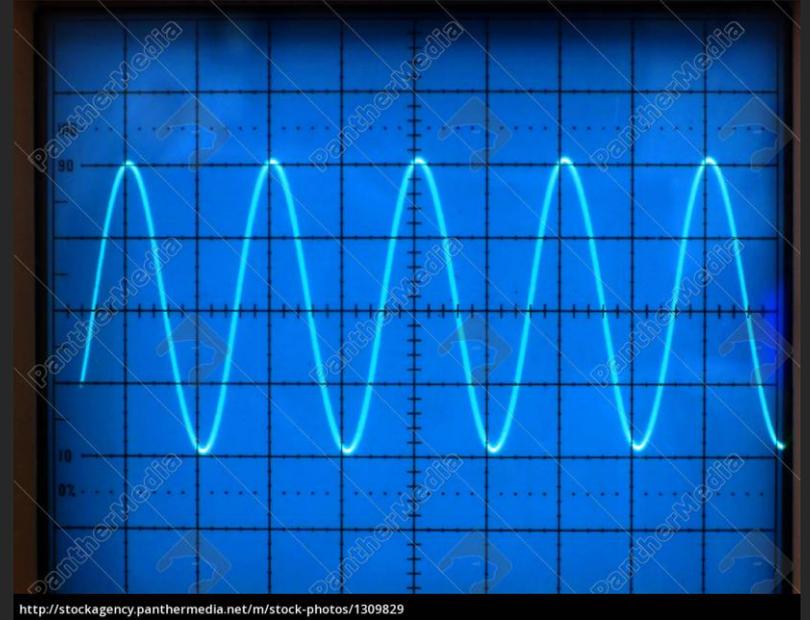


Autocorrelación en C



13 de Mayo de 2021

Pasos a seguir (clase pasada)

- Termalización:

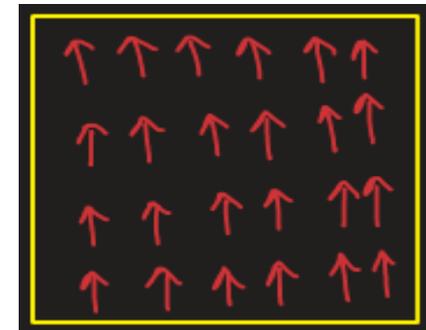
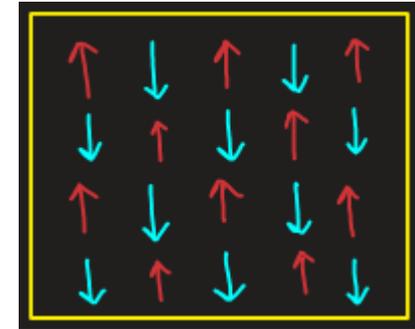
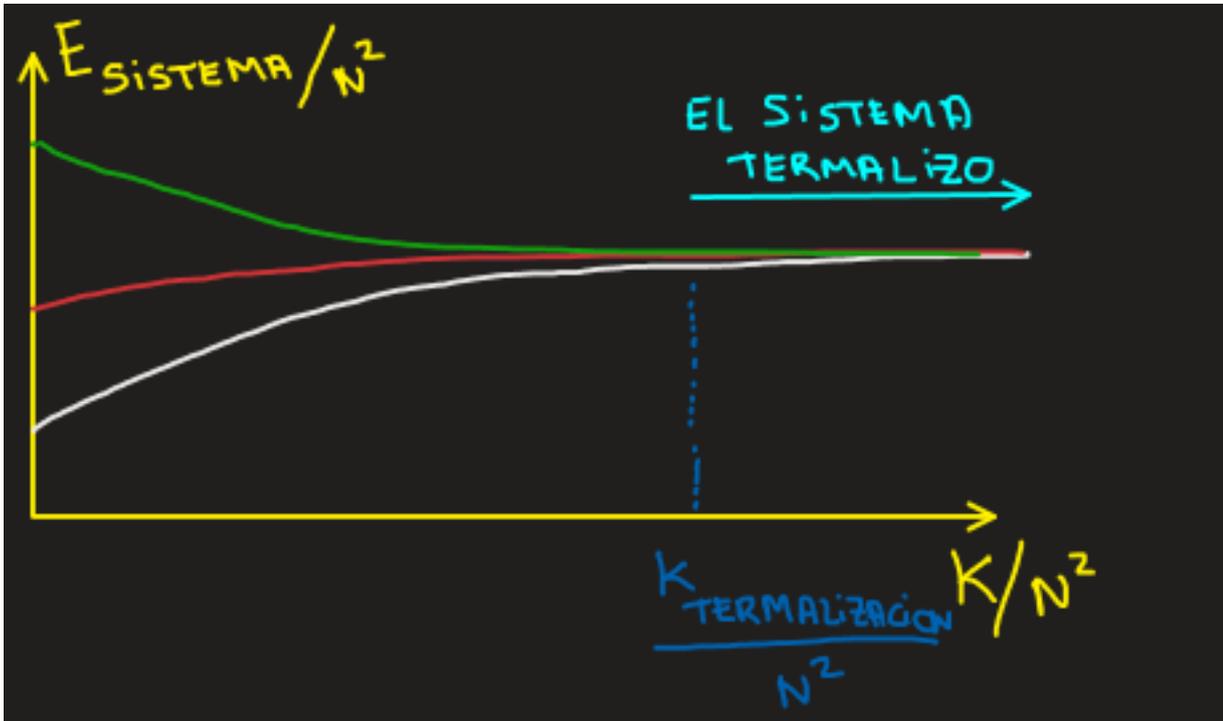
Levantar las curvas de E y M en función de los pasos de Metropolis (k/N^2) ¿convergen?

```
loop sobre temperaturas
{
    creo_tabla();
    calculo_E_inicial();
    calculo_M_inicial();

    termalizacion();
}
}
```

