



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ
ΜΑΘΗΜΑΤΙΚΩΝ
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ

ΕΚΕΦΕ «ΔΗΜΟΚΡΙΤΟΣ»

ΙΝΣΤΙΤΟΥΤΟ ΝΑΝΟΕΠΙΣΤΗΜΗΣ ΚΑΙ
ΝΑΝΟΤΕΧΝΟΛΟΓΙΑΣ

ΙΝΣΤΙΤΟΥΤΟ ΠΥΡΗΝΙΚΗΣ ΚΑΙ
ΣΩΜΑΤΙΔΙΑΚΗΣ ΦΥΣΙΚΗΣ



Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών

«Φυσική και Τεχνολογικές Εφαρμογές»

Μελέτη του Μοντέλου Ising σε 2 διαστάσεις με τη μέθοδο Multi Histogram

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του Σάμου Ιωάννη

Ακαδημαϊκός Επιβλέπων

Αν. Καθηγητής Κωνσταντίνος Ν. Αναγνωστόπουλος

Αθήνα

Σεπτέμβριος 2015

Στον Πατέρα μου

Ευχαριστίες

Είμαι ευγνώμων για όλους τους Καθηγητές του Προγράμματος που συνέβαλλαν στη περάτωση των Μεταπτυχιακών σπουδών μου παρέχοντάς μου τις γνώσεις τους. Κλείνοντας να ευχαριστήσω συμφοιτητές και φίλους μου για την επικοινωνιακή τους στάση και στήριξη.

Περίληψη

Σκοπός της παρούσας είναι η χρήση της Multi Histogram για τη μελέτη του προτύπου Ising στην περιοχή μετάβασης φάσης. Η μέθοδος Multi Histogram χρησιμοποιείται για τον ακριβή προσδιορισμό της ψευδοκρίσιμης θερμοκρασίας καθώς και τον υπολογισμό των κρίσιμων εκθετών. Έπειτα αναδεικνύονται οι αδυναμίες της μεθόδου Single Histogram και τα προτερήματα της Multi Histogram ως προς τα όρια εφαρμογής της καθεμιάς. Δοκιμάσαμε να γράψουμε σε C την προσομοίωση Monte Carlo και σε MatLab την εφαρμογή των μεθόδων Single και Multi Histogram.

Περιεχόμενα

1. Στατιστική Φυσική
2. Το μοντέλο Ising
3. Προσομοίωση Monte Carlo
4. Μέθοδος Single - histogram
5. Μέθοδος Multi – histogram
6. Υπολογισμός κρίσιμων εκθετών και κρίσιμης θερμοκρασίας
7. Σφάλματα που προκύπτουν από την εφαρμογή της μεθόδου
8. Συμπεράσματα
9. Κώδικας υλοποίησης
10. Πίνακες Μετρούμενων Μεγεθών

Κεφάλαιο 1^ο

Θεωρώντας ένα σύστημα N αριθμού σωματιδίων, τα οποία αλληλεπιδρούν μεταξύ τους, μπορούμε μέσω της στατιστικής φυσικής να μελετήσουμε το σύστημα μετρώντας διάφορες θερμοδυναμικές ποσότητες, όπως η μέση ενέργεια ανά σωματίδιο και η μαγνήτιση λόγω σπιν.

Για να καταλήξουμε στον υπολογισμό αυτών των ποσοτήτων, αρχικά ας ορίσουμε συγκεκριμένα μεγέθη τα οποία είναι απαραίτητα για τη πλήρη κατανόηση της παρούσας εργασίας.

Ξεκινώντας από τα βασικά, θεωρούμε μια δεξαμενή θερμότητας η οποία αλληλεπιδρά με ένα σύστημα αποτελούμενο από έναν αριθμό N σωματιδίων, και το όλο σύστημα είναι απομονωμένο, δηλαδή δεν αλληλεπιδρά με το περιβάλλον.

Αναφερόμαστε στη κανονική κατανομή (συλλογή) στην οποία τα σωματίδια αλληλεπιδρούν μεταξύ τους ανταλλάζοντας ενέργεια με τη δεξαμενή θερμότητας.

Θεωρώντας δεξαμενή θερμότητας A' , και σύστημα A που είναι πολύ μικρότερο σε σχέση με το A' και τα φέρνουμε σε επαφή, προσεγγίζουμε τη κανονική συλλογή την οποία θα αναλύσουμε παρακάτω.

$$A^* = A + A'$$

$$E^* = E + E'$$

Η πιθανότητα να βρίσκεται το σύστημα A σε μια κατάσταση r με ενέργεια E_r σε ισορροπία με το A' :

$$A: E_r$$

$$A': E' = E^* - E_r$$

$$P_r = P(E_r) = \frac{1}{\Omega^*(E^*)} \Omega(E_r) \Omega'(E^* - E_r)$$

$$\sum_r P_r = 1$$

$$\text{Τότε } E_r \ll E^* \text{ και } P(E_r) = c' \Omega'(E^* - E_r)$$

$$\ln \Omega'(E^* - E_r) = \ln \Omega'(E^*) - \left. \frac{\partial \ln \Omega'(E')}{\partial E'} \right|_{E'=E^*} E_r + \dots$$

Επειδή το A' είναι δεξαμενή θερμότητας ($E_r \ll E^*$) αγνοούμε τους όρους ανώτερης τάξης στο ανάπτυγμα του $\ln \Omega$. Το $\left. \frac{\partial \ln \Omega'(E')}{\partial E'} \right|_{E'=E^*} \equiv \beta$ παίρνει την τιμή για $E'=E^*$ και άρα είναι σταθερή ποσότητα ανεξάρτητη της E_r του συστήματος A' . Το $\beta = 1/KT$ χαρακτηρίζει τη δεξαμενή θερμότητας A' . Αυτό σημαίνει ότι η T του A δεν επηρεάζεται από την ανταλλαγή ενέργειας με το A' . Άρα:

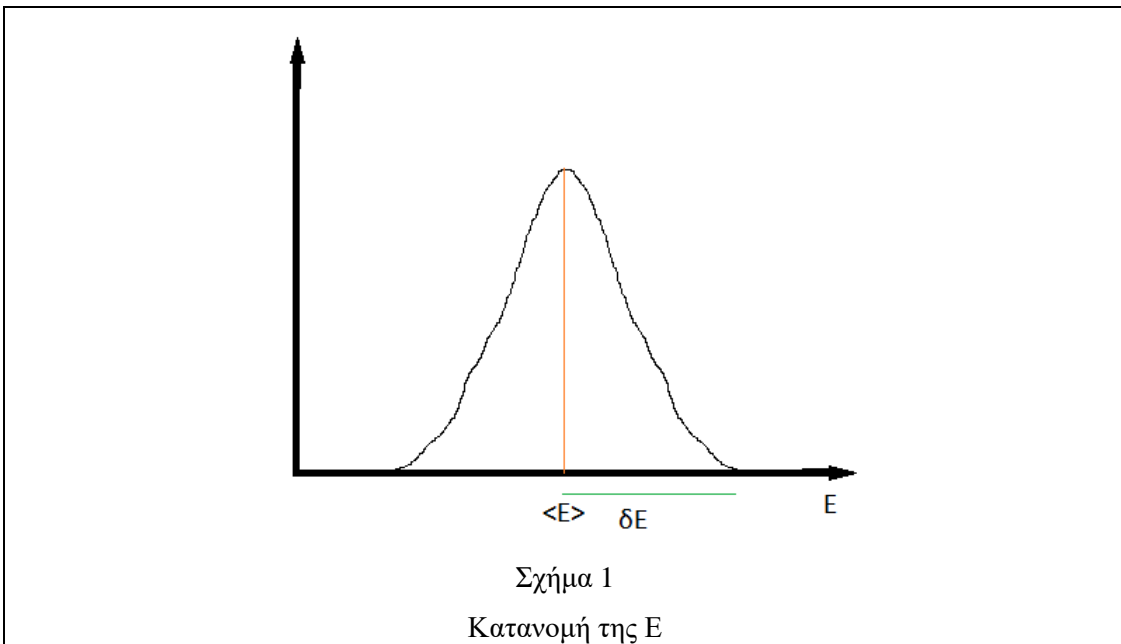
$$\ln \Omega'(E^* - E_r) = \ln \Omega'(E^*) + \beta E_r \Leftrightarrow \Omega'(E^* - E_r) = \Omega'(E^*) e^{\beta E_r}$$

Τότε $P_r = C * e^{-\beta E_r}$, όπου C ανεξάρτητη του r. $C^{-1} = \sum_r e^{-\beta E_r}$

Συνεπώς έχουμε την κανονική κατανομή $P_r = P(E_r) = \frac{e^{-\beta E_r}}{\sum_r e^{-\beta E_r}}$,

όπου $e^{-\beta E_r}$ ονομάζεται παράγοντας Boltzmann και P_r είναι κανονική κατανομή πιθανοτήτων. Ένα σύνολο συστημάτων σε επαφή με δεξαμενή θερμότητας με θερμοκρασία T ονομάζεται κανονική κατανομή. Η πιθανότητα να βρεθεί το σύστημα A σε ενέργεια μεταξύ E και E+δE δίνεται από το άθροισμα των P_r με ενέργειες μεταξύ των E και E+δE, με $P(E) = \sum_r P_r$. Όπου \sum σημαίνει άθροιση στα r με ενέργειες μεταξύ E και E+δE ($E < E_r < E + \delta E$), $P(E_r) = C \Omega(E) e^{-\beta E_r}$.

Η $\Omega(E)$ αυξάνει ραγδαία, ενώ η $e^{-\beta E_r}$ μειώνεται ραγδαία.



Όσον αφορά τις μέσες τιμές οποιαδήποτε ποσότητας y_r που μπορεί να υπάρξει στην κατάσταση r του συστήματος A:

$$\bar{y} = \sum_r P_r y_r = \frac{\sum_r y_r e^{-\beta E_r}}{\sum_r e^{-\beta E_r}}$$

Ο παρονομαστής του παραπάνω κλάσματος ονομάζεται συνάρτηση επιμερισμού Z και ισούται:

$$Z = \sum_r e^{-\beta E_r}$$

Έχουμε τα παρακάτω μεγέθη για τη μελέτη ενός θερμοδυναμικού συστήματος:

$$\langle E \rangle = \frac{\sum_r E_r e^{-\beta E_r}}{\sum_r e^{-\beta E_r}}, \quad \langle M \rangle = \frac{\sum_r M_r e^{-\beta E_r}}{\sum_r e^{-\beta E_r}},$$

$$\langle E^2 \rangle = \frac{\sum_r E_r^2 e^{-\beta E_r}}{\sum_r e^{-\beta E_r}}, \quad \langle M^2 \rangle = \frac{\sum_r M_r^2 e^{-\beta E_r}}{\sum_r e^{-\beta E_r}}.$$

Παραμαγνητισμός

Έχοντας N μαγνητικά άτομα ανά μονάδα όγκου σε εξωτερικό μαγνητικό πεδίο \vec{H} , κάθε άτομο έχει σπιν $\frac{1}{2}$ και μαγνητική ροπή μ . Το κάθε άτομο θα έχει $\vec{\mu} \parallel \vec{H}$ ή $\vec{\mu} \parallel -\vec{H}$. Θεωρούμε ότι το σύστημα A είναι το ένα άτομο και το A' (η δεξαμενή θερμότητας) είναι τα υπόλοιπα $N-1$ άτομα. Το (κάθε) άτομο μπορεί να είναι στην κατάσταση (+) $\mu \parallel \vec{H}$ ή (-) $\mu \parallel -\vec{H}$, με αντίστοιχη ενέργεια $E_+ = -\mu H$ και $E_- = +\mu H$. Η πιθανότητα το άτομο να βρίσκεται στην κατάσταση (+) $\mu \parallel \vec{H}$ είναι:

$P_+ = C e^{-\beta E_+} = C e^{\beta \mu H}$. Η πιθανότητα το άτομο να βρίσκεται στην κατάσταση (-) $\mu \parallel -\vec{H}$ είναι: $P_- = C e^{-\beta E_-} = C e^{-\beta \mu H}$.

Έχουμε δυο καταστάσεις για το σύστημα A . $P_+ + P_- = 1 \Leftrightarrow C e^{\beta \mu H} + C e^{-\beta \mu H} = 1 \Leftrightarrow C = (e^{-\beta \mu H} + e^{\beta \mu H})^{-1} = Z^{-1}$

Η συνάρτηση επιμερισμού είναι: $Z = (e^{\beta \mu H} + e^{-\beta \mu H})$

$$P_+ = \frac{e^{\beta \mu H}}{e^{\beta \mu H} + e^{-\beta \mu H}} \text{ και } P_- = \frac{e^{-\beta \mu H}}{e^{\beta \mu H} + e^{-\beta \mu H}}$$

Συνεπώς η μέση μαγνητική ροπή είναι:

$$\bar{\mu} = \frac{P_-(-\mu) + P_+(+\mu)}{P_- + P_+} = \mu \frac{e^{\beta \mu H} - e^{-\beta \mu H}}{e^{\beta \mu H} + e^{-\beta \mu H}} = \mu \tanh\left(\frac{\mu H}{kT}\right)$$

Μαγνήτιση

Η μέση μαγνητική ροπή για N άτομα ανα μονάδα όγκου: $\bar{M} = N \bar{\mu}$

Η συμπεριφορά του $\tanh\left(\frac{\mu H}{kT}\right)$

α) $\frac{\mu H}{kT} \ll 1$ (Υψηλές θερμοκρασίες)

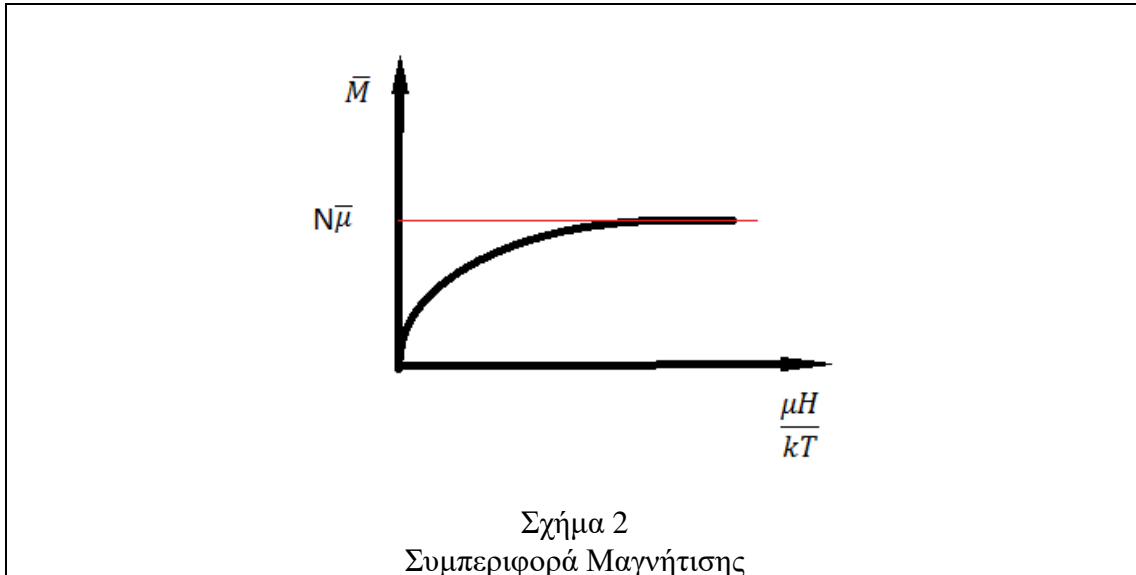
$$\tanh\left(\frac{\mu H}{kT}\right) \approx \frac{\mu H}{kT}$$

β) $\frac{\mu H}{kT} \gg 1$ (Χαμηλές θερμοκρασίες)

$$\tanh\left(\frac{\mu H}{kT}\right) \approx 1$$

Αρα

$$\bar{\mu} = \mu \tanh\left(\frac{\mu H}{kT}\right) = \begin{cases} \mu, & kT \ll \mu H \text{ (χαμηλές θερμοκρασίες)} \\ \frac{\mu^2 H}{kT}, & kT \gg \mu H \text{ (υψηλές θερμοκρασίες)} \end{cases}$$



Η μέση ενέργεια για το σύστημα A είναι:

$$\bar{E}_1 = \frac{(-\mu H)e^{\beta\mu H} - (\mu H)e^{-\beta\mu H}}{e^{\beta\mu H} + e^{-\beta\mu H}} = -\mu H \tanh\left(\frac{\mu H}{kT}\right)$$

Η μέση ενέργεια για N άτομα είναι:

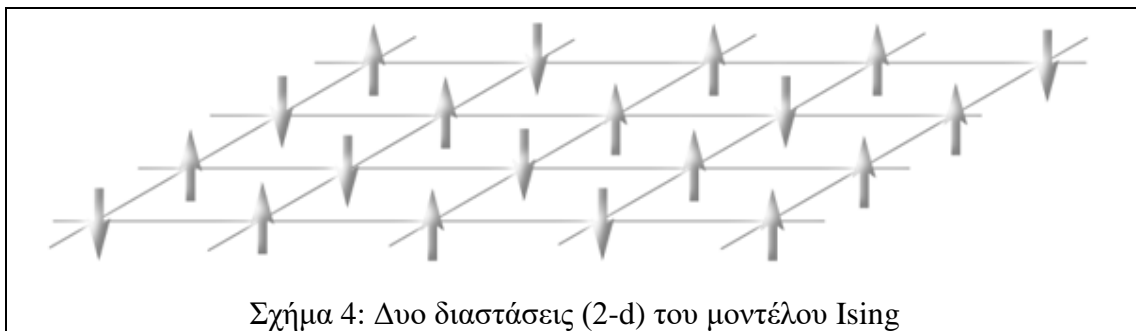
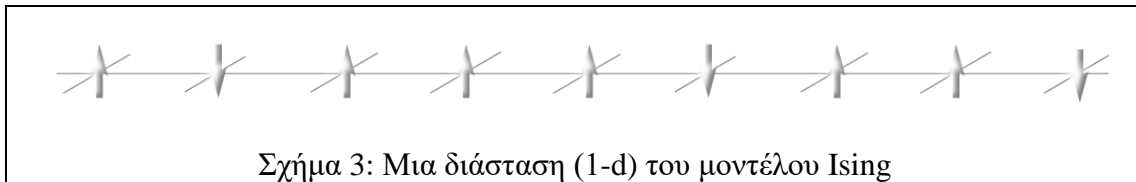
$$\bar{E} = N\bar{E}_1 = -N\mu H \tanh\left(\frac{\mu H}{kT}\right)$$

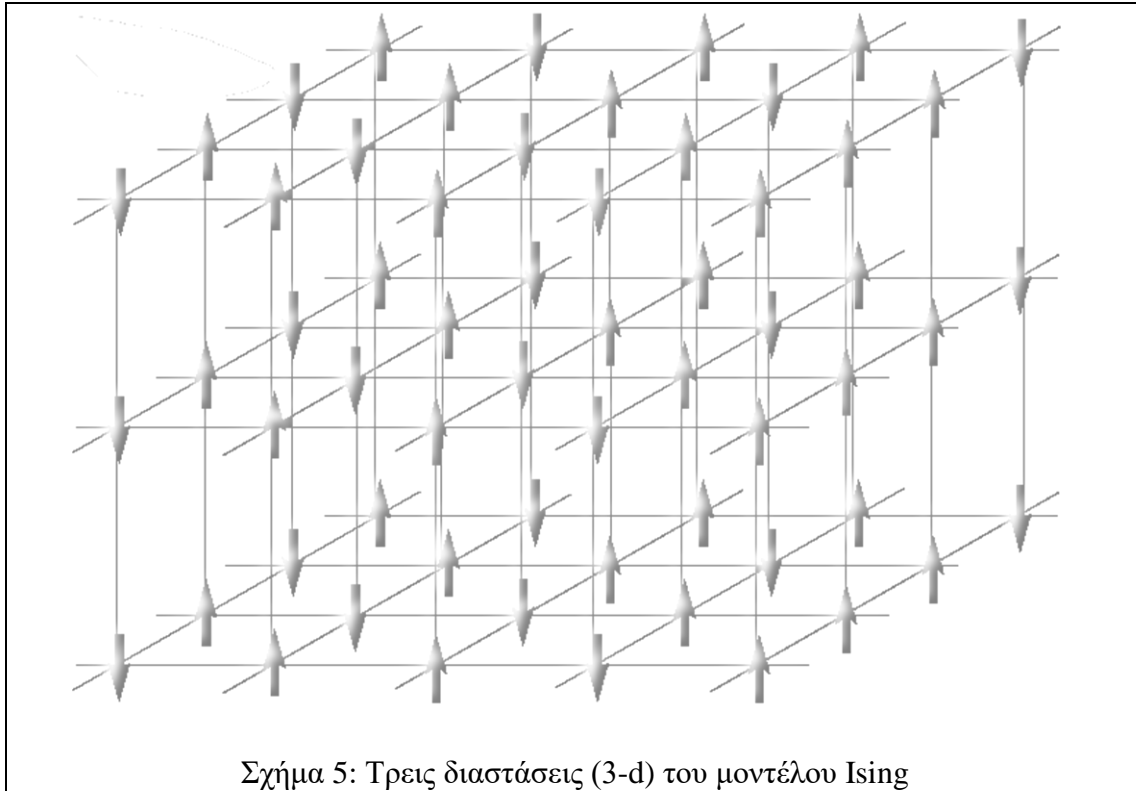
Καθώς $T \rightarrow \infty, P \rightarrow 0, P(E_1) = \frac{1}{2}, P(E_2) = \frac{1}{2}$

Καθώς $T \rightarrow 0, P \rightarrow \infty, P(E_1) = 1, P(E_2) = 0$

Κεφάλαιο 2^ο

Το μοντέλο Ising αναφέρεται σε ένα σύστημα σωματιδίων, τα οποία μπορούν να βρίσκονται στο μονοδιάστατο (1-D), δισδιάστατο (2-D) ή και τρισδιάστατο (3-D) χώρο, με σπιν είτε «+1» ή «-1», έχοντας μια ιδιαιτερότητα. Η χαμιλτονιανή που εκφράζει την ενέργεια του συστήματος εξαρτάται μόνο από την αλληλεπίδραση των σπιν των σωματιδίων των πρώτων γειτόνων και μόνο και από τη συνεισφορά του εξωτερικού μαγνητικού πεδίου. Παρατίθεται ένα πλέγμα σε μια διάσταση, ένα πλέγμα σε δυο διαστάσεις που απεικονίζει τους διάφορους προσανατολισμούς που θα μπορούσε ένα σύστημα να έχει, ενώ μια προσπάθεια απεικόνισης σε τρεις διαστάσεις υπάρχει στη συνέχεια.





