

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

ESTUDIO DE LA TRANSICIÓN DE FASE KOSTERLITZ-THOULESS
EN EL MODELO XY DILUIDO BIDIMENSIONAL UTILIZANDO
SIMULACIONES DE MONTE CARLO.

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
FÍSICO

PROYECTO DE INVESTIGACIÓN

JEFFERSON ALFONSO VACA SANTANA

jefferson.vaca01@epn.edu.ec

Director: DR. LEONARDO ALBERTO BASILE CARRASCO

leonardo.basile@epn.edu.ec

QUITO, DICIEMBRE 2018

DECLARACIÓN

Yo, JEFFERSON ALFONSO VACA SANTANA, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Jefferson Alfonso Vaca Santana

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por JEFFERSON ALFONSO VACA SANTANA, bajo mi supervisión.

Dr. Leonardo Alberto Basile Carrasco
Director del Proyecto

AGRADECIMIENTOS

Agradezco mi familia por su apoyo incondicional en cada una de mis etapas estudiantiles. A mi Colegio por permitirme tener los conocimientos y habilidades necesarias para continuar con mis estudios. A la Escuela Politécnica Nacional y todos los profesores que me han compartido sus conocimientos con paciencia, con especial mención a mi Tutor el Dr. Ramon Xulvi y mi Director el Dr. Leonardo Basile. Y al Centro de Modelización Matemática ModeMat por colaborarme con la prestación 420 núcleos de procesamiento.

DEDICATORIA

Dedicado a mis padres por su perseverancia y compromiso.

Índice general

1. Marco teórico	5
1.1. Mecánica estadística	5
1.1.1. Ensamble canónico	5
1.1.2. Transición de Fase	7
1.1.2.1. Límite termodinámico	8
1.1.2.2. Hipótesis de universalidad	9
1.2. Sistemas Magnéticos	11
1.2.1. Modelo de Ising	12
1.2.2. Modelo XY	15
1.2.2.1. Vórtices	17
1.2.2.2. Transición Kosterlitz y Thouless	21
1.2.3. Modelo XY diluido	22
1.3. Métodos Montecarlo	24
1.3.1. Cadena de Márkov	25
1.3.2. Hipótesis Ergódica	26
1.3.3. Método Montecarlo Metrópolis	27
2. Metodología	28
2.1. Implementación del sistema	28
2.1.1. Implementación de la red	28
2.1.2. Propiedades físicas del sistema	29
2.1.2.1. Energía	29
2.1.2.2. Magnetización	30

2.1.2.3.	Función de correlación	30
2.1.2.4.	Vórtices	30
2.2.	Implementación del Ensamble	31
2.2.1.	Evolución del sistema	31
2.2.2.	Calculo de propiedades físicas del Ensamble	32
2.2.2.1.	Regresión por ley Exponencial	33
2.2.2.2.	Regresión por ley de Potencias	34
2.3.	Calculo de la transición	35
2.3.1.	Método Cv	35
2.3.2.	Método Rigidez de Giro	35
2.3.3.	Efectos de tamaño en la temperatura crítica	36
2.3.4.	Efectos de dilución	36
3.	Análisis de Resultados	37
3.1.	Energía	38
3.1.1.	Efectos de tamaño en la energía	39
3.1.2.	Efectos de impurezas en la energía	40
3.2.	Capacidad calorífica	41
3.2.1.	Efectos de tamaño en la capacidad calorífica	41
3.2.2.	Efectos de las impurezas en la capacidad calorífica	42
3.3.	Magnetización	43
3.3.1.	Efectos de tamaño en magnetización	44
3.3.2.	Efectos de impurezas en la magnetización	45
3.4.	Susceptibilidad magnética	46
3.4.1.	Efectos de tamaño en la susceptibilidad magnética	47
3.4.2.	Efectos de la las impurezas en la susceptibilidad magnética	48
3.5.	Rigidez de giro	49
3.5.1.	Efectos de tamaño en la rigidez de giro	49
3.5.2.	Efectos de las impurezas en la rigidez de giro por dilución	50
3.6.	Vórtices	52

3.7.	Función de correlación	56
3.7.1.	Función de correlación sin impurezas	57
3.7.2.	Función de correlación concentración 0,96	58
3.7.3.	Función de correlación concentración 0,88	60
3.7.4.	Función de correlación concentración 0,80	62
3.7.5.	Efectos de las impurezas en los exponentes críticos	64
3.7.5.1.	Efectos de las impurezas en longitud de correlación	64
3.7.5.2.	Efectos impurezas en el exponente Eta	65
3.8.	Temperatura crítica de transición	66
3.8.1.	Efectos de tamaño en la temperatura crítica de transición	67
3.8.2.	Efectos de la concentración de espines en la temperatura de transición	68
4.	Conclusiones	72
A.	Suplementos Matemáticos	74
A.1.	Topología	74
A.1.1.	Espacio topológico	74
A.1.2.	Triangulación	74
A.1.3.	Característica de Euler	75
A.2.	Geometría	75
A.2.1.	Variedad diferencial	75
A.2.2.	Vector tangente	75
A.2.3.	Campo de vectores	75
A.2.4.	Conexión	76
A.2.5.	Curvatura	76
A.2.6.	Teorema de Gauss-Bonnet	77
A.2.7.	Defectos Topológicos en el modelo xy	77
A.3.	Probabilidad	78
A.3.1.	Espacio de eventos	78
A.3.2.	Espacio de probabilidad	79

A.3.3. Variable Aleatoria	79
A.3.4. Proceso Estocástico	79
A.3.5. Cadena de Márkov	79
B. Manual	80
B.1. Modo de uso	80
B.2. Estructuras	80
B.2.1. sis	80
B.2.2. sist	81
B.2.3. dat	81
B.3. Funciones	81
B.3.1. crear sis montecarlo	81
B.3.2. Maximo maximal	82
C. Código	83
C.1. Main	83
C.2. xy.c	86
C.3. xy.h	118

Índice de figuras

1.1. Cambio de fase ferromagnética y paramagnética	8
1.2. Energía y capacidad calorífica en en el modelo de Ising	13
1.3. Magnetización por espín modelo de Ising	13
1.4. Dominios magnéticos	14
1.5. Red de espines en el modelo xy caracterizados por un ángulo	15
1.6. Espacio del parámetro de orden	18
1.7. Vórtices en el espacio del parámetro de orden	19
1.8. Vortices y antivórtices	20
1.9. Percolación	23
1.10. Efectos de impurezas reportados.	24
3.1. Tiempo en función del tamaño	38
3.2. Efectos de tamaño en la energía	39
3.3. Efectos de impurezas en la energía	40
3.4. Efectos de tamaño en la capacidad calorífica	42
3.5. Capacidad calorífica por dilución	43
3.6. Efectos de tamaño en magnetización	44
3.7. Efectos de impurezas en la magnetización	46
3.8. Susceptibilidad magnética por tamaño	47
3.9. Efectos de la las impurezas en la susceptibilidad magnética	48
3.10. Efectos de tamaño en la rigidez de giro	50
3.11. Efectos de la las impurezas en la susceptibilidad magnética	51
3.12. Efectos de tamaño en el número de vórtices	52

3.13. Efectos de las impurezas en el número de vórtices	53
3.14. Red temperatura de $0,765k_b/J$ sin impurezas	54
3.15. Red temperatura de $1,02k_b/J$ sin impurezas	54
3.16. Red temperatura de $0,765k_b/J$ concentración 0,96	55
3.17. Red temperatura de $1,02k_b/J$ concentración 0,96	56
3.18. Función de correlación sin impurezas	57
3.19. Exponentes críticos sin impurezas	58
3.20. Función de correlación concentración 0,96	59
3.21. Exponentes críticos	60
3.22. Función de correlación concentración 0,88	61
3.23. Exponentes críticos	62
3.24. Función de correlación concentración 0,80	63
3.25. Exponentes críticos	64
3.26. Efectos de las impurezas en en longitud de correlación	65
3.27. Vórtices por tamaño	66
3.28. Efectos de tamaño en la temperatura crítica de transición	68
3.29. Efectos de la concentración de espines en la temperatura de transición	69
3.30. Temperatura crítica de transición en función de la concentración de espines.	70

Índice de tablas

1.1. Exponentes críticos	10
3.1. Temperatura crítica de transición en función de la concentración de espines	71

Resumen

En el presente trabajo se realiza un estudio de la transición de fase KT en el modelo XY - diluido, en una red cuadrada con condiciones de frontera periódica, utilizando una simulación con el método Monte Carlo Metrópolis. Esta simulación está principalmente programada en C. Se determinó, la energía, la capacidad calorífica, la magnetización, la susceptibilidad magnética, la rigidez de Giro, y la correlación entre espines con sus respectivos exponentes críticos para cada temperatura, tamaño y dilución.

Para determinar la temperatura de transición se ha utilizado dos métodos, que nos permitieron encontrar la temperatura crítica de transición, después de realizar varias simulaciones y las respectivas correcciones por el efecto del tamaño finito de la red.

El primer método lo realizamos midiendo la capacidad calorífica en función de la temperatura, para encontrar la temperatura maximal de cada gráfica, donde obtuvimos que la temperatura de transición es de $T_{kt} = 0,89$. Para el segundo método utilizamos la intersección entre la función lineal de pendiente $2/\pi$ y la curva de la rigidez de Giro en función de la temperatura, donde se determinó la temperatura crítica dependiente del tamaño, misma que fue $T_{kt} = 0,87$. Es decir, una diferencia porcentual del 2,77 %

Abstract

In this work, we show an study of the KT phase transition at the diluted XY model due to an Monte Carlo Metropoli simulation. The simulation was carried out in C. We determine properties of the system like energy, heat capacity, the magnetization, the magnetic susceptibility and the correlation between spins according to temperature and dilution. To determine the transition temperature we use two methods. As a first method, the dependence on the network size of the critical temperature is used, which is determined by the maximum of the heat capacity at a given temperature. Making corrections for finite size effects we determine that critical temperature in this method is $T_{kt} = 0,89$. To determine the transition temperature we use two methods. As a first method, the dependence on the network size of the critical temperature is used, which is the temperature at which the rotation stiffness intersects the straight line that originates from the origin with slope $2/\pi$. Making corrections for finite size effects we determine that critical temperature in this method is $T_{kt} = 0,89$. That means, a porcential difference of 2,77 %

Introducción

Las transiciones de fase están entre los fenómenos más notables en la naturaleza. Una transición de fase involucra un cambio súbito en determinadas propiedades macroscópicas medibles de un sistema, como resultado de la variación de parámetros externos como la temperatura o la presión. Entre las transiciones de fase más conocidas se encuentran la evaporación, la condensación, la solidificación, la transición entre las fases ferromagnética y paramagnética. Todos estos cambios de fase presentan una ruptura en la simetría del espacio de los parámetros de orden; por ejemplo, un material en fase paramagnética pierde su simetría rotacional en la fase ferromagnética debido a una variación en la temperatura [1].

Este fenómeno está descrito matemáticamente por el comportamiento no analítico de las funciones termodinámicas, es decir, derivadas de la energía libre, como la capacidad calorífica o la susceptibilidad. Así, se pueden clasificar en transiciones de primer orden (discontinuas) cuando la primera derivada de la energía libre tiene comportamiento no analítico o de orden superior (continuas) cuando las derivadas de orden superior de la energía libre tienen comportamiento no analítico. Bajo esta clasificación, Las transiciones de orden infinito no rompen ninguna simetría del sistema [2]; la más famosa de estas es la transición de Kosterlitz-Thouless (KT) en el modelo XY bidimensional [3]. En el presente trabajo se realizará un estudio de la transición (KT) en el modelo XY diluido en una red cuadrada con condiciones de fronteras periódicas utilizando una simulación Monte Carlo Metrópolis, y las regresiones por el método de mínimos cuadrados.

El modelo XY diluido objeto del presente estudio, consiste en una red cuadrada con condiciones de fronteras periódicas. En este modelo cada sitio de red puede estar ocupada por una partícula magnética o por una impureza no magnética. La partícula magnética esta caracterizada por su espín que esta representado por un vector que puede girar libremente en el plano. La impureza está representada por una vacancia en el sitio de red que no interactúa con las partículas [4]. Como todas las partículas magnéticas son consideradas idénticas en este modelo, para diferencia

una partícula de otra, es suficiente saber el ángulo que forman con algún eje fijo. En este trabajo utilizaremos el eje x como fijo. Para modelar la interacción entre los espines en el plano se utiliza una variante al modelo de Ising, por la ecuación descrita a continuación:

$$H = -J \sum_{\langle k,l \rangle} \cos(\theta_k - \theta_l) \quad (1)$$

donde H representa el Hamiltoniano, J la constante de acoplamiento entre espines, el sumatorio se realiza entre los vecinos más cercanos y θ_i son los ángulos de los espines.

El modelo XY es la variante más simple del modelo de Ising, con simetría continua. Cabe recalcar que se ha demostrado matemáticamente que en un modelo de simetría continua en dos dimensiones no presenta ruptura espontánea de simetría [5], es decir, la magnetización es cero a cualquier temperatura. El modelo XY no tiene un verdadero orden de largo alcance, más bien el sistema se encuentra en un cuasi orden de largo alcance, donde un defecto topológico se forma por pares de vórtice-antivórtice. Estos defectos topológicos los definiremos precisamente en el tema 1.2.2.1. La transición (KT) consiste en la formación pares vórtice-antivórtice a partir de cierta temperatura de transición (TKT). Bajo TKT no se forman pares vórtice-antivórtice y la longitud de correlación decae como una función de potencia para todas las temperaturas bajo TKT [3]. Este comportamiento es distinto al de una transición de fase en el modelo de Ising, donde la longitud de correlación se comporta como una función de potencia solo en la temperatura de transición.

En el presente trabajo se estudiará, las propiedades necesarias para comprender la transición Kosterlitz-Thouless, tales como: la energía, la capacidad calorífica, la magnetización y susceptibilidad magnética el número de vórtices, la rigidez de giro y la función de correlación con sus respectivos exponentes críticos en función de la temperatura. [6].

Estudiar el modelo XY diluido tiene interés práctico por dos motivos: a) debido a la hipótesis de universalidad que la detallaremos más adelante, con este simple modelo podremos estudiar fenómenos como la superconductividad, la superfluididad y la coherencia en un láser. b) debido a la inclusión de impurezas y sus efectos en la transición de fase KT. La presencia de impurezas está relacionada con sistemas reales, donde estos defectos están siempre presentes. Las impurezas pueden activar la formación de vórtices y en principio disminuir la temperatura de transición. Exis-

ten diferentes conclusiones acerca de la densidad de impurezas, deja de ocurrir la transición. Para tratar de aclarar estas diferencias, en este estudio se utilizará dos métodos que determinarán la interrogante [7], posterior a la realización de varias simulaciones y las respectivas correcciones necesarias debido al efecto del tamaño finito de la red. El primer método se basa en encontrar la temperatura maximal de cada gráfica midiendo la capacidad calorífica en función de la temperatura [8, sección 7.9.4]. Para el segundo método se utilizará la intersección entre la curva de la rigidez de giro en función de la temperatura y la función lineal de pendiente $2/\pi$, para determinar la temperatura crítica relativa al tamaño.

El principal objetivo de este trabajo realizar un estudio preliminar de la transición de fase (KT) en el modelo XY-diluido bidimensional utilizando simulaciones de Monte Carlo mediante el desarrollo de un software que permita estudiar el modelo XY-diluido. Entre los usos que se prevén para este programa informático es el calculo de factores referentes al modelo XY-diluido en el equilibrio como la energía, la capacidad calorífica, la magnetización y susceptibilidad magnética, el número de vórtices, la rigidez de giro y la función de correlación con sus respectivos exponentes críticos. Realizando varias simulaciones se determinará la temperatura critica de transición para diferentes concentraciones de impureza y se analizará los resultados. Este programa con una interfase amigable por consola servirá también como futura herramienta para nuevos estudiantes de Física de la Escuela Politécnica Nacional que pretendan realizar nuevas investigaciones en torno al mismo tema, además este software será diseñado para que sea de fácil modificación y de total autoría de la universidad. Entre los usos que se prevén para este programa informático es el calculo de factores referentes al modelo XY como la energía, la capacidad calorífica, la magnetización y susceptibilidad magnética, el número de vórtices, la rigidez de giro y la función de correlación con sus respectivos exponentes críticos.

El Primer capitulo es una recopilación de resultados previos para comprender la problemática de este trabajo. Comenzando con una breve revisión de forma general sobre la mecánica estadística tratando el ensamble canónico y las transiciones de fase. El segundo subtítulo de este capitulo son los sistemas magnéticos. Primero se muestra el modelo de Ising como un preámbulo para el modelo XY y finalmente el modelo XY diluido.

El segundo subtítulo de este capitulo son los sistemas magnéticos. Primero se muestra el modelo de Ising como un preámbulo para el modelo XY y finalmente el modelo XY diluido.

En el segundo capítulo se indican los detalles metodológicos y los principales algoritmos utilizados en el programa. Se describe como se implementa la red en la simulación, como se logra la evolución del sistema incluyendo el número de pasos y los parámetros que se utilizaron. Finalmente se indica como se calculan las propiedades termodinámicas.

En el tercer capítulo se muestran los resultados mas importantes extraídos de los cálculos a distintas concentraciones de impurezas y tamaños de red. Los resultados extraídos son: energía, capacidad calorífica, susceptibilidad magnética, y exponentes críticos en función de la temperatura. Al final de este capítulo se realizará un análisis de la temperatura de transición.

En el capítulo cuatro se presentan las conclusiones extraídas del análisis de resultados y posteriormente los anexos. El primer anexo abarca 4 temas: el primer tema muestra como se consiguen las regresiones por ley de potencia y por ley exponencial implementadas en el código de estos algoritmos utilizando calculo vectorial. El segundo y tercer tema permiten entender el teorema de Gauss-Bonnet y el cuarto tema define un proceso estocástico. El segundo anexo es una descripción de las principales funciones y estructuras que posee el código realizado. El tercer anexo es el código fuente que permitió la realización de la simulación.

Capítulo 1

Marco teórico

1.1. Mecánica estadística

1.1.1. Ensemble canónico

Para el estudio de un sistema compuesto de un gran número de partículas que presentan propiedades similares se debe usar técnicas estadísticas. Un sistema puede existir en cualquier estado (configuración de partículas) de una colección de estados cuánticos. Al conjunto de estados se lo denomina ensemble o colectividad. Un ensemble canónico se define como la agrupación de estados posibles en que se encuentran el sistema al entrar en contacto con un reservorio térmico. La temperatura del ensemble está determinada por la temperatura del reservorio térmico, debido a que se llega a un estado de equilibrio térmico entre el ensemble y el reservorio. Además, en el ensemble canónico el número de partículas y su tamaño no cambian. Dentro de un ensemble podemos encontrar varias configuraciones a las cuales los denominamos micro-estado. Si los micro-estados comparten una característica específica en común o no; pero, si la comparten su conjunto crea un macro-estado. Para el caso de ensemble canónico la característica en común es la energía. En un sistema estadístico se pueden definir probabilidades aplicando el principio de Máxima entropía,

$$S = \sum_i p(E_i) \ln(p(E_i)) \quad (1.1)$$

con la restricción de que el promedio de energía sea constante.

$$U = \sum_i p(E_i)E_i \quad (1.2)$$

La probabilidad $p(E_i)$ de que el micro-estado esta con emergía E_i está dada por:

$$p(E_i) = \frac{e^{-\frac{E_i}{k_b T}}}{\mathcal{Z}} \quad (1.3)$$

donde k_b es la constante de Boltzmann; por simplicidad de notación, se define $\beta = \frac{1}{k_b T}$, y \mathcal{Z} es la condición de normalización de la probabilidad, denominada función de partición, descrita en la ecuación a continuación.

$$\mathcal{Z} = \sum_i e^{-\beta E_i} \quad (1.4)$$

Estas probabilidades definen promedios de cualquier magnitud en el ensamble denominadas variables termodinámicas. Los cuales en este trabajo tendrán la notación $\langle \rangle$, por ejemplo si E es la energía su promedio es $\langle E \rangle$. Esto será aplicable para cualquier magnitud como en la siguiente ecuación.

$$\langle x \rangle = \sum_i x(E_i)p(E_i) \quad (1.5)$$

Donde $\langle x \rangle$ es la variable termodinámica Como el ensamble tiene una temperatura dada por el reservorio térmico, todos los promedios dependen de su temperatura. En el ensamble es importante saber como influyen los cambios de temperatura en los promedios de energía; para esto, se define la capacidad calorífica (C_v) del ensamble como la variación de la energía promedio, con respecto a la temperatura (T).

$$C_v = \frac{\partial \langle E \rangle}{\partial T} \quad (1.6)$$

Tomando en cuenta que la función de partición nos permite calcular las variables termodinámicas que son determinantes en el ensamble con simples manipulaciones algébricas se puede obtener que:

$$C_v = \frac{1}{k_b T^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right) \quad (1.7)$$

1.1.2. Transición de Fase

Los sistemas termodinámicos están caracterizados por propiedades macroscópicas medibles. Cuando se varía un parámetro externo denominado variable de control estas propiedades macroscópicas varían drásticamente, esto ocurre alrededor de un punto crítico. Por ejemplo la transición líquido-gas en el agua donde la propiedad macroscópica es la densidad y la variable de control es la temperatura. Mientras el punto crítico es $373,15\text{K}$ en condiciones normales por debajo de este valor la densidad es mayor y por encima de este valor la densidad disminuye. Es decir, la densidad tiene una discontinuidad o un comportamiento no analítico. La transición de fase es el comportamiento no analítico de alguna variable termodinámica por la variación de alguna otra variable de control. Generalmente esta variable de control es la temperatura que, cuando alcanza el punto crítico, se denomina temperatura crítica.

Existen transiciones de fase de primer orden o discontinuas y de orden superior o continua. Una transición de fase es de primer orden cuando, la primera derivada de la energía libre de Gibbs es discontinua. Esta transición suelen caracterizarse por la liberación o absorción de calor latente por parte del sistema. Esta es proporcional al tamaño de éste. Se debe tomar en cuenta que la energía no se puede transmitir instantáneamente, por este motivo durante la transición coexisten regiones o dominios con diferentes fases; por ejemplo, las transiciones de primer orden son los cambios de estado: la evaporación, la licuefacción, la fusión, la ebullición o la sublimación. Transición de fase de orden superior: Estas van del segundo orden hasta un orden infinito, cuya energía libre de Gibbs es continua. Para determinar el orden de la transición se utiliza el orden de la primera derivada de la energía libre discontinua; estas no tienen asociado un calor latente; por ejemplos, tenemos la transición ferromagnética-paramagnética o la condensación de Bose-Einstein.

La transición ferromagnética-paramagnética se produce cuando el sistema alcanza una temperatura crítica denominada Temperatura de Curie. A esta temperatura los momentos magnéticos cambian de orientación cambiando la magnetización del sistema produciendo un cambio de simetría. Los momentos magnéticos cambian de una simetría esférica a una simetría cilíndrica que se conoce como fase paramagnética y fase ferromagnética respectivamente. Los cambios de simetría de este tipo se denominan ruptura espontánea de simetría que se puede apreciar en la siguiente figura

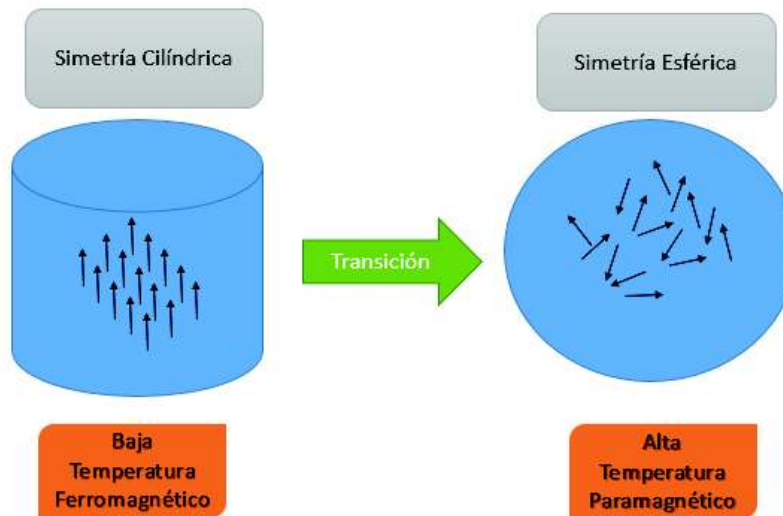


Figura 1.1: Cambio de fase ferromagnética y paramagnética

En la fase ferromagnética, casi todos los momentos magnéticos apuntan en una misma dirección. Si un momento magnético apunta en una determinada dirección entonces, momentos magnéticos muy lejanos apuntan a la misma dirección. Esta interacción de largo alcance indica que los momentos están correlacionados. En otras palabras, para que exista este cambio de simetría es necesario que exista una correlación de largo alcance entre las partículas caracterizadas por su momento magnético.

Sin embargo, en los sistemas magnéticos existen transiciones de fase que no implican un cambio de simetría; es más, está demostrado por el teorema de Mermin-Wagner que no puede haber magnetización para sistemas de dos dimensiones con simetría continua. El caso más sencillo que se puede citar es el modelo-XY, que lo discutiremos más a fondo en el siguiente capítulo.

En un sistema al rededor del punto crítico pueden coexistir dos fases. Para estos sistemas se puede determinar la diferencia en la cantidad sustancia entre un estado y el otro dividido para la cantidad total de sustancia, esto se denomina parámetro de orden. Este parámetro tiene un valor aproximadamente a cero una vez que se sobrepasa el punto crítico y el sistema se encuentra en una fase homogénea [9].

1.1.2.1. Límite termodinámico

En los sistemas suficientemente grandes, la energía libre es extensiva; es decir, la energía es proporcional al tamaño del sistema. Para un sistema finito se puede añadir correcciones debidas al área o dimensiones inferiores. Esto queda matemáti-

camente expresado por la siguiente ecuación [9],

$$F = Vf_v + Sf_s + O \quad (1.8)$$

donde F es la energía del sistema, V es el volumen, f_v es la energía libre por unidad de volumen a tamaño infinito, S es la superficie, f_s es la energía libre por unidad de superficie y O es un error de la aproximación. Los coeficientes f_i son propiedades intensivas que caracterizan el sistema. Cuando un sistema admite una expansión utilizando estos coeficientes f_i se puede definir sistemas de tamaño infinito, a lo que se le conoce como límite termodinámico.

Por su parte, los sistemas reales son siempre finitos, aunque con un número muy grande de partículas. Sin embargo, esta descripción formal proporciona una aproximación del fenómeno real. Si dispusiéramos de una resolución instrumental perfecta, en efecto se podrían apreciar los efectos de tamaño, por ejemplo el calor específico, presenta un máximo finito para algún valor de la temperatura. La resolución instrumental necesaria para observar esto depende del tamaño del sistema. Una de las características más distintivas de un fenómeno crítico es la divergencia en la longitud de correlación ζ al aproximarnos del punto crítico. Lejos del punto crítico toma valores pequeños. Dado que las correlaciones decaen exponencialmente, para un sistema razonablemente grande, los efectos de superficie puedan ser despreciados, es decir el comportamiento global es similar al de un sistema infinito. A medida que nos acercamos al punto crítico, ζ crece, pero este crecimiento no puede ser indefinido, ya que ζ no puede ser superior a L . De esta manera, cuando los efectos de borde cobran relevancia y observaremos un comportamiento diferente al predicho por el límite termodinámico.

1.1.2.2. Hipótesis de universalidad

Uno de los aspectos más importantes del comportamiento de los sistemas. Cuando se encuentran a temperaturas cercanas de a un punto crítico es la hipótesis de universalidad, que se deriva de las fluctuaciones del parámetro de orden. Las fluctuaciones son grandes a temperaturas cercanas de un punto crítico y superiores al alcance del potencial de interacción, por lo que los detalles microscópicos dejan de ser relevantes y los sistemas en el límite termodinámico, independientemente de su naturaleza, presentan el mismo comportamiento. Formalmente se aplica a sistemas con la misma dimensionalidad espacial d y con la misma simetría del parámetro de

orden. Numerosas observaciones experimentales [10] apoyan dicha hipótesis.

En esta situación, la longitud de correlación ζ (distancia máxima a la cual puede detectarse correlación) es la única escala característica del sistema. Esta hipótesis nos permite expresar el comportamiento de muchas variables termodinámicas mediante funciones potenciales, que les simplifica notablemente. Estas funciones potenciales dependen de depende de la temperatura reducida [10]

$$\epsilon = \frac{T - T_c}{T_c} \quad (1.9)$$

donde T_c es la temperatura crítica. Por lo general, cuando una variable termodinámica se encuentra a una temperatura cercana a la crítica es posible escribirla como:

$$f(\epsilon) \approx \epsilon^\lambda \quad (1.10)$$

donde a λ se le conoce cómo el exponente crítico asociado a $f(t)$. En la tabla 1.1 se presenta ejemplos de algunos exponentes críticos.

Magnitud	Exponente crítico	Definición
Capacidad calorífica (V constante)	α	$c_v \approx \epsilon ^\alpha$
Diferencia de densidades	β	$\rho_L - \rho_G \approx \epsilon ^\alpha$
Susceptibilidad osmótica	γ	$\chi \approx \epsilon ^\gamma$
Longitud de correlación	ν	$\zeta \approx \epsilon ^\nu$

Tabla 1.1: Recoge la definición de algunos exponentes críticos. **Tabla modificada de [10]**

En el contexto de la Mecánica Estadística, un modelo se encuentra completamente definido a través de un Hamiltoniano. La existencia de universalidad en el comportamiento crítico posibilita una simplificación adicional; ya que, las propiedades entorno al punto critico son básicamente independientes de la mayoría de los detalles. De esta manera, se trata de formular para cada clase de universalidad un modelo mínimo que describa correctamente las propiedades criticas. Lejos de la criticidad estos modelos pueden resultar insuficientes para describir la física del sistema y será necesario introducir parámetros adicionales en el Hamiltoniano. Por otra parte, varios de estos modelos pueden adaptarse para la descripción de sistemas no magnéticos [9].

1.2. Sistemas Magnéticos

Un sistema magnético es un sistema cuyas partículas en sus estados fundamentales tienen momentos magnéticos permanentes (espín), dichas partículas interactúan entre ellas y con algún campo magnético externo que se le aplique al sistema. Los materiales pueden ser agrupados, en función de sus propiedades magnéticas [11] por ejemplo :

- Paramagnéticas.- En presencia de un campo magnético externo los momentos magnéticos se alinean en dirección del campo.
- Diamagnéticas.- En presencia de un campo magnético externo los momentos magnéticos se alinean en dirección contraria al campo.
- Ferromagnéticas.- Sin necesidad de un campo magnético externo los momentos magnéticos se alinean en una sola dirección, produciendo un propio campo magnético.
- Antiferromagnéticas.- En ausencia de campo magnético externo los momentos se alinean de forma alternada.

Para modelar los sistemas magnéticos se considera que el sistema tiene dos contribuciones. La primera contribución considera todas las interacciones internas. La segunda contribución corresponde a la interacción de los momentos magnéticos de cada molécula o átomo que interactúan con un campo magnético. Con estas consideraciones se puede definir el hamiltoniano H de la siguiente manera

$$H = H_0 - B \sum_i s_i \quad (1.11)$$

donde H_0 es la energía correspondiente a las interacciones internas, B es el campo magnético y s_i es el momento magnético de cada partícula.

Para estos modelos se definen variables importantes para su estudio como: la magnetización (1.12) de un estado la cual se define, como la suma de los momentos magnéticos del sistema.

$$M = \frac{1}{N} \sum_i s_i \quad (1.12)$$

Donde N es el número de partículas del sistema. Para la magnetización, se define la susceptibilidad magnética; como, la variación de la magnetización con respecto al campo magnético:

$$\chi = \frac{\partial \langle M \rangle}{\partial B} \quad (1.13)$$

Utilizando la ecuación (1.13), la definición de cantidades promedios (1.5) y la función de partición (1.4) con unas simples manipulaciones algébricas se tiene (1.14).

$$\chi = \frac{1}{k_b T^2} \left(\langle M^2 \rangle - \langle M \rangle^2 \right) \quad (1.14)$$

1.2.1. Modelo de Ising

El modelo de Ising, es uno de los modelos más simples e importantes utilizados para modelar un sistema magnético y este consiste en un sistema partículas cuyo momento magnético está caracterizado por su espín. En este modelo se estudian partículas de espín $1/2$ que se encuentran en dos estados $s = 1/2$ que representa el espín que apunta hacia arriba y $s = -1/2$ que representa el espín que apunta hacia abajo. Además en este modelo se considera que, cada partícula solo interactúa con sus vecinos más cercanos. Al considerar únicamente las interacciones internas, el Hamiltoniano en el modelo de Ising es:

$$H = -J \sum_{\langle k,l \rangle} s_k \cdot s_l \quad (1.15)$$

El modelo de Ising en una dimensión no presenta transición de fase; mientras que, en dos dimensiones en ausencia de campo magnético presenta transición de fase. La solución analítica del modelo de Ising en dos dimensiones sin campo magnético externo se denomina solución de Onsager [12].

La función de partición es la suma de exponenciales, y la suma de funciones continuas e infinitamente diferenciales es analítica, entonces aparentemente sus derivadas no podrían ser discontinuas, es decir no podría existir transiciones de fase. Sin embargo la solución de Onsager utiliza el límite termodinámico, es decir el sistema tiene un número infinito de estados. En este límite la suma no converge y se puede obtener una transición de fase [13].

