



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ

Ανάλυση Δεδομένων Monte Carlo του 2D-Ising  
προτύπου με τη μέθοδο Multiple Histogram

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΙΟΣ Σ. ΜΠΑΧΤΗΣ

Επιβλέπων: Αναγνωστόπουλος Κωνσταντίνος  
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών  
Τομέας Φυσικής

## Ανάλυση Δεδομένων Monte Carlo του 2D-Ising προτύπου με τη μέθοδο Multiple Histogram

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΙΟΣ Σ. ΜΠΑΧΤΗΣ

Επιβλέπων: Αναγνωστόπουλος Κωνσταντίνος  
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10η Οκτωβρίου 2016.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Αναγνωστόπουλος Κωνσταντίνος  
Αν. Καθηγητής Ε.Μ.Π.

.....  
Ήργες Νίκος  
Αν.Καθηγητής Ε.Μ.Π.

.....  
Κουτσούμπας Γεώργιος  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016

*(Υπογραφή)*

.....  
**ΔΗΜΗΤΡΙΟΣ ΜΠΑΧΤΗΣ**

Διπλωματούχος Φυσικός Εφαρμογών Ε.Μ.Π.

© 2016 – All rights reserved



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών  
Τομέας Φυσικής

Copyright ©–All rights reserved Δημήτριος Μπαχτής, 2016.

Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.



# Περίληψη

Αντικείμενο της διπλωματικής εργασίας είναι η υλοποίηση της μεθόδου Multiple Histogram των Alan M. Ferrenberg και Robert H. Swendsen στο δισδιάστατο πρότυπο Ising. Η Multiple Histogram εκμεταλλεύεται τη διαθεσιμότητα δεδομένων από πολλαπλές προσομοιώσεις Monte Carlo που έχουν εκτελεσθεί σε διαφορετικές αλλά κοντινές τιμές μιας μεταβλητής του συστήματος. Γίνεται τότε δυνατός ο υπολογισμός των εκτιμητών των παρατηρήσιμων ποσοτήτων με επέκταση ή παρεμβολή στη μεταβλητή αυτή για ένα επιτρεπτό εύρος.

Η μέθοδος εφαρμόζεται στο σιδηρομαγνητικό δισδιάστατο πρότυπο Ising με παρεμβολή στη μεταβλητή της θερμοκρασίας. Αρχικά με χρήση των διαθέσιμων προσομοιώσεων μπορούν να υπολογιστούν οι συναρτήσεις επιμερισμού τους. Με δεδομένες αυτές γίνεται δυνατή η εκτίμηση των συναρτήσεων επιμερισμού με παρεμβολή στη θερμοκρασία για όλο το εύρος τιμών ανάμεσα σε αυτές των αρχικών. Επειδή η αναμενόμενη τιμή κάθε παρατηρήσιμης ποσότητας του προτύπου μπορεί να υπολογιστεί σε όρους της συνάρτησης επιμερισμού, ακολουθεί ο υπολογισμός της ελεύθερης ενέργειας, της εσωτερικής ενέργειας, της ειδικής θερμότητας, της μαγνήτισης και της μαγνητικής επιδεκτικότητας.

Η μέθοδος επιτρέπει την ευκολότερη μελέτη των κρίσιμων φαινομένων που χαρακτηρίζουν τη μετάβαση φάσης του προτύπου. Με τη δυνατότητα παρεμβολής σε ένα μεγάλο εύρος μπορούν να εντοπιστούν οι ψευδοκρίσιμες θερμοκρασίες για τις οποίες εμφανίζονται τα μέγιστα της μαγνητικής επιδεκτικότητας και της ειδικής θερμότητας. Εφαρμόζεται τότε η βάθμιση πεπερασμένου μεγέθους για τον υπολογισμό των κρίσιμων εκθετών και της κρίσιμης θερμοκρασίας του προτύπου.

Οι εκτιμήσεις σφαλμάτων των δεδομένων γίνονται με τη μέθοδο Blocking/Binning. Παρατίθεται για πληρότητα και η μέθοδος Bootstrap η οποία όμως δεν είναι καθόλου οικονομική από άποψη ταχύτητας και η χρήση της θα είχε νόημα στη συγκεκριμένη μέθοδο μόνο αν είχε υλοποιηθεί στα πλαίσια παράλληλου προγραμματισμού.

Συνοψίζοντας, η μέθοδος προσφέρει ακριβείς υπολογισμούς με ελαχιστοποίηση των σφαλμάτων. Παρατηρώντας ότι είναι αρκετά ταχύτερη συγκριτικά με την εκτέλεση μιας νέας προσομοίωσης Monte Carlo σε κάποια θερμοκρασία, η χρησιμότητα της διαφαίνεται καθώς το μέγεθος των πλεγμάτων αυξάνει. Μπορεί επίσης να χρησιμοποιηθεί σε πλέγματα μικρού μεγέθους για την πρόβλεψη συμπεριφοράς του μοντέλου σε μεγάλα εύρη θερμοκρασιών. Στο παράρτημα παρατίθεται ο κώδικας υλοποίησης της μεθόδου σε C.





# Abstract

The main purpose of this diploma thesis is the implementation of the Multiple Histogram method of Alan M. Ferrenberg and Robert H. Swendsen in the 2D-Ising Model. Multiple Histogram exploits the availability of data from Monte Carlo simulations that have been executed at different but adequately close values of a specific variable. It is then possible to calculate the estimators of observable quantities by extrapolating or interpolating in that variable for a limited range.

The method is applied in the ferromagnetic 2D-Ising Model by interpolating in the variable of temperature. At first the available simulations can be used to achieve an estimation of their partition functions. Given these one can interpolate in temperatures that lie between those of the initial simulations to estimate the partition functions of the whole range. Since the expectation value of any observable quantity of the model can be given in terms of the partition function, it is simple to calculate the estimators for the free energy, internal energy, specific heat, magnetization and magnetic susceptibility.

Multiple histogram allows for a better study of the phase transition of the model. Considering it is possible to interpolate in a whole range with great accuracy, it is much faster, especially for large lattices, to pinpoint the pseudo-critical temperatures where the maximum values of magnetic susceptibility and specific heat appear. The finite size scaling method can then be used for the calculation of the critical exponents and the critical temperature of the model.

The error estimation of the data has been done with the Blocking/Binning method. The Bootstrap method has also been developed for generality but its use is not advised because it is computationally slow. A parallel computing approach for the bootstrap method would have been ideal.

Overall multiple histogram gives great results with minimum errors. Considering it is much faster than executing a Monte Carlo simulation at any temperature it really shines as the size of lattices becomes larger. It can also be used in small sized lattices to predict the behaviour of the model in an extremely huge range. In the appendix the code of the method is given in C.



# Ευχαριστίες

Ολοκληρώνοντας τις σπουδές μου θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή Κωνσταντίνο Αναγνωστόπουλο για την πολύτιμη βοήθεια, τις γνώσεις φυσικής αλλά και προγραμματισμού που μου μετέφερε καθώς και την καθαρότητα σκέψης με την οποία απαντούσε σε όλες μου τις ερωτήσεις. Επίσης για την εξαιρετική διδασκαλία των μαθημάτων Υπολογιστική Φυσική I και II που αποτέλεσαν το λόγο έναρξης της δικής μου ενασχόλησης με το συναρπαστικό αυτό τομέα.

Επιπλέον θα ήθελα να ευχαριστήσω ιδιαίτερα τους καθηγητές Ήργε Νίκο και Κουτσούμπα Γεώργιο για τη συμμετοχή τους στην τριμελή επιτροπή.

Τέλος την οικογένειά μου, για την ατέρμονη υποστήριξη τους και την παρότρυνση να ασχοληθώ με ότι με γεμίζει και αγαπώ.



# Περιεχόμενα

Περίληψη	i
Abstract	iii
Ευχαριστίες	v
Περιεχόμενα	viii
Κατάλογος Σχημάτων	xi
Κατάλογος Πινάκων	xiii
<b>1 Αρχές Στατιστικής Φυσικής</b>	<b>1</b>
1.1 Θερμοδυναμική και Στατιστική Φυσική . . . . .	1
1.2 Συστήματα σε Ισορροπία . . . . .	2
1.3 Διακυμάνσεις, Αποκρίσεις και Συναρτήσεις Συσχετισμού . . . . .	4
<b>2 Αρχές Monte Carlo Στατιστικής Φυσικής</b>	<b>7</b>
2.1 Ο Εκτιμητής της Παρατηρήσιμης Ποσότητας . . . . .	7
2.2 Δειγματοληψία με Κριτήριο Σημαντικότητας . . . . .	8
2.2.1 Διαδικασίες Markov . . . . .	8
2.2.2 Η Συνθήκη της Εργοδικότητας . . . . .	9
2.2.3 Η Συνθήκη της Λεπτομερούς Ισορροπίας . . . . .	9
2.3 Λόγοι Αποδοχής . . . . .	9
<b>3 Το Πρότυπο Ising</b>	<b>11</b>
3.1 Εισαγωγή . . . . .	11
3.2 Ο Αλγόριθμος Metropolis . . . . .	12
3.2.1 Περιγραφή του Αλγορίθμου Metropolis . . . . .	12
3.3 Θερμοδυναμική Ισορροπία . . . . .	13
3.4 Υπολογισμός Ποσοτήτων της Προσομοίωσης . . . . .	14
3.4.1 Συναρτήσεις Αυτοσυσχετισμού . . . . .	15
3.5 Η Μετάβαση Φάσης . . . . .	16

3.6	Κρίσιμες Διακυμάνσεις και Κρίσιμη Επιβράδυνση . . . . .	17
<b>4</b>	<b>Κρίσιμοι Εκθέτες και ο Αλγόριθμος Wolff</b>	<b>19</b>
4.1	Κρίσιμοι Εκθέτες . . . . .	19
4.2	Ο Αλγόριθμος Wolff . . . . .	20
4.2.1	Λόγος Αποδοχής του Αλγόριθμου Συμπλεγμάτων . . . . .	21
4.3	Ιδιότητες του Αλγορίθμου Wolff . . . . .	23
4.3.1	Ο Χρόνος Συσχετισμού και ο Δυναμικός Εκθέτης . . . . .	24
4.3.2	Ο Δυναμικός Εκθέτης και η Επιδεκτικότητα . . . . .	24
<b>5</b>	<b>Ανάλυση Δεδομένων</b>	<b>27</b>
5.1	Εισαγωγή . . . . .	27
5.2	Υπολογισμός Σφαλμάτων . . . . .	27
5.2.1	Εκτιμήσεις Στατιστικών Σφαλμάτων . . . . .	27
5.2.2	Η μέθοδος Blocking . . . . .	28
5.2.3	Η μέθοδος Bootstrap . . . . .	29
5.2.4	Η μέθοδος Jackknife . . . . .	29
5.2.5	Συστηματικά Σφάλματα . . . . .	30
5.3	Η Μέθοδος Single Histogram . . . . .	30
5.3.1	Επέκταση σε Διαφορετικές Μεταβλητές . . . . .	33
5.4	Η Μέθοδος Multiple Histogram . . . . .	37
5.5	Κρίσιμα Φαινόμενα, Παγχοσιμότητα και Βάθμιση . . . . .	48
	<b>Βιβλιογραφία</b>	<b>55</b>
	<b>Παράρτημα: Κώδικας</b>	<b>57</b>

# Κατάλογος Σχημάτων

3.1	Διαδικασία εύρεσης της θερμικής ισορροπίας. Κάθε χρώμα αντιστοιχεί στη μαγνήτιση (πάνω) και εσωτερική ενέργεια (κάτω) ανά πλεγματοειδή θέση της ίδιας προσομοίωσης για το πρότυπο Ising με $L = 40$ και $\beta = 0.50$ . Έχουν επιλεγεί δύο αρχικές καταστάσεις θερμές ( $\beta = 0$ ) και μια παγωμένη ( $\beta = \infty$ ). Οι δύο θερμές καταστάσεις ακολουθούν διαφορετική στοχαστική εξέλιξη. Το διάγραμμα έχει ως στόχο να δείξει ότι μια διαφορετική στοχαστική εξέλιξη για τις ίδιες ιδιότητες μιας προσομοίωσης μπορεί να οδηγήσει το σύστημα μετά από διαφορετικό χρόνο σε ισορροπία. Προφανώς τα 1000 βήματα δεν είναι αρκετά για να θεωρήσουμε ότι έχει επιτευχθεί θερμική ισορροπία. . . . .	14
3.2	Διατάξεις των spin για το πρότυπο Ising με $L = 400$ μετά από 5000, 20000, 40000 βήματα αντίστοιχα με τον αλγόριθμο Metropolis. Παρατηρείται η δημιουργία συμπλεγμάτων μεγάλου μεγέθους όπου δεν είναι δυνατό να καταστραφούν και να δημιουργηθούν αλλού με ένα αλγόριθμο αλλαγής μοναδικών spin ανά βήμα. Για το λόγο αυτό οι χρόνοι αυτοσυσχετισμού είναι μεγάλοι. . . . .	18
5.1	Εφαρμογή της μεθόδου single histogram(ευθεία σχήματος) για πλέγμα μεγέθους $N = L * L = 40 * 40 = 1600$ και σύγκριση με δεδομένα που προκύπτουν από απευθείας υπολογισμούς με εκτελέσεις προσομοιώσεων Monte Carlo (σχήμα ο). Ο συμπαγής κύκλος δηλώνει την προσομοίωση που χρησιμοποιήθηκε για την επέκταση (extrapolation) των δεδομένων. Οι δύο κάθετες ευθείες είναι το επιτρεπτό εύρος επέκτασης όπως δίνεται μέσω της σχέσης (5.3.10). Παρατηρούμε ότι η μέθοδος χάνει την αξιοπιστία της όσο μεγαλώνει η απόσταση. . . . .	35
5.2	Ανάλογο σχήμα με το 5.1 για $N = L * L = 80 * 80 = 6400$ . Παρατηρούμε ότι με την αύξηση του μεγέθους του πλέγματος το επιτρεπτό εύρος επέκτασης μειώνεται, κάτι που είναι αναμενόμενο λόγω της μεταβολής της τιμής της ειδικής θερμότητας για μεγαλύτερα πλέγματα και της σχέσης (5.3.10). . . . .	36
5.3	Ιστογράμματα $N(E)$ των ενεργειών για το διδιάστατο πρότυπο Ising σε πλέγμα $N = L * L = 20 * 20 = 400$ για θερμοκρασίες(από δεξιά προς αριστερά) $\beta = 0.40, 0.42, 0.44, 0.46, 0.48$ . Είναι εμφανής η αλληλοεπικάλυψη των ιστογραμμάτων η οποία αποτελεί απαραίτητη προϋπόθεση για την εξαγωγή σωστών αποτελεσμάτων με τη μέθοδο multiple histogram. . . . .	39

- 5.4 Εφαρμογή της μεθόδου multiple histogram (ευθεία σχήματος) για πλέγμα μεγέθους  $N = L * L = 20 * 20 = 400$  και σύγκριση με δεδομένα που προκύπτουν από απευθείας υπολογισμούς με εκτελέσεις προσομοιώσεων Monte Carlo (σχήμα ο). Αναπαρίσταται η ενέργεια  $\langle e \rangle$  στο άνω σχήμα και η ειδική θερμότητα  $c$  στο κάτω σχήμα. Οι συμπαγείς κύκλοι (●) δηλώνουν τις προσομοιώσεις που χρησιμοποιήθηκαν για την εφαρμογή της μεθόδου και την παρεμβολή (interpolation) στη μεταβλητή της θερμοκρασίας. Η συγκεκριμένη εκτέλεση αναδεικνύει την δυνατότητα χρήσης της μεθόδου με στρατηγική επιλογή προσομοιώσεων για παρεμβολή σε μεγάλο εύρος. Για το συγκεκριμένο μέγεθος πλέγματος εφόσον υπάρχει μεγάλη αλληλοεπικάλυψη στα ιστογράμματα ενεργειών το δυνατό εύρος επέκτασης είναι κατά πολύ μεγαλύτερο από το εικονιζόμενο. . . . . 42
- 5.5 Αναπαρίσταται η μαγνήτιση  $\langle m \rangle$  στο πάνω σχήμα και η μαγνητική επιδεκτικότητα  $\chi$  στο κάτω σχήμα για την ίδια περίπτωση πλέγματος του σχήματος 5.4. . . . . 43
- 5.6 Εφαρμογή της μεθόδου multiple histogram η οποία αναπαρίσταται ως η ενδιάμεση ευθεία. Επίσης έχουν σχεδιαστεί τα σφάλματα όπως υπολογίστηκαν με τη μέθοδο blocking ως δύο ευθείες για καλύτερη παρουσίαση. Το πλέγμα στο πάνω σχήμα είναι μεγέθους  $N = L * L = 80 * 80 = 6400$  και στο κάτω σχήμα  $N = L * L = 140 * 140 = 19600$ . Τα δεδομένα που προκύπτουν από απευθείας υπολογισμούς σε κάθε περίπτωση με εκτελέσεις προσομοιώσεων Monte Carlo αναπαρίστανται στο σχήμα ως κύκλοι (σχήμα ο). Τα σφάλματα από τη μέθοδο jackknife για τα προσομοιωμένα δεδομένα είναι πολύ μικρά και δεν εμφανίζονται στο σχήμα. Οι συμπαγείς κύκλοι (●) δηλώνουν τις προσομοιώσεις που χρησιμοποιήθηκαν για την εφαρμογή της μεθόδου και την παρεμβολή (interpolation) στη μεταβλητή της θερμοκρασίας. Η συγκεκριμένη εκτέλεση αναδεικνύει τον τρόπο μελέτης πλεγμάτων μεγαλύτερου μεγέθους όπου το εύρος παρεμβολής είναι μικρότερο. Παρατηρούμε ότι οι τιμές βρίσκονται εντός ορίων σφαλμάτων της multiple histogram. . . . . 44
- 5.7 Αναπαρίσταται η ειδική θερμότητα  $c$  για μέγεθος πλέγματος  $N = L * L = 80 * 80 = 6400$  για την περίπτωση  $L = 80$  του σχήματος 5.6. Για τα σφάλματα των δεδομένων της multiple histogram έχει χρησιμοποιηθεί η μέθοδος blocking ενώ για αυτά των προσομοιωμένων δεδομένων η μέθοδος jackknife. Παρατηρούμε ότι οι τιμές βρίσκονται εντός ορίων των σφαλμάτων. . . . . 45
- 5.8 Αναπαρίσταται η μαγνήτιση  $\langle m \rangle$  για μέγεθος πλέγματος  $N = L * L = 80 * 80 = 6400$  και  $N = L * L = 140 * 140 = 19600$  στο πάνω και κάτω διάγραμμα αντίστοιχα για τις ίδιες περιπτώσεις του σχήματος 5.6. . . . . 46
- 5.9 Αναπαρίσταται η μαγνητική επιδεκτικότητα  $\chi$  για μέγεθος πλέγματος  $N = L * L = 80 * 80 = 6400$  και  $N = L * L = 140 * 140 = 19600$  στο πάνω και κάτω διάγραμμα αντίστοιχα για τις ίδιες περιπτώσεις του σχήματος 5.6. . . . . 47



- 5.10 Προσδιορισμός της κρίσιμης θερμοκρασίας  $\beta_c$  και του κρίσιμου εκθέτη  $\nu$  από τη σχέση (5.5.12). Η προσαρμογή έχει γίνει στη συνάρτηση  $a - c(1/L)^b$  όπου  $b = 1/\nu$  και  $a = \beta_c$ . Η τομή της ευθείας με τον άξονα  $y$  δίνει την ακριβή τιμή της κρίσιμης θερμοκρασίας  $\beta_c = \log(1 + \sqrt{2})/2 = 0.4407$ . . . . . 52
- 5.11 Προσδιορισμός της κρίσιμης θερμοκρασίας  $\beta'_c$  και του κρίσιμου εκθέτη  $\nu$  από τη σχέση (5.5.12). Η προσαρμογή έχει γίνει στη συνάρτηση  $a - c(1/L)^b$  όπου  $b = 1/\nu$  και  $a = \beta_c$ . Η τομή της ευθείας με τον άξονα  $y$  δίνει την ακριβή τιμή της κρίσιμης θερμοκρασίας  $\beta_c = \log(1 + \sqrt{2})/2 = 0.4407$ . . . . . 52
- 5.12 Υπολογισμός του κρίσιμου εκθέτη  $\gamma/\nu$  από τα μέγιστα της μαγνητικής επιδεκτικότητας  $\chi_{max}$ . Γίνεται προσαρμογή σε μια συνάρτηση  $\chi_{max} = aL^b$ . Το αποτέλεσμα της προσαρμογής δίνει  $\gamma/\nu = 1.751 \pm 0.003$  . . . . . 53
- 5.13 Υπολογισμός του κρίσιμου εκθέτη  $\alpha/\nu$  από τα μέγιστα της ειδικής θερμότητας. Γίνεται προσαρμογή σε μια συνάρτηση  $c_{max} = a \log L + b - c/L$ . Το αποτέλεσμα της προσαρμογής δίνει  $\alpha = 0.10 \pm 0.01$ ,  $b = 0.19 \pm 0.05$  και  $c = 2.2 \pm 0.9$ . Για προσαρμογή σε συνάρτηση  $c_{max} = aL^d + b - c/L$  παίρνουμε  $d = 0.046 \pm 0.65$  δηλαδή εκθέτη συμβατό με την τιμή μηδέν. Τα δεδομένα είναι συμβατά με τον εκθέτη  $\alpha/\nu$  να είναι μηδέν. . . . . 53



# Κατάλογος Πινάκων

5.1	Υπολογισμός των ψευδοκρίσιμων θερμοκρασιών $\beta_c(L)$ από τα μέγιστα της μαγνητικής επιδεκτικότητας $\chi_{max}$ για διάφορα μεγέθη πλέγματος. Χρησιμοποιήθηκε η μέθοδος Multiple Histogram με χρήση δύο προσομοιώσεων για τον εντοπισμό της περιοχής που εμφανίζονται τα μέγιστα. Μετά εκτελέστηκε με βήμα $10^{-5}$ για την αναζήτηση της ψευδοκρίσιμης θερμοκρασίας. Δίνονται επίσης και αντίστοιχες τιμές των μεγίστων. Σημειώνεται ότι η μέθοδος Blocking εφαρμόζεται σε αυτή την περίπτωση για τον υπολογισμό των σφαλμάτων των ποσοτήτων. . . . .	51
5.2	Πίνακας ανάλογος του 5.1 για τις ψευδοκρίσιμες θερμοκρασίες $\beta'_c(L)$ από τα μέγιστα της ειδικής θερμότητας $c_{max}$ για διάφορα μεγέθη πλέγματος. . . . .	51



# Κεφάλαιο 1

## Αρχές Στατιστικής Φυσικής

### 1.1 Θερμοδυναμική και Στατιστική Φυσική

Η στατιστική φυσική μελετά συστήματα με πολύ μεγάλο αριθμό βαθμών ελευθερίας. Η κύρια δυσκολία αντιμετώπισης τέτοιων συστημάτων εντοπίζεται στο ότι αποτελούνται από πάρα πολλά μέρη τα οποία διέπονται μεν από απλές εξισώσεις αλλά ο αριθμός αυτών των εξισώσεων κάνει την αναλυτική επίλυση απαγορευτική. Η μελέτη αυτή πραγματοποιείται μέσω δύο τρόπων προσέγγισης. Αρχικά μέσω της κλασικής θερμοδυναμικής με φαινομενολογικούς νόμους που περιγράφουν τις μακροσκοπικές ιδιότητες του συστήματος και οι οποίοι δικαιώνονται από την εφαρμογή τους και κατά δεύτερον μέσω της στατιστικής φυσικής όπου αντιμετωπίζει το σύστημα πιθανολογικά και προσπαθεί να εξάγει τις μακροσκοπικές ιδιότητες από το μικροσκοπικό επίπεδο.

Το πρότυπο που θα μελετηθεί σε αυτή τη διπλωματική και για το οποίο θα εφαρμοστεί η μέθοδος multiple histogram περιγράφεται από μια χαμιλτονιανή  $H$  που δίνει την ολική ενέργεια του συστήματος όταν αυτό βρίσκεται σε μια δεδομένη κατάσταση. Επίσης βρίσκεται σε επαφή με μία δεξαμενή η οποία αποτελεί ένα εξωτερικό σύστημα το οποίο ανταλλάσσει διαρκώς θερμότητα με το πρότυπο. Ουσιαστικά η δεξαμενή είναι μια ασθενής διαταραχή στη χαμιλτονιανή, όπου αγνοείται για τον υπολογισμό των επιπέδων ενέργειας αλλά οδηγεί το σύστημα σε μεταβάσεις διακριτών καταστάσεων. Στο κεφάλαιο αυτό και πριν παρουσιαστεί το πρότυπο Ising θα εξετασθούν κάποιες έννοιες οι οποίες έχουν γενική ισχύ.

Θεωρούμε ότι το σύστημα βρίσκεται σε μια κατάσταση  $\mu$ . Ορίζεται ως  $R(\mu \rightarrow \nu)dt$  η πιθανότητα μετάβασης του συστήματος σε μια κατάσταση  $\nu$  σε χρόνο  $dt$ . Για κάθε πιθανή κατάσταση  $\nu$  του συστήματος μπορεί να οριστεί ένας ρυθμός μετάβασης  $R(\mu \rightarrow \nu)$ , και αυτοί οι ρυθμοί συνήθως αποτελούν τη μοναδική πληροφορία για τη δυναμική του συστήματος. Η πιθανολογική προσέγγιση του προβλήματος γίνεται με τον ορισμό των βαρών  $w_\mu(t)$  που εκφράζουν την πιθανότητα το σύστημα να βρίσκεται στην κατάσταση  $\mu$  στο χρόνο  $t$  και τα οποία αποτελούν την κωδικοποίηση της μικροσκοπικής φυσικής στη στατιστική φυσική.

Μπορούμε να γράψουμε μια δεσπόζουσα εξίσωση για την εξέλιξη των  $w_\mu(t)$  συναρτήσει

των ρυθμών  $R(\mu \rightarrow \nu)$ :

$$\frac{dw_\mu(t)}{dt} = \sum_\nu \left( w_\nu(t)R(\nu \rightarrow \mu) - w_\mu(t)R(\mu \rightarrow \nu) \right) \quad (1.1.1)$$

Ο πρώτος όρος στο δεύτερο μέλος εκφράζει το ρυθμό με τον οποίο το σύστημα μεταβαίνει στην κατάσταση  $\mu$  και ο δεύτερος όρος εκφράζει το ρυθμό με τον οποίο το σύστημα μεταβαίνει από την κατάσταση  $\mu$  σε άλλες καταστάσεις.

Τα βάρη  $w_\mu(t)$  υπακούουν επίσης στη σχέση:

$$\sum_\mu w_\mu(t) = 1 \quad (1.1.2)$$

για όλους τους χρόνους  $t$ , εφόσον το σύστημα οφείλει να βρίσκεται πάντα σε μια κατάσταση. Η επίλυση της εξίσωσης (1.1.1) υπό τον περιορισμό (1.1.2) εκφράζει τη μεταβολή των βαρών στο χρόνο.

Τα βάρη  $w_\mu$  σχετίζονται με τις μακροσκοπικές ιδιότητες του συστήματος σύμφωνα με την σχέση της αναμενόμενης τιμής μιας ποσότητας  $Q$  η οποία παίρνει την τιμή  $Q_\mu$  για κατάσταση  $\mu$  σε χρόνο  $t$ :

$$\langle Q \rangle = \sum_\mu Q_\mu w_\mu(t) \quad (1.1.3)$$

Η παραπάνω εξίσωση δίνει σημαντική πληροφορία για την πραγματική τιμή της  $Q$  που αναμένεται να μετρηθεί σε ένα πραγματικό πείραμα.

## 1.2 Συστήματα σε Ισορροπία

Σύμφωνα με την εξίσωση (1.1.1) αν το σύστημα βρεθεί σε μία δεδομένη κατάσταση για την οποία αλληλοαναιρούνται οι όροι στο δεξί μέλος για όλες τις καταστάσεις  $\mu$ , τότε οι ρυθμοί  $\frac{dw_\mu(t)}{dt}$  μηδενίζονται και τα βάρη  $w_\mu(t)$  παίρνουν σταθερές τιμές για κάθε επόμενη χρονική στιγμή. Μια τέτοια κατάσταση ονομάζεται κατάσταση ισορροπίας. Οι ρυθμοί μετάβασης  $R(\mu \rightarrow \nu)$  θεωρούνται ανεξάρτητοι του χρόνου και προσομοιώνονται με κατάλληλες επιλογές κατά τη διάρκεια των υπολογισμών Monte Carlo. Το σύστημα για τα βάρη  $w_\mu(t)$  είναι γραμμικό και οι τιμές τους είναι περιορισμένες στο διάστημα  $[0, 1]$ . Άρα από τα παραπάνω μπορούμε να αποφανθούμε ότι όλα τα συστήματα τα οποία περιγράφονται από αυτές τις εξισώσεις καταλήγουν σε ισορροπία. Ορίζονται λοιπόν οι πιθανότητες κατάληψης ισορροπίας για τις οποίες:

$$p_\mu = \lim_{t \rightarrow \infty} w_\mu(t) \quad \sum_\mu p_\mu = 1 \quad (1.2.1)$$

Για τις πιθανότητες  $p_\mu$  για ένα σύστημα σε θερμοκή ισορροπία με δεξαμενή θερμοκρασίας  $\beta = 1/\kappa T$ ,  $\kappa = 1.38 \times 10^{-23} JK^{-1}$  αποδεικνύεται (Gibbs 1902) ότι ακολουθούν την κατανομή Boltzmann:

$$p_\mu = \frac{1}{Z} e^{-\beta E_\mu} \quad (1.2.2)$$

Η παράμετρος  $\beta$  θα αποκαλείται για το υπόλοιπο της διπλωματικής απλά ως η θερμοκρασία του συστήματος. Η σταθερά  $Z$  ονομάζεται συνάρτηση επιμερισμού και αποτελεί την σταθερά κανονικοποίησης της κατανομής  $p_\mu$  η τιμή της οποίας δίνεται από τη σχέση:

$$Z = \sum_{\mu} e^{-E_\mu/\kappa T} = \sum_{\mu} e^{-\beta E_\mu} \quad (1.2.3)$$

Η  $Z$  αποτελεί μια ποσότητα που σύμφωνα με την παραπάνω σχέση απαρτιμεί με σχετικό βάρος τις δυνατές καταστάσεις του συστήματος. Η γνώση της συνάρτησης επιμερισμού συνεπάγεται και την γνώση οποιασδήποτε ενδιαφέρουσας μακροσκοπικής πληροφορίας του συστήματος. Στα συστήματα με πολύ μεγάλο βαθμό ελευθερίας  $N$  το ενδιαφέρον βρίσκεται στη μέση τιμή μιας ποσότητας, λόγω του ότι η πιθανότητα μέτρησης μιας τιμής που διαφέρει σημαντικά από τη μέση τιμή είναι αμελητέα. Η αναμενόμενη/μέση τιμή μιας ποσότητας  $Q$  για ένα σύστημα σε ισορροπία είναι:

$$\langle Q \rangle = \sum_{\mu} Q_\mu p_\mu = \frac{1}{Z} \sum_{\mu} Q_\mu e^{-\beta E_\mu} \quad (1.2.4)$$

Η μέση τιμή της ενέργειας  $\langle E \rangle$ , που στην θερμοδυναμική ονομάζεται εσωτερική ενέργεια  $U$  δίνεται από τη σχέση:

$$U = \langle E \rangle = \frac{1}{Z} \sum_{\mu} E_\mu e^{-\beta E_\mu} = -\frac{1}{Z} \sum_{\mu} \frac{\partial}{\partial \beta} e^{-\beta E_\mu} = -\frac{1}{Z} \frac{\partial}{\partial \beta} \sum_{\mu} e^{-\beta E_\mu} = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = -\frac{\partial \ln Z}{\partial \beta} \quad (1.2.5)$$

Με τον ίδιο τρόπο υπολογίζεται η ειδική θερμότητα:

$$C = \frac{\partial U}{\partial T} = \frac{\partial \beta}{\partial T} \frac{\partial U}{\partial \beta} = (-\kappa\beta^2) \left( -\frac{\partial^2 \ln Z}{\partial \beta^2} \right) = \kappa\beta^2 \frac{\partial^2 \ln Z}{\partial \beta^2} \quad (1.2.6)$$

Από τη θερμοδυναμική γνωρίζουμε ότι η ειδική θερμότητα σχετίζεται με την εντροπία μέσω της σχέσης:

$$C = T \frac{\partial S}{\partial T} = -\beta \frac{\partial S}{\partial \beta} \quad (1.2.7)$$

Άρα εξισώνοντας τις δύο εξισώσεις για την ειδική θερμότητα θα έχουμε την εξής σχέση για την εντροπία:

$$S = -\kappa\beta \frac{\partial \log Z}{\partial \beta} + \kappa \log Z \quad (1.2.8)$$

Μπορεί επίσης να εξαχθεί και μια σχέση για την ελεύθερη ενέργεια  $F$  του συστήματος:

$$F = U - TS = -\kappa T \log Z \quad (1.2.9)$$

Παρατηρούμε τη σημασία της συνάρτησης επιμερισμού καθώς οι ποσότητες  $U$ ,  $F$ ,  $C$  και  $S$  μπορούν όλες να υπολογιστούν απευθείας με χρήση της. Προχωρώντας ένα βήμα παρακάτω, και βάση της τελευταίας εξίσωσης, είναι επίσης δυνατός ο υπολογισμός άλλων παραμέτρων που αλληλεπιδρούν με το σύστημα. Στην κλασική θερμοδυναμική ένας αλληλεπιδρών παράγοντας

με το σύστημα(π.χ ένα μαγνητικό πεδίο με ένα μαγνήτη) έχει συζυγείς μεταβλητές που εκφράζουν την ανταπόκριση του συστήματος στη διαταραχή που προκαλείται. Η μαγνήτιση  $M$  ενός μαγνήτη αλλάζει συναρτήσει της επίδρασης ενός εξωτερικού πεδίου  $B$ : άρα η  $M$  και η  $B$  είναι συζυγείς. Σύμφωνα με τη θερμοδυναμική μπορούμε να υπολογίσουμε τις τιμές συζυγών μεταβλητών από παραγώγους της ελεύθερης ενέργειας:

$$M = \frac{\partial F}{\partial B} \quad (1.2.10)$$

Η σημασία της συνάρτησης επιμερισμού είναι λοιπόν εμφανής. Η μέθοδος Multiple Histogram που αποτελεί το θέμα αυτής της διπλωματικής είναι μια μέθοδος εκτίμησης της συνάρτησης επιμερισμού.