Σχήμα 5: Τρεις διαστάσεις (3-d) του μοντέλου Ising

Η χαμιλτονιανή που εκφράζει το μοντέλο Ising είναι:

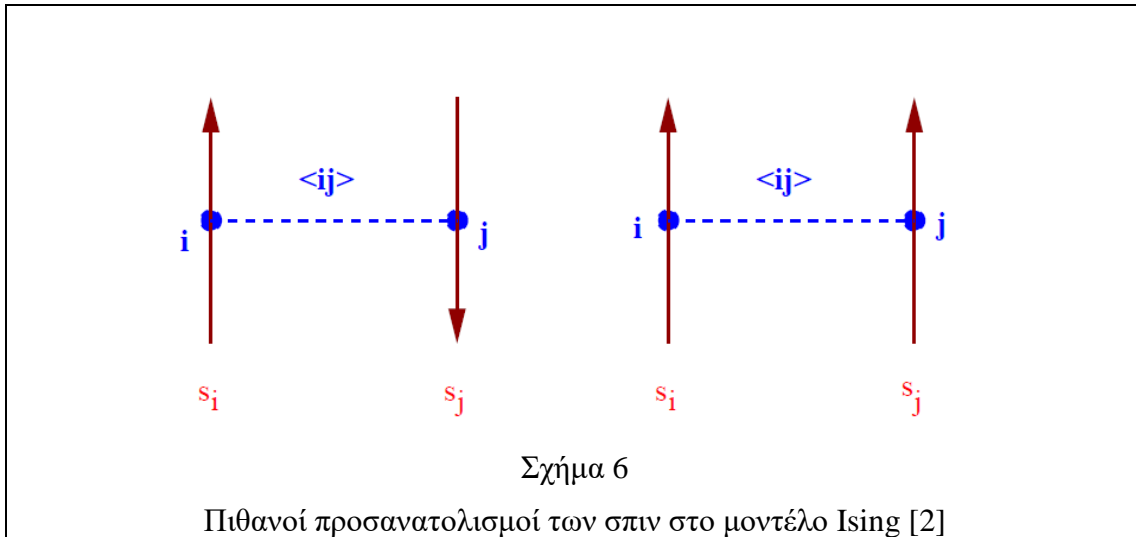
$$H = -J \sum_{\langle ij \rangle} s_i s_j - B \sum_i s_i$$

Ο πρώτος όρος της Χαμιλτονιανής είναι η αλληλεπίδραση σπιν – σπιν και ο δεύτερος όρος είναι η αλληλεπίδραση του μαγνητικού πεδίου. Για $J > 0$ το σύστημα είναι σιδηρομαγνητικό και μόνο αυτή τη περίπτωση μελετάμε.

Η συνάρτηση επιμερισμού Z είναι:

$$Z = \sum_{s_1 = \pm 1} \dots \sum_{s_2 = \pm 1} \dots \sum_{s_N = \pm 1} e^{-\beta H[\{s_i\}]}$$

$$Z = \sum_{\{s_i\}} e^{\beta J \sum_{\langle ij \rangle} s_i s_j + \beta \sum_i s_i}$$



Παραπάνω φαίνονται οι πιθανοί προσανατολισμοί των σπιν ανάμεσα σε δυο σωματίδια. Η μελέτη του συστήματος στη παρούσα εργασία γίνεται για μηδενικό μαγνητικό πεδίο $\vec{B} = 0$.

Η λύση του μοντέλου Ising σε δυο διαστάσεις για την αλλαγή φάσης, σύμφωνα με τη βιβλιογραφία [4], έχει λύση και είναι η θερμοκρασία $\beta_c = \frac{\ln(1+\sqrt{2})}{2}$.

Κεφάλαιο 3ο

Η μέθοδος Monte Carlo, είναι μια μέθοδος προσομοίωσης ενός στοχαστικού συστήματος. Η βασική ιδέα στη προσομοίωση με τη μέθοδο Monte Carlo είναι να μεταβούμε τυχαία από μια κατάσταση των σπιν σε μια άλλη. Παράγουμε ένα αντιπροσωπευτικό δείγμα κατασκευάζοντας μια αλυσίδα Markov, ώστε τα δεδομένα να υπακούουν σε στατιστική προσέγγιση της επιθυμητής κατανομής πιθανότητας, στην περίπτωση μας $p_\mu = \frac{1}{Z} e^{-\frac{E_\mu}{kT}}$.

Για την υλοποίηση των παραπάνω, είναι απαραίτητη η επιλογή ενός «κανόνα επιλογής» μετάβασης από τη μια κατάσταση στην επόμενη για την προσομοίωση Monte Carlo, έτσι ώστε η πιθανότητα κατανομής στο δείγμα να είναι ίδια με τη κατανομή του δείγματος, στα όρια μεγάλου δείγματος. Η βασική ιδέα είναι να προσομοιώσουμε τις φυσικές διαδικασίες που ανταποκρίνονται στην εξίσωση

$$\frac{dw_\mu}{dt} = \sum_\nu [w_\nu(t)R(\nu \rightarrow \mu) - w_\mu R(\mu \rightarrow \nu)] = 0$$

όπου w_μ είναι το στατιστικό βάρος που μας δίνει την πιθανότητα να είναι το σύστημά μας στην κατάσταση μ , και συνδέει τη μικροσκοπική περιγραφή με τη στατιστική περιγραφή του συστήματος.

Επιλέγουμε ένα σύνολο ρυθμών αλλαγής καταστάσεων από μ σε ν $R(\mu \rightarrow \nu)$ για μεταβάσεις από τη μια κατάσταση στην άλλη, και τις επιλέγουμε με τέτοιο τρόπο, ώστε οι λύσεις της παραπάνω εξίσωσης να υπακούουν στη κατανομή του Boltzmann.

$$p_\mu = \frac{1}{Z} e^{-\frac{E_\mu}{kT}}$$

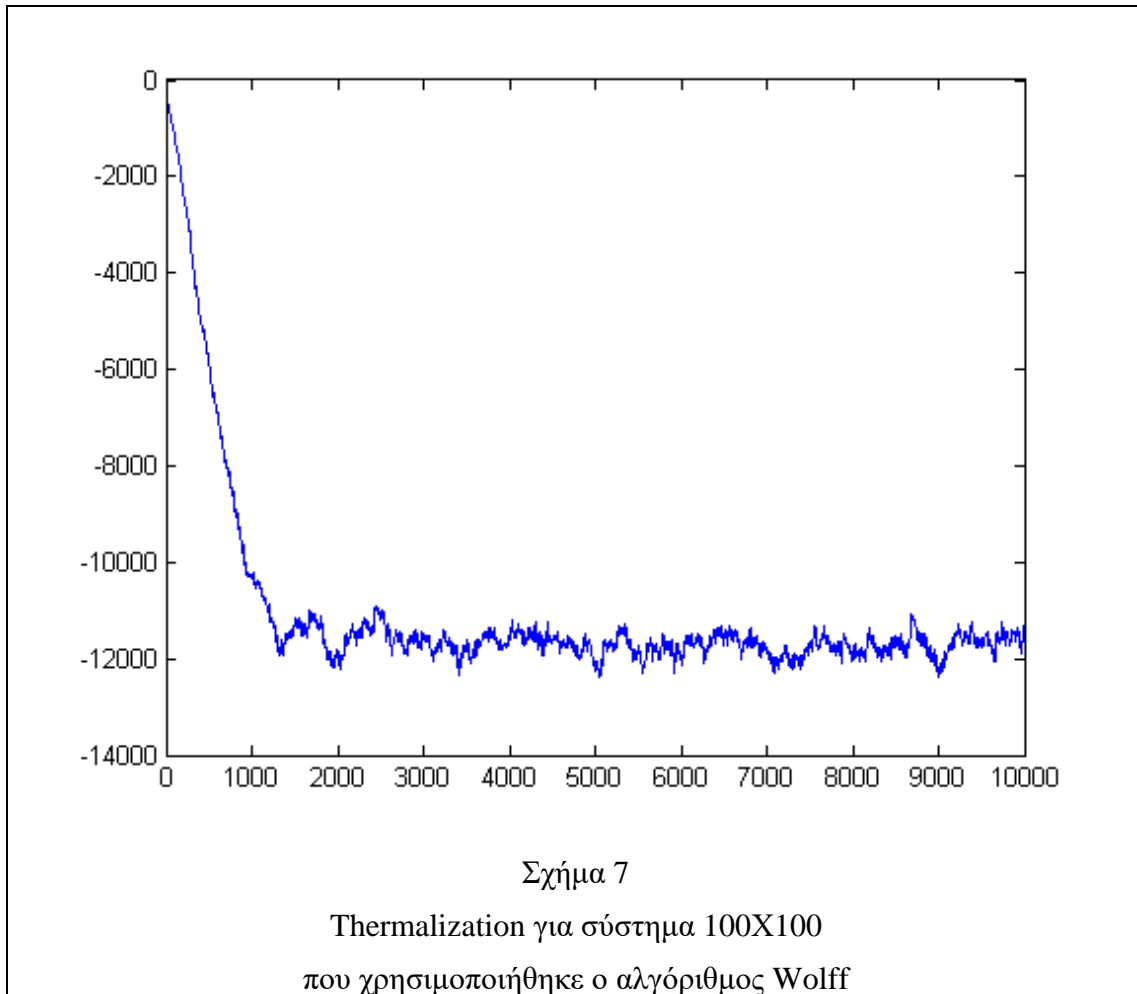
Με τη παραπάνω διαδικασία, επιλέγονται από τον υπολογιστή εκείνες οι καταστάσεις που υπακούουν στα παραπάνω, και υπολογίζονται κάθε φορά οι τιμές που μας ενδιαφέρουν για το σύστημα.

Στην παρούσα μελέτη, η παραγωγή των τυχαίων καταστάσεων μας επιτρέπει να προσομοιώνουμε τη θερμοδυναμική συμπεριφορά κάθε φορά σε τετράγωνο πλέγμα $L \times L = N$ μέσω των αλγορίθμων Metropolis και Wolff.

Ο πιο απλός αλγόριθμος που μπορεί να προσομοιώσει το μοντέλο Ising είναι ο Metropolis, ενώ για τη παρούσα μελέτη χρησιμοποιήθηκε ο αλγόριθμος Wolff όπου υλοποιήθηκε με κώδικα C.

Η διαφορά των δυο αλγορίθμων είναι πως ο Metropolis για τη μετάβαση από μια κατάσταση των σπιν σε μια άλλη αλλάζει κάθε φορά ένα μόνο σημείο του πλέγματος από σπιν « s_i » σε « $-s_i$ », ενώ ο αλγόριθμος Wolff, ο οποίος είναι cluster αλγόριθμος, αυτό που κάνει είναι να κατασκευάζει clusters στοχαστικού μεγέθους (μεταβαλλόμενου) μέσα στο πλέγμα, δηλαδή μια συστοιχία από θέσεις στις οποίες αντιστρέφει τις τιμές όλων των σπιν.

Για να δημιουργήσουμε μια σειρά από ανεξάρτητες μετρήσεις χρησιμοποιώντας μια αλυσίδα Markov είναι απαραίτητο να μην υπάρχει αυτοσυσχέτιση των μετρήσεων του δείγματος. Για τον αλγόριθμο Metropolis αυτό γίνεται πολύ αργά, μιας και οι επόμενες καταστάσεις διαφέρουν από τις προηγούμενες κατά το πολύ μια διαφοροποίηση μιας κατάστασης ενός σπιν. Για το λόγο αυτό ο αλγόριθμος Wolff υπερτερεί σημαντικά. Από την άλλη, από την αρχή της διαδικασίας προσομοίωσης χρειάζεται μια κατάσταση από την οποία θα ξεκινήσει ο αλγόριθμος. Ο αλγόριθμος Metropolis υστερεί κατά πολύ έναντι του Wolff να φέρει το σύστημα σε θερμική ισορροπία. Η διαδικασία αυτή αναφέρεται ως Thermalization.



Για τη κρίσιμη θερμοκρασία ενδεικτικά φαίνεται στο παραπάνω σχήμα πως για να επέλθει το σύστημα σε ενέργειες που αντιστοιχούν στο σύστημα μας για δεδομένη θεοκρασία και μέγεθος πλέγματος, χρειάζονται τουλάχιστον 1000 τρεξίματα του αλγορίθμου. Για το αντίστοιχο σύστημα με τον αλγόριθμο Metropolis, δεδομένου πως αλλάζει μόνο μια κατάσταση ενός σπιν, και όχι σπιν clusters όπως ο Wolff, διαφαίνεται πως πρακτικά μας διευκολύνει ο Wolff. Ο Wolff υπερτερεί του Metropolis μόνο στη κρίσιμη περιοχή. Για $\beta \ll \beta_c$ και $\beta \gg \beta_c$ οι δυο αλγόριθμοι είναι συγκρίσιμης απόδοσης.

Κεφάλαιο 4^ο

Η μέθοδος Single – Histogram.

Η μέθοδος αυτή είναι μια τεχνική, η οποία μας επιτρέπει αφού κάνουμε μια προσομοίωση Monte Carlo σε συγκεκριμένη θερμοκρασία και για δεδομένο μέγεθος πλέγματος L, να υπολογίσουμε άμεσα ποσότητες, όπως η εσωτερική ενέργεια U, σε θερμοκρασίες διαφορετικές της προσομοιωμένης, σε ένα εύρος θερμοκρασιών κοντά της αρχικής.

Ξεκινώντας από τη κανονική κατανομή για μια ποσότητα Q, έχουμε:

$$\bar{Q} = \sum_r P_r Q_r = \frac{\sum_r Q_r e^{-\beta E_r}}{\sum_r e^{-\beta E_r}}$$

Ο παρονομαστής του παραπάνω κλάσματος ονομάζεται συνάρτηση επιμερισμού Z και ισούται:

$$Z = \sum_r e^{-\beta E_r}$$

Μια εκτίμηση για την παρατηρούμενη ποσότητα στην κατάσταση M του συστήματος είναι:

$$Q_M = \frac{\sum_i Q_{\mu_i} \rho_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}{\sum_j \rho_{\mu_j}^{-1} e^{-\beta E_{\mu_j}}}$$

Όπου ρ_{μ_i} είναι η πιθανότητα δειγματοληψίας της ποσότητας Q.

Την πιθανότητα δειγματοληψίας της i-οστής κατάστασης την παίρνουμε να είναι:

$$\rho_{\mu_i} = \frac{1}{Z_0} e^{-\beta E_{\mu_i}}$$

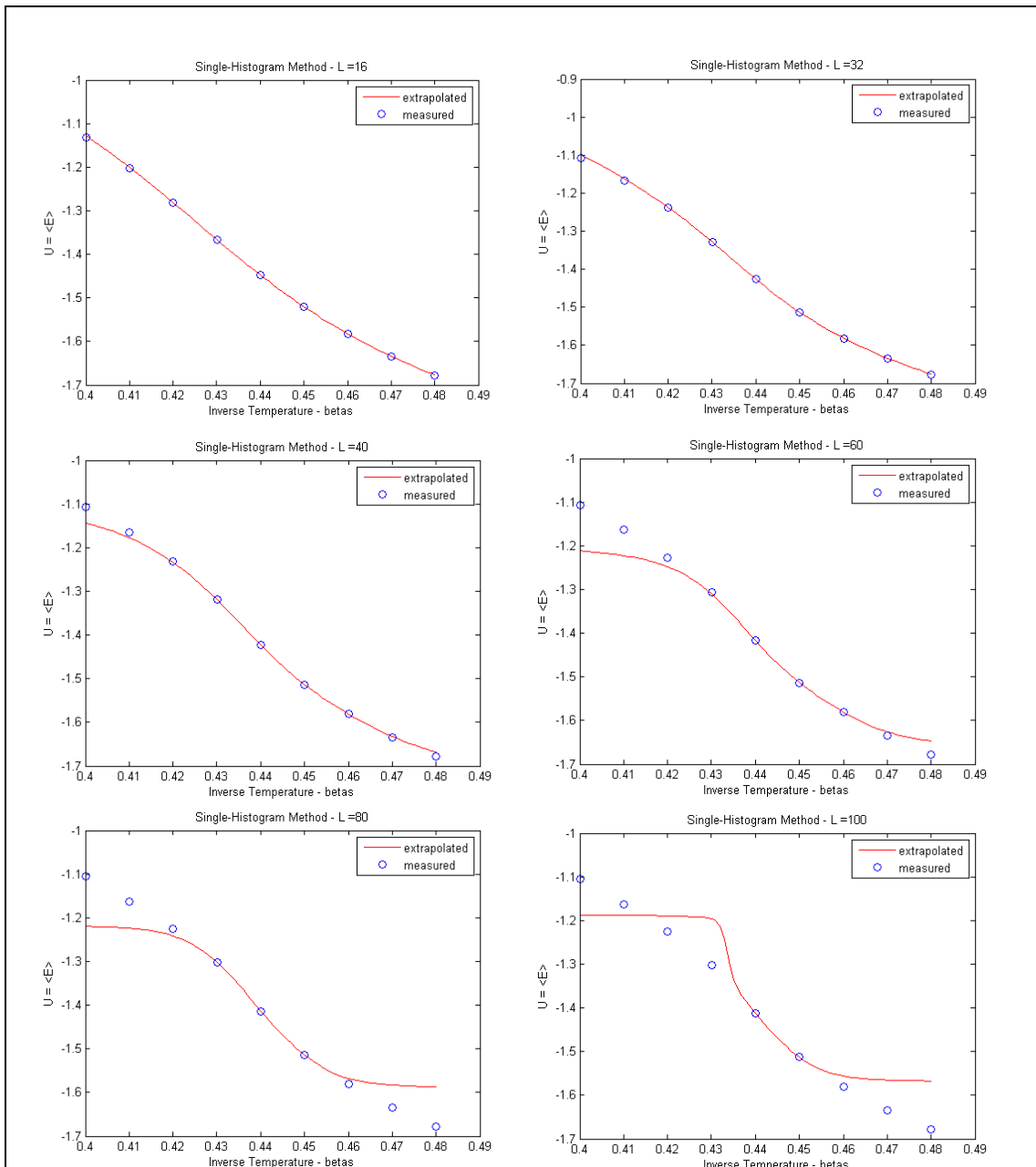
Άρα:

$$Q_M = \frac{\sum_i Q_{\mu_i} \left(\frac{1}{Z_0} e^{-\beta_0 E_{\mu_i}}\right)^{-1} e^{-\beta E_{\mu_i}}}{\sum_j \left(\frac{1}{Z_0} e^{-\beta_0 E_{\mu_j}}\right)^{-1} e^{-\beta E_{\mu_j}}} \Leftrightarrow Q_M = \frac{\sum_i Q_{\mu_i} e^{(\beta_0 - \beta) E_{\mu_i}}}{\sum_j e^{(\beta_0 - \beta) E_{\mu_j}}}$$

Για τον υπολογισμό της εσωτερικής ενέργειας U έχουμε:

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

Όπου \sum_E είναι το άθροισμα σε όλο το φάσμα των ενεργειών και N(E) η συχνότητα της κάθε ενέργειας E.



Εφαρμογή μεθόδου Single Histogram για θερμοκρασία $\beta_c = 0.44$.