A partir de la solución de Onsager se pueden calcular el propiedades termo-

dinámicas como: la magnetización, la capacidad calorífica y la susceptibilidad magnética. Las soluciones exactas de este modelo se observan en la siguientes figuras.

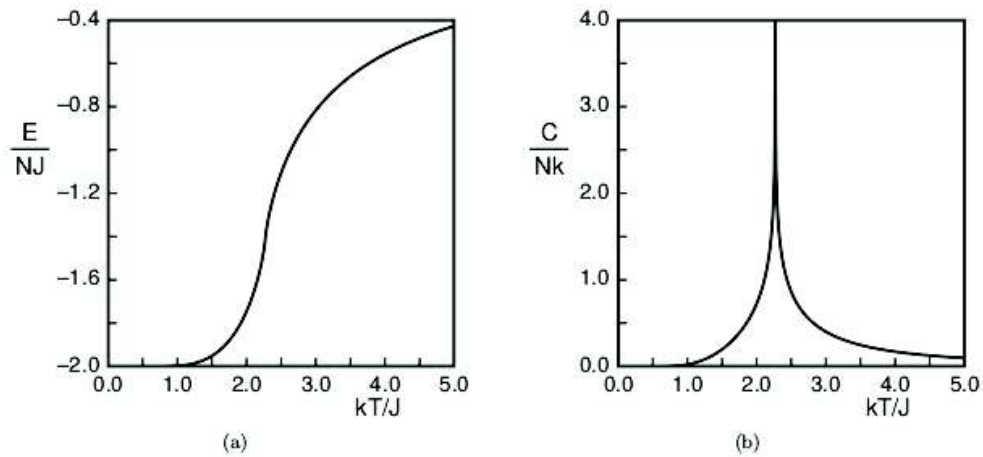


Figura 1.2: Soluciones analíticas en el modelo de Ising a) Energía b) Capacidad calorífica **Figura tomada de [12]**

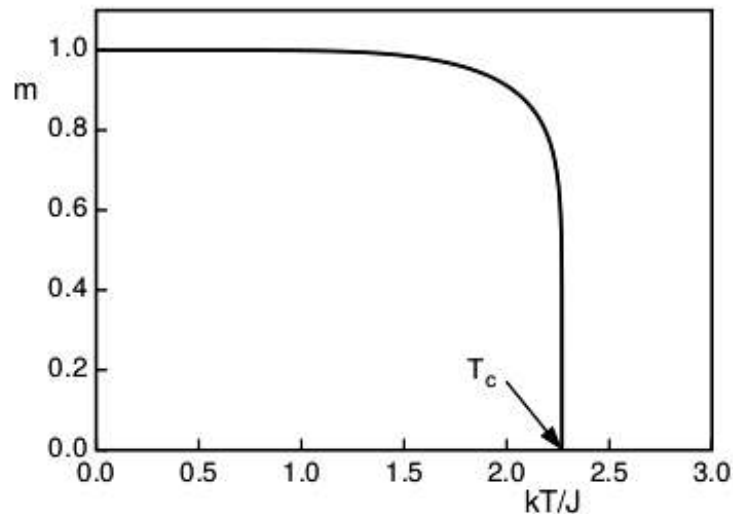


Figura 1.3: Magnetización por espín modelo de Ising **Figura tomada de [12]**

En la figura 1.2.b se puede observar la transición de fase dada por una divergencia en la capacidad calorífica en la temperatura crítica. La figura 1.2 muestra que la magnetización decae drásticamente a temperatura crítica, donde se produce la transición de fase.

En el modelo de Ising los momentos magnéticos se alinean de forma que la energía disminuye. Esto ocurre solo entre los vecinos más cercanos. A altas temperaturas, debido a fluctuaciones térmicas, los espines muy lejanos no se correlacionan.

Mientras a bajas temperaturas los espines lejanos pueden estar correlacionados. Esto permite la formación de dominios magnéticos. Se definen dos clases de dominios magnéticos de espín, hacia arriba (+1/2) y hacia abajo (-1/2). Estos dominios magnéticos pueden ser visto como regiones en el estado ferromagnético.

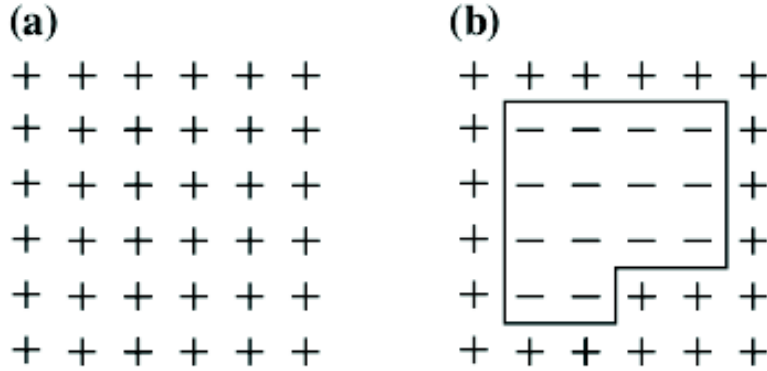


Figura 1.4: Dominios magnéticos **Figura tomada de [14]**

Estos dominios magnéticos coexisten alrededor de la temperatura crítica. Como dos estados coexisten entonces se puede definir un parámetro de orden, que está determinado por la diferencia entre espines que apuntan hacia arriba y espines que apuntan hacia abajo dividido para la cantidad total de espines. Entonces, la magnetización es el parámetro de orden en el modelo de Ising. Tomando en cuenta que en el modelo de Ising el parámetro de orden son momentos magnéticos o espines, se define el espacio del parámetro de orden el conjunto de todos los valores que puede tomar los espines en este caso es el conjunto $\{1/2, -1/2\}$

Una clase de Universalidad es una agrupación de fenómenos que se pueden modelizar de la misma forma utilizando exponentes críticos. El modelo de Ising pertenece a una clase de universalidad donde dos estados coexisten. El modelo de Ising permite modelizar otros fenómenos que pertenecen a esta clase de Universalidad como la opalescencia crítica de líquido.

Uno de los exponentes críticos más importantes es el relacionado a la longitud de correlación que determina el alcance de la interacción y se define como

$$\langle s(x)s(x+r) \rangle \approx e^{-1/\zeta} \quad (1.16)$$

donde el exponente crítico asociado a la longitud de correlación está dado por

$$\zeta = |\epsilon|^\nu \quad (1.17)$$

donde $\nu = -1$ para el modelo de Ising es decir la longitud de correlación diverge en el punto crítico esto significa que se tiene una interacción de largo alcance y permite que espines muy lejanos estén correlacionados y se produzca la transición.

Para sistemas reales la longitud de correlación no puede ser infinita dado que un sistema tiene un tamaño finito. Es decir el tamaño del sistema es importante para analizar las transiciones que se producen, ya que en el punto crítico se alejan del límite termodinámico.

1.2.2. Modelo XY

El modelo XY es una modificación al modelo de Ising donde se tiene un arreglo de espines con diferente orientación hacia arriba o hacia abajo. Mientras, en el modelo XY las partículas magnéticas pueden tener cualquier orientación en el plano como se puede observar en la siguiente figura.

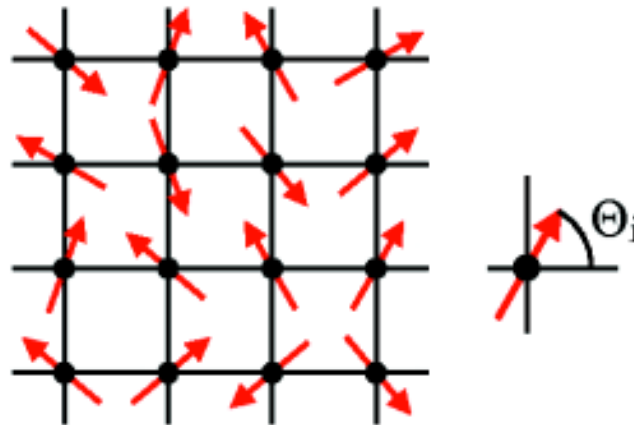


Figura 1.5: Red de espines en el modelo xy caracterizados por un ángulo

La energía de interacción entre dos espines se dan; tal como, en el modelo de Ising, por el producto de sus espines. Como todas las partículas se consideran idénticas el espín vectorial localmente queda caracterizado únicamente por el ángulo formado con el eje x , es decir $\vec{s} = s \cdot \vec{u}_\theta$. Este ángulo permite escribir la interacción entre dos espines como $-j\vec{s}_k \cdot \vec{s}_l = js^2 \cos(\theta_k - \theta_l) = -J \cdot \cos(\theta_k - \theta_l)$, obteniendo así la energía o hamiltoniano H_0 total del sistema como [15]:

$$H_0 = -J \sum_{\langle k,l \rangle} \cos(\theta_k - \theta_l) \quad (1.18)$$

En este hamiltoniano el termino J debe tener unidades de energía, y se denomina constante de acople entre los espines. El modelo XY tiene la siguiente función de partición [15].

$$\begin{aligned} \mathcal{Z} &= \sum_i e^{-\beta J \sum_{\langle k,l \rangle} \cos(\theta_k - \theta_l)} \\ &= \sum_i \prod_{\langle k,l \rangle} e^{-\beta J \cos(\theta_k - \theta_l)} \end{aligned} \quad (1.19)$$

La correlación entre los espines es una magnitud que es máxima cuando los espines están alineados y mínima cuando están anti-alineados. De esta forma se puede expresar como un producto punto. Además para medir si la interacción es de corto o largo alcance se debe considerar que los espines se encuentran a una determinada longitud r . Como existen varios espines que se encuentran a una misma distancia r en un modelo bidimensional se tiene que tomar un promedio espacial entre todos los espines que estén a esa longitud r . Finalmente como es una magnitud del ensamble se requiere tomar un promedio en el ensamble [15]. De esta forma se puede definir la función de correlación de espines como:

$$G(r) = \frac{1}{n_r} \left\langle \sum_r \vec{s}_i \cdot \vec{s}_{j(r)} \right\rangle \quad (1.20)$$

donde n_r es el numero de espines que están a distancia r , el sumatorio considera todos los posibles pares de espines que estén a distancia r .

Los defectos topológicos son metaestables [16]. Y disminuyen la correlación del sistema, lo que impide que a bajas temperaturas, se tenga una ruptura espontánea de simetría y se obtenga una magnetización. Por ese motivo se tiene solo cuasi-orden de largo alcance y la función correlación tiene dos comportamientos asintóticos [15]:

$$\lim_{r \rightarrow \infty} G(r) = \begin{cases} c_1 r^{-\eta} & T < T_c \\ c_2 e^{-r/\zeta} & T > T_c \end{cases} \quad (1.21)$$

Donde el exponente ζ es longitud de correlación del sistema y diverge en la temperatura crítica, η es un exponente que mide el decaimiento de la correlación y c_1 y c_2 son constantes.

El comportamiento asintótico de la función de correlación es, para temperaturas mayores a la temperatura crítica es una función exponencial. Por debajo de la tem-

peratura crítica es una función de potencia, es decir se puede aplicar la Hipótesis de Universalidad.

Gracias a la Hipótesis de Universalidad y el comportamiento de algunas variables termodinámicas que se comportan como exponentes críticos por debajo de la temperatura crítica se puede modelar sistemas como la superconductividad, superfluidez y la coherencia de láseres. Todos los fenómenos que pertenecen a esta misma clase de universalidad tiene los siguientes exponentes críticos:

$$\zeta_{\infty}(t) \approx e^{bt^{\nu}} \quad (1.22)$$

$$C_{\infty}(t) \approx \zeta^{-2} \quad (1.23)$$

$$\chi_{\infty}(t) \approx \zeta^{2-\eta_c} \quad (1.24)$$

donde b es un parámetro no universal, $\nu = 1/2$ y en la temperatura crítica $\eta = 1/4$ [1]. El estudio del presente trabajo se enfoca en el modelo XY en una red cuadrada con condiciones de fronteras periódicas.

1.2.2.1. Vórtices

A baja temperatura, los efectos térmicos son menos importantes y el sistema tiende a un estado de menor energía. La mínima energía de interacción entre dos espines es cuando están alineados, esto ocasiona que a baja temperatura los espines tiendan a alinearse con sus vecinos más cercanos. Esta tendencia a estar alineados en una vecindad es análogo a la continuidad de un campo. Pero existen puntos en la red que no pueden estar alineados con ninguno de sus vecinos, estos puntos se denominan vórtices.

En el modelo XY se denomina espacio del parámetro de orden, al igual que en el modelo de Ising al conjunto de estados que puede tener el espín. Sin embargo en el modelo XY la magnetización no es el parámetro de orden.

Como el espín puede tomar cualquier orientación y está caracterizado por un ángulo este queda representado como un punto en una circunferencia, es decir que el parámetro de orden es una circunferencia. Como se puede observar en la siguiente figura.

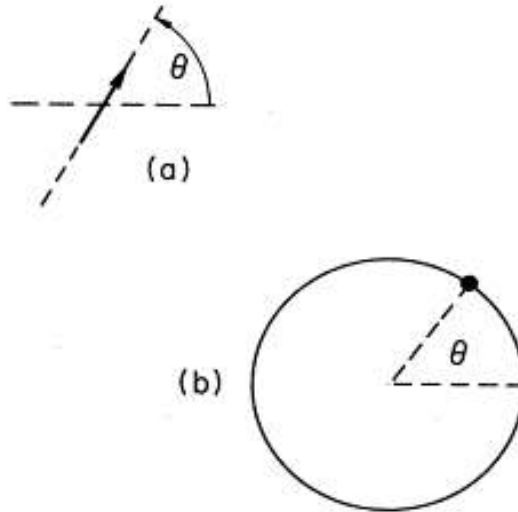


Figura 1.6: Se puede observar como la dirección del espín representa un punto en el parámetro de orden **Figura tomada de [16]**

Una curva que une espines cercanos y termina en el mismo punto se denomina lazo en la red de espines, este lazo nos determina otro lazo en el espacio de parámetros de orden. Si nos imaginamos una red de espines y un lazo circular, fijándonos únicamente en los espines que atraviesan el lazo nos queda algo similar a la figura del lado izquierdo de 1.7. Recorriendo el lado izquierdo de 1.7 se puede identificar con un punto en el lado derecho de 1.7. Por ejemplo, en la sub-figura (a) de 1.7 en el lado izquierdo todas las flechas apuntan hacia arriba en consecuencia en el lado derecho existe un solo punto de las flechas que apuntan hacia arriba. El lado izquierdo representa la red de espines y el derecho representa el espacio del parámetro de orden.

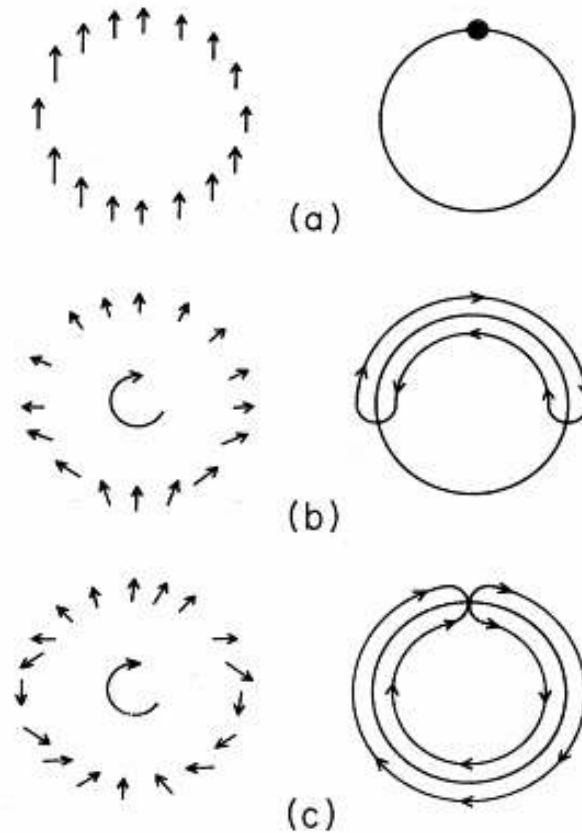


Figura 1.7: Vórtices en el espacio del parámetro de orden. **Figura tomada de [16]**

Un lazo en la red de espines determina un lazo en el espacio del parámetro de orden. Si se contrae el lazo en la red de espines, también se puede contraer el lazo en el espacio del parámetro de orden, pero hay ocasiones en las que el lazo en el parámetro de orden da un giro a la circunferencia y no se puede contraer lo que impide que el lazo la red tampoco pueda contraer lo que determina que el lazo encierra al menos un vórtice.

El índice de la curva en la red de espines es el número de vueltas que da la curva en el espacio del parámetro de orden. El índice de la curva en la red de espines que encierra un sola vórtice, se denomina índice del vórtice (v).

Otra forma de caracterizar v es realizando una integral de línea cerrada al rededor del centro del vórtice:

$$v = \frac{1}{2\pi} \oint_C d\vec{r} \cdot \vec{\nabla} \theta(\vec{r}) \quad (1.25)$$

Hay dos tipos de vórtices, los vórtices con un índice mayor que cero se denominan vórtices, y vórtices con índice menor que cero se denominan antivórtices que se

presentan en pares. En la siguiente figura se observa una configuración simple de vórtices en sentido horario y antivórtices en sentido antihorario en el modelo XY.

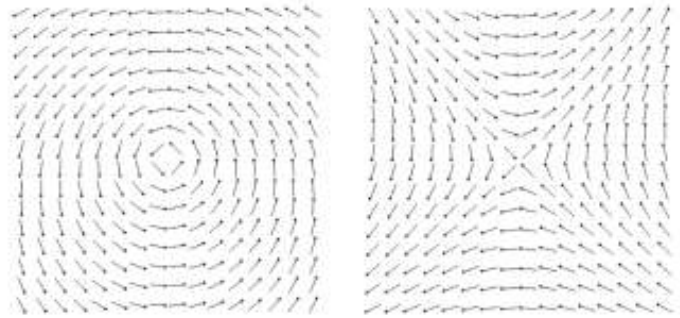


Figura 1.8: Una configuración simple de vórtices y antivórtices en el modelo XY
Figura tomada de [17]

Utilizando la descripción matemática descrita en el apéndice A.2.7. Se obtiene

$$\sum_i v_i = 0 \quad (1.26)$$

El teorema de Euler Poncaire nos demuestra que con el paso al continuo y condiciones de frontera periódicas existe una conservación entre vórtices y anti-vórtices y eso solo depende de la topología. Esto determina una ley de conservación que no depende de una simetría, es así que se muestra que el teorema de Noether no puede ser bi-condicional, es decir toda simetría tiene asociada una magnitud conservada, pero no toda magnitud conservada tiene una simetría asociada.

Debido a que los vórtices solo pueden desaparecer cuando un par antivortice-antivórtice se une y aniquila. Un vórtice es una formación de espín topológicamente estable [16].

Físicamente estos vórtices tienen asociada una energía [18]

$$E_v = \frac{J}{2} \int d^2r \left(\frac{1}{r} \right)^2 = J\pi \ln \frac{L}{a} \quad (1.27)$$

donde L es el tamaño del sistema, J la constante de acoplamiento y a el parámetro de red.

La entropía del vórtice es [18]:

$$S = 2k_b \ln \frac{L}{a} \quad (1.28)$$

El giro de los espines en el vórtice producen un cambio de orientación en los espines, estos cambios reducen la correlación del sistema.

1.2.2.2. Transición Kosterlitz y Thouless

El espacio del parámetro de orden en el modelo de Ising es discreto y en el modelo XY es continuo. Aunque es similar al modelo de Ising, Mermin Wagner demostraron que no existe magnetización en sistemas de dos dimensiones con simetría continua, es decir, la magnetización es cero a toda temperatura en el límite termodinámico.

En sistemas de dos dimensiones correlacionados de corto y cuasi-largo alcance esta demostrado que no puede haber presencia de magnetización en el límite termodinámico. En dos dimensiones la susceptibilidad magnética puede se expresada como una función de correlación de espines mediante la siguiente ecuación.

$$\chi = \frac{1}{2T} \sum_r G(r) \quad (1.29)$$

Como la longitud de correlación diverge en la temperatura crítica, la función susceptibilidad magnética también lo hace, eso produce que la susceptibilidad diverge y se tenga una transición de fase. Es decir, Al igual que el modelo Ising, la susceptibilidad magnética diverge a temperatura crítica, es decir existe un cambio de fase. Este cambio de fase es conocido como la transición (KT) .

La transición KT presenta un cambio en el comportamiento de la función de correlación entre espines. La correlación cambia de ser, de corto alcance en altas temperaturas que corresponde a un decaimiento exponencial, a cuasi largo alcance a bajas temperaturas decaimiento logarítmico.

A pesar que en la teoría de Landau no se puede definir un parámetro de orden en el sentido usual [19]. Utilizando la teoría de renormalización se puede definir un parámetro de orden gracias al grupo de Kosterlitz y Thouless. En la región de criticidad por encima de la temperatura crítica se forman vórtices que están en pares y muy juntos, bajo la temperatura crítica estos vórtices aportan energía al sistema, de forma que la energía crece con la aparición de vórtices, lo que impide que existan vórtices por debajo de la temperatura crítica [18].

Existen varias formas de calcular la temperatura de transición Kosterlitz y Thouless (TKT) utilizando simulaciones numéricas, en todas estas se requiere una corrección por efectos de tamaño. Una forma de calcular la temperatura crítica es calculan-

do la capacidad calorífica en función de la temperatura y fijarse en su máximo Otra forma de calcular la temperatura crítica de transición es la rigidez de giro que puede ser entendida como una fuerza que intenta imponer los giros en borde del sistema. La rigidez de giro puede ser calculada como se indica en la siguiente ecuación:

$$Y = -\frac{1}{2} \langle E \rangle - \frac{1}{T} \left\langle \left(\sum_{\langle i,j \rangle} \text{sen}(\theta_i - \theta_j) \vec{r}_{ij} \cdot \vec{u} \right)^2 \right\rangle \quad (1.30)$$

donde \vec{r}_{ij} es el vector que une el espín i con el espín j y \vec{u} es un unitario arbitrario. [6]. La rigidez de giro permite calcular la temperatura de crítica de transición porque cumple la siguiente ecuación.

$$Y(T_{KT}) = \frac{2}{\pi} T_{KT} \quad (1.31)$$

Generalmente los sistemas de tamaño finito tienen efectos de tamaño. Para poder eliminar estos efectos generalmente se realiza una corrección por efectos de tamaño con la siguiente ecuación (1.32).

$$T = T_{KT} + \frac{c_3}{\ln(L)} \quad (1.32)$$

donde L es la longitud de uno de los lados del sistema en una red cuadrada que determina el tamaño del sistema.

1.2.3. Modelo XY diluido

El modelo XY diluido consiste en una red cuadrada, donde cada posición de la red puede tener una partícula magnética o una impureza no magnética. Al igual que en el modelo XY, La partícula magnética tiene un momento magnético, representado por un vector de modulo unitario que puede girar libremente. Y la impureza es un espacio vacío que no interacciona.

Se define una unidad de área, de modo que, en cada unidad de área solo puede haber una partícula magnética o una impureza. De este modo se puede definir una concentración de espines ρ como el número de espines que tiene la red, dividido para el área total de la red o número de puntos de red. De esta misma manera, se define la dilución como el número de impurezas por unidad de área (c). En efecto $\rho + c = 1$.

Las impurezas en la red permiten relacionar la concentración de espines con la

teoría de percolación de materiales porosos. Esta teoría nos indica que a una concentración de orificios mínima que en nuestro caso, estaría representada por impurezas, el material poroso permite la formación de caminos entre una frontera y otra. A esta concentración mínima se denomina concentración de percolación. Para el caso de una red cuadrada la concentración de percolación es $\rho_c = 0,592746$ [20].

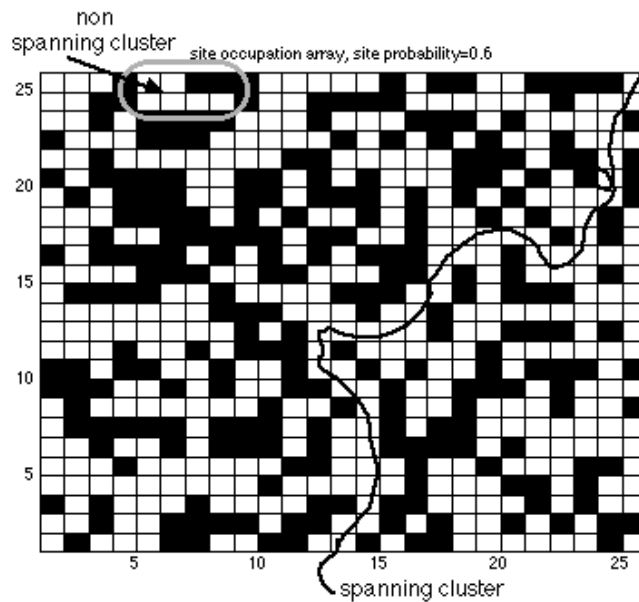


Figura 1.9: Percolación, se puede observar qué a una concentración de $\rho = 0,6$ el sistema permite la creación de caminos

Cabe mencionar que al día de hoy no existe acuerdo si la transición deja de ocurrir a concentración de percolación o una concentración menor, como se puede observar en la figura siguiente figura.

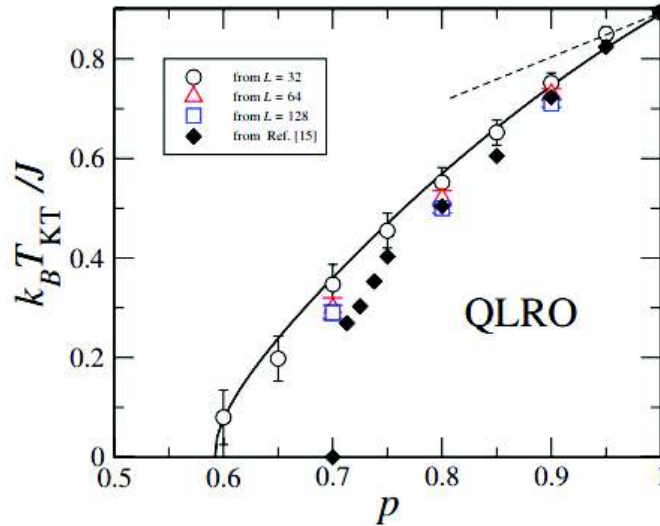


Figura 1.10: Efectos de impurezas para diferentes tamaños de red en la temperatura crítica **Figura tomada de [21]**

1.3. Métodos Montecarlo

Las simulaciones numéricas pueden ser consideradas como auténticos experimentos numéricos, en los cuales se estudia el comportamiento de un modelo simplificado. La ventaja frente a un experimento real es que en este caso poseemos un completo control sobre las condiciones microscópicas del problema. Así, por ejemplo, en tanto que un experimental puede controlar los parámetros macroscópicos (tales como temperatura o presión) sobre su muestra, en principio le resulta sumamente difícil determinar los efectos de una variación en el potencial inter-atómico. Por otra parte, dichas simplificaciones introducen una dificultad adicional ya que es posible que el modelo simplificado a analizar presente comportamientos nuevos, los cuales no estaban presentes en el modelo original. Por otra parte, de las limitaciones que se tiene en las simulaciones numéricas son las limitaciones del calculo computacional. Por ejemplo si quisiéramos realizar todos los casos posibles del espacio de fases a temperatura fija en el modelo de Ising en una red cuadrada de 10×10 se podría demorar tiempos en el orden de la edad del universo. Frente a esta limitación se realizan métodos numéricos que no exploran todos los casos; si no, que toman una muestra de forma aleatoria de estos. Los métodos que utilizan números pseudo aleatorios se denominan Montecarlo. Existen diferentes formas de tomar esta muestra, una forma podría simplemente explorar el espacio de forma totalmente aleatoria pero este también requiere gran número de simulaciones. Desde que se comenzaron a

realizar estas simulaciones computacionales se buscan métodos que otorguen soluciones fiables. Uno de los métodos más usuales y conocido por su relativa eficiencia y fácil implementación es el Método Monte Carlo Metrópolis.

El método Montecarlo Metrópolis, nos permite calcular promedios en una Cadena de Márkov. Para poder aplicar este método en la mecánica estadística es necesario entender las Cadenas de Márkov y la hipótesis ergódica.

1.3.1. Cadena de Márkov

Una cadena de Márkov es un proceso estocástico en el cual la probabilidad de transición de un estado a otro solo depende del estado anterior. Esto se define formalmente en el apéndice A.3. Una cadena de Markov puede ser entendida físicamente como una sucesión de experimentos, además cada experimento puede ser realizado sin una relación temporal específica [22].

Para entender la cadena de Markov tomaremos el simple ejemplo de lanzar una moneda, si el objeto de estudio es una moneda nuestro espacio muestral es cara o sello, el experimento aleatorio consiste en lanzar una moneda, la medición es observar si fue cara o sello. Cada lanzamiento de la misma moneda tiene que realizarse un lanzamiento después de otro, pero el tiempo de espera entre un lanzamiento y otro puede ser distinto en cada lanzamiento. En este ejemplo el espacio muestral es: No se lanza la moneda, sale cara, sale sello, sale cara o sello. Aun que los lanzamientos de la moneda sean independientes del tiempo, por su naturaleza secuencial, se denomina en teoría de los procesos estocásticos como variable temporal. En nuestro caso el espacio muestral representa el ensamble. Es decir un micro-estado representa un punto muestral y un macro-estado es un evento. Cuando se termina de realizar una serie de experimentos uno después de otro se determina una serie de eventos para ese experimento. Esta serie de eventos puede ser caracterizado por una función denominada, función muestra $x(t)$, que es totalmente determinista en la variable t , que es la variable temporal. No olvidemos que con t fijo, antes de realizar el experimento su resultado es aleatorio.

La probabilidad $P(x_1, t | x_2, t + 1)$, es la probabilidad condicional de que estando en el estado x_1 a tiempo t pase al estado x_2 al tiempo $t + 1$ y se notará $P_t(x_1 \rightarrow x_2)$. Además por simplicidad se elige la probabilidad de tal manera que al cabo de ciertos tiempos sucesivos los estados obedecen una distribución estacionaria $P(x)$. Para llegar a esta distribución se requiere que $P(x_1 \rightarrow x_2)$ satisfaga las hipótesis de balance detallado y accesibilidad, detallados a continuación

1) Accesibilidad: Dados dos estados x y x' , existe una probabilidad de que se pueda llegar de un estado a otro. Matemáticamente esto significa que existe una sucesión finita de estados x_1, x_2, \dots, x_n que satisfacen.

$$P(x \rightarrow x_1)P(x_1 \rightarrow x_2)\dots P(x_n \rightarrow x') \neq 0 \quad (1.33)$$

2) Balance detallado (o micro reversibilidad): Esta hipótesis nos dice que el sistema puede llegar a un equilibrio y no depender de una variable temporal.

$$P(x)P(x \rightarrow x') = p(x')P(x' \rightarrow x) \quad (1.34)$$

1.3.2. Hipótesis Ergódica

Para un tiempo fijo t la variable X_t es una variable aleatoria, y se puede tomar un promedio. Multiplicando cada valor x_i por la probabilidad de que la variable aleatoria X_t tome el valor x_i y sumando todos los productos de los posibles valores. $\langle X(t) \rangle$ se denomina promedio estadístico de la variable aleatoria X al tiempo t como se indica en la siguiente ecuación:

$$\langle X_t \rangle = \sum x_i P(X_t = x_i) \quad (1.35)$$

por otro lado supongamos que el sistema, esta evolucionando en el tiempo y cada cierto tiempo se toma una medida y al final se toma un promedio de estas medidas. Esto es totalmente análogo a la función muestra, definida en el inciso anterior. En cada posible función muestra, se puede definir un promedio denominado, promedio temporal como:

$$\langle x(t) \rangle = \lim_{n \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N x(t) \quad (1.36)$$

Un proceso estocástico es ergódico cuando se cumple la ecuación $\langle X_t \rangle = \langle x(t) \rangle$, en cualquier función muestra y cualquier variable aleatoria.

La hipótesis ergódica en física se refiere a que un promedio en el ensamble estadístico es equivalente a un promedio en el tiempo.

1.3.3. Método Montecarlo Metrópolis

El proceso Montecarlo Metrópolis nos permite crear una función nuestra y gracias a la Hipótesis ergódica, esta función muestra, representa un ensamble. Este método se lo realiza mediante los siguientes pasos:

1. Se define un estado inicial aleatorio.
2. A partir del estado anterior se genera un nuevo estado.
3. Se calcula la probabilidad de transición $P_{anterior \rightarrow nuevo}$ (P_{an}) entre el estado anterior y el nuevo.
4. Se compara la probabilidad de transición con un número r entre 0 y 1 distribuido aleatoriamente.
 - Si $r < P_{an}$ se acepta el cambio, es decir el nuevo se convierte en el viejo.
 - Si $r > P_{an}$ se rechaza el cambio, es decir no se altera el estado quedándonos con el estado viejo y descartando el nuevo estado.
5. Se repite desde el segundo al cuarto paso, una gran cantidad de veces que dependerá del sistema de estudio.

