

# Υπολογιστική Φυσική II

Μια Εισαγωγή στις Προσομοιώσεις Monte Carlo  
στη Στατιστική Φυσική

Για το μάθημα Υπολογιστικής Φυσικής II  
8ο εξάμηνο Σχολής ΕΜΦΕ  
Εθνικό Μετσόβιο Πολυτεχνείο

Κωνσταντίνος Ν. Αναγνωστόπουλος

25 Φεβρουαρίου 2011



# ΠΕΡΙΕΧΟΜΕΝΑ

<b>I</b>	<b>ΤΑ ΘΕΜΕΛΙΑ</b>	<b>1</b>
<b>1</b>	<b>ΣΤΑΤΙΣΤΙΚΗ ΦΥΣΙΚΗ</b>	<b>3</b>
1.1	ΕΙΣΑΓΩΓΗ . . . . .	3
1.2	ΕΝΤΡΟΠΙΑ . . . . .	6
1.3	ΔΙΑΚΥΜΑΝΣΕΙΣ . . . . .	10
1.4	ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΣΧΕΤΙΣΜΟΥ . . . . .	13
1.5	ΠΡΟΤΥΠΟ ISING . . . . .	15
1.6	ΠΡΟΒΛΗΜΑΤΑ . . . . .	21
<b>2</b>	<b>ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΜΟΝΤΕ CARLO</b>	<b>23</b>
2.1	ΔΕΙΓΜΑΤΟΛΗΨΙΑ . . . . .	24
2.1.1	ΑΠΛΗ ΔΕΙΓΜΑΤΟΛΗΨΙΑ . . . . .	24
2.1.2	IMPORTANCE SAMPLING . . . . .	25
2.2	ΔΙΑΔΙΚΑΣΙΕΣ ΜΑΡΚΟΝ . . . . .	26
2.3	ΣΥΝΘΗΚΗ ΛΕΠΤΟΜΕΡΟΥΣ ΙΣΟΖΥΓΗΣΗΣ . . . . .	28
2.4	ΠΡΟΒΛΗΜΑΤΑ . . . . .	30
<b>3</b>	<b>Ο ΤΥΧΑΙΟΣ ΠΕΡΙΠΑΤΗΤΗΣ</b>	<b>31</b>
3.1	ΤΥΧΑΙΕΣ ΔΙΑΔΡΟΜΕΣ . . . . .	32
3.2	(ΨΕΥΔΟ)ΤΥΧΑΙΟΙ ΑΡΙΘΜΟΙ . . . . .	35
3.2.1	ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ . . . . .	37
3.2.2	ΑΠΟΤΕΛΕΣΜΑΤΑ . . . . .	46
3.3	ΜΕΛΕΤΗ ΤΟΥ ΤΥΧΑΙΟΥ ΠΕΡΙΠΑΤΗΤΗ . . . . .	54
3.3.1	Η ΓΕΩΜΕΤΡΙΑ ΤΩΝ ΤΥΧΑΙΩΝ ΔΙΑΔΡΟΜΩΝ . . . . .	61
3.3.2	ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ . . . . .	66
3.3.3	ΠΑΡΑΛΛΑΓΕΣ ΤΟΥ ΜΟΝΤΕΛΟΥ . . . . .	80
3.4	ΠΡΟΒΛΗΜΑΤΑ . . . . .	84
<b>4</b>	<b>ΠΡΟΣΟΜΟΙΩΣΗ ΤΟΥ ΠΡΟΤΥΠΟΥ ISING</b>	<b>89</b>
4.1	ΑΛΓΟΡΙΘΜΟΣ ΜΕΤΡΟΠΟΛΙΣ . . . . .	89
4.2	ΣΧΕΔΙΑΣΜΟΣ ΚΩΔΙΚΑ . . . . .	93
4.2.1	Ο ΚΩΔΙΚΑΣ . . . . .	96
4.2.2	ΒΕΛΤΙΩΣΗ ΤΟΥ INTERFACE . . . . .	103

4.3	ΕΝ ΑΡΧΗ ΗΝ ΨΥΧΟΣ (Η ... ΖΕΣΤΗ)	116
4.4	ΤΟ ΖΗΤΟΥΜΕΝΟ, Η ΑΝΕΞΑΡΤΗΣΙΑ	119
4.5	ΑΝΑΖΗΤΩΝΤΑΣ ΤΟ ... ΣΦΑΛΜΑ	123
4.5.1	ΣΦΑΛΜΑΤΑ ΑΝΕΞΑΡΤΗΤΩΝ ΜΕΤΡΗΣΕΩΝ	124
4.5.2	Ο ΣΟΥΓΙΑΣ ΕΙΝΑΙ ΧΡΗΣΙΜΟΣ	125
4.5.3	ΟΤΑΝ ΔΕΝΕΙΣ ΤΙΣ ΜΠΟΤΕΣ ΣΟΥ...	126
4.6	ΠΡΟΒΛΗΜΑΤΑ	127
<b>5</b>	<b>CLUSTER ALGORITHMS</b>	<b>139</b>
5.1	ΚΡΙΣΙΜΗ ΕΠΙΒΡΑΔΥΝΣΗ	139
5.2	Ο ΑΛΓΟΡΙΘΜΟΣ ΤΟΥ WOLFF	141
5.3	ΣΧΕΔΙΑΣΜΟΣ ΚΩΔΙΚΑ	148
5.3.1	Ο ΚΩΔΙΚΑΣ	150
5.4	ΣΥΛΛΟΓΗ ΔΕΔΟΜΕΝΩΝ (και shell scripting)	156
5.5	ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ	159
5.6	ΧΡΟΝΟΙ ΑΥΤΟΣΥΣΧΕΤΙΣΜΟΥ	163
5.7	ΠΡΟΒΛΗΜΑΤΑ	167
<b>6</b>	<b>ΚΡΙΣΙΜΟΙ ΕΚΘΕΤΕΣ</b>	<b>173</b>
6.1	ΕΙΣΑΓΩΓΗ	173
6.2	ΒΑΘΜΙΣΗ ΘΕΡΜΟΚΡΑΣΙΑΣ	177
6.3	ΒΑΘΜΙΣΗ ΠΕΠΕΡΑΣΜΕΝΟΥ ΜΕΓΕΘΟΥΣ	182
6.4	ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΗΣ $\beta_c$	185
6.5	ΜΕΛΕΤΗ ΒΑΘΜΙΣΗΣ ΜΕ ... ΚΑΤΑΡΡΕΥΣΗ	190
6.6	BINDER CUMULANT ΚΑΙ ... ΟΧΙ ΜΟΝΟ	198
6.7	ΠΡΟΒΛΗΜΑΤΑ	202
<b>II</b>	<b>ΠΑΡΑΡΤΗΜΑΤΑ</b>	<b>207</b>
<b>A'</b>	<b>ΣΥΝΑΡΤΗΣΗ ΚΑΙ ΧΡΟΝΟΣ ΑΥΤΟΣΥΣΧΕΤΙΣΜΟΥ</b>	<b>209</b>
<b>B'</b>	<b>ΥΠΟΛΟΓΙΣΜΟΣ ΣΦΑΛΜΑΤΩΝ</b>	<b>217</b>
B'.1	ΜΕΘΟΔΟΣ JACKKNIFE	217
B'.2	ΜΕΘΟΔΟΣ BOOTSTRAP	221
B'.3	Η ΣΥΓΚΡΙΣΗ ΤΩΝ ΜΕΘΟΔΩΝ	225
B'.4	ΠΡΟΒΛΗΜΑΤΑ	229
<b>Γ'</b>	<b>ΒΑΘΜΙΣΗ ΚΑΙ ΚΡΙΣΙΜΟΙ ΕΚΘΕΤΕΣ</b>	<b>235</b>
Γ'.1	BINDER CUMULANT	235
Γ'.2	ΒΑΘΜΙΣΗ	238
Γ'.3	ΒΑΘΜΙΣΗ ΠΕΠΕΡΑΣΜΕΝΟΥ ΜΕΓΕΘΟΥΣ	240
Γ'.4	ΠΡΟΒΛΗΜΑΤΑ	243

<b>Δ' ΚΡΙΣΙΜΟΙ ΕΚΘΕΤΕΣ</b>	<b>245</b>
Δ'.1 ΟΡΙΣΜΟΙ . . . . .	245
Δ'.2 ΣΧΕΣΕΙΣ . . . . .	245
Δ'.3 ΠΡΟΒΛΗΜΑΤΑ . . . . .	246



# ΒΙΒΛΙΟΓΡΑΦΙΑ

## [Βασικά συγγράματα για το μάθημα]

- [1] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics*, Clarendon Press, Oxford. Πολύ καλό βιβλίο στο σωστό επίπεδο για το μάθημα (Κεφ. 1-4).
- [2] B. A. Berg, *Markov Chain Monte Carlo Simulations and Their Statistical Analysis. With Web-Based Fortran Code*, World Scientific, 2004. Σύγχρονο, λεπτομερές και με προγράμματα σε γλώσσα Fortran. Καλύπτει σύγχρονες μεθόδους ανάλυσης όπως multihistogramming, multicanonical simulations κ.α. Το συστήνω στους πιο απαιτητικούς φοιτητές.
- [3] D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, 3rd Edition, 2009. Εξαιρετικό βιβλίο, εκτενές και πλήρες, καλύπτει πάρα πολλά θέματα στην προσομοίωση συστημάτων στη στατιστική φυσική.
- [4] K. Binder, D.W. Heermann, *Monte Carlo Simulation in Statistical Physics*, Fourth Edition, Springer. Εξάίρετο βιβλίο για σύντομη και περιεκτική εισαγωγή σε Μόντε Κάρλο συμπληρωματικό του [1]. Περιέχει ωραία εισαγωγική κάλυψη του θέματος του τυχαίου περπατητή.
- [5] H. Gould, J. Tobochnik και H. Christian, *Computer Simulation Methods, Application to Physical Systems*, Third Edition, Addison Wesley. Πλήρες βιβλίο υπολογιστικής φυσικής με ειδικά κεφάλαια σε προσομοιώσεις Monte Carlo. Το λογισμικό των προσομοιώσεων είναι γραμμένο σε Java και προσφέρεται για εφαρμογές με εικονικές και διαδραστικές δυνατότητες. Διατίθεται ελεύθερα στο [opensourcephysics.org](http://opensourcephysics.org)