Termalización

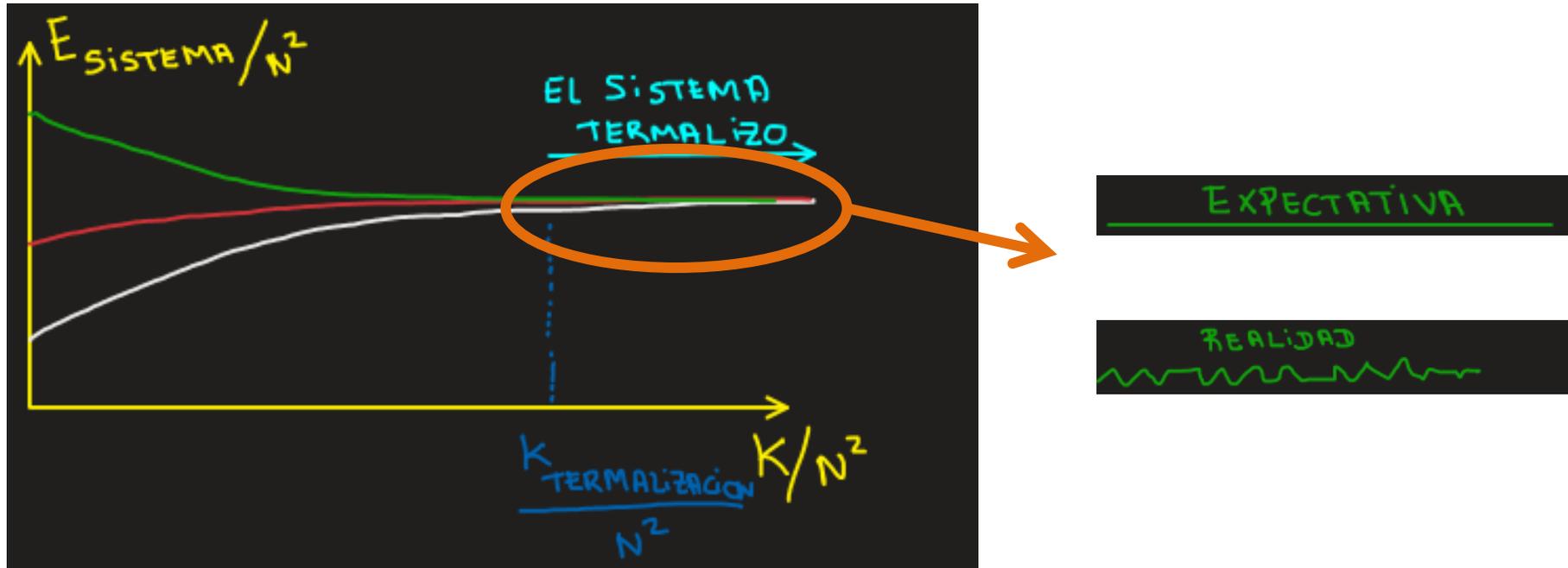
En cada paso de Metropolis (k) calculan la energía y la magnetización del sistema.