### 1.3 Διακυμάνσεις, Αποκρίσεις και Συναρτήσεις Συσχετισμού

Οι ροπές μιας παρατηρήσιμης ποσότητας  $\langle Q \rangle$  είναι στατιστικές συναρτήσεις οι οποίες έχουν μεγάλο φυσικό ενδιαφέρον και δίνονται από τη σχέση:

$$\langle (Q - \langle Q \rangle)^n \rangle, n = 1, 2, 3... \quad (1.3.1)$$

Για  $n = 2$  έχουμε τη ροπή:

$$(\Delta Q)^2 = \langle (Q - \langle Q \rangle)^2 \rangle, \quad (1.3.2)$$

η οποία δίνει ένα μέτρο της απόκλισης της  $Q$  από τη μέση τιμή  $\langle Q \rangle$ . Για την ενέργεια  $Q = E$  θα ισχύει:

$$(\Delta E)^2 = \langle (E - \langle E \rangle)^2 \rangle = \langle E^2 \rangle - \langle E \rangle^2, \quad (1.3.3)$$

όπου ο υπολογισμός της ποσότητας  $\langle E^2 \rangle$  μπορεί να γίνει από παραγώγους της συνάρτησης επιμερισμού με παρόμοιο τρόπο όπως η  $\langle E \rangle$ . Άρα θα ισχύει:

$$\langle E \rangle = \frac{1}{Z} \sum_{\mu} E_{\mu} e^{-\beta E_{\mu}} = -\frac{1}{Z} \frac{\partial}{\partial \beta} \sum_{\mu} e^{-\beta E_{\mu}} = -\frac{1}{Z} \frac{\partial Z}{\partial \beta}, \quad (1.3.4)$$

$$\langle E^2 \rangle = \frac{1}{Z} \sum_{\mu} E_{\mu}^2 e^{-\beta E_{\mu}} = \frac{1}{Z} \frac{\partial^2}{\partial \beta^2} \sum_{\mu} e^{-\beta E_{\mu}} = \frac{1}{Z} \frac{\partial^2 Z}{\partial \beta^2} \quad (1.3.5)$$

$$(\Delta E)^2 = \langle E^2 \rangle - \langle E \rangle^2 = \frac{1}{Z} \frac{\partial^2 Z}{\partial \beta^2} - \left( -\frac{1}{Z} \frac{\partial Z}{\partial \beta} \right)^2 = \frac{\partial^2 \ln Z}{\partial \beta^2}, \quad (1.3.6)$$

όπου σύμφωνα με τη σχέση 1.2.6 είναι η ειδική θερμότητα

$$C = \frac{\partial \langle E \rangle}{\partial T} = \kappa \beta^2 (\Delta E)^2 \quad (1.3.7)$$



Το αποτέλεσμα είναι ενδιαφέρον γιατί δίνει το μέγεθος των διακυμάνσεων συναρτήσει της ειδικής θερμότητας  $C$  ή της  $\log Z = -\beta F$ . Άρα ο υπολογισμός των διακυμάνσεων είναι δυνατός από ποσότητες που είναι διαθέσιμες μέσω της κλασικής θερμοδυναμικής. Η αναμενόμενη τιμή δίνει μια πάρα πολύ καλή εκτίμηση του τι αναμένουμε να δούμε στο πείραμα. Στο όριο ενός πολύ μεγάλου συστήματος (θερμοδυναμικό όριο) είναι δυνατό να αγνοηθούν πλήρως οι διακυμάνσεις. Το ενδιαφέρον σε συστήματα συμπυκνωμένης ύλης επικεντρώνεται στο θερμοδυναμικό όριο. Η χρήση των προσομοιώσεων Monte Carlo συνήθως δεν επιτρέπει την αναπαράσταση ενός συστήματος αρκετά μεγάλου ώστε η συμπεριφορά του να είναι συγκρίσιμη με ένα πραγματικά μεγάλο σύστημα.

Για τις συζυγείς μεταβλητές όπως αναφέρθηκε ο υπολογισμός τους γίνεται μέσω των παραγώγων της ελεύθερης ενέργειας. Οι παράγωγοι αυτής της γενικής μορφής δημιουργούνται από όρους της χαμιλτονιανής της μορφής  $-XY$  όπου  $Y$  είναι το πεδίο με ορισμένη τιμή και  $X$  η συζυγής μεταβλητή. Για παράδειγμα, η επίδραση ενός μαγνητικού πεδίου σε ένα μαγνήτη θα εμφανιστεί ως ένας όρος μαγνητικής ενέργειας στη χαμιλτονιανή της μορφής  $-MB$ , όπου  $M$  η μαγνήτιση του συστήματος και  $B$  το μαγνητικό πεδίο. Η αναμενόμενη τιμή της  $X$  δίνεται ως:

$$\langle X \rangle = \frac{1}{\beta Z} \sum_{\mu} X_{\mu} e^{-\beta E_{\mu}} = \frac{1}{\beta Z} \frac{\partial}{\partial Y} \sum_{\mu} e^{-\beta E_{\mu}} \quad (1.3.8)$$

με την ενέργεια  $E_{\mu}$  να περιλαμβάνει τον όρο  $-X_{\mu}Y$  όπου η παράγωγος επιδρά. Η σχέση μπορεί να γραφεί συναρτήσει της ελεύθερης ενέργειας άρα:

$$\langle X \rangle = \frac{1}{\beta} \frac{\partial \log Z}{\partial Y} = -\frac{\partial F}{\partial Y} \quad (1.3.9)$$

Με αυτό τον τρόπο μπορεί να γίνει υπολογισμός της αναμενόμενης τιμής μιας συζυγούς μεταβλητής απλά προσθέτοντας ένα υποθετικό πεδίο και τον κατάλληλο όρο στη χαμιλτονιανή και μηδενίζοντας το μετά την παραγωγή.

Μια επιπλέον παραγωγή του  $\log Z$  συναρτήσει του  $Y$  θα δώσει:

$$-\frac{1}{\beta} \frac{\partial^2 F}{\partial Y^2} = \frac{1}{\beta} \frac{\partial \langle X \rangle}{\partial Y} = \langle X^2 \rangle - \langle X \rangle^2 \quad (1.3.10)$$

άρα μπορούμε να βρούμε τις διακυμάνσεις όλων των ποσοτήτων που μας ενδιαφέρουν από δεύτερες παραγώγους της ελεύθερης ενέργειας με χρήση κατάλληλων πεδίων.

Η παράγωγος  $\partial \langle X \rangle / \partial Y$  που είναι μια μέτρηση του μεγέθους της απόκρισης της  $X$  σε αλλαγές του  $Y$  ονομάζεται επιδεικτικότητα της  $X$  στο  $Y$  και συμβολίζεται:

$$\chi = \frac{\partial \langle X \rangle}{\partial Y} \quad (1.3.11)$$

Οι σχέσεις για την μαγνήτιση θα είναι:

$$(\Delta M)^2 = \langle (M - \langle M \rangle)^2 \rangle = \langle M^2 \rangle - \langle M \rangle^2, \quad (1.3.12)$$

$$\langle M^2 \rangle = \frac{1}{\beta^2 Z} \frac{\partial^2 Z}{\partial B^2} \quad (1.3.13)$$

$$(\Delta M)^2 = \frac{1}{\beta^2} \left( \frac{1}{Z} \frac{\partial^2 Z}{\partial B^2} - \frac{1}{Z^2} \left( \frac{\partial Z}{\partial B} \right)^2 \right) = \frac{1}{\beta^2} \frac{\partial^2 \ln Z}{\partial B^2} = \frac{1}{\beta} \frac{\partial \langle M \rangle}{\partial B} \quad (1.3.14)$$

$$\chi = \frac{1}{N} \frac{\partial \langle M \rangle}{\partial B} = \frac{\beta}{N} \langle (M - \langle M \rangle)^2 \rangle \quad (1.3.15)$$

Επεκτείνοντας την ιδέα της επιδεκτικότητας και θεωρώντας ένα πλέγμα  $N$  θέσεων είναι δυνατή η μελέτη της μεταβολής των συζυγών μεταβλητών του συστήματος κατά την αλλαγή μιας παραμέτρου σε μια συγκεκριμένη θέση του πλέγματος.

Υποθέτουμε ένα χωρικά μεταβαλλόμενο πεδίο που παίρνει τις τιμές  $Y_i$  σε κάθε θέση του πλέγματος  $i$ . Οι συζυγείς μεταβλητές του συστήματος ορίζονται ως  $x_i$  και συνδέονται μέσω του όρου  $-\sum_i x_i Y_i$ . Η μέση τιμή του  $x_i$  είναι:

$$\langle x_i \rangle = \frac{1}{Z} \sum_{\mu} x_i^{\mu} e^{-\beta E_{\mu}} = \frac{1}{\beta} \frac{\partial \log Z}{\partial Y_i} \quad (1.3.16)$$

όπου  $x_i^{\mu}$  είναι η τιμή του  $x_i$  για κάποια κατάσταση  $\mu$ . Μπορεί να οριστεί μια γενικευμένη επιδεκτικότητα  $\chi_{ij}$  που αποτελεί μέτρηση της απόκρισης της  $x_i$  σε μια μεταβολή του πεδίου  $Y_j$  σε μια διαφορετική πλεγματοική θέση  $i$ , της μορφής:

$$\chi_{ij} = \frac{\partial \langle x_i \rangle}{\partial Y_j} = \frac{1}{\beta} \frac{\partial^2 \log Z}{\partial Y_i \partial Y_j} \quad (1.3.17)$$

Αντικαθιστώντας τη συνάρτηση επιμερισμού καταλήγουμε στη σχέση:

$$\begin{aligned} \chi_{ij} &= \frac{\beta}{Z} \sum_{\mu} x_i^{\mu} x_j^{\mu} e^{-\beta E_{\mu}} - \beta \left[ \frac{1}{Z} \sum_{\mu} x_i^{\mu} e^{-\beta E_{\mu}} \right] \left[ \frac{1}{Z} \sum_{\nu} x_j^{\nu} e^{-\beta E_{\nu}} \right] \implies \\ \chi_{ij} &= \beta (\langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle) = \beta G_c^{(2)}(i, j) \end{aligned} \quad (1.3.18)$$

Η ποσότητα  $G_c^{(2)}(i, j)$  ονομάζεται συνάρτηση συσχετισμού δύο σημείων (connected two point correlation function) της  $x$  ανάμεσα στις θέσεις  $i$  και  $j$ . Ο εκθέτης (2) χρησιμοποιείται για να την ξεχωρίσει από συναρτήσεις συσχετισμού ανώτερης τάξης. Η συνάρτηση παίρνει θετικές τιμές αν οι τιμές της  $x$  στις δύο θέσεις διακυμαίνονται προς την ίδια κατεύθυνση και αρνητική προς αντίθετη. Ο μηδενισμός της συνάρτησης αντιστοιχεί σε δυο ασυσχέτιστες ποσότητες.

## Κεφάλαιο 2

# Αρχές Monte Carlo Στατιστικής Φυσικής

### 2.1 Ο Εκτιμητής της Παρατηρήσιμης Ποσότητας

Κύριος στόχος μιας προσομοίωσης Monte Carlo είναι η εύρεση της αναμενόμενης τιμής  $\langle Q \rangle$  μιας παρατηρήσιμης ποσότητας  $Q$  του συστήματος, όπως η εσωτερική ενέργεια ή η μαγνήτιση. Ο υπολογισμός γίνεται με την άθροιση πάνω σε όλες τις καταστάσεις  $\mu$  του συστήματος, χρησιμοποιώντας και το κατάλληλο βάρους:

$$\langle Q \rangle = \frac{\sum_{\mu} Q_{\mu} e^{-\beta E_{\mu}}}{\sum_{\mu} e^{-\beta E_{\mu}}} \quad (2.1.1)$$

Η παραπάνω σχέση έχει νόημα για συστήματα μικρού μεγέθους. Σε συστήματα μεγάλου μεγέθους γίνεται άθροιση σε κάποιο υποσύνολο των καταστάσεων, το οποίο επιφέρει και ένα σφάλμα στον υπολογισμό. Οι τεχνικές Monte Carlo υλοποιούνται με την επιλογή ενός υποσυνόλου καταστάσεων με τυχαίο τρόπο από κάποια κατανομή πιθανότητας  $p_{\mu}$  που ορίζεται καταλλήλως. Η επιλογή λοιπών δείγματος από  $M$  καταστάσεις  $\{\mu_1, \mu_2, \dots, \mu_M\}$  δίνει τον εκτιμητή της ποσότητας  $Q$  σύμφωνα με τη σχέση:

$$Q_{\mu} = \frac{\sum_{i=1}^M Q_{\mu_i} p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}{\sum_{j=1}^M p_{\mu_j}^{-1} e^{-\beta E_{\mu_j}}} \quad (2.1.2)$$

Καθώς ο αριθμός των καταστάσεων  $M$  αυξάνει, ο εκτιμητής τείνει σε μια ακριβέστερη τιμή της  $\langle Q \rangle$  και για  $M \rightarrow \infty$  θα ισχύει  $Q_M = \langle Q \rangle$ .

Κύριος στόχος τώρα είναι η σωστή επιλογή της κατανομής πιθανότητας  $p_{\mu}$ . Ο τρόπος σκέψης είναι να γίνει επιλογή των καταστάσεων οι οποίες έχουν την μεγαλύτερη συνεισφορά στην (2.1.1) αγνοώντας όσο είναι δυνατό καταστάσεις που δεν συνεισφέρουν σημαντική πληροφορία για το σύστημα. Η τεχνική για την επιλογή αυτών των καταστάσεων ονομάζεται δειγματοληψία με κριτήριο σημαντικότητας.

## 2.2 Δειγματοληψία με Κριτήριο Σημαντικότητας

Επειδή η επιλογή κατάλληλων καταστάσεων συνεισφέρει σημαντικά στον υπολογισμό του  $\langle Q \rangle$ , όταν η πιθανότητα να επιλεγεί μια κατάσταση  $\mu$  ισούται με  $p_\mu = Z^{-1}e^{-\beta E_\mu}$  ο εκτιμητής γίνεται:

$$Q_\mu = \frac{\sum_{i=1}^M Q_{\mu_i} (e^{-\beta E_{\mu_i}})^{-1} e^{-\beta E_{\mu_i}}}{\sum_{j=1}^M (e^{-\beta E_{\mu_j}})^{-1} e^{-\beta E_{\mu_j}}} = \frac{1}{M} \sum_{i=1}^M Q_{\mu_i} \quad (2.2.1)$$

Καταλήξαμε σε μια απλή εξίσωση με την απαλοιφή των όρων Boltzmann, η οποία λειτουργεί ικανοποιητικά, ειδικά όταν το σύστημα βρίσκεται για το μεγαλύτερο μέρος του χρόνου του σε μικρό αριθμό καταστάσεων. Αυτές οι καταστάσεις θα επιλεγθούν συχνότερα και με σχετική συχνότητα που θα αντιστοιχεί στον χρόνο όπου το πραγματικό σύστημα ξοδεύει σε αυτές.

Η ερώτηση που απομένει είναι με ποιον τρόπο ακριβώς γίνεται η επιλογή αυτών των καταστάσεων έτσι ώστε να εμφανίζονται με τη σωστή πιθανότητα Boltzmann. Η απάντηση θα δοθεί μεσώ των διαδικασιών Markov.

### 2.2.1 Διαδικασίες Markov

Μια διαδικασία Markov αποτελεί ένα μηχανισμό όπου για μια δεδομένη κατάσταση  $\mu$  παράγει μια νέα κατάσταση του συστήματος  $\nu$  κατά τυχαίο τρόπο, δηλαδή δεν θα παραχθεί αναγκαστικά η ίδια κατάσταση  $\nu$  κάθε φορά που το σύστημα θα βρίσκεται στη  $\mu$ . Κάθε πιθανότητα μετάβασης  $P(\mu \rightarrow \nu)$  οφείλει να ικανοποιεί τις συνθήκες:

1. Να είναι ανεξάρτητη του χρόνου.
2. Να εξαρτάται μόνο από τις καταστάσεις  $\mu$  και  $\nu$  και όχι από την διαδρομή του συστήματος μέχρι την κατάσταση  $\mu$ .
3. Να ικανοποιείται η σχέση:

$$\sum_{\nu} P(\mu \rightarrow \nu) = 1 \quad (2.2.2)$$

όπου η πιθανότητα  $P(\mu \rightarrow \mu)$  μπορεί να είναι θετική. Δηλαδή υπάρχει η δυνατότητα το σύστημα να παραμείνει στην κατάσταση που βρίσκεται ήδη.

4. Για  $M \rightarrow \infty$  το δείγμα  $\{\mu_i\}$  ακολουθεί την κατανομή  $P_\mu$

Η επανάληψη διαδικασιών Markov ονομάζεται αλυσίδα Markov. Η επιλογή της αρχικής κατάστασης είναι σημαντική αφού αν δεν είναι μια τυπική κατάσταση του τελικού δείγματος τότε θα χρειαστεί κάποιος χρόνος ώστε το σύστημα να οδηγηθεί σε κατάσταση ισορροπίας. Για να προκύψει η επιθυμητή κατανομή από το δείγμα απαιτείται η ισχύς δυο σημαντικών συνθηκών των διαδικασιών Markov, της εργοδικότητας και της συνθήκης λεπτομερούς ισορροπίας.

### 2.2.2 Η Συνθήκη της Εργοδικότητας

Η συνθήκη της εργοδικότητας απαιτεί να είναι κάθε κατάσταση του συστήματος προσβάσιμη μέσω μιας διαδικασίας Markov.

Η συνθήκη επιτρέπει το μηδενισμό κάποιων πιθανοτήτων μετάβασης, αρκεί να υπάρχει μια μη μηδενική διαδρομή ανάμεσα σε δυο καταστάσεις που επιλέγουμε. Πρακτικά οι περισσότεροι αλγόριθμοι Monte Carlo κάνουν ακριβώς αυτό. Η παραβίαση της συνθήκης οδηγεί σε σημαντικά λάθη, καθώς είναι πιθανό καταστάσεις που συνεισφέρουν σημαντικά να μην είναι προσβάσιμες και η ζητούμενη κατανομή δεν προκύπτει.

### 2.2.3 Η Συνθήκη της Λεπτομερούς Ισορροπίας

Η συνθήκη που ορίζει ότι το σύστημα βρίσκεται σε ισορροπία είναι:

$$\sum_{\nu} p_{\mu} P(\mu \rightarrow \nu) = \sum_{\nu} p_{\nu} P(\nu \rightarrow \mu), \quad (2.2.3)$$

δηλαδή ο ρυθμός με τον οποίο το σύστημα μεταβαίνει από και προς την κατάσταση  $\mu$  είναι ίσος. Σύμφωνα με την (2.2.2) η παραπάνω εξίσωση απλοποιείται στην:

$$p_{\mu} = \sum_{\nu} p_{\nu} P(\nu \rightarrow \mu) \quad (2.2.4)$$

Η παραπάνω εξίσωση δεν είναι αρκετή για τη δημιουργία της ζητούμενης κατανομής. Η επιπλέον συνθήκη που χρειάζεται, αυτή δηλαδή της λεπτομερούς ισορροπίας είναι η:

$$p_{\mu} P(\mu \rightarrow \nu) = p_{\nu} P(\nu \rightarrow \mu), \quad (2.2.5)$$

Κάθε σύνολο πιθανοτήτων μετάβασης που ικανοποιεί την παραπάνω συνθήκη θα ικανοποιεί και την (2.2.3). Το σύστημα θα μεταβαίνει από την κατάσταση  $\mu$  στην κατάσταση  $\nu$  σχεδόν όσο συχνά μεταβαίνει από την  $\nu$  στην  $\mu$ . Οι καταστάσεις που θα παράγονται μέσω της διαδικασίας Markov θα ακολουθούν την επιθυμητή κατανομή  $p_{\mu}$ . Εφόσον στόχος είναι να ακολουθούν την κατανομή Boltzmann θα πρέπει να ικανοποιείται η εξίσωση:

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{p_{\nu}}{p_{\mu}} = e^{-\beta(E_{\nu} - E_{\mu})} \quad (2.2.6)$$

Η παραπάνω εξίσωση καθώς και η (2.2.2) αποτελούν τους δυο απαραίτητους περιορισμούς που πρέπει να τηθούν στην επιλογή των πιθανοτήτων μετάβασης  $P(\mu \rightarrow \nu)$ . Με την ικανοποίηση και της συνθήκης της εργοδικότητας θα έχουμε την κατανομή Boltzmann.

## 2.3 Λόγοι Αποδοχής

Αν και είναι δυνατό να προταθούν πολλές διαδικασίες Markov για τις οποίες οι πιθανότητες μετάβασης θα ικανοποιούν τις (2.2.2) και (2.2.6) είναι δύσκολο να βρεθεί κάποια που θα έχει

ακριβώς το επιθυμητό σύνολο πιθανοτήτων μετάβασης. Αυτό το πρόβλημα μπορεί να ξεπεραστεί με την χρήση του λόγου αποδοχής κάνοντας δυνατή τη χρήση οποιουδήποτε αλγορίθμου επιλογής νέων καταστάσεων.

Όπως προαναφέρθηκε το σύστημα μπορεί να παραμείνει στην ίδια κατάσταση κατά μια μετάβαση αν αυτό είναι επιθυμητό. Για  $\nu = \mu$  στην (2.2.6) η συνθήκη της λεπτομερούς ισορροπίας ικανοποιείται ανεξάρτητα από την τιμή που επιλέγουμε για αυτή. Άρα έχουμε την δυνατότητα να επιλέξουμε πιο ελεύθερα τις τιμές πιθανοτήτων μετάβασης για  $\nu \neq \mu$  ενώ η (2.2.2) θα συνεχίζει να ικανοποιείται αρκεί για κάθε μεταβολή να γίνεται μια ίση και αντίθετη στην πιθανότητα μετάβασης για  $\nu = \mu$ . Είναι σημαντική η προσοχή στη διατήρηση της τιμής της  $P(\mu \rightarrow \mu)$  μεταξύ μηδέν και ένα.

Η πιθανότητα μετάβασης χωρίζεται σε δύο μέρη:

$$P(\mu \rightarrow \nu) = g(\mu \rightarrow \nu)A(\mu \rightarrow \nu) \quad (2.3.1)$$

Η ποσότητα  $g(\mu \rightarrow \nu)$  είναι η πιθανότητα επιλογής της κατάστασης  $\nu$  και  $A(\mu \rightarrow \nu)$  είναι ο λόγος αποδοχής που εκφράζει το αν θα γίνει αποδεκτή η νέα κατάσταση  $\nu$  και θα γίνει η μετάβαση.

Ένας ιδανικός αλγόριθμος θα επιλέγει νέες καταστάσεις με σωστές πιθανότητες μετάβασης και ο λόγος αποδοχής του θα είναι πάντα ένα.

## Κεφάλαιο 3

# Το Πρότυπο Ising

### 3.1 Εισαγωγή

Το πρότυπο Ising είναι το μοντέλο ενός μαγνήτη. Έχει τη γεωμετρία ενός πλέγματος  $d$  διαστάσεων και υποθέτουμε ότι η μαγνήτιση του προκύπτει από την συμπεριφορά των μαγνητικών διπόλων που το απαρτίζουν. Σε κάθε θέση του πλέγματος βρίσκεται μια βαθμωτή ποσότητα  $s_i$  που μπορεί να πάρει τις τιμές  $\pm 1$  και που αντιστοιχεί σε ένα δίπολο με κατεύθυνση πάνω ή κάτω. Η αλληλεπίδραση μεταξύ των σπιν αναπαρίσταται στο πρότυπο Ising μέσω της εμφάνισης ενός όρου  $J$  με διαστάσεις ενέργειας όπου θεωρούμε αλληλεπίδραση μόνο μεταξύ πλησιέστερων γειτόνων  $\langle ij \rangle$ , ίσης ποσότητας.

Επίσης είναι δυνατή η επίδραση ενός εξωτερικού πεδίου  $B$  με τη χαμιλτονιανή να παίρνει τη μορφή:

$$H = -J \sum_{\langle ij \rangle} s_i s_j - B \sum_i s_i \quad (3.1.1)$$

Τα αρνητικά πρόσημα είναι συμβατικά, με την θετική τιμή του  $J$  να ορίζει ένα σιδηρομαγνητικό μοντέλο και τα σπιν να έχουν ίδια κατεύθυνση με το εξωτερικό πεδίο  $B$ .

Οι καταστάσεις του προτύπου ορίζονται από τα σύνολα των τιμών των σπιν σε κάθε θέση με  $2^N$  τον συνολικό αριθμό των καταστάσεων για ένα πλέγμα με  $N$  σπιν. Η συνάρτηση επιμερισμού του μοντέλου είναι:

$$Z = \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} \dots \sum_{s_N=\pm 1} e^{-\beta H[\{s_i\}]} = \sum_{\{s_i\}} e^{\beta J \sum_{\langle ij \rangle} s_i s_j + \beta B \sum_i s_i} \quad (3.1.2)$$

όπου  $\{s_i\}$  η διάταξη των σπιν στο πλέγμα. Στόχος είναι η προσομοίωση του προτύπου Ising πεπερασμένου μεγέθους για την εκτίμηση ποσοτήτων όπως η εσωτερική ενέργεια, η μαγνήτιση, η ειδική θερμότητα και η μαγνητική επιδεκτικότητα. Για το υπόλοιπο της διπλωματικής η τιμή  $B$  του εξωτερικού μαγνητικού πεδίου θεωρείται ίση με μηδέν.

## 3.2 Ο Αλγόριθμος Metropolis

Σύμφωνα με τα όσα αναφέρθηκαν για τις αρχές Monte Carlo ο αλγόριθμος Metropolis προκύπτει αρχικά με την επιλογή συνόλου πιθανοτήτων επιλογής  $g(\mu \rightarrow \nu)$  για κάθε πιθανή μετάβαση από μια κατάσταση σε άλλη οι οποίες θα ικανοποιούν τη συνθήκη της εργοδικότητας. Επιπλέον επιλέγεται ένα κατάλληλο σύνολο λόγων αποδοχής  $A(\mu \rightarrow \nu)$  που ικανοποιεί την συνθήκη της λεπτομερούς ισορροπίας. Ο αλγόριθμος λειτουργεί με τη συνεχόμενη επιλογή κάποιας νέας κατάσταση  $\nu$  και αποφασίζει αν θα την αποδεχτεί ή όχι σύμφωνα με την πιθανότητα αποδοχής.

Το μοντέλο Ising στην πραγματικότητα παραμένει σε ένα υποσύνολο των καταστάσεων με μικρό εύρος διαφοράς ενεργειών χωρίς δραματικές αλλαγές. Αυτό μπορεί να προσομοιωθεί με την αλλαγή ενός σπίν όπου η ενέργεια της νέας κατάστασης  $E_\nu$  θα διαφέρει από την  $E_\mu$  κατά μέγιστο  $2J$  για κάθε δεσμό μεταξύ του σπίν και των πλησιέστερων γειτόνων. Η επιλογή μεταβολής ενός μόνο σπίν ικανοποιεί επίσης και την συνθήκη της εργοδικότητας, αφού όλες οι καταστάσεις είναι προσβάσιμες.

Στον αλγόριθμο Metropolis οι πιθανότητες επιλογής  $g(\mu \rightarrow \nu)$  για κάθε πιθανή κατάσταση  $\nu$  θέτονται ίσες μεταξύ τους ενώ για τις υπόλοιπες καταστάσεις μηδενίζονται. Για τον αριθμό  $N$  όλων των σπίν που μπορούν να αλλάξουν κατεύθυνση υπάρχουν  $N$  πιθανές καταστάσεις  $\nu$  που μπορούμε να μεταβούμε από την  $\mu$  και  $N$  πιθανότητες μετάβασης  $g(\mu \rightarrow \nu)$  που είναι μη μηδενικές με τιμή:

$$g(\mu \rightarrow \nu) = \frac{1}{N} \quad (3.2.1)$$

Η συνθήκη λεπτομερούς ισορροπίας γίνεται τότε:

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{g(\mu \rightarrow \nu)A(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu)A(\nu \rightarrow \mu)} = \frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)} \quad (3.2.2)$$

Η επιλογή των λόγων αποδοχής ως:

$$A(\mu \rightarrow \nu) = \begin{cases} \exp(-\beta(E_\nu - E_\mu)), & \text{αν } E_\nu - E_\mu > 0. \\ 1, & \text{σε κάθε άλλη περίπτωση} \end{cases} \quad (3.2.3)$$

μας δίνει τον αλγόριθμο Metropolis. Η επιλογή μιας νέας κατάστασης με ενέργεια μικρότερη ή ίση της τωρινής θα γίνεται πάντα αποδεκτή ενώ μια κατάσταση με μεγαλύτερη ενέργεια θα γίνεται αποδεκτή με την ανάλογη πιθανότητα που δίνεται από την παραπάνω σχέση.

### 3.2.1 Περιγραφή του Αλγορίθμου Metropolis

Αρχικά πρέπει να οριστεί η δομή δεδομένων, ένα διάνυσμα ακέραιων αριθμών  $N$ , που αντιστοιχούν στα σπιν των πλεγματικών θέσεων, το οποίο θα παίρνει τιμές  $\pm 1$ . Σημαντικός είναι και ο ορισμός των σχέσεων γειτονίας των πλεγματικών θέσεων και των συνοριακών συνθηκών. Μπορεί να γίνει μια επιλογή ανάμεσα σε περιοδικές, ελικοειδείς ή τοροειδείς. Ο τρόπος επιλογής καθορίζει αν το σύστημα θα έχει τοπική γεωμετρία και κάθε πλεγματική θέση θα έχει τον ίδιο αριθμό πλησιέστερων γειτόνων.



Έπειτα γίνεται επιλογή της θερμοκρασίας  $\beta = \frac{1}{\kappa T}$  για την οποία θα εκτελεστεί η προσομοίωση και των αρχικών τιμών των σπιν, δηλαδή της αρχικής κατάστασης. Δυο συχνές επιλογές αρχικών καταστάσεων είναι για μηδενική ή άπειρη θερμοκρασία. Για  $T = 0$  το σύστημα βρίσκεται σε θεμελιώδη κατάσταση. Όταν η ενέργεια αλληλεπίδρασης  $J$  είναι μεγαλύτερη του μηδενός και το εξωτερικό μαγνητικό πεδίο είναι ίσο με μηδέν υπάρχουν δύο θεμελιώδεις καταστάσεις που αντιστοιχούν σε όλα τα σπιν να βρίσκονται πάνω ή κάτω. Για  $T = \infty$  η θερμική ενέργεια  $\kappa T$  είναι απείρως μεγαλύτερη της ενέργειας αλληλεπίδρασης των σπιν  $J$ , άρα τα σπιν προσανατολίζονται τυχαία πάνω ή κάτω κατά ασυσχέτιστο τρόπο.

Στο επόμενο βήμα γίνεται η έναρξη της προσομοίωσης. Γίνεται παραγωγή μιας νέας κατάστασης  $\nu$  η οποία διαφέρει με αλλαγή κατεύθυνσης ενός σπιν το οποίο διαλέγεται τυχαία από το πλέγμα.

Ακολουθεί ο υπολογισμός της διαφοράς ενέργειας  $E_\nu - E_\mu$ . Η μεταβολή ενέργειας ανάμεσα σε δύο καταστάσεις ισούται με

$$E_\nu - E_\mu = -J \sum_{\langle ij \rangle} s_i^\nu s_j^\nu + J \sum_{\langle ij \rangle} s_i^\mu s_j^\mu = -J \sum_i s_i^\mu (s_i^\nu - s_i^\mu) \quad (3.2.4)$$

Με τον τελευταίο όρο να είναι το άθροισμα ανάμεσα στα σπιν πλησιέστερων γειτόνων και θεωρώντας ότι όλα αυτά τα σπιν δεν αλλάζουν κατεύθυνση ισχύει  $s_i^\nu = s_i^\mu$ . Αν  $s_k^\mu = +1$  τότε όταν το σπιν  $k$  αλλάξει κατεύθυνση θα έχουμε  $s_k^\nu = -1$  έτσι ώστε  $s_k^\nu - s_k^\mu = -2$ . Επίσης αν  $s_k^\mu = -1$  θα έχουμε  $s_k^\nu - s_k^\mu = +2$ . Άρα μπορούμε να γράψουμε:

$$s_k^\nu - s_k^\mu = -2s_k^\mu \quad (3.2.5)$$

και ο υπολογισμός της διαφοράς ενέργειας γίνεται:

$$E_\nu - E_\mu = 2J \sum_i s_i^\mu s_k^\mu = 2J s_k^\mu \sum_i s_i^\mu \quad (3.2.6)$$

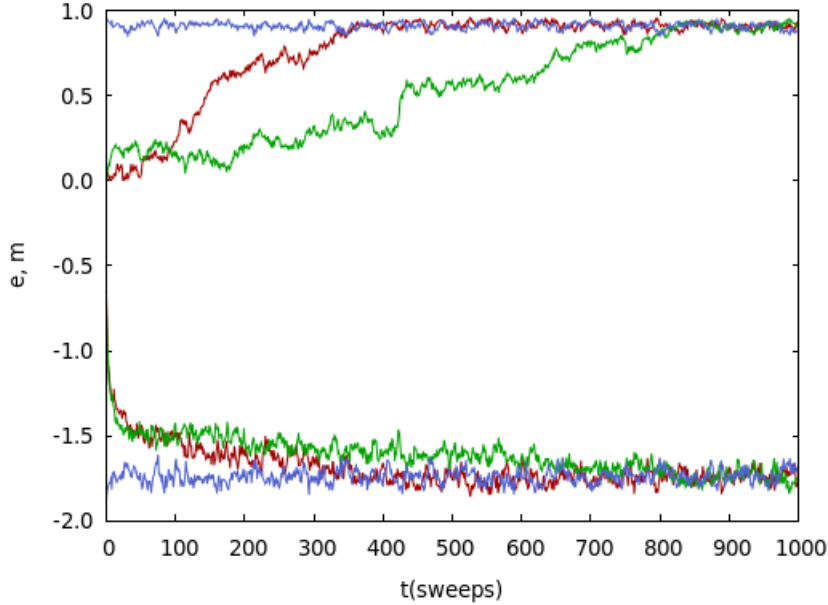
Με αυτό τον τρόπο αθροίζουμε πάνω σε  $z$  όρους και όχι  $\frac{1}{2}Nz$  που αποτελούν τους συνολικούς δεσμούς του συστήματος.