## [Κεφάλαιο 1]

- [6] L. Kadanoff, *Statistical Physics – Statics, Dynamics and Renormalization*, World Scientific. Εξάριετο βιβλίο στατιστικής φυσικής από τον απόλυτα ειδικό στο πεδίο.
- [7] J. J. Binney, N. J. Dowrick, A. J. Fisher and M. E. J. Newman, *The Theory of Critical Phenomena*, Oxford. Παιδαγωγική εισαγωγή στην θεωρία των κρίσιμων φαινομένων και της ομάδας επανακανονικοποίησης.
- [8] R. K. Pathria, *Statistical Mechanics*, Pergamon. Κλασικό για μεταπτυχιακού επιπέδου μάθημα στατιστικής φυσικής.
- [9] F. Mandl, *Στατιστική Φυσική*, Γ. Α. Πνευματικού. Απλή εισαγωγή σε προπτυχιακό επίπεδο.
- [10] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*. Ανάλυση των (λίγων) προτύπων στατιστικής φυσικής που επιδέχονται ακριβούς αναλυτικής λύσης.
- [11] E. Ising, *Beitrag zur Theorie des Ferromagnetismus*, *Z. Phys.* **31** (1925) 253–258. Το πρότυπο Ising στην πρώτη του διατύπωση με τη λύση του μονοδιάστατου προβλήματος.
- [12] L. Onsager, *Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition*, *Phys. Rev.* **65** (1944) 117–119. Η αναλυτική λύση του διδιάστατου πρότυπου Ising, η λύση Onsager.
- [13] K. Huang, *Statistical Mechanics*, John Wiley & Sons, New York, ed. 1987. Λεπτομερής ανάλυση της λύσης Onsager.

### [Κεφάλαιο 3]

- [14] J. Ambjørn, B. Durhuus, T. Jonsson, *Quantum Geometry*, Cambridge Monographs on Mathematical Physics, Cambridge University Press. Παρουσίαση του τυχαίου περιπατητή από γεωμετρική σκοπιά.
- [15] C. Itzykson, J.M. Drouffe, *Statistical Field Theory*, Volume 1, Cambridge Monographs on Mathematical Physics, Cambridge University Press. Παρουσίαση του τυχαίου περιπατητή από τη σκοπιά της Ευκλείδειας Θεωρίας Πεδίου
- [16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes, The Art of Scientific Computing*, Cambridge University Press. Το βιβλίο είναι ελεύθερα διαθέσιμο στην ιστοσελίδα [www.nr.com](http://www.nr.com).



- [17] D. E. Knuth, *Seminumerical Algorithms*, Vol. 2 of *The Art of Computer Programming*, Addison-Wesley, 1981.
- [18] M. Lüscher, *Comput. Phys. Commun.* **79** (1994) 100; F. James, *Comput. Phys. Commun.* **79** (1994) 111; *Erratum* **97** (1996) 357; The Scalable Parallel Random Number Generators Library (SPRNG) for ASCII Monte Carlo Computations at Florida State University, <http://sprng.cs.fsu.edu/>. Επί πλέον αναφορές για γεννήτριες τυχαίων αριθμών.

#### [Κεφάλαιο 4]

- [19] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E., *J. Chem. Phys.* **21** (1953) 1087.
- [20] Nightingale, M. P. and Blöte, H. W. J., *Phys. Rev. Lett.* **76** (1996) 4548.
- [21] Müller-Krumbhaar, H and Binder, K., *J. Stat. Phys.* **8** (1973) 1.
- [22] Efron, Bradley *SIAM Review* **21** (1979) 460; *Ann. Statist.* **7** (1979) 1; Efron, B and Tibshirani, R. *Statistical Science* **1** (1986) 54. Διαθέσιμα ελεύθερα από το [projecteuclid.org](http://projecteuclid.org).

#### [Κεφάλαιο 5]

- [23] Swendsen, R. H. and Wang J.-S., *Phys. Rev. Lett.* **58** (1987) 86
- [24] U. Wolff, *Phys. Rev. Lett.* **62** (1989) 361
- [25] A. Pelisseto and E. Vicari, “*Critical Phenomena and Renormalization-Group Theory*”, *Phys. Reports* **368** (2002) 549.
- [26] F. Y. Wu, *The Potts Model*, *Rev. Mod. Phys.* **54** (1982) 235
- [27] Coddington, P.D. and Baillie, *Phys. Rev. Lett.* **68** (1992) 962

#### [Κεφάλαιο 6]

- [28] Rieger, H., *Phys. Rev. B* **52** (1995) 6659
- [29] Newman E. J. and Barkema, G. T. *Phys. Rev. E* **53** (1996) 393
- [30] A.E. Ferdinand and M.E. Fisher, *Phys. Rev.* **185** (1969) 832; N.Sh. Izmailian, C.-K. Hu, *Phys. Rev. E* **65** (2002) 036103; J. Salas, *J. Phys. A* **34** (2001) 1311; W. Janke and R. Kenna, *Nucl. Phys. (Proc. Suppl.)* **106** (2002) 929.

- [31] J. Ambjørn, K.N. Anagnostopoulos, *Nucl. Phys.* **B 497** (1997) 445.
- [32] K. Binder, *Phys. Rev. Lett.* **47** (1981) 693.
- [33] K. Binder, *Z. Phys.* **B 43** (1981) 119; G. Kamieniarz and H.W.J. Blöte, *J. Phys. A* **26** (1993) 201.
- [34] J. Cardy, *Scaling and Renormalization in Statistical Physics*, Cambridge University Press, 1st Edition, 1996.
- [35] A. M. Ferrenberg and D. P. Landau, *Phys. Rev.* **B44** (1991) 5081.
- [36] M. S. S. Challa, D. P. Landau and K. Binder, *Phys. Rev.* **B34** (1986) 1841.

### [Παράρτημα Δ']

- [37] H.E. Stanley, *"Introduction to Phase Transitions and Critical Phenomena"*, Oxford 1971.
- [38] R. Creswick and S.-Y. Kim, *J. Phys. A: Math.Gen.* **30** (1997) 8785
- [39] C. Holm and W. Janke, *Phys. Rev.* **B 48** (1993) 932, [arXiv:hep-lat/9301002].
- [40] M. Hasenbusch and S. Meyer, *Phys. Lett.* **B 241** (1990) 238
- [41] M. Kolesik and M. Suzuki arXiv:cond-mat/9411109
- [42] M. Kolesik and M. Suzuki, *Physica* **A 216** (1995) 469.

**Μέρος Ι**  
**ΤΑ ΘΕΜΕΛΙΑ**



# ΚΕΦΑΛΑΙΟ 1

## ΣΤΑΤΙΣΤΙΚΗ ΦΥΣΙΚΗ

Στο κεφάλαιο αυτό γίνεται μια επισκόπηση βασικών εννοιών της στατιστικής φυσικής. Στόχος είναι οι έννοιες αυτές να “φρεσκαριστούν” σε σύντομο χώρο στο κείμενο, οπότε δεν γίνεται προσπάθεια να είμαστε απόλυτα ακριβείς. Αν κάτι δεν είναι κατανοητό στο διάβασμα του κεφαλαίου, πρέπει ο φοιτητής να ανατρέξει στη βιβλιογραφία στατιστικής φυσικής. Ενδεικτικά, καλά βιβλία για το επιθυμητό επίπεδο γνώσεων αναφέρονται στην βιβλιογραφία των σημειώσεων [1]-[9].

### 1.1 ΕΙΣΑΓΩΓΗ

Η Στατιστική Φυσική έχει σαν σκοπό να περιγράψει συστήματα με πολύ μεγάλο αριθμό βαθμών ελευθερίας  $N$ . Απλά συστήματα έχουν τυπικά  $\approx 10^{23} - 10^{44}$ . Για τα συστήματα αυτά οι εξισώσεις που περιγράφουν μικροσκοπικά το σύστημα είναι πρακτικά αδύνατον και τελικά μάλλον άχρηστο να λυθούν. Αρκούν μερικά σωστά ορισμένες “χονδροειδείς ιδιότητες” (bulk properties) του συστήματος για να μας δώσουν τις χρήσιμες φυσικές πληροφορίες για το σύστημα. Λ.χ. σε έναν μαγνήτη πολλές φορές μας αρκεί να γνωρίζουμε την εσωτερική ενέργεια και μαγνήτιση του υλικού, σε ένα ρευστό την ενέργεια και πυκνότητά του κ.ο.κ. και όχι αναλυτικά την θέση, ενέργεια, ορμή και στροφορμή κάθε σωματιδίου που τα αποτελούν. Αυτά είναι γνωστά από τη θερμοδυναμική, στη στατιστική φυσική όμως γίνεται η απόπειρα να παραχθούν οι παραπάνω ιδιότητες των συστημάτων από τις μικροσκοπικές τους ιδιότητες, δηλ. κινηματική - βαθμοί ελευθερίας και τη δυναμική - Hamiltonian του συστήματος.

Στην περίπτωση μας θα κάνουμε τις, όχι ιδιαίτερα περιοριστικές, υποθέσεις ότι το σύστημά μας περιγράφεται από διακριτές καταστάσεις που μπορούν να απαριθμηθούν μέσα σε ένα σύνολο

$\{\mu\}$  με αντίστοιχες ενέργειες  $0 < \epsilon_1 < \dots < \epsilon_\mu < \dots$ . Το σύστημα αυτό είναι σε επαφή με μεγάλη δεξαμενή θερμότητας θερμοκρασίας  $\beta = 1/kT$  με το οποίο μπορεί να αλληλεπιδρά. Η επαφή με τη δεξαμενή έχει σαν αποτέλεσμα να υπάρχουν τυχαίες μεταβάσεις του συστήματος με τρόπο που προσδιορίζεται από τη δυναμική του συστήματος. Η θεμελιώδεις ποσότητες που μας ενδιαφέρουν είναι τα βάρη (weights)  $w_\mu(t)$  που μας δίνουν την πιθανότητα να είναι το σύστημα στην κατάσταση  $\mu$  τη χρονική στιγμή  $t$ . Αυτές κωδικοποιούν την μικροσκοπική φυσική στη στατιστική φυσική.