Measured – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας

Extrapolated – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Single – Histogram

Για $L = 40 - 100$ παρατηρούμε απόκλιση των τιμών που υπολογίζονται από τη μέθοδο Single Histogram και αυτών που υπολογίζονται με απευθείας προσομοίωση Monte Carlo

Με παρόμοιο τρόπο υπολογίζονται και άλλες ποσότητες, όπως η μέση μαγνήτιση καθώς και οι μέσες τιμές των τετραγώνων τους.

Στα παραπάνω σχήματα μπορούμε ενδεικτικά να καταλάβουμε την απόκλιση της εφαρμογής της μεθόδου καθώς εφαρμόζεται σε μεγαλύτερα πλέγματα. Από $L = 40$ και έπειτα ξεκάθαρα οι ακραίες τιμές αποκλίνουν. Μπορούμε να καταλάβουμε πως εφαρμόζοντας τη μέθοδο, η καμπύλη αποκλίνει αυξανόμενου του L καθώς και του εύρους θερμοκρασιών υπολογισμού. Αυτό συμβαίνει διότι η κατανομή πιθανοτήτων για μια συγκεκριμένη T θερμοκρασία είναι διαφορετική για διαφορετική θερμοκρασία T' . Όσο αυξάνει το μέγεθος του πλέγματος, τόσο μειώνεται το εύρος της περιοχής που προσομοιώνουμε όπου η κατανομή πιθανοτήτων είναι παρόμοια με τη μετρούμενη. Πιο συγκεκριμένα, τα ιστογράμματα έχουν εύρος που μειώνεται αυξανόμενου του μεγέθους πλέγματος L , και για το λόγο αυτό μειώνεται και το εύρος των θερμοκρασιών που η μέθοδος Single Histogram δεν επηρεάζεται σημαντικά από το θόρυβο στα άκρα των ιστογραμμάτων.

Κεφάλαιο 5^ο

Η μέθοδος Multi Histogram.

Η μέθοδος Single Histogram είναι αρκετά χρηστική αλλά μας περιορίζει σε ένα σχετικά μικρό εύρος θερμοκρασιών κοντά στην θερμοκρασία που προσομοιώνουμε, για τον υπολογισμό σε διαφορετικές θερμοκρασίες.

Για τον παραπάνω λόγο, η μέθοδος Multi Histogram είναι προέκταση της εφαρμογής της Single Histogram, η οποία μας επιτρέπει μέσω των προσομοιώσεων Monte Carlo σε ένα εύρος θερμοκρασιών, να μπορούμε να υπολογίζουμε με ακρίβεια ποσότητες για συγκεκριμένες θερμοκρασίες εντός της περιοχής θερμοκρασιών όπου έγιναν οι προσομοιώσεις. Αναλυτικά τα βήματα φαίνονται παρακάτω.

Για μια προσομοίωση Monte Carlo η πιθανότητα της κάθε ενέργειας είναι: $P(E) = \rho(E) \frac{e^{-\beta E}}{Z}$, με $\rho(E)$ η πυκνότητα των καταστάσεων με ενέργεια E και Z η συνάρτηση επιμερισμού για τη συγκεκριμένη θερμοκρασία β . Μέσω ενός ιστογράμματος ενεργειών για μια προσομοίωση Monte Carlo, η πιθανότητα είναι $P(E) = \frac{N(E)}{n}$, με $N(E)$ να είναι η συχνότητα εμφάνισης της κάθε ενέργειας E , από n φορές που πραγματοποιήθηκε η προσομοίωση. Χρησιμοποιώντας τις δυο παραπάνω σχέσεις έχουμε:

$\frac{N(E)}{n} = \rho(E) \frac{e^{-\beta E}}{Z} \Leftrightarrow \rho(E) = \frac{N(E)}{n} \frac{Z}{e^{-\beta E}}$. Για i αντίστροφες θερμοκρασίες β_i , οι πυκνότητες των καταστάσεων είναι: $\rho_i(E) = \frac{N_i(E)}{n_i} \frac{Z_i}{e^{-\beta_i E}}$.

Η πυκνότητα των καταστάσεων είναι δεδομένη, δεν εξαρτάται από τη θερμοκρασία, αλλά προσπαθούμε να εξομαλύνουμε τις αποκλίσεις χρησιμοποιώντας την παρακάτω προσέγγιση.

Ο κύριος τρόπος υπολογισμού μιας τέτοιας ποσότητας είναι: $\bar{x} = \frac{\sum_i \frac{x_i}{\sigma_i^2}}{\sum_j \frac{1}{\sigma_j^2}}$.

Η διακύμανση σ εξαρτάται της συχνότητας εμφάνισης $N_i(E)^{-1}$.

Άρα $\rho(E) = \frac{\overline{N(E)}}{n_i} \frac{Z_i}{e^{-\beta_i E}}$. Το σφάλμα είναι $\sigma_i = \frac{\Delta N_i(E)}{n_i} \frac{Z_i}{e^{-\beta_i E}}$ με το $\Delta N_i(E)$ να είναι η εκτίμηση των σφαλμάτων σε κάθε ένα από τα ιστογράμματα. Οι διαφορές προκύπτουν από τα στατιστικά σφάλματα στα ιστογράμματα.

Θεωρώντας πως οι μετρήσεις της ενέργειας κατά τη διαδικασία προσομοίωσης είναι ανεξάρτητες μεταξύ τους, το σφάλμα $\Delta N_i(E)$ του αριθμού των δειγμάτων σε κάθε ιστόγραμμα υπακούει στη κατανομή Poisson: $\Delta N_i(E) = \sqrt{N_i(E)}$. Το πραγματικό σφάλμα $\Delta N_i(E)$ είναι συνάρτηση της μέσης τιμής των ιστογραμμάτων $\overline{N_i(E)}$ για συγκεκριμένη θερμοκρασία, συνεπώς μπορούμε να θεωρήσουμε ότι: $\Delta N_i(E) = \sqrt{N_i(E)} = \sqrt{\overline{N_i(E)}}$

$$\text{Άρα } \sigma_i = \frac{\sqrt{N_i(E)}}{n_i} \frac{Z_i}{e^{-\beta_i E}} \Leftrightarrow \sigma_i^2 = \frac{N_i(E)}{n_i^2} \left[\frac{Z_i}{e^{-\beta_i E}} \right]^2. \text{ Όμως } \rho(E) = \frac{N(E)}{n_i} \frac{Z_i}{e^{-\beta_i E}} \Leftrightarrow \rho^2(E) = \frac{N(E)^2}{n_i^2} \frac{Z_i^2}{e^{-2\beta_i E}}. \text{ Επομένως } \sigma_i^2 = \frac{\rho^2(E)}{N_i(E)}$$

Χρησιμοποιώντας τον τύπο για τον μέσο με τα βάρη έχουμε:

$$\rho(E) = \frac{\Sigma_i \frac{\rho_i(E)}{\sigma_i^2}}{\Sigma_j \frac{1}{\sigma_j^2}} = \frac{\Sigma_i \frac{\frac{N_i(E)}{n_i} \frac{Z_i}{e^{-\beta_i E}}}{\frac{\rho^2(E)}{N_i(E)}}}{\Sigma_j \frac{1}{\frac{\rho^2(E)}{N_j(E)}}} = \frac{\Sigma_i \frac{N_i^2(E)}{n_i} \frac{Z_i}{e^{-\beta_i E}} \frac{1}{\rho^2(E)}}{\Sigma_j \frac{N_j(E)}{\rho^2(E)}} = \frac{\Sigma_i \frac{N_i^2(E)}{n_i} \frac{Z_i}{e^{-\beta_i E}}}{\Sigma_j N_j(E)} = \Sigma_i \frac{N_i(E)}{n_i Z_i^{-1} e^{-\beta_i E}} \Leftrightarrow$$

$$\boxed{\rho(E) = \Sigma_i \frac{N_i(E)}{n_i Z_i^{-1} e^{-\beta_i E}}}$$

Καταλήγουμε σε μια έκφραση της πυκνότητας των καταστάσεων όπου περιέχει τις συναρτήσεις επιμερισμού Z για κάθε β . Ο υπολογισμός της συνάρτησης επιμερισμού Z γίνεται επαναληπτικά χρησιμοποιώντας τον ορισμό της συνάρτησης επιμερισμού:

$$Z_k = \Sigma_E \rho(E) e^{-\beta_k E} \Leftrightarrow Z_k = \Sigma_E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i Z_i^{-1} e^{-\beta_i E}} e^{-\beta_k E} \Leftrightarrow$$

$$Z_k = \Sigma_E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i Z_i^{-1} e^{(\beta_k - \beta_i) E}}$$

Στις παραπάνω σχέσεις θέτουμε αρχική $Z_k = 1$ και η επαναληπτική διαδικασία συγκλίνει στις τιμές $Z_k = Z(\beta_k)$. Στην πράξη σταματάμε τις επαναλήψεις όταν οι μεταβολές $|Z_k^{(MH)} - Z_k^{(n)}| < \varepsilon$, για επιλεγμένο μικρό ε . Τότε η συνάρτηση επιμερισμού δίνεται από τη σχέση:

$$Z(\beta) = \Sigma_E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i Z_i^{-1} e^{(\beta - \beta_i) E}}.$$

Μέσω αυτής μπορούμε να υπολογίσουμε διάφορες ποσότητες, όπως η εσωτερική ενέργεια U :

$$U(\beta) = \Sigma_E E \rho(E) \frac{e^{-\beta E}}{Z(\beta)} \Leftrightarrow U(\beta) = \frac{1}{Z(\beta)} \Sigma_E E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i Z_i^{-1} e^{(\beta - \beta_i) E}}.$$

Οι τιμές της ενέργειας E που χρησιμοποιούνται είναι οι συνολικές ενέργειες του συστήματος, και όχι οι ενέργειες ανά ζεύγους spin. Κατά τη διάρκεια υπολογισμού των συναρτήσεων επιμερισμού Z_k , οι τιμές που μπορούν να παίρνουν είναι πολύ μεγάλες ή πολύ μικρές, είναι πιο πρακτικό να χρησιμοποιήσουμε τις ελεύθερες ενέργειες που δίνονται από τα εκθετικά: $Z_k = e^{-\beta_k F_k}$. Συνεπώς έχουμε:

$$Z_k = \Sigma_E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i Z_i^{-1} e^{(\beta_k - \beta_i) E}} \Leftrightarrow Z_k = \Sigma_E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i (e^{-\beta_j F_j})^{-1} e^{(\beta_k - \beta_i) E}} \Leftrightarrow$$

$$Z_k = \Sigma_E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i e^{\beta_j F_j} e^{(\beta_k - \beta_i) E}} \Leftrightarrow e^{-b_k F_k} = \Sigma_E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i e^{\beta_k E - \beta_i E + \beta_j F_j}} \Leftrightarrow$$

$$\boxed{F_k = \frac{-\ln(\Sigma_E \frac{\Sigma_i N_i(E)}{\Sigma_j n_i e^{\beta_k E - \beta_i E + \beta_j F_j}})}{\beta_k}}$$

$$U_k = \frac{1}{Z_k} \sum_E \frac{E \sum_i N_i(E)}{\sum_j n_i (e^{-\beta_j F_j})^{-1} e^{(\beta_k - \beta_i)E}} \Leftrightarrow \boxed{U_k = \frac{1}{Z_k} \sum_E \frac{E \sum_i N_i(E)}{\sum_j n_i e^{\beta_k E - \beta_i E + \beta_j F_j}}$$

Αντίστοιχα για τον υπολογισμό μιας οποιαδήποτε ποσότητας $\langle f(M) \rangle$ εργαζόμαστε όπως φαίνεται παρακάτω.

Για δεδομένη θερμοκρασία με συγκεκριμένο β , κατ' αντιστοιχία έχουμε πως:

$$h(E, \beta) = \frac{N(E)}{\sum_E N(E)} \quad \rightarrow \quad h(E, M, \beta) = \frac{N(E, M)}{\sum_{E, M} N(E, M)}$$

όπου $h(E, M, \beta)$ η πιθανότητα μέτρησης ενέργειας E όταν η μαγνητική ροπή είναι M .

Άρα η πιθανότητα εμφάνισης είναι:

$$P(E, M, \beta) = \frac{1}{Z(\beta)} p(E, M) e^{\beta E} \quad \rightarrow \quad P(E, M) = Z(\beta) P(E, M, \beta) e^{\beta E}$$

Άρα μπορούμε να γράψουμε:

$$P(E, M) = Z(\beta_k) P(E, M, \beta) e^{\beta_k E} \cong Z_k h(E, M, \beta_k) e^{\beta_k E} \text{ και}$$

$$Z(\beta) = \sum_{E, M} p(E, M) e^{-\beta E} \cong Z_k \sum_{E, M} h(E, M, \beta) e^{-(\beta - \beta_k)E}$$

Για τη διευκόλυνση των αριθμητικών υπολογισμών θέτουμε $Z_k = e^{-F_k}$ και οι παραπάνω σχέσεις μετατρέπονται ως εξής:

$$Z(\beta) = e^{-F_k} \sum_{E, M} h(E, M, \beta) e^{-(\beta - \beta_k)E} \Leftrightarrow$$

$$Z(\beta) = \sum_{E, M} h(E, M, \beta) e^{-(\beta - \beta_k)E - F_k}$$

Άρα:

$$P(E, M, \beta) = \frac{h(E, M, \beta) e^{-(\beta - \beta_k)E - F_k}}{\sum_{E, M} h(E, M, \beta) e^{-(\beta - \beta_k)E - F_k}}$$

Συνεπώς οποιαδήποτε τιμή μπορεί να υπολογιστεί:

$$\langle F(M) \rangle (\beta) = \sum_{E, M} P(E, M, \beta) F(M) \Leftrightarrow$$

$$\boxed{\langle F(M) \rangle (\beta) = \frac{\sum_{E, M} f(M) h(E, M, \beta_k) e^{-(\beta - \beta_k)E - F_k}}{\sum_{E, M} h(E, M, \beta_k) e^{-(\beta - \beta_k)E - F_k}}}$$

Όπου $F(M)$ μπορεί να είναι οποιαδήποτε ποσότητα συνάρτηση του M .

Όπως και για τον υπολογισμό της μέσης ενέργειας παραπάνω, χρησιμοποιώντας τον τύπο για τον μέσο με τα βάρη έχουμε:

$$\begin{aligned} \rho(E, M) &= \frac{\sum_i \frac{\rho_i(E, M)}{\sigma_i^2}}{\sum_j \frac{1}{\sigma_j^2}} = \frac{\sum_i \frac{\frac{N_i(E, M)}{n_i} \frac{Z_i}{e^{-\beta_i E}}}{\rho^2(E, M)}}{\sum_j \frac{1}{\rho^2(E, M) \frac{N_j(E, M)}{N_j(E, M)}}} \\ &= \frac{\sum_i \frac{N_i^2(E, M)}{n_i} \frac{Z_i}{e^{-\beta_i E}} \frac{1}{\rho^2(E, M)}}{\sum_j \frac{N_j(E, M)}{\rho^2(E, M)}} = \frac{\sum_i \frac{N_i^2(E, M)}{n_i} \frac{Z_i}{e^{-\beta_i E}}}{\sum_j N_j(E, M)} = \sum_i \frac{N_i(E, M)}{n_i Z_i^{-1} e^{-\beta_i E}} \end{aligned}$$

Προχωρώντας περαιτέρω έχουμε:

$$\begin{aligned} Z_\kappa &= \sum_{E, M} P(E, M) e^{-\beta_\kappa E} = \sum_{E, M} \frac{\sum_i N_i(E, M)}{\sum_j n_j Z_j^{-1} e^{(\beta - \beta_j) E}} \Leftrightarrow \\ e^{-F_\kappa} &= \sum_{E, M} \frac{\sum_i N_i(E, M)}{\sum_j n_j Z_j^{-1} e^{(\beta - \beta_j) E}} \end{aligned}$$

Αντί για το Z_κ υπολογίζω το F_κ .

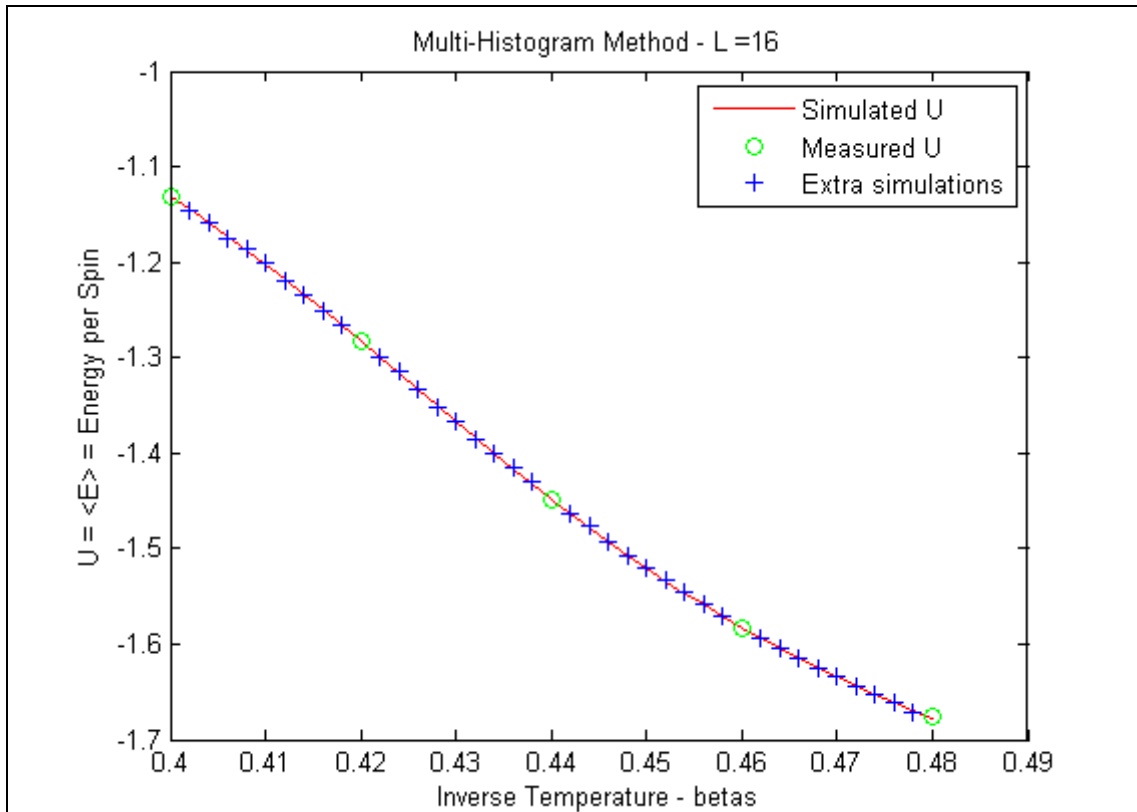
Ακριβώς όπως παραπάνω η διαδικασία υπολογίζει οποιαδήποτε ποσότητα έπειτα.

$$\langle F(M) \rangle (\beta) = \frac{1}{Z(\beta)} \sum_{E, M} F(M) \frac{\sum_i N_i(E, M)}{\sum_j n_j e^{(\beta - \beta_j) E + F_j}}$$

$$\langle M(\beta) \rangle = \frac{1}{Z(\beta)} \sum_{E, M} M \frac{\sum_i N_i(E, M)}{\sum_j n_j e^{(\beta - \beta_j) E + F_j}}$$

$$\langle M^2(\beta) \rangle = \frac{1}{Z(\beta)} \sum_{E, M} M^2 \frac{\sum_i N_i(E, M)}{\sum_j n_j e^{(\beta - \beta_j) E + F_j}}$$

Τα παρακάτω διαγράμματα δείχνουν την αξιοπιστία της μεθόδου, καθώς για πλέγματα $L=16,32,40,60$ και 80 με 10^6 Wolff Clusters, φαίνεται πως η καμπύλη που δημιουργήθηκε με τη μέθοδο Multi – Histogram περιέχει σημεία που υπολόγισε για συγκεκριμένα β , των οποίων οι προσομοιώσεις έρχονται σε συμφωνία αποτελεσμάτων.