Recomendación: T inicial = $5k$ ó $4k$ y $J=1$.

Termalización

En cada paso de Metropolis (k) calculan la energía y la magnetización del sistema.



Recomendación: T inicial = 5k ó 4k y J=1.

Termalización

La cantidad de pasos para que el sistema termalice (ej. VN^2 pasos) se calcula por única vez

Luego de “intentar” reducir la temperatura (loop), hacen esa cantidad de pasos para estar (recién ahí) en la nueva temperatura. Estos pasos son “obligatorios”.

El número de pasos para que mi sistema termalice entre una dada T_1 y otra T_2 ($T_2 < T_1$) ¿es independiente del ΔT ?

Recomendación: T inicial = 5k ó 4k, J=1 y $\Delta T=0.01$ ó 0.001.

```
loop sobre temperaturas
{
    creo_tabla();
    calculo_E_inicial();
    calculo_M_inicial();

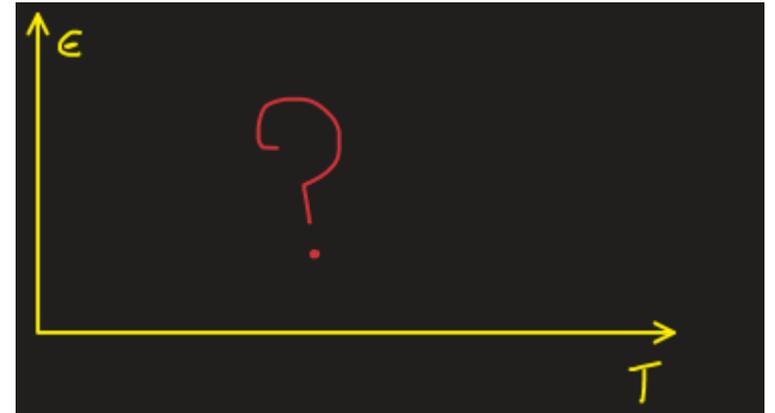
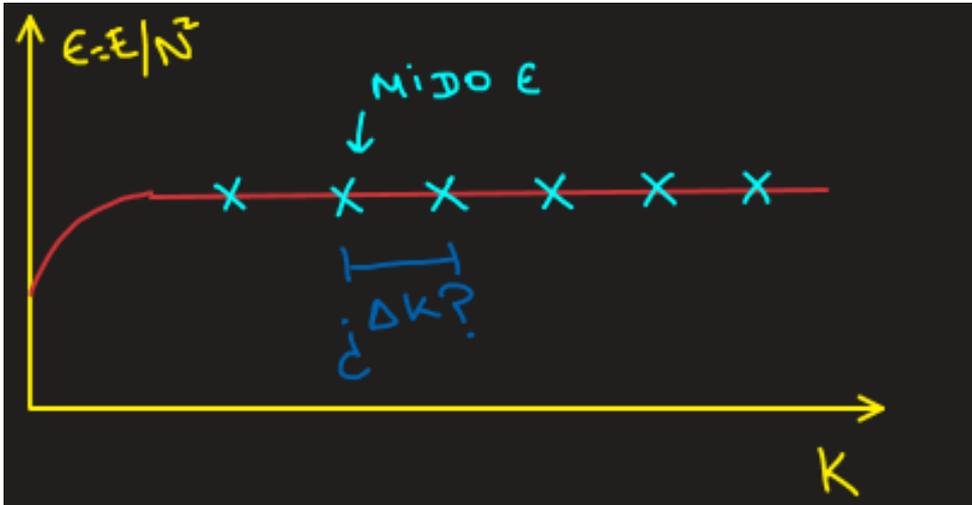
    termalizacion();
}
}
```

Introducción a la autocorrelación

Una vez termalizado mi sistema quiero medir “cosas”.

Quiero medir 10000 veces la energía (y magnetización) del sistema, ¿da lo mismo cada cuanto mido?