Έστω ότι  $R(\mu \rightarrow \nu)$  δίνουν το ρυθμό μετάβασης από την κατάσταση  $\mu \rightarrow \nu$ , δηλ.

$$R(\mu \rightarrow \nu)dt = \text{Πιθανότητα μετάβασης } \mu \rightarrow \nu \text{ στο χρόνο } dt \quad (1.1)$$

Τότε μπορούμε να γράψουμε την πολύ γενική “δεσπόζουσα εξίσωση” (Master Equation):

$$\frac{dw_\mu(t)}{dt} = \sum_\nu \{w_\nu(t)R(\nu \rightarrow \mu) - w_\mu(t)R(\mu \rightarrow \nu)\} \quad (1.2)$$

$$\sum_\mu w_\mu(t) = 1. \quad (1.3)$$

Η πρώτη από τις παραπάνω εξισώσεις μας λέει απλά ότι η μεταβολή του βάρους  $w_\mu(t)$  είναι ίση με το ρυθμό που το σύστημα εισέρχεται στην κατάσταση  $\mu$  από οποιαδήποτε άλλη  $\nu$  μείον το ρυθμό με τον οποίο φεύγει από την κατάσταση  $\mu$ . Η δεύτερη εκφράζει ότι τα βάρη  $w_\mu(t)$  ερμηνεύονται ως πιθανότητα και φυσικά η πιθανότητα να είναι το σύστημα σε κάποια κατάσταση είναι ίση με 1.

Οι ρυθμοί μετάβασης  $R(\mu \rightarrow \nu)$  προκύπτουν από τη θερμική φύση της αλληλεπίδρασης του συστήματος με τη θερμική δεξαμενή. Στην πράξη αυτοί προσομοιώνονται με κατάλληλες επιλογές κατά τη διάρκεια των υπολογισμών Μόντε Κάρλο. Τα  $R(\mu \rightarrow \nu)$  θεωρούνται ανεξάρτητα του χρόνου οπότε το παραπάνω σύστημα εξισώσεων για τα  $w_\mu(t)$  είναι γραμμικό, και ο περιορισμός  $0 \leq w_\mu(t) \leq 1$  οδηγεί στο (μη τετριμμένο) συμπέρασμα ότι σε άπειρο χρόνο τα  $w_\mu(t)$  συγκλίνουν γρήγορα (για μεγάλα συστήματα) σε αριθμούς  $p_\mu$ , τις πιθανότητες κατάληψης ισορροπίας. Δηλαδή μετά από κάποιο χρόνο

$$\frac{dw_\mu(t)}{dt} = 0 \quad (1.4)$$

$$p_\mu = \lim_{t \rightarrow \infty} w_\mu(t) \quad \sum_\mu p_\mu = 1. \quad (1.5)$$

$\beta^{-1}$	Χαρακτηριστική κλίμακα
1/40 eV	$\approx$ Θερμ. Δωματίου/Βιολογία
1-10 eV	$\approx$ Ατομική φυσική/Χημεία
1 MeV	$\approx$ Πυρηνική Φυσική
0.511 MeV	$\approx m_e$
1 GeV	$\approx m_p$
0.1-1 TeV	$\approx m_{\text{Weak}}/\text{Φυσ. Υψηλών Ενέργειών (LHC)}$
$10^{19}$ GeV	$\approx m_{\text{Planck}}/\text{Κβαντική Βαρύτητα}$

Πίνακας 1.1: Χαρακτηριστικές κλίμακες ενέργειας/θερμοκρασίας στη φύση

Οι πιθανότητες  $p_\mu$  για σύστημα σε ισορροπία με θερμική ισορροπία με δεξαμενή θερμοκρασίας  $\beta = 1/kT$ ,  $k = 1.38 \times 10^{-23} JK^{-1}$  μπορεί να δειχθεί (Gibbs 1902) ότι ακολουθούν την κατανομή Boltzman

$$p_\mu = \frac{1}{Z} e^{-\beta E_\mu}. \quad (1.6)$$

Η παράμετρος  $\beta$  θα αναφέρεται απλά ως η θερμοκρασία του συστήματος και βλέπουμε ότι μέσω του εκθετικού στην εξίσωση (1.6) καθορίζει μία χαρακτηριστική ενέργεια για το σύστημα (βλ. Πίνακα 1.1). Η μέτρησή της σε βαθμούς Kelvin κλπ οφείλεται στην αρχική άγνοια του ορισμού της στην θερμοδυναμική.

Η σταθερά  $Z$  στην (1.6) είναι η συνάρτηση επιμερισμού του συστήματος και είναι η σταθερά κανονικοποίησης της κατανομής  $p_\mu$ . Η σχέση  $\sum_\mu p_\mu = 1$  μας δίνει

$$Z(\beta) = \sum_\mu e^{-\beta E_\mu} \quad (1.7)$$

Η τιμή μιας φυσικής ποσότητας που μετριέται στο εργαστήριο έχει στοχαστικό χαρακτήρα. Για συστήματα με πολύ μεγάλο αριθμό βαθμών ελευθερίας  $N$  πρακτικά κανείς ενδιαφέρεται για τη μέση τιμή μιας ποσότητας μια και η πιθανότητα να μετρήσει μια τιμή που να διαφέρει σημαντικά από αυτή είναι αμελητέα. Σύμφωνα με τα παραπάνω η μέση τιμή  $\langle Q \rangle$  μιας φυσικής ποσότητας  $Q$  η οποία παίρνει την τιμή  $Q_\mu$  στην κατάσταση  $\mu$  θα είναι

$$\langle Q \rangle = \sum_\mu p_\mu Q_\mu = \frac{1}{Z} \sum_\mu Q_\mu e^{-\beta E_\mu}. \quad (1.8)$$

Όπως θα δούμε παρακάτω η τυπική απόκλιση  $\Delta Q$  για ένα τυπικό θερμοδυναμικό σύστημα είναι τέτοια ώστε

$$\frac{\Delta Q}{Q} \sim \frac{1}{\sqrt{N}} \quad (1.9)$$

ποσοστό το οποίο είναι αμελητέο για συνήθη μακροσκοπικά συστήματα. (π.χ. για  $N \sim 10^{23}$  έχουμε  $\Delta Q/Q \sim 10^{-11}$ ). Για το λόγο αυτό όταν το σύστημα είναι μεγάλο οι διακυμάνσεις μπορούν να αγνοηθούν. Το όριο  $N \rightarrow \infty$  ονομάζεται το **θερμοδυναμικό όριο** και το ενδιαφέρον μας συνήθως εστιάζεται στην συμπεριφορά του συστήματος στο όριο αυτό. Στην πράξη, ενώ τα συστήματα στο εργαστήριο είναι τις περισσότερες φορές πολύ κοντά στο όριο αυτό, στις προσομοιώσεις μας πολύ συχνά δεν είναι δυνατόν να μελετήσουμε αρκετά μεγάλα συστήματα. Η όλη τέχνη επικεντρώνεται στο σχεδιασμό αλγορίθμων προσομοίωσης και μεθόδων ανάλυσης ;έτσι ώστε να έχουμε εμπιστοσύνη ότι τα αποτελέσματά μας αντανακλούν τη συμπεριφορά του συστήματος στο θερμοδυναμικό όριο.

Η συνάρτηση επιμερισμού κωδικοποιεί λόγω του ορισμού (1.7) όλη τη στατιστική πληροφορία για το σύστημα, μια που δεν είναι μια απλή συνάρτηση μίας μεταβλητής (της  $\beta$ ) αλλά απαριθμεί με σχετικό βάρος τις δυνατές καταστάσεις του συστήματος. Απλό παράδειγμα αποτελεί ο υπολογισμός μέσω αυτής της μέσης ενέργειας  $\langle E \rangle$  (εσωτερικής ενέργειας  $U$  στη θερμοδυναμική) του συστήματος:

$$\begin{aligned} U \equiv \langle E \rangle &= \frac{1}{Z} \sum_{\mu} E_{\mu} e^{-\beta E_{\mu}} = -\frac{1}{Z} \sum_{\mu} \frac{\partial}{\partial \beta} e^{-\beta E_{\mu}} = -\frac{1}{Z} \frac{\partial}{\partial \beta} \sum_{\mu} e^{-\beta E_{\mu}} \\ &= -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = -\frac{\partial \ln Z}{\partial \beta} \end{aligned} \quad (1.10)$$

Με τον ίδιο τρόπο μπορεί κανείς να υπολογίσει την ειδική θερμότητα

$$C = \frac{\partial U}{\partial T} = \frac{\partial \beta}{\partial T} \frac{\partial U}{\partial \beta} = (-k\beta^2) \left( -\frac{\partial^2 \ln Z}{\partial \beta^2} \right) = k\beta^2 \frac{\partial^2 \ln Z}{\partial \beta^2} \quad (1.11)$$

## 1.2 ΕΝΤΡΟΠΙΑ

Από τη θερμοδυναμική ορίζουμε ότι η εντροπία  $S$  θερμοδυναμικού συστήματος ορίζεται από τη σχέση

$$S = -\frac{\partial F}{\partial T}, \quad F = U - TS, \quad (1.12)$$

όπου  $F$  Θα επιχειρήσουμε τώρα να δώσουμε ορισμούς που να είναι συνεπείς με τους παραπάνω και να προκύπτουν από τις μικροσκοπικές καταστάσεις του συστήματος.



Ορίζουμε την ελεύθερη ενέργεια από τη σχέση

$$e^{-\beta F} = Z \equiv \sum_{\mu} e^{-\beta E_{\mu}} \quad (1.13)$$

ή ισοδύναμα

$$F = -\frac{1}{\beta} \ln Z \quad (1.14)$$

Παρατηρήστε ότι ο παραπάνω ορισμός της  $F$  ταυτίζεται με την ενέργεια θεμελιώδους κατάστασης για  $T \rightarrow 0^1$ . Πράγματι τότε  $\beta \rightarrow \infty$  και ο μόνος όρος που επιζεί στο άθροισμα (1.13) είναι ο πρώτος. Για τον λόγο αυτό από την (1.12) θα έχουμε  $\lim_{T \rightarrow 0} S = 0$  που είναι ο δεύτερος νόμος της θερμοδυναμικής.