Τέλος ο αλγόριθμος με δεδομένη την διαφορά ενέργειας αποφασίζει βάση της σχέσης (3.2.3) αν θα αλλάξει την κατεύθυνση του σπιν και θα μεταβεί στη νέα κατάσταση και συνεχίζεται εκ νέου από το βήμα της επιλογής του προς αλλαγή κατεύθυνσης σπιν.

### 3.3 Θερμοδυναμική Ισορροπία

Όπως προαναφέρθηκε το σύστημα βρίσκεται σε ισορροπία όταν η κατά μέσο όρο πιθανότητα να βρεθεί σε μια συγκεκριμένη κατάσταση  $\mu$  είναι ανάλογη του βάρους Boltzmann της  $e^{-\beta E_\mu}$ . Δεδομένης της αρχικής κατάστασης το σύστημα μπορεί να χρειαστεί κάποιο χρόνο για να βρεθεί σε ισορροπία. Σύμφωνα με τον αλγόριθμο Metropolis η αλλαγή ενός μόνο σπιν μπορεί να κάνει αυτή τη διαδικασία ακόμα πιο αργή.

Ένας αξιόπιστος τρόπος υπολογισμού του χρόνου που χρειάζεται για να βρεθεί το σύστημα σε ισορροπία είναι με την σχεδίαση του διαγράμματος της χρονικής εξέλιξης των υπολογισμένων ποσοτήτων (ενέργεια η/και μαγνήτιση).



Σχήμα 3.1: Διαδικασία εύρεσης της θερμικής ισορροπίας. Κάθε χρώμα αντιστοιχεί στη μαγνήτιση (πάνω) και εσωτερική ενέργεια (κάτω) ανά πλεγματική θέση της ίδιας προσομοίωσης για το πρότυπο Ising με  $L = 40$  και  $\beta = 0.50$ . Έχουν επιλεγεί δύο αρχικές καταστάσεις θερμές ( $\beta = 0$ ) και μια παγωμένη ( $\beta = \infty$ ). Οι δύο θερμές καταστάσεις ακολουθούν διαφορετική στοχαστική εξέλιξη. Το διάγραμμα έχει ως στόχο να δείξει ότι μια διαφορετική στοχαστική εξέλιξη για τις ίδιες ιδιότητες μιας προσομοίωσης μπορεί να οδηγήσει το σύστημα μετά από διαφορετικό χρόνο σε ισορροπία. Προφανώς τα 1000 βήματα δεν είναι αρκετά για να θεωρήσουμε ότι έχει επιτευχθεί θερμική ισορροπία.

Η χρονική εξέλιξη εκφράζεται μέσω των βημάτων (sweeps) του συστήματος, δηλαδή στην συγκεκριμένη περίπτωση του αλγορίθμου Metropolis κάθε βήμα αντιστοιχεί στην μεταβολή ενός σπιν. Μια καλύτερη επιλογή είναι η σχεδίαση της χρονικής εξέλιξης μιας ποσότητας πολλαπλών προσομοιώσεων όπως φαίνεται στο σχήμα 3.1.

### 3.4 Υπολογισμός Ποσοτήτων της Προσομοίωσης

Με το σύστημα να έχει βρεθεί σε κατάσταση ισορροπίας είναι δυνατός ο υπολογισμός των ποσοτήτων που μας ενδιαφέρουν, δηλαδή της ενέργειας και της μαγνήτισης. Η ενέργεια  $E_\mu$  της κατάστασης  $\mu$  μπορεί να υπολογιστεί κατευθείαν από τη χαμιλτονιανή αντικαθιστώντας τις τιμές των σπιν που βρίσκονται στη δομή δεδομένων. Ωστόσο κατά την υλοποίηση του αλγορίθμου έγινε υπολογισμός της διαφοράς ενέργειας  $\Delta E = E_\nu - E_\mu$  κατά τη μετάβαση κατάστασης. Άρα με δεδομένη την ενέργεια της τωρινής κατάστασης είναι δυνατός ο υπολογισμός της νέας ενέργειας που θα προκύψει από την αλλαγή κατεύθυνσης ενός σπιν μέσω της σχέσης:

$$E_\nu = E_\mu + \Delta E \quad (3.4.1)$$

Με άλλα λόγια ο υπολογισμός μέσω της Χαμιλτονιανής είναι απαραίτητος μόνο κατά την έναρξη της προσομοίωσης και κάθε νέα ενέργεια θα υπολογίζεται από τη σχέση (3.4.1).

Η ολική μαγνήτιση του συστήματος  $M_\mu$  σε μια κατάσταση  $\mu$  ισούται με:

$$M_\mu = \sum_i s_i^\mu \quad (3.4.2)$$

και παρομοίως με την αλλαγή μόνο ενός σπιν κάθε φορά η μεταβολή της μαγνήτισης ανάμεσα σε δύο καταστάσεις είναι:

$$\Delta M = M_\nu - M_\mu = \sum_i s_i^\nu - \sum_i s_i^\mu = s_\kappa^\nu - s_\kappa^\mu = 2s_\kappa^\nu \quad (3.4.3)$$

Οπότε ο υπολογισμός της μαγνήτισης κατά την έναρξη της προσομοίωσης οδηγεί στον υπολογισμό για κάθε επόμενη κατάσταση μέσω της σχέσης:

$$M_\nu = M_\mu + \Delta M = M_\mu + 2s_\kappa^\nu \quad (3.4.4)$$

Έχοντας υπολογίσει την ενέργεια και τη μαγνήτιση του συστήματος μπορούμε να υπολογίσουμε τον μέσο τους για να βρούμε τους εκτιμητές της εσωτερικής ενέργειας και της μαγνήτισης. Διαιρώντας με τον αριθμό  $N$  των πλεγματικών θέσεων μπορούμε να υπολογίσουμε αυτές τις ποσότητες ανά σπιν και αυτός είναι ο τρόπος με τον οποίο θα παρουσιαστούν τα δεδομένα αυτής της διπλωματικής.

Για  $\kappa = 1$  και με υπολογισμό των τετραγώνων της ενέργειας και της μαγνήτισης είναι επίσης δυνατός ο υπολογισμός της ειδικής θερμότητας και της μαγνητικής επιδεκτικότητας μέσω των σχέσεων:

$$c = \frac{\beta^2}{N} (\langle E^2 \rangle - \langle E \rangle^2) \quad (3.4.5)$$

$$\chi = \beta N (\langle m^2 \rangle - \langle m \rangle^2) \quad (3.4.6)$$

### 3.4.1 Συναρτήσεις Αυτοσυσχετισμού

Μια πολύ σημαντική έννοια στις προσομοιώσεις Monte Carlo είναι αυτή του χρόνου αυτοσυσχετισμού  $\tau$ . Ο χρόνος αυτοσυσχετισμού δίνει μια μέτρηση του πόσο χρόνο θα χρειαστεί το σύστημα για να βρεθεί σε μια κατάσταση η οποία θα διαφέρει σημαντικά από την πρώτη.

Ο αυτοσυσχετισμός (time-displaced autocorrelation) της μαγνήτισης  $m$  του προτύπου Ising δίνεται από τη σχέση:

$$\chi(t) = \int dt' [m(t') - \langle m \rangle][m(t' + t) - \langle m \rangle] = \int dt' [m(t')m(t' + t) - \langle m \rangle^2] \quad (3.4.7)$$

όπου  $m(t)$  η μαγνήτιση για μία δεδομένη στιγμή και  $\langle m \rangle$  η μέση τιμή. Ο αυτοσυσχετισμός δίνει μια παρόμοια μέτρηση του συσχετισμού της ποσότητας για δύο διαφορετικές χρονικές στιγμές. Θετική τιμή συνεπάγεται ότι οι ποσότητες διακυμαίνονται προς την ίδια κατεύθυνση για τις δύο χρονικές στιγμές και αρνητική για διακύμανση σε διαφορετικές κατευθύνσεις. Με

την ολοκλήρωση στον χρόνο, μια μηδενική τιμή αντιστοιχεί σε ασυσχέτιστες ποσότητες. Ο αυτοσυσχετισμός ελαττώνεται εκθετικά:

$$\chi(t) \sim e^{-t/\tau} \quad (3.4.8)$$

Παρατηρείται από την παραπάνω εξίσωση ότι υπάρχει μια έντονη συσχέτιση ακόμα και για χρόνο  $t = \tau$  αφού ο αυτοσυσχετισμός έχει μειωθεί κατά ένα παράγοντα  $1/e$ . Μια σχετικά ασφαλής επιλογή είναι η επιλογή δειγμάτων ανά  $2\tau$ .

Για μία προσομοίωση η οποία διαρκεί χρόνο  $t_{max}$  ο αριθμός των ανεξάρτητων μετρήσεων που μπορούμε να λάβουμε, αφού το σύστημα έχει βρεθεί σε ισορροπία, είναι της τάξης:

$$n = \frac{t_{max}}{2\tau} \quad (3.4.9)$$

Συνηθίζεται στις προσομοιώσεις Monte Carlo να γίνονται μετρήσεις σε διαστήματα μικρότερα του χρόνου αυτοσυσχετισμού. Οι λόγοι για τους οποίους συμβαίνει αυτό είναι ότι συνήθως δεν γνωρίζουμε την τιμή του πριν την προσομοίωση και επιπλέον πρέπει να μπορεί να υπολογιστεί η συνάρτηση αυτοσυσχετισμού για χρόνους μικρότερους του χρόνου αυτοσυσχετισμού έτσι ώστε να μπορεί να γίνει μια καλύτερη εκτίμηση του.

Μπορεί επίσης να οριστεί για την εκτίμηση του χρόνου αυτοσυσχετισμού ο ολοκληρωμένος χρόνος αυτοσυσχετισμού integrated correlation time. Θεωρώντας την (3.4.8) για όλες τις τιμές  $t$  θα έχουμε:

$$\int_0^\infty \frac{\chi(t)}{\chi(0)} dt = \int_0^\infty e^{-t/\tau} dt = \tau \quad (3.4.10)$$

Στον υπολογισμό υπάρχουν στατιστικά σφάλματα που κάνουν τις τιμές να διαφέρουν. Η εξίσωση (3.4.8) είναι αυστηρά σωστή μόνο για μεγάλες τιμές του  $t$ . Συνήθως ο τρόπος υπολογισμού του χρόνου αυτοσυσχετισμού γίνεται με χρήση της εξίσωσης (3.4.10).

### 3.5 Η Μετάβαση Φάσης

Το πρότυπο Ising παρουσιάζει ενδιαφέρουσα αλλαγή των ιδιοτήτων του για την κρίσιμη θερμοκρασία:

$$T_c = \frac{2}{\ln(1 + \sqrt{2})} \approx 2.269 \quad (3.5.1)$$

$$\beta_c = \frac{1}{2} \ln(1 + \sqrt{2}) \approx 0.4406867935 \dots \quad (3.5.2)$$

Για μεγαλύτερες θερμοκρασίες  $T > T_c$  η μέση μαγνήτιση ανά σπιν  $m$  είναι αρκετά μικρή, ενώ για μικρότερες έχει μη μηδενική τιμή και μάλιστα σχεδόν ίση με την μέγιστη τιμή ένα. Η συμπεριφορά αυτή του μοντέλου είναι αναμενόμενη λόγω του προσανατολισμού των σπιν για τις δύο ακραίες καταστάσεις θερμοκρασίας όπως συζητήθηκε κατά την υλοποίηση του αλγορίθμου Metropolis στο υποκεφάλαιο 3.2.1. Ωστόσο για μεγαλύτερα πλέγματα παρατηρείται ότι η μετάβαση της μαγνήτισης  $m$  γίνεται όλο και πιο απότομη, και μάλιστα για το συγκεκριμένο

μοντέλο γνωρίζουμε, λόγω της ακριβής λύσης του, ότι απειρίζεται για το θερμοδυναμικό όριο. Αυτή η δραματική αλλαγή των ιδιοτήτων του συστήματος ονομάζεται μετάβαση φάσης.

Για τιμές θερμοκρασίας μεγαλύτερες της κρίσιμης το μοντέλο βρίσκεται σε παραμαγνητική φάση στην οποία η μέση τιμή της μαγνήτισης ισούται με μηδέν. Για μικρότερες τιμές βρίσκεται σε σιδηρομαγνητική όπου αποκτά μη μηδενική μαγνήτιση λόγω των σπιν του, χωρίς την επίδραση κάποιου εξωτερικού μαγνητικού πεδίου και η οποία παίρνει τη μέγιστη τιμή της για το απόλυτο μηδέν. Κάθε ποσότητα η οποία μηδενίζεται στη μία πλευρά της μετάβασης φάσης και είναι μη μηδενική στην άλλη ονομάζεται παράμετρος τάξης. Η μαγνήτιση είναι λοιπόν μια παράμετρος τάξης.

Όπως προαναφέρθηκε στην 3.2.1 σε υψηλές θερμοκρασίες τα σπιν προσανατολίζονται τυχαία και κατά ασυσχέτιστο τρόπο, ενώ σε χαμηλές θερμοκρασίες έχουν την τάση κοντινά σπιν να προσανατολίζονται προς την ίδια κατεύθυνση ενισχύοντας τον συσχετισμό και δημιουργώντας ένα σύμπλεγμα (cluster). Καθώς το σύστημα οδηγείται στην κρίσιμη θερμοκρασία, το τυπικό μέγεθος  $\xi$ , που ονομάζεται μήκος συσχετισμού, αυτών των συμπλεγμάτων τείνει στο άπειρο, δημιουργώντας πολύ μεγάλες περιοχές όπου τα σπιν προσανατολίζονται πάνω ή κάτω.

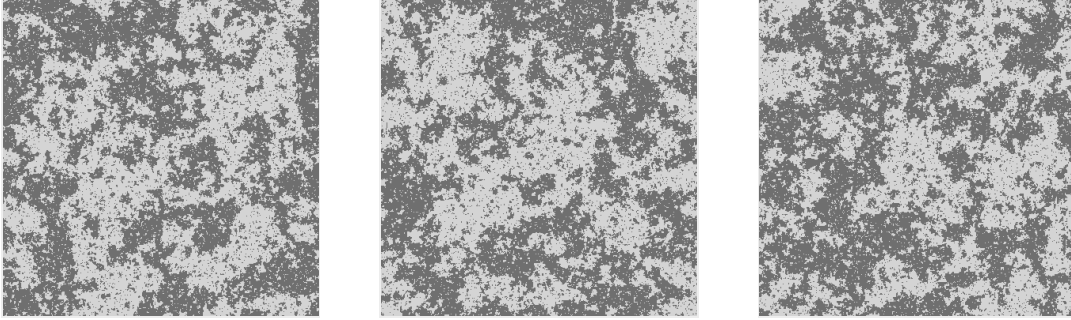
### 3.6 Κρίσιμες Διακυμάνσεις και Κρίσιμη Επιβράδυνση

Το ενδιαφέρον για την μελέτη του προτύπου Ising επικεντρώνεται για τιμές κοντά στη κρίσιμη θερμοκρασία, δηλαδή για την κρίσιμη περιοχή η οποία διέπεται από κρίσιμα φαινόμενα. Η δημιουργία των συμπλεγμάτων στη κρίσιμη περιοχή οδηγεί σε σημαντική μεταβολή των ποσοτήτων της εσωτερικής ενέργειας  $E$  και μαγνήτισης  $m$  δημιουργώντας κρίσιμες διακυμάνσεις. Καθώς το μήκος συσχετισμού αποκλίνει όταν  $T \rightarrow T_c$ , το μέγεθος των διακυμάνσεων αποκλίνει επίσης. Επιπλέον οι ποσότητες της ειδικής θερμότητας και μαγνητικής επιδεκτικότητας εφόσον εξαρτώνται από τις  $E$  και  $m$  εμφανίζουν παρόμοια συμπεριφορά. Αυτές οι αποκλίσεις αποτελούν κάποια από τα πιο ενδιαφέροντα κρίσιμα φαινόμενα που μελετούνται με χρήση των προσομοιώσεων Monte Carlo.

Δυστυχώς η χρήση του αλγόριθμου Metropolis είναι απαγορευτική για τη μελέτη αυτής της περιοχής. Τα στατιστικά σφάλματα στις ποσότητες της  $E$  και  $m$  είναι ανάλογα του μεγέθους των κρίσιμων διακυμάνσεων και άρα αυξάνουν όταν  $T \rightarrow T_c$ . Σε ένα σύστημα πεπερασμένου μεγέθους το μέγεθος των διακυμάνσεων δεν αποκλίνει ποτέ αλλά αυξάνεται σημαντικά με αποτέλεσμα την αύξηση των στατιστικών σφαλμάτων.

Μια πιθανή λύση θα ήταν η αύξηση των στατιστικά ανεξάρτητων μετρήσεων μιας προσομοίωσης. Ωστόσο ο χρόνος αυτοσυσχετισμού  $\tau$  της προσομοίωσης είναι επίσης μεγάλος για τιμές κοντά στη κρίσιμη θερμοκρασία αφού δημιουργούνται συμπλέγματα. Για την ακρίβεια και ο χρόνος αυτοσυσχετισμού αποκλίνει για την κρίσιμη θερμοκρασία στο θερμοδυναμικό όριο και γίνεται πολύ μεγάλος για συστήματα πεπερασμένου μεγέθους. Άρα ένας μεγάλος χρόνος αυτοσυσχετισμού συνεπάγεται μικρότερο αριθμό στατιστικά ανεξάρτητων μετρήσεων.

Οι κρίσιμες διακυμάνσεις αποτελούν ιδιότητα του προτύπου Ising και δεν μπορούν να αντιμετωπιστούν. Κάθε αλγόριθμος Monte Carlo με Boltzmann κατανομή θα εμφανίζει κρίσιμες διακυμάνσεις. Ωστόσο η αύξηση του χρόνου αυτοσυσχετισμού  $\tau$ , η οποία ονομάζεται κρίσι-



Σχήμα 3.2: Διατάξεις των σπιν για το πρότυπο Ising με  $L = 400$  μετά από 5000, 20000, 40000 βήματα αντίστοιχα με τον αλγόριθμο Metropolis. Παρατηρείται η δημιουργία συμπλεγμάτων μεγάλου μεγέθους όπου δεν είναι δυνατό να καταστραφούν και να δημιουργηθούν αλλού με ένα αλγόριθμο αλλαγής μοναδικών σπιν ανά βήμα. Για το λόγο αυτό οι χρόνοι αυτοσυσχετισμού είναι μεγάλοι.

μη επιβράδυνση, αποτελεί ιδιότητα του αλγορίθμου και μπορεί να αντιμετωπιστεί. Αρα για τη μελέτη της κρίσιμης περιοχής χρειαζόμαστε ένα νέο αλγόριθμο, ο οποίος θα προσφέρει μεγαλύτερη ακρίβεια στις μετρήσεις μας.

## Κεφάλαιο 4

# Κρίσιμοι Εκθέτες και ο Αλγόριθμος Wolff

### 4.1 Κρίσιμοι Εκθέτες

Όπως προαναφέρθηκε τα σπιν έχουν την τάση να δημιουργούν συμπλέγματα μήκους συσχετισμού  $\xi$ , το οποίο αποκλίνει για  $T \rightarrow T_c$ . Ορίζεται μια αδιάστατη παράμετρος, η ανηγμένη θερμοκρασία, η οποία μετρά την απόσταση από την  $T_c$ :

$$t = \frac{T - T_c}{T_c} \quad (4.1.1)$$

Το μήκος συσχετισμού κοντά στη μετάβαση φάσης απειρίζεται σύμφωνα με τη σχέση:

$$\xi \sim |t|^{-\nu} \quad (4.1.2)$$

Η θετική ποσότητα  $\nu$  ονομάζεται κρίσιμος εκθέτης. Η απόλυτη τιμή μας επιτρέπει να χρησιμοποιούμε τη σχέση για τιμές μεγαλύτερες ή μικρότερες της κρίσιμης θερμοκρασίας. Ο κρίσιμος εκθέτης  $\nu$  είναι μια ιδιότητα του συστήματος και είναι ανεξάρτητος του  $J$  ή του σχήματος του συστήματος (τριγωνικό, τετραγωνικό, εξαγωνικό κτλ), αλλά εξαρτάται από τις διαστάσεις (2D, 3D) του πλέγματος.

Η απόκλιση της μαγνητικής επιδεκτικότητας αλλά και της ειδικής θερμότητας δίνουν δύο ακόμα κρίσιμους εκθέτες:

$$\chi \sim |t|^{-\gamma} \quad (4.1.3)$$

$$c \sim |t|^{-\alpha} \quad (4.1.4)$$

Για τον χρόνο αυτοσυσχετισμού  $\tau$  ορίζεται ένας διαφορετικός τύπος εκθέτη που συμβολίζεται  $z$ :

$$\tau \sim |t|^{-z\nu} \quad (4.1.5)$$

όπου  $\tau$  μετρείται σε βήματα της προσομοίωσης για κάθε πλεγματική θέση.

Ο εκθέτης  $z$  ονομάζεται δυναμικός εκθέτης και μας δίνει ένα τρόπο να ποσοτικοποιήσουμε το φαινόμενο της κρίσιμης επιβράδυνσης. Κάθε αλγόριθμος έχει ένα διαφορετικό εκθέτη  $z$  και μια μεγάλη τιμή σημαίνει ότι ο χρόνος αυτοσυσχετισμού μεγαλώνει ραγδαία καθώς πλησιάζουμε την μετάβαση φάσης, κάνοντας την προσομοίωση πιο αργή και μειώνοντας την ακρίβεια της. Η μηδενική τιμή του  $z$  συνεπάγεται απουσία του φαινομένου της κρίσιμης επιβράδυνσης και ο αλγόριθμος μπορεί να χρησιμοποιηθεί μέχρι την κρίσιμη θερμοκρασία χωρίς την αύξηση του χρόνου αυτοσυσχετισμού.

Χρησιμοποιώντας τις σχέσεις (4.1.2) και (4.1.5), μπορούμε να γράψουμε την εξίσωση:

$$\tau \sim \xi^z \quad (4.1.6)$$

η οποία μας δίνει την αύξηση του χρόνου αυτοσυσχετισμού όταν το μήκος συσχετισμού πλησιάζει το κρίσιμο σημείο. Σε ένα σύστημα πεπερασμένου μεγέθους  $L$  το μεγαλύτερο σύμπλεγμα που μπορούμε να πάρουμε είναι  $L^d$  όπου  $d$  ή διάσταση του συστήματος. Αυτό σημαίνει ότι η απόκλιση του μήκους συσχετισμού, και άρα και του χρόνου αυτοσυσχετισμού, αποκόπτεται στη περιοχή για την οποία  $\xi > L$ . Άρα για όλες τις θερμοκρασίες ικανοποιητικά κοντά στη κρίσιμη θα ισχύει:

$$\tau \sim L^z \quad (4.1.7)$$

Για τον αλγόριθμό Metropolis υπολογίστηκε με γραφική επίλυση απο τους Nightingale-Blöte η τιμή  $z = 2.1665 \pm 0.0012$  η οποία είναι μια αρκετά μεγάλη τιμή που οδήγησε στην απόρριψη του αλγόριθμου Metropolis για τη μελέτη της κρίσιμης περιοχής.

## 4.2 Ο Αλγόριθμος Wolff

Ο υπολογιστικός χρόνος που χρειάζεται για την εκτέλεση συγκεκριμένου αριθμού βημάτων Monte Carlo ανά πλεγματική θέση αυξάνει όπως ο αριθμός των θέσεων, δηλαδή ως  $L^d$ , με  $d$  την διάσταση. Άρα ο υπολογιστικός χρόνος  $\tau_{CPU}$  που χρειάζεται για την μέτρηση ενός χρόνου αυτοσυσχετισμού αυξάνει με το μέγεθος του συστήματος σύμφωνα με τη σχέση:

$$\tau_{CPU} \sim L^{d+z} \quad (4.2.1)$$

όπου για το υπό μελέτη διδιάστατο πρότυπο Ising είναι περίπου  $L^4$ . Άρα οι μετρήσεις σε συστήματα πολύ μεγάλου μεγέθους είναι δύσκολες για την κρίσιμη περιοχή.

Η μεγάλη τιμή  $z$  του αλγόριθμου Metropolis οφείλεται στην απόκλιση του μήκους συσχετισμού και στις κρίσιμες διακυμάνσεις κοντά στη μετάβαση φάσης. Είναι πολύ δύσκολο για τον αλγόριθμο να αλλάξει την κατεύθυνση των σπιν για τις μεγάλες περιοχές που δημιουργούνται στη κρίσιμη περιοχή, αφού αναγκάζεται να αλλάξει ένα σπιν κάθε φορά, και υπάρχει μεγάλη πιθανότητα κάθε αλλαγή να απορριφθεί λόγω της σιδηρομαγνητικής συμπεριφοράς. Οι πιθανότητες μεταβολής ενός σπιν στο κέντρο της περιοχής είναι πολύ χαμηλές, αφού περικυκλώνεται από άλλα 4 σπιν που δείχνουν προς την ίδια κατεύθυνση. Στις δύο διαστάσεις η



αλλαγή ενός τέτοιου σπιν κοστίζει  $8J$  ενέργεια, και χρησιμοποιώντας την τιμή  $T_c \sim 2.269$  η πιθανότητα επιλογής της, όταν βρισκόμαστε κοντά στη κρίσιμη θερμοκρασία, είναι:

$$A(\mu \rightarrow \nu) \simeq e^{-8J/T_c} = 0.0294 \dots \quad (4.2.2)$$

Η πιθανότητα αλλαγής κατεύθυνσης ενός σπιν σε ακραία θέση της περιοχής είναι υψηλότερη γιατί απαιτεί χαμηλότερη ενέργεια, και αποδεικνύεται ότι αυτή η διαδικασία κυριαρχεί για την αλλαγή κατεύθυνσης σπιν ολόκληρων περιοχών. Είναι όμως μια πάρα πολύ αργή διαδικασία.

Η επίλυση αυτών των προβλημάτων βρέθηκε από τον Ulli Wolff με την πρόταση του αλγόριθμου Wolff. Η βασική ιδέα είναι η αναζήτηση συμπλεγμάτων με σπιν παρόμοιας κατεύθυνσης και η αλλαγή της κατεύθυνσης ολόκληρου του συμπλέγματος. Η χρήση του αλγορίθμου σχεδόν αναιρεί το πρόβλημα της κρίσιμης επιβράδυνσης.

Η αναζήτηση των συμπλεγμάτων γίνεται με την επιλογή ενός τυχαίου σπιν από το πλέγμα και τον έλεγχο των γειτονικών σπιν του. Ακολουθώντας ελέγχονται τα γειτονικά σπιν αυτών των γειτονικών σπιν κτλ. μέχρι τη δημιουργία ενός συμπλέγματος. Ο αριθμός των σπιν που θα αλλάξει κατεύθυνση εξαρτάται από τη θερμοκρασία του συστήματος και της συμπεριφοράς του όπως συζητήθηκε. Η προσθήκη των σπιν σε αυτό το σύμπλεγμα γίνεται με κάποια πιθανότητα  $P_{add}$  που αυξάνει με τη μείωση της θερμοκρασίας. Με την ολοκλήρωση του συμπλέγματος γίνεται η απόφαση αλλαγής της κατεύθυνσης βάση ενός λόγου αποδοχής που εξαρτάται από το ενεργειακό κόστος αυτής της μεταβολής. Το ερώτημα είναι τώρα, δεδομένης μιας πιθανότητας  $P_{add}$  ποιος είναι ο σωστός λόγος αποδοχής για να ικανοποιεί ο αλγόριθμος την συνθήκη λεπτομερούς ισορροπίας, και ποια είναι η καλύτερη επιλογή της  $P_{add}$  για να είναι ο λόγος αποδοχής πιο κοντά σε μοναδιαία τιμή;

### 4.2.1 Λόγος Αποδοχής του Αλγόριθμου Συμπλεγμάτων

Θεωρούμε δύο καταστάσεις του συστήματος  $\mu$  και  $\nu$  οι οποίες διαφέρουν κατά την εναλλαγή κατεύθυνσης ενός συμπλέγματος σπιν. Είναι σημαντική η παρατήρηση της κατεύθυνσης των σπιν γύρω από τα όρια του συμπλέγματος. Σε κάθε μια από τις δυο καταστάσεις, κάποια από τα σπιν εκτός του ορίου έχουν την ίδια κατεύθυνση με τα σπιν εσωτερικά. Οι δεσμοί ανάμεσα σε αυτά τα σπιν και τα εσωτερικά πρέπει να αποκοπούν κατά την εναλλαγή κατεύθυνσης του συμπλέγματος.

Θεωρούμε μια μετάβαση από την κατάσταση  $\mu$  στη κατάσταση  $\nu$ . Υπάρχουν πολλοί τρόποι μετάβασης, αφού θα μπορούσε να επιλεγεί ένα οποιοδήποτε σπιν εσωτερικά του συμπλέγματος και να προστεθούν τα υπόλοιπα με μια ποικιλία διατάξεων. Θα θεωρήσουμε τώρα μια συγκεκριμένη επιλογή αρχικού σπιν και την προσθήκη των επόμενων μέσω συγκεκριμένης διάταξης. Επίσης θεωρείται και η αντίστροφη μετάβαση από την  $\nu$  στη  $\mu$  με την επιλογή του ίδιου αρχικού σπιν και της ίδιας διάταξης προσθήκης σπιν. Η πιθανότητα επιλογής του ίδιου αρχικού σπιν καθώς και η πιθανότητα προσθήκης ενός νέου είναι ακριβώς ίδιες για τις δύο μεταβάσεις. Η μόνη διαφορά βρίσκεται στην πιθανότητα αποκοπής δεσμών με τα σπιν εξωτερικά του ορίου του συμπλέγματος. Θεωρούμε ότι για την κανονική μετάβαση υπάρχουν

$m$  δεσμοί που πρέπει να αποκοπούν για να γίνει η αλλαγή κατεύθυνσης του συμπλέγματος. Αυτοί οι αποκομμένοι δεσμοί αντιστοιχούν σε ζευγάρια σπιν τα οποία έχουν ίδια κατεύθυνση και τα οποία δεν προστέθηκαν στον αλγόριθμο. Η πιθανότητα να μην προστεθεί ένα τέτοιο σπιν είναι  $1 - P_{add}$ . Άρα η πιθανότητα να μην προστεθούν όλα, είναι ανάλογη της πιθανότητας μετάβασης  $g(\mu \rightarrow \nu)$  για την κανονική μετάβαση και ισούται με  $(1 - P_{add})^m$ . Αν υπάρχουν  $n$  δεσμοί που πρέπει να αποκοπούν κατά την αντίστροφη μετάβαση τότε η πιθανότητα θα είναι  $(1 - P_{add})^n$ . Η συνθήκη της λεπτομερούς ισορροπίας θα είναι τότε:

$$\frac{g(\mu \rightarrow \nu)A(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu)A(\nu \rightarrow \mu)} = (1 - P_{add})^{m-n} \frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)} \quad (4.2.3)$$

όπου  $A(\mu \rightarrow \nu)$  και  $A(\nu \rightarrow \mu)$  οι λόγοι αποδοχής για τις μεταβάσεις στις δύο καταστάσεις. Η μεταβολή στην ενέργεια  $E_\nu - E_\mu$ , ανάμεσα σε δύο καταστάσεις επίσης εξαρτάται από τους δεσμούς που αποκόπτονται. Για κάθε έναν από τους  $m$  δεσμούς η ενέργεια μεταβάλλεται κατά  $2J$  ενώ για τους  $n$  δεσμούς κατά  $-2J$ . Άρα:

$$E_\nu - E_\mu = 2J(m - n) \quad (4.2.4)$$

Με την αντικατάσταση στην εξίσωση (4.2.3) παίρνουμε τη συνθήκη για τους λόγους αποδοχής

$$\frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = [e^{2\beta J}(1 - P_{add})]^{n-m} \quad (4.2.5)$$

Παρατηρούμε ότι η επιλογή:

$$P_{add} = 1 - e^{-2\beta J} \quad (4.2.6)$$

δίνει τιμή μονάδα στο αριστερό μέλος της (4.2.5) και υπάρχει ανεξαρτησία από τις ιδιότητες κάθε κατάστασης  $\mu$  και  $\nu$ . Με αυτή την επιλογή οι λόγοι αποδοχής για την κανονική και αντίστροφη μετάβαση γίνονται μονάδα, που είναι η βέλτιστη τιμή που μπορούμε να έχουμε. Κάθε μετάβαση που επιλέγουμε γίνεται αποδεκτή και ο αλγόριθμος ικανοποιεί την συνθήκη λεπτομερούς ισορροπίας.

Η περιγραφή του αλγόριθμου Wolff είναι:

1. Επιλογή ενός αρχικού σπιν τυχαία από το πλέγμα.
2. Εξέταση των γειτονικών σπιν κατά σειρά. Αν έχουν την ίδια κατεύθυνση όπως το αρχικό σπιν: προσθήκη στο σύμπλεγμα με πιθανότητα  $P_{add} = 1 - e^{-2\beta J}$ .
3. Για κάθε σπιν που προστέθηκε στο προηγούμενο βήμα γίνεται έλεγχος των γειτονικών του σπιν και προστίθενται στο σύμπλεγμα πάλι βάση της κατεύθυνσης και της πιθανότητας του προηγούμενου βήματος. Παρατηρείται ότι όσο το σύμπλεγμα μεγαλώνει κάποια από τα γειτονικά σπιν θα έχουν ήδη ελεγχθεί και θα είναι ήδη μέλη, οπότε δεν θα προστεθούν ξανά. Κάποια από τα σπιν που έχουν απορριφθεί ωστόσο θα επανεξεταστούν για προσθήκη. Γίνεται επανάληψη της διαδικασίας μέχρι να εξεταστούν όλα τα σπιν και τα γειτονικά τους.

