Diffusion Limited Aggregation(유한 확산 집합체)

목적 : 프랙탈의 차원과 random walk, 프랙탈의 한 종류인 유한 확산 집합체(DLA)를 이해하고 DLA를 형성하는 프로그램을 만들어 그 차원을 계산해 본다.

1. 이론

1)프랙탈(fractal)의 정의
프랙탈의 첫 번째 특징은, 위에서 언급한 여러 현상들의 복잡한 형태가 형성하는 기본 구성요소(fundamental building blocks)의 분포(distribution)가 아래의 멱법칙(power law)을 따른다. 임의의 한 점을 중심으로 반경 R인 원 내부에 존재하는 입자의 수를 N(R)이라 할 때, 그 입자의 평균 은 $< N(R) >= AR^{d_f}$ 을 만족하며, 이 식을 만족하는 분포를 하는 어떤 물리량이의 분포의 가 그 분포가 존재하는 바닥 공간의 위상차원 보다 작은 경우(즉, 인 경우)를 이러한 분포를 프랙탈 분포(fractal distribution) 또는 단순히 프랙탈(fractal)이라 부른다. 이것 이외에 프랙탈을 정의하는 또 하나의 특징은 프랙탈의 구조가 자기상자성(Self-similarity, 자기 유사성)를 가지고 있어야 한다. 위에 말은 프랙탈 구조의 일부분을 확대했을 때, 확대된 부분에서 전체의 구조를 다시 볼 수 있음을 뜻한다.

2)프랙탈 차원
유클리드 기하학에서 점은 0차원, 직선은 1차원, 평면은 2차원, 공간은 3차원이라고 한다. 또한 우리는 여러 가지 양, 즉 길이나 면적, 부피 등을 계산하는 법을 배운다. 이러한 여러 가지 양의 크기를 통틀어 측도라고 말한다. 1차원 도형의 측도는 '길이'이며, 2차원 도형의 측도는 '넓이'이다. 이처럼 도형은 그 차원에 따라 측도가 달라진다. 1차원 도형은 길이만이 관심의 대상이 된다. 예를 들어, 실이나 열차는 그 길이만이 문제이고, 그 굵기는 무시된다. 2차원의 도형은 가로와 세로의 길이, 즉 넓이가 문제가 된다. 옷감이나 종이는 특별한 경우가 아니면 두께는 문제 삼지 않고 넓이만을 주로 문제 삼는다. 차원이 다른 도형을 확대할 때, 그 크기, 즉 측도가 달라진다. 가령 일정한 길이의 1차원 도형인 선분을 3배로 확대하면 그 길이는 그대로 3배가 된다. 그러나 2차원 도형인 정사각형을 3배로 확대하면 넓이는 3배가 아니라 9배가 되며, 3차원 도형인 정육면체의 경우에는 3배로 확대하면 부피는 무려 27배로 늘어난다. 단순히 3배로 확대했을 뿐인데 이렇게 차원에 따라 그 크기는 3배, 9배, 27배로 달라지는 것이다.
여기서 어떤 법칙성을 발견할 수 있다. 확대의 비율 3의 지수로서 각 도형의 차원을 잡으면 다음 관계가 성립함을 누구나 쉽게 짐작할 수 있다.
3=3(1차원)
3=9(2차원)

3=27(3차원)
여기서 이 등식들의 양변에 로그를 취하면 다음과 같은 도형의 차원을 구하는 식을 유도할 수 있다.