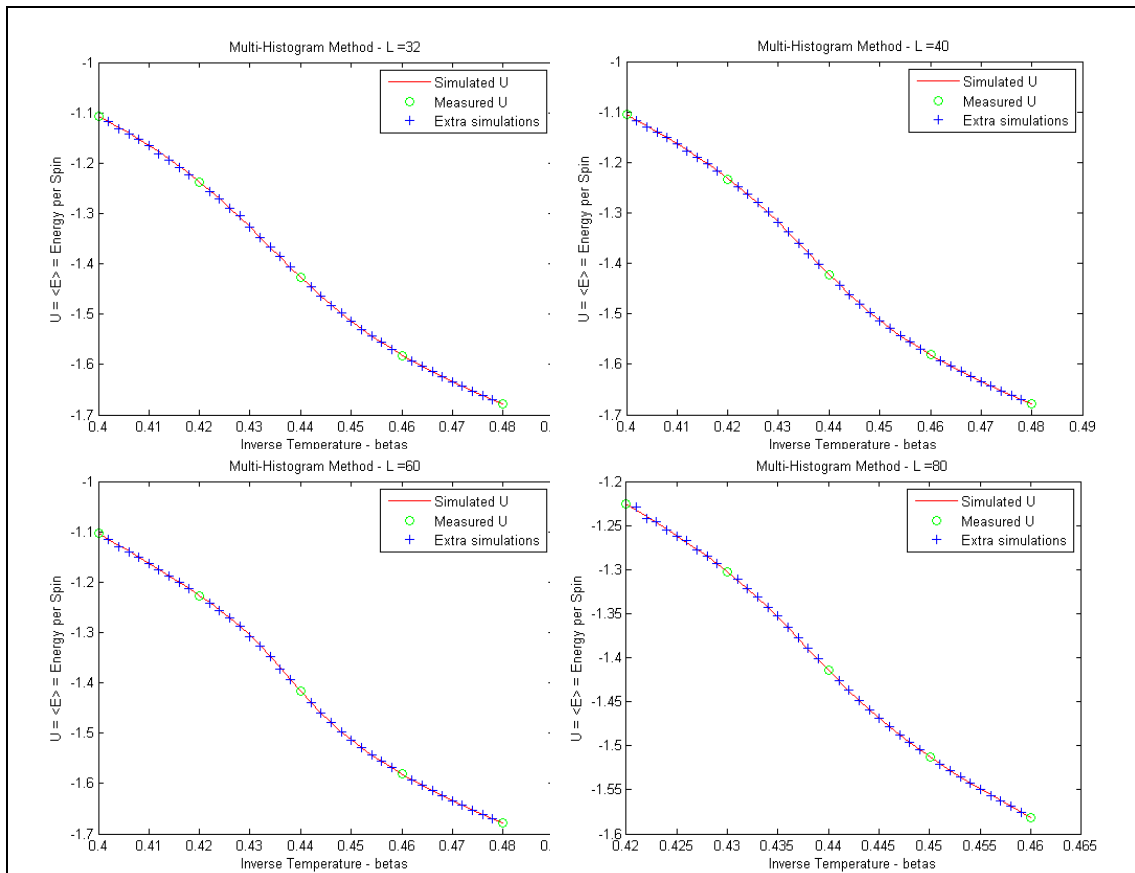
Εφαρμογή μεθόδου Multi Histogram για πλέγμα $L=16$

Measured U – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας που χρησιμοποιείται στη μέθοδο Multi Histogram

Simulated U – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Multi – Histogram

Extra Simulations – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τα αρχεία προσομοίωσης της κάθε θερμοκρασίας προς επαλήθευση της μεθόδου

Ο κατακόρυφος άξονας έχουμε τις συνολικές τιμές ενέργειας ενώ στον οριζόντιο άξονα τις θερμοκρασίες.



Εφαρμογή μεθόδου Multi Histogram για πλέγμα

$L=32, 40, 60 \text{ \& } 80$

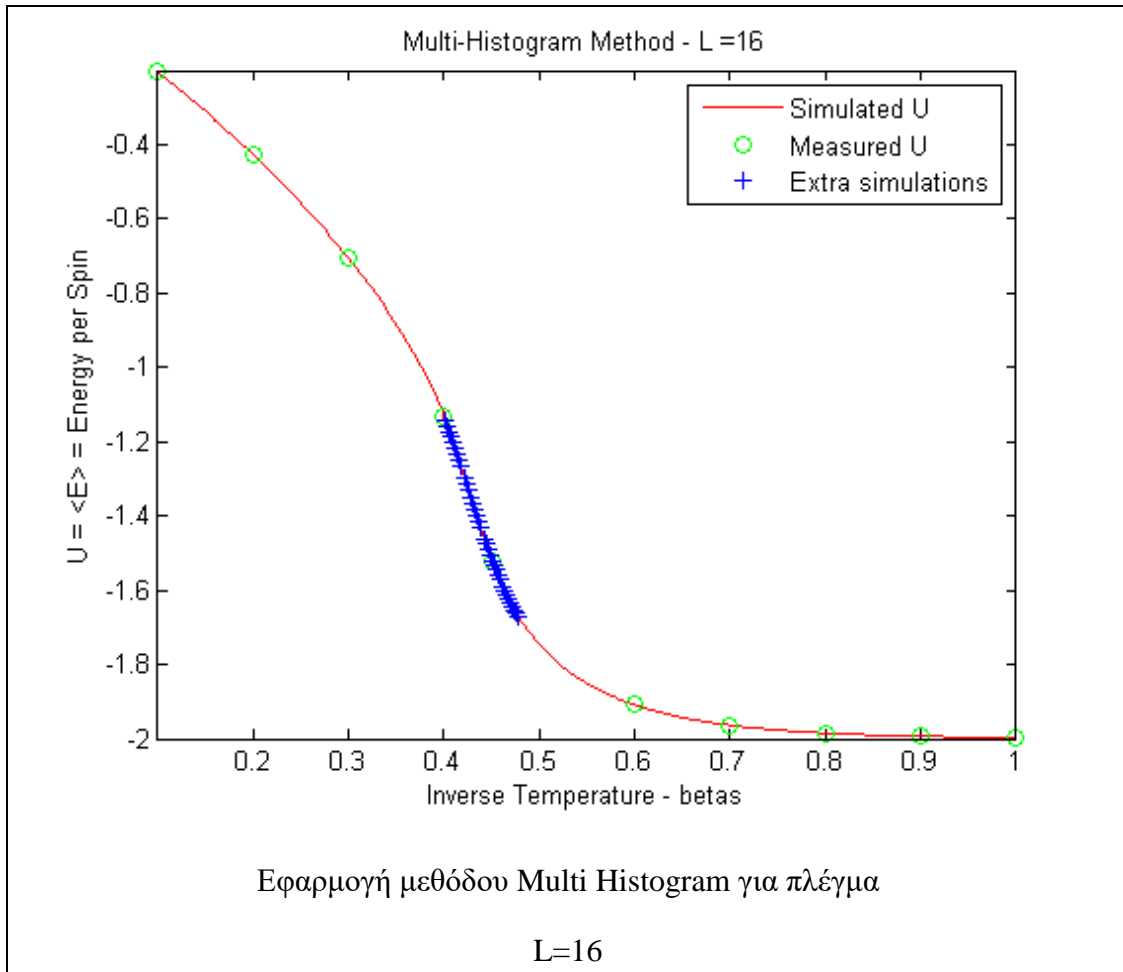
Measured U – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας που χρησιμοποιείται στη μέθοδο Multi Histogram

Simulated U – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Multi – Histogram

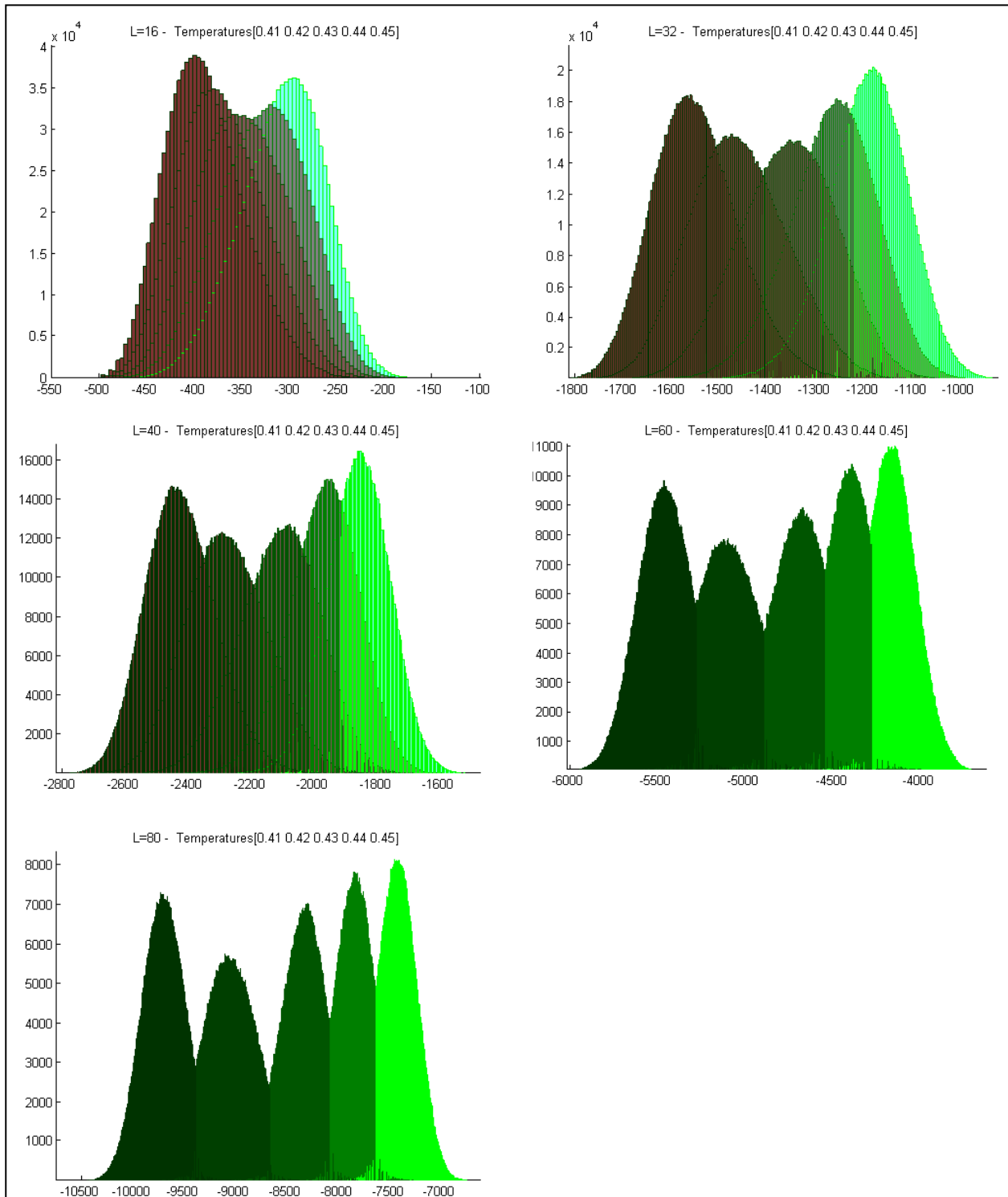
Extra Simulations – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τα αρχεία προσομοίωσης της κάθε θερμοκρασίας προς επαλήθευση της μεθόδου

Στον κατακόρυφο άξονα έχουμε τις συνολικές τιμές ενέργειας ενώ στον οριζόντιο άξονα τις θερμοκρασίες.

Ενδεικτικά για $L=16$, λόγω πυκνής επικάλυψης των ιστογραμμάτων και του μεγάλου εύρους τους για τις αντίστροφες θερμοκρασίες, η μέθοδος μπορεί να εφαρμοστεί με ακρίβεια για μεγάλο εύρος θερμοκρασιών:



Εφαρμόζοντας τη μέθοδο για μεγαλύτερα L , η διαδικασία απαιτεί προσομοιώσεις με πιο πυκνά β , ώστε να επιτυγχάνεται η επικάλυψη των ιστογραμμάτων. Ενδεικτικά παραθέτονται ιστογράμματα όπου φαίνεται η επικάλυψη για δεδομένες τιμές του β , αλλά αλλάζοντας κάθε φορά το L με 10^6 Wolff Clusters.



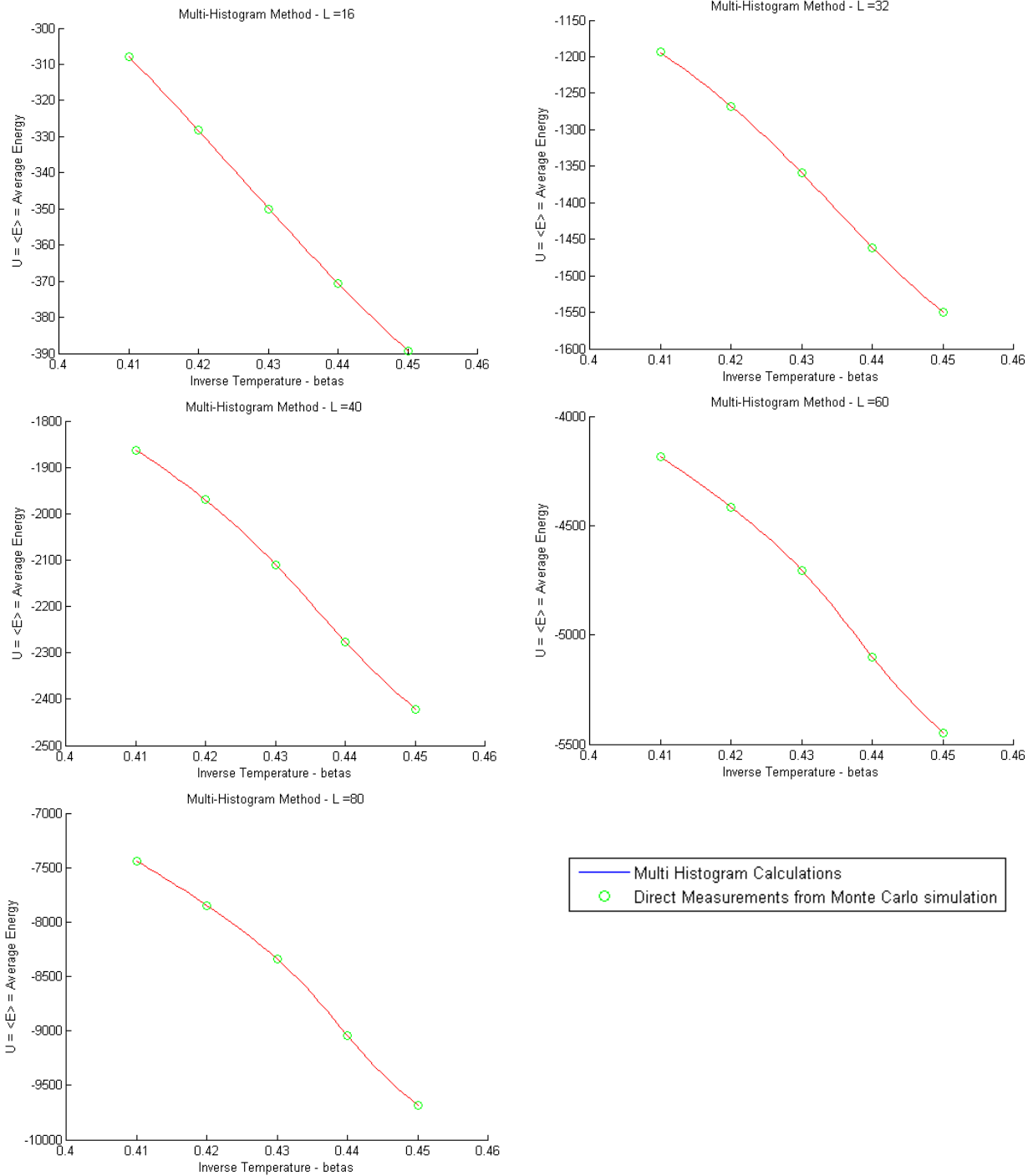
Ιστογράμματα επικάλυψης θερμοκρασιών για τη μέθοδο Multi Histogram για $L = 16, 32, 40, 60, 80$

Κατακόρυφος Άξονας: συχνότητα εμφάνισης ενέργειας

Οριζόντιος Άξονας: εύρος ενεργειών

Η επικάλυψη των ιστογραμμάτων είναι ένα σημαντικό κομμάτι της εφαρμογής της μεθόδου Multi Histogram. Το εύρος των προσεγγίσεων για τις κατανομές πιθανοτήτων σε κάθε θερμοκρασία θα μας δώσει σωστά αποτελέσματα, μόνο αν τα ιστογράμματα της προσομοίωσης για κάθε θερμοκρασία επικαλύπτονται.

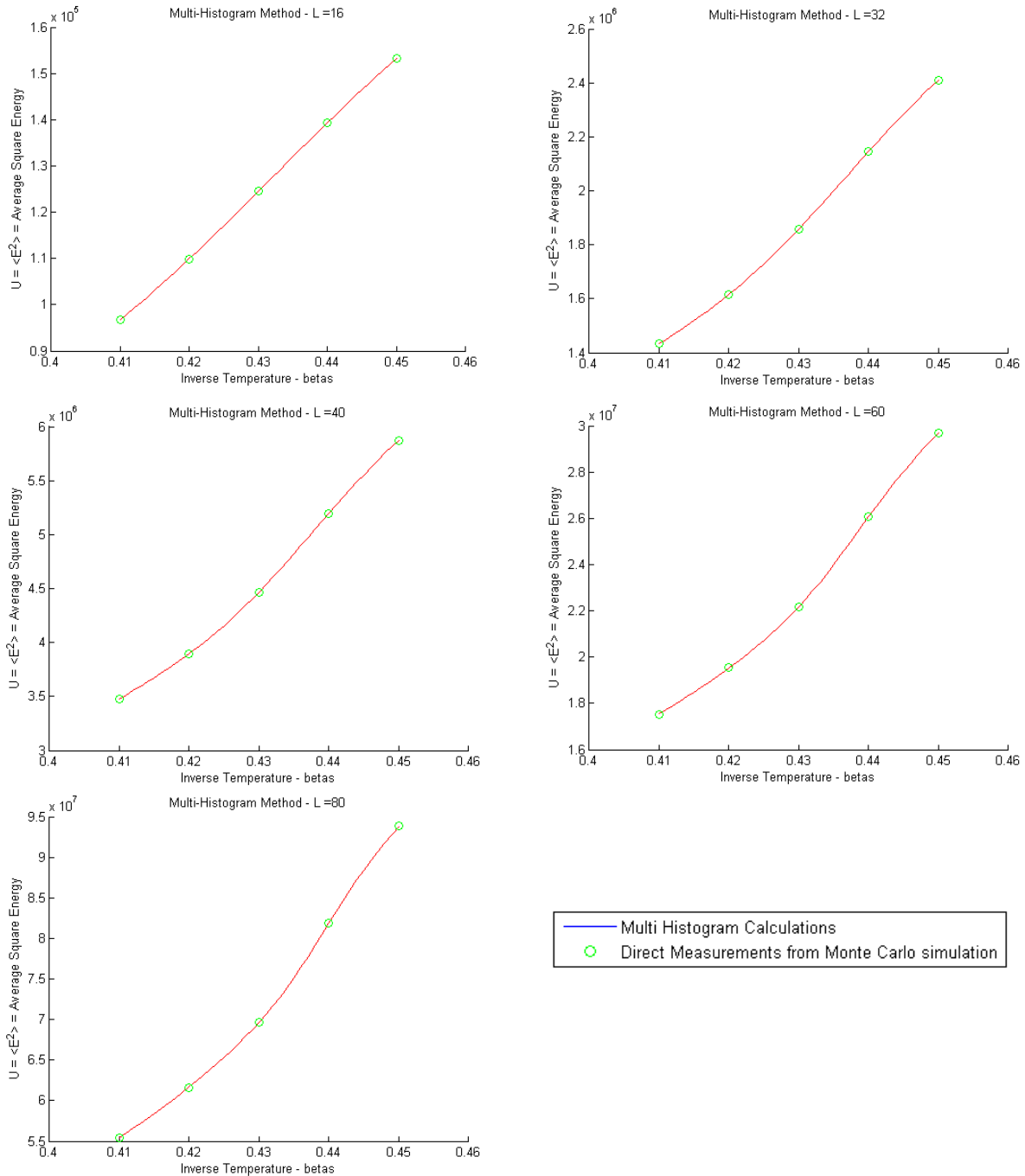
Διαγράμματα υπολογισμού μέσης ενέργειας συστήματος με τη μέθοδο Multi Histogram



Direct Measurements – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας

Multi Histogram Calculations – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Multi – Histogram