4. Γίνεται αλλαγή κατεύθυνσης του συμπλέγματος.

Ο αλγόριθμος ικανοποιεί επίσης τη συνθήκη της εργοδικότητας. Μπορεί να επιλεγεί ένα σπιν και να δημιουργήσει ένα σύμπλεγμα μόνο με τον εαυτό του το οποίο θα αλλάξει κατεύθυνση. Άρα είναι προσβάσιμες όλες οι πιθανές καταστάσεις του συστήματος σε πεπερασμένο χρόνο και ο αλγόριθμος επίσης θα καταλήξει μέσω μιας ακολουθίας καταστάσεων να παράγει την κατανομή Boltzmann.

Με την ολοκλήρωση της περιγραφής του αλγορίθμου Wolff είναι σημαντικό να δούμε κατά πόσο οι επιλογές που έγιναν οδηγούν σε μικρότερο δυναμικό εκθέτη αλλά και μικρότερη κρίσιμη επιβράδυνση.

### 4.3 Ιδιότητες του Αλγορίθμου Wolff

Ο αλγόριθμος Wolff αλλάζει κατεύθυνση επιτυχώς σε συμπλέγματα σπιν μεγάλου μεγέθους στη κρίσιμη περιοχή. Επίσης για  $T > T_c$  η πιθανότητα  $P_{add}$  είναι αρκετά μικρή και τα συμπλέγματα μικρού μεγέθους. Για ακόμα μεγαλύτερες θερμοκρασίες, το σύμπλεγμα τείνει να αποτελείται από ένα μόνο σπιν και άρα εναλλάσσεται μόνο αυτό, δηλαδή έχουμε παρόμοια συμπεριφορά με τη διαδικασία του αλγορίθμου Metropolis. Για αυτές τις υψηλές θερμοκρασίες ο λόγος αποδοχής του Metropolis είναι περίπου μονάδα, άρα στο όριο των υψηλών θερμοκρασιών οι δύο αλγόριθμοι έχουν ίδιες ιδιότητες. Ωστόσο σε αυτή την περιοχή θερμοκρασιών ο αλγόριθμος Wolff θα έχει μικρότερη υπολογιστική ταχύτητα λόγω του ελέγχου των γειτονικών σπιν, άρα θα ήταν προτιμότερη η χρήση του Metropolis.

Για χαμηλές θερμοκρασίες το μοντέλο *Ising* αποκτά μια πεπερασμένη μαγνήτιση συγκεκριμένης κατεύθυνσης την οποία αποκτούν όλα τα σπιν. Με τη χρήση του αλγορίθμου Wolff για αυτή την περιοχή θερμοκρασιών η επιλογή του τυχαίου σπιν και του ελέγχου των γειτονικών θα οδηγήσει στη δημιουργία ενός συμπλέγματος φιλτραρίσματος (percolating cluster) που θα εμπεριέχει σχεδόν όλο το πλέγμα. Άρα θα έχουμε μια σχεδόν πλήρη αλλαγή κατεύθυνσης των σπιν στην επόμενη κατάσταση.

Ωστόσο το μοντέλο σε αυτή την περιοχή έχει κάποιο μικρό αριθμό διεγερμένων σπιν τα οποία έχουν διαφορετική κατεύθυνση από αυτή του συμπλέγματος φιλτραρίσματος, και που αλλάζουν κατεύθυνση τυχαία ανά διαστήματα. Το ίδιο μπορεί να συμβεί φυσικά και μέσα στο σύμπλεγμα. Αυτή η συμπεριφορά προσομοιώνεται πλήρως με τη χρήση του Metropolis εφόσον κάθε φορά υπάρχει εναλλαγή ενός μόνο σπιν αλλά ο αλγόριθμος Wolff τα εξαφανίζει με την αλλαγή κατεύθυνσης σχεδόν όλου του πλέγματος. Αυτό όμως δεν σημαίνει ότι ο Wolff δεν δημιουργεί τέτοια διεγερμένα σπιν. Η πιθανότητα να μην επιλεγθεί ένα σπιν μέσα στο σύμπλεγμα είναι υπαρκτή και κατά την αλλαγή κάποια σπιν θα γίνουν διεγερμένα.

Συνοψίζοντας, μετά από ένα βήμα του αλγορίθμου Wolff στις χαμηλές θερμοκρασίες θα έχουμε μια ολοκαίνουργια διάταξη του συστήματος. Όπως και στην περιοχή των υψηλών θερμοκρασιών ο αλγόριθμος Metropolis είναι καλύτερη επιλογή λόγω της απλότητας του, αφού προσφέρει καλύτερη ταχύτητα για ίδια αποτελέσματα.

Άρα μένει για εξέταση η κρίσιμη περιοχή του συστήματος για την οποία κατασκευάστηκε ο αλγόριθμος Wolff και θα αναδειχθεί η αποτελεσματικότητά του με σύγκριση των δύο χρόνων

αυτοσυσχετισμού.

### 4.3.1 Ο Χρόνος Συσχετισμού και ο Δυναμικός Εκθέτης

Πριν τον υπολογισμό του χρόνου συσχετισμού του αλγορίθμου Wolff είναι σημαντικό να σκεφτούμε πως θα τον ορίσουμε. Ενδιαφερόμαστε για την σύγκριση των χρόνων των δύο αλγορίθμων ωστόσο δεν μπορούμε να τους συγκρίνουμε βάση του αριθμού βημάτων Monte Carlo. Ένα βήμα για τον αλγόριθμο Wolff είναι πολύ πιο περίπλοκο από αυτό του αλγορίθμου Metropolis και απαιτεί πολύ περισσότερο υπολογιστικό χρόνο αφού εναλλάσσονται πολλά σπιν.

Ο χρόνος που απαιτείται είναι ανάλογος του αριθμού των σπιν  $n$  του συμπλέγματος. Ένα σύμπλεγμα καλύπτει ένα κλάσμα  $n/L^d$  του πλέγματος, και άρα κατά μέσο όρο θα κάνει  $\langle n \rangle / L^d$  βήματα για να αλλάξει κάθε σπιν μία φορά, όπου  $\langle n \rangle$  είναι το μέσο μέγεθος του συμπλέγματος σε ισορροπία. Αυτό αντιστοιχεί σε μια σάρωση (sweep) του πλέγματος με τον Metropolis άρα ο σωστός τρόπος ορισμού του χρόνου συσχετισμού είναι  $t \propto t_{steps} \langle n \rangle / L^d$ , όπου  $t_{steps}$  είναι ο χρόνος συσχετισμού μετρημένος σε βήματα (δηλαδή εναλλαγές συμπλεγμάτων) στον αλγόριθμο Wolff. Η συμβατική επιλογή για την σταθερά της παραπάνω αναλογίας είναι η τιμή της μονάδας όπου κάνει τους χρόνους συσχετισμού των δύο αλγορίθμων ίσους στα όρια χαμηλής και υψηλής θερμοκρασίας. Ωστόσο, αν και ο Metropolis είναι ταχύτερος σε αυτά τα όρια υπάρχει τεράστια διαφορά για την κρίσιμη περιοχή, οπότε έχουμε:

$$\tau = \tau_{steps} \frac{\langle n \rangle}{L^d} \quad (4.3.1)$$

Είναι πια δυνατό να συγκρίνουμε τους δύο αλγορίθμους στην κρίσιμη περιοχή. Για ένα  $100 \times 100$  σύστημα στη κρίσιμη θερμοκρασία, ο χρόνος συσχετισμού του αλγορίθμου Wolff ισοδυναμεί με  $\tau = 2.80 \pm 0.03$  εναλλαγές σπιν ανά πλεγματική θέση ενώ για τον αλγόριθμο Metropolis  $\tau = 2570 \pm 330$ . Αυτή η διαφορά είναι δραματική, άρα προβλέπεται η χρήση του αλγορίθμου Wolff για τη μελέτη αυτής της περιοχής.

### 4.3.2 Ο Δυναμικός Εκθέτης και η Επιδεκτικότητα

Κατά την μελέτη του μοντέλου Ising με τον αλγόριθμο Wolff μπορούμε να ορίσουμε έναν δυναμικό εκθέτη  $z_{steps}$  συναρτήσει του χρόνου συσχετισμού  $\tau_{steps}$ :

$$\tau_{steps} \sim \xi^{z_{steps}} \quad (4.3.2)$$

Ο εκθέτης σχετίζεται με την πραγματικό εκθέτη δυναμικής  $z$  του αλγορίθμου σύμφωνα με τη σχέση:

$$z = z_{steps} + \frac{\gamma}{\nu} - d \quad (4.3.3)$$

όπου  $\gamma$  και  $\nu$  είναι οι κρίσιμοι εκθέτες των αποκλίσεων της μαγνητικής επιδεκτικότητας και του μήκους συσχετισμού (4.1.2), (4.1.3) και  $d$  η διάσταση του προβλήματος. Για την περίπτωση του 2D Ising μοντέλου, επειδή η λύση είναι γνωστή, οι τιμές των  $\nu$  και  $\gamma$  είναι γνωστές και

άρα ο υπολογισμός του  $z$  μπορεί να γίνει αποφεύγοντας τον υπολογισμό του μέσου μεγέθους συμπλέγματος και άρα μειώνοντας το σφάλμα.

Η μαγνητική επιδεκτικότητα  $\chi$ , για θερμοκρασίες  $T \geq T_c$ , σχετίζεται με το μέσο μέγεθος  $\langle n \rangle$  σύμφωνα με τη σχέση:

$$\chi = \beta \langle n \rangle \quad (4.3.4)$$

Άρα υπάρχει η επιλογή υπολογισμού της μαγνητικής επιδεκτικότητας μέσω του μέσου μεγέθους των συμπλεγμάτων όταν γίνεται χρήση του αλγορίθμου Wolff.



## Κεφάλαιο 5

# Ανάλυση Δεδομένων

### 5.1 Εισαγωγή

Είναι σημαντικό πριν την παρουσίαση της multiple histogram να γίνει μια αναφορά στις μεθόδους που θα χρησιμοποιηθούν για την εκτίμηση σφαλμάτων. Ο λόγος για τον οποίο πρέπει να αναφερθούν σε αυτό το σημείο είναι επειδή και οι προσομοιώσεις Monte Carlo που εκτελούνται με χρήση των αλγορίθμων Metropolis και Wolff και από τις οποίες θα εξαχθούν μετρήσεις ενέργειας και μαγνήτισης, διέπονται από στατιστικά και συστηματικά σφάλματα.

### 5.2 Υπολογισμός Σφαλμάτων

Στο πρόβλημα μας υπάρχουν τόσο στατιστικά όσο και συστηματικά σφάλματα. Τα στατιστικά σφάλματα εμφανίζονται από τυχαίες μεταβολές στο προσομοιωμένο σύστημα ανάμεσα σε μετρήσεις (π.χ θερμικές διακυμάνσεις) και μπορούν να εκτιμηθούν με την πολλαπλή μέτρηση της ποσότητας που μας ενδιαφέρει. Τα συστηματικά σφάλματα εμφανίζονται από τη διαδικασία που χρησιμοποιείται για τη μέτρηση των ποσοτήτων, και επηρεάζουν ολόκληρη την προσομοίωση (για παράδειγμα ένα σφάλμα υπεισέρχεται από την αναμονή μέχρι το σύστημα να φτάσει σε ισορροπία).

#### 5.2.1 Εκτιμήσεις Στατιστικών Σφαλμάτων

Στις προσομοιώσεις Monte Carlo η κύρια πηγή στατιστικού σφάλματος σε μία μετρήσιμη ποσότητα είναι η διακύμανση της ποσότητας από ένα χρονικό βήμα στο επόμενο. Είναι συνήθως άμεση η δυνατότητα υπολογισμού του στατιστικού σφάλματος της παρατηρήσιμης ποσότητας αφού σε μια τέτοια περίπτωση μπορεί να υπολογιστεί η μέση τιμή της ποσότητας και το σφάλμα της.

Άρα για μια προσομοίωση στο Ising με  $n$  μετρήσεις της μαγνήτισης  $m$ , η καλύτερη εκτίμηση της παρατηρήσιμης ποσότητας είναι η μέση τιμή  $\bar{m}$  (που αντιστοιχεί στον εκτιμητή) και η καλύτερη εκτίμηση της τυπικής απόκλισης δίνεται από τη σχέση:

$$\sigma = \sqrt{\frac{\frac{1}{n} \sum_{i=0}^n (m_i - \bar{m})^2}{n-1}} = \sqrt{\frac{1}{n-1} (\overline{m^2} - \bar{m}^2)} \quad (5.2.1)$$

Ωστόσο για τη σχέση αυτή υποθέτουμε ότι τα δείγματα  $m$  είναι στατιστικά ανεξάρτητα, κάτι το οποίο δεν συμβαίνει στην περίπτωση που μελετάμε αφού οι μετρήσεις γίνονται σε χρόνο μικρότερο του χρόνου αυτοσυσχετισμού. Ένας τρόπος επίλυσης του προβλήματος είναι με χρήση της εξίσωσης :

$$\sigma = \sqrt{\frac{1 + 2\tau/\Delta t}{n-1} (\overline{m^2} - \bar{m}^2)} \quad (5.2.2)$$

με  $\tau$  τον χρόνο αυτοσυσχετισμού και  $\Delta t$  το χρονικό διάστημα στο οποίο πήραμε τα δείγματα. Για  $\Delta t \gg \tau$  έχουμε την εξίσωση (5.2.1), αλλά συνήθως ισχύει  $\Delta t \ll \tau$  οπότε μπορεί να αγνοηθεί ο όρος 1 στον αριθμητή της (5.2.2). Για μια εκτέλεση  $t_{max}$  αφού το σύστημα βρεθεί σε ισορροπία, το  $\Delta t$  σχετίζεται με τον ολικό αριθμό των δειγμάτων μέσω της σχέσης:

$$n = \frac{t_{max}}{\Delta t} \quad (5.2.3)$$

Για μεγάλα  $n$  θα ισχύει:

$$\sigma = \sqrt{\frac{2\tau}{t_{max}} (\overline{m^2} - \bar{m}^2)} \quad (5.2.4)$$

Σημειώνεται ότι η σωστή επιλογή χρονικού διαστήματος για την λήψη ανεξάρτητων μετρήσεων είναι δύο φορές ο χρόνος αυτοσυσχετισμού. Επίσης είναι σημαντικό να αναφερθεί ότι η τιμή  $s$  της εξίσωσης (5.2.4) είναι ανεξάρτητη του χρονικού διαστήματος  $\Delta t$ , άρα μπορούμε να επιλέξουμε οποιοδήποτε  $\Delta t$  συμφέρει.

### 5.2.2 Η μέθοδος Blocking

Υπάρχουν περιπτώσεις για τις οποίες δεν είναι δυνατή η εκτίμηση του σφάλματος όπως συζητήθηκε στο προηγούμενο υποκεφάλαιο. Μια τέτοια περίπτωση για παράδειγμα είναι η μέτρηση για την ειδική θερμότητα  $c$  η οποία δεν υπολογίζεται απευθείας ως μια μέση τιμή αλλά εξάγεται από μετρήσεις που έχουν γίνει κατά τη διάρκεια της προσομοίωσης. Η ειδική θερμότητα ορίζεται συναρτήσει των μέσων τιμών των  $E$  και  $E^2$ , τα σφάλματα των οποίων είναι συσχετισμένα και αυξάνονται μαζί. Η μέθοδος Blocking ή Binning μας δίνει μια πρώτη καλή εκτίμηση των σφαλμάτων των ποσοτήτων με ένα απλό τρόπο και μπορεί να εφαρμοστεί σε οποιαδήποτε ποσότητα, όπως και θα συμβεί κατά την υλοποίηση της multiple histogram.

Η ιδέα πίσω από την Blocking είναι ο χωρισμός των μετρήσεων της ενέργειας  $E$  που ε-ξάχθηκαν κατά την εκτέλεση της προσομοίωσης σε μέρη, ονομαζόμενα blocks. Για παράδειγμα για μια προσομοίωση 1.000.000 μετρήσεων θα γίνει χωρισμός σε 10 blocks των 100.000 μετρήσεων. Μετά γίνεται υπολογισμός της ειδικής θερμότητας ξεχωριστά για κάθε block, άρα καταλήγουμε με 10 διαφορετικές τιμές της ειδικής θερμότητας και η μέση τιμή υπολογίζεται ακριβώς όπως και για την μαγνήτιση στο προηγούμενο υποκεφάλαιο. Μετά γίνεται υπολογισμός του σφάλματος της μέσης τιμής με χρήση της εξίσωσης (5.2.1) όπου η τιμή  $n$  αντικαθίσταται



από τον αριθμό  $n_b$  των blocks που είναι 10 για την περίπτωση που συζητήθηκε. Οι εκτιμήσεις αυτές διαφέρουν σύμφωνα με τον αριθμό των blocks που έχουν επιλεγεί. Η απλότητα της εφαρμογής της μεθόδου, την κάνει ιδανική για μια πρώτη καλή εκτίμηση της τάξης σφαλμάτων του προβλήματος.

### 5.2.3 Η μέθοδος Bootstrap

Για την εφαρμογή της πιο γενικής μεθόδου bootstrap έστω ότι έχουμε  $n$  μετρήσεις σε μια προσομοίωση. Τότε θα εξάγουμε από αυτές τις  $n$  μετρήσεις με τυχαίο τρόπο  $n$  νέες μετρήσεις. Κανονικά αυτές οι νέες μετρήσεις θα έπρεπε να επιλεγθούν κατά στατιστικά ανεξάρτητο τρόπο, ωστόσο πρακτικά και τουλάχιστον για την συγκεκριμένη περίπτωση που εξετάζεται η χρήση μη στατιστικά ανεξάρτητων μετρήσεων δεν επηρεάζει σημαντικά τη μέθοδο. Από αυτές τις νέες  $n$  μετρήσεις που επιλέγονται τυχαία είναι δυνατή η επιλογή της ίδιας τιμής παραπάνω από μία φορές. Περίπου ένα ποσοστό  $1 - 1/e \approx 63\%$  θα είναι πανομοιότυπες μετρήσεις. Ακολούθως γίνεται ο υπολογισμός της ειδικής θερμότητας ακριβώς όπως έχει συζητηθεί αλλά με τη χρήση των νέων δεδομένων, και μετά επαναλαμβάνεται η διαδικασία της τυχαίας επιλογής και της δημιουργίας ενός νέου συνόλου δεδομένων. Αποδείχθηκε από τον Efron(1979) ότι με την εκτέλεση αυτής της διαδικασίας αρκετές φορές, η τυπική απόκλιση των τελικών μετρήσεων της ειδικής θερμότητας είναι μια μέτρηση του σφάλματος της. Με άλλα λόγια η εφαρμογή πολλαπλών bootstrap υπολογισμών της ειδικής θερμότητας θα δώσει το σφάλμα σύμφωνα με τη σχέση:

$$\sigma = \sqrt{c^2 - \bar{c}^2} \quad (5.2.5)$$

Παρατηρούμε ότι δεν υπάρχει όρος  $1/(n - 1)$  αφού αυτό θα σήμαινε ότι με την εφαρμογή της μεθόδου περισσότερες φορές η εκτίμηση του σφάλματος θα ήταν ένας μικρότερος αριθμός.

Όπως αναφέρθηκε, για την εφαρμογή της μεθόδου δεν είναι απαραίτητο οι μετρήσεις να είναι στατιστικά ανεξάρτητες, δηλαδή μετρημένες σε διάστημα δύο χρόνων αυτοσυσχετισμού. Η τελική εκτίμηση του σφάλματος θα είναι ουσιαστικά ίδια σε κάθε περίπτωση καθώς κατά την αναδειγματοληψία θα έχουμε πανομοιότυπες τιμές τόσο στατιστικά ανεξάρτητων όσο και μη στατιστικά ανεξάρτητων μετρήσεων. Μια αρχική επιλογή αριθμού αναδειγματοληψίας με την bootstrap για μια εκτίμηση σφάλματος είναι 100, η οποία θα έδινε ένα σφάλμα ακρίβειας περίπου 10%. Ωστόσο ακόμα και η τιμή 1000 δεν μπορεί να θεωρηθεί υπερβολική και συνήθως επιλέγονται τιμές πιο κοντά στη 1000. Η μέθοδος bootstrap είναι αξιόπιστη γενική μέθοδος εκτίμησης σφαλμάτων σε προσομοιώσεις Monte Carlo.

### 5.2.4 Η μέθοδος Jackknife

Για την εφαρμογή της μεθόδου jackknife χρειαζόμαστε ένα σύνολο  $n$  στατιστικά ανεξάρτητων μετρήσεων. Για την περίπτωση της ειδικής θερμότητας λοιπόν, θα υπολογιστεί  $n$  φορές παίρνοντας μια τιμή  $c_i$  με τον ακόλουθο τρόπο. Αρχικά για το σύνολο των  $n$  μετρήσεων θα αφαιρεθεί η πρώτη μέτρηση, άρα θα μείνουν  $n - 1$  και θα γίνει ο υπολογισμός της  $c_1$  σύμφωνα με αυτό το υποσύνολο. Μετά επανατοποθετείται η πρώτη μέτρηση και αφαιρείται η δεύτερη

υπολογίζοντας την  $c_2$  από το νέο υποσύνολο. Κάθε  $c_i$  είναι η ειδική θερμότητα υπολογισμένη με την  $i$  μέτρηση της ενέργειας αφαιρεμένη από το σύνολο των μετρήσεων, το οποίο καταλήγει να αποτελείται από  $n - 1$  μετρήσεις. Η εκτίμηση του σφάλματος της ειδικής θερμότητας είναι τότε:

$$\sigma = \sqrt{\sum_{i=1}^n (c_i - c)^2} \quad (5.2.6)$$

όπου  $c$  είναι η εκτίμηση της ειδικής θερμότητας με χρήση ολόκληρου του συνόλου των μετρήσεων.

Οι μέθοδοι jackknife και bootstrap δίνουν καλές εκτιμήσεις σφαλμάτων για μεγάλα σύνολα δεδομένων, και όταν το μέγεθος αυτών των συνόλων τείνει στο άπειρο δίνουν ακριβείς εκτιμήσεις. Η επιλογή σε κάθε περίπτωση βασίζεται στο ποια από τις δύο είναι ευκολότερα εφαρμόσιμη στο πρόβλημα. Για την jackknife η ποσότητα πρέπει να υπολογιστεί ξανά  $n$  φορές οπότε αν ο αριθμός  $n$  είναι μεγαλύτερος του αριθμού αναδειγματοληψίας που αναφέρθηκε στο προηγούμενο υποκεφάλαιο τότε επιλέγεται η bootstrap. Αλλιώς επιλέγεται η jackknife.

### 5.2.5 Συστηματικά Σφάλματα

Η αντιμετώπιση και η παρατήρηση των συστηματικών σφαλμάτων είναι αρκετά δυσκολότερη αφού αυτά δεν εμφανίζονται στις διακυμάνσεις των παρατηρήσιμων ποσοτήτων. Η κύρια πηγή προσθήκης συστηματικού σφάλματος στη προσομοίωση του προτύπου Ising είναι η αναμονή πεπερασμένου χρόνου για να οδηγηθεί το σύστημα σε ισορροπία. Δεν υπάρχει κάποια γενική μέθοδος αντιμετώπισης των συστηματικών σφαλμάτων και κάθε πηγή τέτοιου σφάλματος οφείλει να αντιμετωπιστεί ξεχωριστά. Κατά μία έννοια αυτό συνέβη ήδη κατά τη συζήτηση για το χρόνο που χρειάζεται το σύστημα να βρεθεί σε ισορροπία καθώς και για το χρόνο αυτοσυσχετισμού αφού ο περιορισμένος χρόνος που εκτελείται η προσομοίωση μετά την ισορροπία δεν επιτρέπει τη λήψη αρκετών στατιστικά ανεξάρτητων μετρήσεων.

## 5.3 Η Μέθοδος Single Histogram

Πριν την παρουσίαση της μεθόδου multiple histogram είναι σημαντική η αναφορά στη μέθοδο single histogram. Η single histogram είναι μια τεχνική όπου με χρήση των δεδομένων μιας Monte Carlo προσομοίωσης για μία δεδομένη θερμοκρασία στο Ising πρότυπο, επιτρέπει την εκτίμηση παρατηρήσιμων ποσοτήτων για κοντινές θερμοκρασίες. Η μέθοδος, η οποία είναι πάρα πολύ απλή στην υλοποίηση της, οδηγεί στην εκτίμηση ποσοτήτων με πολύ καλή προσέγγιση χωρίς την ανάγκη εκτέλεσης μιας νέας προσομοίωσης για την εξαγωγή τους.

Όπως αναφέρθηκε ο εκτιμητής μιας παρατηρήσιμης ποσότητας  $Q$  από  $M$  μετρήσεις  $Q_\mu$  μιας προσομοίωσης δίνεται από τη σχέση:

$$Q_M = \frac{\sum_{i=1}^M Q_{\mu_i} p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}{\sum_{j=1}^M p_{\mu_j}^{-1} e^{-\beta E_{\mu_j}}} \quad (5.3.1)$$

όπου  $p_{\mu_i}$  οι πιθανότητες Boltzmann για δεδομένη θερμοκρασία. Υποθέτουμε λοιπόν ότι οι  $p_{\mu_i}$  είναι οι πιθανότητες μιας διαφορετικής θερμοκρασίας  $\beta_0 = 1/\kappa T_0$  η οποία βρίσκεται αρκετά κοντά στην αρχική και ισούνται με:

$$p_{\mu_i} = \frac{1}{Z_0} e^{-\beta_0 E_{\mu_i}} \quad (5.3.2)$$

όπου  $Z_0$  είναι η συνάρτηση επιμερισμού για τη θερμοκρασία  $T_0$ . Η αντικατάσταση στην (5.3.1) θα δώσει:

$$Q_M = \frac{\sum_{i=1}^M Q_{\mu_i} e^{-(\beta-\beta_0)E_{\mu_i}}}{\sum_{j=1}^M e^{-(\beta-\beta_0)E_{\mu_j}}} \quad (5.3.3)$$

Η (5.3.3) είναι η εξίσωση της single histogram στην απλούστερη μορφή της. Επιτρέπει μέσω των μετρήσεων  $Q_{\mu_i}$  της παρατηρήσιμης ποσότητας  $Q$  μιας προσομοίωσης σε θερμοκρασία  $\beta_0$ , την εξαγωγή της αναμενόμενης τιμής  $Q$  σε μια νέα θερμοκρασία  $\beta$ . Είναι σημαντικό να σημειωθεί ότι οι ενέργειες  $E_{\mu_i}$  που εμφανίζονται στην εξίσωση είναι ολικές ενέργειες των καταστάσεων  $\mu_i$  και όχι ενέργειες ανά σπιν.

Η εκτίμηση των σφαλμάτων μπορεί να γίνει με οποιαδήποτε από τις μεθόδους αναφέρθηκαν στο κεφάλαιο αυτό. Για τις νέες τιμές που προκύπτουν από την παραπάνω επέκταση (extrapolation) μπορεί να γίνει επεξεργασία ακριβώς με τις ίδιες τεχνικές όπως και με τις μετρήσεις των προσομοιώσεων.

Για την περίπτωση της εσωτερικής ενέργειας, οι μετρήσεις  $Q_{\mu_i}$  της (5.3.3) αντιστοιχούν στις  $E_{\mu_i}$ , μετρήσεις της χαμιλτονιανής του συστήματος. Η εξίσωση (5.3.3) μπορεί να γραφεί τότε:

$$U = \langle E \rangle = \frac{\sum_E E N(E) e^{-(\beta-\beta_0)E}}{\sum_E N(E) e^{-(\beta-\beta_0)E}} \quad (5.3.4)$$

όπου το άθροισμα είναι τώρα πάνω σε όλες τις πιθανές ενέργειες  $E$  των καταστάσεων του συστήματος και  $N(E)$  είναι ο αριθμός που εμφανίζεται η ενέργεια  $E$  στο σύνολο των καταστάσεων. Άρα μέσω των  $N(E)$  μπορεί να δημιουργηθεί το ιστόγραμμα των ενεργειών του συστήματος που έχει δειγματιστεί. Είναι πολύ σημαντικό να σημειωθεί ότι η χρήση ιστογραμμάτων δεν είναι απαραίτητη αφού μπορεί να γίνει χρήση απευθείας της (5.3.3). Η εξίσωση (5.3.3) ωστόσο απαιτεί την επεξεργασία σε όλες τις μετρήσεις  $Q_{\mu_i}$  και άρα η μέθοδος χάνει σε ταχύτητα όσο το δείγμα μεγαλώνει ενώ η δημιουργία του ιστογράμματος οδηγεί σε καλύτερες ταχύτητες. Συγκεκριμένα για το υπό μελέτη πρόβλημα του προτύπου Ising η χρήση ιστογραμμάτων ενδείκνυται και με αυτό τον τρόπο θα υλοποιηθούν οι μέθοδοι.

Ένας παράγοντας που προσθέτει σφάλμα και πρέπει να μελετηθεί είναι το επιτρεπτό εύρος επέκτασης θερμοκρασιών. Για μεγάλη απόσταση η εκτίμηση θα χάνει την αξιοπιστία της. Άρα πρέπει να βρεθεί ένας τρόπος υπολογισμού του επιτρεπτού εύρους επέκτασης με χρήση των δεδομένων της ήδη υπάρχουσας προσομοίωσης στη θερμοκρασία  $T_0$ . Θεωρούμε την εξίσωση (5.3.4) ως:

$$U = \sum_E W(E) \quad (5.3.5)$$

με τα κατάλληλα κανονικοποιημένα βάρη  $W(E)$  που εξαρτώνται από τη διαφορά θερμοκρασιών  $\beta - \beta_0$ :

$$W(E) = \frac{N(E)e^{-(\beta-\beta_0)E}}{\sum_E N(E)e^{-(\beta-\beta_0)E}} \quad (5.3.6)$$

Τα βάρη  $W(E)$  είναι ανάλογα των  $N(E)$  για τη θερμοκρασία  $\beta = \beta_0$  όπου βρίσκεται η αρχική προσομοίωση. Ουσιαστικά η μέθοδος single histogram προσπαθεί να εκτιμήσει ποια θα ήταν τα ιστογράμματα των ενεργειών σε μια θερμοκρασία  $T$  σύμφωνα με τα ιστογράμματα στην θερμοκρασία  $T_0$ , βαραίνοντας κατάλληλα τα bins των ιστογραμμάτων με εκθετικούς όρους που εξαρτώνται από τη διαφορά θερμοκρασιών. Ωστόσο όπως προαναφέρθηκε για το εύρος ενεργειών στο πρότυπο Ising, το σύστημα επιλέγει να βρίσκεται σε ένα μικρό σύνολο καταστάσεων. Με την απομάκρυνση από το κέντρο αυτού του εύρους τα ιστογράμματα θα καταλήξουν να είναι μηδενικά και κοντά στα όρια θα έχουν πολύ μικρές τιμές με το σφάλμα να γίνεται μεγαλύτερο.

Ποια είναι λοιπόν η διαφορά θερμοκρασίας  $\Delta T = T - T_0$  όπου μπορούμε να επεκτείνουμε και να έχουμε αξιόπιστα αποτελέσματα: Το κέντρο της κατανομής  $W(E)$  που δίνεται προσεγγιστικά από την εσωτερική ενέργεια  $\langle E \rangle$ , εξίσωση (5.3.5), θα πρέπει να βρίσκεται στο εύρος όπου τα  $N(E)$  θα είναι αρκετά μεγαλύτερα της μονάδας. Αν αναπαραστήσουμε αυτό το εύρος με την τυπική απόκλιση  $\sigma_E$  της  $N(E)$ , τότε το κριτήριο μας θα είναι:

$$|U(T) - U(T_0)| \leq \sigma_E \quad (5.3.7)$$

Το τετράγωνο της τυπικής απόκλισης  $\sigma_E^2$  είναι ανάλογο της ειδικής θερμότητας  $C$  σε θερμοκρασία  $T_0$  άρα μπορούμε να γράψουμε το κριτήριο συναρτήσει των θερμοδυναμικών ποσοτήτων  $U$  και  $C$ :

$$|U(T) - U(T_0)|^2 \leq \frac{C(T_0)}{\beta_0^2} \quad (5.3.8)$$

Συνεχίζοντας με την προσέγγιση:

$$U(T) - U(T_0) \approx \left. \frac{dU}{dT} \right|_{T_0} (T - T_0) = C(T_0)\Delta T \quad (5.3.9)$$

η επιτρεπτή απόσταση  $\Delta T$  για την οποία μπορούμε να επεκτείνουμε μακριά από την  $T_0$  δίνεται από τη σχέση:

$$\left[ \frac{\Delta T}{T_0} \right]^2 = \frac{1}{C(T_0)} \quad (5.3.10)$$

Σημειώνεται ότι η παραπάνω ποσότητα αποτελεί μια πολύ συντηρητική επιλογή και για το συνηθισμένο αριθμό δειγμάτων (π.χ μεγαλύτερων του ενός εκατομμυρίου) μπορούμε να πάρουμε καλές εκτιμήσεις για το διπλάσιο της (5.3.10).