Capítulo 2

Metodología

2.1. Implementación del sistema

2.1.1. Implementación de la red

El sistema está caracterizado por una red de $L \times L$ espines con condiciones de frontera periódica. Para este objetivo se genera una matriz $\{s_{ij}\}$ con $i, j = 0, 1, \dots, L + 1$. Esta matriz guarda el ángulo del espín.

Las filas y columnas correspondientes a 0 y $L + 1$ permiten definir las condiciones de frontera. Al comenzar la simulación, toda la fila 1 es copiada en la fila $L + 1$, toda la fila L es copiada en la fila 0, toda la columna 1 es copiada en la columna $L + 1$ y toda la columna L es copiada en la columna 0. Las filas y columnas 0 y $L + 1$ son filas auxiliares que sirven para determinar las condiciones de frontera periódicas y no son consideradas parte de la red. En efecto la frontera de la red son las filas y columnas correspondientes a 0 y 1. Cada vez que en el transcurso de la simulación un espín de la frontera de la red cambia, se cambia su correspondiente fila o columna auxiliar.

Para considerar una dilución del sistema, es necesario guardar en la memoria la ubicación de cada espín e impureza; para esto, se genera una matriz sd_{ij} que contiene únicamente 0 y 1. El número 1 representa que el espín existe y 0 para denotar que es una impureza en esa posición. Esta red tiene las mismas características que la red principal con respecto su tamaño y las condiciones de frontera. Para generar esta matriz primero se considera la concentración de espines. Para concentraciones de espines menores que 0,9, se llena la matriz completamente de 1 y después por cada impureza se coloca aleatoriamente dentro de la red un 0. Para concentraciones de

espines mayores que 0,9 se llena la red con 1 y las fronteras auxiliares con 0, se encoje aleatoriamente un sitio de la red, si alrededor de este sitio no existen 0 se coloca un 0 en ese sitio de red. Finalmente se aplican las condiciones de frontera como se indica en el inciso anterior.

Para poder acceder de forma aleatoria solo a cada espín sin considerar las impurezas, se enumera todos los sitios de la matriz incluyendo la frontera auxiliar de forma consecutiva con un solo índice. Después se genera una lista que contiene solo los índices de los sitios de red que contienen espines, es decir solo los los puntos de la matriz que tienen 1 dentro de la red. A esta lista le denominaremos índice del espín k

2.1.2. Propiedades físicas del sistema

Para el calculo de propiedades físicas del sistema. Se realiza una suma utilizando solo los valores en la lista índice del espín k como se indica en cada propiedad en cada sistema.

2.1.2.1. Energía

Como en el modelo XY, la interacción solo se realiza con los vecinos más sécanos, para cada espín k en la red se suma la contribución de sus cuatro vecinos. Cómo existen impurezas, la contribución de la impureza no debe ser tomada en cuenta, para esto, se multiplica por 0 a la contribución de la impureza. Como la matriz s_{ij} contiene cero para cada impureza, basta con multiplicar a la contribución de cada interacción entre espines por el valor de la matriz sd_{ij} . Cabe recalcar qué, como la matriz sd_{ij} contiene un 1 en cada espín, al multiplicarla por esta no produce efecto alguno.

$$\begin{aligned}
 E = \frac{1}{2} \sum_k \{ & \cos(s_{i(k),j(k)} - s_{i(k),j(k)+1}) \cdot sd_{i(k),j(k)+1} \\
 & + \cos(s_{i(k),j(k)} - s_{i(k),j(k)-1}) \cdot sd_{(k)i,j(k)-1} \\
 & + \cos(s_{i(k),j(k)} - s_{i(k)+1,j(k)}) \cdot sd_{i(k)+1,j(k)} \\
 & + \cos(s_{i(k),j(k)} - s_{i(k)-1,j(k)}) \cdot sd_{i(k)-1,j(k)} \}
 \end{aligned} \tag{2.1}$$

2.1.2.2. Magnetización

Para el cálculo de la magnetización se hace por separado la magnetización en x y la magnetización en y , y se realiza la misma consideración de impurezas que con la energía con la matriz $sd_{i,j}$.

$$M_x = \sum_k \cos(s_{i(k),j(k)}) \cdot sd_{i(k),j(k)} \quad (2.2)$$

$$M_y = \sum_k \sin(s_{i(k),j(k)}) \cdot sd_{i(k),j(k)} \quad (2.3)$$

Finalmente:

$$M = \sqrt{M_x^2 + M_y^2} \quad (2.4)$$

2.1.2.3. Función de correlación

1. Formar todas las posibles parejas de espines y clasificarlos de acuerdo a su distancia.
2. Se promedia el producto de cada pareja de espines clasificados por distancia.

Con este procedimiento se obtiene una función $G_e(r)$ que depende de cada estado del sistema.

2.1.2.4. Vórtices

En cada sistema se cuenta el número de vórtices de la siguiente manera.

Sin impurezas: Se inicializa un contador de vórtices en cero, y se recorre cada uno de las celdas de la red, cuando cumple la siguiente condición el contador se incrementa en uno

1. La mínima celda de la red es un arreglo de 2×2 se comienza en la esquina superior izquierda en sentido antihorario a contabilizar cambio de ángulo;
2. Se suma los cambios al rededor de un ciclo, se dice que se tiene un vórtice cuando el cambio es 2π y un antivortice cuando el cambio es de -2π
3. Se guarda la posición de la singularidad.

Con impurezas: Se recorre cada una de las impurezas y se incrementa el mismo contador cuando se cumple la siguiente condición.

1. Se recorre el contorno del hueco;
2. Se suma los cambios al rededor de un ciclo;
3. Se dice que se tiene un vórtice cuando el cambio es múltiplo entero de 2π y antivórtice cuando el cambio es múltiplo entero de $2\pi..$

2.2. Implementación del Ensamble

2.2.1. Evolución del sistema

1. Parámetros fijos para todo el algoritmo: temperatura, tamaño de red.
2. Generar un arreglo aleatorio de espines S_0 .
3. Calcular el hamiltoniano del arreglo con la ecuación 2.1.
4. Modificar el sistema anterior S_0 y generar un sistema levemente modificado S_f , cambiando un solo espín. El espín es escogido de forma uniformemente aleatoria.
5. Se calcula la diferencia de energía con el sistema anterior $\Delta H = H_f - H_0$
6. Se calcula $p_0 = e^{-\beta\Delta H}$ y se genera un número aleatorio r entre cero y uno.
7. Si $p_0 > r$ se acepta el cambio, entonces, el sistema modificado se convierte en el nuevo S_0 . Caso contrario se mantiene el anterior.
8. Se repite los pasos del 3 hasta el 7
9. Se repite el paso anterior 600 000 veces en total (300 000 veces hasta que el sistema llegue a un equilibrio y 300 000 para promediar).
10. Se calculan las propiedades del estado y se promedia los últimos 300 000 pasos.

2.2.2. Cálculo de propiedades físicas del Ensemble

Cantidades promedio

Para el ensemble gracias a la hipótesis ergódica, se puede calcular los promedios en el ensemble tomando promedios entre los pasos de la simulación Montecarlo. Esta es modificación de la ecuación (1.36), donde el límite al ser una simulación computacional se calcula con un número muy grande de pasos que en la bibliografía [8] sugiere en el orden de 10^5 .

$$\langle x(t) \rangle = \frac{1}{N} \sum_{n=1}^N x(t) \quad (2.5)$$

Capacidad Calorífica

Utilizando el cálculo de propiedades y la ecuación (1.7), se obtiene la siguiente ecuación.

$$C_v = \frac{k_b}{T^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right) \quad (2.6)$$

Susceptibilidad Magnética

Utilizando el cálculo de propiedades y la ecuación (1.14) se obtiene la siguiente ecuación.

$$\chi = \frac{k_b}{T^2} \left(\langle M^2 \rangle - \langle M \rangle^2 \right) \quad (2.7)$$

Rigidez de giro

Utilizando el cálculo de propiedades y la ecuación (1.30). Se considera únicamente los vecinos más cercanos. Para el vector unitario \vec{u} se utiliza el vector unitario en el eje x . Con estas consideraciones se obtiene la siguiente ecuación:

$$Y = -\frac{1}{2} \langle E \rangle - \frac{1}{T} \left\langle \left(\sum_k \text{sen}(s_{i(k),j(k)} - s_{i(k)-1,j(k)}) - \text{sen}(s_{i(k),j(k)} - s_{i(k)+1,j(k)}) \right)^2 \right\rangle \quad (2.8)$$

Exponentes críticos

Primero se promedia la función de correlación de forma temporal para cada distancia.

$$G(r) = \langle G_e(r) \rangle \quad (2.9)$$

Como se han realizado muchas mediciones debido al costo computacional y por simplicidad no se realiza una regresión por “minimum likelihood”, aún que no se ha comprobado si la distribución de errores son gaussianos se realiza una regresión por mínimos cuadrados. No obstante, para mejorar la regresión no se linealizan los valores de los puntos y se realiza una optimización numérica no lineal de la ecuación que se genera al aplicar el método de mínimos cuadrados con el método de Newton Raphson, como se describe a continuación; de cada caso.

2.2.2.1. Regresión por ley Exponencial

Para realizar esta regresión, se utiliza mínimos cuadrados con el siguiente modelo.

$$f(x) = a \cdot e^{b \cdot x} \quad (2.10)$$

Utilizando un conjunto de puntos $\{(x_i, y_i)\}$ de datos, la distancia a optimizar es:

$$d(a, b) = \sum_i (y_i - a \cdot e^{b \cdot x_i})^2 \quad (2.11)$$

En efecto la función de distancia solo depende de a y b , Tomando el gradiente e igualando a cero se obtiene las ecuaciones que esta función satisface:

$$\nabla d = \begin{pmatrix} \frac{\partial d}{\partial a} \\ \frac{\partial d}{\partial b} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.12)$$

Se obtiene un sistema de dos ecuaciones con dos incógnitas no lineales

$$\frac{\partial d}{\partial a} = 2 \sum_i (y_i - a \cdot e^{b \cdot x_i}) \cdot e^{b \cdot x_i} = 0 \quad (2.13)$$

$$\frac{\partial d}{\partial b} = -2 \sum_i (y_i - a \cdot e^{b \cdot x_i}) (a \cdot x_i \cdot e^{b \cdot x_i}) = 0 \quad (2.14)$$

De la ecuación (2.13) se puede despejar a y reemplazar en (2.14) y después de manipulaciones algébricas se obtiene:

$$\sum_{ij} \left((y_i x_i - y_j x_j) \cdot e^{b(x_i + 2x_j)} \right) = 0 \quad (2.15)$$

El lado izquierdo de la ecuación (2.15) es una función $F(b)$ que solo depende de b . Se puede utilizar el método de Newton para poder encontrar un cero para esta función. El método consiste en a partir de $b_0 = 0$ se aproxima el valor de b con un b_k con k suficientemente grande mediante la siguiente formula de recursividad:

$$b_{i+1} = b_i - \frac{F(b)}{F'(b)} \quad (2.16)$$

El criterio de suficientemente grande de modo que $\frac{F(b)}{F'(b)} < 10^{-10}$ o $k < 1000$.

2.2.2.2. Regresión por ley de Potencias

Para realizar esta regresión se utiliza mínimos cuadrados con el siguiente modelo.

$$f(x) = a \cdot x^b \quad (2.17)$$

Utilizando un conjunto de puntos $\{(x_i, y_i)\}$ de datos, la distancia a optimizar es:

$$d(a, b) = \sum_i (y_i - a \cdot x_i^b)^2 \quad (2.18)$$

En efecto, la función de distancia depende de a y b , Tomando el gradiente e igualando a cero se obtiene las ecuaciones que esta función cumple:

$$\nabla d = \begin{pmatrix} \frac{\partial d}{\partial a} \\ \frac{\partial d}{\partial b} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.19)$$

Se obtiene un sistema de dos ecuaciones con dos incógnitas no lineales

$$\frac{\partial d}{\partial a} = 2 \sum_i (y_i - a \cdot x_i^b) \cdot x_i^b = 0 \quad (2.20)$$

$$\frac{\partial d}{\partial b} = -2 \sum_i (y_i - a \cdot x_i^b) (a b \cdot x_i^{b-1} \ln(x_i)) = 0 \quad (2.21)$$

De la ecuación (2.20) se puede despejar a y reemplazar en (2.21) y después de manipulaciones algébricas la ecuación (2.21) queda como:

$$\sum_{ij} \left((y_j \ln(x_i) - y_i \ln(x_j)) \cdot x_j^b x_i^{2b} \right) = 0 \quad (2.22)$$

El lado izquierdo de la ecuación (2.22) es una función $F(b)$ que solo depende de b . Se puede utilizar el método de Newton para poder encontrar un cero para esta función. El método consiste en a partir de $b_0 = 0$ se aproxima el valor de b con un b_k con k suficientemente grande mediante la siguiente formula de recursividad:

$$b_{i+1} = b_i - \frac{F(b)}{F'(b)} \quad (2.23)$$

El criterio de suficientemente grande de modo que $\frac{F(b)}{F'(b)} < 10^{-10}$ o $k < 1000$. Los resultados obtenidos mediante este algoritmo se compararon con los obtenidos mediante el comando fit de GNU PLOT

2.3. Cálculo de la transición

2.3.1. Método Cv

Para calcular temperatura crítica de transición se puede tomar en cuenta el máximo de la capacidad calorífica. Numéricamente se calcula la capacidad calorífica para cada temperatura utilizando el método descrito anteriormente.

2.3.2. Método Rigidez de Giro

Para calcular la temperatura crítica de transición se puede tomar el punto de intersección entre la Rigidez de giro y la recta $\zeta = \frac{T}{\pi}$. Tomando en cuenta que no se puede encontrar exactamente el punto que cruza con la recta, se toma rectas desde el origen a cada punto calculado se compara las pendientes, la diferencia mas pequeña de la pendiente es el punto que se considera como el punto de transición.

2.3.3. Efectos de tamaño en la temperatura crítica

Una vez calculada la temperatura crítica con cada método para diferentes tamaños. Se utiliza la ecuación 1,32 descrita en el marco teórico.

$$T = T_{KT} + \frac{c_3}{\ln(L)} \quad (2.24)$$

donde c_3 es una constante de ajuste del modelo.

2.3.4. Efectos de dilución

Observando los resultados de la figura 1.10, por simplicidad se realiza una regresión lineal por mínimos cuadrados a la temperatura crítica de transición que resulta de realizar la corrección por efectos de tamaño. Se repite el procedimiento para otras diluciones tomando en cuenta que las diluciones por efecto de tamaño sean las mismas. Finalmente se realiza una regresión lineal por mínimos cuadrados de la temperatura crítica en función de la concentración de espines.

$$T_{KT} = \alpha\rho \quad (2.25)$$

Capítulo 3

Análisis de Resultados

A continuación se presentan los resultados para una red cuadrada de $L \times L$ con condiciones de frontera periódica. En cada punto de red se colocan espines o impurezas aleatoriamente. Este sistema se estudia como un proceso estocástico en su forma más simple como una cadena de Márkov en la cual la probabilidad de estar en un estado solo depende del estado anterior. La simulación se ha conseguido utilizando el Método Monte-Carlo. Para inicializar la simulación se requiere, como datos: la temperatura, el número de impurezas y el tamaño de red. Variando estos parámetros se analiza el comportamiento de la energía, capacidad calorífica, magnetización, susceptibilidad magnética, rigidez de giro, vórtices, función de correlación y temperatura de transición.

En la figura 3.1 se muestra el tiempo que se demora en realizar en un núcleo de procesamiento una simulación sin impurezas en cualquier temperatura.

La regresión presentada indica que el tiempo con respecto al tamaño esta ajustada por la siguiente ecuación $t = 32,5L^2$. Esta regresion se realizo utilizando el comando fit de GNUPLOT. Esto significa que para realizar una simulación Monte Carlo se requiere aproximadamente 32,5 segundos por espín. En efecto en una red cuadrada de 10×10 que tiene 100 espines se demora aproximadamente $3000s = 50$ minutos y una red de 30×30 se demora $29000s \approx 8$ horas . Como se requiere realizar varias simulaciones, una para cada temperatura. Se realizo primero 24 simulaciones de con pasos de $\Delta T = 0,85J/k_b$ y después al rededor del punto más próximo a al pico de la capacidad calorífica se realizo adicionalmente 33 simulaciones con pasos de $\Delta T = 0,004J/k_b$. Finalmente, alrededor del punto más próximo a la intersección entre la rigidez de giro y la recta con pendiente $2/\pi$ se realizaron 33 simulaciones más.

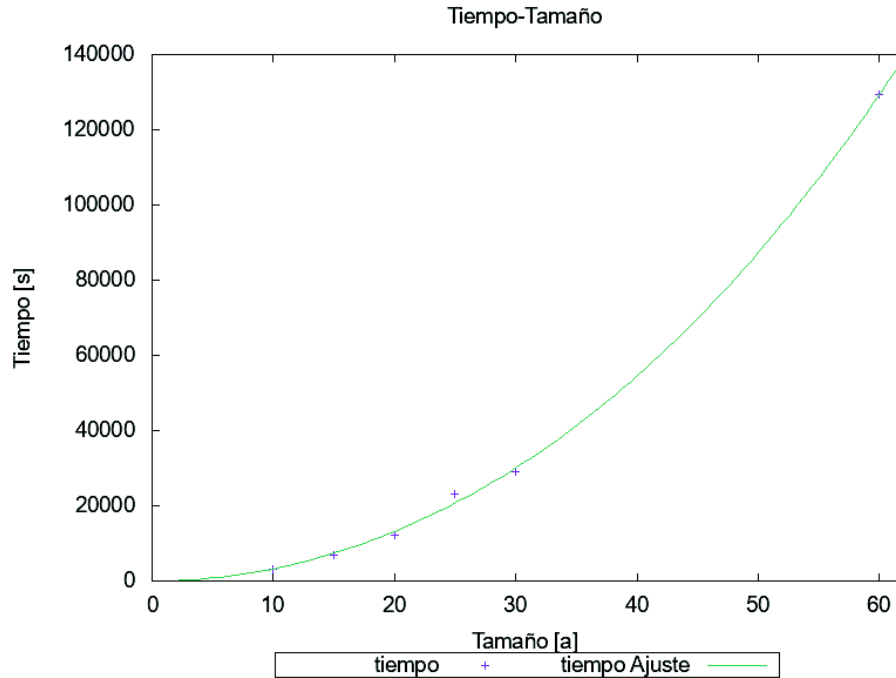


Figura 3.1: Tiempo

Dando un total de 90 simulaciones para una concentración constante y un tamaño de red constante. Las simulaciones se realizaron en Centro de de Modelización Matemática MODEMAT de la Escuela Politécnica Nacional se usaron un total de 480 núcleos de procesamiento distribuidos en 20 nodos de octava generación CPU INTEL 2695, se logró realizar simulaciones de hasta un tamaño de 60×60 . Este proyecto utilizó los servidores por dos meses, en el primer mes se realizaron pruebas de ensayo para la utilización del servidor y el segundo mes se utilizó para conseguir los resultados: las 3780 simulaciones presentadas a continuación. Realizando varias simulaciones con todos los los parámetros fijos, únicamente cambiando la semilla que genera los números aleatorios se podría realizar un análisis de errores que debido al coste computacional que se requeriría para realizar para todos los casos presentados, no se realiza un análisis de errores en todas las propiedades y solo se calculan los errores en la regresión por efectos de tamaño para encontrar la temperatura crítica.

3.1. Energía

En la presente sección se discuten los resultados para la energía por espín del sistema.

3.1.1. Efectos de tamaño en la energía

La figura 3.2 muestra cuatro gráficas de la energía en función de la temperatura para redes de tamaño $L = 10, 20, 30$ y 60 espines de forma que se puede observar los efectos de tamaño de la red. Cada gráfica corresponde a una diferente dilución y en cada una de ellas se gráfica la energía en función de la temperatura para diferentes L . En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta la energía en constante de acoplamiento J .

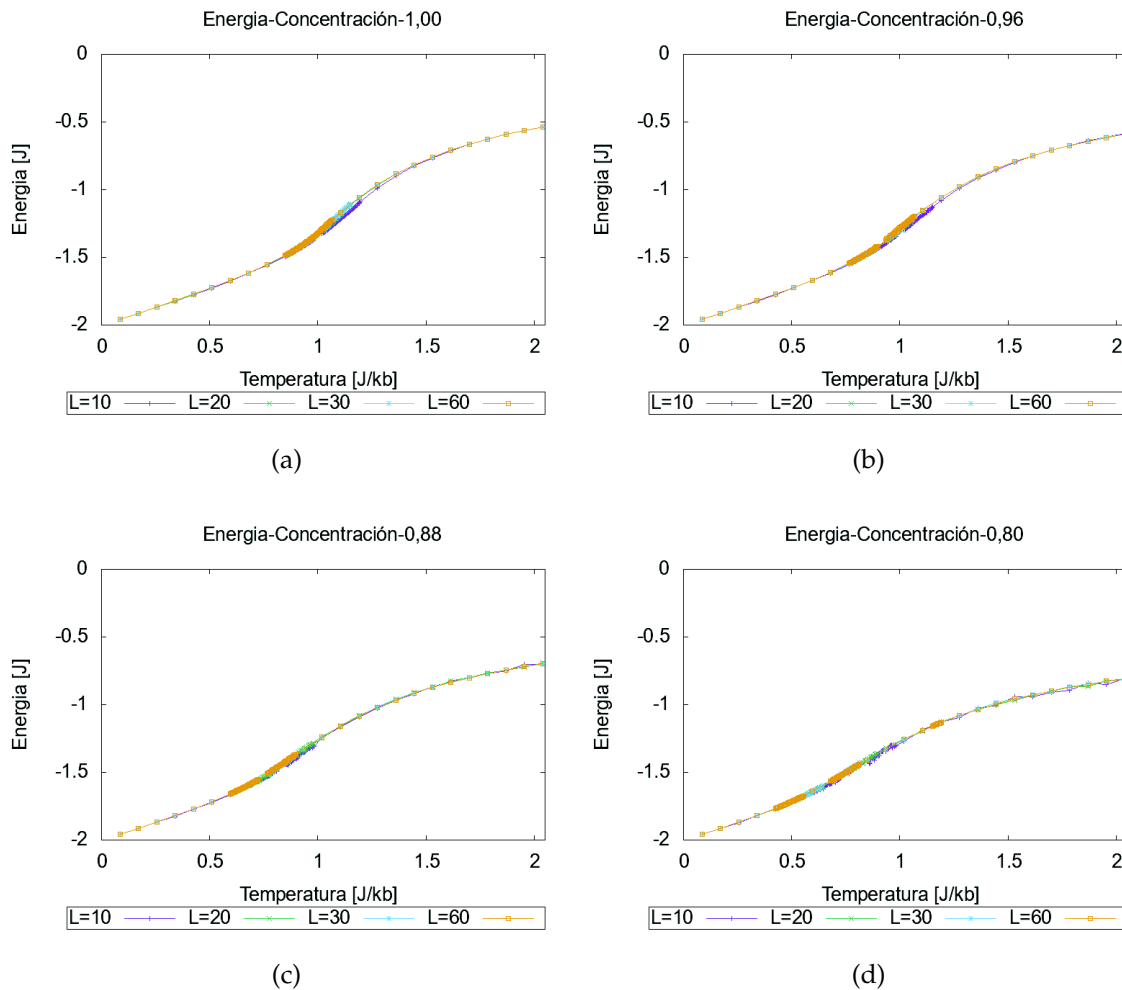


Figura 3.2: Energía por tamaño a diferentes concentraciones a) $L = 1.00$, b) $L = 0.96$, c) $L = 0.88$, d) $L = 0.80$

Se puede observar en la figura 3.2 que a medida que aumenta la temperatura aumenta la energía. Además a medida que se incrementa el tamaño de la red, los efectos en la energía son inapreciables para cada una de las concentraciones presentadas. La poca influencia del tamaño de red en la energía puede deberse a que el

fenómeno es de carácter topológico.

3.1.2. Efectos de impurezas en la energía

La figura 3.3 muestra cuatro gráficas de la energía en función de la temperatura para distintas concentraciones correspondientes a 1.00, 0.96, 0.88, 0.88 espines por unidad de área, de forma que se puede observar los efectos del porcentaje de impurezas. Las gráficas a continuación corresponden a los tamaños de red cuadrada de 10, 20, 30 y 60 espines por lado.

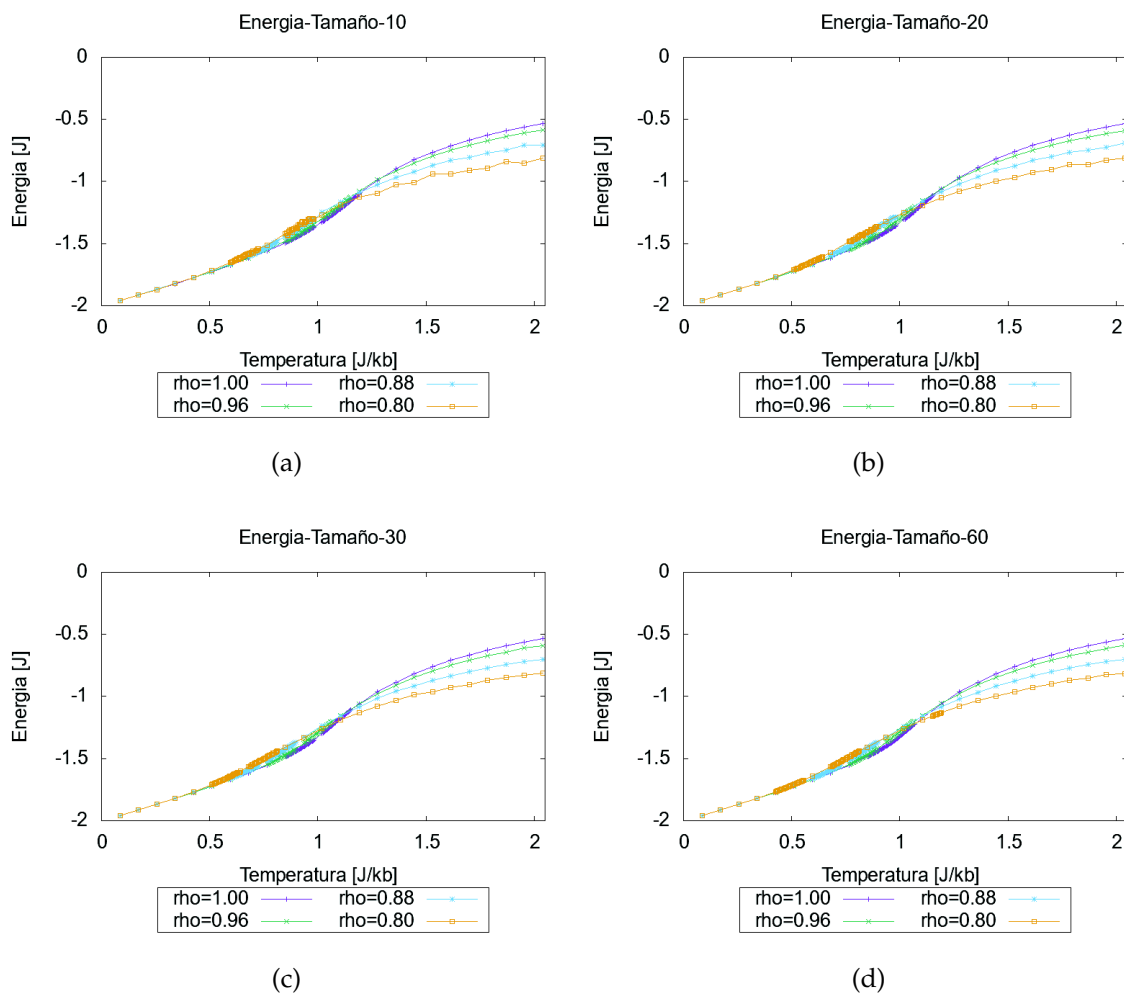


Figura 3.3: Energía en función de la temperatura por concentración a diferentes tamaños a) Tamaño 10 , b) Tamaño 20 , c) Tamaño 30 , d) Tamaño 60. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta la energía en constante de acoplamiento J .

Se puede observar en la figura 3.3, en todos los tamaños presentados, los efectos

de la dilución son más significativos con forme aumenta la temperatura. También se puede observar que a temperaturas superiores a 1,25 la energía es menor cuando se presenta una menor concentración de espines por unidad de área. Esto puede deberse a que a mayor concentración de impurezas el sistema tiene menos partículas magnéticas para almacenar energía.

3.2. Capacidad calorífica

En la presente sección se discuten los resultados de la capacidad calorífica por espín del sistema. Es importante tener en cuenta que la presencia de un pico en la capacidad calorífica nos puede indicar el punto donde la transición de fase ocurre.

3.2.1. Efectos de tamaño en la capacidad calorífica

La figura 3.4 muestra la capacidad calorífica en función de la temperatura para redes de tamaño $L = 10, 20, 30$ y 60 espines de forma que se puede observar los efectos de tamaño de la red. Cada gráfica corresponde a una diferente dilución y en cada una de ellas se gráfica la energía en función de la temperatura para diferentes L . En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas está la capacidad calorífica medida en constantes de Boltzmann k_b .

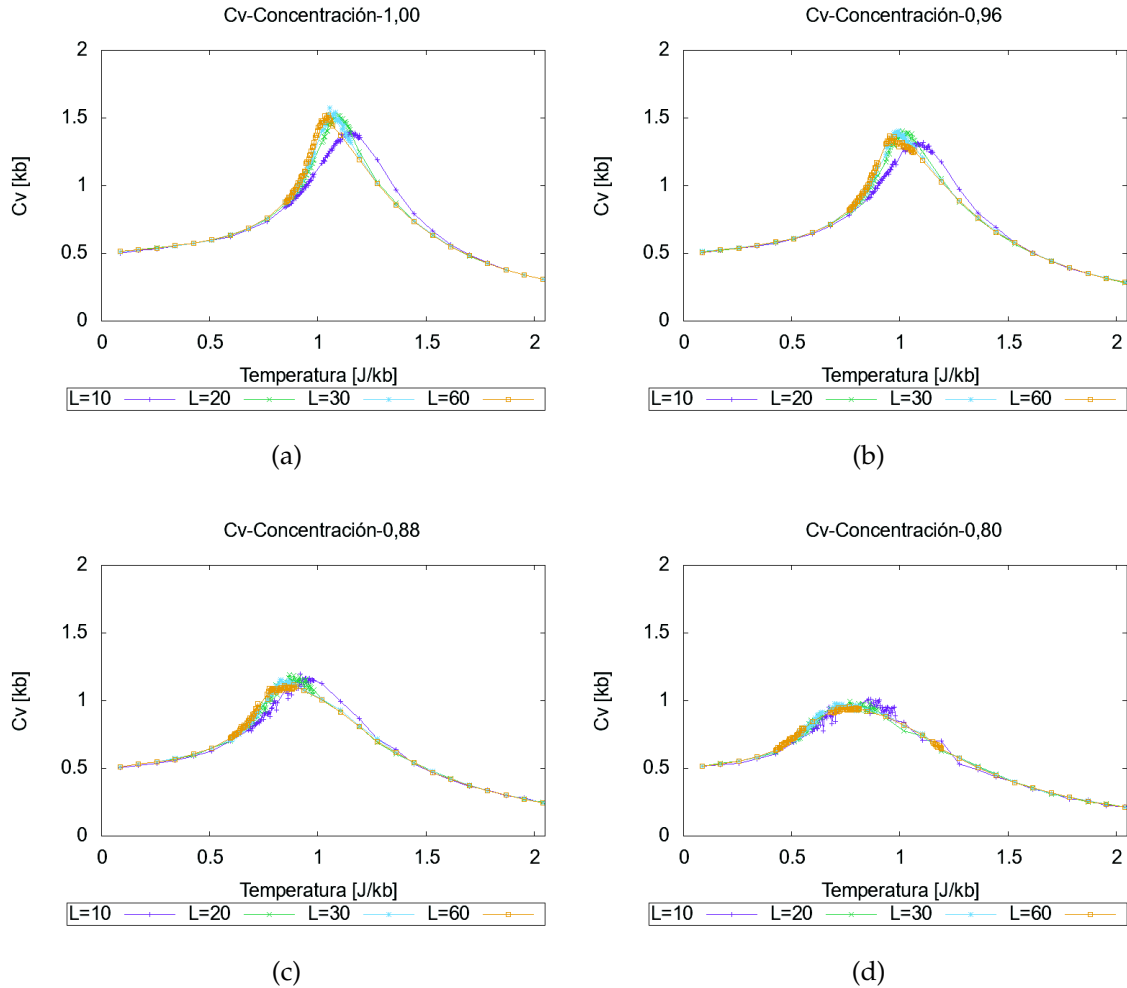


Figura 3.4: Capacidad calorífica por tamaño a diferentes concentraciones a) concentración 1.00, b) concentración 0.96, c) concentración 0.88, d) concentración 0.80

Se puede observar en la figura 3.4.a que, para una red sin impurezas, a medida que se incrementa el tamaño de red, el pico crece y tiene un corrimiento hacia temperaturas menores. También se puede observar en la figura 3.4 que los efectos de tamaño para menor concentración de espines por unidad de área parecen disminuir. Es decir, las impurezas en los rangos simulados disminuyen los efectos de tamaño en la capacidad calorífica.