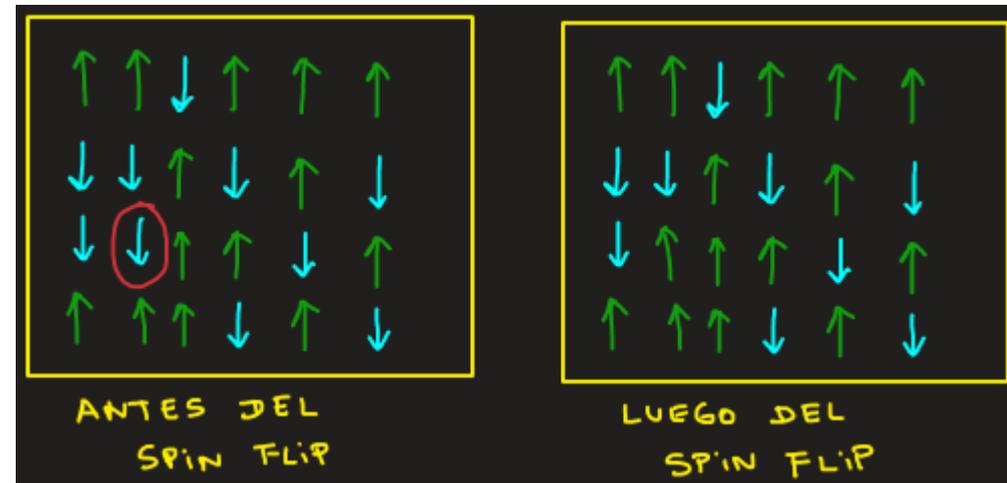
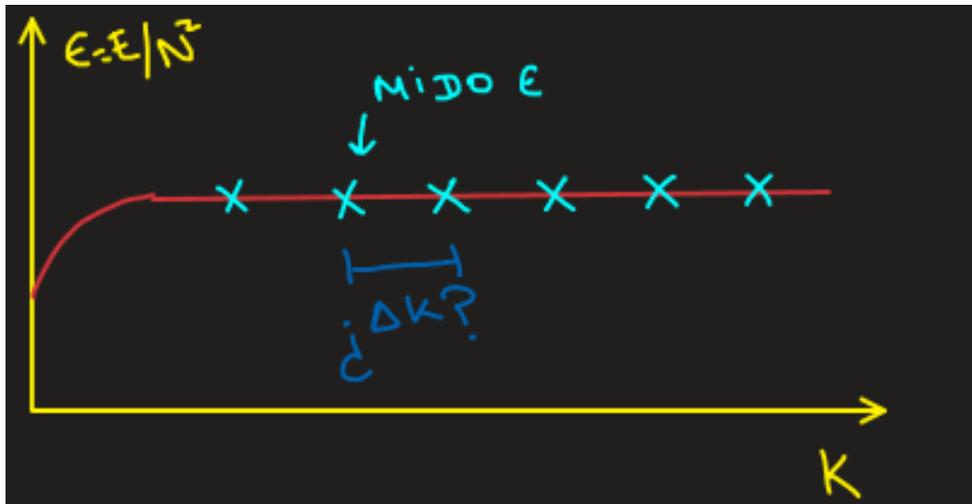
Quiero que mis mediciones sean estadísticamente independientes



Introducción a la autocorrelación

Si solo doy vuelta un spin (con la probabilidad adecuada) mi sistema original es muy parecido al nuevo (i.e. están muy correlacionados).