Ο ορισμός (1.13) είναι συνεπής με την (1.12) γιατί

$$U = -\frac{\partial \ln Z}{\partial \beta} = -\frac{\partial}{\partial \beta}(-\beta F) = F + \beta \frac{\partial F}{\partial \beta} = F - T \frac{\partial F}{\partial T} = F + TS. \quad (1.15)$$

Η σύνδεση της εντροπίας  $S$  με τη μικροφυσική γίνεται με τις σχέσεις (1.13) και (1.12):

$$\frac{S}{k} = \frac{U - F}{kT} = \beta(U - F) = \beta \left( \sum_{\mu} p_{\mu} E_{\mu} + \frac{1}{\beta} \ln Z \right). \quad (1.16)$$

Αλλά

$$p_{\mu} = \frac{e^{-\beta E_{\mu}}}{Z} \Rightarrow E_{\mu} = -\frac{1}{\beta} (\ln p_{\mu} + \ln Z), \quad (1.17)$$

οπότε

$$\begin{aligned} \frac{S}{k} &= \beta \sum_{\mu} \left( -\frac{1}{\beta} (\ln p_{\mu} + \ln Z) p_{\mu} + \frac{1}{\beta} \ln Z \right) \\ &= -\sum_{\mu} p_{\mu} \ln p_{\mu} - \ln Z \sum_{\mu} p_{\mu} + \ln Z \\ &= -\sum_{\mu} p_{\mu} \ln p_{\mu}. \end{aligned} \quad (1.18)$$

Τελικά

$$S = -k \sum_{\mu} p_{\mu} \ln p_{\mu} \quad (1.19)$$

Ας κάνουμε μία διερεύνηση του παραπάνω τύπου. Ας υποθέσουμε ότι ένα (ομολογουμένως ασυνήθιστο <sup>2</sup>) σύστημα όλες οι

<sup>1</sup>Δηλ. δεν υπάρχει αυθόρμητο σπάσιμο συμμετρίας που οδηγεί σε εκφυλισμό της ενέργειας θεμελιώδους κατάστασης

<sup>2</sup>π.χ. η διδιάστατη κβαντική βαρύτητα απουσία ύλης

καταστάσεις έχουν την ίδια ενέργεια. Στην περίπτωση αυτή με απλή αντικατάσταση στην εξίσωση (1.19) παίρνουμε ότι

$$p_\mu = \frac{1}{g} = \text{σταθ.} \Rightarrow S = k \ln g. \quad (1.20)$$

Δηλαδή η εντροπία μετράει τον αριθμό των καταστάσεων του συστήματος, όπως ακριβώς και στην περίπτωση της μικροκανονικής συλλογής. Πράγματι η τελευταία σχέση προκύπτει και για την κατανομή

$$p_\mu = \begin{cases} \frac{1}{g(E)} & E_\mu = E \\ 0 & E_\mu \neq E \end{cases} \quad (1.21)$$

που μπορεί να θεωρηθεί πως δίνει την μικροκανονική συλλογή αφού βάζει τον περιορισμό  $E_\mu = E = \text{σταθ.}$ . Η συνάρτηση  $g(E)$  (σε πολλά βιβλία συμβολίζεται με  $\Omega(E)$ ) μετράει τον αριθμό των καταστάσεων με ενέργεια ίση με  $E$ . Από αυτή προκύπτει η πιθανότητα  $p(E)$  το σύστημα να βρεθεί να έχει ενέργεια  $E$

$$p(E) = \langle \delta_{E,E_\mu} \rangle = \sum_\mu p_\mu \delta_{E,E_\mu} = \frac{1}{Z} \sum e^{-\beta E_\mu} \delta_{E,E_\mu} = \frac{1}{Z} e^{-\beta E} \sum \delta_{E,E_\mu}. \quad (1.22)$$

Επειδή προφανώς  $\sum_\mu \delta_{E,E_\mu} = g(E)$  παίρνουμε

$$p(E) = \langle \delta_{E,E_\mu} \rangle = \frac{g(E) e^{-\beta E}}{Z}. \quad (1.23)$$

Για ένα τυπικό σύστημα σαν αυτά που θα μελετήσουμε ισχύει

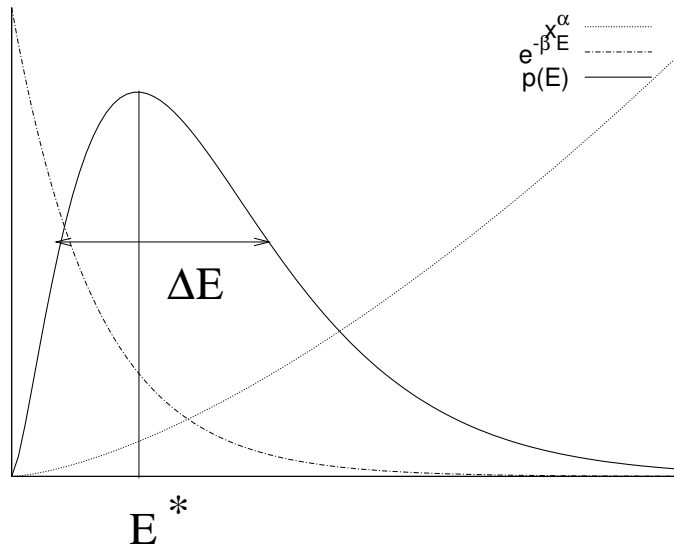
$$g(E) \sim E^{\alpha N} \quad (1.24)$$

όπου  $N$  ο αριθμός των βαθμών ελευθερίας του συστήματος και  $\alpha$  μια σταθερά. Για παράδειγμα  $\alpha = 3/2$  για το ιδανικό αέριο. Η ποιοτική συμπεριφορά της κατανομής (1.23) φαίνεται στο Σχήμα 1.2. Γενικά παρατηρούμε ότι οι πιθανές τιμές της ενέργειας επικεντρώνονται γύρω από μια τιμή  $E^*$  και η απόκλιση  $\Delta E$  είναι ένα μέτρο της διασποράς των τιμών που όπως θα δείξουμε παρακάτω ο λόγος  $\Delta E/E$  μειώνεται με το  $N$  με χαρακτηριστική συμπεριφορά  $1/\sqrt{N}$ . Πράγματι η συνάρτηση (ανάλογη της  $p(E)$ )

$$\tilde{p}(E) = E^{\alpha N} e^{-\beta E} = e^{-\beta E - \alpha N \ln E} \quad (1.25)$$

έχει μέγιστο όταν

$$\left. \frac{\partial \ln \tilde{p}(E)}{\partial E} \right|_{E=E^*} = 0 \Rightarrow \left. \frac{\partial}{\partial E} (-\beta E + \alpha N \ln E) \right|_{E=E^*} = -\beta + \frac{\alpha N}{E^*} = 0 \quad (1.26)$$



Σχήμα 1.1: Η πιθανότητα  $p(E)$  όπως προκύπτει από τον ανταγωνισμό του παράγοντα Boltzmann  $e^{-\beta E}$  και της πυκνότητας καταστάσεων  $g(E) \sim E^{\alpha N}$  για μία τυπική περίπτωση.

ή

$$E^* = \frac{\alpha}{\beta} N. \quad (1.27)$$

Το  $E^*$  μετατοπίζεται προς μεγαλύτερες τιμές με την αύξηση της θερμοκρασίας (μείωση του  $\beta$ ) και είναι όπως αναμένεται ανάλογο του μεγέθους του συστήματος. Αναπτύσσοντας κατά Taylor

$$\begin{aligned} \ln \tilde{p}(E) &= \ln \tilde{p}(E^*) + (E - E^*) \left. \frac{\partial \ln \tilde{p}(E)}{\partial E} \right|_{E=E^*} + \frac{1}{2} (E - E^*)^2 \left. \frac{\partial^2 \ln \tilde{p}(E)}{\partial E^2} \right|_{E=E^*} + \dots \\ &= \ln \tilde{p}(E^*) + \frac{1}{2} (E - E^*)^2 \left( -\frac{\alpha N}{(E^*)^2} \right) + \dots \end{aligned} \quad (1.28)$$

όπου χρησιμοποιήσαμε την συνθήκη (1.26) και υπολογίσαμε την  $\left. \frac{\partial^2 \ln \tilde{p}(E)}{\partial E^2} \right|_{E=E^*}$ . Οπότε προκύπτει ότι

$$p(E) \approx p(E^*) e^{-\alpha N \frac{(E-E^*)^2}{2(E^*)^2}} \quad (1.29)$$

Η παραπάνω κατανομή είναι Gaussian με τυπική απόκλιση

$$\Delta E \sim \sqrt{\frac{(E^*)^2}{\alpha N}} = \sqrt{\frac{(\frac{\alpha N}{\beta})^2}{\alpha N}} \sim \frac{\sqrt{N}}{\beta} \quad (1.30)$$

όπου κρατήσαμε μόνο την εξάρτηση από το μέγεθος του συστήματος  $N$  και τη θερμοκρασία  $\beta$ . Οπότε πράγματι επιβεβαιώνουμε τη σχέση (1.9) ότι

$$\frac{\Delta E}{E^*} \sim \frac{\frac{\sqrt{N}}{\beta}}{\frac{N}{\beta}} = \frac{1}{\sqrt{N}}. \quad (1.31)$$

Η παραπάνω ανάλυση υπέθεσε αναλυτική συμπεριφορά (ανάπτυγμα Taylor, σχέση (1.28)) η οποία δεν ισχύει όταν είμαστε σε ένα κρίσιμο σημείο μιας μετάβασης φάσης.