$$log3 = log3 \rightarrow 1 = \frac{\log 3}{\log 3}$$

$$log3 = log9 \rightarrow 2 = \frac{\log 9}{\log 3}$$

$$log3 = log27 \rightarrow 3 = \frac{\log 27}{\log 3}$$

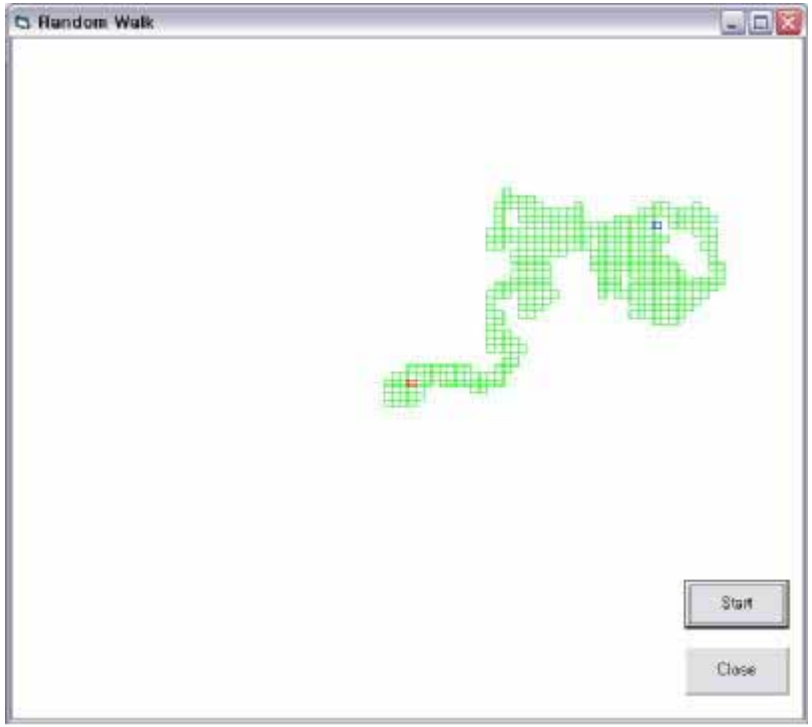
즉,

$$\text{차원} = \frac{\log \text{측도}}{\log \text{확대율}}$$

인 것이다. 그러니까 차원은 복잡성의 정도를 나타낸다는 것을 알 수 있다.
이것이 용량 차원인데 좀 더 정확하게 말하자면, 우선 1차원의 선분을 생각해 보자. 이 선분을 길이가 ε인 작은 길이 요소로 덮는 데 필요한 길이 요소의 개수를 N(ε)이라고 표시하자. 길이 요소를 가능한 작게 만들 때(ε→0) 길이 요소의 개수는 N(ε) 1/ε 이라고 말할 수 있다. 이제 2차원의 면적을 생각해 보자. 이 면적을 한 변의 길이가 ε인 작은 벽돌로 덮는다고 하자. 1차원의 경우와 마찬가지로, 주어진 면적을 전부 덮는데 필요한 면적 요소의 개수는 ε이 작아질 때 N(ε) 1/ε²이 된다. 이 아이디어를 차원이 d인 기하학적 구조에 대하여 다음과 같이 일반화시키면 위 식을 차원 d에 대해 풀어 쓰면 다음과 같이 쓸 수 있다

$$N(\varepsilon) \approx 1/\varepsilon^{d} \ , \ \varepsilon \rightarrow 0 \quad d = \lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log (1/\varepsilon)}$$

3) Random walk(멋대로 걷기)
DLA(확산-제한 응집체) model을 성장(growth) 시키기 위해서는, 확산(diffusion, 퍼짐)현상을 통해 입자가 움직이므로 이러한 확산 현상을 잘 설명하는 통계역학적 모형인 Random walk model을 사용한다. 편향성(bias)이 없는 현상의 무작위적(random), 무질서적(disordered), 비평형적(nonequilibrium) 운동성을 확률론적인 입장에서 설명을 시도할 때 필연적으로 나타나는 기본적인 모형이기 때문이다.
다음 그림은 Random walk를 computer simulation 한 것이다. 시작점이 빨간 사각형이며 도착점은 파란 사각형으로 나타내었다.



Random walk by computer simulation method

4)Diffusion-Limited Aggregates (DLA)