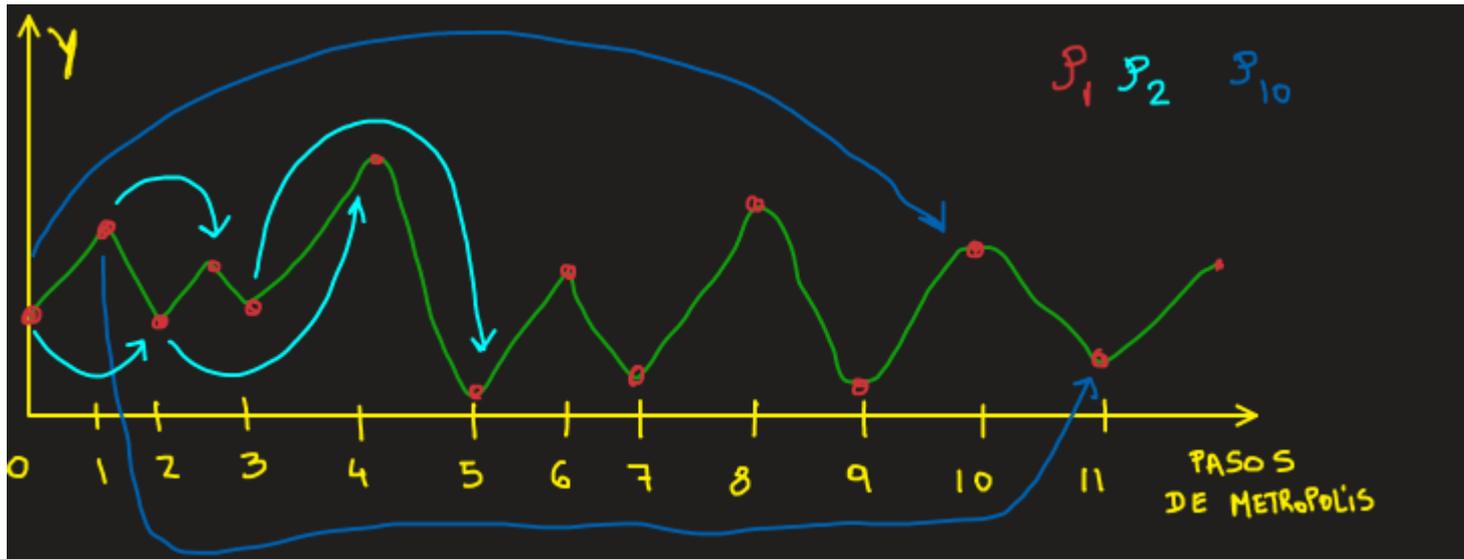
Quisiera ver la forma de medir cuantos pasos de Metropolis hacer para descorrelacionar mi sistema (y mis mediciones).



Autocorrelación

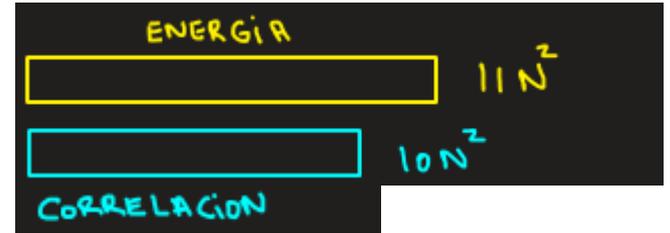
Uno compara las mediciones realizadas en el paso i respecto al paso $i+k$.

$$\rho_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^N (Y_i - \bar{Y})^2}$$



Implementación en C

$$\rho_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^N (Y_i - \bar{Y})^2}$$



- 1) Con el sistema termalizado, nos creamos un vector de energías (y magnetizaciones) de V componentes (ej. $V=11N^2$) y otro de correlación (ej. $10N^2$).
- 2) Hacemos V pasos de Metropolis y en cada paso guardamos E y M .
- 3) Calculamos la autocorrelación.

Esto lo pueden hacer para 4 temperaturas: 5k, 3k, 2.3k y 2k y comparar los resultados.

Recuerden que si quieren llegar a $T=2k$ deben empezar a T alta (ej. 5k) e ir bajando la temperatura de a poco.

Implementación en C

$$\rho_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^N (Y_i - \bar{Y})^2}$$

Lo calculamos por partes:

Primero calculan el valor medio.

Después el denominador: solo visitar cada componente del vector.

Por último el numerador:

```
for (k = 0; k < n_vec_corr; k++)
//ej. n_vec_corr = 10N*N
{
    num_e = 0.0;
    num_m = 0.0;
    for (ki = 0; ki < nsamp-k; ki++)
    //nsamp = mediciones de E y M (ej. 11*N*N)
    {
        num_e=num_pk_e + (vec_e[ki]-e_media)*(vec_e[ki+k]-e_media);
        ¿? //magnetizacion
    }
    vec_corr_e[k]= vec_corr_e[k] + num_e/(double)(den_e*1000);
    vec_corr_m[k]= vec_corr_m[k] + num_m/(double)(den_m*1000);
}
```

IMPORTANTE

$$\rho_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^N (Y_i - \bar{Y})^2}$$

Con lo anterior estaríamos calculando un valor de ρ_0 otro de ρ_1 , etc.

