

Procedure:

- 1. Select a site randomly from the **perimeter** list and move it to the **cluster** list
- 2. Determine the nearest neighbours to the selected site and add those nearest neighbour sites not already in the **cluster** or **perimeter** lists to the **perimeter** list
- Steps 1 and 2 are repeated until the **cluster** list reaches a certain size, say n .

■ 6.3 Single Percolation Cluster Model (variation of basic model applicable to epidemics)

■ 6.3.1 Outline of procedure or How model works

- 1. Select a site randomly from the **perimeter** list
Decide with probability **p** whether the selected site will be moved to the **cluster** list
Even if selected site is not moved to the **cluster**, remove it from the **perimeter**
 - 2. Determine the nearest neighbours to the selected site (if it has not been rejected) and add those nearest neighbour sites not already in the **cluster** or **perimeter** lists (or already rejected) to the **perimeter** list
 - Steps 1 and 2 are repeated until the **cluster** list reaches a certain size, say **n**.
- Note: This variation specialises to the basic model when **p** = 1.

■ 6.3.2 Specific Algorithm

Step 1: Create an ordered pair, consisting of a **cluster** list containing the seed site and a **perimeter** list of its nearest neighbours:

$$\{\{0, 0\}, \{0, 1\}, \{1, 0\}, \{0, -1\}, \{-1, 0\}\}$$

Also, create an empty list called **reject**

Step 2: Randomly choose a site in the **perimeter** list

Step 3: Generate a random number and compare it with the input value **p**. If less than or equal to **p**, follow Steps 3.A.i and 3.A.ii; otherwise follow Steps 3.B.i and 3.B.ii.

3.A.i: Determine the nearest neighbours to the selected site and add those nearest neighbour sites not already in the **cluster**, **perimeter**, or **reject** lists to the **perimeter** list.

Also, remove the selected site from the **perimeter** list.

3.A.ii: Place the selected site in the **cluster** list and compute the ordered pair consisting of the new **cluster** and **perimeter** lists.

3.B.i: Place the selected site in the **reject** list

3.B.ii: Remove the selected site from the **perimeter** list and compute the ordered pair consisting of the old **cluster** and the new **perimeter** lists.

Step 4:

Execute the sequence of Steps 1 to 3 are repeated until the **cluster** list reaches a certain size, say n , or until the **perimeter** list becomes empty.

■ 6.3.3 *Mathematica* implementation of the algorithm [Don't worry about coding details]

```
In[3]:= Epidemic[n_, p_] :=
Module[{choices = {{0, 1}, {1, 0}, {0, -1}, {-1, 0}},
  reject, pickAndChoose, select, newPers},
  reject = {};
  pickAndChoose :=
  (select = #[[2, Random[Integer, {1, Length#[[2]]}]]];
   If[Random[] ≤ p,
     newPers =
      Complement[Union[Map[Function[y, y + select],
        choices], #[[2]]],
        {select}, #[[1]], reject];
     {Join#[[1]], {select}, newPers},
     reject = Join[reject, {select}];
     {#[[1]], Complement#[[2], {select}}] &;
  FixedPoint[pickAndChoose, {{0, 0}}, choices], n,
  SameTest → (#2[[2]] == {} ||
    Length[#2[[1]]] == n &)] [[1]]
]
```