3.2.2. Efectos de las impurezas en la capacidad calorífica

La figura 3.5 muestra cuatro gráficas de la capacidad calorífica en función de la temperatura para distintas concentraciones correspondientes a 1.00, 0.96, 0.88, 0.88 espines por unidad de área. De de forma que se puede observar los efectos del porcentaje de impurezas. Las gráficas a continuación corresponden a $L = 10, 20, 30$

y 60 espines por lado. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta la capacidad calorífica en constante de Boltzmann k_b .

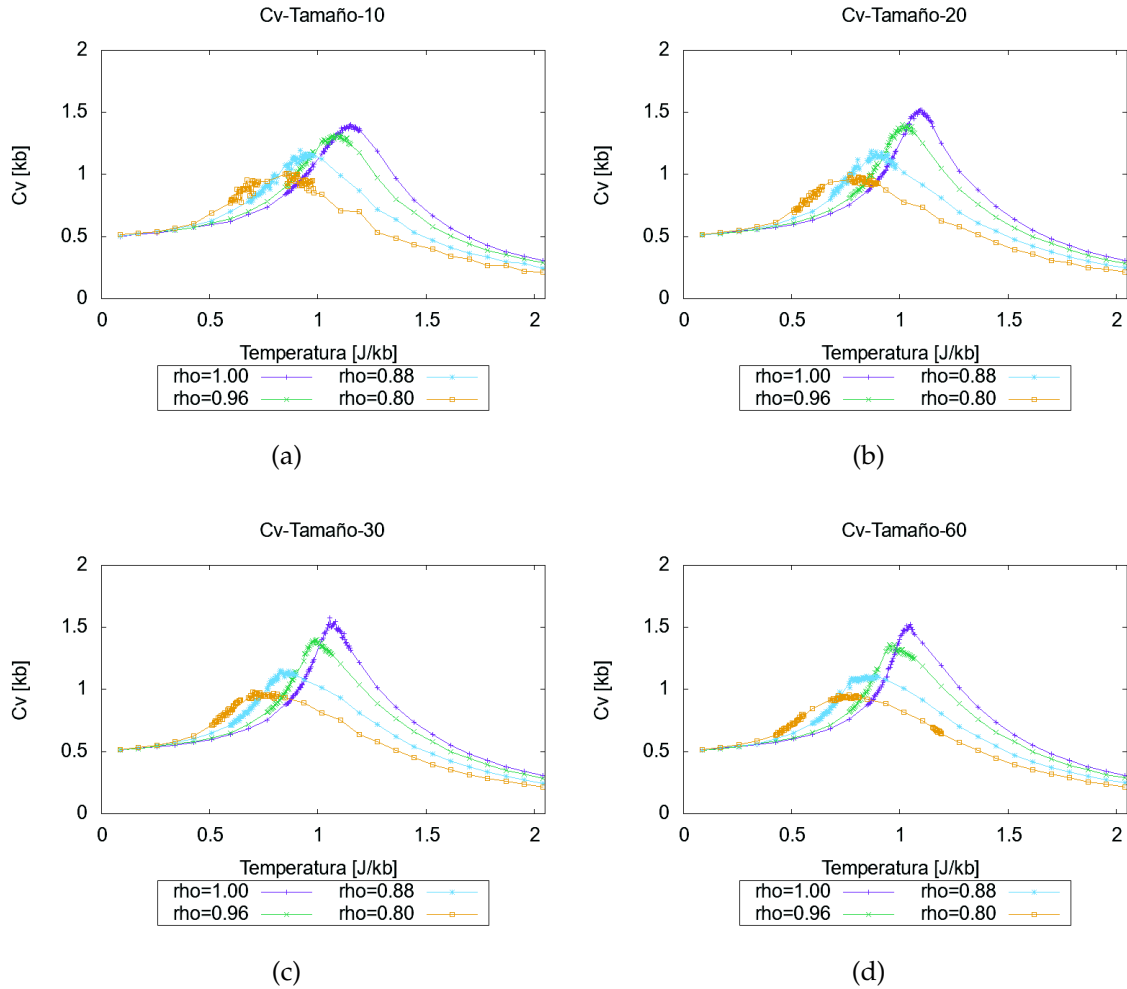


Figura 3.5: Capacidad calorífica por dilución a diferentes tamaños a) $L = 10$, b) $L = 20$, c) $L = 30$, d) $L = 60$

En la figura 3.5 se puede notar que a medida que la concentración de espines disminuye, el pico en la capacidad calorífica disminuye y tiene un corrimiento a temperaturas menores. También se puede observar que los efectos de la dilución son similares en todos los tamaños mostrados.

3.3. Magnetización

En la presente sección se discuten los resultados de la magnetización del sistema.

3.3.1. Efectos de tamaño en magnetización

La figura 3.6 muestra la magnetización de forma que se puede observar los efectos de tamaño de red. En cada gráfica están las longitudes de 10, 20, 30 y 60 espines por lado. Las gráficas a continuación son las correspondientes a las diluciones 1.00, 0.96, 0.88, 0.88 espines por unidad de área. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b . En el eje de las ordenadas está la magnetización. El valor de 1 corresponde a que todos los espines están orientados en una misma dirección.

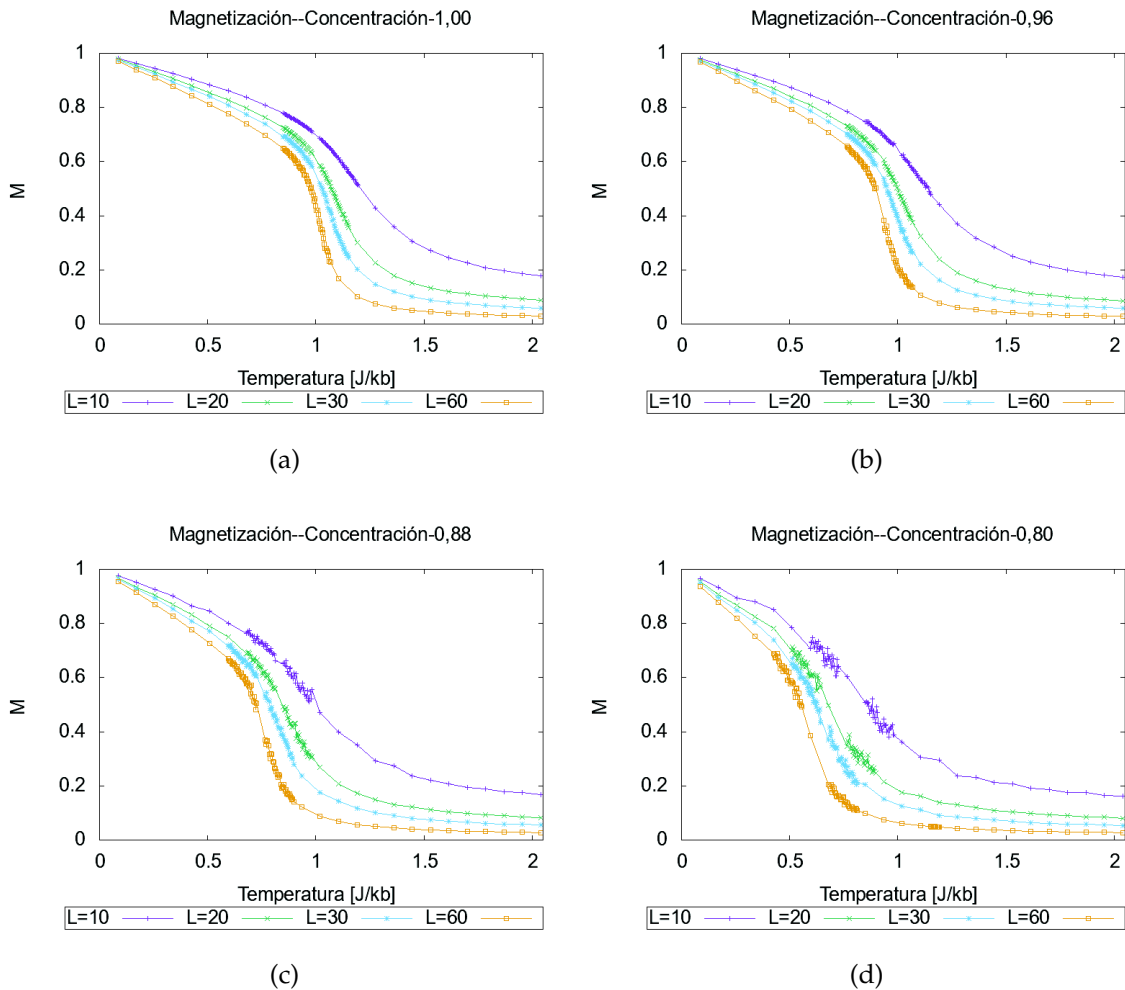


Figura 3.6: Magnetización en función de la temperatura, por tamaño a diferentes concentraciones a) concentración 1.00, b) concentración 0.96, c) concentración 0.88, d) concentración 0.80e

En la figura 3.6 se puede observar que conforme aumenta el tamaño de la red la magnetización decae a cero más rápido a menores temperaturas. Podría esperarse que a mayores tamaños de red se cumpla el teorema de Mermin-Wagner que nos

indica que la magnetización es cero a toda temperatura, ya que no existe una ruptura espontánea de simetría para sistemas de simetría continua. Cabe recalcar que una simulación siempre es una aproximación discreta de cualquier sistema continuo. En este caso la continuidad en el ángulo de rotación de los espines está limitada por la generación de números aleatorios y el número de pasos en la evolución Monte Carlo.

3.3.2. Efectos de impurezas en la magnetización

La figura 3.7 muestra cuatro gráficas de la capacidad calorífica en función de la temperatura para distintas concentraciones correspondientes a 1.00, 0.96, 0.88, 0.88 espines por unidad de área, de forma que se puede observar los efectos del porcentaje de impurezas. Las gráficas a continuación corresponden a los tamaños de red cuadrada de 10, 20, 30 y 60 espines por lado. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta la capacidad calorífica en constante de Boltzmann k_b .

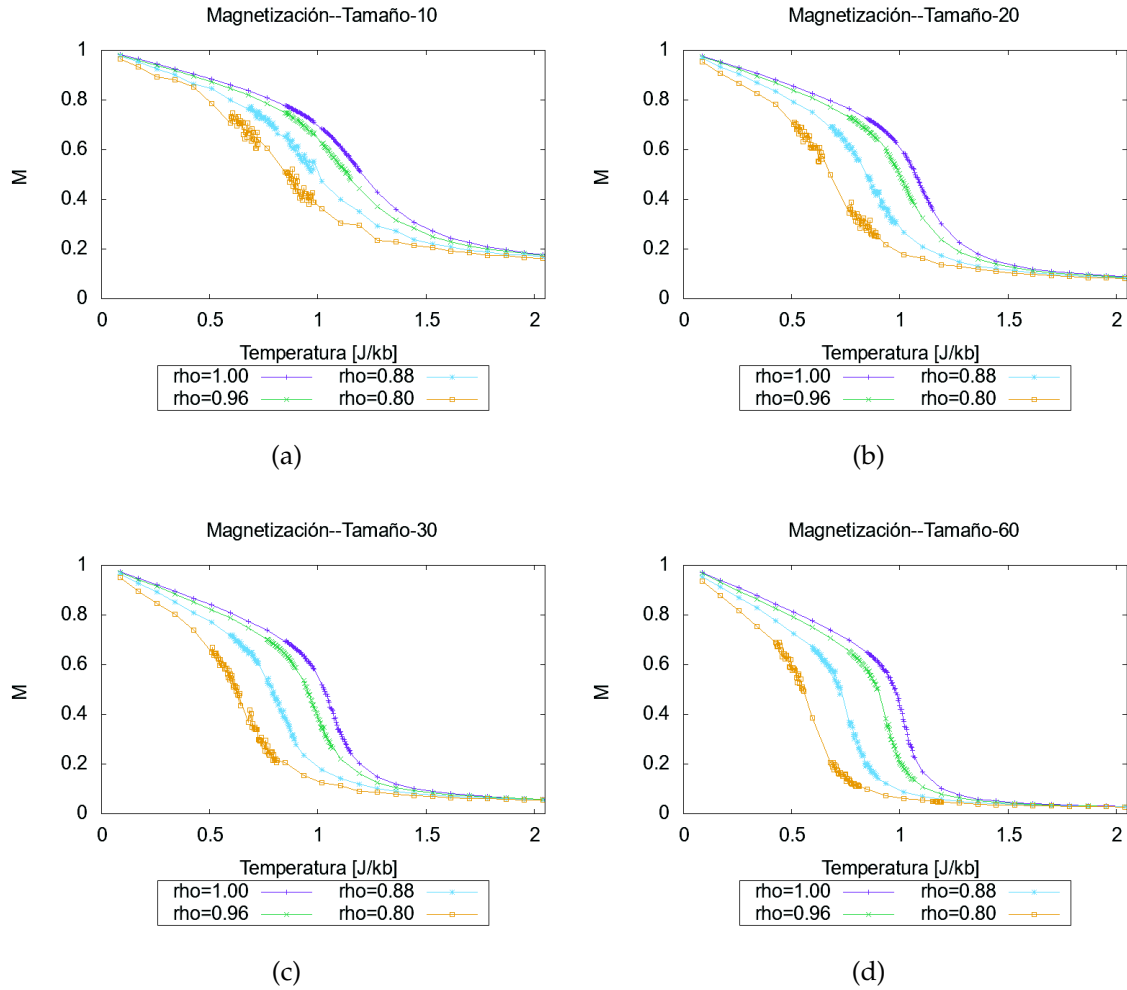


Figura 3.7: Magnetización en función de la temperatura, por dilución a diferentes tamaños a) $L = 10$, b) $L = 20$, c) $L = 30$, d) $L = 60$

En la figura 3.7 se puede observar que la magnetización decae más rápidamente conforme aumenta la concentración de impurezas. Este fenómeno se puede apreciar en todos los tamaños presentados.

3.4. Susceptibilidad magnética

En la presente sección se discuten los resultados de la susceptibilidad magnética del sistema. Es importante tener en cuenta que la presencia de un pico en la susceptibilidad magnética nos puede indicar el punto donde la transición de fase ocurre.

3.4.1. Efectos de tamaño en la susceptibilidad magnética

La figura 3.8 muestra la susceptibilidad magnética de forma que se puede observar los efectos de tamaño de red. En cada gráfica están las longitudes de 10, 20, 30 y 60 espines por lado. Las gráficas a continuación son las correspondientes a las diluciones 1.00, 0.96, 0.88, 0.88 espines por unidad de área. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta la susceptibilidad magnética.

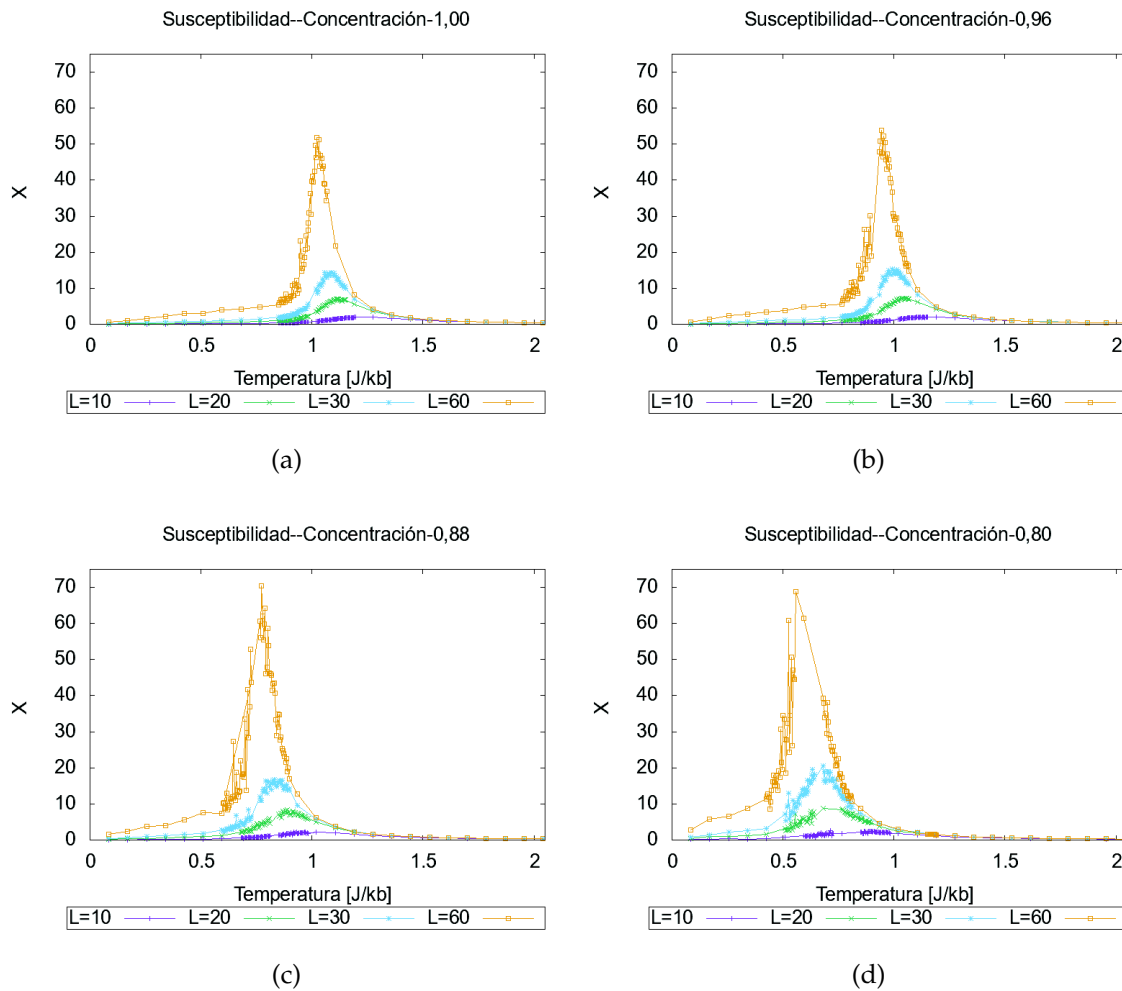


Figura 3.8: Susceptibilidad magnética en función de la temperatura, por tamaño a diferentes concentraciones a) $L = 1.00$, b) $L = 0.96$, c) $L = 0.88$, d) $L = 0.80$

En la figura 3.8 se puede observar que, a medida que el tamaño de red se incrementa, el pico de la susceptibilidad magnética se incrementa y tiene un corrimiento a temperaturas menores. Este pico nos indica que existe una transición de fase a esta temperatura. Éste fenómeno se puede observar en todas las concentraciones presentadas.

3.4.2. Efectos de la las impurezas en la susceptibilidad magnética

La figura 3.9 muestra cuatro gráficas de la susceptibilidad magnética en función de la temperatura para distintas concentraciones correspondientes a 1.00, 0.96, 0.88, 0.88 espines por unidad de área, de de forma que se puede observar los efectos del porcentaje de impurezas. Las gráficas a continuación corresponden a los tamaños de red cuadrada de 10, 20, 30 y 60 espines por lado. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta la susceptibilidad magnética .

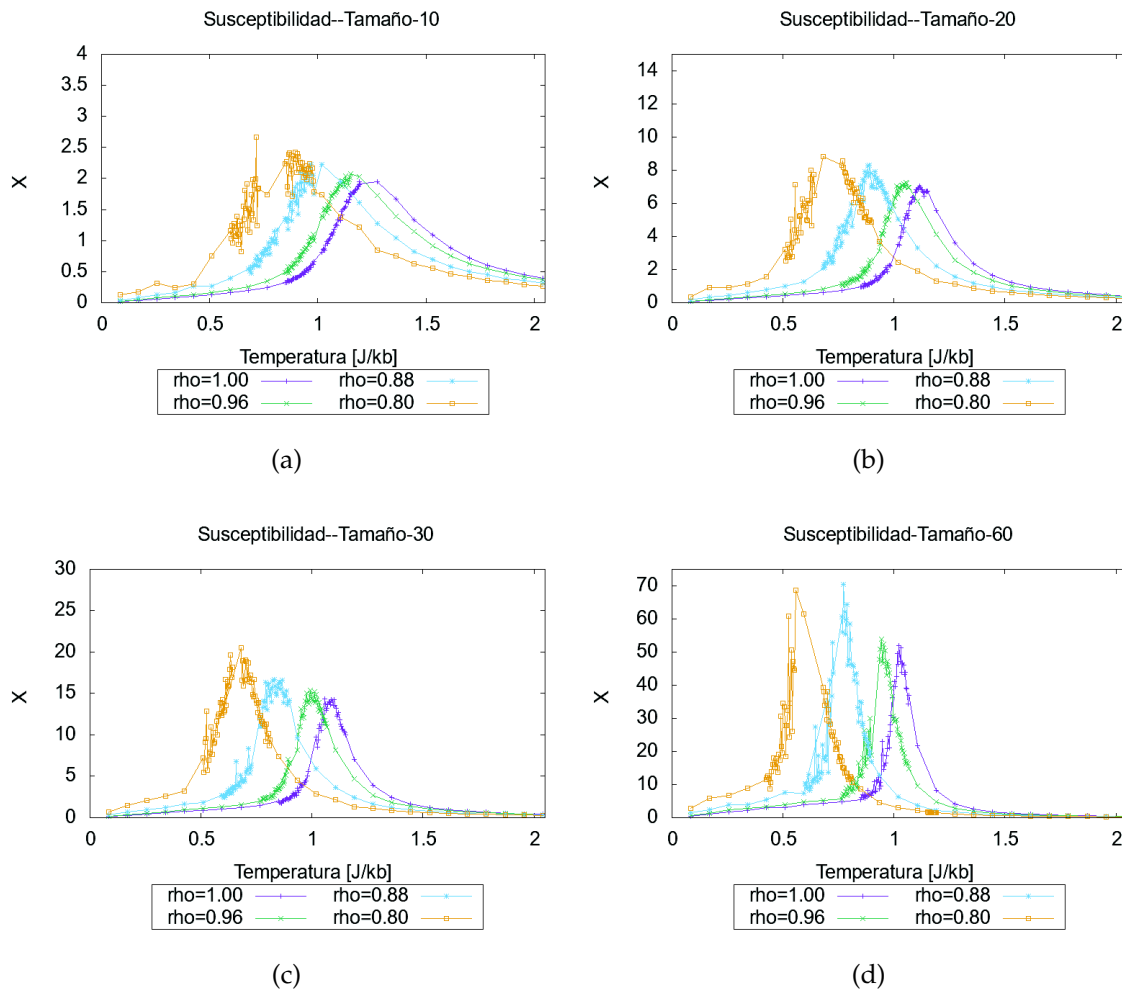


Figura 3.9: Susceptibilidad magnética en función de la temperatura, por dilución a diferentes tamaños a) $L = 10$, b) $L = 20$, c) $L = 30$, d) $L = 60$

En la figura 3.9 se puede observar que, a medida que la concentración de espines por unidad de área disminuye, el pico de la susceptibilidad magnética se incrementa y tiene un corrimiento a temperaturas menores. Éste fenómeno se puede observar en todas los tamaños presentados.

3.5. Rigidez de giro

En la presente sección se discuten los resultados de la rigidez de giro magnética del sistema. Es importante tener en cuenta que, según la bibliografía citada en el marco teórico, la temperatura a la cual la rigidez de giro se interseca con la recta que parte del origen con pendiente $\frac{2}{\pi}$ indica la temperatura de transición de fase.

3.5.1. Efectos de tamaño en la rigidez de giro

La figura 3.10 muestra la rigidez de giro de forma que se puede observar los efectos de tamaño de red. En cada gráfica están las longitudes de 10, 20, 30 y 60 espines por lado. Las gráficas a continuación son las correspondientes a las diluciones 1.00, 0.96, 0.88, 0.88 espines por unidad de área. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta la rigidez de giro. Además en cada una de las gráficas esta la recta de pendiente $2/\pi$ en color amarillo.

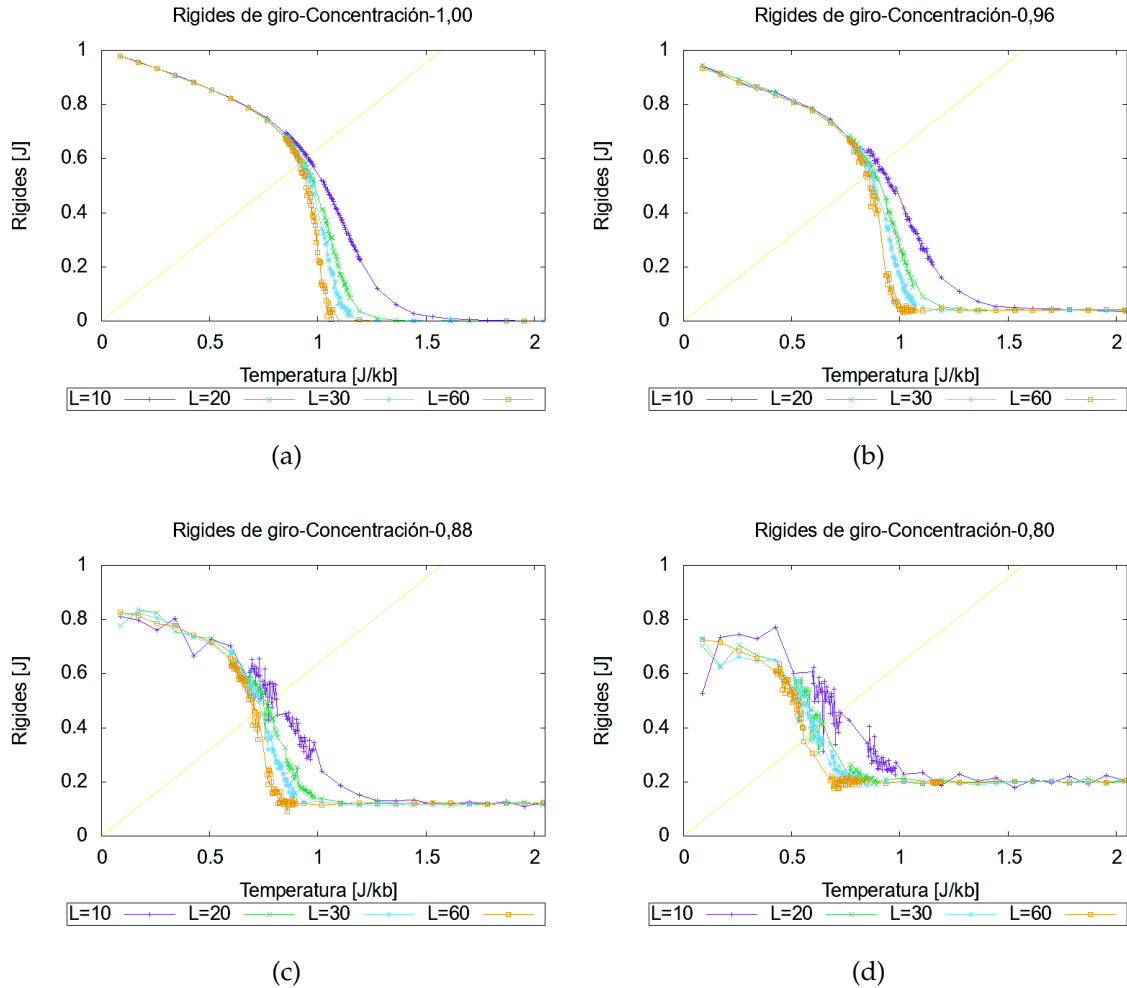


Figura 3.10: Susceptibilidad magnética en función de la temperatura, por tamaño a diferentes concentraciones a) concentración 1.00, b) concentración 0.96, c) concentración 0.88, d) concentración 0.80

En la figura 3.10 se puede observar que conforme aumenta el tamaño la rigidez de giro decae más rápido. Por otro lado también se puede observar que al comparar en la figura 3.10.a con 3.10.c a mayor dilución la rigidez de giro presenta más ruido.

3.5.2. Efectos de las impurezas en la rigidez de giro por dilución

La figura 3.11 muestra cuatro gráficas de la rigidez de giro en función de la temperatura para distintas concentraciones correspondientes a 1.00, 0.96, 0.88, 0.88 espines por unidad de área. De forma que se puede observar los efectos del porcentaje de impurezas. Las gráficas a continuación corresponden a los tamaños de red cuadrada de 10, 20, 30 y 60 espines por lado. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann

k_b , en el eje de las ordenadas esta la rigidez de giro .

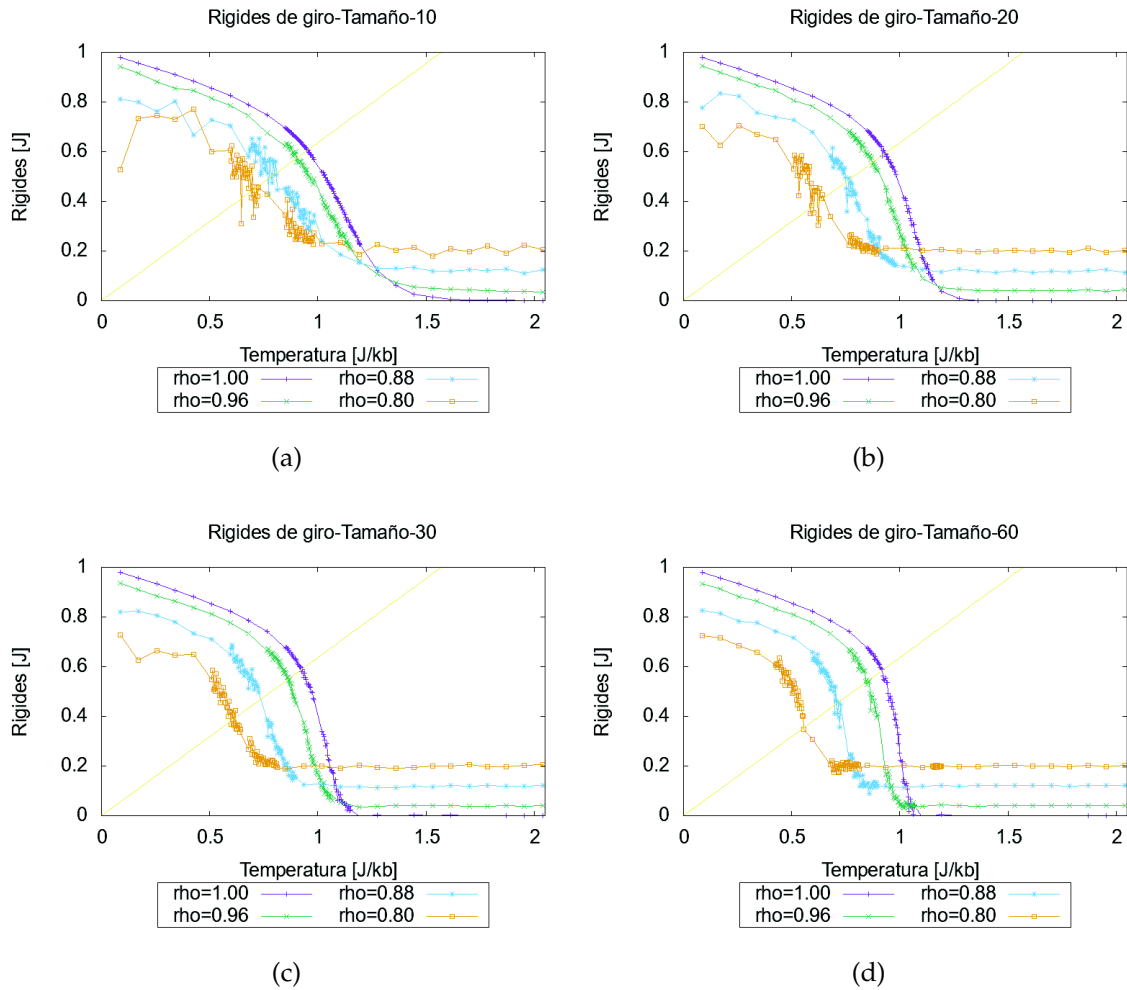


Figura 3.11: Rigidez de giro en función de la temperatura, por dilución a diferentes tamaños a) $L = 10$, b) $L = 20$, c) $L = 30$, d) $L = 60$

En la figura 3.11 se puede observar que la rigidez de giro decae más rápido cuando disminuye la concentración de espines por unidad de área, pese a esto la rigidez de giro parece tener un comportamiento constante que no decae a cero cuando se tiene mayor número de impurezas. La rigidez de giro a altas temperaturas es mayor conforme aumenta el número de impurezas en las concentraciones presentadas.

También se puede observar que mientras aumenta el número de impurezas, aumenta el ruido en la simulación. Esto puede deberse; a que, para cada red generada se le otorga diferentes temperatura con la misma cantidad de impurezas aunque colocadas en diferentes posiciones. Se recomienda que en estudios posteriores se analice la relación entre la posición de las impurezas y la rigidez de giro.

3.6. Vórtices

En esta sección se discute el número de vórtices formados en la red. Es importante recordar que los vórtices aparecen sobre la temperatura de transición. Y bajo la temperatura de transición no se espera encontrar vórtices.

La figura 3.12 muestra el número de vórtices por unidad de superficie de forma que se puede observar los efectos de tamaño. En cada gráfica están las longitudes de 15, 20, 25 y 30 espines por lado. Las gráficas a continuación son las correspondientes a las diluciones 1.00, 0.96 espines por unidad de área. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta el número de vórtices por unidad de área.