Η μέθοδος single histogram μπορεί να χρησιμοποιηθεί και για την εκτίμηση και άλλων παρατηρήσιμων ποσοτήτων, με χρήση της (5.3.3). Το επιτρεπτό εύρος επέκτασης (5.3.10)

συνεχίζει να ισχύει και για άλλες παρατηρήσιμες ποσότητες. Ωστόσο η μόνη διαφορά βρίσκεται στην περίπτωση χρήσης ιστογραμμάτων όπου η σωστή γενίκευση είναι:

$$Q_M = \frac{\sum_{E,Q} Q N(E, Q) e^{-(\beta-\beta_0)E}}{\sum_{E,Q} N(E, Q) e^{-(\beta-\beta_0)E}} \quad (5.3.11)$$

όπου  $N(E, Q)$  είναι το διδιάστατο ιστόγραμμα της ενέργειας  $E$  και της παρατηρήσιμης ποσότητας  $Q$ . Στην περίπτωση της μαγνήτισης  $M$  για παράδειγμα το ιστόγραμμα  $N(E, M)$  θα αντιστοιχούσε σε πόσες φορές εμφανίζεται μια κατάσταση με ενέργεια  $E$  και μαγνήτιση  $M$ . Ωστόσο η δημιουργία διδιάστατων ιστογραμμάτων αποφεύγεται γιατί οδηγεί σε πολύ μεγάλο αριθμό bins για ένα σύστημα μεγάλου μεγέθους, και συνήθως επιλέγεται η αναλυτική επίλυση.

Κατά την υλοποίηση της single histogram και καθώς το σύστημα μεγαλώνει σε μέγεθος είναι πιθανό να προκύψουν αριθμοί οι οποίοι ξεφεύγουν από τα όρια αναπαράστασης ενός υπολογιστή λόγω της εκθετικής εξάρτησης από την ενέργεια  $E$ .

### 5.3.1 Επέκταση σε Διαφορετικές Μεταβλητές

Η μέθοδος single histogram μπορεί να χρησιμοποιηθεί με επέκταση σε άλλες μεταβλητές αντί της θερμοκρασίας. Η χαμιλτονιανή μπορεί να γραφεί ως:

$$H = \sum_k J^{(k)} E^{(k)} \quad (5.3.12)$$

κάνοντας δυνατή την επέκταση σε οποιαδήποτε από τις μεταβλητές  $J^{(k)}$ . Οι μεταβλητές  $J^{(k)}$  είναι οι σταθερές σύζευξης, αντίστοιχες της παραμέτρου  $J$  στο πρότυπο Ising, και  $E^{(k)}$  είναι οι αδιάστατες συναρτήσεις των σπιν ή άλλων βαθμών ελευθερίας ανάλογα με το μοντέλο. Για ένα εξωτερικό μαγνητικό πεδίο η χαμιλτονιανή του συστήματος είναι:

$$H = -J \sum_{\langle ij \rangle} s_i s_j - B \sum_i s_i \quad (5.3.13)$$

όπου αντιστοιχίζοντας με την (5.3.12) έχουμε:

$$J^{(1)} = -J \quad E^{(1)} = \sum_{\langle ij \rangle} s_i s_j \quad (5.3.14)$$

$$J^{(2)} = -B \quad E^{(2)} = \sum_i s_i \quad (5.3.15)$$

Μπορούμε να επεκτείνουμε σε οποιαδήποτε από τις μεταβλητές  $J^{(k)}$  με μια απλή γενίκευση της (5.3.3). Η εξίσωση (5.3.1) γράφεται:

$$Q_M = \frac{\sum_{i=1}^M Q_{\mu_i} p_{\mu_i}^{-1} \exp(-\beta \sum_k J^{(k)} E_{\mu_i}^{(k)})}{\sum_{j=1}^M p_{\mu_j}^{-1} \exp(-\beta \sum_k J^{(k)} E_{\mu_j}^{(k)})} \quad (5.3.16)$$

Υποθέτουμε ότι θέλουμε να υπολογίσουμε τα  $Q_M$  σε κάποιο σημείο του διαστήματος  $J^{(k)}$  αλλά η προσομοίωση είχε εκτελεσθεί για το σημείο  $J_0^{(k)}$ . Τότε οι πιθανότητες  $p_{\mu_i}$  δίνονται από την:

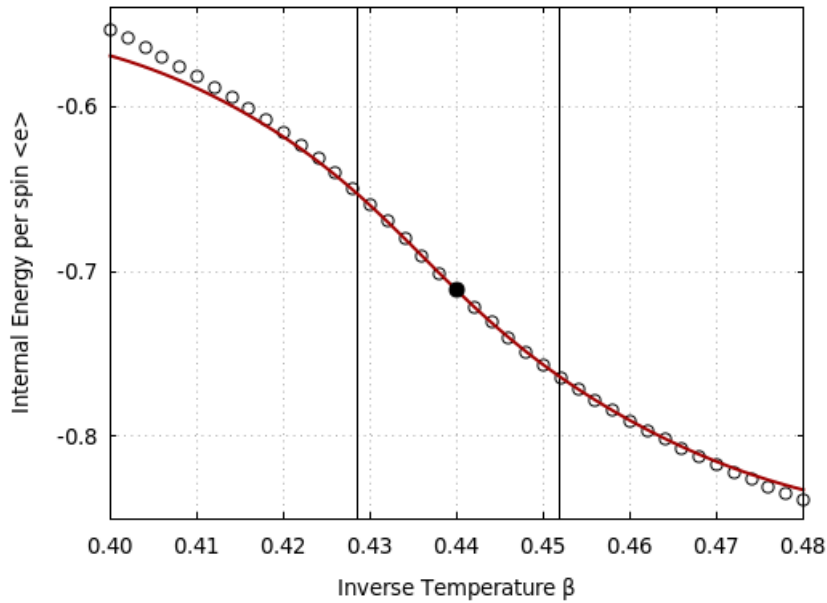
$$p_{\mu_i} = \frac{1}{Z_0} \exp(-\beta \sum_k J_0^k E_{\mu_i}^k) \quad (5.3.17)$$

όπου με αντικατάσταση στην εξίσωση (5.3.16) έχουμε:

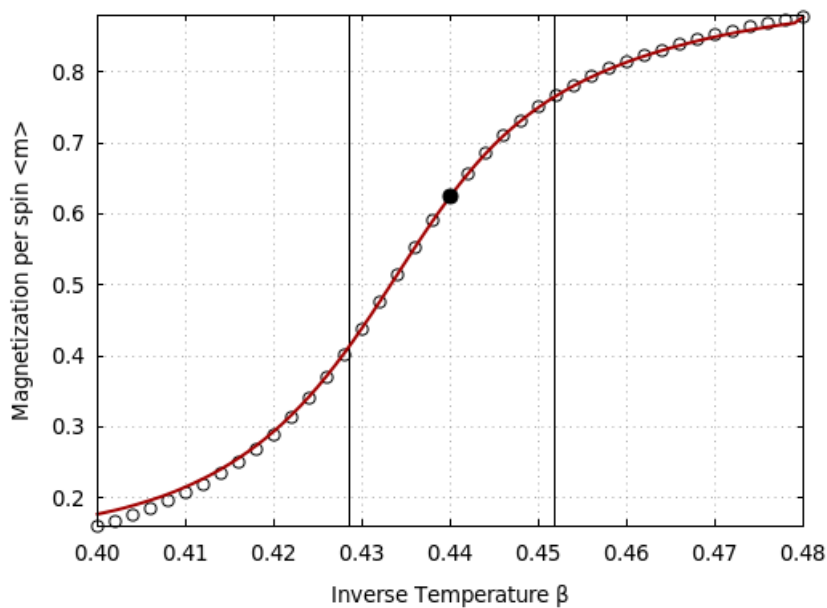
$$Q_M = \frac{\sum_{i=1}^M Q_{\mu_i} \exp\left[-\beta \sum_k (J^{(k)} - J_0^{(k)}) E_{\mu_i}^{(k)}\right]}{\sum_{j=1}^M \exp\left[-\beta \sum_k (J^{(k)} - J_0^{(k)}) E_{\mu_j}^{(k)}\right]} \quad (5.3.18)$$

όπου αποτελεί τη σωστή γενίκευση της (5.3.3) για την επέκταση σε διαφορετικές μεταβλητές. Σημειώνεται ότι σε αυτή την περίπτωση δεν αρκεί μόνο η αποθήκευση των τιμών της ενέργειας  $E^{(k)}$  κάθε κατάστασης.

Διάγραμμα της εσωτερικής ενέργειας ανά σπιν  $\langle e \rangle$  συναρτήσει της θερμοκρασίας  $\beta$  για  $L = 40$

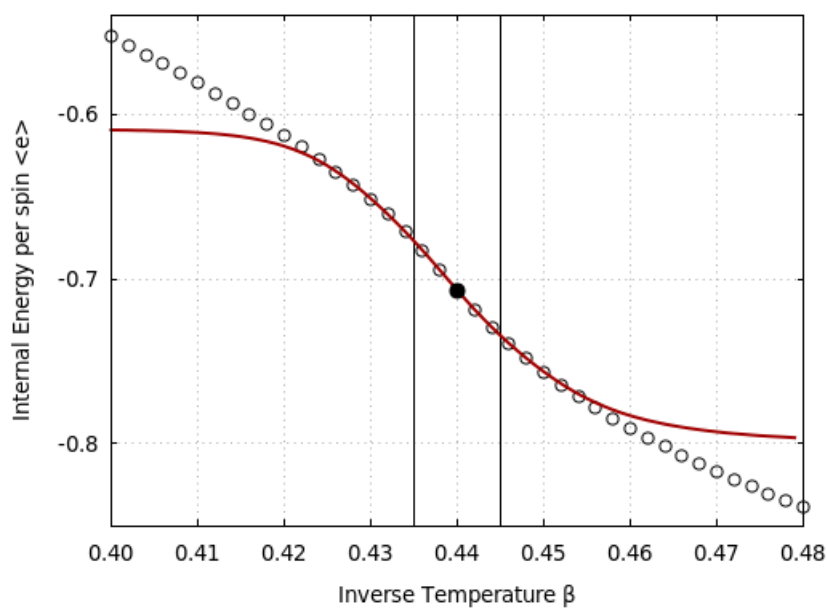


Διάγραμμα της μαγνήτισης ανά σπιν  $\langle m \rangle$  συναρτήσει της θερμοκρασίας  $\beta$  για  $L = 40$

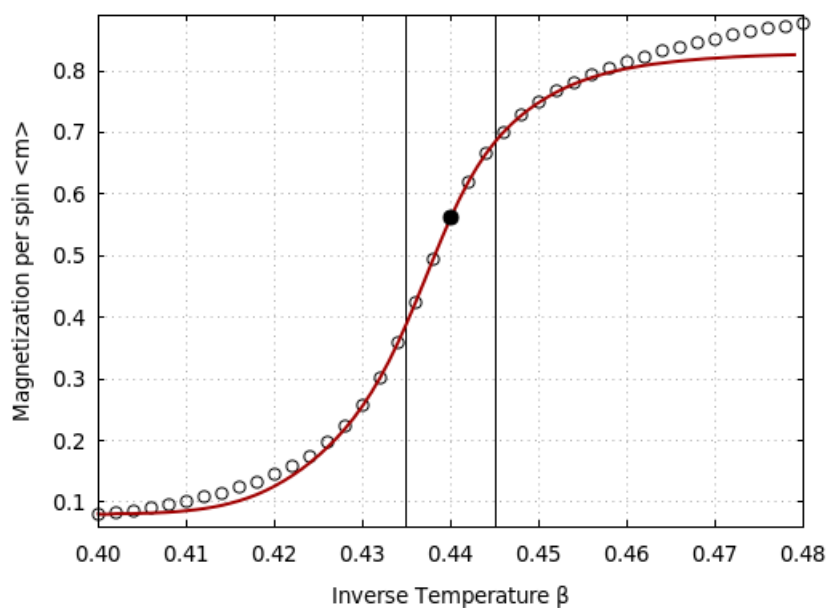


Σχήμα 5.1: Εφαρμογή της μεθόδου single histogram (ευθεία σχήματος) για πλέγμα μεγέθους  $N = L * L = 40 * 40 = 1600$  και σύγκριση με δεδομένα που προκύπτουν από απευθείας υπολογισμούς με εκτελέσεις προσομοιώσεων Monte Carlo (σχήμα ο). Ο συμπαγής κύκλος δηλώνει την προσομοίωση που χρησιμοποιήθηκε για την επέκταση (extrapolation) των δεδομένων. Οι δύο κάθετες ευθείες είναι το επιτρεπτό εύρος επέκτασης όπως δίνεται μέσω της σχέσης (5.3.10). Παρατηρούμε ότι η μέθοδος χάνει την αξιοπιστία της όσο μεγαλώνει η απόσταση.

Διάγραμμα της εσωτερικής ενέργειας ανά σπιν  $\langle e \rangle$  συναρτήσει της  $\beta$  για  $L = 80$



Διάγραμμα της μαγνήτισης ανά σπιν  $\langle m \rangle$  συναρτήσει της  $\beta$  για  $L = 80$



Σχήμα 5.2: Ανάλογο σχήμα με το 5.1 για  $N = L * L = 80 * 80 = 6400$ . Παρατηρούμε ότι με την αύξηση του μεγέθους του πλέγματος το επιτρεπτό εύρος επέκτασης μειώνεται, κάτι που είναι αναμενόμενο λόγω της μεταβολής της τιμής της ειδικής θερμότητας για μεγαλύτερα πλέγματα και της σχέσης (5.3.10).



## 5.4 Η Μέθοδος Multiple Histogram

Με τη μέθοδο single histogram το διάστημα των θερμοκρασιών στο οποίο μπορούμε να εκτιμήσουμε παρατηρήσιμες ποσότητες  $Q$  είναι κάπως περιορισμένο. Το επόμενο λογικό βήμα είναι η προσπάθεια υλοποίησης μιας μεθόδου που θα επιτρέψει τη δυνατότητα επέκτασης σε πολύ μεγαλύτερο εύρος θερμοκρασιών. Για να επιτευχθεί ο στόχος θα γίνει χρήση δεδομένων από πολλαπλές προσομοιώσεις. Αν και η multiple histogram ακούγεται σαν μια γενίκευση της single histogram υλοποιείται με διαφορετικό τρόπο και είναι σωστό να θεωρηθεί ως μια διαφορετική μέθοδος.

Αν είχαμε ένα σύνολο δεδομένων από πολλαπλές προσομοιώσεις σε κοντινές θερμοκρασίες έτσι ώστε οι ενέργειες στα άκρα να αλληλεπικαλύπτονται ανά δύο προσομοιώσεις τότε με συνεχόμενη εφαρμογή της single histogram θα μπορούσαμε να εκτιμήσουμε παρατηρήσιμες ποσότητες για ένα μεγάλο εύρος θερμοκρασιών. Στα άκρα του εύρους επέκτασης λοιπόν θα ήταν δυνατές δύο εκτιμήσεις για μια δεδομένη θερμοκρασία, μια από το ανώτερο όριο επέκτασης της πρώτης προσομοίωσης και μια από το κατώτερο όριο επέκτασης της δεύτερης. Αυτές οι δύο εκτιμήσεις δεν θα είχαν καλή ακρίβεια, ωστόσο γεννιέται το ερώτημα: Μήπως θα ήταν δυνατό να τις συνδυάσουμε δίνοντας μεγαλύτερο βάρος σε αυτή με τη μεγαλύτερη ακρίβεια έτσι ώστε να λάβουμε μια καλύτερη εκτίμηση της  $\langle Q \rangle$ . Αυτή είναι η ιδέα πίσω από την multiple histogram όπως προτάθηκε από τους Ferrenberg και Swendsen το 1989.

Για μία προσομοίωση Monte Carlo με Boltzmann κριτήριο σημαντικότητας, η πιθανότητα  $p(E)$  να παραχθεί μια κατάσταση με ενέργεια  $E$  σε κάθε χρονικό βήμα είναι:

$$p(E) = \rho(E) \frac{e^{-\beta E}}{Z} \quad (5.4.1)$$

όπου  $\rho(E)$  είναι η πυκνότητα των καταστάσεων ενέργειας του συστήματος, δηλαδή ο αριθμός των καταστάσεων του συστήματος με ενέργεια  $E$ , όπου το συγκεκριμένο σύστημα έχει διακριτές ενέργειες, και  $Z$  είναι η συνάρτηση επιμερισμού. Οι δύο παραπάνω ποσότητες μας είναι άγνωστες κατά την προσομοίωση. Με την εκτέλεση μιας προσομοίωσης σε θερμοκρασία  $\beta$  και την λήψη  $n$  ανεξάρτητων μετρήσεων της ολικής ενέργειας  $E$  του συστήματος, μπορούμε να κάνουμε μια εκτίμηση της  $p(E)$ :

$$p(E) = \frac{N(E)}{n} \quad (5.4.2)$$

Η πυκνότητα των καταστάσεων ενέργειας με χρήση της (5.4.1) είναι τότε:

$$\rho(E) = \frac{N(E)}{n} \frac{Z}{e^{-\beta E}} \quad (5.4.3)$$

Τώρα υποθέτουμε ότι εκτελούμε πολλαπλές προσομοιώσεις σε διαφορετικές τιμές θερμοκρασίας  $\beta_i$ , παίρνοντας κάθε φορά μια εκτίμηση της πυκνότητας καταστάσεων:

$$\rho_i(E) = \frac{N_i(E)}{n_i} \frac{Z_i}{e^{-\beta_i E}} \quad (5.4.4)$$

Σημειώνεται ότι η  $\rho(E)$  είναι μια συνάρτηση που εξαρτάται από το υπό μελέτη σύστημα, και όχι από τη θερμοκρασία, άρα ουσιαστικά οι παραπάνω αντιστοιχούν σε εκτιμήσεις

της ίδιας ποσότητας. Αφού έχουμε πολλές εκτιμήσεις από πολλαπλές προσομοιώσεις η προσπάθεια επικεντρώνεται τώρα στην εύρεση της καλύτερης εκτίμησης για την πραγματική τιμή της πυκνότητας καταστάσεων της ενέργειας. Για να γίνει πιο εμφανής η σχέση αυτή στο διάγραμμα 5.3 έχουν σχεδιασθεί τα ιστογράμματα ενέργειας  $N_i(E)$  για πέντε προσομοιώσεις σε πέντε διαφορετικές θερμοκρασίες. Οι προσομοιώσεις έχουν δειγματίσει διαφορετικά αλλά αλληλοεπικαλυπτόμενα διαστήματα ενεργειών. Η εξίσωση (5.4.4) θα δώσει μια καλή μέτρηση της πυκνότητας καταστάσεων της ενέργειας σε μια περιοχή όπου το ιστόγραμμα  $N_i(E)$  θα έχει πολλά δείγματα και μια μέτρηση μικρής ακρίβειας σε περιοχές με μικρές τιμές δειγμάτων. Στόχος είναι να εφαρμοστεί ένας μέσος όρος με κατάλληλα βάρη στις εκτιμήσεις  $\rho_i(E)$  για την τελική λήψη μιας καλής εκτίμησης από όλο το εύρος ενεργειών. Θα δίνεται μεγαλύτερο βάρος σε περιοχές όπου τα αντίστοιχα ιστογράμματα έχουν περισσότερα δείγματα.

Αν έχουμε ένα αριθμό μετρήσεων  $x_i$  μιας ποσότητας  $x$ , όπου κάθε όρος έχει ένα τυπικό σφάλμα  $\sigma_i$  τότε η καλύτερη εκτίμηση της  $x$  είναι:

$$\bar{x} = \frac{\sum_i x_i / \sigma_i^2}{\sum_j 1 / \sigma_j^2} \quad (5.4.5)$$

Οι διασπορές  $\sigma_i^2(E)$  των εκτιμήσεων  $\rho_i(E)$  είναι ανάλογες των  $N_i^{-1}(E)$ , άρα για την καλύτερη εκτίμηση της  $\rho(E)$  δίνουμε βάρος βάση του αριθμού δειγμάτων του ιστογράμματος για την δεδομένη ενέργεια.

Υποθέτοντας ότι οι μετρήσεις της ενέργειας είναι στατιστικά ανεξάρτητες, το σφάλμα  $\Delta N_i(E)$  του αριθμού δειγμάτων σε κάθε bin του ιστογράμματος θα είναι Poissonian. Δηλαδή:

$$\Delta N_i(E) = \sqrt{N_i(E)} \quad (5.4.6)$$

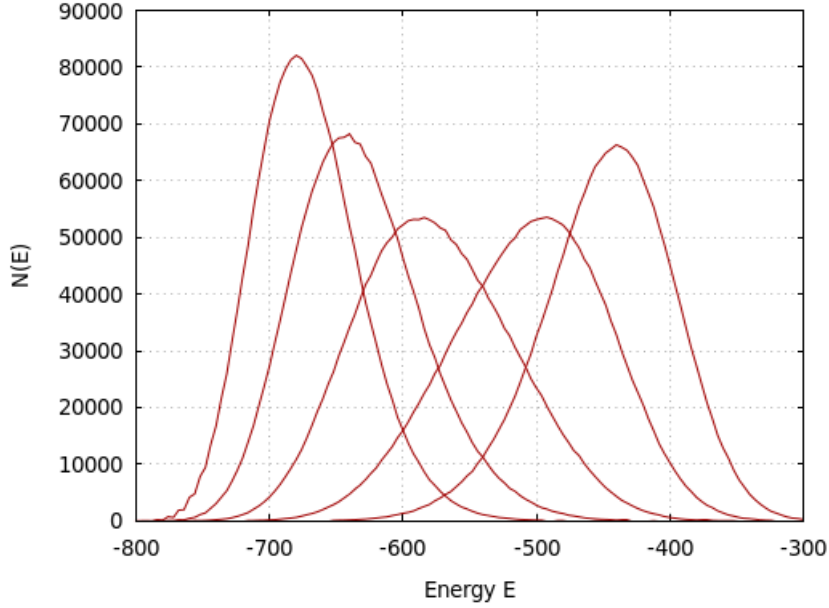
Το πραγματικό σφάλμα  $\Delta N_i(E)$  είναι συνάρτηση του μέσου ιστογράμματος  $\overline{N_i(E)}$  στη θερμοκρασία  $\beta_i$  που έχει ληφθεί από πολλαπλές εκτελέσεις.

$$\Delta N_i(E) = \sqrt{\overline{N_i(E)}} \quad (5.4.7)$$

Δηλαδή υποθέτουμε ένα πολύ μεγάλο αριθμό προσομοιώσεων που έχουν εκτελεσθεί για τη θερμοκρασία  $\beta_i$ , με  $n_i$  μετρήσεις της ενέργειας  $E$  για την κάθε μία, δημιουργούμε ιστογράμματα για κάθε εκτέλεση και υπολογίζουμε τη μέση τιμή μεταξύ των bins. Η τετραγωνική ρίζα αυτής της μέσης τιμής θα δώσει τη σωστή εκτίμηση σφάλματος  $\Delta N_i(E)$  σε κάθε ιστόγραμμα. Ο λόγος που αυτό είναι σημαντικό είναι η εξάρτηση της πυκνότητας καταστάσεων ενέργειας από τα  $\overline{N_i(E)}$ :

$$\rho(E) = \frac{\overline{N_i(E)}}{n_i} \frac{Z_i}{e^{-\beta_i E}} \quad (5.4.8)$$

όπου  $\rho(E)$  είναι η ακριβής τιμή της πυκνότητας καταστάσεων ενέργειας. Με άλλα λόγια για άπειρες εκτελέσεις προσομοιώσεων στη  $\beta_i$  θα μπορούσε να υπολογιστεί η ακριβής τιμή πυκνότητας καταστάσεων ενέργειας από τα αποτελέσματα. Επειδή αυτό δεν είναι δυνατό η χρήση της (5.4.8) απλοποιεί τους υπολογισμούς σε σημαντικό βαθμό.



Σχήμα 5.3: Ιστογράμματα  $N(E)$  των ενεργειών για το δισδιάστατο πρότυπο Ising σε πλέγμα  $N = L * L = 20 * 20 = 400$  για θερμοκρασίες (από δεξιά προς αριστερά)  $\beta = 0.40, 0.42, 0.44, 0.46, 0.48$ . Είναι εμφανής η αλληλοεπικάλυψη των ιστογραμμάτων η οποία αποτελεί απαραίτητη προϋπόθεση για την εξαγωγή σωστών αποτελεσμάτων με τη μέθοδο multiple histogram.

Η διασπορά στα  $N_i(E)$  αποτελεί τη μόνη πηγή σφάλματος στην εξίσωση (5.4.4) άρα το σφάλμα  $\sigma_i$  στη  $\rho_i(E)$  είναι:

$$\sigma_i = \frac{\Delta N_i(E)}{n_i} \frac{Z_i}{e^{-\beta_i E}} = \frac{\sqrt{N_i(E)}}{n_i} \frac{Z_i}{e^{-\beta_i E}} \quad (5.4.9)$$

και η διασπορά:

$$\sigma_i^2 = \frac{N_i(E)}{n_i^2} \left[ \frac{Z_i}{e^{-\beta_i E}} \right]^2 = \frac{\rho^2(E)}{N_i(E)} \quad (5.4.10)$$

με χρήση της εξίσωσης (5.4.8). Ακολουθώντας λογική παρόμοια με την (5.4.5) η καλύτερη εκτίμηση της  $\rho(E)$  δίνεται από τη σχέση:

$$\rho(E) = \frac{\sum_i \overline{N_i(E)} [N_i(E)/n_i] [Z_i/e^{-\beta_i E}]}{\sum_j \overline{(N_j E)}} = \frac{\sum_i N_i(E)}{\sum_j n_j Z_j^{-1} e^{-\beta_j E}} \quad (5.4.11)$$

όπου έγινε χρήση της (5.4.8) αφού δεν γνωρίζουμε τις ποσότητες  $\overline{N_i(E)}$ .

Η εξίσωση (5.4.11) εμπεριέχει την συνάρτηση επιμερισμού  $Z_j$  του συστήματος της κάθε προσομοίωσης σε θερμοκρασία  $\beta_j$  η οποία είναι μια άγνωστη ποσότητα που όπως προαναφέρθηκε είναι αρκετά δύσκολο να υπολογιστεί. Παρατηρούμε όμως ότι η συνάρτηση επιμερισμού δίνεται από τη σχέση:

$$Z_k = \sum_E \rho(E) e^{-\beta_k E} = \sum_E \frac{\sum_i N_i(E)}{\sum_j n_j Z_j^{-1} e^{(\beta_k - \beta_j)E}} \quad (5.4.12)$$

Η υλοποίηση της multiple histogram γίνεται με την επαναλαμβανόμενη επίλυση της παραπάνω εξίσωσης για κάθε τιμή της θερμοκρασίας  $\beta_k$  των προσομοιώσεων. Γίνεται μια αρχική εκτίμηση των τιμών  $Z_k$  οι οποίες επανατοποθετούνται ξανά στην εξίσωση οδηγώντας σε μια καλύτερη εκτίμηση των  $Z_k$ . Αυτή η επανάληψη γίνεται αρκετές φορές μέχρι να καταλήξουμε στις σωστές τιμές των συναρτήσεων επιμερισμού.

Συγκεκριμένα η επιλογή των αρχικών τιμών των  $Z_k$  μπορεί να γίνει αυθαίρετα αρκεί να είναι θετικές. Για την εφαρμογή της μεθόδου επιλέχθηκαν αρχικές τιμές  $Z_k = 1$  οι οποίες με την επαναλαμβανόμενη εκτέλεση θα συγκλίνουν στις πραγματικές. Θέτεται ως προϋπόθεση για τη διακοπή επανάληψης της (5.4.12) η κλασματική μεταβολή ανάμεσα σε δύο τιμές της συνάρτησης επιμερισμού οι οποίες διαφέρουν κατά ένα βήμα επανάληψης να είναι μικρότερη του  $e = 10^{-7}$ . Ουσιαστικά αυτή η επιλογή δεν προσφέρει κάποια ασφάλεια μιας και δεν είναι γνωστή η ταχύτητα με την οποία οι τιμές συγκλίνουν ωστόσο πρακτικά παρατηρείται ότι η σύγκλιση συμβαίνει με εκθετικό ρυθμό και τα αποτελέσματα είναι ακριβή. Επιπλέον η εκθετική εξάρτηση της συνάρτησης επιμερισμού από την ολική ενέργεια  $E$  των καταστάσεων του συστήματος συνεπάγεται ότι οι συναρτήσεις επιμερισμού παίρνουν είτε πολύ μεγάλες είτε πολύ μικρές τιμές, δηλαδή τιμές που δεν μπορούν να αναπαρασταθούν στα όρια ενός υπολογιστή. Στο πλαίσιο της διπλωματικής και εφόσον δεν είναι αναγκαίος ο υπολογισμός πλεγμάτων μεγάλου μεγέθους αυτό το πρόβλημα αντιμετωπίστηκε με τον πολλαπλασιασμό των συναρτήσεων επιμερισμού σε κάθε βήμα εκτέλεσης της (5.4.12) κατά ένα παράγοντα:

$$A = \frac{1}{\sqrt{Z_{max} Z_{min}}} \quad (5.4.13)$$

όπου  $Z_{max}$ ,  $Z_{min}$  η συνάρτηση επιμερισμού με τη μεγαλύτερη και μικρότερη τιμή αντίστοιχα σε κάθε βήμα εκτέλεσης. Σημειώνεται ότι ο παράγοντας  $A$  αναιρείται κατά τον υπολογισμό της εξίσωσης (5.4.15) που δίνεται παρακάτω.

Με δεδομένες τις συναρτήσεις επιμερισμού των προσομοιώσεων που χρησιμοποιήθηκαν είναι πια εύκολο να υπολογιστεί η συνάρτηση επιμερισμού σε οποιαδήποτε άλλη θερμοκρασία με μία γενίκευση της (5.4.12):

$$Z(\beta) = \sum_E \frac{\sum_i N_i(E)}{\sum_j n_j Z_j^{-1} e^{(\beta - \beta_j)E}} \quad (5.4.14)$$

Σε συνδυασμό με την εξίσωση (5.4.11) γίνεται εκτίμηση της εσωτερικής ενέργειας μέσω της σχέσης:

$$U(\beta) = \sum_E E \rho(E) \frac{e^{-\beta E}}{Z(\beta)} = \frac{1}{Z(\beta)} \sum_E E \frac{\sum_i N_i(E)}{\sum_j n_j Z_j^{-1} e^{(\beta - \beta_j)E}} \quad (5.4.15)$$

Όπως και με τη μέθοδο single histogram είναι δυνατή η εκτίμηση των παρατηρήσιμων ποσοτήτων με αναλυτικό υπολογισμό. Συγκεκριμένα για την συνάρτηση επιμερισμού θα ισχύει:

$$Z(\beta) = \sum_E \frac{\sum_i N_i(E)}{\sum_j n_j Z_j^{-1} e^{(\beta-\beta_j)E}} = \sum_{i,E} \frac{N_i(E)}{\sum_j n_j Z_j^{-1} e^{(\beta-\beta_j)E}} = \sum_{i,s} \frac{1}{\sum_j n_j Z_j^{-1} e^{(\beta-\beta_j)E_{is}}} \quad (5.4.16)$$

όπου το άθροισμα  $s$  είναι πια σε όλες τις καταστάσεις που δειγματίστηκαν κατά την  $i$  προσομοίωση και  $E_{is}$  είναι η ολική ενέργεια κάθε κατάστασης. Ακολουθώντας την ίδια λογική η εσωτερική ενέργεια για αναλυτικό υπολογισμό δίνεται μέσω της σχέσης:

$$U(\beta) = \frac{1}{Z(\beta)} \sum_{i,s} \frac{E_{is}}{\sum_j n_j Z_j^{-1} e^{(\beta-\beta_j)E_{is}}} \quad (5.4.17)$$

Παρομοίως προκύπτει η γενική εξίσωση αναλυτικού υπολογισμού για κάθε παρατηρήσιμη ποσότητα  $\langle Q \rangle$ :

$$\langle Q \rangle = \frac{1}{Z(\beta)} \sum_{i,s} \frac{Q_{is}}{\sum_j n_j Z_j^{-1} e^{(\beta-\beta_j)E_{is}}} \quad (5.4.18)$$

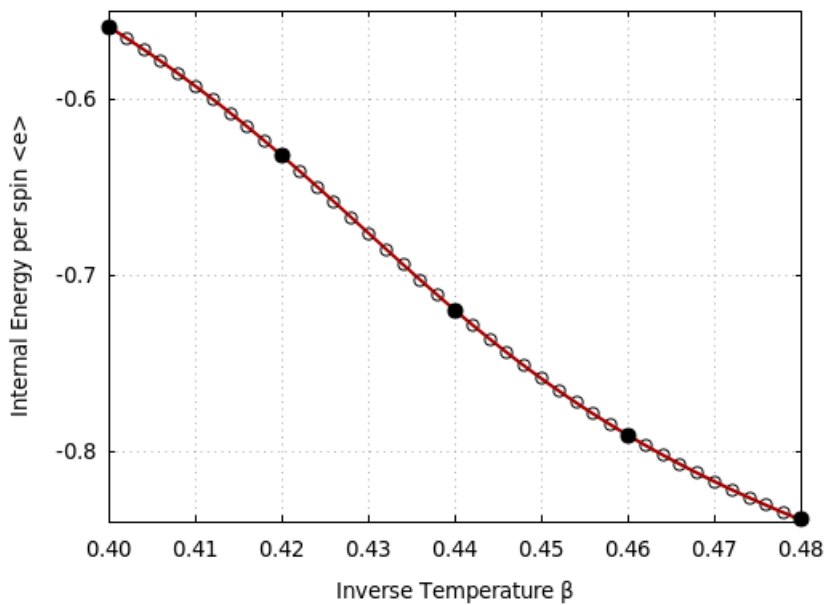
Οι εξισώσεις αναλυτικού υπολογισμού είναι άξιες αναφοράς επειδή έχουν γενική χρήση. Όπως έχει προαναφερθεί στη single histogram η χρήση ιστογραμμάτων μπορεί να μην ενδεύονται σε συγκεκριμένες περιπτώσεις. Για μία παρατηρήσιμη ποσότητα  $\langle Q \rangle$  με χρήση ιστογραμμάτων η γενίκευση της εξίσωσης (5.4.15) είναι:

$$\langle Q \rangle = \frac{1}{Z(\beta)} \sum_E Q \frac{\sum_i N_i(E, Q)}{\sum_j n_j Z_j^{-1} e^{(\beta-\beta_j)E}} \quad (5.4.19)$$

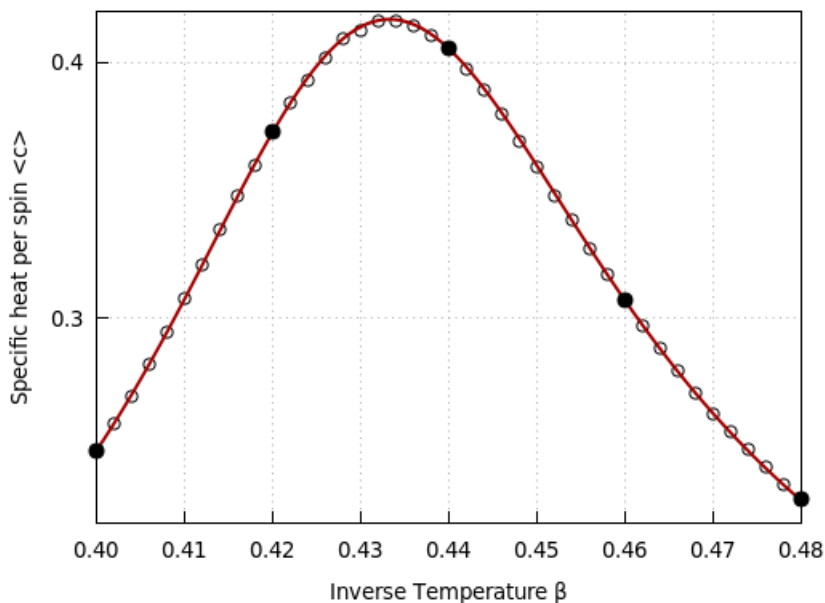
και βάση αυτής ή της αναλυτικής σχέσης υπολογίζεται η ενέργεια, το τετράγωνο της ενέργειας, η μαγνήτιση και το τετράγωνο της μαγνήτισης ώστε να γίνει δυνατός ο υπολογισμός και των ποσοτήτων της ειδικής θερμότητας και μαγνητικής επιδεκτικότητας.