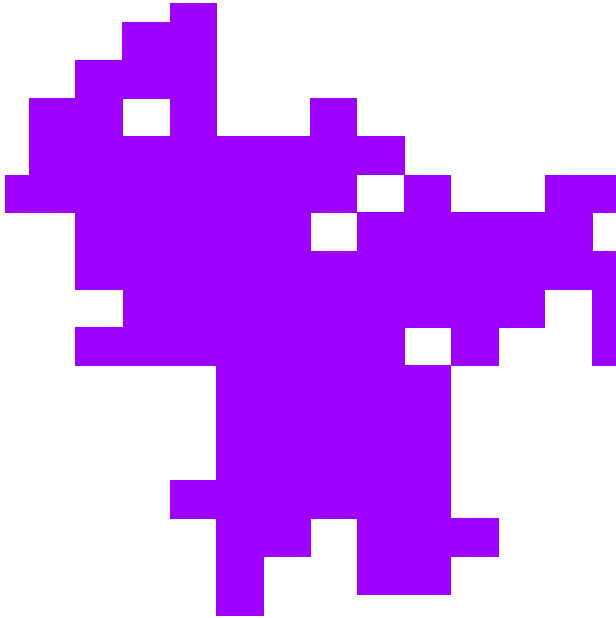
■ 6.3.4 Code for graphical output of "Epidemic" [Don't worry about coding details]

```
In[4]:= ShowSpread[list_, opts___] :=
Show[Graphics[{Hue[0.77],
  Map[(Rectangle[# - {0.5, 0.5},
    # + {0.5, 0.5}]) &, list]}],
  opts,
  AspectRatio → 1,
  PlotRange → Map[({Min[#], Max[#]}) &, Transpose[list]]]
```

■ 6.3.5 Results from some example runs

■ 6.3.5.1.a Basic Model ($p = 1$) until cluster length is 100

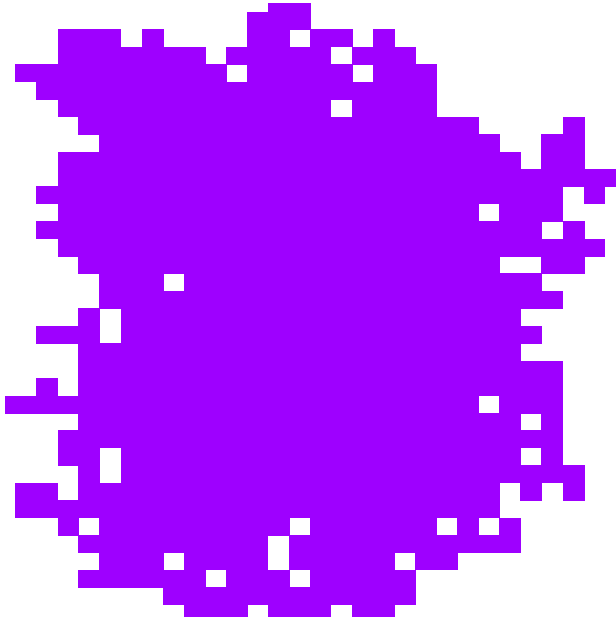
```
In[17]:= ShowSpread[Epidemic[100, 1.0]]
```



```
Out[17]= - Graphics -
```

■ 6.3.5.1.b Basic Model ($p = 1$) until cluster length is 700

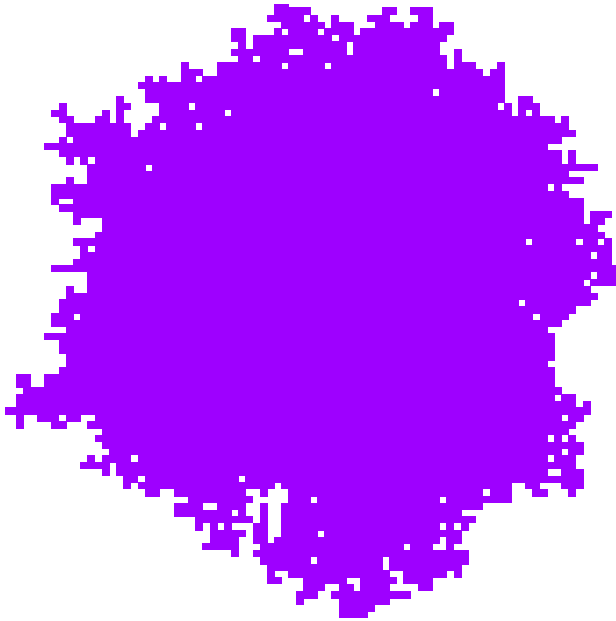
```
In[18]:= ShowSpread[Epidemic[700, 1.0]]
```



```
Out[18]= - Graphics -
```