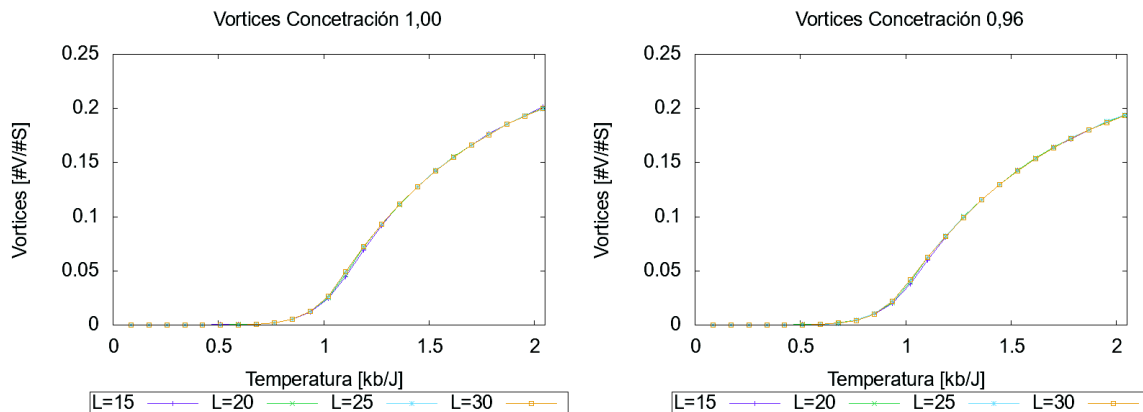


Figura 3.12: Rigidez de giro en función de la temperatura, a concertación fija, a diferentes tamaños a) $L = 10$, b) $L = 20$, c) $L = 30$, d) $L = 60$

En la figura 3.12 se puede observar que con los parámetros mostrados no hay efectos de tamaño en el número de vórtices por unidad de superficie. Se recomienda realizar en estudios posteriores más simulaciones para corroborar esta observación de forma estadística.

La figura 3.13 se muestra el número de vórtices por unidad de área en función de la temperatura para distintas concentraciones correspondientes a 1.00, 0.98, 0.96 espines por unidad de área, de forma que se puede observar los efectos del porcentaje de impurezas. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta el número de vórtices por unidad de superficie .

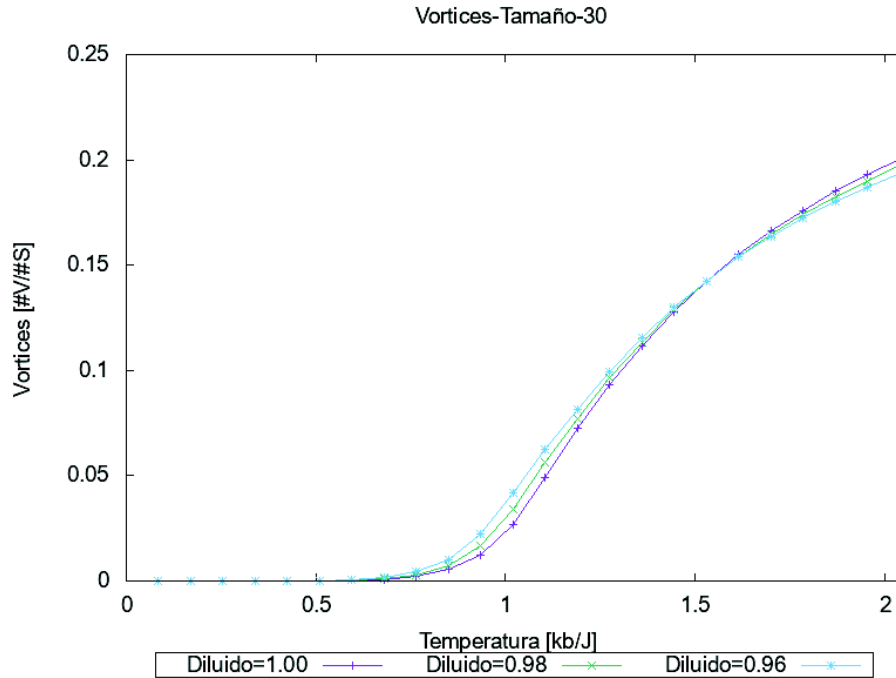


Figura 3.13: Efectos de las impurezas en el número de vórtices

En la figura 3.13 se puede observar que en el rango de temperatura de $0,51k_b/J$ a $1,445k_b/J$ el número de vórtices es mayor cuando el número de impurezas aumenta. En la temperatura de $0,595k_b/J$ en las tres concentraciones presentadas el número de vórtices por unidad de área es el mismo. Finalmente en el rango de $1,445k_b/J$ a $2,04k_b/J$ el número de vórtices es menor cuando aumenta el número de impurezas. Este comportamiento puede ocasionarse porque las impurezas disminuyen la correlación del sistema y permite la formación de más vórtices a baja temperatura, pero al rededor de la temperatura de $1,445k_b/J$ las impurezas disminuyen el número de espines que pueden formar los vórtices. Por este motivo, a partir de esta temperatura, se forman más vórtices con menor número de impurezas. Se recomienda realizar más simulaciones con mayor precisión al rededor de la temperatura de $1,445k_b/J$ y diferente número de impurezas para corroborar esta observación de forma estadística.

En la figura 3.14 se muestra la configuración final en una red cuadrada de 30 posiciones por lado sin impurezas a una temperatura de $0,765k_b/J$.

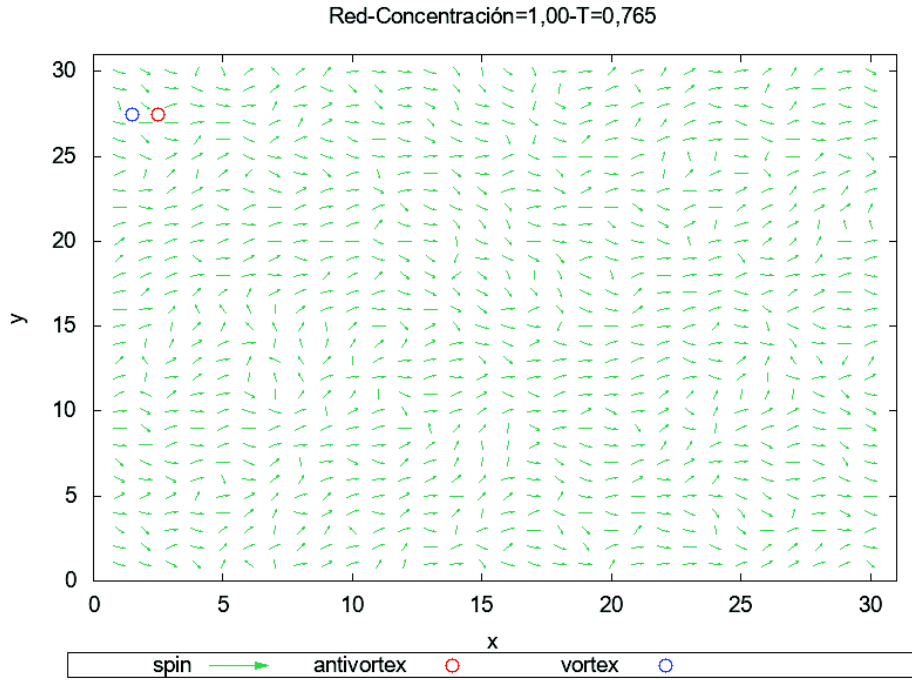


Figura 3.14: Red temperatura de $0,765k_b/J$ sin impurezas

La figura 3.14 muestra una red con una temperatura bajo la temperatura de transición. Se observa que no presenta formación de vórtices.

En la figura 3.15 muestra la configuración final en una red cuadrada de 30 posiciones por lado sin impurezas a una temperatura de $1,02k_b/J$.

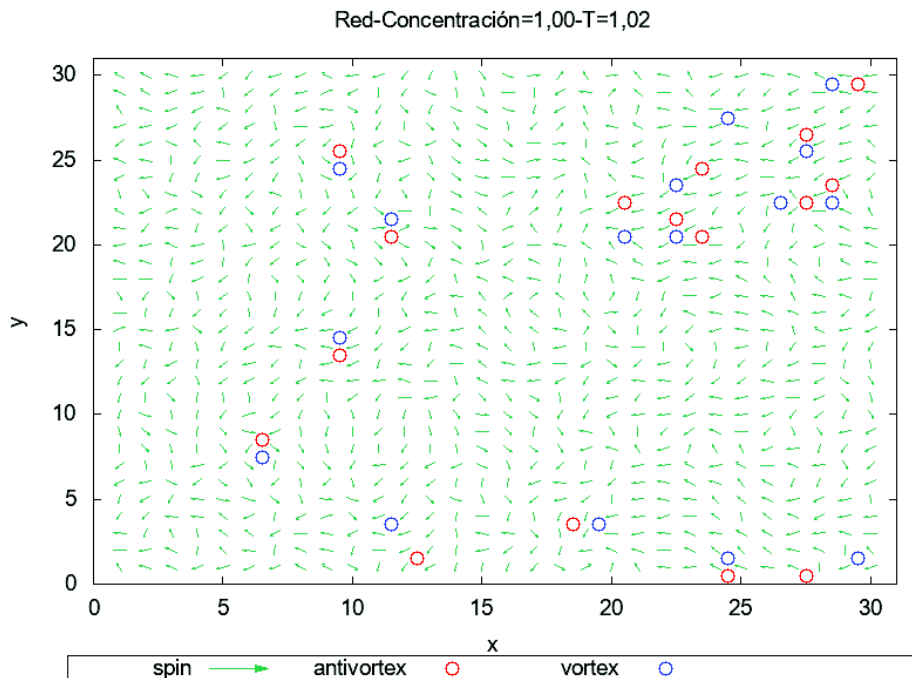


Figura 3.15: Red temperatura de $1,02k_b/J$ sin impurezas

La figura 3.15 muestra una red con una temperatura por encima de la temperatura de transición, se observa la formación de vórtices. También se puede observar que los vórtices están muy cerca a los anti-vórtices en pares.

La figura 3.16 muestra la configuración final en una red cuadrada de 30 posiciones por lado con una concentración de 0,96 espines por unidad de área a una temperatura de $0,765k_b/J$.

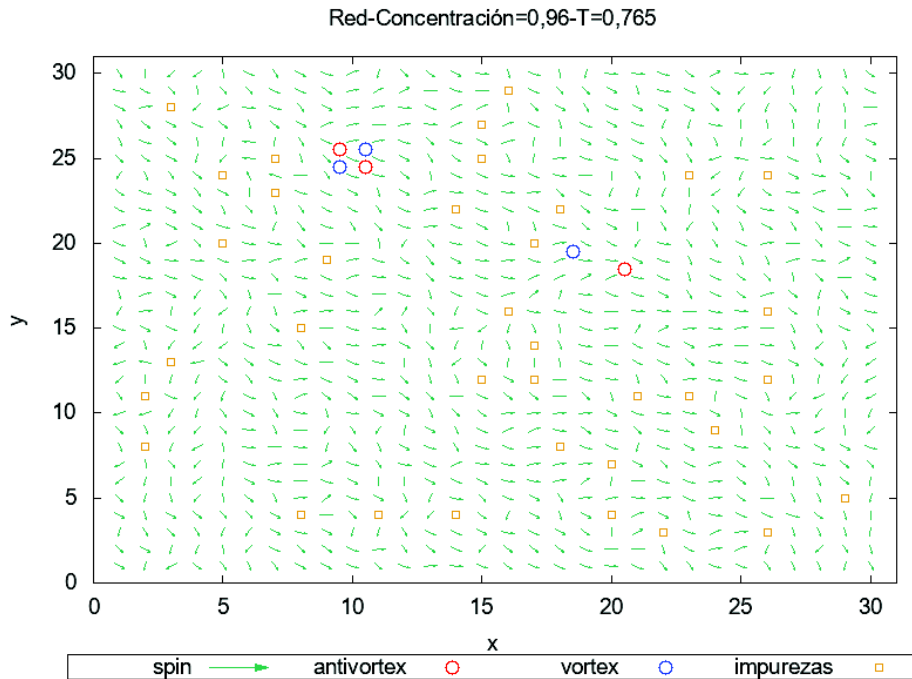


Figura 3.16: Red temperatura de $0,765k_b/J$ concentración 0,96

En la figura 3.16 se muestra una red con una temperatura por debajo de la temperatura de transición, se observa que presenta tan solo 3 pares de vórtices anti-vórtices.

En la figura 3.17 se muestra la configuración final en una red cuadrada de 30 posiciones por lado con una concentración de 0,96 espines por unidad de área a una temperatura de $1,02k_b/J$.

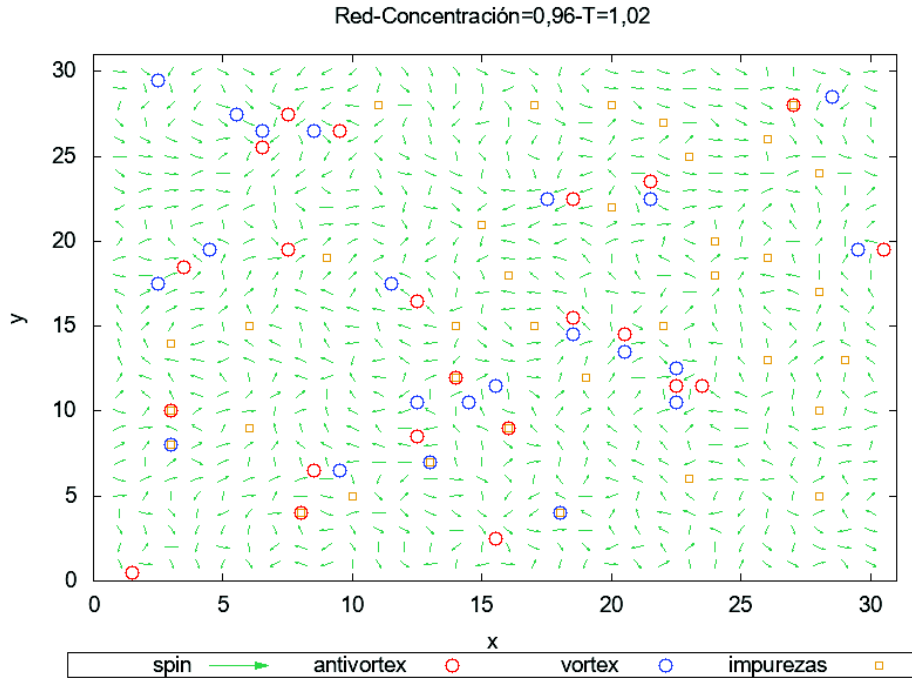


Figura 3.17: Red temperatura de $1,02k_b/J$ concentración 0,96

La figura 3.17 muestra una red con una temperatura sobre la temperatura de transición, se observa la formación de pares de vórtice-antivórtice. También se puede observar que los vórtices están muy cerca a los anti-vórtices en pares. Además se puede notar que existen vórtices y anti-vórtices que se forman en las impurezas.

3.7. Función de correlación

En esta sección se realiza un estudio de la función de correlación y los exponentes críticos de una regresión exponencial y con ley de potencia. El estudio de esta función es muy importante porque la transición T_{KT} es una transición donde el sistema pasa de ser de corto alcance, es decir, ser ajusta solo a una ley exponencial, a ser de cuasi-largo alcance, es decir se puede ajustar a una ley de potencia.

Recordemos ley exponencial $e^{-r/\zeta}$, cuyo exponente crítico ζ se denomina longitud de correlación y ley de potencia $r^{-\eta}$, cuyo exponente crítico es η .

Para poder realizar las regresiones presentadas en esta sección se utiliza el método descrito e implementadas en el código del anexo C.2.

3.7.1. Función de correlación sin impurezas

En la figura 3.18 se muestra la función de correlación de una red cuadrada de 30×30 para dos temperaturas con sus respectivas regresiones. Para temperaturas $T = 0,51J/k_b$ menor a la temperatura de transición, en color celeste están los puntos simulados, en color negro se encuentra la regresión por ley de potencia y en color rojo la regresión por ley exponencial. Para temperaturas $T = 1,70J/k_b$ mayor que la temperatura crítica se muestra en color verde están los puntos, en color azul el ajuste por ley de potencia y en violeta el ajuste por ley exponencial.

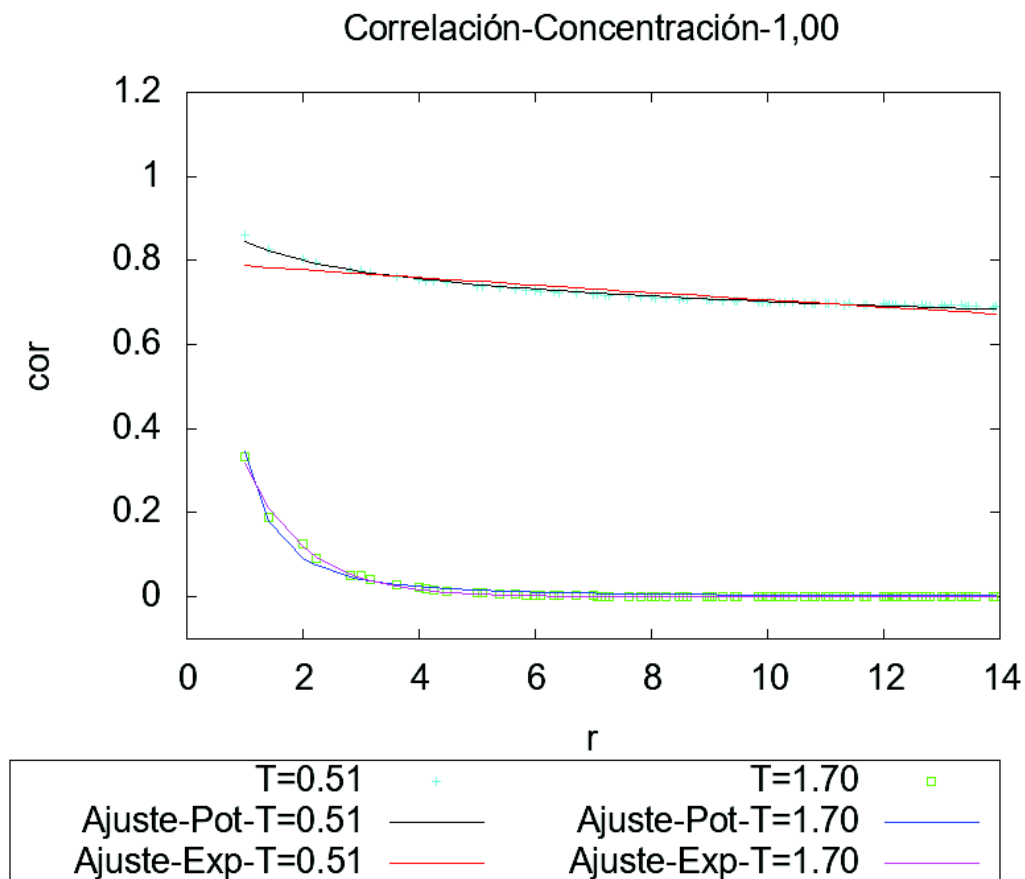


Figura 3.18: Función de correlación sin impurezas

En la figura 3.18 se puede observar que la línea roja está alejada de los puntos celestes, es decir, la ley exponencial no se ajusta bien a bajas temperaturas. También se puede observar que a altas temperaturas se ajusta mejor la ley exponencial.

En la figura 3.19 se muestra los exponentes críticos en función de la temperatura para una red sin impurezas. Se presentan los exponentes críticos para redes cuadradas de 10, 20, 30 y 60 puntos de red por lado. La figura 3.19.a presenta el exponente

crítico para una ley exponencial y la figura 3.19.b presenta el exponente crítico para a una ley de potencia. Cabe recalcar que no se muestran valores para la longitud de correlación bajo la temperatura crítica en la ecuación (1.21) solo se define para temperaturas mayores a la temperatura crítica.

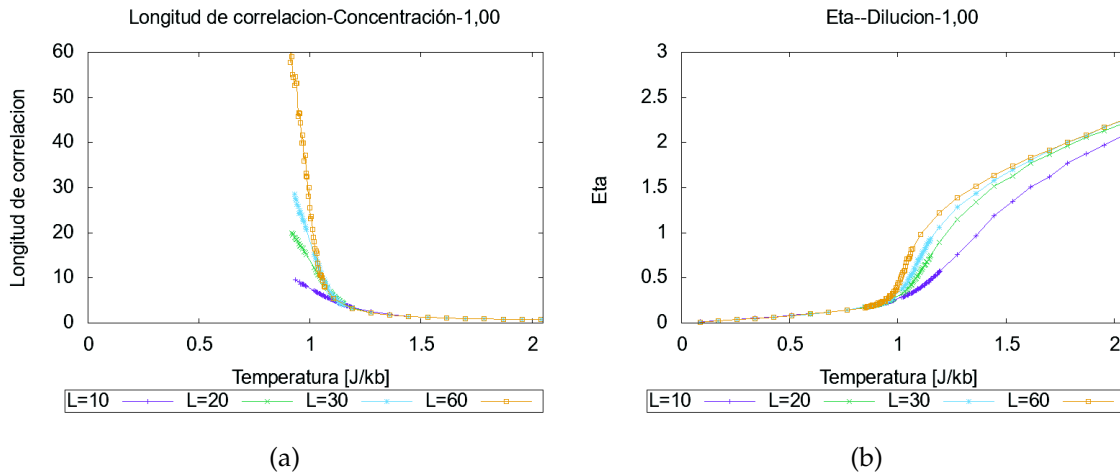


Figura 3.19: Exponentes críticos sin impurezas a) Ley exponencial b) Ley de potencia

En la figura 3.19.a se puede observar que la longitud de correlación tiende a divergir tal como se espera en la ecuación (1.21). En la figura 3.19.b se puede observar que bajo la temperatura crítica los efectos de tamaño no son apreciables en los rangos de tamaño mostrado, y para temperaturas sobre la temperatura crítica el exponente crítico eta crece conforme el tamaño de la red aumenta.

3.7.2. Función de correlación concentración 0,96

En la figura 3.18 se muestra la función de correlación de una red cuadrada de 30×30 para dos temperaturas con sus respectivas regresiones. Para temperaturas $T = 0,51J/k_b$ menor a la temperatura de transición, en color celeste están los puntos simulados, en color negro se encuentra la regresión por ley de potencia y en color rojo la regresión por ley exponencial. Para temperaturas $T = 1,70J/k_b$ mayor que la temperatura crítica se muestra en color verde están los puntos, en color azul el ajuste por ley de potencia y en violeta el ajuste por ley exponencial.

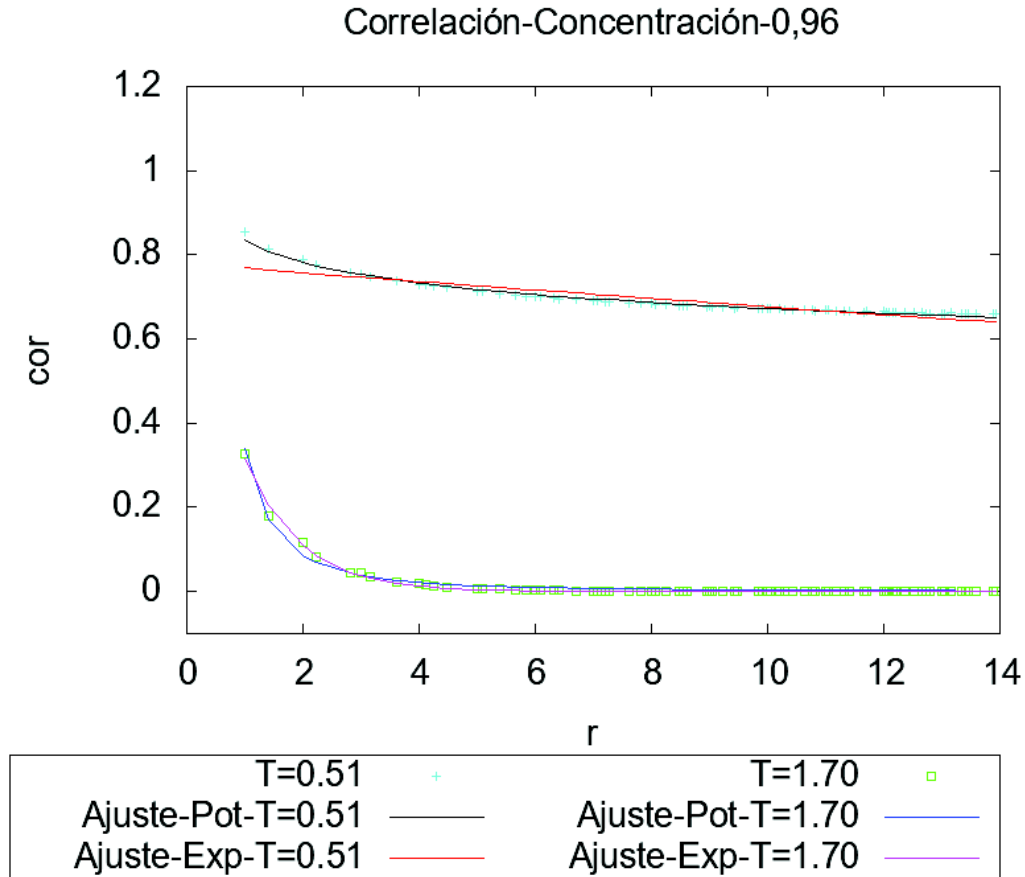


Figura 3.20: Función de correlación concentración 0,96

En la figura 3.20.a se puede observar que la línea roja está alejada de los puntos celestes es decir, la ley exponencial no se ajusta bien a altas bajas temperaturas. También se puede observar que a altas temperaturas se ajusta mejor la ley exponencial.

En la figura 3.21 se muestra los exponentes críticos en función de la temperatura para una red con 0,04% de impurezas. Se presentan los exponentes críticos para redes cuadradas de 10, 20, 30 y 60 puntos de red por lado. La figura 3.21.a presenta el exponente crítico para una ley exponencial y la figura 3.21.b presenta el exponente crítico para a una ley de potencia. Cabe recalcar que no se muestran valores para la longitud de correlación bajo la temperatura crítica en la ecuación (1.21) solo se define para temperaturas mayores a la temperatura crítica.

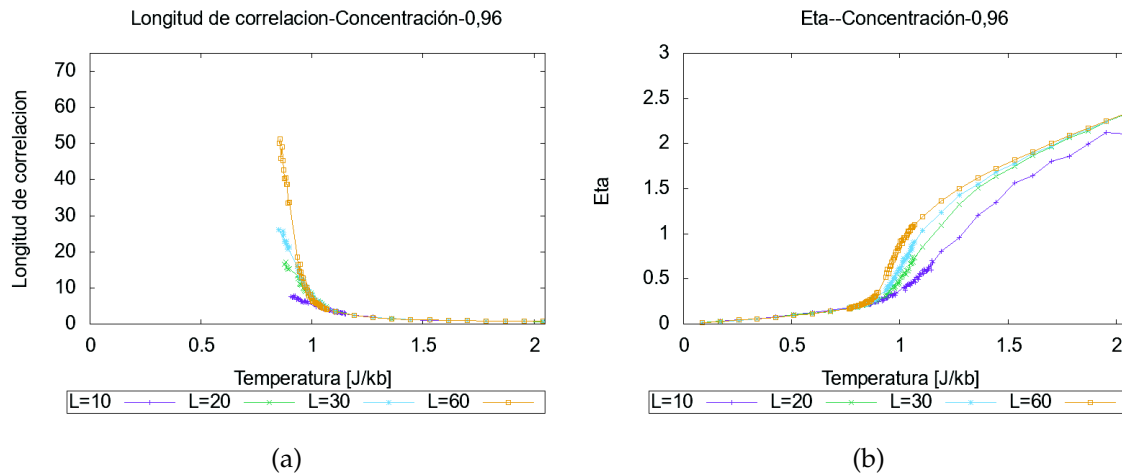


Figura 3.21: En la presente figura se puede notar

En la figura 3.21.a se puede observar que la longitud de correlación tiende a divergir tal como se espera en la ecuación (1.21). En la figura 3.21.b se puede observar que bajo la temperatura crítica los efectos de tamaño no son apreciables en los rangos de tamaño mostrado, y para temperaturas sobre la temperatura crítica el exponente crítico η crece conforme el tamaño de la red aumenta.

3.7.3. Función de correlación concentración 0,88

En la figura 3.22 se muestra la función de correlación de una red cuadrada de 30×30 para dos temperaturas con sus respectivas regresiones. Para temperaturas $T = 0,51J/k_b$ menor a la temperatura de transición, en color celeste están los puntos simulados, en color negro se encuentra la regresión por ley de potencia y en color rojo la regresión por ley exponencial. Para temperaturas $T = 1,70J/k_b$ mayor que la temperatura crítica se muestra en color verde están los puntos, en color azul el ajuste por ley de potencia y en violeta el ajuste por ley exponencial.

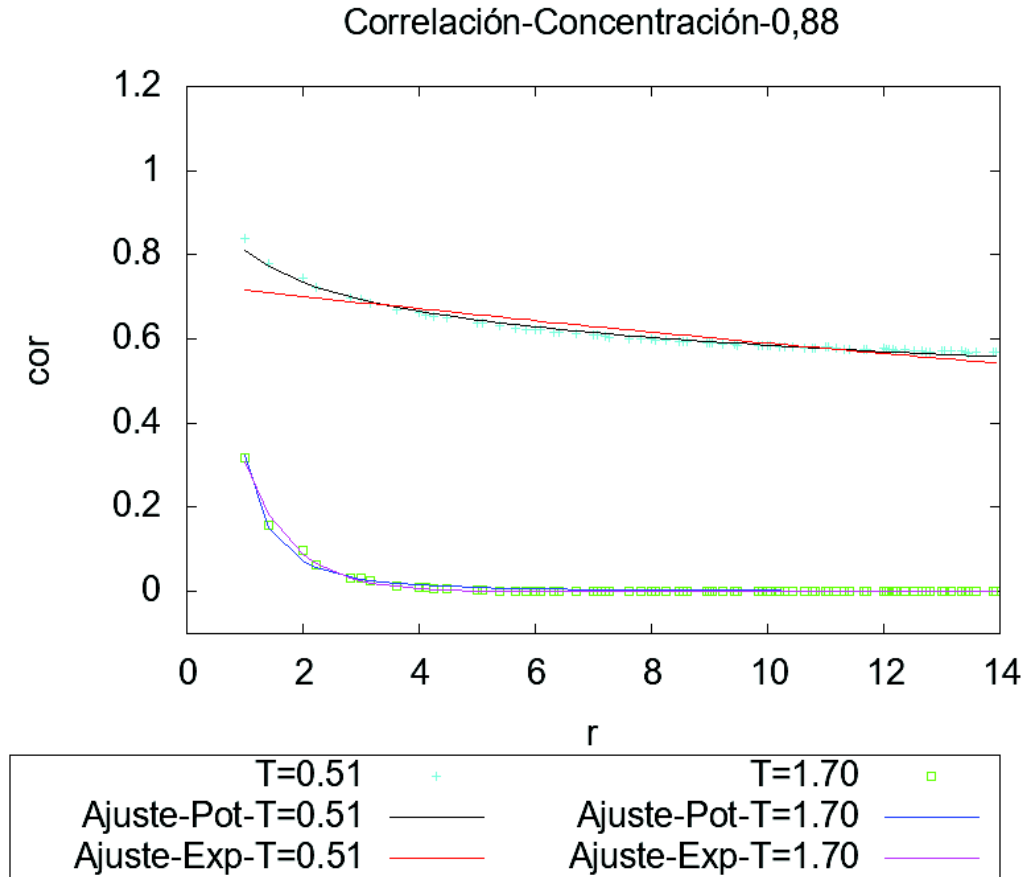


Figura 3.22: Función de correlación concentración 0,88

En la figura 3.22 se puede observar que la línea roja está alejada de los puntos celestes es decir, la ley exponencial no se ajusta bien a altas bajas temperaturas. También se puede observar que a altas temperaturas se ajusta mejor la ley exponencial.

En la figura 3.23 se muestra los exponentes críticos en función de la temperatura para una red con 0,12% de impurezas. Se presentan los exponentes críticos para redes cuadradas de 10, 20, 30 y 60 puntos de red por lado. La figura 3.23.a presenta el exponente crítico para una ley exponencial y la figura 3.23.b presenta el exponente crítico para a una ley de potencia. Cabe recalcar que no se muestran valores para la longitud de correlación bajo la temperatura crítica en la ecuación (1.21) solo se define para temperaturas mayores a la temperatura crítica.