Η εκτίμηση σφαλμάτων γίνεται με τη μέθοδο blocking όπως περιγράφηκε στο υποκεφάλαιο 5.2.2. Τα δεδομένα κάθε προσομοίωσης χωρίζονται σε δέκα μέρη, από τα οποία εκτελείται κάθε φορά η μέθοδος multiple histogram οδηγώντας σε δέκα διαφορετικές εκτιμήσεις μιας παρατηρήσιμης ποσότητας και τον υπολογισμό του σφάλματος της.

Διάγραμμα της εσωτερικής ενέργειας ανά σπιν  $\langle e \rangle$  συναρτήσει της  $\beta$  για  $L = 20$

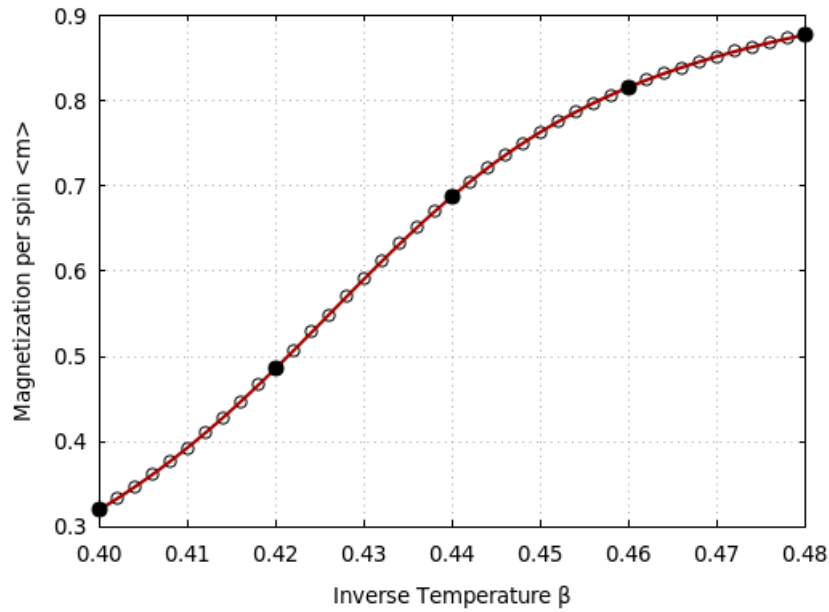


Διάγραμμα της ειδικής θερμότητας  $c$  συναρτήσει της  $\beta$  για  $L = 20$

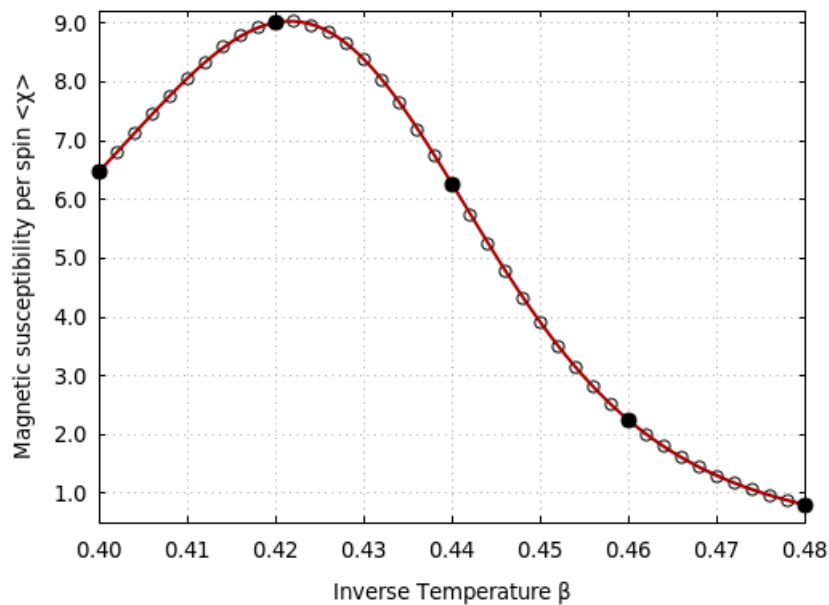


Σχήμα 5.4: Εφαρμογή της μεθόδου multiple histogram (ευθεία σχήματος) για πλέγμα μεγέθους  $N = L * L = 20 * 20 = 400$  και σύγκριση με δεδομένα που προκύπτουν από απευθείας υπολογισμούς με εκτελέσεις προσομοιώσεων Monte Carlo (σχήμα  $\circ$ ). Αναπαρίσταται η ενέργεια  $\langle e \rangle$  στο άνω σχήμα και η ειδική θερμότητα  $c$  στο κάτω σχήμα. Οι συμπαγείς κύκλοι ( $\bullet$ ) δηλώνουν τις προσομοιώσεις που χρησιμοποιήθηκαν για την εφαρμογή της μεθόδου και την παρεμβολή (interpolation) στη μεταβλητή της θερμοκρασίας. Η συγκεκριμένη εκτέλεση αναδεικνύει την δυνατότητα χρήσης της μεθόδου με στρατηγική επιλογή προσομοιώσεων για παρεμβολή σε μεγάλο εύρος. Για το συγκεκριμένο μέγεθος πλέγματος εφόσον υπάρχει μεγάλη αλληλοεπικάλυψη στα ιστογράμματα ενεργειών το δυνατό εύρος επέκτασης είναι κατά πολύ μεγαλύτερο από το εικονιζόμενο.

Διάγραμμα της μαγνήτισης ανά σπιν  $\langle m \rangle$  συναρτήσει της  $\beta$  για  $L = 20$

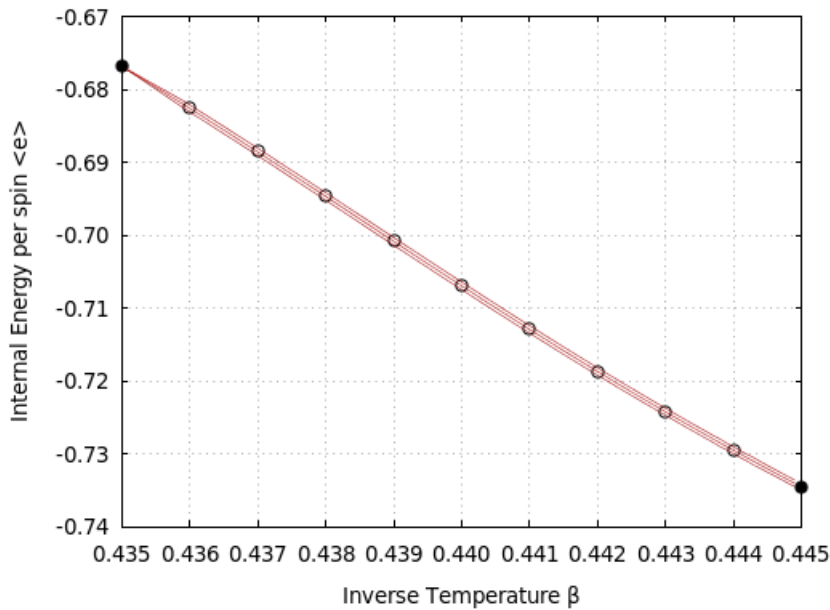


Διάγραμμα της μαγνητικής επιδεκτικότητας  $\chi$  συναρτήσει της  $\beta$  για  $L = 20$

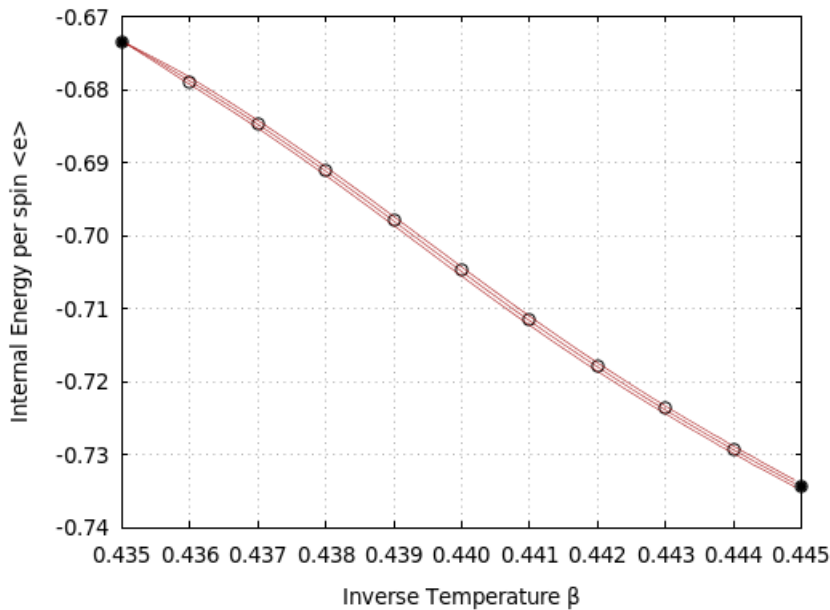


Σχήμα 5.5: Αναπαρίσταται η μαγνήτιση  $\langle m \rangle$  στο πάνω σχήμα και η μαγνητική επιδεκτικότητα  $\chi$  στο κάτω σχήμα για την ίδια περίπτωση πλέγματος του σχήματος 5.4.

Διάγραμμα της εσωτερικής ενέργειας ανά σπιν  $\langle e \rangle$  συναρτήσει  $\beta$  για  $L = 80$



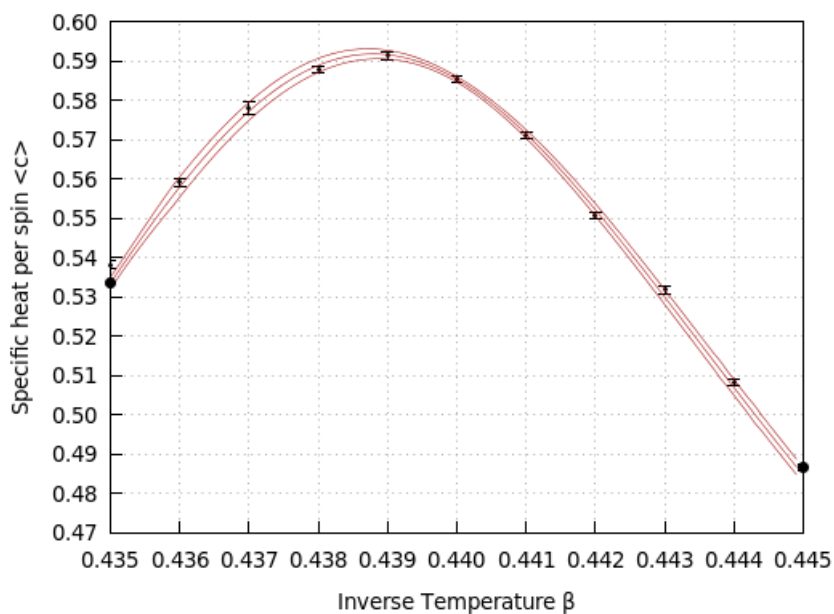
Διάγραμμα της εσωτερικής ενέργειας ανά σπιν  $\langle e \rangle$  συναρτήσει της  $\beta$  για  $L = 140$



Σχήμα 5.6: Εφαρμογή της μεθόδου multiple histogram η οποία αναπαρίσταται ως η ενδιάμεση ευθεία. Επίσης έχουν σχεδιαστεί τα σφάλματα όπως υπολογίστηκαν με τη μέθοδο blocking ως δύο ευθείες για καλύτερη παρουσίαση. Το πλέγμα στο πάνω σχήμα είναι μεγέθους  $N = L * L = 80 * 80 = 6400$  και στο κάτω σχήμα  $N = L * L = 140 * 140 = 19600$ . Τα δεδομένα που προκύπτουν από απευθείας υπολογισμούς σε κάθε περίπτωση με εκτελέσεις προσομοιώσεων Monte Carlo αναπαρίστανται στο σχήμα ως κύκλοι (σχήμα ο). Τα σφάλματα από τη μέθοδο jackknife για τα προσομοιωμένα δεδομένα είναι πολύ μικρά και δεν εμφανίζονται στο σχήμα. Οι συμπαγείς κύκλοι(●) δηλώνουν τις προσομοιώσεις που χρησιμοποιήθηκαν για την εφαρμογή της μεθόδου και την παρεμβολή(interpolation) στη μεταβλητή της θερμοκρασίας. Η συγκεκριμένη εκτέλεση αναδεικνύει τον τρόπο μελέτης πλεγμάτων μεγαλύτερου μεγέθους όπου το εύρος παρεμβολής είναι μικρότερο. Παρατηρούμε ότι οι τιμές βρίσκονται εντός ορίων σφαλμάτων της multiple histogram.

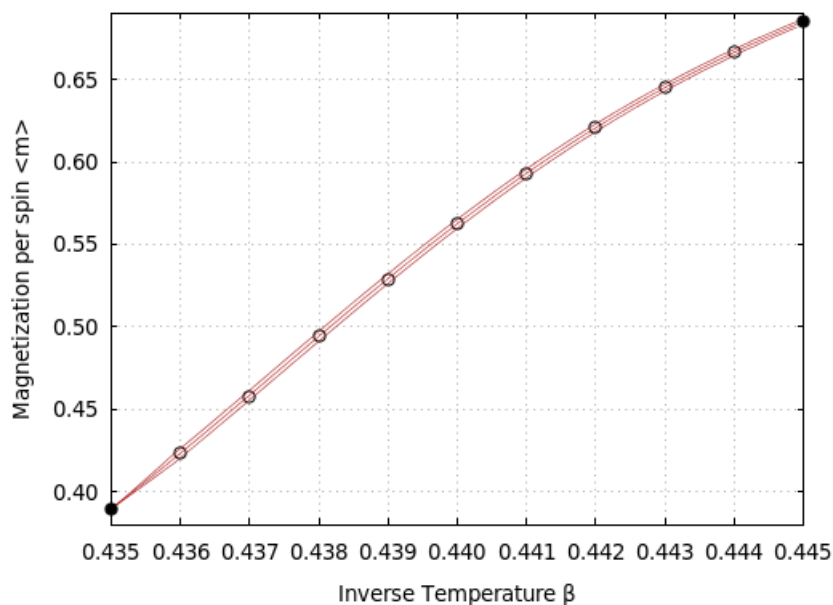


Διάγραμμα της ειδικής θερμότητας  $c$  συναρτήσει της θερμοκρασίας  $\beta$  για  $L = 80$

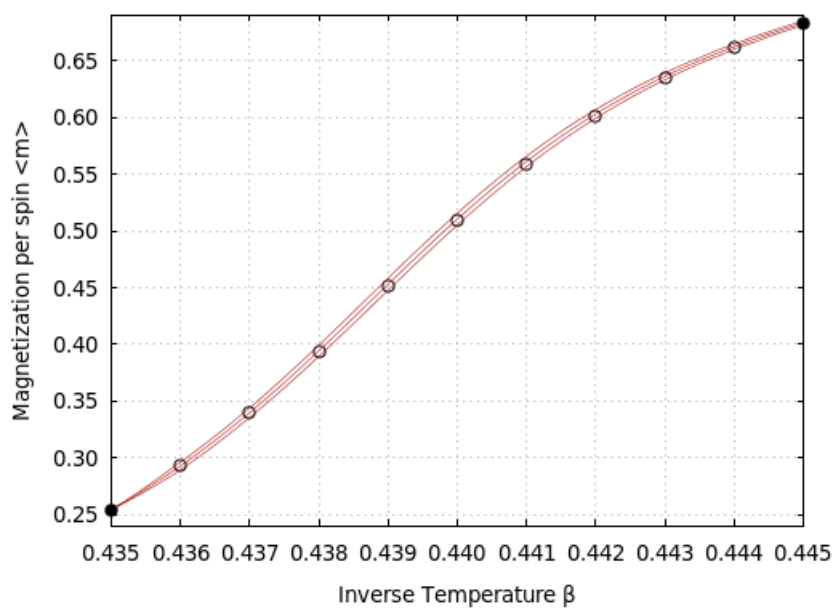


Σχήμα 5.7: Αναπαρίσταται η ειδική θερμότητα  $c$  για μέγεθος πλέγματος  $N = L * L = 80 * 80 = 6400$  για την περίπτωση  $L = 80$  του σχήματος 5.6. Για τα σφάλματα των δεδομένων της multiple histogram έχει χρησιμοποιηθεί η μέθοδος blocking ενώ για αυτά των προσομοιωμένων δεδομένων η μέθοδος jackknife. Παρατηρούμε ότι οι τιμές βρίσκονται εντός ορίων των σφαλμάτων.

Διάγραμμα της μαγνήτισης ανά σπιν  $\langle m \rangle$  συναρτήσει της θερμοκρασίας  $\beta$  για  $L = 80$

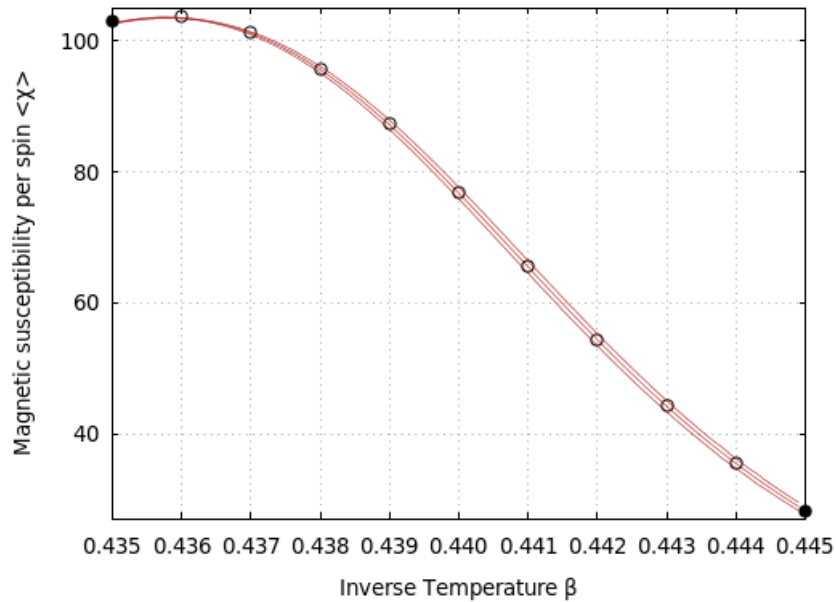


Διάγραμμα της μαγνήτισης ανά σπιν  $\langle m \rangle$  συναρτήσει της θερμοκρασίας  $\beta$  για  $L = 140$

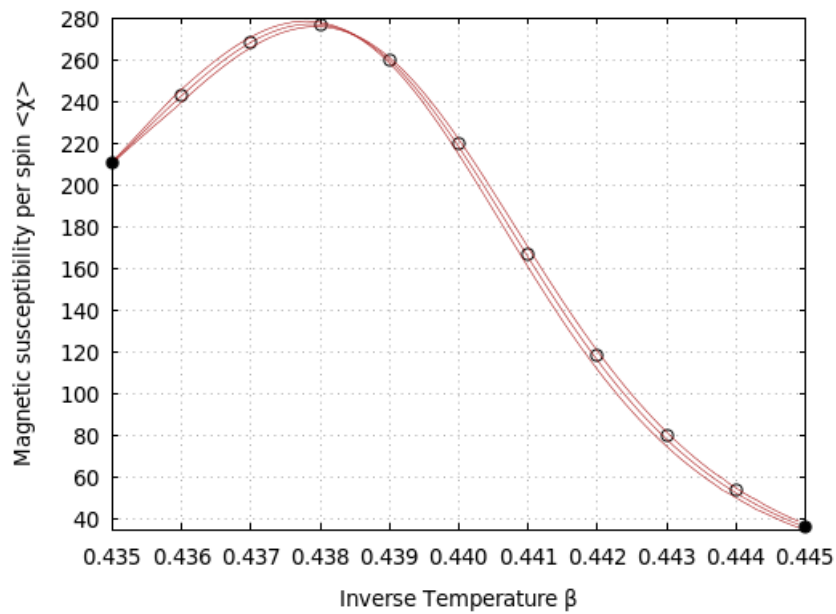


Σχήμα 5.8: Αναπαρίσταται η μαγνήτιση  $\langle m \rangle$  για μέγεθος πλέγματος  $N = L * L = 80 * 80 = 6400$  και  $N = L * L = 140 * 140 = 19600$  στο πάνω και κάτω διάγραμμα αντίστοιχα για τις ίδιες περιπτώσεις του σχήματος 5.6.

Διάγραμμα της μαγνητικής επιδεκτικότητας  $\chi$  συναρτήσει της θερμοκρασίας  $\beta$  για  $L = 80$



Διάγραμμα της μαγνητικής επιδεκτικότητας  $\chi$  συναρτήσει της θερμοκρασίας  $\beta$  για  $L = 140$



Σχήμα 5.9: Αναπαρίσταται η μαγνητική επιδεκτικότητα  $\chi$  για μέγεθος πλέγματος  $N = L * L = 80 * 80 = 6400$  και  $N = L * L = 140 * 140 = 19600$  στο πάνω και κάτω διάγραμμα αντίστοιχα για τις ίδιες περιπτώσεις του σχήματος 5.6.

## 5.5 Κρίσιμα Φαινόμενα, Παγκοσμιότητα και Βάθμιση

Όπως έχει προαναφερθεί τα κρίσιμα φαινόμενα εμφανίζονται σε περιοχές ενός συστήματος κοντά στη συνεχή μετάβαση φάσης. Ορίζεται η ανηγμένη θερμοκρασία ως:

$$t = \frac{\beta_c - \beta}{\beta_c} \quad (5.5.1)$$

Όταν  $t \rightarrow 0$  ή ισοδύναμα  $\beta \rightarrow \beta_c$  το μήκος συσχετισμού  $\xi = \xi(\beta, L = \infty)$  στο θερμοδυναμικό όριο απειρίζεται σύμφωνα με τη σχέση:

$$\xi \sim |t|^{-\nu} \quad (5.5.2)$$

και το φυσικό σύστημα βρίσκεται σε μία κλάση παγκοσμιότητας η οποία χαρακτηρίζεται από κρίσιμους εκθέτες. Συστήματα με διαφορετικές μικροσκοπικές χαμιλτονιανές τα οποία όμως βρίσκονται στην ίδια κλάση έχουν ίδιους κρίσιμους εκθέτες. Άρα το πρότυπο Ising παραμένει στην ίδια κλάση παγκοσμιότητας είτε με την αλλαγή της τοπολογίας (τετραγωνικό πλέγμα σε τριγωνικό, εξαγωνικό κτλ), είτε με την προσθήκη όρων στην αλληλεπίδραση πλησιέστερων γειτόνων από απομακρυσμένες πλεγματικές θέσεις οι οποίες όμως βρίσκονται σε απόσταση μικρότερη του μήκους συσχετισμού. Επίσης η μετάβαση φάσης διαφορετικών συστημάτων μπορεί να βρίσκεται στην ίδια κλάση όπως συμβαίνει για το πρότυπο Ising με υγρό/ατμό στο τρικρίσιμο σημείο. Ουσιαστικά μέσω του φαινομένου της παγκοσμιότητας γίνεται δυνατή η μελέτη περίπλοκων συστημάτων μέσω των απλούστερων δυνατών μοντέλων.

Οι κρίσιμοι εκθέτες του προτύπου Ising για πλέγμα το οποίο απειρίζεται  $L \rightarrow \infty$  είναι:

$$\chi \sim |t|^{-\gamma}, \gamma = 7/4 \quad (5.5.3)$$

$$c \sim |t|^{-\alpha}, \alpha = 0 \quad (5.5.4)$$

$$m \sim |t|^{-\beta}, t < 0, \beta = 1/8 \quad (5.5.5)$$

Όταν το πλέγμα είναι πεπερασμένου μεγέθους για μήκος συσχετισμού  $1 \ll \xi \ll L$  η συμπεριφορά του μοντέλου είναι προσεγγιστικά ίδια όπως του απείρου συστήματος και για  $\beta \approx \beta_c$ ,  $\xi \sim L$  εμφανίζονται φαινόμενα επίδρασης πεπερασμένου μεγέθους. Για το πλέγμα πεπερασμένου μεγέθους οι διακυμάνσεις  $\chi$ ,  $c$  παρουσιάζουν μέγιστο για μια ψευδοκρίσιμη θερμοκρασία  $\beta_c(L)$  για την οποία  $\lim_{L \rightarrow \infty} \beta_c(L) = \beta_c$ .

Χρησιμοποιώντας τη σχέση (5.5.2) για  $\beta = \beta_c(L)$  έχουμε  $\xi(t, L) \sim L$  και άρα  $|t| \sim L^{-1/\nu}$ , άρα οι σχέσεις (5.5.3) - (5.5.5) μετατρέπονται στις ακόλουθες:

$$\chi \sim L^{\gamma/\nu} \quad (5.5.6)$$

$$c \sim L^{\alpha/\nu} \quad (5.5.7)$$

$$m \sim L^{-\beta/\nu} \quad (5.5.8)$$

Στις περισσότερες ενδιαφέρουσες περιπτώσεις υπολογισμού κρίσιμων εκθετών, η κρίσιμη παράμετρος δεν είναι γνωστή. Συγκεκριμένα για το δισδιάστατο πρότυπο Ising η γνώση

της κρίσιμης θερμοκρασίας από τη λύση του Onsager επιτρέπει ικανοποιητικούς υπολογισμούς μέσω των σχέσεων (5.5.6) - (5.5.8) για τις τιμές της μαγνητικής επιδεικτικότητας, ειδικής θερμότητας και μαγνήτισης στη κρίσιμη θερμοκρασία. Η προσαρμογή σε δεδομένα από πλέγματα διαφορετικού μεγέθους δίνει τους κρίσιμους εκθέτες.

Στο υπόλοιπο του υποκεφαλαίου θα γίνει υπολογισμός των κρίσιμων εκθετών μέσω της ψευδοκρίσιμης περιοχής χωρίς να απαιτείται η γνώση της κρίσιμης θερμοκρασίας και μάλιστα η προσαρμογή στα δεδομένα θα προσφέρει έναν υπολογισμό για την  $\beta_c$ . Είναι σημαντική η αναφορά στην αιτία της βάθμισης των φυσικών ποσοτήτων. Η ψευδοκρίσιμη συμπεριφορά ενός πεπερασμένου συστήματος που εμφανίζει συνεχή μετάβαση τάξης στο θερμοδυναμικό όριο οφείλεται στην αύξηση του μήκους συσχετισμού  $\xi$  σε μέγεθος που συγκρίνεται με το μέγεθος του συστήματος  $L$  με την παρουσίαση των φαινομένων επίδρασης πεπερασμένου μεγέθους. Στη περιοχή αυτή, η οποία γίνεται όλο και πιο στενή με την αύξηση του μεγέθους του συστήματος, οι ποσότητες που απειρίζονται στο θερμοδυναμικό όριο ή παρουσιάζουν μη αναλυτική συμπεριφορά έχουν πεπερασμένη, μέγιστη τιμή. Η επιλογή μιας τιμής στη περιοχή με συστηματικό τρόπο για αρκετά μεγάλα  $L$  δίνει τις ασυμπτωτικές συμπεριφορές των σχέσεων (5.5.6) - (5.5.8). Η επιλογή αυτής της τιμής για την εφαρμογή της μεθόδου είναι ο υπολογισμός των ποσοτήτων στην ψευδοκρίσιμη θερμοκρασία  $\beta_c(L)$ , δηλαδή η τιμή που παρουσιάζεται το μέγιστο των διακυμάνσεων της παραμέτρου τάξης:

$$\chi(\beta_c(L), L) \equiv \chi_{max}(L) \quad (5.5.9)$$

Επίσης είναι δυνατό να οριστεί μια διαφορετική  $\beta'_c(L)$ , για παράδειγμα μέσω του μεγίστου της ειδικής θερμότητας  $c$ :

$$c(\beta'_c(L), L) \equiv c_{max}(L) \quad (5.5.10)$$

Οι δύο επιλογές για πλέγματα μεγάλου μεγέθους θα δώσουν τις ίδιες τιμές αφού ισχύει:

$$\lim_{L \rightarrow \infty} \beta_c(L) = \lim_{L \rightarrow \infty} \beta'_c(L) = \beta_c \quad (5.5.11)$$

Ωστόσο αυτό δεν συμβαίνει με την ίδια ταχύτητα για τις δύο επιλογές και τελικώς αναδεικνύεται η μία ως καλύτερη.

Στην ψευδοκρίσιμη περιοχή ισχύει  $\xi \approx L$  οπότε σύμφωνα με τη σχέση (5.5.2):

$$|t| = \left| \frac{\beta_c - \beta_c(L)}{\beta_c} \right| \sim \xi^{-\frac{1}{\nu}} \sim L^{-\frac{1}{\nu}} \implies \beta_c(L) = \beta_c - \frac{c}{L^{\frac{1}{\nu}}} \quad (5.5.12)$$

Ο υπολογισμός της ψευδοκρίσιμης θερμοκρασίας για την οποία εμφανίζεται μέγιστο στις διακυμάνσεις της παραμέτρου τάξης γίνεται ευκολότερος και ακριβέστερος με την εφαρμογή της μεθόδου multiple histogram. Εφόσον είναι δυνατή η παρεμβολή για οποιαδήποτε τιμή της θερμοκρασίας ανάμεσα σε δύο προσομοιώσεις για τον υπολογισμό παρατηρήσιμων ποσοτήτων και επειδή η μέθοδος προσφέρει ικανοποιητική ταχύτητα μπορεί να επιλεγεί ένα βήμα  $10^{-4}$  ή και αρκετά μικρότερο (ανάλογα με το μέγεθος πλέγματος και πάντα βάση των σφαλμάτων) έτσι ώστε να υπολογιστεί με τη μεγαλύτερη ακρίβεια η ψευδοκρίσιμη θερμοκρασία.

Ο υπολογισμός μέσω της σχέσης (5.5.12) για την ψευδοκρίσιμη θερμοκρασία των μεγίστων της μαγνητικής επιδεκτικότητας φαίνεται στο σχήμα (5.10). Τα αποτελέσματα είναι:

$$\begin{aligned}\beta_c &= 0.44074 \pm 0.00010 \\ \frac{1}{\nu} &= 0.983 \pm 0.019\end{aligned}\tag{5.5.13}$$

Για την ψευδοκρίσιμη θερμοκρασία των μεγίστων της ειδικής θερμότητας του σχήματος (5.11) τα αποτελέσματα είναι:

$$\begin{aligned}\beta_c &= 0.4406 \pm 0.0002 \\ \frac{1}{\nu} &= 1.00 \pm 0.12\end{aligned}\tag{5.5.14}$$

Ακολουθεί ο υπολογισμός του εκθέτη  $\gamma/\nu$  από τα μέγιστα της μαγνητικής επιδεκτικότητας συναρτήσει του μεγέθους πλέγματος όπου γίνεται μία προσαρμογή όπως φαίνεται στο σχήμα (5.12) όπου δίνει το αποτέλεσμα:

$$\frac{\gamma}{\nu} = 1.751 \pm 0.003\tag{5.5.15}$$

που βρίσκεται μέσα στα όρια σφάλματος με την αναμενόμενη τιμή 7/4.

Τέλος γίνεται ο υπολογισμός του εκθέτη  $\alpha/\nu$ . Επειδή ο εκθέτης  $\alpha = 0$  η μορφή της ασυμπτωτικής σχέσης αναμένεται να δίνεται από την :

$$c(\beta_c, L) \sim \log L\tag{5.5.16}$$

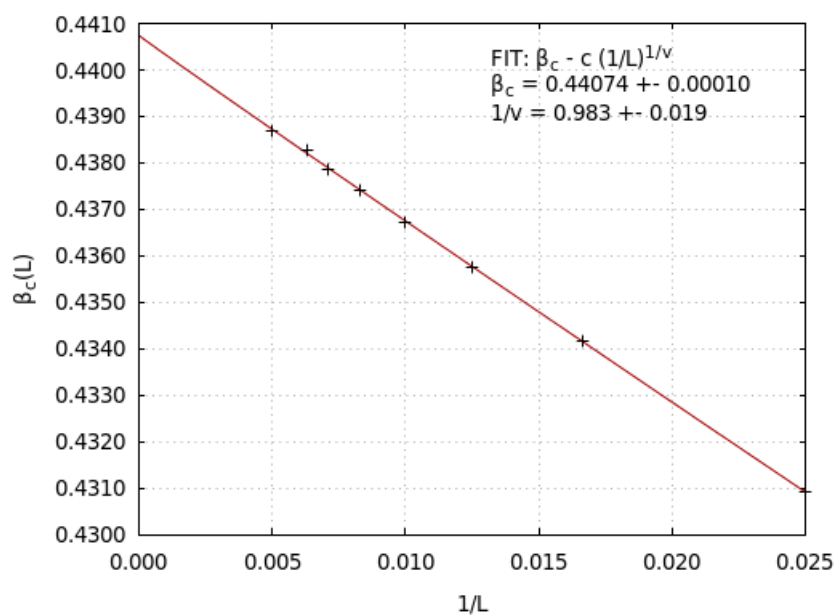
Η προσαρμογή γίνεται στο σχήμα (5.13). Επειδή τα δεδομένα δεν προσαρμόζονται σε συνάρτηση της μορφής  $\log L$  υποθέτουμε ότι υπάρχουν όροι οι οποίοι συνεισφέρουν για αμελητέα  $L$  αλλά μηδενίζονται στο θερμοδυναμικό όριο. Η προσαρμογή στη συνάρτηση  $a \log L + b - c/L$  δίνει σωστά αποτελέσματα ενώ η προσαρμογή σε συνάρτηση της μορφής  $aL^d + b - c/L$  δεν είναι επιτυχής και ο εκθέτης είναι συμβατός με την τιμή μηδέν.