■ 6.3.5.1.c Basic Model ($p = 1$) until cluster length is 4900

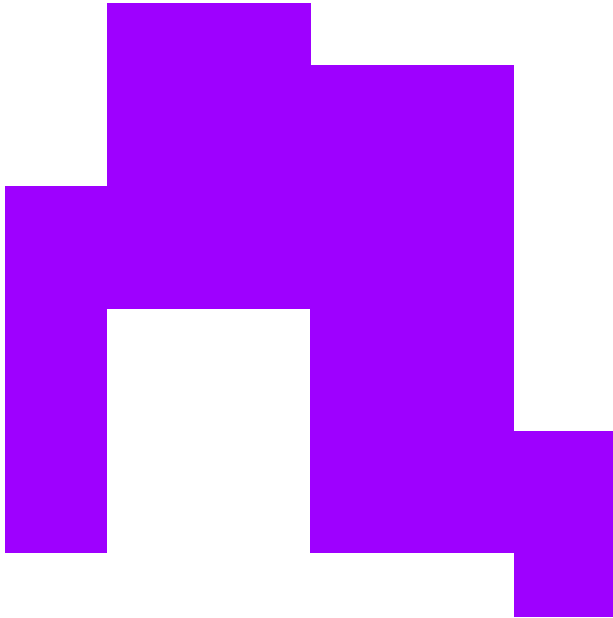
```
In[19]:= ShowSpread[Epidemic[4900, 1.0]]
```



```
Out[19]= - Graphics -
```

■ 6.3.5.2.a Epidemic Model with $p = 0.58$ until cluster length is 100

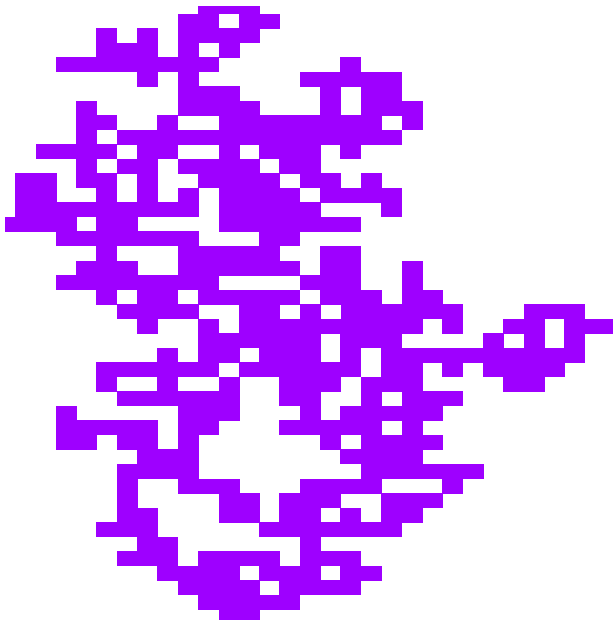
```
In[23]:= ShowSpread[Epidemic[100, 0.58]]
```



```
Out[23]= - Graphics -
```

■ 6.3.5.2.b Epidemic Model with $p = 0.58$ until cluster length is 700

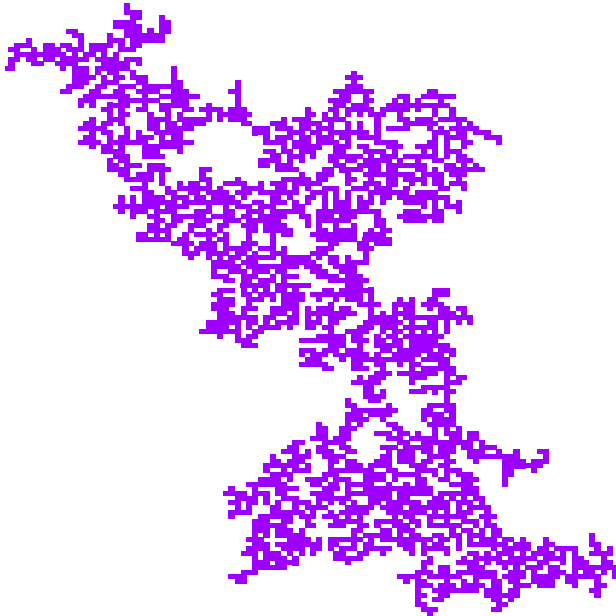
```
In[21]:= ShowSpread[Epidemic[700, 0.58]]
```



```
Out[21]= - Graphics -
```