Μια άλλη σημαντική περίπτωση που η παραπάνω ανάλυση δεν ισχύει είναι όταν η κατανομή  $p(E)$  έχει παραπάνω από ένα μέγιστα<sup>3</sup>. Αυτό συμβαίνει όταν το σύστημα υπόκειται σε μετάβαση φάσης πρώτης τάξης, π.χ. όταν ο πάγος γίνεται νερό ή όταν ένα μαγνητικό υλικό που βρίσκεται σε ένα μαγνητικό πεδίο χάνει τη μαγνητισή του λόγω αύξησης της θερμοκρασίας του. Στην περίπτωση αυτή οι δύο καταστάσεις πάγος - νερό/μαγνήτης - παραμαγνήτης έχουν την ίδια πιθανότητα εμφάνισης ("συνυπάρχουν") και μία τυπική κατανομή με δομή δύο κορυφών φαίνεται στο Σχήμα 1.2

### 1.3 ΔΙΑΚΥΜΑΝΣΕΙΣ

Κάθε παρατηρήσιμη ποσότητα  $Q$  έχει στοχαστική συμπεριφορά σύμφωνα με μια κατανομή πιθανότητας  $p(Q)$  που προκύπτει από την κατανομή Boltzman (1.6). Μια τέτοια κατανομή χαρακτηρίζεται πλήρως από τη μέση τιμή  $\langle Q \rangle$  και τις ροπές ανώτερης τάξης, δηλ. τις μέσες τιμές  $\langle (Q - \langle Q \rangle)^n \rangle$ ,  $n = 1, 2, 3, \dots$ . Η πιο χρήσιμη ποσότητα από αυτές δίνεται από τη διακύμανση γύρω από τη μέση τιμή για  $n = 2$

$$(\Delta Q)^2 \equiv \langle (Q - \langle Q \rangle)^2 \rangle = \langle Q^2 \rangle - \langle Q \rangle^2. \quad (1.32)$$

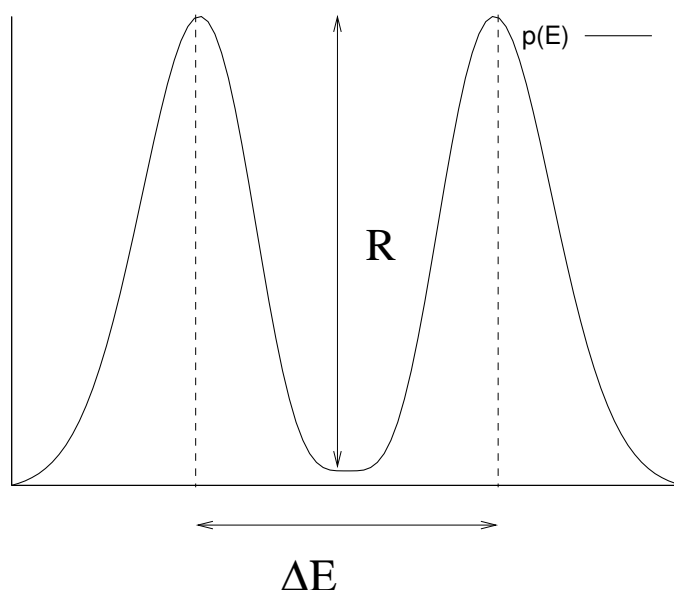
Η ποσότητα αυτή δίνει τις περισσότερες φορές ένα μέτρο της απόκλισης της  $Q$  από τη μέση τιμή της  $\langle Q \rangle$ . Για την περίπτωση της ενέργειας  $Q = E$  παίρνουμε

$$(\Delta E)^2 \equiv \langle (E - \langle E \rangle)^2 \rangle = \langle E^2 \rangle - \langle E \rangle^2, \quad (1.33)$$

και από τις σχέσεις

$$\langle E^2 \rangle = \frac{1}{Z} \sum_{\mu} E_{\mu}^2 e^{-\beta E_{\mu}} = \frac{1}{Z} \frac{\partial^2}{\partial \beta^2} \sum_{\mu} e^{-\beta E_{\mu}} = \frac{1}{Z} \frac{\partial^2 Z}{\partial \beta^2} \quad (1.34)$$

<sup>3</sup>Όταν έχει περισσότερα από ένα τοπικά ελάχιστα, το ολικό ελάχιστο επικρατεί των υπολοίπων στο θερμοδυναμικό όριο  $N \rightarrow \infty$ .



Σχήμα 1.2: Η πιθανότητα  $p(E)$  με δομή δύο κορυφών σε σύστημα με μετάβαση φάσης 1ης τάξης. Τα δύο μέγιστα αντιστοιχούν στις δύο συνυπάρχουσες καταστάσεις (“πάγος”-“νερό”) και  $\Delta E/N$  αντιστοιχεί στη λανθάνουσα θερμότητα (latent heat). Στο θερμοδυναμικό όριο  $N \rightarrow \infty$  το χάσμα  $R$  γίνεται μεγαλύτερο, συνήθως σαν  $R \sim e^{-fA}$  όπου  $A$  η ελάχιστη επιφάνεια που χωρίζει τις δύο φάσεις και  $f$  η διεπαφική τάση (interface tension).

και

$$\langle E \rangle = \frac{1}{Z} \sum_{\mu} E_{\mu} e^{-\beta E_{\mu}} = -\frac{1}{Z} \frac{\partial}{\partial \beta} \sum_{\mu} e^{-\beta E_{\mu}} = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} \quad (1.35)$$

προκύπτει ότι

$$(\Delta E)^2 = \langle E^2 \rangle - \langle E \rangle^2 = \frac{1}{Z} \frac{\partial^2 Z}{\partial \beta^2} - \left( -\frac{1}{Z} \frac{\partial Z}{\partial \beta} \right)^2 = \frac{\partial^2 \ln Z}{\partial \beta^2}. \quad (1.36)$$

που σύμφωνα με τη σχέση (1.11) είναι η ειδική θερμότητα

$$C = \frac{\partial \langle E \rangle}{\partial T} = k\beta^2 (\Delta E)^2. \quad (1.37)$$

Άρα καταλήγουμε στο ενδιαφέρον συμπέρασμα ότι η ειδική θερμότητα ενός συστήματος (θερμοδυναμική ποσότητα) συνδέεται άμεσα με τις μικροσκοπικές στατιστικές διακυμάνσεις της ενέργειας του συστήματος. Αυτό είναι γνωστό σαν το θεώρημα γραμμικής απόκρισης (linear response theorem) και ισχύει γενικά για οποιαδήποτε ποσότητα έχει γραμμική σύζευξη με το σύστημα.

Μια τέτοια ποσότητα για ένα μαγνητικό σύστημα που βρίσκεται μέσα σε ομογενές μαγνητικό πεδίο  $B$  είναι η μαγνήτιση  $M$ . Αν  $M_\mu$  είναι η μαγνήτιση του συστήματος στην κατάσταση  $\mu$  και υποθέσουμε ότι είναι στην διεύθυνση του  $\vec{B}$  τότε η Hamiltonian του συστήματος γίνεται

$$H = E - BM \quad (1.38)$$

και η συνάρτηση επιμερισμού

$$Z = \sum_{\mu} e^{-\beta E_{\mu} + \beta B M_{\mu}}. \quad (1.39)$$

Ο όρος “γραμμική σύζευξη” οφείλεται στη γραμμική απόκριση του συστήματος στο μαγνητικό πεδίο δηλ. στον όρο  $BM$  που είναι γραμμικός στη μαγνήτιση  $M$ . Οι ποσότητες  $B$  και  $M$  ονομάζονται συζυγείς. Παρόμοια σχέση έχουν η όγκος/πίεση ( $V/P$ ) σε ένα αέριο ή το χημικό δυναμικό/αριθμός σωματιδίων ( $\mu/N$ ) στη μεγαλοκανονική συλλογή.

Εξ' αιτίας αυτής της γραμμικής σύζευξης παίρνουμε

$$\langle M \rangle = \frac{1}{Z} \sum_{\mu} E_{\mu} e^{-\beta E_{\mu} + \beta B M_{\mu}} = \frac{1}{\beta Z} \frac{\partial Z}{\partial B} = -\frac{\partial F}{\partial B} \quad (1.40)$$

σχέση που είναι ανάλογη με την (1.10). Η αντίστοιχη της (1.36) προκύπτει από (1.32) για  $Q = M$

$$(\Delta M)^2 \equiv \langle (M - \langle M \rangle)^2 \rangle = \langle M^2 \rangle - \langle M \rangle^2, \quad (1.41)$$

και από

$$\langle M^2 \rangle = \frac{1}{Z} \sum_{\mu} M_{\mu}^2 e^{-\beta E_{\mu} + \beta B M_{\mu}} = \frac{1}{\beta^2 Z} \frac{\partial^2 Z}{\partial B^2}, \quad (1.42)$$

οπότε

$$(\Delta M)^2 = \frac{1}{\beta^2} \left\{ \frac{1}{Z} \frac{\partial^2 Z}{\partial B^2} - \frac{1}{Z^2} \left( \frac{\partial Z}{\partial B} \right)^2 \right\} = \frac{1}{\beta^2} \frac{\partial^2 \ln Z}{\partial B^2} = \frac{1}{\beta} \frac{\partial \langle M \rangle}{\partial B}. \quad (1.43)$$

Η μαγνητική επιδεκτικότητα  $\chi$  ορίζεται από τη σχέση

$$\chi = \frac{1}{N} \frac{\partial \langle M \rangle}{\partial B} = \frac{\beta}{N} (\langle (M - \langle M \rangle)^2 \rangle) \quad (1.44)$$

και σχετίζεται άμεσα με τις διακυμάνσεις της μαγνήτισης. Η παραπάνω ανάλυση μπορεί να γίνει πανομοιότυπα για οποιοδήποτε ζεύγος συζυγών ποσοτήτων.

## 1.4 ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΣΧΕΤΙΣΜΟΥ

Οι συναρτήσεις συσχετισμού προκύπτουν από τη συζήτηση της προηγούμενης παραγράφου όταν θεωρήσουμε μαγνητικά πεδία τα οποία όμως έχουν τιμή που εξαρτάται από τη θέση στο χώρο που θεωρούμε μέσα στο σύστημα. Για λόγους απλότητας (αλλά και επειδή πρόκειται να μελετήσουμε μόνο τέτοια συστήματα) το σύστημα μας βρίσκεται μέσα σε ένα “χώρο” στον οποίο οι δυνατές θέσεις είναι οι διακριτές θέσεις ενός πλέγματος τις οποίες αντιστοιχούμε σε φυσικούς αριθμούς<sup>4</sup>  $i = 1, \dots, N$ . Τότε το μαγνητικό πεδίο θα είναι συνάρτηση της θέσης στο πλέγμα  $B_i$  και αλληλεπιδρά με το σπιν  $s_i$

$$H = E - \sum_i B_i s_i \quad (1.45)$$

και η μαγνήτιση  $m_i \equiv s_i$ <sup>5</sup> στη θέση πλέγματος  $i$  είναι

$$\langle s_i \rangle = \frac{1}{\beta} \frac{\partial \ln Z}{\partial B_i} \quad (1.46)$$

και η συνάρτηση συσχετισμού δύο σημείων (connected two point correlation function) ορίζεται ως

$$G_c^{(2)}(i, j) = \langle (s_i - \langle s_i \rangle)(s_j - \langle s_j \rangle) \rangle = \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle = \frac{1}{\beta^2} \frac{\partial^2 \ln Z}{\partial B_i \partial B_j}. \quad (1.47)$$

Η παραπάνω συνάρτηση αυτή έχει μεγάλη θετική τιμή όταν οι τιμές  $s_i, s_j$  είναι ισχυρά συσχετισμένες, δηλ. “μεταβάλλονται μαζί” στα τυχαία δείγματα που παίρνουμε από το σύστημα, ενώ αντίθετα είναι σχεδόν μηδέν όταν η τιμή της  $s_i$  δεν εξαρτάται ελάχιστα από την  $s_j$  (ασυσχέτιστες τυχαίες μεταβλητές). Υπάρχει φυσικά και οι περίπτωση οι  $s_i, s_j$  να είναι ισχυρά αντι-συσχετισμένες και η συνάρτηση συσχετισμού να είναι αρνητική.