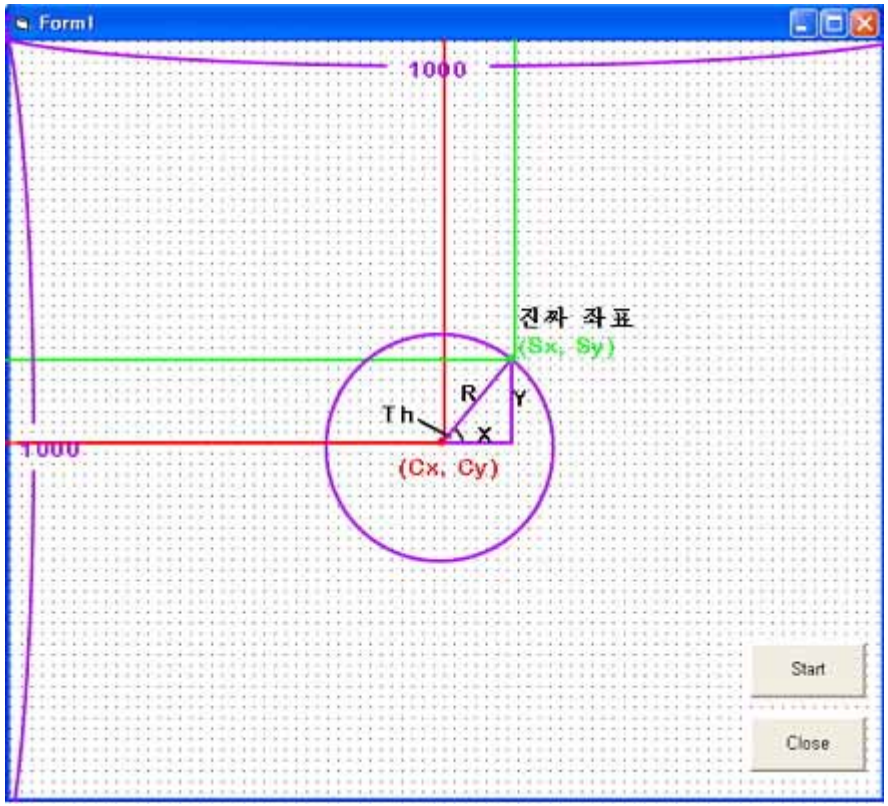
DLA란 무작위적(random), 비평형적(nonequilibrium), 비가역적(irrevirsible) 물질의 성장(growth)을 대표하는 프랙탈 응집체(fractal aggregates) 중 가지치기(ramified pattern) 구조를 갖는 물질 성장을 대표하는 모형이다. 이러한 가지치기(ramified pattern) 구조의 특징은 내부와 외부를 구별하기 힘든 것이 그 특징이며, 번개와 유전체 파괴(dielectric breakdown) 가 일어날 때의 모양과 금속 재료 등에서 나타나는 크랙 전파(crack propagation), 생물학에서의 Bacteria colony의 성장, 나무 가지 성장 등등에서 볼 수 있다. DLA를 성장 시키는 방법은 다음과 같다. 일단 중심을 원점으로 하는 seed particle(씨앗 입자)를 놓은 다음 그 입자를 중심으로 하는 원(반경 Rs - starting radius, 프로그램에선 seed 반경의 7배)으로부터 한 점을 무작위로 잡아, 그 점에서부터 random walk(멋대로 걷기)를 시행한다. 이 입자가 seed particle에 충분히 붙을 수 있는 거리에 오게 되었을 때 흡착시킨다. 이러한 과정을 반복하여 성장을 지속시키며, 정해진 수의 입자가 붙었을 경우 성장을 정지시킨다.

성장시키는 과정에 있어서, random walk의 특성상 입자가 일정한 반경(Rk - Killing radius)에 도달하게 되면 이 입자를 버리고, 새로운 입자를 출발시킨다. 또한 seed particle에 계속하여 입자가 붙게 되므로 Aggregates의 최대반경을 고려하여 입자가 출발하는 원의 반경을 점점 크게 만들어 준다.

2. 프로그램 설명

먼저 변수부터 지정해 준다. 이 Form의 전체는 9000 by 9000인데, 이것을 (1000, 1000)으로 잡는다. Th는 입자의 random-work를 출발시키는 원의 중심각이고, X는 원 위에서 출발하는 입자의 원 중심에서부터 입자가 위치하는 x좌표까지의 길이, Y는 원 위에서 출발하는 입자의 원 중심에서부터 입자가 위치하는 y좌표까지의 길이, Q는 중심과 붙은 입자 사이의 거리, I는 한 입자를 얼마나 random-work 시킬지 루프를 돌려주는 변수이고 R은 입자가 발생하는 반경, D는 Q의 값에서 정수부분만 추출한 값을 나타내주는 변수이다.
그리고 K는 난수를 발생시키는 변수이고 Cx, Cy는 Form의 중심, Sx, Sy는 계산에 직접 참여하는 변수이다. info는 결과를 레이블에 나타내주는 변수이다.

```
Private Sub command1_Click()
Dim P(1000, 1000) As Single
Dim Th, X, Y, Q As Double
Dim I, R, D, Count As Integer
Dim K As Double
Dim Cx, Cy, Sx, Sy, Max As Integer
Dim info As String
```



