Exercício 3 – Dimensão Fractal

Cláudio Santos nº 42208 MIEF

claudiostb7@hotmail.com

Sumário: Dimensão fractal de um agregado percolativo, implementando o método de *“box counting”* e o método de *ensemble*. Geração de agregação limitada por difusão (DLA) numa rede quadrada e cálculo da sua dimensão fractal usando o método de *“sandbox”*.

Exercício 3.1

O objectivo começa por gerar configurações de percolação na probabilidade crítica $\mathbf{p_c=0.592746}$. Utilizou-se o método de queima de baixo para cima (tal como no exercício 2.2) para criar a rede. Depois implementou-se novamente o método da queima mas agora de cima para baixo para isolar o agregado percolativo. Os sítios ocupados sem percolação passaram a sítios vazios ($1 \to 0$) e os sítios queimados passaram a sítios ocupados ($4 \to 1$).

Exercício 3.1.1

O objectivo é implementar o método do ensemble para calcular a dimensão fractal. Considerou-se uma amostra de 250 configurações e calculou-se o número médio de sítios **M** para um dado tamanho do sistema **L**. Fez-se por isso um ciclo para mudar a *seed* do gerador aleatório que cria a rede e para cada *seed* era guardado o número de sitios de cada configuração. Depois criou-se mais um ciclo que ia correr este *array* do número de sitios de cada seed e calculava o número médio de sítios **M** das 250 configurações. Tudo isto foi implementado numa função chamada **ensemble** que tinha como argumento de entrada **L** e retornava **M**. Por fim, fez-se na função main um ciclo que a cada iteração fazia variar **L** e chamava a função **ensemble**. Foram guardados num ficheiro .txt os valores de log(**M**) vs log(**L**), fez-se um ajuste linear dos pontos e o declive da recta é a dimensão fractal **df**.

$$df \propto \frac{\log(M)}{\log(L)}$$

Foi utilizado o programa gnumeric para analisar M vs L.

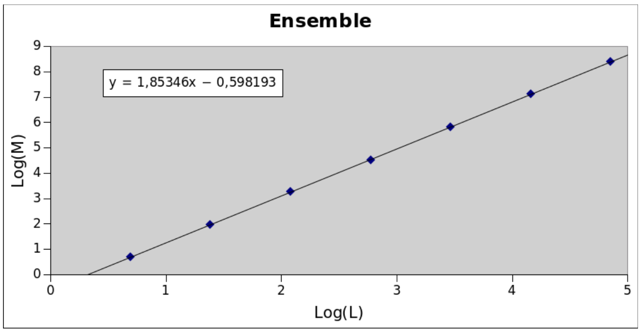

Figura 1: Recta de log (M) vs log (L) calculados para $L=2^n$, n=1,2,…,7 (L_{max}=128) em p_c.

Foi obtido uma dimensão fractal **df=1,85** pelo método do ensemble.

Exercício 3.1.2

O objectivo é implementar o método de “box counting” para calcular a dimensão fractal. O algoritmo consiste em dividir a **rede**[**L*****L**] em caixas de tamanho $\mathbf{l_i * l_i}$ e contar o número de sítios **N**i correspondentes em cada caixa. Fez-se um ciclo que corre os inteiros todos desde 1 até L. Se o resto da divisão inteira $\mathbf{L/l_i}$ tiver resto zero, é definido um *array* **box**[($\mathbf{L/l_i}$)*($\mathbf{L/l_i}$)] e corre-se a **rede**[] de modo a encontrar os sítios ocupados (**i**,**j**). Quando se encontrar, é feito uma mudança de coordenadas para os pontos (**x**,**y**) de **box**[], com **x**=int **i**/**l** e **y**=int **j**/**l**, e põe se também a posição de **box**[**x**+**y***(**L**/**l**)] como sítio ocupado. Por fim corre-se as posições todas de **box**[] e conta-se o número de sítios ocupados $\mathbf{N_i}$. Foram guardados num ficheiro .txt os valores de log($\mathbf{N_i}$) vs log($\mathbf{L/l_i}$), fez-se um ajuste linear dos pontos e o declive da recta corresponde à dimensão fractal **df**.

$$df \propto \frac{\log(N_i)}{\log\left(\frac{L}{l_i}\right)}$$

Foi utilizado igualmente o gnumeric para análise.