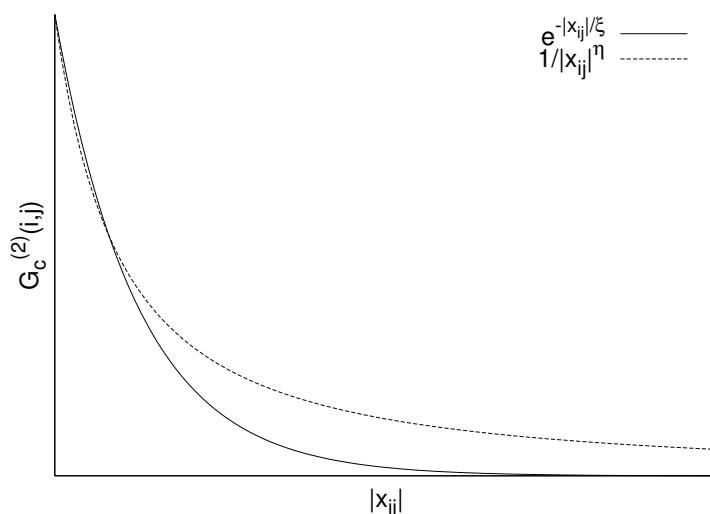
Η συνάρτηση συσχετισμού  $G_c^{(2)}(i, j)$  παίρνει τη μέγιστη τιμή της  $\langle (s_i - \langle s_i \rangle)^2 \rangle$  για  $i = j$ . Στη συνέχεια πέφτει γρήγορα κατά απόλυτη τιμή. Για ένα σύνηθες σύστημα

$$G_c^{(2)}(i, j) \sim e^{-|x_{ij}|/\xi}, \quad (1.48)$$

όπου  $|x_{ij}|$  η απόσταση των σημείων  $i, j$ .  $\xi$  είναι ένα χαρακτηριστικό μήκος για το σύστημα που δίνει ένα μέτρο σε πόση απόσταση υπάρχει ουσιαστικός συσχετισμός μεταξύ των τιμών της

<sup>4</sup>Οι βαθμοί ελευθερίας που αναφέραμε προηγουμένως μπορεί να είναι περισσότεροι από  $N$

<sup>5</sup>Οι δύο φυσικές ποσότητες είναι διαφορετικές αλλά ανάλογες. Εδώ απλά αγνοούμε τη σταθερά αναλογίας.



Σχήμα 1.3: Η συνάρτηση συσχετισμού  $G_c^{(2)}(i, j)$  για  $\xi < \infty$  και  $\xi \rightarrow \infty$

μαγνήτισης σε δύο πλεγματικές θέσεις. Εξαρτάται από τις παραμέτρους που ορίζουν το σύστημα  $\xi = \xi(\beta, B, N, \dots)$ . Είναι σημαντικό να κατανοηθεί ότι το μήκος συσχετισμού ορίζει μια κλίμακα μήκους που προκύπτει δυναμικά και όχι από τον ορισμό του συστήματος όπως λ.χ. το μέγεθος  $L$  του συστήματος ή η απόσταση  $a$  μεταξύ δύο πλεγματικών σημείων (πλεγματική σταθερά). Συνήθως το  $\xi$  είναι της τάξης μεγέθους της πλεγματικής σταθεράς  $a$  και το σύστημα δεν παρουσιάζει συσχετισμούς σε μακροσκοπικές κλίμακες (δηλ. της τάξης του  $L$ ).

Πολύ ενδιαφέρουσα φυσική προκύπτει όταν ρυθμίσουμε με λεπτότητα τις παραμέτρους από τις οποίες εξαρτάται το  $\xi$  έτσι ώστε  $\xi \rightarrow \infty$  (στην πράξη  $\xi \approx L$ ). Αυτό γίνεται στην περιοχή μίας συνεχούς (όχι πρώτης τάξης) μετάβασης φάσης. Στην περίπτωση αυτή η εκθετική συμπεριφορά χάνεται και έχουμε πολύ βραδύτερη πτώση της  $G_c^{(2)}(i, j)$  (βλ. Σχήμα 1.4), που σε  $d$  χωρικές διαστάσεις δίνεται από

$$G_c^{(2)}(i, j) \sim \frac{1}{|x_{ij}|^{d-2+\eta}}. \quad (1.49)$$

Καθώς πλησιάζουμε την περίπτωση αυτή, συμβαίνει οι συσχετισμοί να εκτείνονται σε αποστάσεις  $|x_{ij}| \gg a$ . Τότε το σύστημα πάυει να “βλέπει” τις λεπτομέρειες του πλέγματος και συμπεριφέρεται με πολύ καλή προσέγγιση σαν το πλέγμα να ήταν ένας συνεχής χώρος. Το όριο αυτό αναφέρεται ως το “συνεχές όριο” (continuum limit) μιας θεωρίας που ορίζεται σε ένα πλέγμα. Εξαιτίας του ότι η λεπτομέρειες του πλέγματος γίνονται ασήμαντες

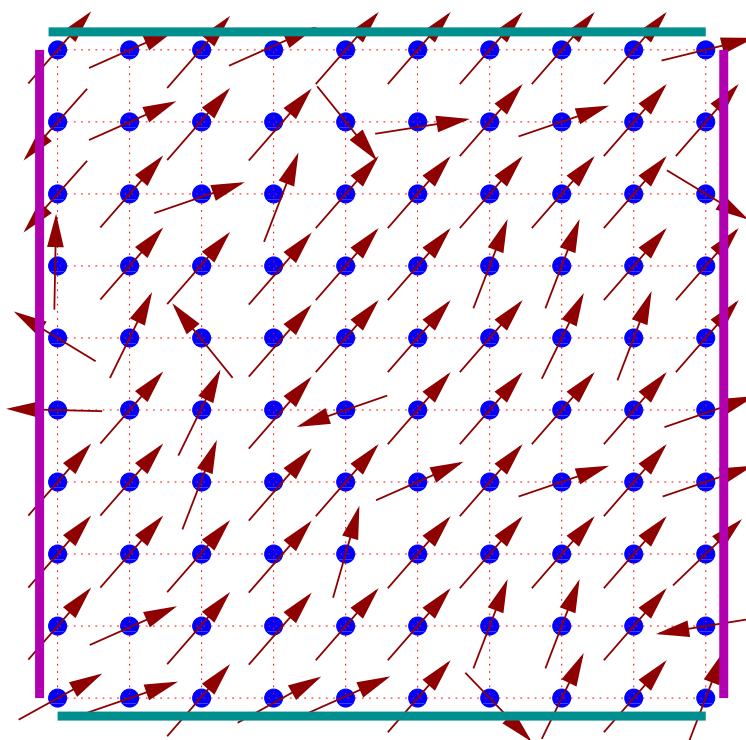


στο όριο αυτό, θεωρίες που ορίζονται μικροσκοπικά με διαφορετικό τρόπο (λ.χ. μια σε τετραγωνικό πλέγμα και μία σε εξαγωνικό) έχουν το ίδιο συνεχές όριο. Το φαινόμενο αυτό ονομάζεται παγκοσμιότητα (universality) και παίζει κεντρικό ρόλο στη μελέτη στατιστικών συστημάτων καθώς και στην κβαντική θεωρία πεδίου.

## 1.5 ΠΡΟΤΥΠΟ ISING

Το πρότυπο Ising (1925) [11] έχει παίξει ιστορικά πολύ σημαντικό ρόλο στις εξελίξεις των ιδεών της στατιστικής φυσικής και της κβαντικής θεωρίας πεδίου. Ειδικότερα το πρότυπο στις δύο διαστάσεις είναι αρκετά σύνθετο ώστε να έχει ενδιαφέρουσες μη τετριμμένες ιδιότητες και αρκετά απλό ώστε να μπορούμε να αντλήσουμε πολύτιμες πληροφορίες για τη φυσική των μεταβάσεων φάσης. Το πρότυπο εκδηλώνει μετάβαση φάσης 2ης τάξης με αποτέλεσμα να μπορούν να μελετηθούν οι ιδιότητες των συνεχών μεταβάσεων φάσης (κρίσιμοι εκθέτες, παγκοσμιότητα, ομάδα επανακανονικοποίησης, συνεχές όριο). Με την αναλυτική λύση του Onsager (1948) [12] και άλλων παίρνουμε αποτελέσματα με τα οποία μπορούμε να ελέγξουμε προσεγγιστικές μεθόδους, όπως οι προσομοιώσεις Μόντε Κάρλο, θεωρία διαταραχών, mean field theory κλπ. Επί πλέον έχει και φυσικό ενδιαφέρον μια και είναι ένα πρότυπο ενός μαγνητικού υλικού που παρ' όλη την απλότητά του περιγράφει ποιοτικά πολλές από τις μη τετριμμένες ιδιότητές τους. Επίσης λόγω της παγκοσμιότητας μπορεί να θεωρηθεί πρότυπο για μετάβαση φάσης σε υγρό-αέριο (τριπλό σημείο) απεικονίζεται στο Coulomb Gas ..... Εξάριετο βιβλίο για πρότυπα της στατιστικής φυσικής που λύνονται επακριβώς με αναλυτικές μεθόδους αποτελεί το βιβλίο του Baxter [10].