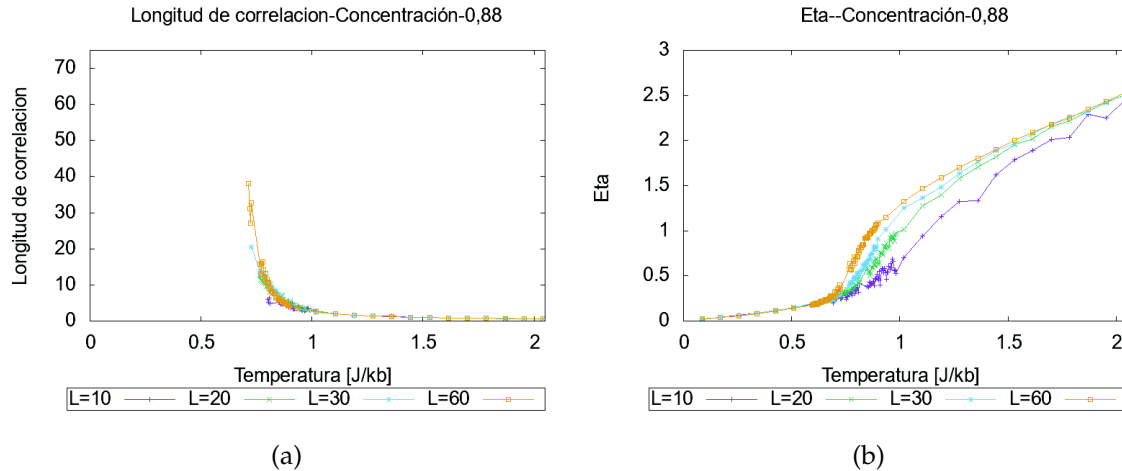


Figura 3.23: En la presente figura se puede notar

En la figura 3.23.a se puede observar que la longitud de correlación tiende a divergir tal como se espera en la ecuación (1.21). En la figura 3.23.b se puede observar que bajo la temperatura crítica los efectos de tamaño no son apreciables en los rangos de tamaño mostrado, y para temperaturas sobre la temperatura crítica el exponente crítico η crece conforme el tamaño de la red aumenta.

3.7.4. Función de correlación concentración 0,80

En la figura 3.24 se muestra la función de correlación de una red cuadrada de 30×30 para dos temperaturas con sus respectivas regresiones. Para temperaturas $T = 0,51J/k_b$ menor a la temperatura de transición, en color celeste están los puntos simulados, en color negro se encuentra la regresión por ley de potencia y en color rojo la regresión por ley exponencial. Para temperaturas $T = 1,70J/k_b$ mayor que la temperatura crítica se muestra en color verde están los puntos, en color azul el ajuste por ley de potencia y en violeta el ajuste por ley exponencial.

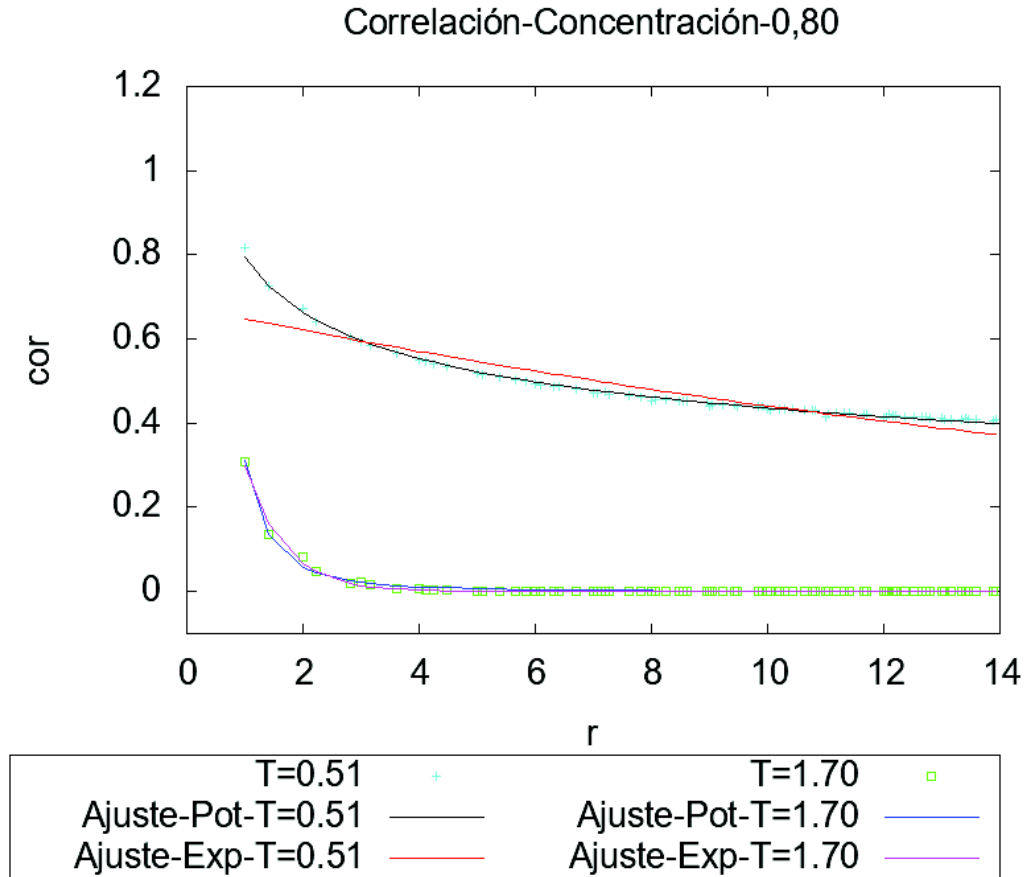


Figura 3.24: Función de correlación concentración 0,80

En la figura 3.24 se puede observar que la línea roja está alejada de los puntos celestes es decir, la ley exponencial no se ajusta bien a altas bajas temperaturas. También se puede observar que a altas temperaturas se ajusta mejor la ley exponencial.

En la figura 3.25 se muestra los exponentes críticos en función de la temperatura para una red con 0,20% de impurezas. Se presentan los exponentes críticos para redes cuadradas de 10, 20, 30 y 60 puntos de red por lado. La figura 3.25.a presenta el exponente crítico para una ley exponencial y la figura 3.25.b presenta el exponente crítico para a una ley de potencia. Cabe recalcar que no se muestran valores para la longitud de correlación bajo la temperatura crítica en la ecuación (1.21) solo se define para temperaturas mayores a la temperatura crítica.

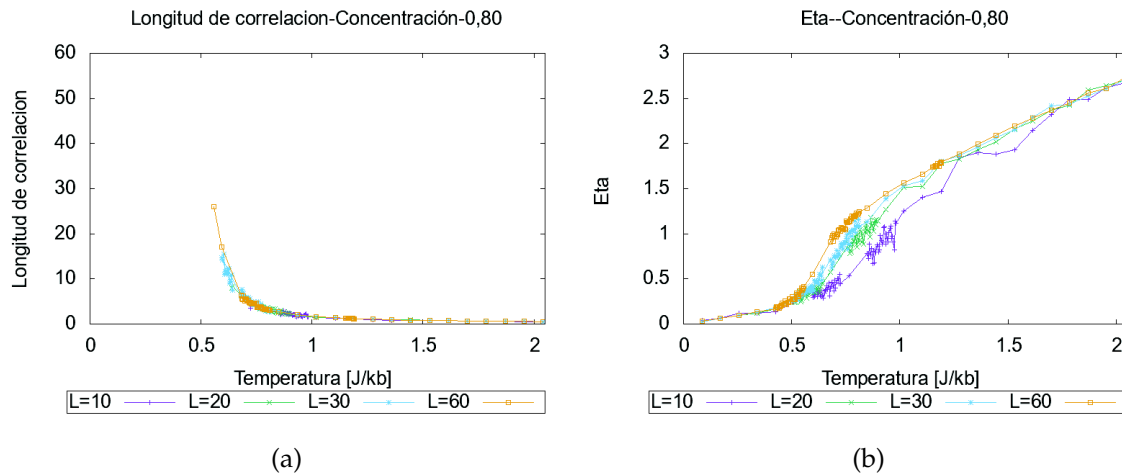


Figura 3.25: En la presente figura se puede notar

En la figura 3.25.a se puede observar que la longitud de correlación tiende a divergir tal como se espera en la ecuación (1.21). En la figura 3.25.b se puede observar que bajo la temperatura crítica los efectos de tamaño no son apreciables en los rangos de tamaño mostrado, y para temperaturas sobre la temperatura crítica el exponente crítico η crece conforme el tamaño de la red aumenta.

3.7.5. Efectos de las impurezas en los exponentes críticos

En la presente sección se estudia los efectos de tamaño en los exponentes críticos tanto para el ajuste por ley exponencial como para ley de potencia.

3.7.5.1. Efectos de las impurezas en longitud de correlación

La figura 3.26 muestra cuatro gráficas de la longitud de correlación en función de la temperatura para distintas concentraciones correspondientes a 1.00, 0.96, 0.88, 0.88 espines por unidad de área. De de forma que se puede observar los efectos del porcentaje de impurezas. Las gráficas a continuación corresponden a los tamaños de red cuadrada de 10, 20, 30 y 60 espines por lado. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta la longitud de correlación.

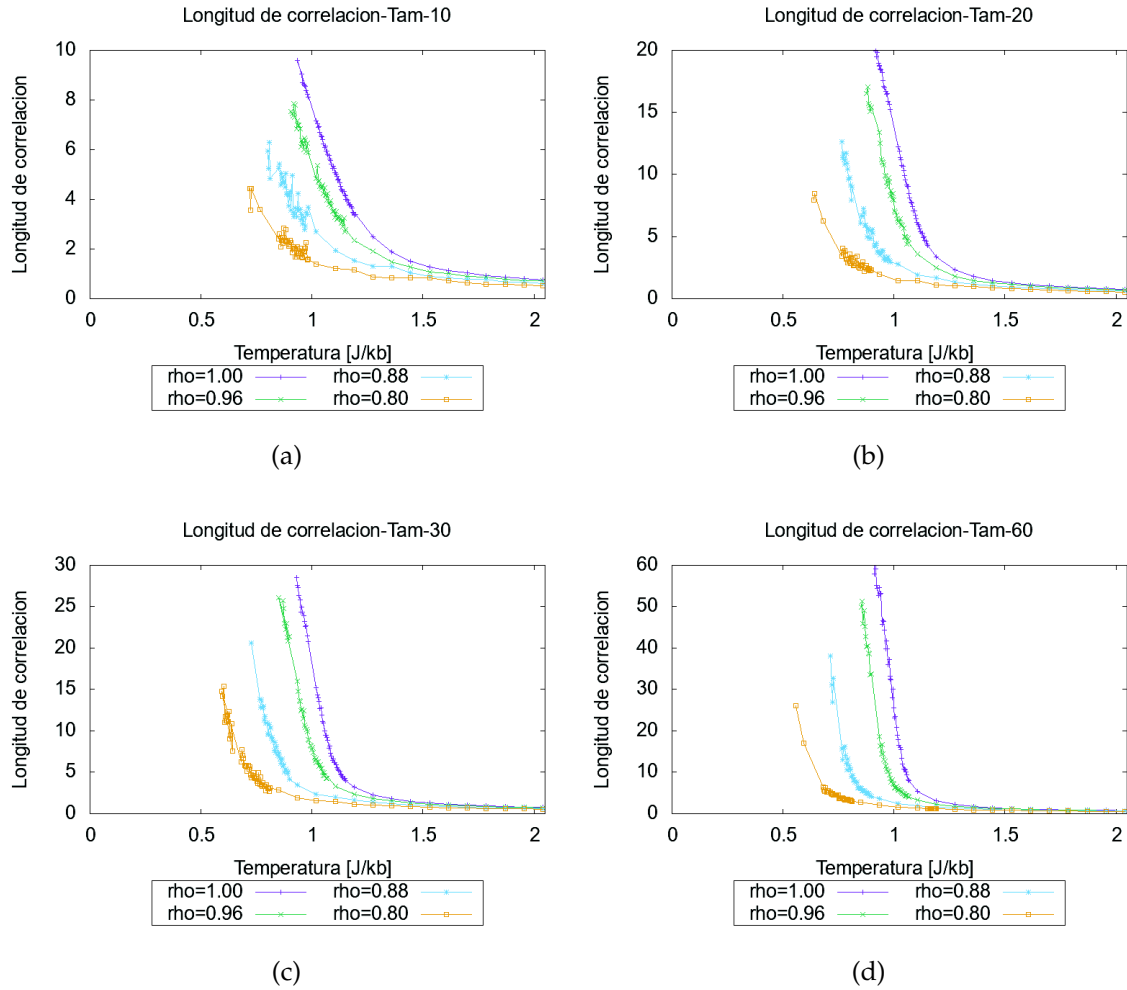


Figura 3.26: Efectos de las impurezas en en longitud de correlación a) Tamaño 10 , b) Tamaño 20 , c) Tamaño 30 , d) Tamaño 60

En la figura 3.26 se puede observar que la longitud de correlación disminuye conforme disminuye la concentración de espines en todos los tamaños presentados.

3.7.5.2. Efectos impurezas en el exponente Eta

La figura 3.27 muestra cuatro gráficas de la longitud de correlación en función de la temperatura para distintas concentraciones correspondientes a 1.00, 0.96, 0.88, 0.88 espines por unidad de área, de forma que se puede observar los efectos del porcentaje de impurezas. Las gráficas a continuación corresponden a los tamaños de red cuadrada de 10, 20, 30 y 60 espines por lado. En el eje de las abscisas se encuentra la temperatura medida en constantes de acoplamiento J por constante de Boltzmann k_b , en el eje de las ordenadas esta el exponente η .

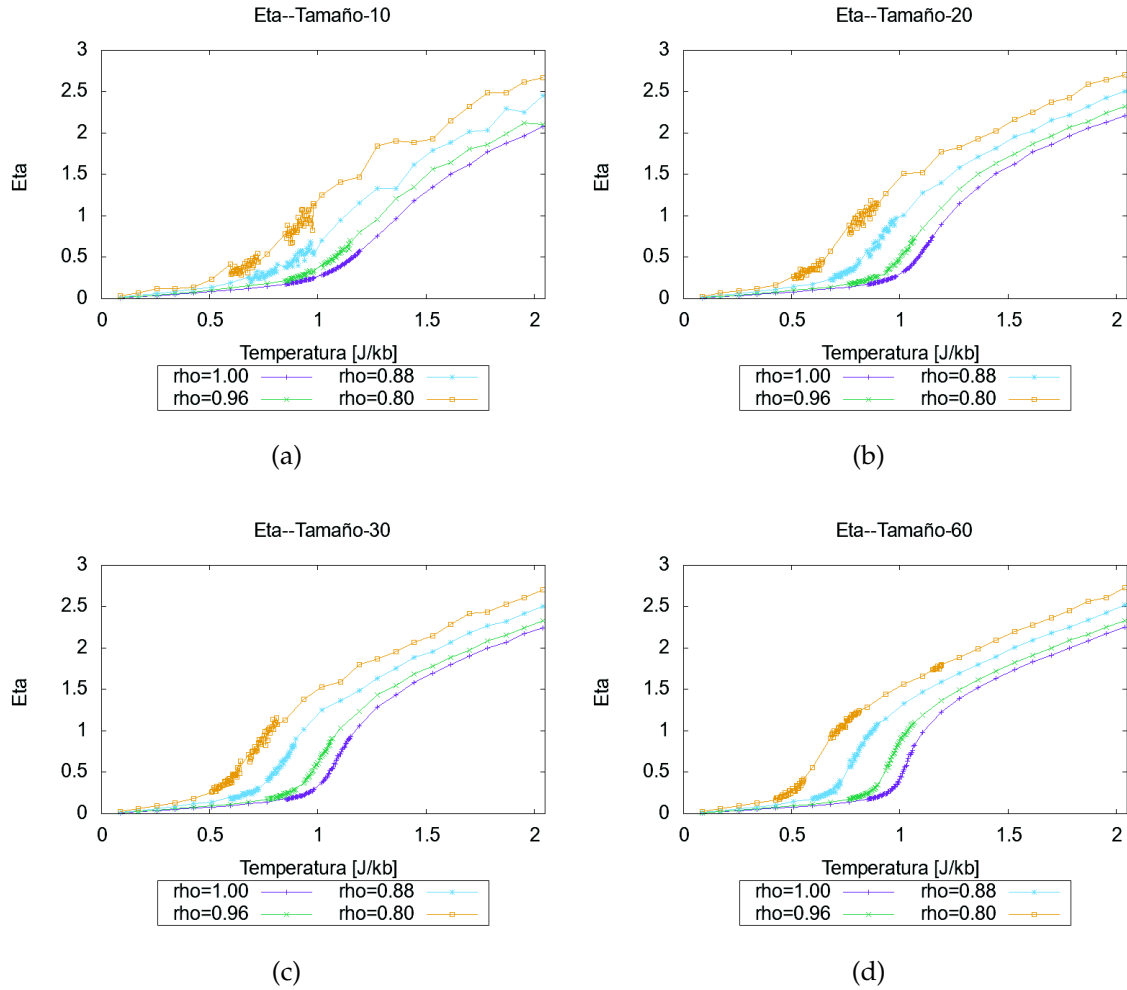


Figura 3.27: En la presente figura se puede notar

En la figura 3.27 se puede observar que el exponente η es mayor con forme disminuye la concentración de espines. Esto comprueba que las impurezas disminuyen la correlación del sistema ya que la correlación decae mas fuertemente conforme aumenta el exponente η .

3.8. Temperatura crítica de transición

En esta sección se analiza la temperatura critica de transición. Como se ha discutido en secciones anteriores se puede podría utilizar el pico de la capacidad calorífica, el pico de la susceptibilidad magnética, el valor en donde la longitud de correlación diverge, el valor en donde el exponente η es $1/4$, la intersección entre la rigidez de giro y la recta que parte del origen con pendiente $2/\pi$, incluso se podría utilizar el número de vórtices ya que todas esta propiedades pueden caracterizar la

transición de fase aún que por efecto de tamaño la temperatura no coincide entre las diferentes propiedades. Entre todas estas propiedades se han escogido únicamente dos métodos que sugiere la bibliografía. El primer método consiste en utilizar el pico en la capacidad calorífica, y el segundo método en utilizar la intersección de la rigidez de giro con la recta que parte del origen con pendiente $2/\pi$.

3.8.1. Efectos de tamaño en la temperatura crítica de transición

En la figura 3.28 muestra cuatro gráficas de la temperatura crítica de transición en función del tamaño de red. Los puntos con morado corresponden al método de la capacidad calorífica. Los puntos en celeste corresponden al método de la rigidez de giro y las líneas continuas son las regresiones utilizando la ecuación (2.24). Para realizar estas regresiones se utilizó el comando «fit» en gnuplot. Este comando utiliza el método Marquardt-Levenberg no lineal de mínimos cuadrados. Las gráficas a continuación corresponden a las concentraciones 1.00, 0.96, 0.88, 0.88 espines por unidad de área. En el eje de las abscisas se encuentra la longitud del lado de la red medida en parámetros de red, en el eje de las ordenadas encuentra la temperatura crítica de transición dependiente del tamaño medida en constantes de acoplamiento J por constante de Boltzmann k_b .

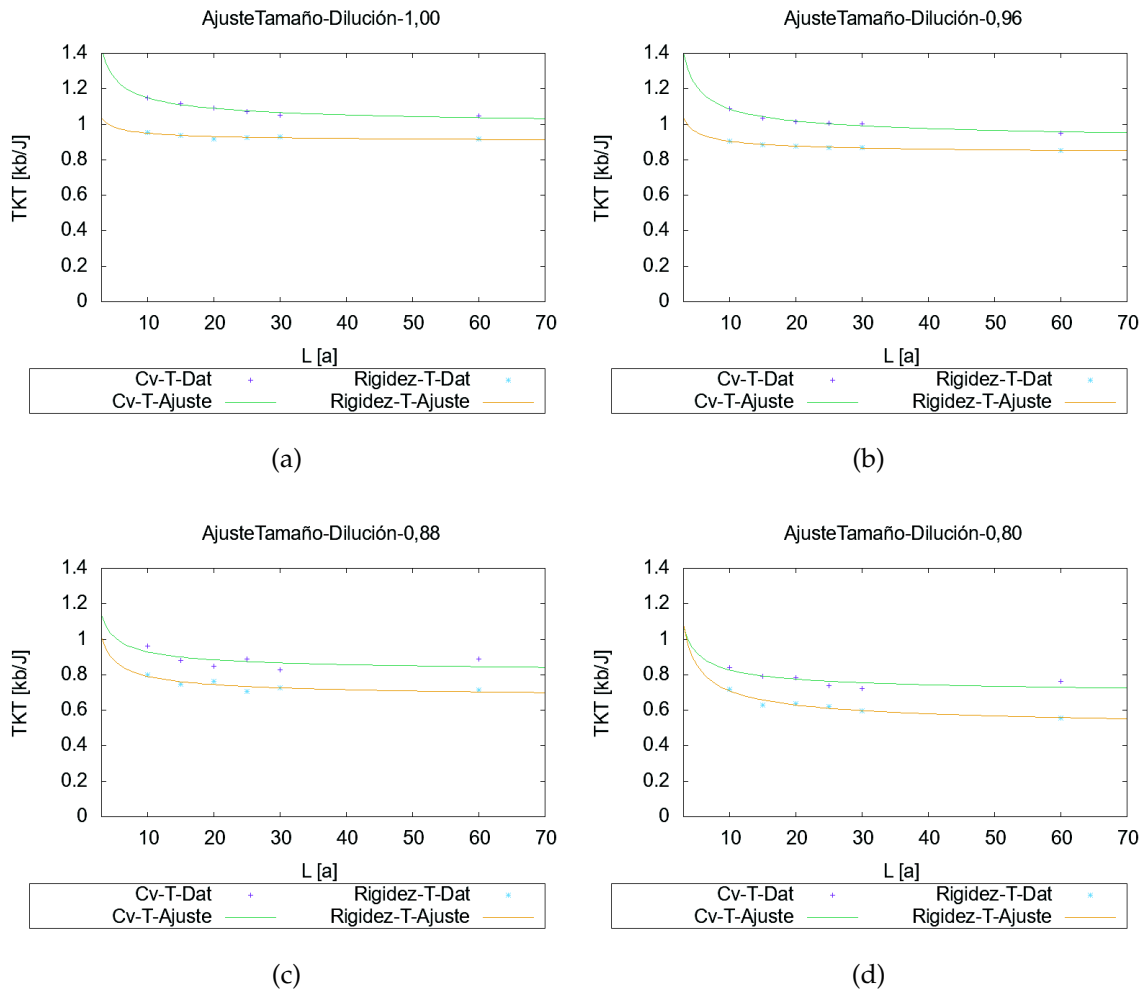


Figura 3.28: Efectos de tamaño en la temperatura crítica de transición, por tamaño a diferentes concentraciones a) concentración 1.00, b) concentración 0.96, c) concentración 0.88, d) concentración 0.80

En la figura 3.28 se puede observar que los ajustes realizados tienen menos puntos que no coinciden con las líneas del ajuste para altas concentraciones de espines. Esto se ve posteriormente reflejado en las barras de error de la figura 3.30. También se puede observar que con los tamaños de red utilizados no se ha logrado que la temperatura crítica por los dos métodos converja a una misma temperatura de transición en el infinito.

3.8.2. Efectos de la concentración de espines en la temperatura de transición

En la figura 3.28 muestra cuatro gráficas de la temperatura crítica de transición en función de la contracción de espines. Los puntos con morado corresponden a el

método de la capacidad calorífica. Los puntos en celeste corresponden al método de la rigidez de giro y las líneas continuas son las regresiones utilizando la ecuación (2.24). Para realizar estas regresiones se utilizó el comando "fit" en gnuplot. Las gráficas a continuación corresponden a redes cuadradas de 10, 20, 30 y 60 puntos de red por lado. En el eje de las abscisas se encuentra la concentración de espines medida en espines por unidad de área, en el eje de las ordenadas encuentra la temperatura crítica de transición dependiente del tamaño medida en constantes de acoplamiento J por constante de Boltzmann k_b .

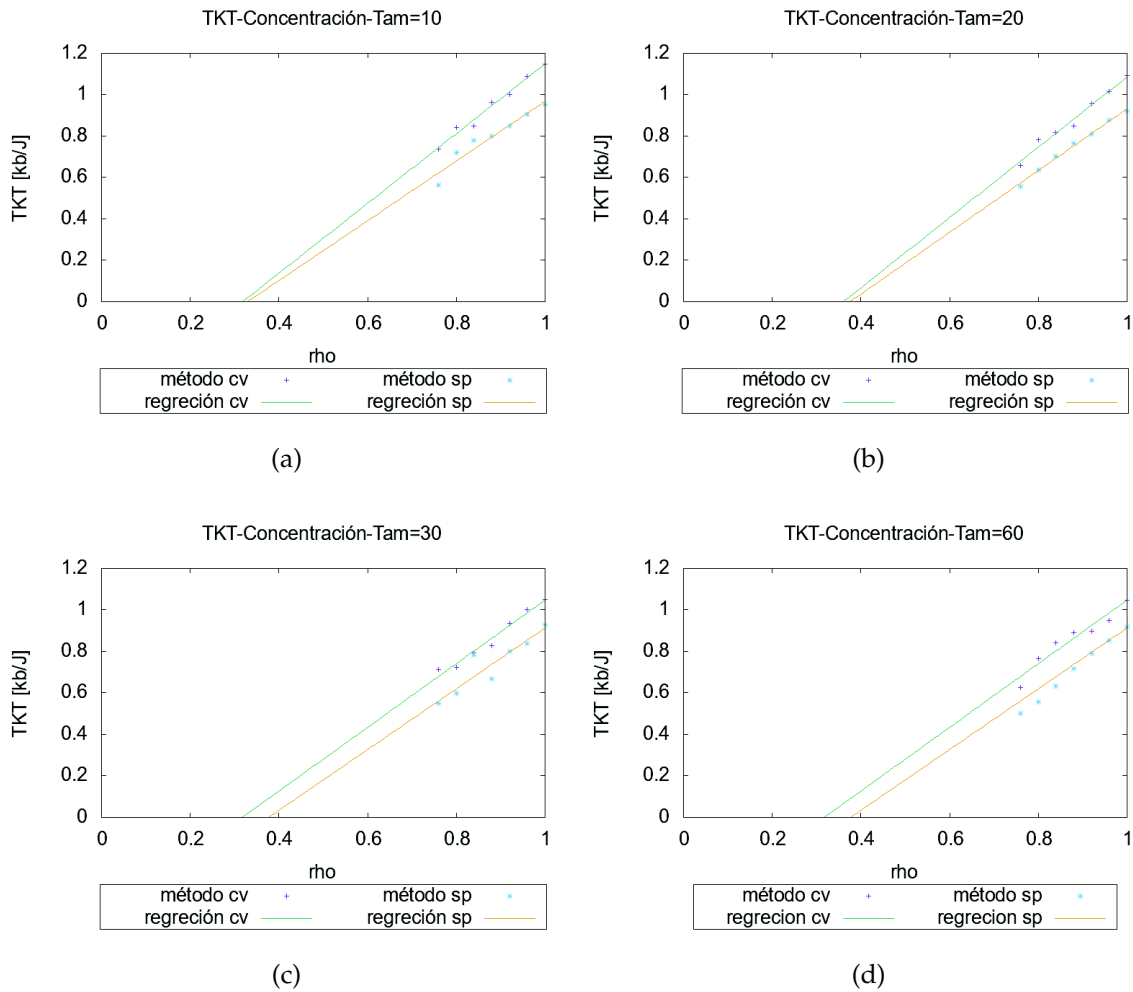


Figura 3.29: Efectos de la concentración de espines en la temperatura de transición a los tamaños: a) Tamaño 10 , b) Tamaño 20 , c) Tamaño 30 , d) Tamaño 60

En la figura 3.29 se puede observar que en todos los tamaños de red presentados se cumple un comportamiento lineal para los dos métodos de la temperatura crítica de transición.

En la figura 3.30 se muestra el comportamiento de la temperatura crítica de tran-

sición en función de la concentración de espines. Del mismo modo que en la figura 3.28 se realizan los ajustes para las concentraciones 1.00 , 0.96 , 0.88 , 0.88 se realizaron para las concentraciones 0.92 , 0.84 y 0.76. Para menores concentraciones el ruido introducido por la aleatoriedad en la posición de las impurezas, imposibilita detectar efectos de tamaño para encontrar la temperatura crítica de transición.

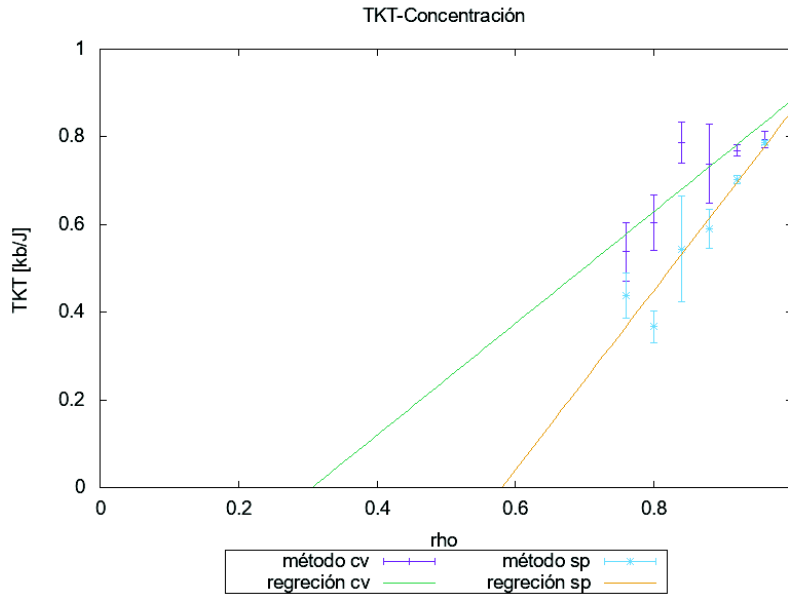


Figura 3.30: Temperatura crítica de transición en el límite termodinámico en función de la concentración de espines.

En la figura 3.30 se puede observar que los errores en el ajuste se incrementan conforme disminuye la concentración de espines. Esto puede deberse a una pérdida de linealidad en los efectos de la concentración de espines o a la aleatoriedad de la presencia de impurezas. Además se puede observar que a mayores diluciones, es decir, a mayor número de impurezas no magnéticas por unidad de área la diferencia entre los dos métodos es mayor.

Para encontrar la temperatura a la que la transición deja de existir se realiza una extrapolación de los datos con las regresiones realizadas. Se encuentra la intersección de la recta con el eje de las x porque ningún sistema puede llegar al cero absoluto y se logra obtener para el método de la capacidad calorífica que transición deja de ocurrir al rededor de $\rho = 0,31$ y con el método de la rigidez de giro que la transición deja de ocurrir al rededor de $\rho = 0,58$.

En la siguiente tabla 3.1 se muestran los datos gráfica dos en la figura 3.30

ρ	TK_{cv}	Error TK_{cv}	Tk_{sp}	Error Tk_{sp}
1.00	0.893581	0.02166	0.870883	0.01786
0.96	0.79327	0.01873	0.786331	0.004612
0.92	0.769112	0.01332	0.702924	0.008772
0.88	0.738681	0.09052	0.590854	0.04431
0.84	0.786628	0.04684	0.544163	0.1208
0.80	0.603698	0.0627	0.366577	0.03723
0.76	0.537608	0.06636	0.438171	0.05186

Tabla 3.1: Temperatura crítica de transición en función de la concentración de espines

Donde ρ es la concentración de impureza, TK_{cv} es la temperatura crítica en el límite termodinámico utilizando el método de la capacidad calorífica y Tk_{sp} es la temperatura crítica en el límite termodinámico utilizando el método de la rigidez de giro.

Capítulo 4

Conclusiones

En este proyecto se generó un software programado en C, que nos sirve para estudiar una lámina de un material magnético con presencia de impurezas no magnéticas, utilizando el modelo XY diluido implementando el Método Montecarlo Metrópolis. Mediante este programa se estudió la transición de fase Kosterlitz-Thouless, mediante el cálculo de las siguientes propiedades termodinámicas en función de la temperatura: energía, capacidad calorífica, magnetización, susceptibilidad magnética, rigidez de giro, función de correlación con sus respectivos exponentes críticos, y número de vórtices. Todas estas propiedades se calcularon para redes de 10, 15, 20, 30 y 60 puntos de red por lado y a concentraciones de 1.00, 0.96, 0.92, 0.88, 0.84, 0.80, 0.76 espines por unidad de área.

Con la utilización del programa creado, se han podido calcular la temperatura crítica mediante dos métodos, donde el primero mide la capacidad calorífica y el segundo la rigidez de giro. En ambos métodos se observan que estas propiedades tienen comportamiento crítico a la misma temperatura cuando el experimento computacional se lo realiza sin presencia de impurezas; pero, comportamientos críticos diferentes con presencia de impurezas, es más, a mayor número de impurezas, más diferencia en sus temperaturas crítica. El resultado del método de la rigidez de giro el que más se asemeja al resultado obtenido previamente en el artículo [21].

Las simulaciones realizadas por el programa nos han permitido constatar que los dos métodos utilizados nos arrojan medidas distintas, para la concentración de impurezas a la cual la transición deja de ocurrir a dilución de 0,31 en el método de la capacidad calorífica, y 0,58 en el método de la rigidez de giro.

Se observa que los datos obtenidos en la gráfica 3.30 de la temperatura crítica en el límite termodinámico en función de la concentración de espines tiene semejanzas

en la temperatura crítica, reportada en la figura 1.10 de A.R. Pereira L.A.S. Mól B.V. Costa S.A. Leonel, P. Zimmermann Coura [21].

Después de varias simulaciones realizadas, pudimos visualizar que las diferencias de tamaño de la red no produce cambios de energía y número de vórtices notables, a ninguna concentración de impurezas.

Los picos presentados en las gráficas de la capacidad calorífica, y la susceptibilidad magnética, tienen un corrimiento a menores temperaturas; tanto por, disminución del tamaño de red como por disminución de la concentración de espines, con los resultados obtenidos el programa parece ser que los efectos de tamaño produce efectos similares que la concentración de espines.