Pero, queremos calcular un promedio sobre J valores (ej. 1000 valores).

```
for (int s = 0; s < ns; s++)
{
    for (int is = 0; is < nsamp; is++)
    {
        metropolis();
        vec_e[is] = energ;
        vec_m[is] = m;
    }
    correlacion();
}
```

Pseudo-código para Ising

```
poblamos_red();  
  
loop sobre temperaturas  
{  
    creo_tabla();  
    calculo_E_inicial();  
    calculo_M_inicial();  
  
    termalizacion();  
  
    loop sobre mediciones  
    {  
        loop sobre pasos de metropolis  
        {  
            metropolis();  
        }  
        vec_e[s] = energ;  
        vec_m[s] = m;  
    }  
}
```

Me “aseguro” que
estoy a esa
temperatura (hay que
averiguar cuantos
pasos hacer)

Para descorrelacionar
las mediciones
(hay que averiguar
cuantos pasos hacer)

Resumiendo: calculan una sola vez los pasos necesarios para termalizar y descorrelacionar y luego los usan. Es decir, hacen esos pasos de Metropolis antes de hacer las mediciones.

Algunos comentarios sobre los informes

- Las citas dan peso a las oraciones (y no te la pueden criticar):
ej. este fenómeno se observa en sistemas magnéticos...[1].
- Captions con información suficiente para no tener que recurrir al texto.
- Si hay alguna zona interesante (ej. vecindad de p_c), hacer un zoom.

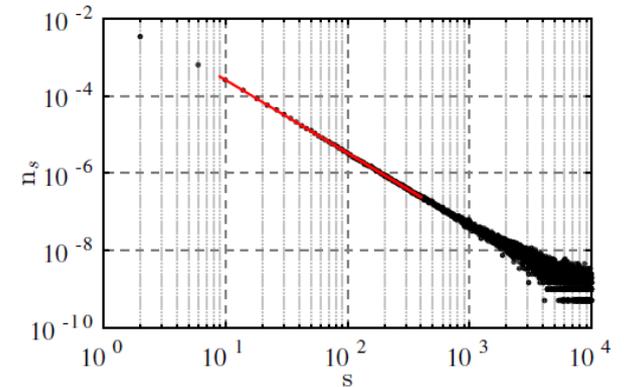


FIG. 5. Número medio de clusters de tamaño s por nodo (n_s) en función del tamaño de los mismos para una red de 256×256 y una probabilidad de ocupación de los nodos igual a $p_c=0.5928$. La línea continua de color rojo representa un ajuste lineal. Se realizaron un total de 30000 iteraciones.

Algunos comentarios sobre los informes

- Si conocemos el valor teórico (ej. $p_c=0.5927$) ó la curva ideal, es útil graficarla junto a los datos experimentales para poder comparar.
- Unir las diferentes secciones para que haya un hilo conductor y sea más fluida la lectura. Guiar al lector. Ej. En la sección II se analizará mas en detalle este efecto.
- Pensar si ustedes lo leen en el futuro, ¿lo pueden reproducir? ¿Qué datos son necesarios que reporte?
- Documentar el código.
- Agregar las incertezas a las magnitudes calculadas.

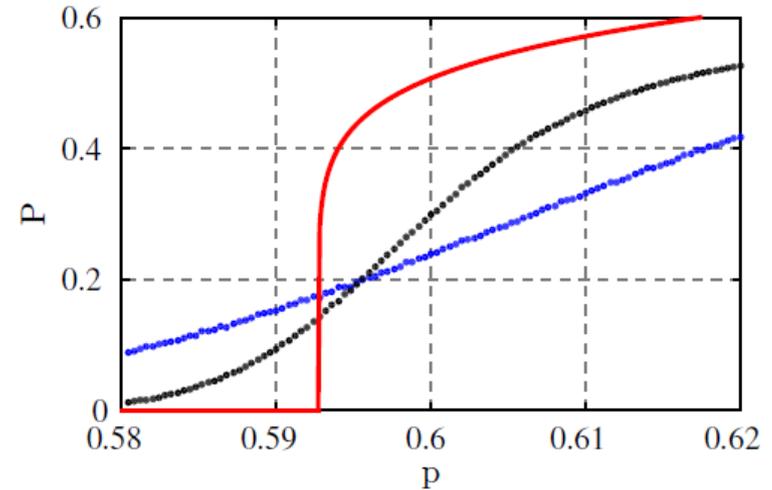


FIG. 7. Intensidad del cluster percolante (P) en función de la probabilidad de ocupación de los nodos (p). Los puntos de color azul y negro representan redes de 64×64 y 256×256 , respectivamente. La línea continua de color rojo representa el resultado teórico para una red de tamaño infinito ($p_c=0.5927$). Para cada probabilidad de ocupación se realizaron un total de 30000 iteraciones.