Για τον ορισμό του προτύπου θεωρήστε ένα διδιάστατο τετραγωνικό πλέγμα όπως αυτό του Σχήματος 1.5 στο οποίο τοποθετείτε σε κάθε πλεγματική θέση (node) "άτομα" ή "μαγνητάκια" με spin  $s_i$ . Η γεωμετρία καθορίζεται από την ελάχιστη απόσταση μεταξύ γειτόνων, την πλεγματική σταθερά  $a$ , και τον αριθμό των πλεγματικών θέσεων  $N$ . Σε κάθε πλευρά έχουμε  $L$  πλεγματικές θέσεις έτσι ώστε  $N = L \times L = L^d$  με  $d = 2$  τη διάσταση του χώρου. Η τοπολογία καθορίζεται από τις σχέσεις γειτονίας, και ιδιαίτερα από τις σχέσεις γειτονίας των πλεγματικών θέσεων που βρίσκονται στο σύνορο του τετραγώνου. Η τοροϊδής τοπολογία λαμβάνεται αν δημιουργήσουμε δεσμό μεταξύ των γειτόνων που ανήκουν στις κάθετες και οριζόντιες πλευρές του τετραγώνου το Σχήματος (1.5). Η δυναμική του συστήματος καθορίζεται από τη μαγνητική αλληλεπίδραση (spin-spin interaction) την οποία απλο-



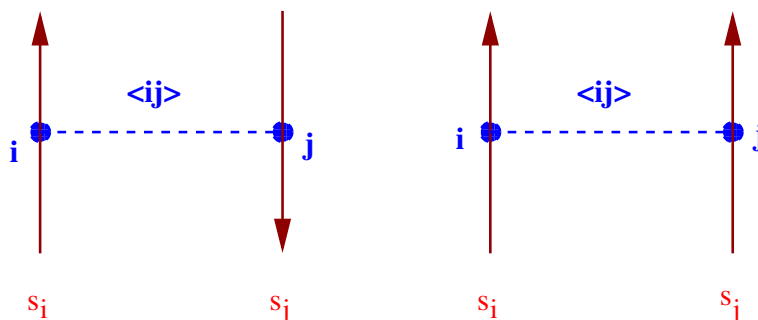
Σχήμα 1.4: Διδιάστατο τετραγωνικό πλέγμα του οποίου οι θέσεις  $i = 1, \dots, N$  καταλαμβάνονται από “άτομα” ή “μαγνητάκια” με σπιν  $s_i$  οποία εδώ έχουν τυχαίο προσανατολισμό στο επίπεδο (πρότυπο XY). Συνήθως λαμβάνονται υπ’ όψη μόνο οι αλληλεπιδράσεις πλησιεστέρων γειτόνων  $-J\vec{s}_i \cdot \vec{s}_j$  για κάθε σύνδεσμο  $\langle ij \rangle$ . Η τοπολογία είναι τοροϊδής (toroidal) όταν ταυτίσουμε τις οριζόντιες πλευρές μεταξύ τους και τις κάθετες μεταξύ τους, δηλ. τα ίδια χρώματα στο σχήμα.

ποιούμε να είναι κοντινής εμβέλειας και ιδιαίτερα μόνο μεταξύ των πλησιεστέρων γειτόνων.

Στο σιδηρομαγνητικό πρότυπο Ising θεωρούμε τις δυνατές τιμές των σπιν να είναι δύο, “πάνω” και “κάτω” ή  $+1$  και  $-1$ . Στην πιο απλή περίπτωση η αλληλεπίδραση είναι αλληλεπίδραση σπιν-σπιν μόνο μεταξύ πλησιεστέρων γειτόνων με την κατάσταση με τα σπιν ομόρροπα να έχει τη χαμηλότερη ενέργεια<sup>6</sup>. Αυτό απεικονίζεται στο Σχήμα 1.5. Το σύστημα μπορεί να βρίσκεται υπό την επίδραση ομογενούς μαγνητικού πεδίου  $B$  του οποίου η διεύθυνση θεωρείται να είναι παράλληλη ή αντιπαράλληλη με αυτή των σπιν. Είμαστε τώρα έτοιμοι να γράψουμε την Hamiltonian και τη συνάρτηση επιμερισμού του συστήματος.

Θεωρούμε τετραγωνικό πλέγμα από  $N$  πλεγματικές θέσεις (sites ή vertices) διατεταγμένες σε τετράγωνο με πλευρές με  $L$  πλεγμα-

<sup>6</sup>Το αντίστροφο ισχύει για το αντισιδηρομαγνητικό πρότυπο Ising.



Σχήμα 1.5: Τα σπιν στο πρότυπο Ising παίρνουν δύο μόνο τιμές “πάνω” και “κάτω” και η ενέργεια του συστήματος προκύπτει από τη συνεισφορά κάθε συνδέσμου (link)  $\langle ij \rangle$ . Αυτή για το σιδηρομαγνητικό πρότυπο παίρνει δύο δυνατές τιμές  $+J$  και  $-J$  για αντίρροπα και ομόρροπα σπιν αντίστοιχα. Το σύστημα έχει τη συμμετρία της διακριτής ομάδας  $Z_2$ .

τικές θέσεις τις οποίες απαριθμούμε με αριθμούς  $i = 1, 2, \dots, N$ . Το πλέγμα έχει  $N_l$  δεσμούς (bonds ή links) μεταξύ των πλησιέστερων γειτόνων. Αυτοί χαρακτηρίζονται από τα ζεύγη των πλεγματικών θέσεων  $i, j$  που ενώνουν και θα τους συμβολίζουμε με  $\langle ij \rangle$ . Ταυτίζουμε τις πλευρές του τετραγώνου όπως στο Σχήμα 1.5. Επειδή τότε κάθε δεσμός ενώνει ακριβώς δύο πλεγματικές θέσεις και από κάθε πλεγματική θέση ξεκινούν ακριβώς τέσσερις δεσμοί θα ισχύει

$$2N_l = 4N \Rightarrow N_l = 2N. \quad (1.50)$$

Σε κάθε πλεγματική θέση τοποθετούμε σπιν  $s_i = \pm 1$ .

Η Hamiltonian του συστήματος θα δίνεται από τη σχέση

$$H = -J \sum_{\langle ij \rangle} s_i s_j - B \sum_i s_i. \quad (1.51)$$

Ο πρώτος όρος δίνει την αλληλεπίδραση μεταξύ των σπιν και για  $J > 0$  – το οποίο υποθέτουμε στο βιβλίο αυτό – το σύστημα είναι σιδηρομαγνητικό. Κάθε δεσμός που ενώνει ομόρροπα σπιν έχει ενέργεια  $-J$  που είναι κατά  $2J$  μικρότερη από ένα δεσμό με αντίρροπα σπιν. Το σύστημα ενεργειακά προτιμά καταστάσεις με δεσμούς ομόρροπους, δηλ. τα σπιν να είναι όλα προσανατολισμένα προς την ίδια κατεύθυνση. Η ελάχιστη ενέργεια αντιστοιχεί στη μοναδική<sup>7</sup> κατάσταση με όλα τα σπιν να κοιτούν προς την κατεύθυνση

<sup>7</sup>Στην περίπτωση του αντισιδηρομαγνητικού συστήματος  $J < 0$  η θεμελιώδης ενέργεια είναι εκφυλισμένη και οι αντίστοιχες καταστάσεις είναι πιο πολύπλοκο να περιγραφούν.

του  $B$ , την θεμελιώδη κατάσταση<sup>8</sup>. Η ενέργειά της είναι

$$E_0 = -JN_l - BN = -(2J + B)N. \quad (1.52)$$

Η συνάρτηση επιμερισμού είναι

$$Z = \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} \dots \sum_{s_N=\pm 1} e^{-\beta H\{s_i\}} \equiv \sum_{\{s_i\}} e^{\beta J \sum_{\langle ij \rangle} s_i s_j + \beta B \sum_i s_i}, \quad (1.53)$$

όπου  $\{s_i\} \equiv \{s_1, s_2, \dots, s_N\}$  είναι μια διάταξη (configuration) των σπιν στο πλέγμα. Το παραπάνω άθροισμα μοιάζει απλό να γίνει, ιδιαίτερα όταν  $B = 0$ , μια και η Hamiltonian παίρνει διακριτές τιμές και απλά μετράει τον αριθμό των ομόροπων σπιν (δες πρόβλημα 3). Ο αριθμός των όρων του είναι ίσος με τον αριθμό των δυνατών διατάξεων  $\{s_i\}$  των σπιν που είναι 2, δηλ. αυξάνουν εκθετικά με το  $N$ . Για ένα  $5 \times 5$  πλέγμα ο αριθμός των όρων είναι  $2^{25} \approx 3.4 \times 10^6$ . Σύμφωνα με τη γνώση του συγγραφέα αυτό είναι και το μεγαλύτερο πλέγμα στο οποίο έχει γίνει απ' ευθείας υπολογισμός του αθροίσματος αυτού.

Το διδιάστατο πρότυπο Ising με  $B = 0$  έχει την εξαιρετικά ενδιαφέροντα ιδιότητα ότι για  $\beta = \beta_c$  όπου

$$\beta_c = \frac{1}{2} \ln(1 + \sqrt{2}) \approx 0.4406867935 \dots \quad (1.54)$$

το σύστημα παρουσιάζει μετάβαση φάσης από την διατεταγμένη φάση ή φάση χαμηλής θερμοκρασίας όπου το σύστημα είναι μαγνητισμένο ( $\langle |M| \rangle > 0$ ) στην άτακτη φάση ή φάση υψηλής θερμοκρασίας όπου η μαγνήτιση χάνεται ( $\langle |M| \rangle = 0$ ). Η θερμοκρασία  $\beta_c$  είναι η κρίσιμη θερμοκρασία η θερμοκρασία Curie. Η μετάβαση φάσης είναι συνεχής, δευτέρας τάξης, γιατί η συνάρτηση  $\langle |M| \rangle(\beta)$  είναι συνεχής για  $\beta = \beta_c$  αλλά όχι η παράγωγός της. Αυτό φαίνεται ποιοτικά στο Σχήμα 1.5

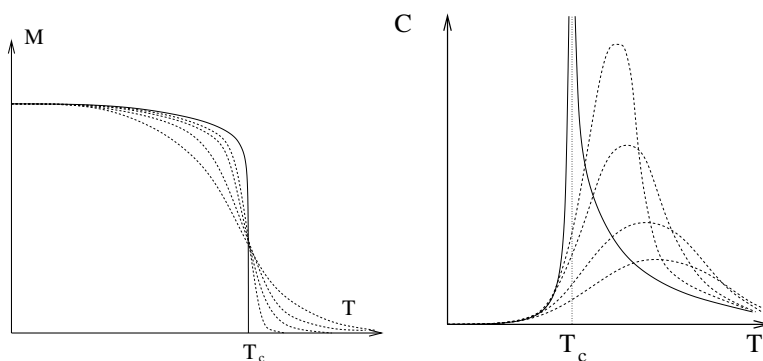
Για  $\beta \neq \beta_c$  η συνάρτηση συσχετισμού (1.47) συμπεριφέρονται όπως στη σχέση (1.48) με πεπερασμένο μήκος συσχετισμού  $\xi(\beta)$ . Καθώς πλησιάζουμε την κρίσιμη θερμοκρασία το μήκος συσχετισμού τείνει στο άπειρο και μάλιστα συμπεριφέρεται ασυμπτωτικά σαν

$$\xi(\beta) \equiv \xi(t) \sim |t|^{-\nu} \quad t = \frac{\beta_c - \beta}{\beta_c}. \quad (1.55)$$

Η συνάρτηση συσχετισμού συμπεριφέρεται σύμφωνα με την εξίσωση (1.49)

$$G_c^{(2)}(i, j) \sim \frac{1}{|x_{ij}|^\eta}. \quad (1.56)$$

<sup>8</sup>Για  $B = 0$  το σύστημα έχει "πάνω-κάτω" συμμετρία (συμμετρία  $Z_2$ ). Δύο καταστάσεις που προκύπτουν με την εφαρμογή της συμμετρίας αυτής (αντιστροφή όλων των σπιν) ταυτίζονται.