Las figuras 3.18 , 3.20 , 3.22 y 3.24, muestran indicios que la función de correlación decae como una ley de potencia para temperaturas que se encuentran bajo la temperatura de transición y como una ley exponencial para temperaturas sobre la misma.

En las figuras 3.19.b , 3.21.b , 3.23.b , 3.25.b , nos podemos dar cuenta que no existen efectos de tamaño por debajo de la temperatura crítica; y que, al sobrepasarla los efecto de tamaño son apreciables. Lo que podría generar que esta diferencia sirva como un nuevo método para determinar la temperatura crítica

Al realizar los experimentos computacionales, se percibe que la densidad de vórtices es independiente del tamaño de red, además que a bajas temperaturas las impurezas incrementa la formación de vórtices mientras que a altas las disminuye.

Utilizando este mismo programa, creado en este trabajo, se puede realizar un estudio detallado del número de vórtices con respecto al número de impurezas sobretudo al rededor de la temperatura $1,445k_b/J$. Con pequeñas modificaciones al programa se puede estudiar la influencia de la posición de las impurezas en las propiedades termodinámicas.

Debido a que en la rigidez de giro en función de la temperatura, cuando tiene un mayor porcentaje de impurezas presenta más ruido, se recomienda que se estudie la influencia de la ubicación de impurezas en la la red para altas concentraciones de las mismas.

Apéndice A

Suplementos Matemáticos

A.1. Topología

A.1.1. Espacio topológico

Un espacio topológico es un par ordenado (X, τ) donde X es un conjunto no vacío y $\tau \subseteq \text{pot}(X)$ tal que:

1. $X \in \tau, \emptyset \in \tau$
2. $A, B \in \tau \rightarrow A \cup B \in \tau$
3. $\forall s \subseteq \tau, \bigcup_{A \in s} A \in \tau$

A.1.2. Triangulación

En un espacio topológico de dos dimensiones continuo, a la intersección de dos curvas sobre el espacio se denomina vértices. La curva restringida entre dos vértices que no contiene ningún vértice intermedio se denomina arco o arista. Un triángulo es un subespacio topológico limitado por tres vértices y tres arcos. Al interior del triángulo se nomina cara, es decir un triángulo tiene una sola arista. Una triangulación consiste en recubrir el espacio por triángulos que solo tengan intersección en las fronteras.

A.1.3. Característica de Euler

La característica de Euler es un invariante topológico que puede ser definido o calculado de diferentes maneras. La forma más usual es utilizar una triangulación arbitraria. Para una triangulación dada, la característica de Euler \mathcal{X} , se define como el número de caras C , más el número de vértices V , menos el número de lados L . Es decir,

$$\mathcal{X} = C + V - L \quad (\text{A.1})$$

A.2. Geometría

A.2.1. Variedad diferencial

Una variedad diferencial es un espacio topológico (X, τ) con una colección de funciones $\mathcal{A} = \{\phi_i\}$ denominado Atlas maximal que cumple:

- En cada punto de la variedad se puede encontrar una vecindad tal que existe un homeomorfismo ϕ con un abierto de \mathbb{R}^n .
- Si dos de estos homeomorfismos ϕ_1 y ϕ_2 tienen dominios cuya intersección es diferente de vacío, se deben cumplir en su intersección: $\phi_1 \circ \phi_1^{-1}$ es una función infinitamente diferenciable.
- Todos homeomorfismos que cumplen 1 y 2 son elementos de \mathcal{A}

A.2.2. Vector tangente

Un vector tangente es un funcional V que toma; funciones que van de la variedad en los reales; y nos devuelve un real. Esta definición surge de considerar a los vectores tangentes como derivadas. Cabe recalcar que el conjunto de todos los vectores en un punto dado tiene estructura de espacio vectorial.

A.2.3. Campo de vectores

Un campo de vectores x es una aplicación que asigna a cada punto de la variedad un vector del espacio tangente en ese punto. Al conjunto de todos los campos

vectoriales que se pueden definir en una variedad se le nota como $\chi(X)$

A.2.4. Conexión

Una conexión es una aplicación que toma dos campos de vectores y nos entrega un campo de vectores que cumple las siguientes propiedades.

$$\begin{aligned} \nabla : \chi(X) \times \chi(X) &\longrightarrow \chi(X) \\ (x, y) &\longmapsto \nabla_x y \end{aligned} \tag{A.2}$$

- Es lineal en la primera variable: $\nabla_{x+y}Z = \nabla_x Z + \nabla_y Z$
- Es lineal en la segunda variable: $\nabla_x w + z = \nabla_x w + \nabla_x z$
- Cumple la regla de Leibniz en la segunda variable. $\nabla_x fZ = df(x)Z + \nabla_x Z$
- Finalmente $\nabla_{fx}Z = f(x)\nabla_x Z$

En una misma variedad se pueden definir varias conexiones, pero existe una conexión especial denominada conexión de Levi-Civita. La conexión de Levi-Civita conecta la variedad de modo que las nociones de distancia no cambian de punto a punto. Esta conexión se puede definir sobre una variedad bidimensional mediante el embebimiento en R^k para algún k , tomando la proyección de la derivada direccional sobre el plano tangente a la variedad.

$$\nabla_x y = D_x y - N_{D_x y} \tag{A.3}$$

La conexión como cumple una regla de Leibniz también se denomina derivada covariante.

A.2.5. Curvatura

La curvatura que nos interesa es la curvatura Gaussiana K , que tomando una conexión que nace del embebimiento, es el jacobiano de la transformación de Weingarten. La transformación de Weingarten a cada punto le corresponde la derivada covariante del unitario normal a la variedad en ese punto.

A.2.6. Teorema de Gauss-Bonnet

Sea una región \mathcal{R} Sin borde se cumple:

$$\iint_{\mathcal{R}dA} K = 2\pi\chi \quad (\text{A.4})$$

Donde χ es la característica de Euler.

A.2.7. Defectos Topológicos en el modelo xy

Dentro del modelo xy, a cada estado del sistema se le otorga una función Θ , donde a cada punto de red se le asigna una dirección particular. Esta dirección puede estar representada por un ángulo o un punto sobre la circunferencia, además estas direcciones tienen la propiedad de sumarse, esto constituye un tipo de grupo de Lie llamado U_1 el cual puede tener varias representaciones obteniendo un mismo resultado.

Al número de espines que tenemos en cada lado del sistema se le llamará p , para modelar se procede a realizar la formación de grupos cíclicos, que se da al utilizar las condiciones de frontera periódica en un cada lado de la red. El número de elementos que tiene el grupo cíclico es p . Para caracterizar a la red se requiere el producto cartesiano de dos grupos cíclicos; ya que, en este caso la red sera bidimensional, representado en la siguiente ecuación (A.5).

$$\begin{aligned} \Theta : \mathbb{Z}_p \times \mathbb{Z}_p &\longrightarrow U_1 \\ (i, j) &\longmapsto \Theta(i, j) = \theta_{ij} \end{aligned} \quad (\text{A.5})$$

Para realizar una aproximación al continuo, la red con condiciones de frontera periódicas son transformadas en un toro y el grupo de Lie se transforma en un campo de vectores tangentes sobre el toro. Representado en la siguiente ecuación (A.6).

$$\begin{aligned} s : \mathbb{T}^2 &\longrightarrow T\mathbb{T}^2 \\ x &\longmapsto s(x) \in T_x\mathbb{T}^2 \end{aligned} \quad (\text{A.6})$$

Los puntos en los cuales la función se hace cero se denominan singularidades. Si alrededor de esta no existe otra singularidad se denomina singularidades aisladas.

Si se proporciona a un toro una conexión, cualquiera, se puede tener nociones de: ángulos, paralelismo y perpendicularidad.

A una función continua $\gamma : [0, 1] \rightarrow \mathbb{T}^2$ se denomina *lazo* si el inicio y el final de esta coinciden $\gamma(0) = \gamma(1)$. La composición $s \circ \gamma$ define el campo restringido a la curva con un parámetro t . Frente a un vector fijo, en cada punto de la curva se puede definir un ángulo entre el campo s y el vector fijo. Una función continua asigna, a cada parámetro t el ángulo al cual lo llamaremos $\phi(t)$. Al número de giros orientados que realiza el campo a lo largo de la curva, considerando, la curva como positiva, se denomina índice I_γ de la curva. En efecto:

$$2\pi I_\gamma = \phi(1) - \phi(0) \quad (\text{A.7})$$

Donde $\chi(\mathbb{T}^2)$ es la característica de Euler del toro que es 0. Los puntos de singularidad con un índice mayor que cero se denominan *vórtices*, y los puntos de singularidad con índice menor que cero se denominan *antivórtices*. El teorema de Euler Poncaire nos demuestra que con el paso al continuo y condiciones de frontera periódicas existe una conservación entre vórtices y anti-vórtices y eso solo depende de la topología. A estas singularidades se les denomina defectos topológicos.

A.3. Probabilidad

A.3.1. Espacio de eventos

Se tiene un espacio muestral Ω que contiene todos los resultados de un determinado fenómeno. El espacio de eventos \mathcal{F} está compuesto por todas las posibles combinaciones de eventos que pueden ocurrir. En efecto el espacio de eventos está conformado por subconjuntos del espacio muestral. Y cumple las siguientes propiedades.

- Existe una probabilidad de que algo ocurra es decir $\Omega \in \mathcal{F}$
- Si dos eventos pueden ocurrir el uno o el otro o pueden ocurrir es decir $A \in \mathcal{F}$ y $B \in \mathcal{F}$ entonces $A \cup B \in \mathcal{F}$
- Si algo puede ocurrir también puede no ocurrir, es decir $A \in \mathcal{F}$ entonces $A^c \in \mathcal{F}$

En efecto el espacio de eventos tiene estructura de sigma álgebra [23].

A.3.2. Espacio de probabilidad

Un espacio de probabilidad es un espacio medido. Es decir es una triplete (Ω, \mathcal{F}, P) donde: Ω es el espacio muestral, \mathcal{F} es el espacio de eventos y P es una función aditiva llamada probabilidad que va del espacio de eventos al intervalo cero uno.

A.3.3. Variable Aleatoria

Una variable aleatoria es una función que va del espacio muestral a los reales.

A.3.4. Proceso Estocástico

Un proceso estocástico es una función asigna un numero de un subconjunto de los de los reales, a una variable aleatoria.

A.3.5. Cadena de Márkov

La definición matemática de la cadena de Márkov; Sea (Ω, \mathcal{F}, P) un espacio de probabilidades y S un conjunto no vacío. Una sucesión de variables aleatorias $\{X_t : \Omega \rightarrow \mathbf{R}, n \in \mathbb{N}\}$, donde para toda sucesión x_i que valores de S , se cumple la ecuación [23] (A.8)

$$P(X_{i+1} = x_{i+1} | X_i = x_i, X_{i-1} = x_{i-1}, \dots, X_0 = x_0) = P(X_{i+1} = x_{i+1} | X_i = x_i) \quad (\text{A.8})$$

Si en cada variable aleatoria se toma un punto muestral se obtiene una función, denominada, función muestra $x(t)$, que es totalmente determinista en la variable t , que es la variable temporal

Apéndice B

Manual

B.1. Modo de uso

En el apéndice C se encuentran el código para tres archivos de nombre : main, xy.c y xy.h . El archivo main contiene las instrucciones básicas y puede ser entendido y configurado con el presente manual. El código del apéndice calcula las propiedades de una red 10×10 espines, con 5 impurezas, a las temperaturas 0,2, 0,4, 0,6, ..., 2. Para realizar estos cálculos el programa debería demorarse al rededor de dos horas en una computadora de escritorio.

Para utilizar el programa creado, se necesita compilar el archivo main en gcc C std c99, en una distribución de linux, añadiendo la biblioteca xy.h.

Para compilar y ejecutar simplemente copie los tres archivos en una carpeta y en una terminal de Linux, ejecute:

```
|| gcc -g -std=c99 xy.c xy.h main.c -lm -lpthread ; ./a.out
```

El código principalmente consiste en una biblioteca, que contiene 3 estructuras , 2 funciones principales y 27 funciones auxiliares.

B.2. Estructuras

B.2.1. sis

La estructura sis representa un estado del sistema. Esta estructura está en constante actualización en cada evolución Monte Carlo. Contiene la información correspondiente a la red, su tamaño, el ángulo, el hamiltoniano asociado a ese estado,

su magnetización. Además cuenta con las condiciones de frontera y el sistema de indización necesario para des-criminar entre huecos

B.2.2. `sist`

La estructura `sist` contiene la información más relevante sobre el ensamble contiene después de ser creadas las magnitudes promedio tales como: energía, magnetización, capacidad calorífica, susceptibilidad magnética, función de correlación y exponentes críticos.

B.2.3. `dat`

La estructura `dat` contiene toda la información que se requiere para crear una estructura `sist`. Los datos que se requieren son: la temperatura a la que se encuentra el sistema, el tamaño de la red tanto para x como para y , el número de huecos, y el número de pasos que se requiere para termalizar y el número de pasos que se requiere para ejecutar.

B.3. Funciones

En esta sección solo se indicará el uso de las 3 funciones más importantes.

B.3.1. `crear_sis_montecarlo`

Esta función es la principal función de ejecución la cual tiene la siguiente declaración

```
void* crear_sis_montecarlo(void* arg);
```

Esta función puede ser usada como:

```
datos dat;  
...  
...  
...  
crear_sis_montecarlo((void*)&(dat));
```

Esta función crea la estructura `sist`.

B.3.2. Maximo maximal

Esta función permite encontrar el máximo y el punto maximal. Tiene la siguiente declaración

```
|| void maximoMaximal(double * x, double *y, int n ,  
|| double* maximo, double *maximoMaximal);
```

El arreglo x contiene la variable independiente, el arreglo y contiene la variable dependiente, n es el número de datos que tienen los dos arreglos, «maximo» . Es el valor máximo de la variable dependiente y «maximoMaximal» es la pre-imagen del valor máximo.

Apéndice C

Código

C.1. Main

```
#include "xy.h"
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <time.h>
#include <math.h>
#include <sys/stat.h>
#include <sys/types.h>

int main()
{
    printf("Comenso Programa");
    clock_t tiempo_inicio, tiempo_final;
    double segundos;
    tiempo_inicio = time(NULL);

    srand(time(NULL));
    int NTemp = 10;
    int NDiluido = 1;
    int NLong = 1;
    pthread_t mont[NDiluido][NLong][NTemp];
    datos dat[NDiluido][NLong][NTemp];

    for (int i = 0; i < NDiluido; i++)
        for (int l = 0; l < NLong; l++)
            for (int j = 0; j < NTemp; j++)
```

```

    {
        dat[i][l][j].n = 300000; // n mero de pasos para
            termalizar
        dat[i][l][j].nt = 300000; // n mero de pasos para
            promediar
        dat[i][l][j].t1 = 0.2 + 0.2 * j; // temperatura
            frontera 1
        dat[i][l][j].t2 = dat[i][l][j].t1; // temperatura
            frontera 2
        dat[i][l][j].nx = 10; // tama o del lado x
        dat[i][l][j].ny = dat[i][l][j].nx; // tama o del y
        dat[i][l][j].num_huecos = 5; // n mero de
            impurezas
        pthread_create(
            &(mont[i][l][j]), NULL, crear_sis_montecarlo, (
                void*)&(dat[i][l][j]));
    }

for (int i = 0; i < NDiluido; i++)
    for (int l = 0; l < NLong; l++)
        for (int j = 0; j < NTemp; j++)
            {
                pthread_join(mont[i][l][j], NULL);
            }

double x[NLong][NTemp];
double y_cv[NLong][NTemp];
double temp_Critica_cv[NLong][NDiluido];
double y_sl[NLong][NTemp];
double temp_Critica_sl[NLong][NDiluido];
char dirDatos [] = "Datos";
char dirDiluido [200];
char dirLong [200];
double max;
double maximal;
double* regrecion_aux;
FILE* archivo;
char nombre [500];
for (int i = 0; i < NDiluido; i++)
    for (int l = 0; l < NLong; l++)
        {
            sprintf(
                dirDiluido, "%s%s%03.10f", dirDatos, "/Diluido=",
                    ((dat[i][l][0].sist).a).diluido);

```

```

sprintf(dirLong, "%s%i", dirDiluido, "/Longitud=", ((
    dat[i][l][0].sist).a).nx);
sprintf(nombre, "%s%s", dirLong, "/Energia.dat");
archivo = fopen(nombre, "a");
for (int j = 0; j < NTemp; j++)
{
    x[l][j] = (dat[i][l][j].sist).t1;
    y_cv[l][j] = (dat[i][l][j].sist).cv;
    y_sl[l][j] = (dat[i][l][j].sist).tan;

    regrecion_aux = (dat[i][l][j].sist).regrecion;
    fprintf(archivo, "%03.10f ", dat[i][l][j].t1); //
        temperatura
    fprintf(archivo, "%03.10f ", (dat[i][l][j].sist).Em
        ); // Energ a
    fprintf(archivo, "%03.10f ", (dat[i][l][j].sist).cv
        ); // Capacidad calor fica
    fprintf(archivo, "%03.10f ", (dat[i][l][j].sist).Mm
        ); // Magnetizaci n
    fprintf(archivo, "%03.10f ", (dat[i][l][j].sist).x)
        ; // Susceptibilidad magnetica
    fprintf(archivo, "%03.10f ",
        (dat[i][l][j].sist).Em2); // Desviaci n
        estandar Energ a
    fprintf(archivo, "%03.10f ",
        (dat[i][l][j].sist).Mm2); // Desviaci n
        estandar magnetizaci n
    fprintf(archivo, "%03.10f ", regrecion_aux[0]); //
        Eta
    fprintf(archivo, "%03.10f ", regrecion_aux[1]); //
        Longitud de correlaci n
    fprintf(archivo, "%03.10f ", regrecion_aux[2]); //
        Constante de proporcionalidad Exp
    fprintf(archivo, "%03.10f ", regrecion_aux[3]); //
        Constante de proporcionalidad
    fprintf(archivo, "%03.10f ", ((dat[i][l][j].sist).
        sl)); // Rigides de giro
    fprintf(archivo, "%03.10f ", ((dat[i][l][j].sist).
        sl2));
    fprintf(archivo, "%03.10f ",
        ((dat[i][l][j].sist).tan)); // Distancia entre
        pendientes, Rigides de giro y 2/pi
    fprintf(archivo, "%03.10f ", (dat[i][l][j].sist).
        vor); // Dencidad de Vortices

```

```

        fprintf(archivo, "\n");
    }

    maximoMaximal(x[l], y_cv[l], NTemp, &max, &maximal);
    temp_Critica_cv[i][l] = maximal;
    maximoMaximal(x[l], y_sl[l], NTemp, &max, &maximal);
    temp_Critica_sl[i][l] = maximal;
    fclose(archivo);
}

for (int i = NDiluido - 1; i >= 0; i--)
{
    sprintf(dirDiluido, "%s%s%03.10f", dirDatos, "/Diluido=",
        ((dat[i][0][0].sist).a).diluido);
    sprintf(nombre, "%s%s", dirDiluido, "/TempCriticaLongitud.
        dat");
    FILE* archivo3 = fopen(nombre, "w");
    for (int l = 0; l < NLong; l++)
    {
        fprintf(archivo3, "%i %f %f %i\n", ((dat[i][l][0].sist)
            .a).nx, temp_Critica_cv[i][l],
            temp_Critica_sl[i][l], ((dat[i][l][0].sist).a).n_h)
            ;
    }
}

tiempo_final = time(NULL);
segundos = (tiempo_final - tiempo_inicio);
FILE* archivo2 = fopen("Datos/tiempo.txt", "a");
fprintf(archivo2, "%i %f", dat[0][0][0].nx, segundos);
printf("Programa Totalmente terminado\n");
return 0;
}

```

C.2. xy.c

```

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <time.h>
#include <math.h>
#include <string.h>
#include <unistd.h>

```

```

#include <sys/stat.h>
#include <sys/types.h>

int mkdir(const char* pathname, mode_t mode);

const double pi = 3.1415926535897932384626433832795028841971;
const double pi2 = 6.28318530717958647692528676655;
int permitir = 0;

typedef struct
{
    int nx;
    int ny;
    int ns;
    int n_cor;
    int n_vortices;
    int n_huecos;
    int n_h;
    int v_en_h;

    double E;
    double Mx;
    double My;
    double diluido;
    double J;
    double B;
    double sl;

    int* indice;
    double* regrecion;

    double** s;
    double** sx;
    double** sy;
    double** corr;
    double** vortice;
    short** sd;
    short* f;

    int** ij;
} sis;

typedef struct

```

```

{
    sis a;
    int n;
    int nt;
    double Mm;
    double Em;
    double Em2;
    double Mm2;
    double cv;
    double x;
    double t1;
    double t2;
    double s1;
    double tan;
    double s12;
    double** Mag_franja;
    double vor;
    int n_cor;
    double** corr;
    double* regrecion;
} sis_montecarlo;

typedef struct
{
    int n;
    int nt;
    double t1;
    double t2;
    int nx;
    int ny;
    double J;
    double B;
    int num_huecos;
    sis_montecarlo sist;
} datos;

double r();
double r2();
double ri();
double rci(double spin);
double rd();

```

```

void barajar(int baraja[], int* n);
sis crear_sis(int nx, int ny, double J, double B, int);
int index2(int i, int j, int nx);
void frontera(sis* a);
void guardar_matriz(const char* file_name, double** matriz, int nx,
    int ny);
void abrir_matriz(const char* nombre, double** matriz);
void calcular_sis(sis* a);
void montecarlo(sis* a, int k, double* beta);
void montecarlo1(sis* a, int k, double* beta);
void guardar_matrizi(const char* file_name, short** matriz, int nx,
    int ny);
void* crear_sis_montecarlo(void* arg);
void cambiar_frontera(int i, int j, double sAux, double sxAux,
    double syAux, sis* a);
void calcular_corr(sis* a);
void ordenar(double* A, int N);
void guardar_sistema(char* apodo, sis_montecarlo sist);
void regrecion_lineal(double* x, double* y, double n, double* m,
    double* b);
void calc_longitud_corr(sis* a, char* dir);
void maximoMaximal(double* x, double* y, int n, double* maximo,
    double* maximoMaximal);
void xyGuardar(char* nombre, sis* a);
void guardar_vortice(const char* dir, sis a);
void contar_vortice(sis* a);
void periodico(int* i, int* j, int nx, int ny);
void guardar_Parametros(const char* dir, sis_montecarlo sist);
void calc_longitud_corr_mon(sis_montecarlo* sist, char* dir);
void exp_reg(double* x, double* y, int n_corr, double* a, double* b
    );
void pot_reg(double* x, double* y, int n, double* a, double* b);

sis crear_sis(int nx, int ny, double J, double B, int n_h)
{
    //
    // =====

    // Copia y declara variables

    sis a;
    a.nx = nx;

```

```

a.ny = ny;
a.J = J;
a.B = B;
int nxy = nx * ny;
a.ns = nxy - n_h;
a.n_h = n_h;
a.n_vortices = 0;
a.n_huecos = 0;
a.indice = malloc((a.ns) * sizeof(int*));

a.s = malloc((nx + 2) * sizeof(double*));
a.sx = malloc((nx + 2) * sizeof(double*));
a.sy = malloc((nx + 2) * sizeof(double*));
a.sd = malloc((nx + 2) * sizeof(short*));
for (int i = 0; i < (nx + 2); i++)
{
    a.s[i] = malloc((ny + 2) * sizeof(double));
    a.sx[i] = malloc((ny + 2) * sizeof(double));
    a.sy[i] = malloc((ny + 2) * sizeof(double));
    a.sd[i] = malloc((ny + 2) * sizeof(short));
}

a.ij = malloc(2 * sizeof(int*));
a.ij[0] = malloc(((nx + 2) * (ny + 2)) * sizeof(int));
a.ij[1] = malloc(((nx + 2) * (ny + 2)) * sizeof(int));

a.corr = malloc(3 * sizeof(double*));
a.corr[0] = malloc(nxy * sizeof(double)); // radio
a.corr[1] = malloc(nxy * sizeof(double)); // correlacion
a.corr[2] = malloc(nxy * sizeof(double)); // desviacion
        correlacion

a.vortice = malloc(3 * sizeof(double*));
a.vortice[0] = malloc(a.ns * sizeof(double)); // i
a.vortice[1] = malloc(a.ns * sizeof(double)); // j
a.vortice[2] = malloc(a.ns * sizeof(double)); // pos
a.regrecion = malloc(4 * sizeof(double));

// Fin copia y declara variables

//
=====

```



```

// Crear Red
for (int i = 0; i < nx + 2; i++)
{
    for (int j = 0; j < ny + 2; j++)
    {
        a.s[i][j] = 0; // r2();
        a.sx[i][j] = cos(a.s[i][j]);
        a.sy[i][j] = sin(a.s[i][j]);
        a.sd[i][j] = 1;
        if ((j == 0 || i == 0) || (i == nx + 1 || j == ny + 1))
            a.sd[i][j] = 0;
    }
}
// Fin Crear red
// crear indices

int ives[8] = { -1, -1, 0, 1, 1, 1, 0, -1 };
int jves[8] = { 0, 1, 1, 1, 0, -1, -1, -1 };

a.diluido = (nxy - n_h) / (double)nxy;

if (a.diluido < 0.9)
{
    printf("\n");
    permitir = 1;
    printf("Advertencia, sistema con muchas impurezas. Este
           programa solo puede contar "
           "v rtices hasta el 10 porciento de impurezas. El
           resto de funciones funcionan con "
           "normalidad pero el numero de v rtices ser un
           numero aleatorio que no corresponde "
           "al numero de v rtices del sistema.\n");
    printf("Esta simulaci n tiene el %f porciento de impurezas
           ", (1 - a.diluido) * 100);
}

for (int i = 0; i < nx + 2; i++)
{
    for (int j = 0; j < ny + 2; j++)
    {
        a.ij[0][index2(i, j, nx)] = i;
        a.ij[1][index2(i, j, nx)] = j;
    }
}
}

```

```

int i1, j1, j2, i2, suma;
for (int k = 0; k < n_h; k++)
{
    do
    {
        i1 = (rand() % nx) + 1;
        j1 = (rand() % ny) + 1;
        suma = 0;
        for (int l = 0; l < 8; l++)
        {
            i2 = i1 + ives[l];
            j2 = j1 + jves[l];
            suma += a.sd[i2][j2];
        }
        if (permitir == 1)
            suma = 8;
        if (suma == 8 && a.sd[i1][j1] != 0)
        {
            a.sd[i1][j1] = 0;
            break;
        }
    } while (0 == 0);
}

int aux2 = 0;

for (int i = 1; i < nx + 1; i++)
{
    for (int j = 1; j < ny + 1; j++)
    {
        if (a.sd[i][j])
        {
            a.indice[aux2] = index2(i, j, nx);
            aux2++;
        }
    }
}

frontera(&a);
calcular_sis(&a);
if (permitir == 0)
    contar_vortice(&a);

```

```

    printf("Creo Sistema\n");
    // Fin crear indices
    return a;
}

double r()
{
    return (double)rand() / (double)RAND_MAX;
}

double ri()
{
    if (rand() % 2 == 0)
        return 0.0;
    return pi;
}

double rci(double spin)
{
    if (spin == 0)
        return pi;
    return 0;
}

double r2()
{
    return ((double)rand() / (double)RAND_MAX) * pi2;
}

double rd()
{
    return (((double)rand() / (double)RAND_MAX) * 0.1) - 0.0;
}

int index2(int i, int j, int nx)
{
    return i + j * (nx + 2);
}

void barajar(int baraja[], int* n)
{
    int aux[*n];

```

```

for (int i = 0; i < *n; i++)
    aux[i] = baraja[i];

short usado[*n];
for (int i = 0; i < *n; i++)
    usado[i] = 0;
int index = 0;
for (int i = 0; i < *n; i++)
{
    do
    {
        index = (rand() % *n);
    } while (usado[index]);
    baraja[i] = aux[index];
    usado[index] = 1;
}
}

void frontera(sis* a)
{
    int nx = (*a).nx;
    int ny = (*a).ny;
    a->f = malloc(sizeof(unsigned short) * ((nx + 2) * (ny + 2)));
    for (int l = 0; l < (nx + 2) * (ny + 2); l++)
        a->f[l] = 0;

    for (int j = 1; j < ny + 1; j++)
    {
        (*a).s[0][j] = (*a).s[nx][j];
        (*a).s[nx + 1][j] = (*a).s[1][j];

        (*a).sx[0][j] = (*a).sx[nx][j];
        (*a).sx[nx + 1][j] = (*a).sx[1][j];

        (*a).sy[0][j] = (*a).sy[nx][j];
        (*a).sy[nx + 1][j] = (*a).sy[1][j];

        (*a).sd[0][j] = (*a).sd[nx][j];
        (*a).sd[nx + 1][j] = (*a).sd[1][j];
        a->f[index2(1, j, nx)] = 1;
        a->f[index2(nx, j, nx)] = 1;
    }
}

```

```

for (int i = 1; i < nx + 1; i++)
{
    (*a).s[i][0] = (*a).s[i][ny];
    (*a).s[i][ny + 1] = (*a).s[i][1];

    (*a).sx[i][0] = (*a).sx[i][ny];
    (*a).sx[i][ny + 1] = (*a).sx[i][1];

    (*a).sy[i][0] = (*a).sy[i][ny];
    (*a).sy[i][ny + 1] = (*a).sy[i][1];

    (*a).sd[i][0] = (*a).sd[i][ny];
    (*a).sd[i][ny + 1] = (*a).sd[i][1];

    a->f[index2(i, 1, nx)] = 1;
    a->f[index2(i, ny, nx)] = 1;
}
(*a).s[0][0] = (*a).s[nx][ny];
(*a).s[nx + 1][0] = (*a).s[1][ny];
(*a).s[0][ny + 1] = (*a).s[nx][1];
(*a).s[nx + 1][ny + 1] = (*a).s[1][1];

(*a).sd[0][0] = (*a).sd[nx][ny];
(*a).sd[nx + 1][0] = (*a).sd[1][ny];
(*a).sd[0][ny + 1] = (*a).sd[nx][1];
(*a).sd[nx + 1][ny + 1] = (*a).sd[1][1];

(*a).sx[0][0] = (*a).sx[nx][ny];
(*a).sx[nx + 1][0] = (*a).sx[1][ny];
(*a).sx[0][ny + 1] = (*a).sx[nx][1];
(*a).sx[nx + 1][ny + 1] = (*a).sx[1][1];

(*a).sy[0][0] = (*a).sy[nx][ny];
(*a).sy[nx + 1][0] = (*a).sy[1][ny];
(*a).sy[0][ny + 1] = (*a).sy[nx][1];
(*a).sy[nx + 1][ny + 1] = (*a).sy[1][1];
}

void cambiar_frontera(int i, int j, double sAux, double sxAux,
    double syAux, sis* a)
{
    if (i == 1)
    {

```

```

        a->s[a->nx + 1][j] = sAux;
        a->sx[a->nx + 1][j] = sxAux;
        a->sy[a->nx + 1][j] = syAux;
    }
    if (i == a->nx)
    {
        a->s[0][j] = sAux;
        a->sx[0][j] = sxAux;
        a->sy[0][j] = syAux;
    }
    if (j == 1)
    {
        a->s[i][a->ny + 1] = sAux;
        a->sx[i][a->ny + 1] = sxAux;
        a->sy[i][a->ny + 1] = syAux;
    }
    if (j == a->ny)
    {
        a->s[i][0] = sAux;
        a->sx[i][0] = sxAux;
        a->sy[i][0] = syAux;
    }
}

void guardar_matriz(const char* nombre, double** matriz, int nx,
int ny)
{
    FILE* archivo = fopen(nombre, "wb");
    if (archivo == NULL)
    {
        printf("\nError\n");
        exit(EXIT_FAILURE);
    }

    for (int j = 0; j < ny; j++)
    {
        for (int i = 0; i < nx; i++)
            fprintf(archivo, "%03.10f ", (double)matriz[i][j]);

        fprintf(archivo, "\n");
    }
    fclose(archivo);
}

```

```

void guardar_matrizi(const char* nombre, short** matriz, int nx,
    int ny)
{
    FILE* archivo = fopen(nombre, "wb");
    if (archivo == NULL)
    {
        printf("\nError\n");
        exit(EXIT_FAILURE);
    }

    for (int j = 0; j < ny; j++)
    {
        for (int i = 0; i < nx; i++)

            fprintf(archivo, "%i ", matriz[i][j]);
        fprintf(archivo, "\n");
    }
    fclose(archivo);
}

void abrir_matriz(const char* nombre, double** matriz)
{
    FILE* archivo = fopen(nombre, "rb");
    if (archivo == NULL)
    {
        printf("\nError\n");
        exit(EXIT_FAILURE);
    }

    char c;
    int ny = 0;
    int nx = 0;
    char espacio = ' ';
    char enter = '\n';
    char auxc [] = "4.1206189225";
    while (!feof(archivo))
    {
        c = fgetc(archivo);
        if (c == enter)
            ny++;
        if (c == espacio)
            nx++;
    }
}