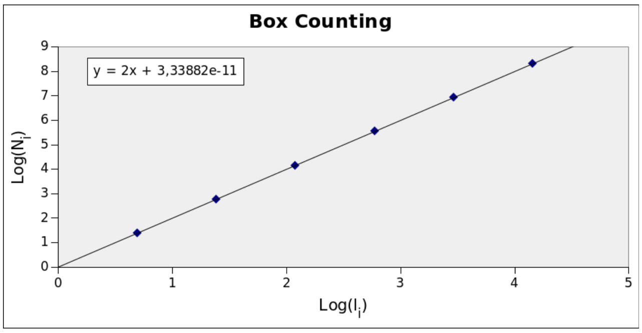

Figura 2: Recta de log(M) vs log(L/l_i) calculados para L=2^7=128 em p_c.

Foi obtido uma dimensão fractal **df=2,0** pelo método de *"box counting"*. Este valor difere do calculado pelo método do *ensemble*. Contudo *"box counting"* é mais rápido a calcular.

Exercício 3.2

Numa primeira parte, o objectivo é gerar configurações de agregação limitada pro difusão (DLA) numa rede quadrada, considerando uma *seed* no centro da rede. Considerou-se o tamanho de sistema **L=100** e número de *walkers* na rede como **N=10^5**. A rede quadrada é criada toda vazia, rede[L*L]={0}. A posição da *seed* corresponde a **rede[(L/2)+(L/2)*L]=1**. As partículas move-se por *random walk* e por isso o tempo de simulação pode ser bastante longo. De modo a reduzir este tempo, considerou-se um raio de largada **r=10** a partir da *seed*. Gera-se o walker numa posição aleatória **(x,y)** desta circuferência. Define-se por isso um ângulo **2*PI*theta** (também aleatório) e usa-se a transformação de coordenadas polares **x=r*cos(theta)+L/2** e **y=r*sin(theta)+L/2** para definir a posição do walker como movimento, **rede[x+y*L]=2**. Defini-se um ciclo de uma variável b*ooleana* **livre=true** para o *walker*. A cada iteração deste ciclo, é gerado um número aleatório que diz se a partícula move para cima (**y++**), baixo(**y--**), direita(**x++**) ou esquerda(**x--**). Se a distância da partícula à seed tornar-se muito grande, esta partícula é destruída e criada outra. Por outro lado, quando o *walker* encontrar primeiros vizinhos ocupados também essa posição fica a pertencer ao *cluster.*

Este algoritmo é repetido para todas as **N** partículas, alterando as sementes dos geradores de números aleatórios para cada uma bem como incrementado a distância de largada **r** a cada 300 walkers para acomodar pelo aumento do *cluster*.

Utilizou-se o código "latticeview.h" para vizualizar o DLA.

Figura 3: DLA de uma rede quadrada 100x100 com seed no centro. Foram utilizados 10^6 partículas.

O programa não está totalmente correcto pois existem pontos ocupados sem agregação. Por outro lado, a distribuição de sítios ocupados não parece uniforme: o *cluster* tem mais pontos no lado direito que no lado esquerdo.

A segunda parte do trabalho consite na implementação do método *"sandbox"* para calcular a dimensão fractal do DLA. O método é semelhante ao *"box counting"*.É colocada uma caixa de tamanho R no centro da rede e conta-se o número de sítios ocupados N(R). Incrementa-se o R aos poucos até apanhar a rede toda, medindo sempre o N(R). Depois-se faz um gráfico de log(N(R)) vs log® e calcula-se a dimensão fractal df.

$$df \propto \frac{\log(N(R))}{\log(R)}$$

Por dificuldade de compreensão e falta de tempo, não foi possível implementar o código desta parte.