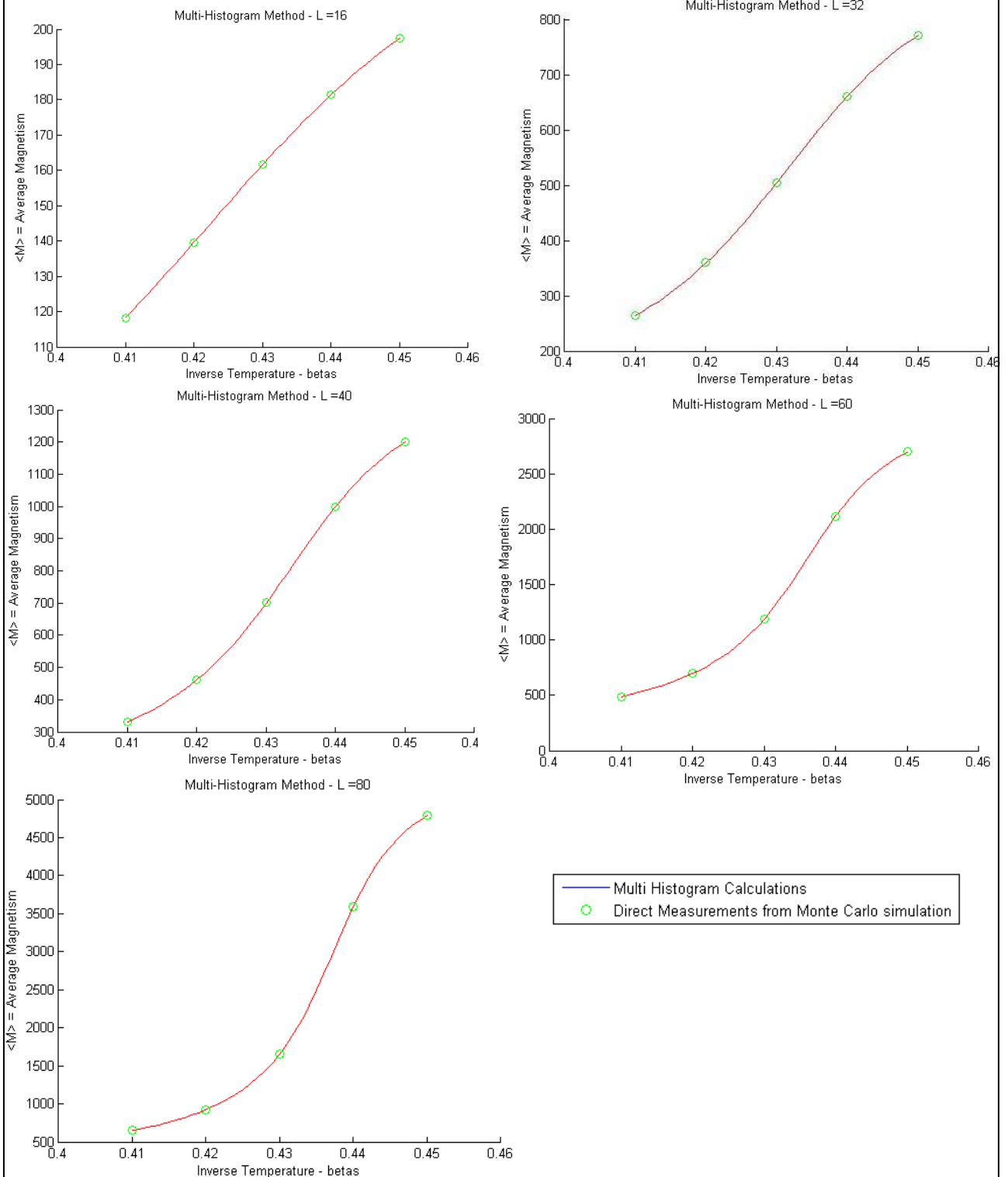
Διαγράμματα υπολογισμού μέσου τετραγώνου ενέργειας συστήματος με τη μέθοδο
 Multi Histogram



Direct Measurements – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας

Multi Histogram Calculations – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Multi – Histogram

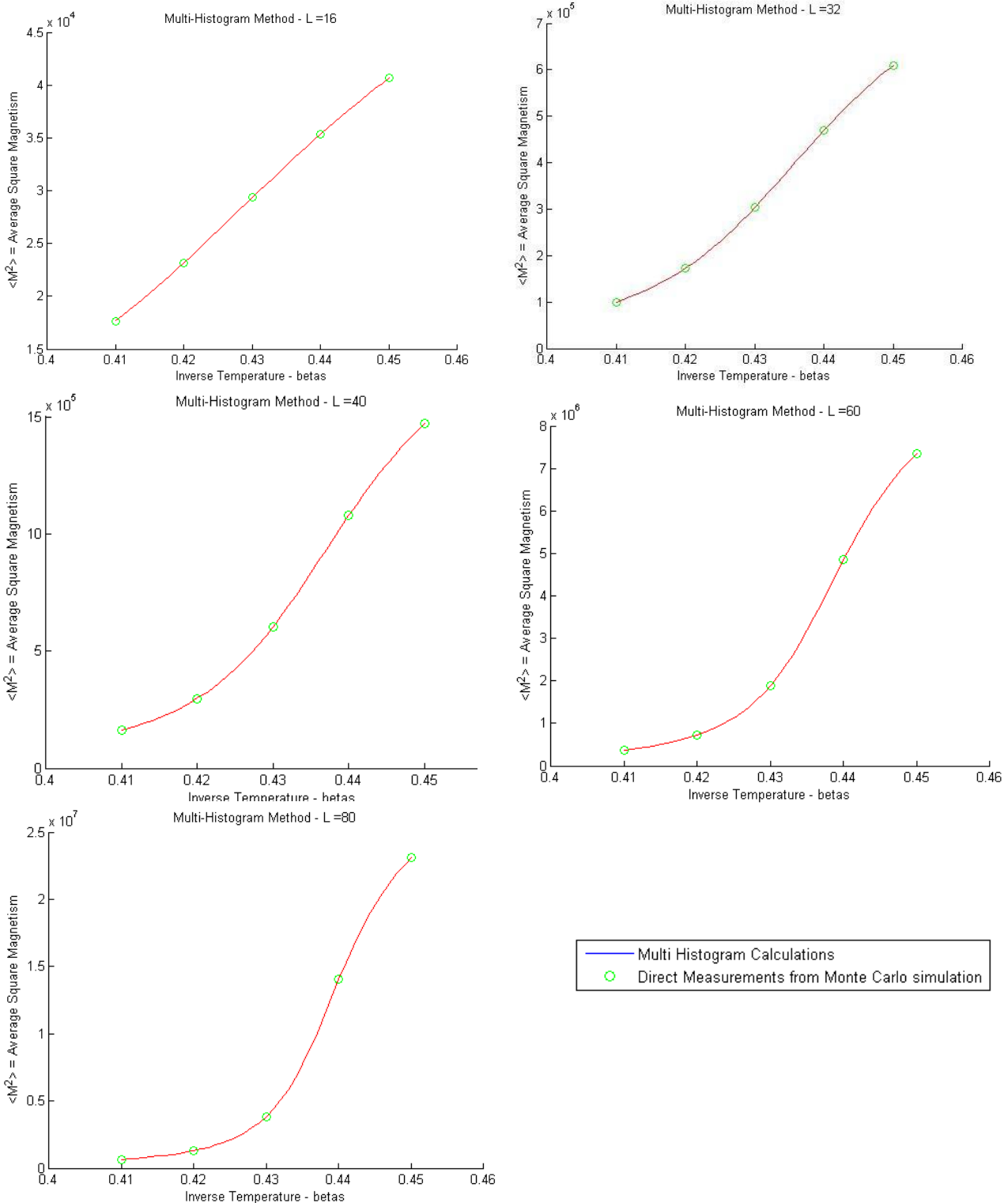
Διαγράμματα υπολογισμού μέσης μαγνήτισης συστήματος με τη μέθοδο Multi Histogram



Direct Measurements – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας

Multi Histogram Calculations – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Multi – Histogram

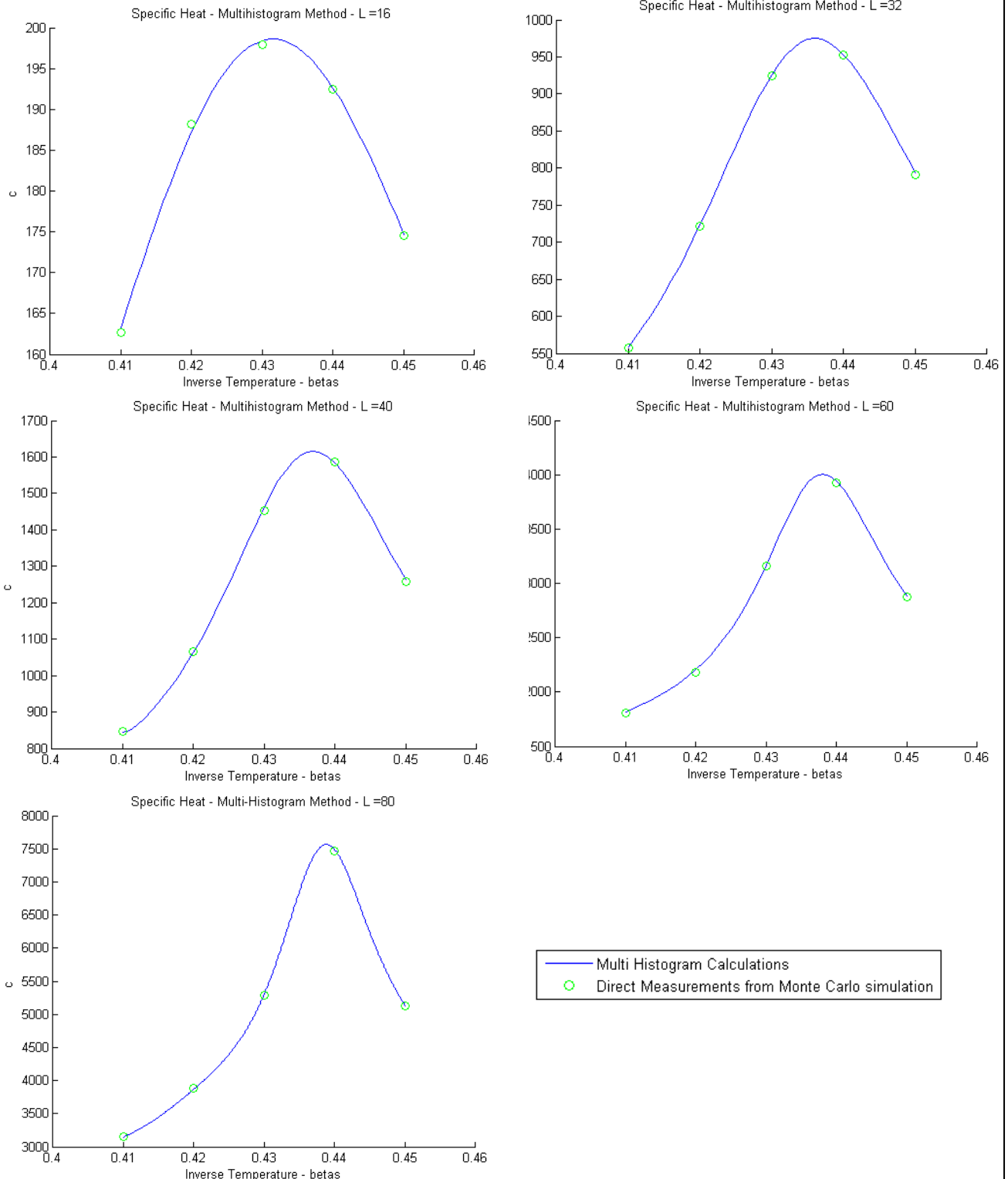
Διαγράμματα υπολογισμού μέσου τετραγώνου μαγνήτισης συστήματος με τη μέθοδο
 Multi Histogram



Direct Measurements – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας

Multi Histogram Calculations – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Multi – Histogram

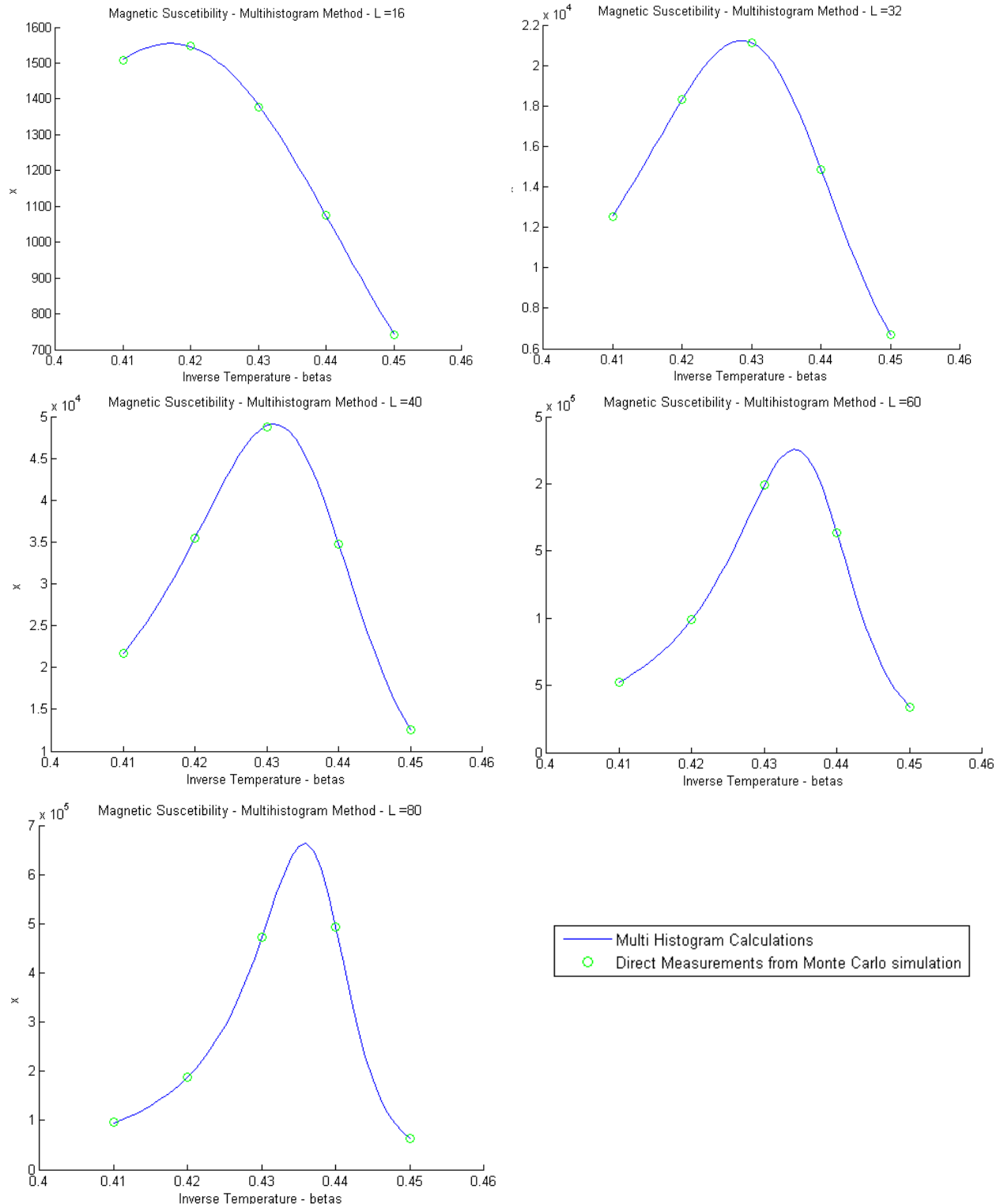
Διαγράμματα υπολογισμού ειδικής θερμότητας συστήματος με τη μέθοδο Multi Histogram



Direct Measurements – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας

Multi Histogram Calculations – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Multi – Histogram

Διαγράμματα υπολογισμού μαγνητικής επιδεκτικότητας συστήματος με τη μέθοδο
 Multi Histogram



Direct Measurements – σημεία που δείχνουν τις τιμές υπολογισμένες απευθείας από τη προσομοίωση Monte Carlo της κάθε θερμοκρασίας

Multi Histogram Calculations – σύνολο σημείων που προκύπτουν από την εφαρμογή της μεθόδου Multi – Histogram

Κεφάλαιο 6^ο

Ο υπολογισμός των κρίσιμων εκθετών θα γίνει με κλιμάκωση πεπερασμένου μεγέθους (Finite Size Scaling) μέσω των υπολογισθέντων τιμών που προέκυψαν από τη μέθοδο Multi – Histogram. Συγκεκριμένα εφαρμόστηκε η μέθοδος, με ακρίβεια υπολογισμού της τάξης του 10^{-6} . Παρακάτω, παρατίθενται σε πίνακα οι τιμές, που προσδιορίζουν αφενός τα μέγιστα της ειδικής θερμότητας και της μαγνητικής επιδεκτικότητας, μαζί με τα σφάλματά τους. Για κάθε μέγιστη τιμή που υπολογίζεται, η κρίσιμη θερμοκρασία που αντιστοιχεί σε αυτή τη τιμή κάθε φορά αλλάζει, τόσο για τις μετρήσεις της ειδικής θερμότητας όσο και για τις μετρήσεις της μαγνητικής επιδεκτικότητας.

L	16	32	40	60	80	100
c max	0,388080	0,475995	0,505088	0,556259	0,591140	0,619926
c max error	0,000499	0,000897	0,000900	0,000939	0,001543	0,001465
beta for c max	0,431470	0,435990	0,436940	0,438050	0,438810	0,439110
x max	6,072769	20,738653	30,692689	62,687564	103,722764	153,356364
x max error	0,003426	0,024706	0,030371	0,072572	0,221397	0,375653
beta for x max	0,416980	0,428530	0,430960	0,434180	0,435770	0,436770

Πίνακας 1: Μετρήσεις για πλέγματα L= 16, 32, 40, 60, 80, 100

L	150	200	300	400	500
c max	0,661956	0,696068	0,752609	0,785584	0,810801
c max error	0,003759	0,002406	0,002043	0,001888	0,001623
Beta for c max	0,439740	0,440010	0,440190	0,440260	0,440420

Πίνακας 2: Μετρήσεις για πλέγματα L= 150, 200, 300, 400, 500

Για το μοντέλο Ising σε δύο διαστάσεις (2D) οι κρίσιμοι εκθέτες περιγράφουν τη συμπεριφορά της ειδικής θερμότητας, της μαγνητικής επιδεκτικότητας και της μέσης μαγνήτισης κοντά στην κρίσιμη περιοχή.

Αντί να έχουμε τις τιμές της αντίστροφης θερμοκρασίας β , εισάγουμε την αδιάστατη θερμοκρασία t :

$$t \equiv \frac{\beta_c - \beta}{\beta_c}$$

Η ασυμπτωτική συμπεριφορά της ειδικής θερμότητας c για μικρό $|t|$ καθορίζεται από τον εκθέτη α : $c \sim |t|^{-\alpha}$

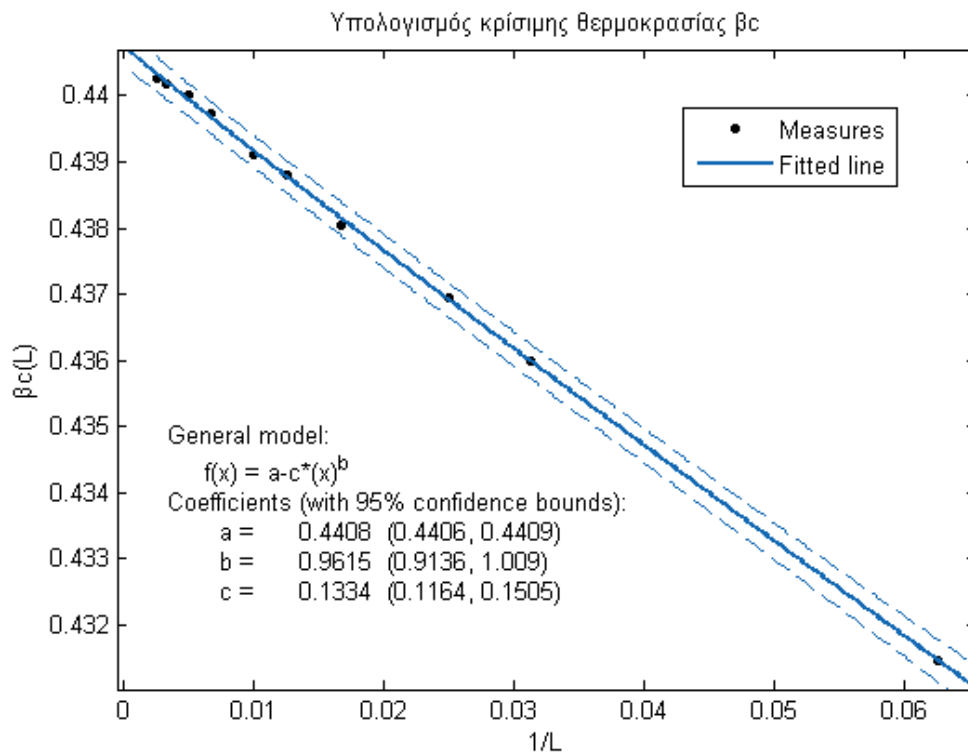
Ομοίως, η ασυμπτωτική συμπεριφορά της μαγνητικής επιδεκτικότητας x για μικρό $|t|$ καθορίζεται από τον εκθέτη γ : $x \sim |t|^{-\gamma}$

Οι τιμές των εκθετών είναι: $\alpha=0$ & $\gamma=7/4$.

Αρχικά όμως θα υπολογίσουμε τη κρίσιμη θερμοκρασία β_c , χρησιμοποιώντας τα αποτελέσματα του πίνακα 1.