이 Form의 전체는 9000 by 9000인데, 이것을 (1000, 1000)으로 잡았으므로 중심좌표는 (500, 500)이 된다. 즉 한 칸당 9인 것이다.

Cx = 500
Cy = 500
중심에 seed를 두어야 하므로
P(Cx, Cy) = 1
일단 각 합들을 초기화 한다.
N = 0
Count = 0

이제 난수를 발생시킨다.

Randomize

그리고 가운데 seed를 빨간색으로 표시한다. 점으로 표시해야 하지만 여기서는 아주 작은 사각형으로 표시했다.
Form1.Line (Cx * 9, Cy * 9)-((Cx + 1) * 9, (Cy + 1) * 9), RGB(255, 0, 0), B

10000개의 입자를 성장시키기 위해 Do While 구문을 사용한다. N이 9999보다 커지면 프로그램이 끝날 것이다.
Do While N < 9999

이제 seed를 중심으로 7보다 큰 원 위에서 입자를 random하게 출발시킨 후 random work시킨다. (0, 1)사이의 임의의 값인 난수를 정의하고, 처음 출발할 땐 I=1일 때므로
K = Rnd
I = 1
여기서 Max는 입자가 붙었을 경우 반경을 늘리기 위한 것이고, Th는 seed를 중심으로 하는 원의 중심각인데, 이 각 또한 random 하게 해야 하므로 난수 K를 곱해준다. 그리고 입자의 위치는 XY 좌표로 나타내준다. X, Y는 Cx, Cy부터 시작된 숫자이므로 Cx, Cy를 더해주어 진짜 좌표값 Sx, Sy를 표시해준다.
R = 7 + Max
Th = 6.283184 * K
X = R * Cos(Th)
Y = R * Sin(Th)
Sx = Cx + Int(X)
Sy = Cy + Int(Y)
그리고 seed에 아주 작은 사각형을 그렸던 것과 마찬가지로 입자도 초록색 사각형으로 그려준다.
Form1.Line (Sx * 9, Sy * 9)-((Sx + 1) * 9, (Sy + 1) * 9), RGB(0, 255, 0), B
Form1.Line (Sx * 9, Sy * 9)-((Sx + 1) * 9, (Sy + 1) * 9), RGB(255, 255, 255), B

이제 원 위에서 랜덤하게 발생된 입자를 random-work 시켜보자.
입자의 운동 방향은 상하좌우 4방향이므로 난수를 4개로 나누면 된다. random-work는 900번 시키고
Do While I < 9000
난수가 0.25보다 작을 땐 오른쪽으로 한칸, 0.25와 0.5보다 사이일 땐 왼쪽으로 한칸, 0.5와 0.75 사이일 땐 아래로 한칸, 0.75보다 클 땐 위로 한칸 이동한다.
K = Rnd
If K <= 0.25 Then
Sx = Sx + 1
ElseIf K > 0.25 And K <= 0.5 Then

```
Sx = Sx - 1
ElseIf K > 0.5 And K <= 0.75 Then
Sy = Sy + 1
Else
Sy = Sy - 1
End If
```
그리고 이것을 마찬가지로 작은 초록색 사각형으로 그린다.
```
Form1.Line (Sx * 9, Sy * 9)-((Sx + 1) * 9, (Sy + 1) * 9), RGB(0, 255, 0), B
Form1.Line (Sx * 9, Sy * 9)-((Sx + 1) * 9, (Sy + 1) * 9), RGB(255, 255, 255), B
```