```

```

rewind(archivo);
nx = nx / ny;
matriz = malloc(sizeof(double*) * nx);
for (int i = 0; i < nx; i++)
    matriz[i] = malloc(sizeof(double) * ny);
int i = 0;
int j = 0;
int k = 0;
while (!feof(archivo))
{
    c = fgetc(archivo);
    auxc[k] = c;
    k++;
    if (c == espacio)
    {
        matriz[i][j] = atof(auxc);
        i++;
        k = 0;
    }
    if (c == enter)
    {
        i = 0;
        j++;
    }
}
printf("\nMatriz de %i por %i abierta\n", nx, ny);
fclose(archivo);
}

void calcular_sis(sis* a)
{
    double E = 0;
    double Mx = 0;
    double My = 0;
    double aux, spin;
    a->sl = 0;
    for (int j = 1; j < a->ny + 1; j++)
        for (int i = 1; i < a->nx + 1; i++)
            {
                spin = (a->s[i][j]) * (a->sd[i][j]);
                aux = cos(a->s[i + 1][j] - spin) * a->sd[i + 1][j]
                    + cos(a->s[i - 1][j] - spin) * a->sd[i - 1][j]
                    + cos(a->s[i][j + 1] - spin) * a->sd[i][j + 1]
                    + cos(a->s[i][j - 1] - spin) * a->sd[i][j - 1];
            }
}

```



```

        Mx = Mx + cos(spin) * a->sd[i][j];
        My = My + sin(spin) * a->sd[i][j];
        E = E - a->J * aux - 2 * a->B * cos(spin) * a->sd[i][j
            ];
        aux = sin(spin - a->s[i + 1][j]) * (a->sd[i + 1][j])
            - sin(spin - a->s[i - 1][j]) * (a->sd[i - 1][j]);
        a->sl += aux;
    }

    a->E = E / 2.0;
    a->sl = a->sl / 2.0;
    a->Mx = Mx;
    a->My = My;
}

void* crear_sis_montecarlo(void* arg)
{
    sis_montecarlo sist;
    datos* dat;
    dat = (datos*)arg;
    sis a;

    sist.n = (*dat).n;
    sist.nt = (*dat).nt;
    sist.t1 = (*dat).t1;
    sist.t2 = (*dat).t2;
    sist.Em = 0.0;
    sist.Mm = 0.0;
    sist.Em2 = 0.0;
    sist.Mm2 = 0.0;
    sist.regrecion = malloc(4 * sizeof(double));
    sist.corr = malloc(3 * sizeof(double*));

    a = crear_sis((*dat).nx, (*dat).ny, 1, 0, dat->num_huecos);

    double beta = 1.0 / ((double)(sist.t1));
    // estabilizar
    for (int l = 0; l < (*dat).nt; l++)
    {
        barajar(a.indice, &(a.ns));
        for (int k = 0; k < a.ns; k++)
        {
            montecarlo(&a, a.indice[k], &beta);
        }
    }
}

```

```

    }
}
///promedio.
calcular_corr(&a);
if (permitir == 0)
    contar_vortice(&a);
sist.corr[0] = malloc(a.n_cor * sizeof(double)); // radio
sist.corr[1] = malloc(a.n_cor * sizeof(double)); // correlacion
sist.corr[2] = malloc(a.n_cor * sizeof(double)); // desviacion
    correlacion
sist.n_cor = a.n_cor;
sist.vor = a.n_vortices;

for (int l = 0; l < a.n_cor; l++)
{
    sist.corr[0][l] = a.corr[0][l]; // radio
    sist.corr[1][l] = a.corr[1][l]; // correlacion
    sist.corr[2][l] = a.corr[2][l]; // desviacion correlacion
}
sist.sl = 0;
int salto_corr = 100;
double pm = 0;
double pasos = 1.0 / ((double)(a.ns * sist.n));
double aux1 = 1.0 / ((double)(a.ns));
double aux3 = 1.0 / ((double)(a.nx * a.ny));

for (int l = 0; l < (*dat).n; l++)
{
    barajar(a.indice, &(a.ns));
    for (int k = 0; k < a.ns; k++)
    {
        montecarlo(&a, a.indice[k], &beta);
        sist.Em += a.E;
        pm = sqrt(a.Mx * a.Mx + a.My * a.My);
        sist.Mm += pm;
        sist.Em2 += a.E * a.E;
        sist.Mm2 += pm * pm;
        sist.sl += a.sl * a.sl;
    }
    if (l % salto_corr == 0)
    {
        calcular_corr(&a);
        if (permitir == 0)
            contar_vortice(&a);
    }
}

```

```

    for (int l = 0; l < a.n_cor; l++)
    {
        sist.corr[0][l] += a.corr[0][l]; // radio
        sist.corr[1][l] += a.corr[1][l]; // correlacion
        sist.corr[2][l] += a.corr[2][l]; // desviacion
            correlacion
    }
    sist.vor += a.n_vortices;
}

double contador_corr = 0;
if ((*dat).n % salto_corr == 0)
    contador_corr = (*dat).n / salto_corr + 1;
else
    contador_corr = (*dat).n / salto_corr + 2;

for (int l = 0; l < a.n_cor; l++)
{
    sist.corr[0][l] = sist.corr[0][l] / contador_corr; // radio
    sist.corr[1][l] = sist.corr[1][l] / contador_corr; //
        correlacion
    sist.corr[2][l] = sist.corr[2][l] / contador_corr; //
        desviacion correlacion
}

sist.vor = sist.vor / contador_corr;
sist.vor = sist.vor * aux3;
sist.Em = sist.Em * pasos;
sist.tan = -fabs((sist.sl / sist.t1) - (2 / pi));
sist.Em2 = sist.Em2 * pasos;
sist.Mm = sist.Mm * pasos;
sist.Mm2 = sist.Mm2 * pasos;
sist.sl = sist.sl * pasos;
sist.Em2 = sist.Em2 - (sist.Em * sist.Em);
sist.Mm2 = sist.Mm2 - (sist.Mm * sist.Mm);
sist.cv = sist.Em2 * beta * beta * aux1;
sist.x = sist.Mm2 * beta * aux1;
sist.Em2 = sqrt(sist.Em2);
sist.Mm2 = sqrt(sist.Mm2);
sist.Em = sist.Em * aux1;
sist.Em2 = sist.Em2 * aux1;
sist.Mm = sist.Mm * aux1;
sist.Mm2 = sist.Mm2 * aux1;
sist.sl2 = sist.sl * aux1;

```

```

sist.sl = -0.5 * sist.Em - beta * sist.sl2;
sist.tan = fabs((sist.sl / sist.t1) - 0.636619772);
sist.tan = -sist.tan;

sist.a = a;

char dirDatos [] = "Datos";
char dirDiluido [200];
char dirLong [200];
char dirTemp [200];
mkdir(dirDatos, 0777);
sprintf(dirDiluido, "%s%s%03.10f", dirDatos, "/Diluido=", sist.
    a.diluido);
mkdir(dirDiluido, 0777);
sprintf(dirLong, "%s%s%i", dirDiluido, "/Longitud=", sist.a.nx)
    ;
mkdir(dirLong, 0777);
sprintf(dirTemp, "%s%s%03.10f", dirLong, "/Temp=", sist.t1);
mkdir(dirTemp, 0777);
calc_longitud_corr_mon(&sist, dirTemp);

sist.a = a;
guardar_sistema(dirTemp, sist);
guardar_Parametros(dirTemp, sist);
(*dat).sist = sist;
printf("termino diluido = %f Long= %i temp = %f \n", dat->
    sist.a.diluido, dat->sist.a.nx,
    (*dat).t1);
return arg;
}

void montecarlo(sis* a, int k, double* beta)
{

int i = a->ij[0][k];
int j = a->ij[1][k];
double cambio = 2.2 / (*beta);
double sAux = a->s[i][j] + 2 * cambio * r() - cambio;
double sxAux = cos(sAux);
double syAux = sin(sAux);

double fin = (cos((*a).s[i + 1][j] - sAux)) * (*a).sd[i + 1][j]
    + (cos((*a).s[i - 1][j] - sAux)) * (*a).sd[i - 1][j]
    + (cos((*a).s[i][j + 1] - sAux)) * (*a).sd[i][j + 1]

```

```

+ (cos((*a).s[i][j - 1] - sAux)) * (*a).sd[i][j - 1];

double inicio = (cos((*a).s[i + 1][j] - (*a).s[i][j])) * (*a).
sd[i + 1][j]
+ (cos((*a).s[i - 1][j] - (*a).s[i][j])) * (*a).sd[i - 1][j
]
+ (cos((*a).s[i][j + 1] - (*a).s[i][j])) * (*a).sd[i][j +
1]
+ (cos((*a).s[i][j - 1] - (*a).s[i][j])) * (*a).sd[i][j -
1];

double dE = -fin + inicio;
double dMx = (sxAux - (*a).sx[i][j]) * (*a).sd[i][j];
double dMy = (syAux - (*a).sy[i][j]) * (*a).sd[i][j];
double dsl = (sin(sAux - a->s[i + 1][j]) * (*a).sd[i + 1][j]
- sin(sAux - a->s[i - 1][j]) * (*a).sd[i - 1][j
])
- (sin((*a).s[i][j] - a->s[i + 1][j]) * (*a).sd[i + 1][j]
- sin((*a).s[i][j] - a->s[i - 1][j]) * (*a).sd
[i - 1][j]);

if (dE < 0)
{
(*a).E = (*a).E + dE;
(*a).Mx += dMx;
(*a).My += dMy;
a->sl += dsl;
(*a).s[i][j] = sAux;
(*a).sx[i][j] = sxAux;
(*a).sy[i][j] = syAux;
if (a->f[k])
cambiar_frontera(i, j, sAux, sxAux, syAux, a);
}
else if (r() < exp(-((*beta) * dE)))
{
(*a).E += dE;
(*a).Mx += dMx;
(*a).My += dMy;
a->sl += dsl;
(*a).s[i][j] = sAux;
(*a).sx[i][j] = sxAux;
(*a).sy[i][j] = syAux;
if (a->f[k])
cambiar_frontera(i, j, sAux, sxAux, syAux, a);
}
}
}

```

```

void calcular_corr(sis* a)
{
    double l = a->nx;
    if (l > (a->ny))
        l = a->ny;
    l = (l / 2) - 1;
    int n = a->ns;
    int* radioAux = malloc(a->nx * a->ny * sizeof(int));
    double* corrAux = malloc(a->nx * a->ny * sizeof(double));
    double* corr2Aux = malloc(a->nx * a->ny * sizeof(double));
    double* contAux = malloc(a->nx * a->ny * sizeof(double));
    int cont = 0;
    int i2, j2, i1, j1;
    double s1, s2;
    for (int i = 0; i < a->nx * a->ny; i++)
    {
        radioAux[i] = 0;
        corrAux[i] = 0;
        contAux[i] = 0;
        corr2Aux[i] = 0;
    }

    int distancia = 0;
    int maxDis = l * l;
    for (int i = 0; i < l; i++)
        for (int j = 0; j < l; j++)
        {
            distancia = i * i + j * j;
            if (distancia)
            {
                if (distancia < maxDis)
                    radioAux[distancia] += 1;
            }
        }

    for (int k1 = 0; k1 < n; k1++)
    {
        i1 = a->ij[0][a->indice[k1]];
        j1 = a->ij[1][a->indice[k1]];

        for (int k2 = 0; k2 < n; k2++)
        {

```

```

        i2 = a->ij[0][a->indice[k2]];
        j2 = a->ij[1][a->indice[k2]];

        if (!(i2 == i1 && j2 == j1))
        {

            s1 = a->s[i2][j2];
            s2 = a->s[i1][j1];
            distancia = (i2 - i1) * (i2 - i1) + (j2 - j1) * (j2
                - j1);
            if (distancia < maxDis)
            {
                corrAux[distancia] += cos(s1 - s2);
                corr2Aux[distancia] += cos(s1 - s2) * cos(s1 -
                    s2);
                contAux[distancia] += 1;
            }
        }
    }
}

cont = 0;

for (int i = 0; i < a->nx * a->ny; i++)
{
    if (radioAux[i])
    {
        a->corr[0][cont] = sqrt((double)i);
        a->corr[1][cont] = corrAux[i] / contAux[i];
        a->corr[2][cont] = corr2Aux[i] / contAux[i];
        cont++;
    }
}

a->n_cor = cont;
}

void ordenar(double* v, int n)
{
    int arc;
    for (int i = 0; i < n; i++)

```

```

        for (int j = i + 1; j < n; j++)
            if (v[i] > v[j])
            {
                arc = v[i];
                v[i] = v[j];
                v[j] = arc;
            }
    }

void regresion_lineal(double* x, double* y, double n, double* m,
    double* b)
{
    double sumax = 0;
    double sumay = 0;
    double sumaxy = 0;
    double sumax2 = 0;
    for (int i = 0; i < n; i++)
    {
        sumaxy += x[i] * y[i];
        sumax2 += x[i] * x[i];
        sumax += x[i];
        sumay += y[i];
    }
    *m = (n * sumaxy - sumax * sumay) / (n * sumax2 - sumax * sumax
        );
    *b = (sumay - *m * sumax) / n;
}

void calc_longitud_corr(sis* a, char* dir)
{
    double l = a->nx;
    if (l > (a->ny))
        l = a->ny;
    l = (l / 2) - 1;
    int n = a->n_cor;

    double Dat[2][n];
    for (int i = 0; i < n; i++)
    {
        if (fabs(a->corr[0][i]) != 0)
        {
            Dat[0][i] = fabs(a->corr[0][i]);
            Dat[1][i] = fabs(a->corr[1][i]);
        }
    }
}

```



```

        else
        {
            printf("Error Corr %f %f %f\n", fabs(a->corr[1][i]), a
                ->corr[0][i], a->corr[1][i]);
        }
    }
    // regresion polinomial
    double ap, bp;
    pot_reg(Dat[0], Dat[1], n, &ap, &bp);

    // regresion exp
    double ae, be;
    exp_reg(Dat[0], Dat[1], n, &ae, &be);

    a->regresion[0] = -bp;
    a->regresion[1] = -1 / be;
    a->regresion[2] = ap;
    a->regresion[3] = ae;
}

void calc_longitud_corr_mon(sis_montecarlo* sist, char* dir)
{
    int n = sist->n_cor;

    double Dat[2][n];
    for (int i = 0; i < n; i++)
    {
        if (fabs(sist->corr[0][i]) != 0)
        {
            Dat[0][i] = fabs(sist->corr[0][i]);
            Dat[1][i] = fabs(sist->corr[1][i]);
        }
        else
        {
            printf("Error Corr %f %f %f\n", fabs(sist->corr[1][i]),
                sist->corr[0][i],
                sist->corr[1][i]);
        }
    }
    // regresion polinomial
    double ap, bp;
    pot_reg(Dat[0], Dat[1], n, &ap, &bp);
}

```

```

// regresion exp
double ae, be;
exp_reg(Dat[0], Dat[1], n, &ae, &be);

char nombre[100];
sprintf(nombre, "%s%s%s", dir, "/", "ajusteCorr.dat");
FILE* archivo = fopen(nombre, "wb");
double r;
for (int i = 0; i < n; i++)
{
    r = sist->corr[0][i];
    fprintf(archivo, "%f %f %f %f\n", r, sist->corr[1][i], ap *
        pow(r, bp), ae * exp(be * r));
}

fclose(archivo);
sist->regresion[0] = -bp;
sist->regresion[1] = -1 / be;
sist->regresion[2] = ap;
sist->regresion[3] = ae;
}

void maximoMaximal(double* x, double* y, int n, double* maximo,
    double* maximal)
{
    *maximo = -10e200;
    *maximal = 0;
    for (int i = 0; i < n; i++)
    {
        if (y[i] > *maximo)
        {
            *maximo = y[i];
            *maximal = x[i];
        }
    }
}

void guardar_sistema(char* directorio, sis_montecarlo sist)
{
    char nombre[500];
    sprintf(nombre, "%s%s%s", directorio, "/", "ajusteCorr.dat");
    sprintf(nombre, "%s%s", directorio, "/s.dat");
}

```

```

guardar_matriz(nombre, sist.a.s, sist.a.nx + 2, sist.a.ny + 2);
sprintf(nombre, "%s%s", directorio, "/sx.dat");
guardar_matriz(nombre, sist.a.sx, sist.a.nx + 2, sist.a.ny + 2)
;
sprintf(nombre, "%s%s", directorio, "/sy.dat");
guardar_matriz(nombre, sist.a.sy, sist.a.nx + 2, sist.a.ny + 2)
;
sprintf(nombre, "%s%s", directorio, "/cor.dat");
sprintf(nombre, "%s%s", directorio, "/sd.dat");
guardar_matriz(nombre, sist.a.sd, sist.a.nx + 2, sist.a.ny +
2);
sprintf(nombre, "%s%s", directorio, "/xy.dat");
xyGuardar(nombre, &(sist.a));
guardar_vortice(directorio, sist.a);
}

void xyGuardar(char* nombre, sis* a)
{
FILE* archivo = fopen(nombre, "wb");
int i, j;
double x, y, dx, dy, fi;
for (i = 1; i < a->nx + 1; i++)
for (j = 1; j < a->ny + 1; j++)
if (a->sd[i][j])
{
fi = a->s[i][j];
dx = cos(fi);
dy = sin(fi);
dx = dx / 2;
dy = dy / 2;
x = i - (dx / 2);
y = j - (dy / 2);
fprintf(archivo, "%i %i %f %f %f %f %f \n", i, j, x
, y, dx, dy, fi);
}

fclose(archivo);
}

void contar_vortice(sis* a)
{
double f, angulo;
double error = 0.0001;

```

```

double error2 = 0.000001;
int cont = 0;
int completo = 0;

for (int i = 0; i < a->nx + 2; i++)
    for (int j = 0; j < a->ny + 2; j++)
    {
        f = a->s[i][j];
        a->s[i][j] = f - (floor(f / pi2)) * pi2;
    }
double t[4];
cont = 0;
for (int i = 1; i < a->nx + 1; i++)
    for (int j = 1; j < a->ny + 1; j++)
    {

        // 1 4
        // 2 3
        // se toma como 1 al ij
        completo = a->sd[i][j - 1] + a->sd[i][j] + a->sd[i +
            1][j - 1] + a->sd[i + 1][j];
        if (completo == 4)
        {

            t[0] = (a->s[i][j - 1] - a->s[i][j]); // 1->2
            t[1] = (a->s[i + 1][j - 1] - a->s[i][j - 1]); //
                2->3
            t[2] = (a->s[i + 1][j] - a->s[i + 1][j - 1]); //
                3->4
            t[3] = (a->s[i][j] - a->s[i + 1][j]); // 4->1

            for (int k = 0; k < 4; k++)
            {
                if (fabs(t[k]) > pi)
                {
                    if (t[k] > 0)
                        t[k] = t[k] - pi2;
                    else
                        t[k] = t[k] + pi2;
                }
            }
            angulo = t[0] + t[1] + t[2] + t[3];
            if (angulo > 0)
            {

```

```

        if (fabs(angulo - pi2) < error)
        {
            a->vortice[0][cont] = i + 0.5;
            a->vortice[1][cont] = j - 0.5;
            a->vortice[2][cont] = 1;
            cont++;
        }
    }
    else
    {

        if (fabs(-angulo - pi2) < error)
        {
            a->vortice[0][cont] = i + 0.5;
            a->vortice[1][cont] = j - 0.5;
            a->vortice[2][cont] = -1;
            cont++;
        }
    }
}

//+++++ Cuento Huecos
a->n_huecos = 0;
int iAux, jAux, jhAux, ihAux, i1, j1, i2, j2, ciclo, widin;
int nxy = 5 * (a->nx + a->ny);
int* ihves = malloc(nxy * sizeof(int));
int* jhves = malloc(nxy * sizeof(int));
short** utilizado;
double anguloAux;
utilizado = malloc((a->nx + 2) * sizeof(short*));
for (int i = 0; i < (a->nx + 2); i++)
{
    utilizado[i] = malloc((a->ny + 2) * sizeof(short));
    for (int j = 0; j < (a->ny + 2); j++)
        utilizado[i][j] = 0;
}
int veci = 0;
int nves = 0;
int ives[8] = { -1, -1, 0, 1, 1, 1, 0, -1 };
int jves[8] = { 0, 1, 1, 1, 0, -1, -1, -1 };
/*
876
1h5
234

```

```

h=(0,0)
*/
a->v_en_h = 0;
for (int j = 1; j < a->ny + 1; j++)
    for (int i = 1; i < a->nx + 1; i++)
    {
        salir:
            veci = 0;
            nves = 0;
            ihAux = i;
            jhAux = j;

            if (a->sd[i][j] == 0)
            {

                do
                {
                    if (utilizado[ihAux][jhAux])
                    {

                        i++;
                        goto salir;
                    }
                    i2 = ihAux + ives[veci];
                    j2 = jhAux + jves[veci];
                    periodico(&i2, &j2, a->nx, a->ny);
                    if (a->sd[i2][j2] == 0)
                    {
                        for (ciclo = 0; ciclo < 8; ciclo++)
                        {
                            i1 = i2 + ives[ciclo];
                            j1 = j2 + jves[ciclo];
                            if (i1 == ihves[nves - 1] && j1 ==
                                jhves[nves - 1])
                            {

                                break;
                            }
                        }
                    }

                    veci = ciclo + 1;
                    ihAux = i2;
                    jhAux = j2;

```

```

    }
    ihves[nves] = ihAux + ives[veci];
    jhves[nves] = jhAux + jves[veci];
    periodico(&(ihves[nves]), &(jhves[nves]), a->nx
        , a->ny);
    nves++;
    veci++;
    veci = veci % 8;

} while (
    (!(ihves[0] == ihves[nves - 1] && jhves[0] ==
        jhves[nves - 1])) || nves == 1);
nves--;
utilizado[i][j] = 1;
angulo = 0;
for (int k = 0; k < nves; k++)
{

    if (fabs(anguloAux) > pi)
    {
        if (anguloAux > 0)
            anguloAux = anguloAux - pi2;
        else
            anguloAux = anguloAux + pi2;
    }
    angulo += anguloAux;
}

if (angulo > 0)
{
    widin = floor((angulo + error2) / pi2);
    anguloAux = angulo - widin * pi2;
    if (fabs(anguloAux) < error && widin != 0)
    {
        a->vortice[0][cont] = i;
        a->vortice[1][cont] = j;
        a->vortice[2][cont] = widin;
        cont++;
        a->v_en_h++;
    }
}
else
{
    angulo = -angulo;
}

```

```

        widin = floor((angulo + error2) / pi2);
        anguloAux = angulo - widin * pi2;
        if (fabs(anguloAux) < error && widin != 0)
        {
            a->vortice[0][cont] = i;
            a->vortice[1][cont] = j;
            a->vortice[2][cont] = -widin;
            cont++;
            a->v_en_h++;
        }
    }
    a->n_huecos += 1;
}

a->n_vortices = cont;
}

void guardar_vortice(const char* dir, sis a)
{
    char nombre[100];
    sprintf(nombre, "%s%s%s", dir, "/", "vor_pos.dat");
    FILE* archivo1 = fopen(nombre, "wb");

    if (archivo1 == NULL)
    {
        printf("\nError\n");
        exit(EXIT_FAILURE);
    }

    sprintf(nombre, "%s%s%s", dir, "/", "vor_menos.dat");
    FILE* archivo2 = fopen(nombre, "wb");

    if (archivo2 == NULL)
    {
        printf("\nError\n");
        exit(EXIT_FAILURE);
    }

    for (int j = 0; j < a.n_vortices; j++)
    {
        if (a.vortice[2][j] > 0)

```



```

        fprintf(archivo1, "%f %f \n", a.vortice[0][j], a.
            vortice[1][j]);
    else
        fprintf(archivo2, "%f %f \n", a.vortice[0][j], a.
            vortice[1][j]);
}

sprintf(nombre, "%s%s%s", dir, "/", "huecos.dat");
FILE* archivo3 = fopen(nombre, "wb");

if (archivo3 == NULL)
{
    printf("\nError\n");
    exit(EXIT_FAILURE);
}

for (int j = 1; j <= a.ny; j++)
    for (int i = 1; i <= a.nx; i++)
        if (a.sd[i][j] == 0)
            fprintf(archivo3, "%i %i \n", i, j);
}

void periodico(int* i, int* j, int nx, int ny)
{
    if (*i < 0)
        *i = *i + 2 * nx;
    if (*j < 0)
        *j = *j + 2 * ny;
    if (*i > nx)
        *i = *i % nx;
    if (*j > ny)
        *j = *j % ny;
    if (*i == 0)
        *i = nx;
    if (*j == 0)
        *j = ny;
}

void guardar_Parametros(const char* dir, sis_montecarlo sist)
{
    char nombre[100];
    sprintf(nombre, "%s%s%s", dir, "/", "parametros.dat");
    FILE* archivo1 = fopen(nombre, "wb");

```

```

fprintf(archivo1, "%f\n", sist.Mm);
fprintf(archivo1, "%f\n", sist.Em);
fprintf(archivo1, "%f\n", sist.Mm2);
fprintf(archivo1, "%f\n", sist.Em2);
fprintf(archivo1, "%f\n", sist.cv);
fprintf(archivo1, "%f\n", sist.x);
fprintf(archivo1, "%i\n", sist.a.nx);
fprintf(archivo1, "%i\n", sist.a.ny);
fprintf(archivo1, "%i\n", sist.a.n_h);
fprintf(archivo1, "%i\n", sist.a.n_huecos);
fprintf(archivo1, "%i\n", sist.a.v_en_h);
fprintf(archivo1, "%f\n", ((double)sist.a.v_en_h) / ((double)
    sist.a.n_h));
fprintf(archivo1, "%f\n", sist.a.regrecion[0]);
fprintf(archivo1, "%f\n", sist.a.regrecion[2]);
fclose(archivo1);
}

void exp_reg(double* x, double* y, int n, double* a, double* b)
{
    int k = 0;
    double M, mf, db;
    double b_a = 0;

    do
    {
        M = 0;
        mf = 0;
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
            {
                M += (y[j] * x[i] - y[j] * x[j]) * exp(b_a * (x[j]
                    + 2 * x[i]));
                mf += (y[j] * x[i] - y[j] * x[j]) * (x[j] + 2 * x[i]
                    )
                    * exp(b_a * (x[j] + 2 * x[i]));
            }
        k++;
        if (mf == 0)
            mf = 1e-100;
        db = M / mf;
    }
}

```

```

        b_a = b_a - db;
    } while (k < 1000 && fabs(db) > 1e-10);

    *b = b_a;

    M = 0;
    mf = 0;
    for (int i = 0; i < n; i++)
    {
        M += y[i] * exp(b_a * x[i]);
        mf += exp(2 * b_a * x[i]);
    }
    *a = M / mf;
}

void pot_reg(double* x, double* y, int n, double* a, double* b)
{
    int k = 0;
    double M, mf, db;
    double b_a = -0;

    do
    {
        M = 0;
        mf = 0;
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
            {
                M += (y[j] * log(x[i]) - y[j] * log(x[j])) * pow(x[
                    j], b_a) * pow(x[i], 2 * b_a);
                mf += (y[j] * log(x[i]) - y[j] * log(x[j])) * pow(x
                    [j], b_a) * pow(x[i], 2 * b_a)
                    * (log(x[j]) + 2 * log(x[i]));
            }
        k++;
        if (mf == 0)
            mf = 1e-100;
        db = M / mf;

        b_a = b_a - db;
    } while (k < 1000 && fabs(db) > 1e-10);

    *b = b_a;
}

```

```

M = 0;
mf = 0;
for (int i = 0; i < n; i++)
{
    M += y[i] * pow(x[i], b_a);
    mf += pow(x[i], 2 * b_a);
}
*a = M / mf;
}

```

C.3. xy.h

```

#ifndef XY_H_INCLUDED
#define XY_H_INCLUDED

typedef struct
{
    int nx;
    int ny;
    int ns;
    int n_cor;
    int n_vortices;
    int n_huecos;
    int n_h;
    int v_en_h;

    double E;
    double Mx;
    double My;
    double diluido;
    double J;
    double B;
    double sl;

    int* indice;
    double* regrecion;

    double** s;
    double** sx;
    double** sy;
    double** corr;
}

```

```

    double** vortice;
    short** sd;
    short* f;

    int** ij;
} sis;

typedef struct
{
    sis a;
    int n;
    int nt;
    double Mm;
    double Em;
    double Em2;
    double Mm2;
    double cv;
    double x;
    double t1;
    double t2;
    double s1;
    double tan;
    double s12;
    double ** Mag_franja;
        double vor;

    int n_cor;
    double** corr;
    double * regrecion;
} sis_montecarlo;

typedef struct
{
    int n;
    int nt;
    double t1;
    double t2;
    int nx;

```

```

    int ny;
    double J;
    double B;
    int num_huecos;
    sis_montecarlo sist;
} datos;

double r();
double r2();
double ri();
double rci(double spin);
double rd();
int index2 (int i, int j , int nx);
sis crear_sis(int nx, int ny, double J,double B, double diluido);
void guardar_matriz ( const char * nombre , double ** matriz, int
    nx, int ny);
void barajar(int baraja[], unsigned int* n);
void calcular_sis( sis a, double J, double B);
void montecarlo(sis* a, int k, double* beta);
void abrir_matriz ( const char * nombre , double ** matriz);
void cambiar_frontera ( int i, int j , double sAux, double sxAux,
    double syAux, sis *a );
void * crear_sis_montecarlo ( void * arg );
void calcular_corr( sis *a );
void ordenar (double * A, int N);
void guardar_sistema (char * apodo , sis_montecarlo sist);
void regresion_lineal(double *x, double *y, double n, double *m ,
    double * b);
void calc_longitud_corr (sis * a, double * regresion, char *dir);
void maximoMaximal(double * x, double *y, int n , double* maximo,
    double *maximoMaximal);
void guardar_vortice (const char* dir, sis a);
void contar_vortice( sis* a );
#endif // XY_H_INCLUDED

```

Bibliografía

- [1] Ralph Kenna. The xy model and the berezinskii-kosterlitz-thouless phase transition. *Arxiv*, 2005.
- [2] P. W. Anderson. *Basic notions of condensed matter physics*. Frontiers in physics 55. Benjamin/Cummings Pub. Co., Advanced Book Program, 1984.
- [3] Franz Wegner. Spin-ordering in a planar classical heisenberg model. *Zeitschrift für Physik*, 1967.
- [4] B.V. Costa, L.S. Lima, P. Z. Coura, S. A. Leonel, and A.B. Lima. Kosterlitz-thouless transition: The diluted xy model. *J. Phys.: Conf.*, 2014.
- [5] N. D. MERMIN and H. WAGNER. Spin-ordering in a planar classical heisenberg model. *Phys. Rev. Letters*, 1966.
- [6] Tobias Rautenkranz Alexander Käser, Tobias Maier. The 2d xy model, July 2007.
- [7] A. B. Harris and T. C. Lubensky. Randomly diluted xy and resistor networks near the percolation threshold. *PHYSICAL REVIEW B*, 1987.
- [8] MICHEL LE BELLAC, FABRICE MORTESSAGNE, and G. GEORGE BATROUNI. *EQUILIBRIUM AND NON-EQUILIBRIUM STATISTICAL THERMODYNAMICS*. Cambridge University Press, New York, 2004.
- [9] Sergio Alejandro Cannas. *Notas de Mecánica Estadística*. Universidad Nacional de Córdoba, 2017.
- [10] María Paula Hernández Pascual. *COMPORTAMIENTO CRÍTICO DE FLÚIDOS: PROPIEDADES ESTÁTICAS Y DINÁMICAS*. PhD thesis, Universidad de Complutense de Madrid, 2004.
- [11] C. Kittel. *Introduction to Solid State Physics*. Wiley, 2004.

- [12] Harvey Gould and Jan Tobochnik. *Thermal and Statistical Physics*. Princeton University Press., New York, 2015.
- [13] Rondey J. Baxter. *Exactly solverd models in statistical mechanics*. Academic Press, 1989.
- [14] Giuseppe Grosso. *Solid State Physics*. ELSEVIER, 2014.
- [15] Duncan M. Haldane. Topological quantum matter. *Department of Physics, Princeton University*, 2016.
- [16] N. D. Mermin. The topological theory of defects in ordered media. *Reviews of Modern Physics*, 1979.
- [17] Jakub Imriška. Phase diagram of a modified xy model, 2004.
- [18] J M Kosterlitz. The critical properties of the two-dimensional xy model. *Journal of Physics C: Solid State Physics*, 7(6):1046, 1974.
- [19] Juan José Toledo Garrido. La transición de fase de kosterlitzthouless, 2017.
- [20] A.R. Pereira L.A.S. Mól B.V. Costa S.A. Leonel, P. Zimmermann Coura. Kosterlitz-thouless transition: The diluted xy model. *J. Phys*, 2014.
- [21] Fariñas-Sánchez A.I. Holovatch Y. et al. Berche, B. Influence of quenched dilution on the quasi-long-range ordered phase of the 2d xy model. *Eur. Phys. J. B*, 2014.
- [22] Sergio Alejandro Cannas. *Simulaciones Numéricas. Método de Monte Carlo*. Universidad Nacional de Córdoba, 2017.
- [23] Marisol Santana Anaya. Técnicas de reducción de varianza para el método monte carlo aplicado a opciones financieras, Agosto 2013.
- [24] H. E. Stanley. *Introduction to Phase Transitions and Critical Phenomena*. Oxford University Press, Oxford and New York, 1971.
- [25] Horst Stöcker D. Rischke Walter Greiner, Ludwig Neise. *Thermodynamics and statistical mechanics*. Springer, New York, 1995.
- [26] Heinz-Otto et al Peitgen. *Chaos and Fractals: New Frontiers of Science*. Springer-Verlag, 1992.