Γνωρίζουμε πως:

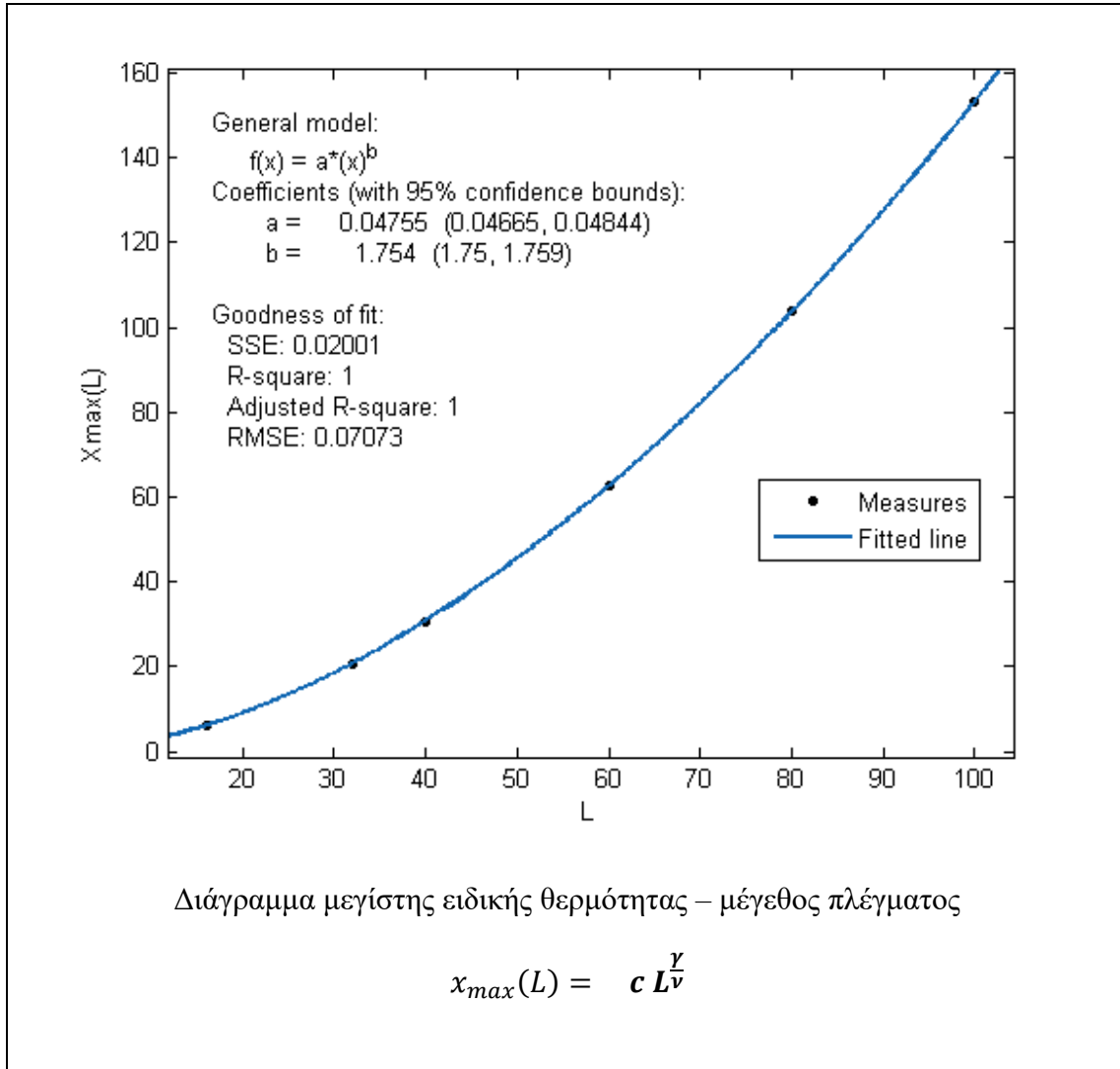
$$|t| = \left| \frac{\beta_c - \beta_c(L)}{\beta_c} \right| \sim L^{\frac{1}{\nu}} \rightarrow \beta_c(L) = \beta_c - \frac{c}{L^{\nu}}$$



Από τα παραπάνω προκύπτει πως $\beta_c = 0.44075 \pm 0.00015$, ενώ η τιμή του $\beta_c = \frac{\ln(1+\sqrt{2})}{2} = 0.440686793509771$, αποτέλεσμα που είναι εντός των ορίων της πραγματικής λύσης.

Χρησιμοποιώντας πάλι τα αποτελέσματα του πίνακα 1,
γνωρίζουμε πως:

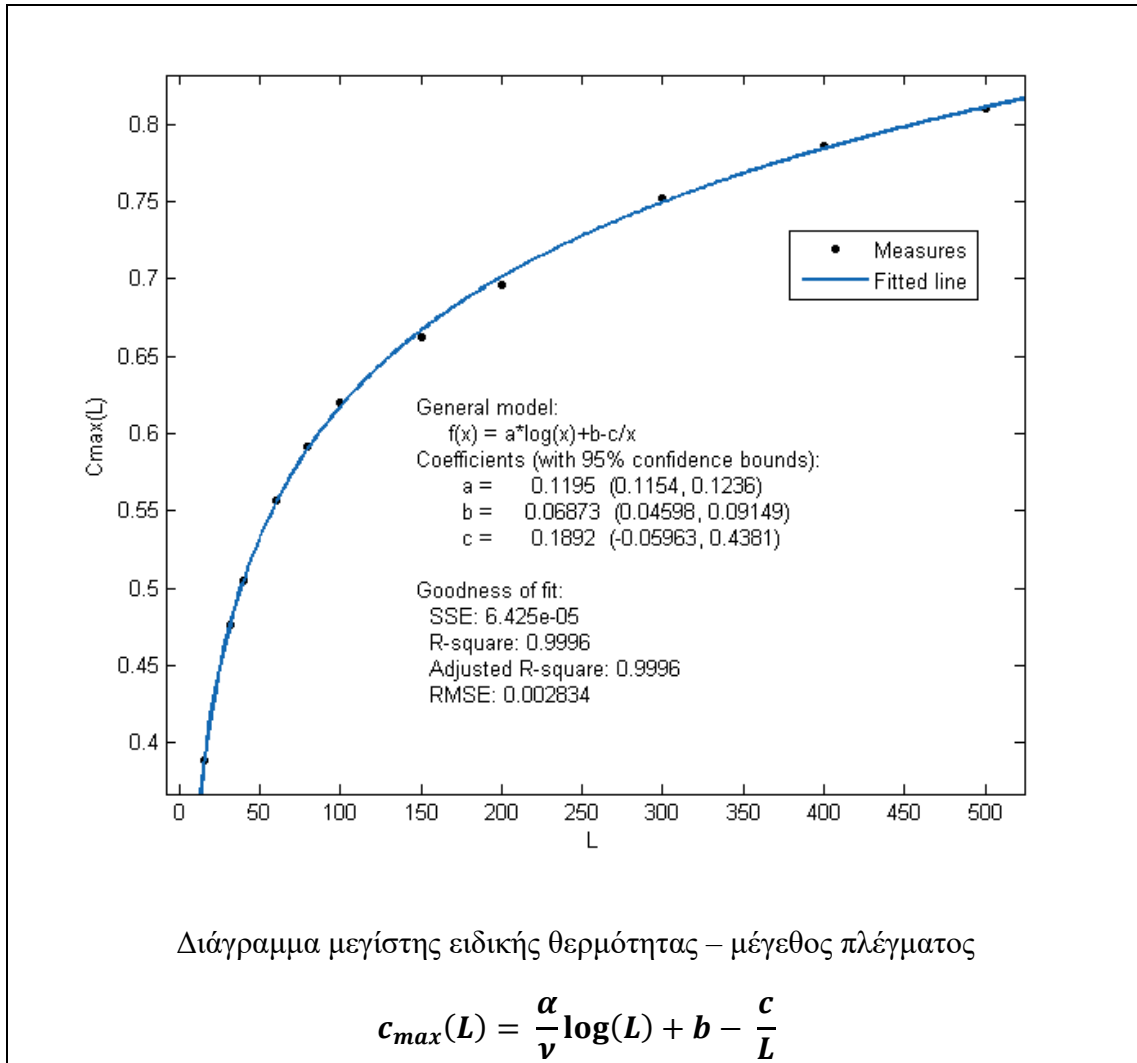
$$x \sim L^{\frac{\gamma}{\nu}} \rightarrow x_{max}(L) = c L^{\frac{\gamma}{\nu}}$$



Από τα παραπάνω, υποθέτοντας πως $\nu=1$ προκύπτει πως $\gamma = 1.7545 \pm 0.0045$, ενώ η τιμή του $\gamma = \frac{7}{4} = 1.75$, αποτέλεσμα που είναι οριακά εντός των ορίων της πραγματικής λύσης.

Χρησιμοποιώντας τα αποτελέσματα του πίνακα 1 και του πίνακα 2, γνωρίζουμε πως:

$$c \sim L^{\frac{\alpha}{\nu}} (\text{δεδομένου πως } \alpha = 0) \rightarrow c_{max}(L) = \frac{\alpha}{\nu} \log(L) + b - \frac{c}{L}$$



Από τα παραπάνω προκύπτει πως για να είναι συνεπή τα δεδομένα ως προς τις μετρήσεις και τους υπολογισμούς $\alpha = 0$.

Κεφάλαιο 7^ο

Για τον υπολογισμό των σφαλμάτων, θα χρησιμοποιήσουμε τη μέθοδο Binning, δηλαδή θα χωρίσουμε τα δεδομένα που πήραμε κατά τις μετρήσεις μας σε 10 επιμέρους, και για αυτά τα 10 επιμέρους αρχεία μετρήσεων θα εφαρμόσουμε τη μέθοδο Multi Histogram και θα έχουμε 10 σετ μετρήσεων. Από τα παραπάνω θα υπολογίσουμε τα σφάλματα τόσο για την εφαρμογή της μεθόδου σε απευθείας ποσότητες όπως η μέση ενέργεια και η μέση μαγνήτιση καθώς και τα τετράγωνα τους και τις μέσες τιμές των τετραγώνων τους, όσο και για έμμεσες όπως τη μαγνητική επιδεκτικότητα και την ειδική θερμότητα καθώς και για τους εκθέτες που αναφέραμε στο προηγούμενο κεφάλαιο.

Η διαδικασία υλοποίησης του αλγορίθμου παρήγαγε δεδομένα τα οποία κι αυτά έχουν ένα σφάλμα. Η σύγκριση των μετρούμενων μεγεθών και των τιμών έπειτα από την εφαρμογή της μεθόδου multi histogram για τις ίδιες θερμοκρασίες και το ίδιο μέγεθος πλέγματος δεν αποκλίνουν σημαντικά και τα αποτελέσματα φαίνονται στους πίνακες στο παράρτημα.

Κεφάλαιο 8^ο

Αναμφισβήτητα η μέθοδος είναι εφαρμόσιμη και ικανοποιητικά ακριβής στο επίπεδο που τη μελετάμε.

Τα ζητούμενα της μεθόδου Multi Histogram στη μελέτη του προτύπου Ising στις δυο διαστάσεις είναι πρωτίστως ο υπολογισμός οποιασδήποτε θερμοδυναμικής ποσότητας σε ζητούμενη θερμοκρασία και πλέγμα χρησιμοποιώντας δεδομένα από διαφορετικές θερμοκρασίες του ίδιου πλέγματος και δευτερογενώς ο προσδιορισμός της κρίσιμης θερμοκρασίας ενός συστήματος και ο υπολογισμός των κρίσιμων εκθετών που διέπουν τη κρίσιμη συμπεριφορά του μοντέλου στην κρίσιμη θερμοκρασία μετάβασης φάσης.

Οπότε πρωτίστως για πλέγματα από $L=16$ έως και $L=100$ με δεδομένα μεγέθους 10^6 Wolff clusters, τα αποτελέσματα της εφαρμογής της μεθόδου Multi Histogram σε σχέση με τα μετρούμενα από την προσομοίωση Monte Carlo έχουν απόκλιση μικρότερη του 1%. Για πλέγματα μεγαλύτερα του $L=100$, με δεδομένα μεγέθους 10^6 απαιτείται καλή επικάλυψη των ιστογραμμάτων ώστε να έχουμε ικανοποιητικά αποτελέσματα.

Από την άλλη ο προσδιορισμός της κρίσιμης θερμοκρασίας ενός συστήματος είχε ακρίβεια και σχετικά πολύ μικρό σφάλμα υπολογισμού, πολύ κοντά στην πραγματική. Το ίδιο για τον κρίσιμο εκθέτη γ .

Συγκεκριμένα στη παρούσα εργασία παρατίθεται στο κεφάλαιο 10 συνοπτικός πίνακας που εμφανίζονται με κόκκινο χρώμα οι αποκλίσεις των τιμών μεγαλύτερες του 1%.

Φαίνεται ξεκάθαρα πως σε υπολογιστικά προβλήματα φυσικής που απαιτείται μεγάλη υπολογιστική ισχύς. Η χρήση MatLab για παραγωγή δεδομένων είναι απαγορευτικά αργή και πρέπει να χρησιμοποιηθεί μια γλώσσα προγραμματισμού όπως η C. Στο επίπεδο αυτό με τη παρούσα διπλωματική επεκτάθηκα στην τροποποίηση του κώδικα της C ώστε να δημιουργεί αρχεία ιστογραμμάτων που χρησιμοποιεί η μέθοδος Multi Histogram, και σε περαιτέρω τροποποίηση ώστε να δημιουργεί ένα συνολικό αρχείο από τα επιμέρους αρχεία ιστογραμμάτων των υπό διερεύνηση θερμοκρασιών.

Κεφάλαιο 9^ο

Παρακάτω παρατίθεται αρχικά ο κώδικας σε γλώσσα προγραμματισμού C για την υλοποίηση του αλγόριθμου Wolff και στη συνέχεια ο κώδικας υλοποίησης της μεθόδου Multi Histogram σε MatLab.

Ο κώδικας που είναι γραμμένος σε C βασίστηκε κυρίως στους M.E.J. Newman & G.T. Barkema, και αναπτύχθηκε στο βιβλίο της αναφοράς 2.

Για τον αλγόριθμο Wolff απαιτείται η εγκατάσταση στο βασικό σύστημα Linux η εντολή make ώστε να είναι δυνατή η εκτέλεση του προγράμματος, το οποίο είναι το ίδιο με εκείνο που παρατίθεται σε γλώσσα Fortran στο βιβλίο Computational Physics, με κάποιες τροποποιήσεις που απλά διευκολύνουν τη διαδικασία εισαγωγής δεδομένων στη πορεία στη MatLab. Επιπλέον παρατίθεται ένα script αρχείο τροποποιημένο όπως τα παραπάνω, το οποίο εξάγει απευθείας ιστογράμματα για την ενέργεια και τη μαγνήτιση, πράγμα που διευκολύνει τη διαδικασία προχωρώντας σε μεγαλύτερα L, λόγω αδυναμίας της MatLab να διαχειριστεί μεγάλα αρχεία. Τα αρχεία που συνθέτουν το πρόγραμμα είναι τα ακόλουθα:

```
/****** init.c *****/
#include "include.h"

/* start= 0 cold
   1 hot
   -1 old config*/
void init(int argc, char **argv){
  int i;
  size_t len =1024;
  char *buf;
  double obeta=-1.0;
  int OL=-1;
  long int oseed=-1;
  FILE *fin;
  /*define parameters from options:*/
  L=-1;beta=-1.;nsweep=-1;start=-1;seed=-1;
  algorithm=0;//default is metropolis, 1 is wolff
  get_the_options(argc,argv);
  if( start == 0 || start == 1){
    if(L < 0 )locerr("L has not been set.");
    if(seed < 0 )locerr("seed has not been set.");
    if(beta < 0 )locerr("beta has not been set.");
    /*derived parameters*/
    N=L*L; XNN=1; YNN = L;
    /*allocate memory space for spins: */
    s = (int *)malloc((size_t) (N*sizeof(int)));
    if (!s) locerr("allocation failure in s[N]");/*if null pointer...*/
  }
}
```

```
if(start < 0 )locerr("start has not been set.");
if(nswEEP < 0 )locerr("nswEEP has not been set");
/*Initialize probabilities:*/
for(i=2;i<5;i+=2) prob[i] = exp(-2*beta*i);
/* wolff adding probability */
padd = 1.0 - exp(-2*beta);
switch(start){
case 0:/*cold start*/
    simmessage(stdout);
    for(i=0;i<N;i++) s[i]=1;
    break;
case 1:/*hot start*/
    simmessage(stdout);
    for(i=0;i<N;i++){
        if(drandom() < 0.5)
            s[i] = 1;
        else
            s[i] = -1;
    }
    break;
case 2:/*start from old configuration*/
    if(!(fin = fopen("conf", "r"))){locerr("Configuration file conf not found");
    buf=(char *)malloc((size_t) (len*sizeof(char)));/*allocate space for buf*/
    getline(&buf,&len,fin); /*do: "man getline" to understand usage...*/
    // to %*s modifier diabasei string kai to agnoi...
    fscanf(fin,"%*s %d %*s %d %*s %f %*s %ld",&OL,&OL,&obeta,&oseed);
    /* we check in case L has also been given by the options */
    if( L < 0 )L=OL;
    if( L != OL )locerr("Given L different from the one read from conf");
    N=L*L; XNN=1; YNN = L;
    if(seed < 0 ) seed = oseed;/*option overrides config*/
    if(beta < 0.) beta = obeta;
    s = (int *)malloc((size_t) (N*sizeof(int)));
    if (!s) locerr("allocation failure in s[N]");
    for(i=0;i<N;i++)
        if( fscanf(fin,"%d",&s[i]) != 1 || (s[i] != 1 && s[i] != -1))
            locerr("conf ended before reading s[N]");
    fclose(fin);
    // free((char *)buf); /*free allocated space for buffer buf*/
    simmessage(stdout);
    break;
default:
    printf("start= %d not valid. Exiting....\n",start);
    exit(1);
    break;
}
}
```

```
/****** main.c *****/
#include "include.h"
int main(int argc, char **argv){
```



```
int isweep;  
init(argc,argv);  
for(isweep=0;isweep<nsweep;isweep++){  
  if(algorithm == 1){wolff();}  
  else{met();}  
  measure();  
}  
simend();  
}
```

```
/****** measure.c *****/  
#include "include.h"  
void measure(){  
  printf ("%d %d\n",E(),M());  
}  
int E(){  
  int e,sum,i,nn;  
  e=0;  
  for(i=0;i<N;i++){  
    /*Sum of neighboring spins: only forward nn necessary in the sum*/  
    if((nn=i+XNN)>= N) nn -= N; sum = s[nn];  
    if((nn=i+YNN)>= N) nn -= N; sum += s[nn];  
    e += sum*s[i];  
  }  
  return -e;  
}  
int M(){  
  int i,m;  
  m=0;  
  for(i=0;i<N;i++) m+=s[i];  
  return m;  
}
```

```
/****** met.c *****/  
#include "include.h"  
void met(){  
  int i,k;  
  int nn,sum,delta;  
  for(k=0;k<N;k++){  
    i=N*drandom();//pick a site  
    /*Sum of neighboring spins*/  
    if((nn=i+XNN)>= N) nn -= N; sum = s[nn];  
    if((nn=i-XNN)< 0) nn += N; sum += s[nn];  
    if((nn=i+YNN)>= N) nn -= N; sum += s[nn];  
    if((nn=i-YNN)< 0) nn += N; sum += s[nn];  
    /*change in energy*/  
    delta = sum*s[i];  
    /*flip*/  
    if(delta <=0 ) {s[i] = -s[i];}
```

```
else if(drandom()<prob[delta]){s[i] = -s[i];}
} //sweep ends
}
```

```
/****** options.c *****/
#include "include.h"
char prog[1024];
/*Get the options function: See "man 3 getopt" for usage*/
/*Option letters are defined with this string*/
#define OPTARGS "?h1234567890.L:b:s:S:n:w"
void get_the_options(int argc,char **argv){
int c,errflg = 0;
strcpy(prog,(char *)basename(argv[0]));
while (!errflg && (c = getopt(argc, argv, OPTARGS)) != -1){
switch(c){
case 'w':
algorithm = 1;
break;
case 'L':
L = atoi(optarg);
break;
case 'b':
beta = atof(optarg);
break;
case 's':
start = atoi(optarg);
break;
case 'S':
seed = atol(optarg);
break;
case 'n':
nsweep = atoi(optarg);
break;
case 'h':
errflg++;/*call usage*/
break;
case '?':
errflg++;/*call usage*/
break;
case '1':case '2':case '3':case '4':case '5':case '6':case '7':case '8':
case '9':case '0':case '.':
/*optind--;/*it was not meant to be an option*/
return;
default:
errflg++;
}/*switch*/
if(errflg) usage(argv);
}/*while...*/
}/*get_the_options()*/
void usage(char **argv){
```

```
/*Careful: New lines end with \n\ : No space after last backslacsh
 indicates l ine is broken...*/
fprintf(stderr,"\
Usage: %s [options]                \n\
-L: Lattice length (N=L*L)         \n\
-b: beta (options beta overrides the one in config) \n\
-s: start (0 cold, 1 hot, 2 old config.) \n\
-S: seed (options seed overrides the one in config) \n\
-n: number of sweeps and measurements of E and M \n\
-w: use wolff algorithm for the updates \n\
Monte Carlo simulation of 2d Ising Model. Metropolis is used by \n\
default, cluster algorithms can be chosen with options. Using the \n\
options, the parameters of the simulations must be set for a new run\n\
(start=0,1). If start=2, a configuration is read from the file conf.\n",prog);
// printed usage message "
exit(1);
}/*usage()*/
void locerr( char *errmes ){
fprintf(stderr,"%s: %s Exiting...\n",prog,errmes);
exit(1);
}
/*print a message about the simulation: */
#include <time.h>
void simmessage(FILE *fp){
char USER[100],HOST[100],MACH[100];
time_t t;
time(&t);/* store time in seconds in *t. see: "man 2 time" */
strcpy(USER,getenv("USER"));
strcpy(MACH,getenv("HOSTTYPE"));
strcpy(HOST,getenv("HOST"));
fprintf(fp,"\
#####\n\
# 2d Ising Model with Metropolis algorithm on square lattice \n\
# Run on %s ( %s ) by %s on %s# \n\
# L = %d (Lattice linear dimension, N=L*L) \n\
# seed = %ld (random number gener. seed) \n\
# nsweeps = %d (No. of sweeps) \n\
# beta = %f \n\
# start = %d (0 cold, 1 hot, 2 old config)\n",
HOST,MACH,USER,ctime(&t),
L,seed,nsweep,beta,start);
fflush(fp);
}/* message()*/
```