$L$	$\beta_c(L)$	$\chi_{max}$
40	$0.4309 \pm 0.0002$	$30.65 \pm 0.02$
60	$0.4342 \pm 0.0002$	$62.58 \pm 0.06$
80	$0.4358 \pm 0.0002$	$103.76 \pm 0.10$
100	$0.4368 \pm 0.0002$	$153.27 \pm 0.11$
120	$0.43743 \pm 0.00002$	$211.67 \pm 0.35$
140	$0.43787 \pm 0.00001$	$277.40 \pm 0.51$
160	$0.43826 \pm 0.00014$	$351.30 \pm 0.59$
200	$0.43870 \pm 0.00009$	$516.89 \pm 0.55$

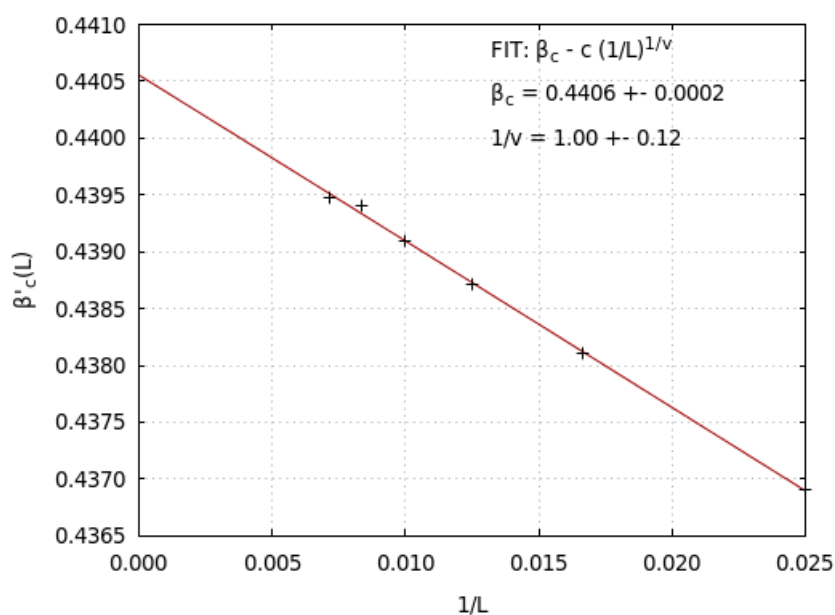
Πίνακας 5.1: Υπολογισμός των ψευδοκρίσιμων θερμοκρασιών  $\beta_c(L)$  από τα μέγιστα της μαγνητικής επιδεκτικότητας  $\chi_{max}$  για διάφορα μεγέθη πλέγματος. Χρησιμοποιήθηκε η μέθοδος Multiple Histogram με χρήση δύο προσομοιώσεων για τον εντοπισμό της περιοχής που εμφανίζονται τα μέγιστα. Μετά εκτελέστηκε με βήμα  $10^{-5}$  για την αναζήτηση της ψευδοκρίσιμης θερμοκρασίας. Δίνονται επίσης και αντίστοιχες τιμές των μεγίστων. Σημειώνεται ότι η μέθοδος Blocking εφαρμόζεται σε αυτή την περίπτωση για τον υπολογισμό των σφαλμάτων των ποσοτήτων.

$L$	$\beta'_c(L)$	$c_{max}$
40	$0.4369 \pm 0.0001$	$0.5035 \pm 0.0005$
60	$0.4381 \pm 0.0002$	$0.5547 \pm 0.0012$
80	$0.43873 \pm 0.00004$	$0.5892 \pm 0.0015$
100	$0.43909 \pm 0.00006$	$0.6183 \pm 0.0013$
120	$0.43937 \pm 0.00002$	$0.6495 \pm 0.0011$
140	$0.4395 \pm 0.0002$	$0.6680 \pm 0.0022$

Πίνακας 5.2: Πίνακας ανάλογος του 5.1 για τις ψευδοκρίσιμες θερμοκρασίες  $\beta'_c(L)$  από τα μέγιστα της ειδικής θερμότητας  $c_{max}$  για διάφορα μεγέθη πλέγματος.

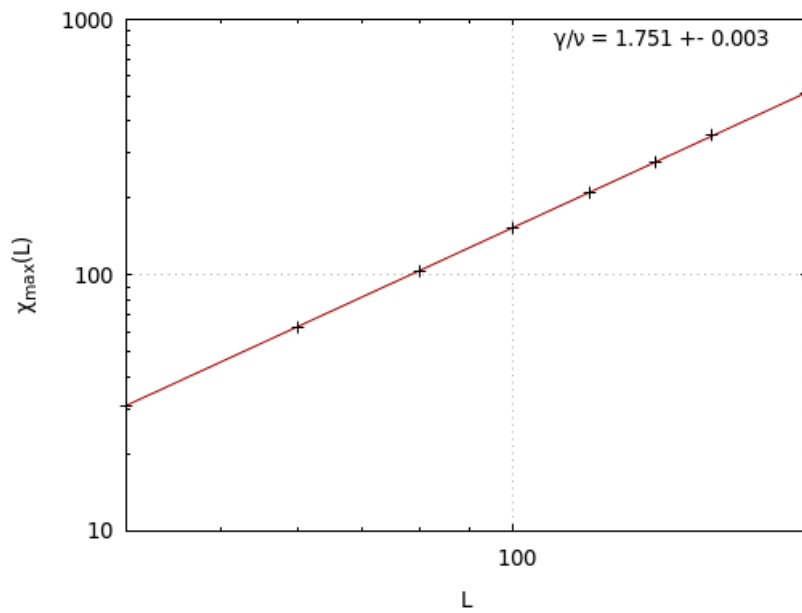


Σχήμα 5.10: Προσδιορισμός της κρίσιμης θερμοκρασίας  $\beta_c$  και του κρίσιμου εκθέτη  $\nu$  από τη σχέση (5.5.12). Η προσαρμογή έχει γίνει στη συνάρτηση  $a - c(1/L)^b$  όπου  $b = 1/\nu$  και  $a = \beta_c$ . Η τομή της ευθείας με τον άξονα  $y$  δίνει την ακριβή τιμή της κρίσιμης θερμοκρασίας  $\beta_c = \log(1 + \sqrt{2})/2 = 0.4407$ .

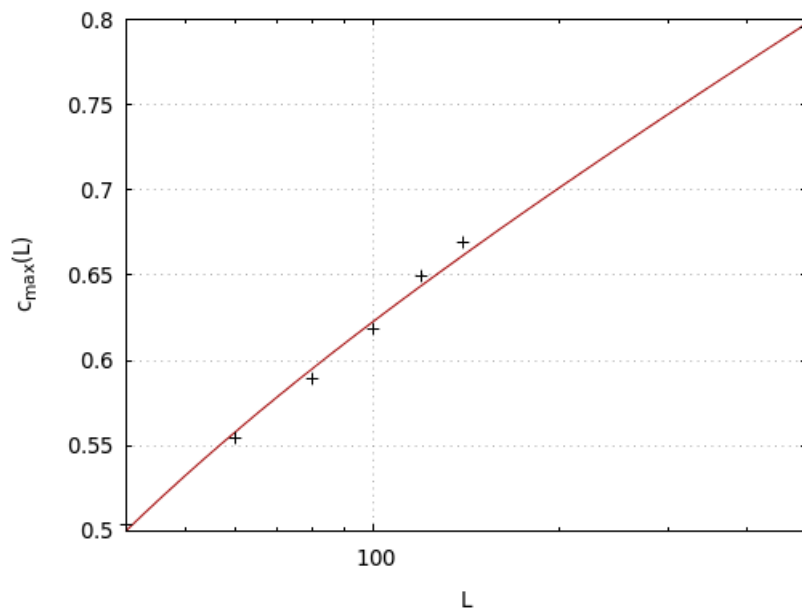


Σχήμα 5.11: Προσδιορισμός της κρίσιμης θερμοκρασίας  $\beta'_c$  και του κρίσιμου εκθέτη  $\nu$  από τη σχέση (5.5.12). Η προσαρμογή έχει γίνει στη συνάρτηση  $a - c(1/L)^b$  όπου  $b = 1/\nu$  και  $a = \beta'_c$ . Η τομή της ευθείας με τον άξονα  $y$  δίνει την ακριβή τιμή της κρίσιμης θερμοκρασίας  $\beta_c = \log(1 + \sqrt{2})/2 = 0.4407$ .





Σχήμα 5.12: Υπολογισμός του κρίσιμου εκθέτη  $\gamma/\nu$  από τα μέγιστα της μαγνητικής επιδεκτικότητας  $\chi_{max}$ . Γίνεται προσαρμογή σε μια συνάρτηση  $\chi_{max} = aL^b$ . Το αποτέλεσμα της προσαρμογής δίνει  $\gamma/\nu = 1.751 \pm 0.003$



Σχήμα 5.13: Υπολογισμός του κρίσιμου εκθέτη  $\alpha/\nu$  από τα μέγιστα της ειδικής θερμότητας. Γίνεται προσαρμογή σε μια συνάρτηση  $c_{max} = a \log L + b - c/L$ . Το αποτέλεσμα της προσαρμογής δίνει  $a = 0.10 \pm 0.01$ ,  $b = 0.19 \pm 0.05$  και  $c = 2.2 \pm 0.9$ . Για προσαρμογή σε συνάρτηση  $c_{max} = aL^d + b - c/L$  παίρνουμε  $d = 0.046 \pm 0.65$  δηλαδή εκθέτη συμβατό με την τιμή μηδέν. Τα δεδομένα είναι συμβατά με τον εκθέτη  $\alpha/\nu$  να είναι μηδέν.



# Βιβλιογραφία

- [1] M. E. J. Newman , G. T. Barkema, *Monte Carlo Methods in Statistical Physics*, Clarendon Press, Oxford(2002).
- [2] K. N. Anagnostopoulos, *Computational Physics: A Practical Introduction to Computational Physics and Scientific Computing*, Published by K. N. Anagnostopoulos, 2014.
- [3] B. A. Berg, *Markov Chain Monte Carlo Simulations and Their Statistical Analysis, With Web-Based Fortran Code*, World Scientific, 2004.
- [4] K. Binder, D. W. Heermann, *Monte Carlo Simulation in Statistical Physics*, 5th Edition, Springer, 2010.
- [5] D. P. Landau, K. Binder *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, 2009.
- [6] B. W. Kernighan, D. M. Ritchie. *The C Programming Language*, Prentice Hall, 1988.
- [7] H. Gould, J. Tobochnik, H. Christian. *Computer Simulation Methods, Application to Physical Systems*, Addison Wesley, 2007.



# Παράρτημα: Κώδικας

Παρατίθεται ο κώδικας υλοποίησης της μεθόδου σε C με τον απαραίτητο σχολιασμό. Επίσης συμπεριλαμβάνονται τα αρχεία κατασκευής διαγραμμάτων μέσω του λογισμικού Gnuplot τα οποία μπορούν να κληθούν κατά την εκτέλεση της μεθόδου. Για περισσότερες πληροφορίες και τη δυνατότητα λήψης του κώδικα με παράδειγμα εφαρμογής σε ένα μικρό πλέγμα, ο αναγνώστης μπορεί να επισκεφθεί το προφίλ [github.com/dbachtis](https://github.com/dbachtis) .

Για κώδικα παραγωγής δεδομένων με χρήση των αλγορίθμων Metropolis και Wolff για μετρήσεις ενέργειας και μαγνήτισης σε μέγεθος πλέγματος  $N = L * L$  ο αναγνώστης μπορεί να επισκεφθεί την ιστοσελίδα του μαθήματος Υπολογιστικής Φυσικής II της σχολής ΕΜΦΕ <http://www.physics.ntua.gr/~konstant/ph47/> την οποία διαχειρίζεται ο επιβλέπων καθηγητής Κωνσταντίνος Αναγνωστόπουλος ή να ανατρέξει στη βιβλιογραφία.

---

```
/* The multiple histogram method used for the estimation of partition functions
 * and observable quantities by interpolating in the variable of temperature.
 * The method uses multiple simulations on the same lattice length of the 2D
 * Ising Model that have been executed adequately close.
 *
 * Necessary flags for the method to run:
 * -L int: Lattice length(N=LxL)
 * -f *char: The input file that consists of the variables of the raw data
 *           in the format:
 *
 *           "datafilepath1 beta1 sweeps1
 *            datafilepath2 beta2 sweeps2
 *            ... "
 *
 *           Notice that the method allows for any number of raw data files.
 *           It also assumes they have measurements of energy and magnetization
 *           in the format:
 *
 *           "Energy1 magnetization1
 *            Energy2 magnetization2
 *            ... "
 *
 *           The raw data files must be evenly spaced for the bootstrap method.
 *
 * Without the extra flags the method will calculate the internal energy and
 * specific heat for a range of temperatures between the minimum and maximum
 * of the simulations set within the input file.
```

```

*
* Extra flags:
* -m: Calculation of magnetization and magnetic susceptibility.
* -e: Calculation of errors with the Blocking method.
* -b int: Calculation of errors with the Bootstrap method where int is the
* number of resampling.
* -p: Plotting of the calculated quantities by using some already made
* Gnuplot scripts.
* -s double: The step on the temperatures of the interpolation. Default is
* 0.001.
*
*
* Folder Structure:
*
* The method will create a subfolder in the Output folder in the format
* LXX.YYMMDDHHFFSS where XX stands for the L value, YY for the year,
* MM for the month, DD for the day, HH for the hour, FF for the minutes
* and SS for the seconds.
* It will then create subfolders for the Results, Plots and the histogram files,
* as well as for the error estimations.
* You are advised, after a successful run, to move the raw data files inside
* the Output folder for future reference.
*
* A typical run using all flags would be(you are advised to avoid using the
* bootstrap method(-b flag)):
*
* ./mh -L 20 -f input.dat -m -e -p -b 1000 -s 0.0001
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <sys/types.h>
#include <sys/stat.h>

struct user_input{
    int L;
    int calculateErrors;
    int calculateMagn;
    int makePlots;
    int bootstrap;
    double step;
    char *inputFile;
};

struct simulations_parameters{
    int numberOfSimulations;
    char **dataFiles;
    char **energyHistogramFiles;
    char **energyMagnHistogramFiles;
    double *betas;
    int *sweeps;
};

};

struct user_input* get_user_input(int argc,char **argv);
struct simulations_parameters* set_input_parameters(char *inputFile,int L,struct tm time_struct);
void make_simulation_en_hist_files(int numberOfSimulations,char **dataFiles, char **histogramFiles);

```

```

void make_simulation_enmagn_hist_files(char **dataFiles, char **energyMagnHistogramFiles, int
    numberOfSimulations);
void make_total_en_hist_file(int numberOfSimulations, char **histogramFiles, char energy_hist_file[])
    ;
void make_total_enmagn_hist_file(int numberOfSimulations, char **histogramFiles, char
    energy_hist_file[]);
//long double* estimate_partition_functions(int numberOfSimulations, double *betas, int *sweeps, char
    **histogramFiles);
long double* estimate_partition_functions(int numberOfSimulations, double *betas, int *sweeps, char
    energy_hist_file[]);
void estimate_internal_energy(double min_beta, double max_beta, int numberOfSimulations, int L, double
    *betas, int *sweeps, char **histogramFiles, long double *partitionFunctions, char energy_hist_file
    [], char outputFile[], double step);
//void estimate_magnetization(double min_beta, double max_beta, int numberOfSimulations, int L, double
    *betas, int *sweeps, char **histogramFiles, long double *partitionFunctions, char energy_hist_file
    [], char outputFile[], double step);
void estimate_magnetization(double min_beta, double max_beta, int numberOfSimulations, int L, double *
    betas, int *sweeps, char **dataFiles, long double *partitionFunctions, char energy_hist_file[], char
    outputFile[], double step);
void make_blocking_data_files(char **dataFiles, int numberOfSimulations, int *sweeps, int L, double
    *betas, struct tm time_struct);
void blocking_method(char **dataFiles, int numberOfSimulations, int *sweeps, int L, double *betas,
    double min_beta, double max_beta, int calculateMagn, struct tm time_struct, double step);
void blocking_results(double *betas, int numberOfSimulations, int L, char errorFile[], char interFile
    []);
int cmpfunc(const void * a, const void * b);
void make_bootstrap_data(double min_beta, double max_beta, char **dataFiles, int *sweeps, double *betas,
    int numberOfSimulations, int L, int bootstrap, int calculateMagn, struct tm time_struct, double
    step);
void calculate_bootstrap_results(double *betas, int numberOfSimulations, int L, char errorFile[], char
    interFile[], int bootstrap);
void make_plots(int L, int calculateMagn, struct tm time_struct);
void create_folder_structure(int L, int makePlots, int calculateErrors, int bootstrap, struct tm
    time_struct);
int main(int argc, char **argv)
{
    int i, L, calculateErrors, calculateMagn, makePlots, bootstrap, numberOfSimulations;
    int *sweeps;
    double min_beta, max_beta, step;
    double *betas;
    char temporaryFileName[100], temporaryFileName2[100];
    char *inputFile;
    char **dataFiles, **energyHistogramFiles, **energyMagnHistogramFiles;
    long double *partitionFunctions;
    struct user_input *arguments;
    struct simulations_parameters *simulations_parameters;

    //Get user input and set the values
    arguments= get_user_input(argc, argv);
    L= arguments->L;
    inputFile= arguments->inputFile;
    calculateErrors=arguments->calculateErrors;
    calculateMagn=arguments->calculateMagn;
    makePlots=arguments->makePlots;
    bootstrap= arguments->bootstrap;
    step= arguments->step;
    free(arguments);
    //get current time and make folder for output data

```

```

time_t t= time(NULL);
struct tm time_struct = *localtime(&t);
create_folder_structure( L,makePlots,calculateErrors,bootstrap,time_struct);
//prepare values
simulations_parameters = set_input_parameters(inputFile,L,time_struct);
numberOfSimulations= simulations_parameters -> numberOfSimulations;
dataFiles = simulations_parameters ->dataFiles;
energyHistogramFiles = simulations_parameters ->energyHistogramFiles;
energyMagnHistogramFiles = simulations_parameters ->energyMagnHistogramFiles;
betas = simulations_parameters->betas;
sweeps = simulations_parameters->sweeps;
free(simulations_parameters);
//Make histogram Files
make_simulation_en_hist_files(numberOfSimulations,dataFiles,energyHistogramFiles);
//make total en hist file
sprintf(temporaryFileName, "./Output/L%d.%02d%02d%02d%02d%02d/Results/energy_histograms.dat",L,
        time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
        time_struct.tm_min,time_struct.tm_sec);
make_total_en_hist_file(numberOfSimulations, energyHistogramFiles,temporaryFileName);
//calculate the converged partitionFunctions
partitionFunctions = malloc(numberOfSimulations*sizeof(long double));
partitionFunctions = estimate_partition_functions(numberOfSimulations,betas,sweeps,
        temporaryFileName);
//Calculate the range of interpolation
min_beta=betas[0];
max_beta=betas[0];
for(i=0; i<numberOfSimulations; i++)
{
    if(min_beta > betas[i])
    {
        min_beta=betas[i];
    }
    if(max_beta < betas[i])
    {
        max_beta=betas[i];
    }
}
sprintf(temporaryFileName2, "./Output/L%d.%02d%02d%02d%02d%02d/Results/energy.dat",L,time_struct
        .tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,
        time_struct.tm_sec);
estimate_internal_energy(min_beta,max_beta,numberOfSimulations,L, betas,sweeps,
        energyHistogramFiles,partitionFunctions,temporaryFileName,temporaryFileName2, step);
if(calculateMagn ==1)
{
    // make_simulation_enmagn_hist_files(dataFiles,energyMagnHistogramFiles,numberOfSimulations);
    sprintf(temporaryFileName, "./Output/L%d.%02d%02d%02d%02d%02d/Results/
        magnetization_histograms.dat",L,time_struct.tm_year-100,time_struct.tm_mon+1,
        time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,time_struct.tm_sec);
    // make_total_enmagn_hist_file(numberOfSimulations, energyMagnHistogramFiles,
        temporaryFileName);
    sprintf(temporaryFileName2, "./Output/L%d.%02d%02d%02d%02d%02d/Results/magnetization.dat",
        L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
        time_struct.tm_min,time_struct.tm_sec);
    // estimate_magnetization(min_beta,max_beta,numberOfSimulations,L, betas,sweeps,
        energyMagnHistogramFiles,partitionFunctions,temporaryFileName,temporaryFileName2, step);
    estimate_magnetization(min_beta,max_beta,numberOfSimulations,L, betas,sweeps,dataFiles,
        partitionFunctions,temporaryFileName,temporaryFileName2, step);
}

```



```

free(partitionFunctions);
if(calculateErrors == 1)
{
    make_blocking_data_files(dataFiles,numberOfSimulations,sweeps,L,betas,time_struct);
    blocking_method(dataFiles,numberOfSimulations,sweeps,L,betas,min_beta,max_beta,calculateMagn,
        time_struct,step);
    sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Results/energy_errors.dat",L,
        time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
        time_struct.tm_min,time_struct.tm_sec);
    sprintf(temporaryFileName2,"./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Results/energyN",L,
        time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
        time_struct.tm_min,time_struct.tm_sec);
    blocking_results(betas,numberOfSimulations,L,temporaryFileName,temporaryFileName2);
    if(calculateMagn==1)
    {
        sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Results/magnetization_errors.
            dat",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour
            ,time_struct.tm_min,time_struct.tm_sec);
        sprintf(temporaryFileName2,"./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Results/magnN",L,
            time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
            time_struct.tm_min,time_struct.tm_sec);
        blocking_results(betas,numberOfSimulations,L,temporaryFileName,temporaryFileName2);
    }
}

if(bootstrap>1)
{
    make_bootstrap_data(min_beta,max_beta,dataFiles,sweeps,betas,numberOfSimulations,L,bootstrap,
        calculateMagn,time_struct,step);

    sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Results/
        energy_errors_bootstrap.dat",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.
        tm_mday,time_struct.tm_hour,time_struct.tm_min,time_struct.tm_sec);
    sprintf(temporaryFileName2,"./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Results/energyN",L,
        time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
        time_struct.tm_min,time_struct.tm_sec);
    calculate_bootstrap_results(betas,numberOfSimulations,L,temporaryFileName,temporaryFileName2,
        bootstrap);
    if(calculateMagn ==1)
    {
        sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Results/
            magnetization_errors_bootstrap.dat",L,time_struct.tm_year-100,time_struct.tm_mon+1,
            time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,time_struct.tm_sec);
        sprintf(temporaryFileName2,"./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Results/magnN",L,
            time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
            time_struct.tm_min,time_struct.tm_sec);
        calculate_bootstrap_results(betas,numberOfSimulations,L,temporaryFileName,temporaryFileName2,
            bootstrap);
    }
}

if(makePlots==1)
{
    make_plots(L,calculateMagn,time_struct);
}

free(sweeps);
free(betas);
free(inputFile);
for(i=0; i<numberOfSimulations; i++)

```

```

    {
        free(dataFiles[i]);
        free(energyHistogramFiles[i]);
    }
free(dataFiles);
free(energyHistogramFiles);
if(calculateMagn==1)
{
    for(i=0; i<numberOfSimulations; i++)
    {
        free(energyMagnHistogramFiles[i]);
    }
    free(energyMagnHistogramFiles);
}
}
struct user_input* get_user_input(int argc, char **argv)
{
    int c,L,bootstrap,calculateErrors,calculateMagn,makePlots;
    char *inputFile;
    struct user_input *arguments;
    double step;

    arguments = malloc(sizeof(struct user_input));
    calculateErrors= 0;
    calculateMagn= 0;
    makePlots=0;
    bootstrap=0;
    step=0.001;
    while ((c = getopt(argc,argv,"L:f:b:s:emp")) != -1)
    {
        switch(c)
        {
        case'L':
            L=atoi(optarg);
            break;
        case'e':
            calculateErrors = 1;
            break;
        case'm':
            calculateMagn = 1;
            break;
        case'p':
            makePlots = 1;
            break;
        case'b':
            bootstrap=atoi(optarg);
            break;
        case's':
            step =atof(optarg);
            break;
        case'f':
            inputFile = malloc(sizeof(char) * (strlen(optarg) + 1 ));
            sprintf(inputFile,"%s",optarg);
            break;
        }
    }
    arguments-> L = L;
    arguments-> inputFile = inputFile;
}

```

```

arguments-> calculateErrors = calculateErrors;
arguments-> calculateMagn = calculateMagn;
arguments-> makePlots = makePlots;
arguments -> bootstrap = bootstrap;
arguments -> step = step;
return arguments;
}
struct simulations_parameters* set_input_parameters(char* inputFile,int L,struct tm time_struct)
{
int i,numberOfSimulations,temporarySweeps;
int *sweeps;
FILE *fp;
float temporaryBeta;
char temporaryFileName[100];
char **dataFiles,**energyHistogramFiles, **energyMagnHistogramFiles;
double *betas;
struct simulations_parameters *sim_param;

sim_param=malloc(sizeof(struct simulations_parameters));
numberOfSimulations=0;
fp=fopen(inputFile, "r");
while(fscanf(fp, "%s %f %d",temporaryFileName,&temporaryBeta,&temporarySweeps) == 3)
{
    numberOfSimulations++;
}
energyHistogramFiles = malloc(numberOfSimulations * sizeof(char*));
energyMagnHistogramFiles = malloc(numberOfSimulations * sizeof(char*));
dataFiles = malloc(numberOfSimulations * sizeof(char*));
betas=malloc(numberOfSimulations * sizeof(double));
sweeps = malloc(numberOfSimulations * sizeof(int));
rewind(fp);
i=0;
while(fscanf(fp, "%s %f %d",temporaryFileName,&temporaryBeta,&temporarySweeps) == 3)
{
    betas[i]=temporaryBeta;
    sweeps[i]=temporarySweeps;
    dataFiles[i] = malloc((strlen(temporaryFileName)+1) * sizeof(char));
    strcpy( dataFiles[i],temporaryFileName);
    sprintf(temporaryFileName, "./Output/L%d.%02d%02d%02d%02d%02d/Energy_Histograms/enhistL%db
        %.4f.dat",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.
        tm_hour,time_struct.tm_min,time_struct.tm_sec,L,betas[i]);
    energyHistogramFiles[i] = malloc((strlen(temporaryFileName)+1) * sizeof(char));
    strcpy(energyHistogramFiles[i], temporaryFileName);
    sprintf(temporaryFileName, "./Output/L%d.%02d%02d%02d%02d%02d/Energy_Histograms/enmagnhistL%
        db%.4f.dat",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct
        .tm_hour,time_struct.tm_min,time_struct.tm_sec,L,betas[i]);
    energyMagnHistogramFiles[i] = malloc((strlen(temporaryFileName)+1) * sizeof(char));
    strcpy(energyMagnHistogramFiles[i], temporaryFileName);
    i++;
}
fclose(fp);
sim_param -> numberOfSimulations= numberOfSimulations;
sim_param -> dataFiles = dataFiles;
sim_param -> energyHistogramFiles = energyHistogramFiles;
sim_param -> energyMagnHistogramFiles = energyMagnHistogramFiles;
sim_param -> betas = betas;
sim_param -> sweeps = sweeps;
return sim_param;

```

```

}
/*
 *NOTICE: There is an alternate function at the end of the file for the
 *creation of the histogram files without the use of arrays but with
 *file manipulation instead. You need only comment this one out and
 *remove commenting from the alternate for the method to run.
 */
void make_simulation_en_hist_files(int numberOfSimulations, char **dataFiles, char **
energyHistogramFiles)
{
    int i,c,j,index,arrayCount,k;
    int *arrayOfEnergies, *arrayOfHistograms;
    FILE *filePath;
    for(i=0; i<numberOfSimulations; i++)
    {
        arrayOfEnergies=malloc(sizeof(int));
        arrayOfHistograms=malloc(sizeof(int));
        filePath=fopen(dataFiles[i],"r");
        arrayCount=0;
        if(fscanf(filePath,"%d %d",&c) == 1)
        {
            arrayOfEnergies[arrayCount]= c;
            arrayOfHistograms[arrayCount]= 1;
            arrayCount++;
        }
        while(fscanf(filePath,"%d %d",&c) == 1)
        {
            index=-1;
            for(k=0; k < arrayCount; k++)
            {
                if(arrayOfEnergies[k] == c)
                {
                    index=k;
                    break;
                }
            }
            if(index == -1)
            {
                arrayOfEnergies=realloc(arrayOfEnergies,(arrayCount+1)*sizeof(int));
                arrayOfHistograms=realloc(arrayOfHistograms,(arrayCount+1)*sizeof(int));
                arrayOfEnergies[arrayCount]= c;
                arrayOfHistograms[arrayCount]= 1;
                arrayCount++;
            }
            else if(index != -1)
            {
                arrayOfHistograms[index]++;
            }
        }
        fclose(filePath);
        filePath=fopen(energyHistogramFiles[i],"w");
        for(j=0; j<arrayCount; j++)
        {
            fprintf(filePath,"%d %d \n",arrayOfEnergies[j],arrayOfHistograms[j]);
        }
        fclose(filePath);
        free(arrayOfEnergies);
        free(arrayOfHistograms);
    }
}

```

```

    }
}
void make_simulation_enmagn_hist_files(char **dataFiles, char **energyMagnHistogramFiles, int
    numberOfSimulations)
{
    int i,c,d,j,index,arrayCount,k;
    FILE *filePath;
    int *arrayOfHistograms;
    int **arrayOfEnergiesMagnetisms;

    for(i=0; i<numberOfSimulations; i++)
    {
        arrayOfEnergiesMagnetisms=malloc(sizeof(int*));
        arrayOfHistograms=malloc(sizeof(int));
        filePath=fopen(dataFiles[i],"r");
        arrayCount=0;
        if(fscanf(filePath,"%d %d",&c, &d) == 2)
        {
            arrayOfEnergiesMagnetisms[arrayCount]= malloc(2 * sizeof(int));
            arrayOfEnergiesMagnetisms[arrayCount][0]= c;
            arrayOfEnergiesMagnetisms[arrayCount][1]= d;
            arrayOfHistograms[arrayCount]= 1;
            arrayCount++;
        }
        while(fscanf(filePath,"%d %d",&c, &d) == 2)
        {
            index=-1;
            for(k=0; k< arrayCount; k++)
            {
                if(arrayOfEnergiesMagnetisms[k][0] == c && arrayOfEnergiesMagnetisms[k][1] == d)
                {
                    index=k;
                    break;
                }
            }
            if(index == -1)
            {
                arrayOfEnergiesMagnetisms=realloc(arrayOfEnergiesMagnetisms, (arrayCount+1)*sizeof(int*));
                arrayOfEnergiesMagnetisms[arrayCount]=malloc(2 * sizeof(int));
                arrayOfHistograms=realloc(arrayOfHistograms, (arrayCount+1)*sizeof(int));
                arrayOfEnergiesMagnetisms[arrayCount][0]= c;
                arrayOfEnergiesMagnetisms[arrayCount][1]= d;
                arrayOfHistograms[arrayCount]= 1;
                arrayCount++;
            }
            else if(index != -1)
            {
                arrayOfHistograms[index]++;
            }
        }
        fclose(filePath);
        filePath=fopen(energyMagnHistogramFiles[i],"w");
        for(j=0; j<arrayCount; j++)
        {
            fprintf(filePath,"%d %d %d \n",arrayOfEnergiesMagnetisms[j][0],arrayOfEnergiesMagnetisms[j][1],
                arrayOfHistograms[j]);
        }
        fclose(filePath);
    }
}

```

```

    for(j=0; j<arrayCount; j++)
    {
        free(arrayOfEnergiesMagnetisms[j]);
    }
    free(arrayOfEnergiesMagnetisms);
    free(arrayOfHistograms);
}
}
void make_total_en_hist_file(int numberOfSimulations, char **histogramFiles, char energy_hist_file[])
{
    FILE *filePath,*filePath2;
    int i,j,energy,histogram,energy2,histogram2,energyMatch,histSum;

    filePath2=fopen(energy_hist_file,"w");
    fclose(filePath2);
    for(i=0; i<numberOfSimulations; i++)
    {
        filePath=fopen(histogramFiles[i],"r");
        while(fscanf(filePath,"%d %d",&energy,&histogram) == 2)
        {
            energyMatch=0;
            filePath2=fopen(energy_hist_file,"r");
            while(fscanf(filePath2,"%d %d",&energy2,&histogram2) ==2)
            {
                if(energy== energy2)
                {
                    energyMatch=1;
                    break;
                }
            }
            fclose(filePath2);
            if(energyMatch==0)
            {
                histSum=histogram;
                for(j=0; j<numberOfSimulations; j++)
                {
                    if( j != i)
                    {
                        filePath2=fopen(histogramFiles[j],"r");
                        while(fscanf(filePath2,"%d %d",&energy2,&histogram2) == 2)
                        {
                            if(energy== energy2)
                            {
                                histSum+=histogram2;
                            }
                        }
                        fclose(filePath2);
                    }
                }
                filePath2=fopen(energy_hist_file,"a");
                fprintf(filePath2,"%d %d \n",energy,histSum);
                fclose(filePath2);
            }
        }
        fclose(filePath);
    }
}
long double* estimate_partition_functions(int numberOfSimulations, double *betas, int *sweeps, char