이제 random-work하는 입자가 seed와 만났을 때 seed 주변에 붙는 것을 만들어보자. 만약 입자가 seed 바로 오른쪽에 오면 입자의 P(Sx, Sy)도 1이 되고 왼쪽에 와도 마찬가지이고, 아래에 와도, 위에 와도 1이 된다고 정의해준다.
```
If P(Sx + 1, Sy) = 1 Then
P(Sx, Sy) = 1
ElseIf P(Sx - 1, Sy) = 1 Then
P(Sx, Sy) = 1
ElseIf P(Sx, Sy + 1) = 1 Then
P(Sx, Sy) = 1
ElseIf P(Sx, Sy - 1) = 1 Then
P(Sx, Sy) = 1
Else
End If
```
이제, 입자가 붙었을 경우 입자를 그 위치에 멈추게 하고 붙은 입자는 빨간색으로 바꿔준다. 일단 붙으면 random-work를 하던 입자는 멈춰야 하니까 I=9000이 되고 새로운 입자를 출발시켜야 하니까 N=N+1이다. 일단, seed와 붙은 입자 사이의 거리를 측정한다.
붙지 않았을 경우에 I는 그냥 가던대로 I+1이 되고 seed와 붙은 입자 사이의 거리는 붙은 입자가 없으므로 0이다.
```
If P(Sx, Sy) = 1 Then
Q = Sqr((Sx - Cx) ^ 2 + (Sy - Cy) ^ 2)
I = 9000
N = N + 1

Form1.Line (Sx * 9, Sy * 9)-((Sx + 1) * 9, (Sy + 1) * 9), RGB(255, 0, 0), B
Else
I = I + 1
```

```
Q = 0
End If
```

그리고 입자의 Killing Time을 정해준다. 입자가 원래 출발하던 반경보다 2배가 멀어졌을 때 그 입자는 없애버리고 다시 새로운 입자를 출발시키는 것이다. 방향은 4개가 있고, 역시 random-work를 멈춰야 하므로 각각 4개의 경우에 대응하여 I=9000이 된다고 하면 된다.

```
If (Sx - Cx) / R < -2 Or (Sx - Cx) / R > 2 Then
I = 9000
ElseIf (Sy - Cy) / R < -2 Or (Sy - Cy) / R > 2 Then
I = 9000
ElseIf Sx > 997 Or Sx < 2 Then
I = 9000
ElseIf Sy > 997 Or Sy < 2 Then
I = 9000
End If

Loop
```

이제 입자가 붙었을 경우 반경을 늘려보자. 위에서 seed와 붙은 입자 사이의 거리를 측정하기 위해 루트를 취해주었으므로 소수가 나올 가능성이 있는데 우리는 그 길이만큼 반경을 넓힐 것이기 때문에, VB에서 길이를 인식할 수 있도록 정수값을 취하도록 만든다.

```
D = Int(Q)
```

D는 오직 길이의 크기만을 따지기 때문에 D가 음수의 경우를 대비하여 절대값을 취하게 만들어준다.

```
If D < 0 Then
D = -1 * D
End If
```

그리고 본격적으로 반경을 늘려준다. R=7+Max인데, Max 초기값은 0이다. 이제 D가 Max보다 크다면 Max 자리에 D를 넣어준다는 코드를 써넣자.

```
If D > Max Then
Max = D
End If
```

그리고 혹시 입자가 한쪽으로만 기형적으로 자라게 되어 Form의 영역을 벗어날 수도 있으므로 Max가 400보다 크면 입자를 그만 출발시키라는 명령을 넣어준다.

```
If Max >= 400 Then
```

```
N = 9999
End If

Loop
```

이제 입자는 다 출발시켰고 레이블에 결과를 출력하는 코드만 써넣으면 된다. 반경이 10증가할 때마다 붙은 입자의 개수를 출력하게 만들기 위해서 Z를 10에서 490까지 10씩 증가하게 만든다. 그리고 붙은 입자, 즉 P(Sx, Sy)=1인 입자 수를 세기 위해서 Count를 돌린다.

```
For Z = 10 To 490 Step 10
For L = 0 To 999
For M = 0 To 999
If Z ^ 2 > (L - 500) ^ 2 + (M - 500) ^ 2 And P(L, M) = 1 Then
Count = Count + 1
P(L, M) = 1
End If
Next
Next
```