wolff.c

```
/* padd = 1 - exp(-2*beta*J)
 */
#include "include.h"
void wolff(){
int i,sp,oldspin,newspin,current,nn;
```

```
int *stack;
int ncluster;
/*ask for the stack memory*/
stack = (int *)malloc((size_t)(N*sizeof(int)));
if(!stack)locerr("allocation failure for stack in wolff()");
/*choose the seed spin for the cluster, put it on the stack and flip it */
i=N*drandom();
stack[0]=i;
sp=1;
oldspin = s[i];
newspin = -s[i];
s[i] = newspin;
ncluster = 1;
/*start loop on spins on the stack: */
while(sp){
  /*Pull a site off the stack */
  current = stack[--sp]; //value of --sp is **after** decrement
  /*check neighbours*/
  if((nn=current+XNN)>=N) nn -= N;
  if(s[nn] == oldspin)
    if(drandom(<padd){
      stack[sp++] = nn; //value of sp++ is **before** increment
      s[nn] = newspin; //flip the spin of cluster
      ncluster++;
    }
  if((nn=current-XNN)<0) nn += N;
  if(s[nn] == oldspin)
    if(drandom(<padd){
      stack[sp++] = nn; //value of sp++ is **before** increment
      s[nn] = newspin; //flip the spin of cluster
      ncluster++;
    }
  if((nn=current+YNN)>=N) nn -= N;
  if(s[nn] == oldspin)
    if(drandom(<padd){
      stack[sp++] = nn; //value of sp++ is **before** increment
      s[nn] = newspin; //flip the spin of cluster
      ncluster++;
    }
  if((nn=current-YNN)<0) nn += N;
  if(s[nn] == oldspin)
    if(drandom(<padd){
      stack[sp++] = nn; //value of sp++ is **before** increment
      s[nn] = newspin; //flip the spin of cluster
      ncluster++;
    }
} /*while(sp)*/
printf("#clu %d\n",ncluster);
free(stack);
} /*wolff()*/
```

```
/****** drandom.c *****/
#include "include.h"
#define a 16807
#define m 2147483647
#define q 127773
#define r 2836
#define conv (1.0/(m-1))
double drandom(){
    long l;
    l=seed/q;
    seed = a*(seed-q*l) - r*l;
    if(seed < 0) seed +=m;
    // printf("r= %f s= %d\n",conv*(seed-1),seed);
    return conv*(seed-1);
}
```

```
/****** end.c *****/
#include "include.h"
void simend(){
    FILE *fout;
    int i;
    /* output the configuration */
    rename("conf","conf.old");
    fout = fopen("conf","w");
    fprintf(fout,"# Configuration of 2d Ising model on square lattice. Parameters (N=Lx*Ly) and
s[N]\n");
    fprintf(fout,"Lx= %d Ly= %d beta= %f seed= %ld\n",L,L,beta,seed);
    for(i=0;i<N;i++){
        fprintf(fout,"%d\n",s[i]);
    }
    fclose(fout);
}
```

```
/****** include.h *****/
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include <string.h>
int L;
int N;
int XNN, YNN;
int *s;
double beta;
int nsweep,start;
double prob[5];
int E(), M();
```

```
double drandom();
long seed;
void init(int, char **), met(), measure(),simmessage(FILE *),locerr(char*);
void usage(char **),get_the_options(int,char**),simend();
//wolff cluster: adding probability
double padd;
int algorithm;
```

Makefile

```
##### Makefile.1 #####
#@ target, $< first dependency, $^ all dependencies,
$* "stem" of target filename e.g. %.f: %.F will five foo for foo.f
OBJS = main.o init.o wolff.o met.o measure.o end.o options.o drandom.o
CFLAGS = -g
LIBS = -lm
is: $(OBJS)
$(CC) $(CFLAGS) $^ -o $@ $(LIBS)
$(OBJS): include.h
clean:
/bin/rm -f *.o is core*
```

Παρακάτω είναι το script που υλοποιεί τον αλγόριθμο Wolff και δημιουργεί τα ιστογράμματα και εκτελείται από οποιοδήποτε τερματικό Linux με την εντολή `./run`.

run

```
#!/bin/tcsh -f
set Ls = (80)
set betas = (0.41 0.42 0.43 0.44 0.45 0.46 0.47)
set nsweeps = 10000000
foreach L ($Ls )
  mkdir Raw_data/${L}
  mkdir Hist_E/${L}
  mkdir Hist_EM/${L}
  mkdir HIST/${L}
set start = "-s 1 -S 3423" #-s: 1 (hot)/0 (cold)/2 (old)
foreach beta ($betas)
  echo "L= $L beta= $beta"
  #ising simulation
  ./is -L $L -b $beta -n $nsweeps -w $start > Raw_data/${L}/outL${L}b${beta}
  #histogram creation for ENERGY and MAGNETISM
  cat Raw_data/${L}/outL${L}b${beta} | sort -n | uniq -c >
  Hist_EM/${L}/histEM_L${L}b${beta}
  #histogram creation for ENERGY only
```

```
cat Raw_data/$(L)/outL$(L)b$(beta) | cut -d " " -f 1 | sort -n | uniq -c >  
Hist_E/$(L)/histE_L$(L)b$(beta)  
set start = "-s 2"  
end  
#multiple histogram creation for all betas  
cat Raw_data/$(L)/*. * | sort -n | uniq -c > HIST/$(L)/L$(L)EM  
cat Raw_data/$(L)/*. * | cut -d " " -f 1 | sort -n | uniq -c > HIST/$(L)/L$(L)E  
end
```

Στη συνέχεια ακολουθεί ένα αρχείο - συνάρτηση κώδικα για την υλοποίηση της μεθόδου Single Histogram και έπειτα ο κώδικας σε MatLab για την υλοποίηση της μεθόδου Multi Histogram.

singlehistfun.m

```
function [U,UU]=singlehistfun(L,i)  
% singlehistfun(L,i)  
% usage singlehistfun(16,0.43)  
a=1;  
s{a,1} = strcat('outL',mat2str(L),'b',mat2str(i))  
s{a,2} = i;  
data2{1,1} = dlmread(s{1},' ',9,0);  
data2{1,2} = s{1,1}  
data2{1,3} = s{1,2}  
data2{1,5} = min(data2{1,1}(:,1)):max(data2{1,1}(:,1));  
[data2{1,4}] = histc(data2{1,1}(:,1),data2{1,5});  
figure  
bar(data2{1,5},data2{1,4},'histc')  
NAMEtitle = strcat('Single-Histogram Method - L = ',mat2str(L));  
title(NAMEtitle)  
betas = [data2{1,3}];  
rangebeta = (betas-0.04):0.001:(betas+0.04);  
[~,rangebetas] = size(rangebeta);  
U = zeros(rangebetas,1);  
for i = 1:rangebetas  
    SumArithmith=0;  
    SumParonomasth = 0;  
    [b,~]=size(data2{1,4});  
    for E = 1:b  
        N_E=[data2{1,4}(E,1)];  
        EE = [data2{1,5}(1,E)];  
        SumArithmith = SumArithmith+EE*N_E*exp(((data2{1,3}]-rangebeta(1,i))*EE);  
        SumParonomasth = SumParonomasth + N_E*exp(((data2{1,3}]-rangebeta(1,i))*EE);  
        U(i,1) = (SumArithmith/SumParonomasth);  
    end  
end  
U = U/(L*L);  
figure  
plot(rangebeta,U,'color','r')  
hold  
a=1;  
for i = min(rangebeta):0.01:max(rangebeta)  
    s{a,1} = strcat('outL',mat2str(L),'b',mat2str(i))
```

```
s{a,2} = i;
a=a+1;
UU(a-1,2)=[i];
end
a = a-1;
[bb,~] = size(s);
for i = 1:bb
    data2{i,1} = dlmread(s{i},'',9,0);
    data2{i,2} = s{i,1}
    data2{i,3} = s{i,2}
end
[a,~] = size(data2);
for i = 1:a
    data2{i,5} = min(data2{i,1}{:,1}):max(data2{i,1}{:,1});
    [data2{i,4}] = histc(data2{i,1}{:,1},data2{i,5});
end
for i = 1:a
    UU(i,1) = mean(data2{i,1}{:,1})/(L^2);
end
plot(UU(:,2),UU(:,1),'o')
legend('extrapolated','measured')
NAMEtitle = strcat('Single-Histogram Method - L = ',mat2str(L));
title(NAMEtitle)
xlabel('Inverse Temperature - betas')
ylabel(' U = <E> ')
hold
clear
```

MULTI_HISTOGRAM_METHOD.m

```
Ls = [16 32 40 60 80];
[~,iii] = size(Ls);
for w = 1:iii;
%% GLOBAL VARIABLE SETUP
cutoff = 10^(-10); %CUTOFF LEVEL FOR Z CALCULATION ITERATION
Extrapolation_step = 0.00001;
temp = [0.43 0.44 0.45]; %TEMPERATURES
L=Ls(1,w);
Q=99500;
%% DATA IMPORT
startQ=1;
endQ=99500;
a=1;
for i = temp
    s{a,1} = strcat('outL',mat2str(L),'b',mat2str(i));
    s{a,2} = i;
    a=a+1;
end
a = a-1;
for i = 1:a
    data2{i,1} = dlmread(s{i},'',5000,0); %RAW DATA FROM SIMULATION
    data2{i,1}{:,2}=abs(data2{i,1}{:,2}); %ABSOLUTE VALUE OF MAGNETISM
```



```

end
for fors= 1:10
for i = 1:a
    data{i,1} = data2{i,1}(startQ:endQ,:);
    data{i,2} = s{i,1}; %NAME OF FILE
    data{i,3} = s{i,2}; %SIMULATED BETA - INVERSE
    TEMPERATURE
    data{i,4} = mean(data{i,1}(:,1)); %AVERAGE ENERGY <E>
    data{i,5} = mean(data{i,1}(:,1).^2); %AVERAGE SQUARE ENERGY <E^2>
    data{i,6} = (data{i,3}.^2)*(data{i,5}-data{i,4}.^2)/2; %c specific heat
    data{i,7}=mean(data{i,1}(:,2)); %MEAN MAGNETISM <M>
    data{i,8}=mean((data{i,1}(:,2)).^2); %MEAN SQUARE MAGNETISM
    <M^2>
    data{i,9} = (data{i,3})*(data{i,8}-data{i,7}.^2); %x magnetic susceptibility
    t(i,1) = (log(1+sqrt(2))/2-data{i,3})/log(1+sqrt(2))/2; %SCALED TEMPERATURE
end
[n,~] = size(data{1,1});
clear i scans
startQ=startQ+Q;
endQ=endQ+Q;
%% HISTOGRAM CREATION
ENERGYcounts= 0;
MAGNETISMcounts= 0;
% FINDING MIN - MAX FOR ENERGY AND MAGNETISM
for i = 1:a
    minE(i,1)=min(data{i,1}(:,1));
    maxE(i,1)=max(data{i,1}(:,1));
    minM(i,1)=min(data{i,1}(:,2));
    maxM(i,1)=max(data{i,1}(:,2));
end
minE = min(minE);
maxE = max(maxE);
minM = min(minM);
maxM= max(maxM);
for i = 1:a
    histogram{i,1} = minE:maxE;
    histogram{i,3} = minM:maxM;
    [histogram{i,2}] = histc(data{i,1}(:,1),histogram{i,1});
    [histogram{i,4}] = histc(data{i,1}(:,2),histogram{i,3});
    ENERGYcounts = ENERGYcounts+histogram{i,2};
    MAGNETISMcounts=MAGNETISMcounts+histogram{i,4};
end
ENERGYrange = minE:maxE;
MAGNETISMrange = minM:maxM;
ENERGYcounts(ENERGYcounts==0)=nan;
id = isnan(ENERGYcounts);
ENERGYrange(id)=nan;
ENERGYcounts(~any(~isnan(ENERGYcounts), 2),:)=[];
ENERGYrange = ENERGYrange';
ENERGYrange(~any(~isnan(ENERGYrange), 2),:)=[];
ENERGYrange = ENERGYrange';
MAGNETISMcounts(MAGNETISMcounts==0)=nan;

```

```
id2 = isnan(MAGNETISMcounts);
MAGNETISMrange(id2)=nan;
MAGNETISMcounts(~any(~isnan(MAGNETISMcounts), 2),:)=[];
MAGNETISMrange = MAGNETISMrange';
MAGNETISMrange(~any(~isnan(MAGNETISMrange), 2),:)=[];
MAGNETISMrange = MAGNETISMrange';
clear id id2
edges={ENERGYrange MAGNETISMrange};
[k,~]=size(ENERGYcounts);
[l,~]=size(MAGNETISMcounts);
hist_matrix = zeros(k,l);
for i=1:a
    hist_matrix =hist_matrix + hist3(data{i,1},'Edges',edges);
end
hist_matrix = transpose(hist_matrix);
figure
hold
width1=4;
for i = 1:a
    c=1/i;
    bar(histogram{i,1},histogram{i,2},width1,'FaceColor',[0.5,c,c],'EdgeColor',[0,c,0])
end
hold
[mag,~]=size(MAGNETISMcounts);
clear minM maxM minE maxE i k l histogram edges
%% ITERATION OF Zk's for ENERGY
figure
for i = 1:a
    betas(i,1) = data{i,3};
    Fk(1,i) = zeros;
    F0k(1,i) = ones;
end
clear i
[~,b] = size(ENERGYrange);
while abs(Fk -F0k)>cuttoff
    for k=1:a
        for EE=1:b
            E = ENERGYrange(1,EE);
            for j = 1:a
                js(j,EE) = n*exp(betas(j,1)*Fk(1,j)+betas(k,1)*E-betas(j,1)*E);
            end
        end
        SUMjs = sum(js);
        paronom = SUMjs.^(-1);
        energieies = bsxfun(@times,ENERGYcounts',paronom);
        SUMenergieies(k,1)= sum(energieies);
        F0k(1,k)=Fk(1,k);
        Fk(1,k) = -log(SUMenergieies(k,1))/(betas(k,1));
        Zk_E(k,1) = exp(-betas(k,1)*Fk(1,k));
    end
    diaforaF =Fk - F0k;
    bar(temp,diaforaF);
```

```
drawnow;
end
% clear diaforaF E EE energieies F0k j js k paronom SUMenergeies SUMjs
%% Z calculation for ENERGY extrapolated betas
beta_min= min(betas);
beta_max=max(betas);
steps = Extrapolation_step;
beta_extrapolated= beta_min:steps:beta_max;
[f,~] = size(beta_extrapolated');
F=zeros(1,f);
clear beta_max beta_min
for k=1:f
    for EE=1:b
        E = ENERGYrange(1,EE);
        for j =1:a
            js(j,EE) = n*exp(betas(j,1)*Fk(1,j)+beta_extrapolated(1,k)*E-betas(j,1)*E);
        end
    end
    SUMjs = sum(js);
    paronom = SUMjs.^(-1);
    energieies = bsxfun(@times,ENERGYcounts',paronom);
    SUMenergeies(k,1)= sum(energieies);
    F(1,k) = -log(SUMenergeies(k,1))/(beta_extrapolated(1,k));
    Z_E(k,1) = exp(-beta_extrapolated(1,k)*F(1,k));
end
%% QUANTITES CALCULATION
U_s =zeros(f,1);
for k=1:f
    for EE=1:b
        E = ENERGYrange(1,EE);
        for j =1:a
            js(j,EE) = n*exp(betas(j,1)*Fk(1,j)+beta_extrapolated(1,k)*E-betas(j,1)*E);
        end
    end
    SUMjs = sum(js);
    paronom = SUMjs.^(-1);
    NE = bsxfun(@times,ENERGYrange,ENERGYcounts');
    energieies = bsxfun(@times,NE,paronom);
    SUMenergeies(k,1)= sum(energieies);
    U_s(k,1)=SUMenergeies(k,1)/Z_E(k,1);
end
U_s2 =zeros(f,1);
for k=1:f
    for EE=1:b
        E = ENERGYrange(1,EE);
        for j =1:a
            js(j,EE) = n*exp(betas(j,1)*Fk(1,j)+beta_extrapolated(1,k)*E-betas(j,1)*E);
        end
    end
    SUMjs = sum(js);
    paronom = SUMjs.^(-1);
    NE = bsxfun(@times,ENERGYrange.^2,ENERGYcounts');
```

```
energieies = bsxfun(@times,NE,paronom);
SUMenergieies(k,1)= sum(energieies);
U_s2(k,1)=SUMenergieies(k,1)/Z_E(k,1);
end
clear E EE energieies j js k N_E paronom SUMenergieies SUMjs NE i
%% ITERATION OF Zk's for MAGNETISM
for i = 1:a
    betas(i,1) = data{i,3};
    Fk(1,i) = zeros;
    F0k(1,i) = ones;
end
clear i
[~,b] = size(ENERGYrange);
while abs(Fk -F0k)>cutoff
    for k=1:a
        for EE=1:b
            E = ENERGYrange(1,EE);
            for j = 1:a
                js(j,EE) = n*exp(Fk(1,j)+betas(k,1)*E-betas(j,1)*E);
            end
        end
        SUMjs = sum(js);
        paronom = SUMjs.^(-1);
        %     for i = 1:b
        %         energieies(:,i)=hist_matrix(:,i)*paronom(:,i);
        %     end
        energieies = bsxfun(@times,hist_matrix,paronom);
        SUMenergieies(k,1)= sum(sum(energieies));
        F0k(1,k)=Fk(1,k);
        Fk(1,k) = -log(SUMenergieies(k,1));
        Zk_M(k,1) = exp(-Fk(1,k));
    end
    diaforaF =Fk - F0k;
    bar(temp,diaforaF);
    drawnow;
end
clear diaforaF E EE energieies F0k j js k paronom SUMenergieies SUMjs
%% Z calculation for MAGNETISM extrapolated betas
beta_min= min(betas);
beta_max=max(betas);
steps = Extrapolation_step;
beta_extrapolated= beta_min:Extrapolation_step:beta_max;
[f,~] = size(beta_extrapolated');
F=zeros(1,f);
clear beta_max beta_min
for k=1:f
    for EE=1:b
        E = ENERGYrange(1,EE);
        for j = 1:a
            js(j,EE) = n*exp(Fk(1,j)+beta_extrapolated(1,k)*E-betas(j,1)*E);
        end
    end
end
```

```
SUMjs = sum(js);
paronom = SUMjs.^(-1);
% for i = 1:b
%   energieies(:,i)=hist_matrix(:,i)*paronom(:,i);
% end
energieies = bsxfun(@times,hist_matrix,paronom);
SUMenergieies(k,1)= sum(sum(energieies));
F(1,k) = -log(SUMenergieies(k,1));
Z_M(k,1) = exp(-F(1,k));
end
%% QUANTITIES CALCULATION
M_s =zeros(f,1);
for k=1:f
    for EE=1:b
        E = ENERGYrange(1,EE);
        for j =1:a
            js(j,EE) = n*exp(Fk(1,j)+beta_extrapolated(1,k)*E-betas(j,1)*E);
        end
    end
    SUMjs = sum(js);
    paronom = SUMjs.^(-1);
    arithmiths = bsxfun(@times,hist_matrix,MAGNETISMrange');
    energieies = bsxfun(@times,arithmiths,paronom);
%   for i = 1:mag
%       arithmiths(i,:)=hist_matrix(i,:)*MAGNETISMrange(:,i);
%   end
%   for i = 1:b
%       energieies(:,i)=arithmiths(:,i)*paronom(:,i);
%   end
    SUMenergieies(k,1)= sum(sum(energieies));
    M_s(k,1)=SUMenergieies(k,1)/Z_M(k,1);
end

M_s2 =zeros(f,1);
for k=1:f
    for EE=1:b
        E = ENERGYrange(1,EE);
        for j =1:a
            js(j,EE) = n*exp(Fk(1,j)+beta_extrapolated(1,k)*E-betas(j,1)*E);
        end
    end
    SUMjs = sum(js);
    paronom = SUMjs.^(-1);
    arithmiths = bsxfun(@times,hist_matrix,MAGNETISMrange'.^2);
    energieies = bsxfun(@times,arithmiths,paronom);
%   for i = 1:mag
%       arithmiths(i,:)=hist_matrix(i,:)*(MAGNETISMrange(:,i))^2;
%   end
%   for i = 1:b
%       energieies(:,i)=arithmiths(:,i)*paronom(:,i);
%   end
```

```
SUMenergeies(k,1)= sum(sum(energeies));
M_s2(k,1)=SUMenergeies(k,1)/Z_M(k,1);
end
clear E EE energeies j js k N_E paronom SUMenergeies SUMjs NE i arithmiths
clear F Fk width1
%% PLOTS
figure
hold
plot(beta_extrapolated,U_s,'color','r')
measures = [data{1:end,3};data{1:end,4}];
plot(measures(1,:),measures(2:),'O','color','g')
NAMEtitle = strcat('Multi-Histogram Method - L = ',mat2str(L));
title(NAMEtitle)
xlabel('Inverse Temperature - betas')
ylabel('U = <E> = Average Energy')
hold
figure
hold
plot(beta_extrapolated,U_s2,'color','r')
measures = [data{1:end,3};data{1:end,5}];
plot(measures(1,:),measures(2:),'O','color','g')
NAMEtitle = strcat('Multi-Histogram Method - L = ',mat2str(L));
title(NAMEtitle)
xlabel('Inverse Temperature - betas')
ylabel('U = <E^2> = Average Square Energy')
hold
figure
hold
c=((beta_extrapolated).^2).*(U_s2-U_s.^2)/2;
plot(beta_extrapolated,c)
measures = [data{1:end,3};[data{1:end,6}]];
plot(measures(1,:),measures(2:),'O','color','g')
NAMEtitle = strcat('Specific Heat - Multi-Histogram Method - L = ',mat2str(L));
title(NAMEtitle)
xlabel('Inverse Temperature - betas')
ylabel(' c ')
hold
figure
hold
plot(beta_extrapolated,M_s,'color','r')
measures = [data{1:end,3};data{1:end,7}];
plot(measures(1,:),measures(2:),'O','color','g')
NAMEtitle = strcat('Multi-Histogram Method - L = ',mat2str(L));
title(NAMEtitle)
xlabel('Inverse Temperature - betas')
ylabel('<M> = Average Magnetism')
hold
figure
hold
plot(beta_extrapolated,M_s2,'color','r')
measures = [data{1:end,3};data{1:end,8}];
plot(measures(1,:),measures(2:),'O','color','g')
```

```
NAMEtitle = strcat('Multi-Histogram Method - L = ',mat2str(L));
title(NAMEtitle)
xlabel('Inverse Temperature - betas')
ylabel('<M^2> = Average Square Magnetism')
hold
figure
hold
x=(beta_extrapolated').*(M_s2-M_s.^2);
plot(beta_extrapolated,x)
measures = [data{1:end,3};data{1:end,9}];
plot(measures(1,:),measures(2,:),'O','color','g')
NAMEtitle = strcat('Magnetic Susceptibility - Multi-Histogram Method - L = ',mat2str(L));
title(NAMEtitle)
xlabel('Inverse Temperature - betas')
ylabel(' x ')
hold
clear b f measures n NAMEtitle edges mag t

Uperspin(fors,:) = U_s/(2*L^2);
U(fors,:) = U_s;
U2(fors,:) = U_s2;
Mperspin(fors,:) = M_s/(L^2);
M(fors,:) = M_s;
M2(fors,:) = M_s2;
cc(fors,:) = c;
xx(fors,:) = x;
end
%% Measure+ERRORS
U_(w,:)=mean(U,1);
U_perspin(w,:)=mean(Uperspin,1);
U2_(w,:)=mean(U2,1);
M_(w,:)=mean(M,1);
M_perspin(w,:)=mean(Mperspin,1);
M2_(w,:)=mean(M2,1);
xx_(w,:)=mean(xx,1);
cc_(w,:)=mean(cc,1);
semU(w,:)=std(U)/sqrt(10);
semUperspin(w,:)=std(Uperspin)/sqrt(10);
semU2(w,:)=std(U2)/sqrt(10);
semM2(w,:)=std(M2)/sqrt(10);
semM(w,:)=std(M)/sqrt(10);
semMperspin(w,:)=std(Mperspin)/sqrt(10);
semxx(w,:)=std(xx)/sqrt(10);
semcc(w,:)=std(cc)/sqrt(10);
end
%% Maximum quantities calculation
for n=1:iii;
A = find(cc_(n,:)==max(cc_(n,:)))
C(1,n)=cc_(n,A)
C(2,n)=semcc(n,A)
C(3,n)=beta_extrapolated(A)
```

```
B = find(xx_(n,:)==max(xx_(n,:)))  
C(4,n)=xx_(n,B)  
C(5,n)=semxx(n,B)  
C(6,n)=beta_extrapolated(B)  
end  
clear data startQ endQ ENERGYcounts ENERGYrange iii fors hist_matrix ...  
MAGNETISMcounts MAGNETISMrange Q w U_s U_s2 Z_E Z_M Zk_E Zk_M ...  
M_s2 M_s L Uperspin Mperspin U U2 M M2 c cc x xx cutoff data2 Extrapolation_step ...  
n s A a B
```