```

```

    energy_hist_file[])
{
FILE *filePath;
int i,j,k,energy,histogram,converged, count;;
long double denominatorSum,resultant_partition,resultant_energy,resultant_energy2,eps,zmax,zmin,
    anor,tmp;
long double *tempPartitionFunctions,*tempPartitionFunctions2;

zmax=1;
zmin=1;
converged=0;
eps=0.0000001;
count=0;
tempPartitionFunctions=malloc(numberOfSimulations*sizeof(long double));
tempPartitionFunctions2=malloc(numberOfSimulations*sizeof(long double));
for(j=0; j<numberOfSimulations; j++)
{
    tempPartitionFunctions2[j]=1;
}
while(1)
{
    for(j=0; j<numberOfSimulations; j++)
    {
        tempPartitionFunctions[j]=tempPartitionFunctions2[j];
        if(zmin > tempPartitionFunctions2[j])
        {
            zmin=tempPartitionFunctions2[j];
        }
        if(zmax < tempPartitionFunctions2[j])
        {
            zmax=tempPartitionFunctions2[j];
        }
    }
    anor=1/(sqrtl(zmax*zmin));
    for(j=0; j<numberOfSimulations; j++)
    {
        resultant_partition=0;
        filePath=fopen(energy_hist_file,"r");
        while(fscanf(filePath,"%d %d",&energy,&histogram)==2)
        {
            denominatorSum=0;
            for(i=0; i<numberOfSimulations; i++)
            {
                denominatorSum+=(sweeps[i]*exp((betas[j]-betas[i])*energy))/tempPartitionFunctions[i];
            }
            resultant_partition+=histogram/denominatorSum;
        }
        fclose(filePath);
        tempPartitionFunctions2[j]=resultant_partition*anor;
    }
    count++;
    for(j=0; j<numberOfSimulations; j++)
    {
        tmp=log(2)+logl(fabsl(tempPartitionFunctions[j]-tempPartitionFunctions2[j]))-logl(fabsl(
            tempPartitionFunctions[j]+tempPartitionFunctions2[j]));
        if( expl(tmp) < eps)
        {
            converged=1;

```

```

        break;
    }
}
if(converged==1)
{
    printf("They converged \n");
    break;
}
}
printf("Final count is %d \n",count);
free(tempPartitionFunctions2);
return tempPartitionFunctions;
}
void estimate_internal_energy(double min_beta,double max_beta,int numberOfSimulations,int L, double
    *betas,int *sweeps,char **histogramFiles,long double *partitionFunctions,char energy_hist_file
    [],char outputFile[], double step)
{
    FILE *filePath,*filePath2;
    int i,j,energy,histogram,energy2,histogram2,energyMatch;
    double tmp_beta;
    long double denominatorSum,resultant_partition,resultant_energy,resultant_energy2;

    filePath=fopen(outputFile,"w");
    fclose(filePath);
    tmp_beta=min_beta;
    while(tmp_beta<max_beta)
    {
        resultant_energy=0;
        resultant_partition=0;
        resultant_energy2=0;
        filePath=fopen(energy_hist_file,"r");
        while(fscanf(filePath,"%d %d",&energy,&histogram)==2)
        {
            denominatorSum=0;
            for(i=0; i<numberOfSimulations; i++)
            {
                denominatorSum+=(sweeps[i]*exp((tmp_beta-betas[i])*energy))/partitionFunctions[i];
            }
            resultant_partition+=histogram/denominatorSum;
            resultant_energy+=energy*(histogram/denominatorSum);
            resultant_energy2+=energy*energy*(histogram/denominatorSum);
            resultant_energy2+=(energy/(2*L*L))*(energy/(2*L*L))*(histogram/denominatorSum);
        }
        fclose(filePath);
        resultant_energy= ((1/resultant_partition)*resultant_energy)/(2*L*L);
        resultant_energy2= ((1/resultant_partition)*resultant_energy2)/(4*L*L*L*L);
        filePath=fopen(outputFile,"a");
        fprintf(filePath,"%lf %.20Lf %.20Lf \n",tmp_beta,resultant_energy, tmp_beta*tmp_beta*L*L*(
            resultant_energy2-(resultant_energy*resultant_energy)));
        fclose(filePath);
        tmp_beta+= step;
    }
}
void make_total_enmagn_hist_file(int numberOfSimulations, char **histogramFiles, char
    energy_hist_file[])
{
    FILE *filePath,*filePath2;
    int i,j,energy,magnetization,histogram,energy2,magnetization2,histogram2,energyMatch,histSum;

```



```

filePath2=fopen(energy_hist_file,"w");
fclose(filePath2);
for(i=0; i<numberOfSimulations; i++)
{
    filePath=fopen(histogramFiles[i],"r");
    while(fscanf(filePath,"%d %d %d",&energy,&magnetization,&histogram) == 3)
    {
        energyMatch=0;
        filePath2=fopen(energy_hist_file,"r");
        while(fscanf(filePath2,"%d %d %d",&energy2,&magnetization2,&histogram2) == 3 )
        {
            if(energy== energy2 && magnetization==magnetization2)
            {
                energyMatch=1;
                break;
            }
        }
        fclose(filePath2);
        if(energyMatch==0)
        {
            histSum=histogram;
            for(j=0; j<numberOfSimulations; j++)
            {
                if( j != i)
                {
                    filePath2=fopen(histogramFiles[j],"r");
                    while(fscanf(filePath2,"%d %d %d",&energy2,&magnetization2,&histogram2) == 3)
                    {
                        if(energy== energy2 && magnetization==magnetization2)
                        {
                            histSum+=histogram2;
                        }
                    }
                    fclose(filePath2);
                }
            }
            filePath2=fopen(energy_hist_file,"a");
            fprintf(filePath2,"%d %d %d \n",energy,magnetization,histSum);
            fclose(filePath2);
        }
    }
    fclose(filePath);
}
}
/*
*Alternate Function. Read comment on function with same name above.
*/
void estimate_magnetization(double min_beta,double max_beta,int numberOfSimulations,int L, double *
    betas,int *sweeps,char **dataFiles,long double *partitionFunctions,char energy_hist_file[],char
    outputFile[], double step)
{
    FILE *filePath,*filePath2;
    int i,j,energy,histogram,magnetism,energy2,histogram2,magnetism2,energyMatch;
    double tmp_beta;
    long double denominatorSum,resultant_partition,resultant_magnetism,resultant_magnetism2;

    filePath=fopen(outputFile,"w");
    fclose(filePath);

```

```

tmp_beta=min_beta;
while(tmp_beta<max_beta)
{
    resultant_magnetism=0;
    resultant_partition=0;
    resultant_magnetism2=0;
    for(j=0; j< numberOfSimulations; j++)
    {
        filePath=fopen(dataFiles[j],"r");
        while(fscanf(filePath,"%d %d",&energy,&magnetism)==2)
        {
            denominatorSum=0;
            for(i=0; i<numberOfSimulations; i++)
            {
                denominatorSum+=(sweeps[i]*exp((tmp_beta-betas[i])*energy))/partitionFunctions[i];
            }
            resultant_partition+=1/denominatorSum;
            resultant_magnetism+=abs(magnetism)*(1/denominatorSum);
            resultant_magnetism2+=magnetism*magnetism*(1/denominatorSum);
        }
        fclose(filePath);
    }
    resultant_magnetism= ((1/resultant_partition)*resultant_magnetism)/(L*L);
    resultant_magnetism2= ((1/resultant_partition)*resultant_magnetism2)/(L*L*L*L);
    filePath=fopen(outputFile,"a");
    fprintf(filePath,"%lf %.20Lf %.20Lf \n",tmp_beta,resultant_magnetism, tmp_beta*L*L*(
        resultant_magnetism2-(resultant_magnetism*resultant_magnetism)));
    fclose(filePath);
    tmp_beta+= step;
}
}

void make_blocking_data_files(char **dataFiles, int numberOfSimulations, int *sweeps, int L, double
    *betas,struct tm time_struct)
{
    int i,j,k,c,d;
    char buf[100],buf2[100];
    FILE *fp, *fp2;
    for(i=0; i<numberOfSimulations; i++)
    {
        k=1;
        j=1;
        sprintf(buf, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Data/blockN%dL%db%.4f.dat",L,
            time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
            time_struct.tm_min,time_struct.tm_sec,j,L,betas[i]);
        fp=fopen(dataFiles[i],"r");
        fp2=fopen(buf,"w");

        while(fscanf(fp,"%d %d",&c,&d) == 2)
        {
            fprintf(fp2,"%d %d \n",c,d);

            if(k == (sweeps[0]/10) && j <10)
            {
                k=0;
                j++;
                fclose(fp2);
            }
        }
    }
}

```

```

        sprintf(buf, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Data/blockN%dL%db%.4f.dat",L,
                time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
                time_struct.tm_min,time_struct.tm_sec,j,L,betas[i]);
        fp2=fopen(buf,"w");
    }
    k++;
}
fclose(fp);
fclose(fp2);
}
}
void blocking_method(char **dataFiles,int numberOfSimulations,int *sweeps,int L,double *betas,double
                    min_beta, double max_beta,int calculateMagn,struct tm time_struct, double step)
{
    FILE *fp, *fp2;
    int i,j,k,c,d;
    int *sweeps2;
    char buf[200], buf2[200];
    char **block_data_files, ** block_energy_hist_files, ** block_energy_magn_hist_files;
    long double *partitionFunctions2;
    sweeps2=malloc(numberOfSimulations * sizeof(int));
    block_data_files = malloc(numberOfSimulations * sizeof(char*));
    block_energy_hist_files = malloc(numberOfSimulations * sizeof(char*));
    if(calculateMagn==1)
    {
        block_energy_magn_hist_files = malloc(numberOfSimulations * sizeof(char*));
    }
    for(i=0; i<numberOfSimulations; i++)
    {
        sweeps2[i]= sweeps[i]/10;
        block_data_files[i]=malloc((200) * sizeof(char));
        block_energy_hist_files[i]=malloc((200) * sizeof(char));

        if(calculateMagn==1)
        {
            block_energy_magn_hist_files[i]=malloc((200) * sizeof(char));
        }
    }
    for(j=1; j<=10; j++)
    {
        for(i=0; i<numberOfSimulations; i++)
        {
            sprintf(buf, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Data/blockN%dL%db%.4f.dat",L,
                    time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
                    time_struct.tm_min,time_struct.tm_sec,j,L,betas[i]);
            strcpy( block_data_files[i],buf);

            sprintf(buf2, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Data/enhistN%dL%db%.4f.dat",L,
                    time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
                    time_struct.tm_min,time_struct.tm_sec,j,L,betas[i]);
            strcpy( block_energy_hist_files[i],buf2);
            if(calculateMagn==1)
            {
                sprintf(buf2, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Data/enmagnhistN%dL%db%.4f.dat"
                        ,L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour
                        ,time_struct.tm_min,time_struct.tm_sec,j,L,betas[i]);
                strcpy( block_energy_magn_hist_files[i],buf2);
            }
        }
    }
}

```

```

    }
}
make_simulation_en_hist_files(numberOfSimulations,block_data_files,block_energy_hist_files);
sprintf(buf, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Data/energy_histogramsN%d.dat",L,
    time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
    time_struct.tm_min,time_struct.tm_sec,j);
make_total_en_hist_file(numberOfSimulations, block_energy_hist_files,buf);
partitionFunctions2 = estimate_partition_functions(numberOfSimulations,betas,sweeps2,buf);
sprintf(buf2, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Results/energyN%d.dat",L,
    time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
    time_struct.tm_min,time_struct.tm_sec,j);
estimate_internal_energy(min_beta,max_beta,numberOfSimulations,L, betas,sweeps2,
    block_energy_hist_files,partitionFunctions2,buf,buf2,step);
if(calculateMagn==1)
{
// make_simulation_enmagn_hist_files(block_data_files,block_energy_magn_hist_files,
    numberOfSimulations);
sprintf(buf, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Data/magnetization_histogramsN%d.
    dat",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour
    ,time_struct.tm_min,time_struct.tm_sec,j);
// make_total_enmagn_hist_file(numberOfSimulations, block_energy_magn_hist_files,buf);
sprintf(buf2, "./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Results/magnN%d.dat",L,time_struct
    .tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.
    tm_min,time_struct.tm_sec,j);
// estimate_magnetization(min_beta,max_beta,numberOfSimulations,L, betas,sweeps2,
    block_energy_magn_hist_files,partitionFunctions2,buf,buf2,step);
estimate_magnetization(min_beta,max_beta,numberOfSimulations,L, betas,sweeps2,block_data_files,
    partitionFunctions2,buf,buf2,step);
}
free(partitionFunctions2);
}
for(i=0; i<numberOfSimulations; i++)
{
free(block_data_files[i]);
free(block_energy_hist_files[i]);
if(calculateMagn==1)
{
free(block_energy_magn_hist_files[i]);
}
}
free(block_data_files);
free(block_energy_hist_files);
if(calculateMagn==1)
{
free(block_energy_magn_hist_files);
}
free(sweeps2);
}
void blocking_results(double *betas, int numberOfSimulations,int L,char errorFile[], char interFile
    [])
{
int i,j;
double beta,beta2,tmp_beta;
long double value1,value2,value3,value4, meanQuantity1, meanQuantity1_2,meanQuantity2,
    meanQuantity2_2, Quantity1_Error, Quantity2_Error;
char fileName[100], fileName2[100], buf[100];
FILE *fp, *fp2;
strcpy(fileName,interFile);

```

```

strcat(fileName,"1.dat");
fp=fopen(errorFile,"w");
fclose(fp);
fp=fopen(fileName,"r");
while(fscanf(fp,"%lf %Lf %Lf",&beta,&value1, &value3) == 3)
{
    meanQuantity1=value1;
    meanQuantity1_2=powl(value1,2);
    meanQuantity2=value3;
    meanQuantity2_2=powl(value3,2);
    for(j=2; j<=10; j++)
    {
        strcpy(fileName2,interFile);
        sprintf(buf,"%d.dat",j);
        strcat(fileName2,buf);
        fp2=fopen(fileName2,"r");
        while(fscanf(fp2,"%lf %Lf %Lf",&beta2,&value2,&value4) == 3)
        {
            if(fabs(beta-beta2) <= 0.001)
            {
                meanQuantity1+=value2;
                meanQuantity1_2+=powl(value2,2);
                meanQuantity2+=value4;
                meanQuantity2_2+=powl(value4,2);
                break;
            }
        }
        fclose(fp2);
    }
    meanQuantity1= meanQuantity1/10;
    meanQuantity1_2= meanQuantity1_2/10;
    meanQuantity2=meanQuantity2/10;
    meanQuantity2_2=meanQuantity2_2/10;
    Quantity1_Error= sqrt1((meanQuantity1_2-powl(meanQuantity1,2))/9);
    Quantity2_Error= sqrt1((meanQuantity2_2-powl(meanQuantity2,2))/9);
    fp2=fopen(errorFile,"a");
    fprintf(fp2,"%lf %.20Lf %.20Lf \n",beta,Quantity1_Error,Quantity2_Error);
    fclose(fp2);
}
fclose(fp);
}
void make_bootstrap_data(double min_beta,double max_beta,char **dataFiles,int *sweeps,double *betas,
    int numberOfSimulations,int L, int bootstrap,int calculateMagn,struct tm time_struct, double
    step)
{
    FILE *fp,*fp2;
    char buf[200],buf2[200],fileLine[200];
    int lineLength,fileSize,energy,magnetization,offset,count,i,j;
    char **block_data_files,**block_energy_hist_files,**block_energy_magn_hist_files;
    long double *partitionFunctions2;

    block_data_files = malloc(numberOfSimulations * sizeof(char*));
    block_energy_hist_files = malloc(numberOfSimulations * sizeof(char*));
    if(calculateMagn ==1)
    {
        block_energy_magn_hist_files = malloc(numberOfSimulations * sizeof(char*));
    }
    for(i=0; i<numberOfSimulations; i++)

```

```

{
    block_data_files[i]=malloc((150) * sizeof(char));
    block_energy_hist_files[i]=malloc((150) * sizeof(char));
    if(calculateMagn==1)
    {
        block_energy_magn_hist_files[i]=malloc((150) * sizeof(char));
    }
}
}
srand(time(NULL));
for(j=1; j<=bootstrap; j++)
{
    for(i=0; i<numberOfSimulations; i++)
    {
        sprintf(buf2, "./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Data/enhistN%dL%db%.4f.dat",L,
            time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
            time_struct.tm_min,time_struct.tm_sec,j,L,betas[i]);
        strcpy( block_energy_hist_files[i],buf2);
        if(calculateMagn==1)
        {
            sprintf(buf2, "./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Data/enmagnhistN%dL%db%.4f.dat
                ",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.
                tm_hour,time_struct.tm_min,time_struct.tm_sec,j,L,betas[i]);
            strcpy( block_energy_magn_hist_files[i],buf2);
        }
        sprintf(buf, "./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Data/bootL%db%.4fN%d.dat",L,
            time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
            time_struct.tm_min,time_struct.tm_sec,L,betas[i],j);
        strcpy( block_data_files[i],buf);
        fp2=fopen(buf,"w");
        fclose(fp2);
        fp=fopen(dataFiles[i],"r");
        fseek(fp, 0L, SEEK_END);
        fileSize = ftell(fp);
        fseek(fp, 0L, SEEK_SET );
        count=0;
        while(count < sweeps[i])
        {
            offset=rand() % fileSize;
            fseek(fp,offset, SEEK_SET);
            if(fgets(fileLine,100,fp)!= NULL)
            {
                if(fgets(fileLine,100,fp) != NULL)
                {
                    lineLength=strlen(fileLine) -1;
                    if(fileLine[lineLength] == '\n')
                    {
                        fileLine[lineLength] == '\0';
                    }
                    sscanf(fileLine,"%d %d",&energy, &magnetization);
                    fp2=fopen(buf,"a");
                    fprintf(fp2,"%d %d \n",energy,magnetization);
                    fclose(fp2);
                    count++;
                }
            }
        }
        else
        {
            fseek(fp,0L, SEEK_SET);
            if(fgets(fileLine,100,fp) != NULL)

```

```

    {
        lineLength=strlen(fileLine) -1;
        if(fileLine[lineLength] == '\n')
        {
            fileLine[lineLength] == '\0';
        }
        sscanf(fileLine,"%d %d",&energy, &magnetization);
        fp2=fopen(buf,"a");
        fprintf(fp2,"%d %d \n",energy,magnetization);
        fclose(fp2);
        count++;
    }
}
}
else
{
    continue;
}
}
fclose(fp);
}
make_simulation_en_hist_files(numberOfSimulations,block_data_files,block_energy_hist_files);
sprintf(buf,"./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Data/enhistN%d",L,time_struct.
tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min
,time_struct.tm_sec,j);
make_total_en_hist_file(numberOfSimulations, block_energy_hist_files,buf);
partitionFunctions2 = estimate_partition_functions(numberOfSimulations,betas,sweeps,buf);
sprintf(buf2,"./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Results/energyN%ddata.dat",L,
time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
time_struct.tm_min,time_struct.tm_sec,j);
estimate_internal_energy(min_beta,max_beta,numberOfSimulations,L, betas,sweeps,
block_energy_hist_files,partitionFunctions2,buf,buf2,step);
if(calculateMagn==1)
{
// make_simulation_enmagn_hist_files(block_data_files,block_energy_magn_hist_files,
// numberOfSimulations);
sprintf(buf,"./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Data/enmagnhistN%d",L,time_struct.
tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min
,time_struct.tm_sec,j);
// make_total_enmagn_hist_file(numberOfSimulations, block_energy_magn_hist_files,buf);
sprintf(buf2,"./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Results/magnN%ddata.dat",L,
time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
time_struct.tm_min,time_struct.tm_sec,j);
// estimate_magnetization(min_beta,max_beta,numberOfSimulations,L, betas,sweeps,
// block_energy_magn_hist_files,partitionFunctions2,buf,buf2,step);
estimate_magnetization(min_beta,max_beta,numberOfSimulations,L, betas,sweeps,block_data_files,
partitionFunctions2,buf,buf2,step);
}
free(partitionFunctions2);
for(i=0; i<numberOfSimulations; i++)
{
remove(block_data_files[i]);
}
}
for(i=0; i<numberOfSimulations; i++)
{
free(block_data_files[i]);
free(block_energy_hist_files[i]);
}

```

```

        if(calculateMagn==1)
        {
            free(block_energy_magn_hist_files[i]);
        }
    }
    free(block_data_files);
    free(block_energy_hist_files);
    if(calculateMagn==1)
    {
        free(block_energy_magn_hist_files);
    }
}
void calculate_bootstrap_results(double *betas, int numberOfSimulations,int L,char errorFile[], char
    interFile[],int bootstrap)
{
    int i,j;
    FILE *fp, *fp2;
    char fileName[100],fileName2[100],buf[100];
    double beta,beta2,tmp_beta;
    long double value1,value2,value3,value4, meanQuantity1, meanQuantity1_2,meanQuantity2,
        meanQuantity2_2, Quantity1_Error, Quantity2_Error;

    strcpy(fileName,interFile);
    strcat(fileName,"1data.dat");
    fp=fopen(errorFile,"w");
    fclose(fp);
    fp=fopen(fileName,"r");
    while(fscanf(fp,"%lf %Lf %Lf",&beta,&value1, &value3) == 3)
    {
        meanQuantity1=value1;
        meanQuantity1_2=powl(value1,2);
        meanQuantity2=value3;
        meanQuantity2_2=powl(value3,2);
        for(j=2; j<=bootstrap; j++)
        {
            strcpy(fileName2,interFile);
            sprintf(buf,"%ddata.dat",j);
            strcat(fileName2,buf);
            fp2=fopen(fileName2,"r");
            while(fscanf(fp2,"%lf %Lf %Lf",&beta2,&value2,&value4) == 3)
            {
                if(fabs(beta-beta2) <= 0.001)
                {
                    meanQuantity1+=value2;
                    meanQuantity1_2+=powl(value2,2);
                    meanQuantity2+=value4;
                    meanQuantity2_2+=powl(value4,2);
                    break;
                }
            }
            fclose(fp2);
        }
        meanQuantity1= meanQuantity1/bootstrap;
        meanQuantity1_2= meanQuantity1_2/bootstrap;
        meanQuantity2=meanQuantity2/bootstrap;
        meanQuantity2_2=meanQuantity2_2/bootstrap;
        Quantity1_Error= sqrtl((meanQuantity1_2-powl(meanQuantity1,2)));
        Quantity2_Error= sqrtl((meanQuantity2_2-powl(meanQuantity2,2)));
    }
}

```



```

        fp2=fopen(errorFile,"a");
        fprintf(fp2,"%lf %.20Lf %.20Lf \n",beta,Quantity1_Error,Quantity2_Error);
        fclose(fp2);
    }
    fclose(fp);
}
void create_folder_structure(int L,int makePlots,int calculateErrors,int bootstrap,struct tm
    time_struct)
{
    char temporaryFileName[100];
    struct stat st = {0};
    sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d",L,time_struct.tm_year-100,
        time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,time_struct.
        tm_sec);
    if (stat(temporaryFileName, &st) == -1)
    {
        mkdir(temporaryFileName, 0700);
    }
    sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Results",L,time_struct.tm_year
        -100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,
        time_struct.tm_sec);
    if (stat(temporaryFileName, &st) == -1)
    {
        mkdir(temporaryFileName, 0700);
    }
    sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Energy_Histograms",L,time_struct.
        tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,
        time_struct.tm_sec);
    if (stat(temporaryFileName, &st) == -1)
    {
        mkdir(temporaryFileName, 0700);
    }
    if(makePlots==1)
    {
        sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Plots",L,time_struct.tm_year
            -100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,
            time_struct.tm_sec);
        if (stat(temporaryFileName, &st) == -1)
        {
            mkdir(temporaryFileName, 0700);
        }
    }
    if(calculateErrors==1)
    {
        sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Data",L,time_struct.
            tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.
            tm_min,time_struct.tm_sec);
        if (stat(temporaryFileName, &st) == -1)
        {
            mkdir(temporaryFileName, 0700);
        }
        sprintf(temporaryFileName,"./Output/L%d.%02d%02d%02d%02d%02d/Blocking_Results",L,
            time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
            time_struct.tm_min,time_struct.tm_sec);
        if (stat(temporaryFileName, &st) == -1)
        {
            mkdir(temporaryFileName, 0700);
        }
    }
}

```

```

}
if(bootstrap>1)
{
    sprintf(temporaryFileName, "./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Data",L,time_struct.
        tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.
        tm_min,time_struct.tm_sec);
    if (stat(temporaryFileName, &st) == -1)
    {
        mkdir(temporaryFileName, 0700);
    }
    sprintf(temporaryFileName, "./Output/L%d.%02d%02d%02d%02d%02d/Bootstrap_Results",L,
        time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
        time_struct.tm_min,time_struct.tm_sec);
    if (stat(temporaryFileName, &st) == -1)
    {
        mkdir(temporaryFileName, 0700);
    }
}
}
void make_plots(int L,int calculateMagn,struct tm time_struct)
{
    int c;
    char command[500];
    sprintf(command,"gnuplot -e \"filename=\\'. /Output/L%d.%02d%02d%02d%02d%02d/Results/energy.dat
        \"; filename2=\\'. /Output/L%d.%02d%02d%02d%02d%02d/Plots/energy2.png\\'\" plot_e2N.plt",L,
        time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
        time_struct.tm_min,time_struct.tm_sec,L,time_struct.tm_year-100,time_struct.tm_mon+1,
        time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,time_struct.tm_sec);
    if(c=system(command) == -1){ }
    sprintf(command,"gnuplot -e \"filename=\\'. /Output/L%d.%02d%02d%02d%02d%02d/Results/energy.dat
        \"; filename2=\\'. /Output/L%d.%02d%02d%02d%02d%02d/Plots/specific_heat.png\\'\" plot_specN.
        plt",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,
        time_struct.tm_min,time_struct.tm_sec,L,time_struct.tm_year-100,time_struct.tm_mon+1,
        time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,time_struct.tm_sec);
    if(c=system(command) == -1){ }
    if(calculateMagn==1)
    {
        sprintf(command,"gnuplot -e \"filename=\\'. /Output/L%d.%02d%02d%02d%02d%02d/Results/
            magnetization.dat\\'; filename2=\\'. /Output/L%d.%02d%02d%02d%02d%02d/Plots/magnetization
            .png\\'\" plot_mN.plt",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday,
            time_struct.tm_hour,time_struct.tm_min,time_struct.tm_sec,L,time_struct.tm_year-100,
            time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,
            time_struct.tm_sec);
        if(c=system(command) == -1){ }
        sprintf(command,"gnuplot -e \"filename=\\'. /Output/L%d.%02d%02d%02d%02d%02d/Results/
            magnetization.dat\\'; filename2=\\'. /Output/L%d.%02d%02d%02d%02d%02d/Plots/magnetic_susc
            .png\\'\" plot_susc.plt",L,time_struct.tm_year-100,time_struct.tm_mon+1,time_struct.tm_mday
            ,time_struct.tm_hour,time_struct.tm_min,time_struct.tm_sec,L,time_struct.tm_year-100,
            time_struct.tm_mon+1,time_struct.tm_mday,time_struct.tm_hour,time_struct.tm_min,
            time_struct.tm_sec);
        if(c=system(command) == -1){ }
    }
}
}
/*
*Alternate Function. Read comment on function with same name above.
*/
/*
*/

```

```

void estimate_magnetization(double min_beta,double max_beta,int numberOfSimulations,int L, double *
    betas,int *sweeps,char **histogramFiles,long double *partitionFunctions,char energy_hist_file
    [],char outputFile[], double step)
{
    FILE *filePath,*filePath2;
    int i,j,energy,histogram,magnetism,energy2,histogram2,magnetism2,energyMatch;
    double tmp_beta;
    long double denominatorSum,resultant_partition,resultant_magnetism,resultant_magnetism2;

    filePath=fopen(outputFile,"w");
    fclose(filePath);
    tmp_beta=min_beta;
    while(tmp_beta<max_beta)
    {
        resultant_magnetism=0;
        resultant_partition=0;
        resultant_magnetism2=0;

        filePath=fopen(energy_hist_file,"r");
        while(fscanf(filePath,"%d %d %d",&energy,&magnetism,&histogram)==3)
        {
            denominatorSum=0;
            for(i=0; i<numberOfSimulations; i++)
            {
                denominatorSum+=(sweeps[i]*exp((tmp_beta-betas[i])*energy))/partitionFunctions[i];
            }
            resultant_partition+=histogram/denominatorSum;
            resultant_magnetism+=abs(magnetism)*(histogram/denominatorSum);
            resultant_magnetism2+=magnetism*magnetism*(histogram/denominatorSum);
        }
        fclose(filePath);

        resultant_magnetism= ((1/resultant_partition)*resultant_magnetism)/(L*L);
        resultant_magnetism2= ((1/resultant_partition)*resultant_magnetism2)/(L*L*L*L);

        filePath=fopen(outputFile,"a");
        fprintf(filePath,"%lf %.20Lf %.20Lf \n",tmp_beta,resultant_magnetism, tmp_beta*L*L*(
            resultant_magnetism2-(resultant_magnetism*resultant_magnetism)));
        fclose(filePath);

        tmp_beta+= step;
    }
}
*/
/*
*Alternate Function. Read comment on function with same name above.
*/
/*
void make_simulation_en_hist_files(int numberOfSimulations,char **dataFiles, char **histogramFiles)
{
    FILE *filePath,*filePath2;
    int i,j,energy,histogram,energy2,histogram2,energyMatch,magnetization,magnetization2, tempEnergy,
        histSum, length;
    unsigned long position;

    for(i=0; i<numberOfSimulations; i++)
    {

```

```

    filePath2=fopen(histogramFiles[i],"w");
    fclose(filePath2);

    filePath=fopen(dataFiles[i],"r");
    while(fscanf(filePath,"%d %d",&energy,&magnetization) == 2)
    {
        energyMatch=0;
        filePath2=fopen(histogramFiles[i],"r");
        while(fscanf(filePath2,"%d %d",&energy2,&histogram) ==2)
        {
            if(energy== energy2)
            {
                energyMatch=1;
                break;
            }
        }
        fclose(filePath2);
        if(energyMatch==0)
        {
            fflush(filePath);
            position=ftell(filePath);
            tempEnergy=energy;
            histSum=1;

            while(fscanf(filePath,"%d %d",&energy,&magnetization) == 2)
            {
                if(energy == tempEnergy)
                {
                    histSum++;
                }
            }

            filePath2=fopen(histogramFiles[i],"a");
            fprintf(filePath2,"%d %d \n",tempEnergy,histSum);
            fclose(filePath2);

            fseek(filePath, position, SEEK_SET);
        }
    }
    fclose(filePath);
}
*/

```

---

```

set terminal pngcairo size 525,389 enhanced font 'Verdana,10'
set output filename2

set xlabel "Inverse Temperature"
set xrange [0.40:0.48]
set ylabel "Internal Energy per spin <e>"
set yrange [-0.9:-0.5]
set key off
set xtics 0.01
set ytics 0.1
set xtics scale 0.5
set ytics scale 0.5
set format x "%.2f"
set format y "%.1f"

```

---

```
set style line 81 lt 0 # dashed
set style line 81 lt rgb "#808080"
set grid back ls 81

plot "L20e" using 3:4 with points pt 6 ps 1.2 lc rgb "black", \
filename using 1:2 with lines lw 2 lc rgb "#A00000", \
"in.dat" using 3:4 with points pt 7 ps 1.5 lc rgb "black"
```

---

```
set terminal pngcairo size 525,389 enhanced font 'Verdana,10'
set output filename2
```

```
set xlabel "Inverse Temperature"
set xrange [0.40:0.48]
set ylabel "Specific heat per spin <c>"
set yrange [0.39:0.60]
set key off
set xtics 0.01
set ytics 0.1
set style line 81 lt 0
set style line 81 lt rgb "#808080"
set format x "%.2f"
set format y "%.1f"
set grid back ls 81
set xtics nomirror
set ytics nomirror
```

```
plot "L20e" using 3:6 with points pt 6 ps 1.2 lc rgb "black", \
filename using 1:3 with lines lw 2 lc rgb "#A00000", \
"in.dat" using 3:6 with points pt 7 ps 1.5 lc rgb "black"
```

---

```
set terminal pngcairo size 525,389 enhanced font 'Verdana,10'
set output filename2
```

```
set xlabel "Inverse Temperature"
set xrange [0.40:0.48]
set ylabel "Magnetization per spin <m>"
set yrange [0.30:0.90]
set key off
set xtics 0.01
set ytics 0.1
set xtics scale 0.5
set ytics scale 0.5
set format x "%.2f"
set format y "%.1f"
set style line 81 lt 0 # dashed
set style line 81 lt rgb "#808080"
set grid back ls 81
```

```
plot "L20m" using 3:4 with points pt 6 ps 1.2 lc rgb "black", \
filename using 1:2 with lines lw 2 lc rgb "#A00000", \
"inm.dat" using 3:4 with points pt 7 ps 1.5 lc rgb "black"
```

---

```
set terminal pngcairo size 525,389 enhanced font 'Verdana,10'
set output filename2
```

```
set xlabel "Inverse Temperature"
```

```
set xrange [0.40:0.48]
set ylabel "Magnetic susceptibility per spin "
set yrange [0.5:9.2]
set key off
set xtics 0.01
set xtics scale 0.5
set ytics scale 0.5
set format x "%.2f"
set format y "%.1f"
set style line 80 lt rgb "#808080"
set style line 81 lt 0
set style line 81 lt rgb "#808080"
set grid back ls 81

plot "L20m" using 3:6 with points pt 6 ps 1.2 lc rgb "black", \
filename using 1:3 with lines lw 2 lc rgb "#A00000", \
"inm.dat" using 3:6 with points pt 7 ps 1.5 lc rgb "black"
```

---