이제 입자의 반경인 Z와 붙은 입자의 개수를 출력하게 만들면 된다. 반경 10에서 490까지 49개가 출력될텐데 한칸씩 밑으로 가게 출력하기 위해서 Chr(13)(한칸 밑으로) + Chr(10) (맨 앞으로)를 써넣어 준다. 그리고 이것을 Info라는 변수로 만든 후 이것을 레이블 1에 출력하도록 만든다.

```
info = info + "-" + "반경" + Str(Z) + " = " + Str(Count) + Chr(13) + Chr(10)
Label1.Caption = info
Count = 0
Next

End Sub

Private Sub Command2_Click()
End
End Sub
```

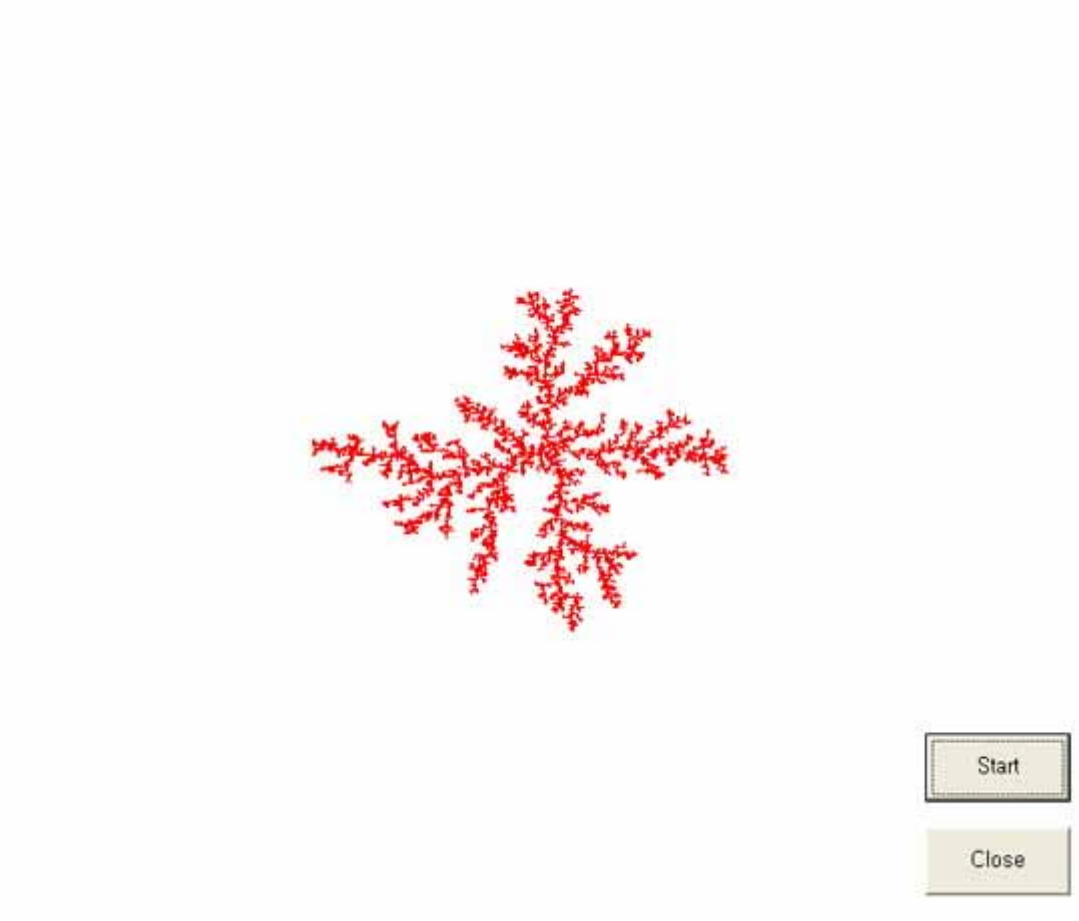
VB 프로그램의 form file, project file 및 execution file은 DLA.frm DLA.vbp DLA.exe와 같다.

3. 프로그램 결과

N=R (N : 붙은 입자수, R :반경 df : 차원)이므로

$$N=R^{df} \quad (N : 붙은\ 입자수,\ R :반경\ df : 차원)$$

$$df=\frac{\log N}{\log R}$$



VB로 짠 프로그램을 실행시켜 얻은 DLA의 모양