■ 6.3.5.2.c Epidemic Model with $p = 0.58$ until cluster length is 4900

```
In[24]:= ShowSpread[Epidemic[4900, 0.58]]
```



```
Out[24]= - Graphics -
```

■ 6.3.5.3.a Epidemic Model with $p = 0.30$ until cluster length is 100

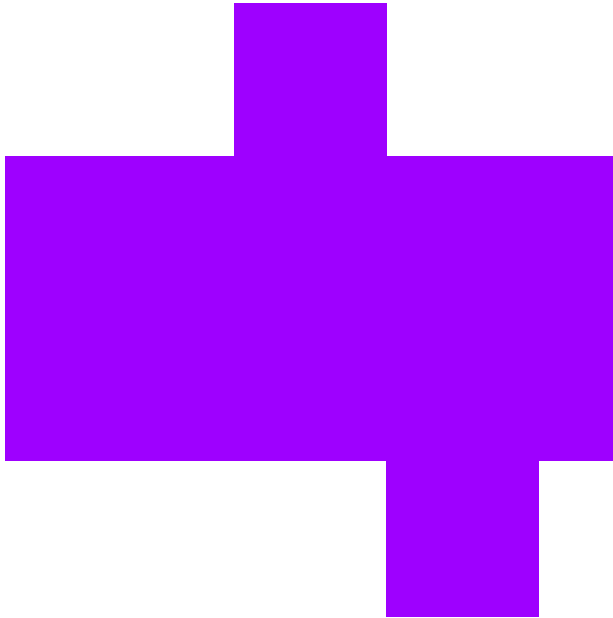
```
In[25]:= ShowSpread[Epidemic[100, 0.30]]
```



```
Out[25]= - Graphics -
```

■ 6.3.5.3.b Epidemic Model with $p = 0.30$ until cluster length is 700

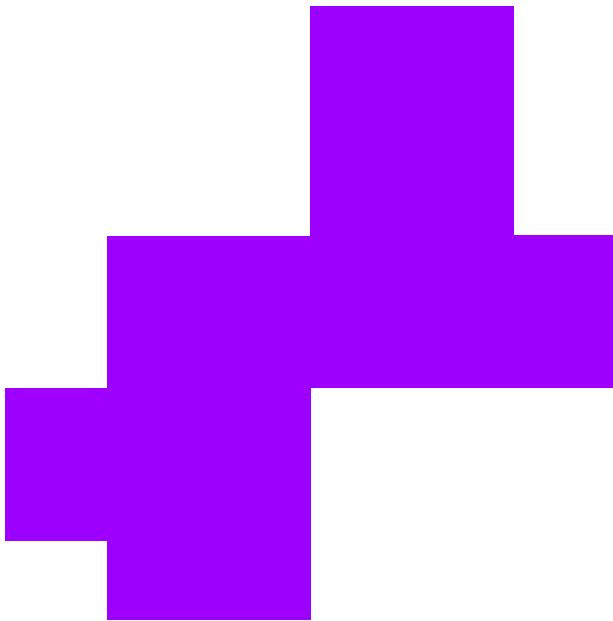
```
In[26]:= ShowSpread[Epidemic[700, 0.30]]
```



```
Out[26]= - Graphics -
```

■ 6.3.5.3.c Epidemic Model with $p = 0.30$ until cluster length is 4900

```
In[27]:= ShowSpread[Epidemic[4900, 0.30]]
```



```
Out[27]= - Graphics -
```