Κεφάλαιο 10^ο

Πίνακες μετρούμενων μεγεθών κατευθείαν από τις προσομοιώσεις και υπολογισμένων μεγεθών με τη μέθοδο Multi – Histogram για 10^6 μετρήσεις με τον αλγόριθμο Wolff.

	beta	<E>		<E ² >		c	
		calculated	measured	calculated	measured	calculated	measured
L=16	0.41	-307,95	-307,92	96775,31	96748,48	163,24	162,57
	0.42	-328,37	-328,27	109947,60	109894,88	187,20	188,21
	0.43	-349,87	-349,99	124555,06	124634,89	198,44	197,78
	0.44	-370,68	-370,71	139396,70	139416,22	192,65	192,49
	0.45	-389,31	-389,28	153289,97	153264,92	174,76	174,46
	0.46	-405,16	-405,19	165604,92	165633,01	153,17	153,57
	0.47	-418,40	-418,37	176261,57	176239,05	133,29	133,19

	beta	<E>		<E ² >		c	
		calculated	measured	calculated	measured	calculated	measured
L=32	0.41	-1194,26	-1194,00	1432903,03	1432265,34	558,73	557,22
	0.42	-1267,89	-1268,08	1615755,15	1616210,55	724,07	721,68
	0.43	-1359,62	-1359,50	1858581,79	1858222,21	926,53	923,71
	0.44	-1461,34	-1461,40	2145371,23	2145533,38	953,63	953,03
	0.45	-1550,25	-1550,29	2411118,76	2411214,48	793,70	791,23
	0.46	-1619,18	-1619,17	2627814,78	2627743,25	641,04	638,39
	0.47	-1673,46	-1673,57	2805347,26	2805703,69	538,58	538,15

	beta	<E>		<E ² >		c	
		calculated	measured	calculated	measured	calculated	measured
L=40	0.41	-1862,52	-1862,93	3478810,28	3480371,15	826,54	829,79
	0.42	-1970,67	-1969,80	3895620,30	3892194,26	1064,72	1066,23
	0.43	-2109,79	-2110,19	4467036,00	4468624,62	1464,40	1452,19
	0.44	-2276,35	-2276,33	5198125,85	5198081,52	1583,85	1587,90
	0.45	-2421,32	-2421,13	5875294,08	5874316,90	1264,63	1258,64
	0.46	-2529,88	-2530,02	6409765,53	6410396,85	1002,56	995,44
	0.47	-2614,67	-2614,80	6844098,78	6844776,28	840,84	841,02

	beta	<E>		<E ² >		c	
		calculated	measured	calculated	measured	calculated	measured
L=60	0.41	-4185,29	-4184,48	17538304,10	17531350,97	1819,38	1805,93
	0.42	-4416,17	-4416,74	19527581,08	19532367,53	2209,04	2183,99
	0.43	-4705,47	-4705,39	22175733,44	22174849,16	3170,81	3157,82
	0.44	-5101,02	-5101,18	26061107,64	26062565,99	3942,98	3926,07
	0.45	-5447,19	-5447,14	29700408,48	29699695,29	2887,54	2873,25
	0.46	-5692,15	-5692,32	32421882,27	32423709,68	2253,45	2247,85

	0.47	-5882,88	-5882,93	34625465,63	34625955,82	1892,28	1891,24
		<M>		<M ² >		x	
L=16	beta	calculated	measured	calculated	measured	calculated	measured
	0.41	118,36	118,27	17690,06	17667,57	1509,66	1508,92
	0.42	139,69	139,61	23192,04	23174,92	1545,11	1546,75
	0.43	161,63	161,76	29338,60	29370,42	1381,71	1377,25
	0.44	181,47	181,44	35370,41	35360,65	1072,69	1074,22
	0.45	197,47	197,49	40643,94	40649,49	742,96	741,51
	0.46	209,44	209,47	44912,46	44923,30	482,13	481,09
	0.47	218,17	218,18	48253,26	48254,56	308,51	305,95

		<M>		<M ² >		x	
L=32	beta	calculated	measured	calculated	measured	calculated	measured
	0.41	263,70	263,29	100249,92	99939,41	12592,62	12552,33
	0.42	359,55	359,87	172909,99	173151,70	18326,27	18330,10
	0.43	504,09	504,13	303236,08	303236,91	21125,20	21106,77
	0.44	660,82	660,79	470400,24	470374,56	14838,29	14842,32
	0.45	771,12	771,17	609483,37	609551,39	6682,38	6684,67
	0.46	834,16	834,08	701861,73	701740,29	2779,16	2781,19
	0.47	872,39	872,48	763987,20	764131,52	1373,52	1368,18

		<M>		<M ² >		x	
L=40	beta	calculated	measured	calculated	measured	calculated	measured
	0.41	330,12	330,91	161793,83	162482,01	21654,28	21721,45
	0.42	462,21	460,62	298255,40	296737,09	35536,29	35519,31
	0.43	700,29	701,07	604228,41	605019,10	48944,34	48816,27
	0.44	1000,02	999,95	1078915,83	1078811,38	34704,39	34722,93
	0.45	1201,08	1200,81	1470370,17	1469718,17	12497,16	12498,75
	0.46	1303,31	1303,60	1708259,03	1708840,20	4439,62	4352,78
	0.47	1363,08	1363,16	1862527,70	1862748,16	2133,46	2137,87

		<M>		<M ² >		x	
L=60	beta	calculated	measured	calculated	measured	calculated	measured
	0.41	489,34	489,03	367607,21	367034,48	52539,40	52431,01
	0.42	697,92	698,61	723200,26	723883,19	99160,29	99050,26
	0.43	1190,81	1190,38	1881777,52	1881162,47	199407,60	199583,88
	0.44	2119,00	2118,60	4861113,62	4859763,52	163212,26	163365,48
	0.45	2697,43	2697,66	7351890,86	7352602,04	34095,86	33850,25
	0.46	2932,61	2932,81	8621796,27	8622854,73	9927,46	9880,96
	0.47	3066,83	3066,84	9415606,75	9415681,55	4776,54	4788,48

	beta	$\langle M \rangle$		$\langle M^2 \rangle$		x	
		calculated	measured	calculated	measured	calculated	measured
L=80	0.43	1647,24	1646,17	3808960,77	3802889,83	471045,65	469990,24
	0.44	3599,68	3601,15	14074638,85	14084297,80	491437,44	491035,37
	0.45	4795,40	4794,99	23135206,94	23131618,93	62702,32	62866,28

	beta	$\langle M \rangle$		$\langle M^2 \rangle$		x	
		calculated	measured	calculated	measured	calculated	measured
L=100	0.43	2078,74	2079,02	6285407,79	6288210,39	844613,89	845334,25
	0.44	5413,32	5413,24	31948530,28	31948564,60	1163537,49	1163966,29
	0.45	7492,24	7492,04	56353224,77	56350387,86	98816,05	98904,77

	beta	$\langle E \rangle$		$\langle E^2 \rangle$		c	
		calculated	measured	calculated	measured	calculated	measured
L=80	0.43	-8339,44	-8339,08	69603713,19	69597444,30	5308,66	5286,90
	0.44	-9048,42	-9048,82	81951260,91	81958349,55	7486,96	7467,55
	0.45	-9683,49	-9683,45	93820624,59	93819804,85	5132,40	5132,57

	beta	$\langle E \rangle$		$\langle E^2 \rangle$		c	
		calculated	measured	calculated	measured	calculated	measured
L=100	0.43	-13015,59	-13015,56	169491231,87	169490461,05	7920,54	7918,78
	0.44	-14117,89	-14117,84	199442013,16	199440471,68	12298,52	12289,82
	0.45	-15130,27	-15130,35	229004202,99	229006515,28	8015,32	8008,47

	beta	<E>		<E ² >		c	
		calculated	measured	calculated	measured	calculated	measured
L=150	0.43	-29259,02	-29258,93	856277291	856271787,1	17293,74	17296,311
	0.44	-31705,07	-31705,14	1005519072	1005522960	29747,416	29720,185
	0.45	-34043,7	-34043,73	1159151367	1159152983	17975,461	17969,324

	beta	<E>		<E ² >		c	
		calculated	measured	calculated	measured	calculated	measured
L=200	0.435	-53820,43	-53820,06	2897036403	2896996276	37619,643	37584,795
	0.44	-56298,85	-56298,68	3170136037	3170114056	55685,382	55447,364
	0.445	-58752,01	-58752,56	3452195004	3452257112	39186,947	39048,963

	beta	<E>		<E ² >		c	
		calculated	measured	calculated	measured	calculated	measured
L=300	0.439	-125124,5	-125121,8	15658778696	15658089890	253159,79	253391,05
	0.44	-126507	-126504,1	16005408015	16004668073	135053,36	134684,92
	0.441	-127884,8	-127890,5	16355861295	16357290042	129256,74	128444

	beta	<E>		<E ² >		c	
		calculated	measured	calculated	measured	calculated	measured
L=400	0.439	-221756	-221744,6	49244538996	49239456177	6325122,4	6626870,4
	0.44	-224750,1	-224748,8	50515188352	50514521082	248999,73	243484,81
	0.441	-227297,6	-227310,3	51666619595	51672369684	236334,32	232155,72

Πίνακες απόκλισης επί τοις εκατό (%) μεγεθών κατευθείαν από τις προσομοιώσεις και υπολογισμένων μεγεθών με τη μέθοδο Multi – Histogram για 10⁶ μετρήσεις με τον αλγόριθμο Wolff.

		<E>	<E ² >	<M>	<M ² >	c	x
L=16	beta	calculated	calculated	calculated	calculated	calculated	calculated
	0.41	0,01%	0,03%	0,07%	0,13%	0,41%	0,05%
	0.42	0,03%	0,05%	0,05%	0,07%	0,54%	0,11%
	0.43	0,04%	0,06%	0,08%	0,11%	0,33%	0,32%
	0.44	0,01%	0,01%	0,02%	0,03%	0,09%	0,14%
	0.45	0,01%	0,02%	0,01%	0,01%	0,17%	0,20%
	0.46	0,01%	0,02%	0,01%	0,02%	0,27%	0,22%
	0.47	0,01%	0,01%	0,01%	0,00%	0,08%	0,84%

		<E>	<E ² >	<M>	<M ² >	c	x
L=32	beta	calculated	calculated	calculated	calculated	calculated	calculated
	0.41	0,02%	0,04%	0,15%	0,31%	0,27%	0,32%
	0.42	0,02%	0,03%	0,09%	0,14%	0,33%	0,02%
	0.43	0,01%	0,02%	0,01%	0,00%	0,31%	0,09%
	0.44	0,00%	0,01%	0,00%	0,01%	0,06%	0,03%
	0.45	0,00%	0,00%	0,01%	0,01%	0,31%	0,03%
	0.46	0,00%	0,00%	0,01%	0,02%	0,41%	0,07%
	0.47	0,01%	0,01%	0,01%	0,02%	0,08%	0,39%

		<E>	<E ² >	<M>	<M ² >	c	x
L=40	beta	calculated	calculated	calculated	calculated	calculated	calculated
	0.41	0,02%	0,04%	0,24%	0,42%	0,39%	0,31%
	0.42	0,04%	0,09%	0,35%	0,51%	0,14%	0,05%
	0.43	0,02%	0,04%	0,11%	0,13%	0,84%	0,26%
	0.44	0,00%	0,00%	0,01%	0,01%	0,25%	0,05%
	0.45	0,01%	0,02%	0,02%	0,04%	0,48%	0,01%
	0.46	0,01%	0,01%	0,02%	0,03%	0,72%	2,00%
	0.47	0,00%	0,01%	0,01%	0,01%	0,02%	0,21%

		<E>	<E ² >	<M>	<M ² >	c	x
L=60	beta	calculated	calculated	calculated	calculated	calculated	calculated
	0.41	0,02%	0,04%	0,06%	0,16%	0,74%	0,21%
	0.42	0,01%	0,02%	0,10%	0,09%	1,15%	0,11%
	0.43	0,00%	0,00%	0,04%	0,03%	0,41%	0,09%
	0.44	0,00%	0,01%	0,02%	0,03%	0,43%	0,09%
	0.45	0,00%	0,00%	0,01%	0,01%	0,50%	0,73%
	0.46	0,00%	0,01%	0,01%	0,01%	0,25%	0,47%
	0.47	0,00%	0,00%	0,00%	0,00%	0,06%	0,25%

		$\langle E \rangle$	$\langle E^2 \rangle$	$\langle M \rangle$	$\langle M^2 \rangle$	c	x
L=80	beta	calculated	calculated	calculated	calculated	calculated	calculated
	0.43	0,00%	0,01%	0,06%	0,16%	0,41%	0,22%
	0.44	0,00%	0,01%	0,04%	0,07%	0,26%	0,08%
	0.45	0,00%	0,00%	0,01%	0,02%	0,00%	0,26%

		$\langle E \rangle$	$\langle E^2 \rangle$	$\langle M \rangle$	$\langle M^2 \rangle$	c	x
L=100	beta	calculated	calculated	calculated	calculated	calculated	calculated
	0.43	0,00%	0,00%	0,01%	0,04%	0,02%	0,09%
	0.44	0,00%	0,00%	0,00%	0,00%	0,07%	0,04%
	0.45	0,00%	0,00%	0,00%	0,01%	0,09%	0,09%

		$\langle E \rangle$	$\langle E^2 \rangle$	c
L=150	beta	calculated	calculated	calculated
	0.43	0,00%	0,00%	0,01%
	0.44	0,00%	0,00%	0,09%
	0.45	0,00%	0,00%	0,03%

		$\langle E \rangle$	$\langle E^2 \rangle$	c
L=200	beta	calculated	calculated	calculated
	0.435	0,00%	0,00%	0,09%
	0.44	0,00%	0,00%	0,43%
	0.445	0,00%	0,00%	0,35%

		$\langle E \rangle$	$\langle E^2 \rangle$	c
L=300	beta	calculated	calculated	calculated
	0.439	0,00%	0,00%	0,09%
	0.44	0,00%	0,00%	0,27%
	0.441	0,00%	0,01%	0,63%

		$\langle E \rangle$	$\langle E^2 \rangle$	c
L=400	beta	calculated	calculated	calculated
	0.439	0,01%	0,01%	4,55%
	0.44	0,00%	0,00%	2,26%
	0.441	0,01%	0,01%	1,80%

Βιβλιογραφία:

1. M.E.J. Newman & G.T. Barkema «Monte Carlo Methods in Statistical Physics» Oxford University Press June 1998
2. Konstantinos N. Anagnostopoulos «Computational Physics» published by Konstantinos N. Anagnostopoulos 2014
3. R. K. Pathria & Paul D. Beale «Statistical Mechanics» Thrid Edition, 2011 Elsevier Ltd. All rights reserved
4. K. Huang, Statistical Mechanics, John Wiley & Sons, New York, ed. 1987. A detailed presentation of the Osanger Solution