Σχήμα 1.6: Ποιοτική συμπεριφορά της μαγνήτισης (αριστερά) και της ειδικής θερμότητας (δεξιά) στην μετάβαση φάσης στο πρότυπο Ising. Με συνεχή γραμμή δείχνεται η (μη αναλυτική) συμπεριφορά στο θερμοδυναμικό όριο ενώ με διακεκομμένες η συμπεριφορά για αυξανόμενο αλλά πεπερασμένο μέγεθος  $N$  του συστήματος. Οι τελευταίες συγκλίνουν ομοιόμορφα προς τις μη αναλυτικές συναρτήσεις.

ενώ παρόμοια συμπεριφορά βάθμισης (scaling) παρουσιάζουν και η ειδική θερμότητα  $C$ , μαγνήτιση  $M \equiv \langle M \rangle$  και μαγνητική επιδεκτικότητα  $\chi$  σύμφωνα με τις σχέσεις

$$C \sim |t|^{-\alpha} \quad (1.57)$$

$$M \sim |t|^\beta \quad (1.58)$$

$$\chi \sim |t|^{-\gamma}. \quad (1.59)$$

ενώ η μαγνήτιση για  $t = 0$  και μαγνητικό πεδίο  $B \neq 0$  συμπεριφέρεται σύμφωνα με τη σχέση

$$M \sim B^{-1/\delta}. \quad (1.60)$$

Οι εκθέτες που παρουσιάζονται στις παραπάνω σχέσεις λέγονται κρίσιμοι εκθέτες ή εκθέτες βάθμισης (critical ή scaling exponents) και οι τιμές τους παρουσιάζουν την ιδιότητα της παγκοσμιότητας. Δηλ. οι τιμές τους δεν εξαρτώνται από τις λεπτομέρειες του πλέγματος (τετραγωνικό, τριγωνικό κλπ) ή της αλληλεπίδρασης (πλησιεστέρων ή μη γειτόνων, μεγαλύτερες δυνάμεις του σπιν κλπ) και μία ολόκληρη κλάση από πρότυπα έχουν την ίδια συμπεριφορά! Αρκεί τα πρότυπα αυτά να έχουν τις ίδιες ιδιότητες συμμετρίας, να ορίζονται σε χώρο ίδιας διάστασης και οι αλληλεπιδράσεις να είναι κοντινής εμβέλειας. Στο συγκεκριμένο πρότυπο

οι εκθέτες παίρνουν τις λεγόμενες τιμές Onsager

$$\begin{aligned}
 \alpha &= 0 \\
 \beta &= \frac{1}{8} \\
 \gamma &= \frac{7}{4} \\
 \delta &= 15 \\
 \nu &= 1 \\
 \eta &= \frac{1}{4}
 \end{aligned}
 \tag{1.61}$$

Η συμπεριφορά (1.55–1.60) χαρακτηρίζεται από τη μη αναλυτικότητα των αντίστοιχων συναρτήσεων. Αυτή δεν είναι δυνατόν να προκύψει από την συνάρτηση επιμερισμού (1.53) για πλέγμα με  $N$  θέσεις μία και ένα πεπερασμένο άθροισμα από εκθετικά είναι αναγκαστικά αναλυτική συνάρτηση (άρα και οι παράγωγοί της). Η μη αναλυτική συμπεριφορά φανερώνεται στο όριο απείρου μεγέθους (θερμοδυναμικό όριο) όπου η παραπάνω συναρτήσεις τείνουν προς μια μη-αναλυτική συνάρτηση όπως στο Σχήμα 1.5. Το χάσιμο της αναλυτικότητας οφείλεται στους συσχετισμούς των σπιν σε μακροσκοπικές αποστάσεις.

Σε πολλά συστήματα που μελετάμε τα παραπάνω μη αναλυτικά σημεία, αναζητούμε μια παράμετρο τάξης (order parameter) η οποία χαρακτηρίζει τη συμμετρία του συστήματος. Στην συγκεκριμένη περίπτωση του πρότυπου Ising η παράμετρος τάξης είναι η μαγνήτιση με τη συμμετρία  $s_i \rightarrow -s_i$ . Συνήθως στη μία φάση η παράμετρος τάξης είναι μη μηδενική ενώ στην άλλη μηδενίζεται. Αυτό συνεπάγεται μη αναλυτική συμπεριφορά αφού μια αναλυτική συνάρτηση που είναι μηδενική σε ένα διάστημα, είναι παντού μηδέν

Η συμπεριφορά παγκοσμιότητας ή ανεξαρτησίας κλίμακας (scale invariance) παρουσιάζεται κάθε φορά που έχουμε απόκλιση του μήκους συσχετισμού  $\xi$ . Στην περίπτωσή μας για να φτάσουμε το κρίσιμο σημείο έχουμε να ρυθμίσουμε μόνο μια παράμετρο, τη θερμοκρασία, οπότε αναμένει κανείς να εμφανιστεί μόνο μια καινούργια κλίμακα στο πρότυπο. Ανεξαρτησία κλίμακας εμφανίζεται όταν το μήκος συσχετισμού γίνει πολύ μεγαλύτερο από τη μικροσκοπική κλίμακα  $a$ , οπότε οποιαδήποτε ποσότητα σαν συνάρτηση της απόστασης  $r$  μπορεί να εξαρτάται μόνο από το λόγο  $r/\xi$ . Η παγκοσμιότητα προκύπτει από το γεγονός ότι στην περίπτωση αυτή φαίνεται τα πάντα να εξαρτώνται από εκείνες τις διακυμάνσεις με μεγάλο μήκος κύματος που απαιτούνται από τη συμμετρία της παραμέτρου τάξης  $M$ . Η σημαντική απλούστευση που

προκύπτει είναι ότι για να μελετήσουμε ένα πραγματικό φυσικό σύστημα σε μια συνεχή μετάβαση φάσης  $\xi \rightarrow \infty$  αρκεί να μελετήσουμε το απλούστερο πρότυπο με τη δεδομένη συμμετρία και (χωρικές) διαστάσεις το οποίο είναι ανεξάρτητο κλίμακας (scale invariant).

## 1.6 ΠΡΟΒΛΗΜΑΤΑ

1. Αποδείξτε τη σχέση (1.20). Επαναλάβετε για την κατανομή (1.21).
2. Αποδείξτε τη σχέση (1.47).
3. Αποδείξτε τη σχέση (1.50). Βρείτε τις ανάλογες σχέσεις για το αντίστοιχο τριγωνικό και εξαγωνικό πλέγμα.
4. Επαναλάβετε την προηγούμενη άσκηση για το κυβικό πλέγμα στις 3 διαστάσεις. Γενικεύστε για το υπερκυβικό στις  $d$  διαστάσεις.
5. Η Hamiltonian του προτύπου Ising για  $B = 0$  μπορεί ισοδύναμα να γραφεί σαν

$$H' = -J' \sum_{\langle ij \rangle} \delta_{i,j} \quad (1.62)$$

Βρείτε τη σχέση μεταξύ των  $J, J', H, H'$  της σχέσης (1.51) ώστε τα συστήματα να είναι φυσικά ισοδύναμα. Επαναλάβετε το ίδιο και για τη Hamiltonian

$$H'' = -J'' \sum_{\langle ij \rangle} (\delta_{i,j} - 1) \quad (1.63)$$

Παρατηρήστε ότι οι  $H'$  και  $H''$  μετρούν ομόρροπους και αντίρροπους δεσμούς αντίστοιχα.





## ΚΕΦΑΛΑΙΟ 2

# ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΜΟΝΤΕ CARLO

Στο κεφάλαιο αυτό γίνεται επισκόπηση των βασικών αρχών μιας προσομοίωσης με τη μέθοδο Μόντε Κάρλο. Δεν υπάρχει ενδιαφέρον φυσικό σύστημα του οποίου ο υπολογισμός της συνάρτησης επιμερισμού να γίνεται με απ' ευθείας υπολογισμό του αθροίσματος (1.7). Στις περισσότερες περιπτώσεις καταφεύγουμε σε στατιστική δειγματοληψία των όρων στην παραπάνω εξίσωση και η πιο διαδεδομένη μέθοδος, λόγω της αποτελεσματικότητας της και της γενικής της εφαρμογής, είναι η μέθοδος Μόντε Κάρλο. Είναι αξιοσημείωτο ότι σε συστήματα όπως το απλό πρότυπο Ising το άθροισμα (1.53) είναι προσβάσιμο μόνο με τη μέθοδο αυτή και ότι χρήσιμα αποτελέσματα μπορούν να εξαχθούν με τη μελέτη δειγμάτων της τάξης του  $\approx 10^{-3000}$  συνολικού χώρου των καταστάσεων<sup>1</sup>. Αυτό γίνεται διαισθητικά αποδεκτό λαμβάνοντας υπ' όψη ότι και στο εργαστήριο ένα πραγματικό στατιστικό σύστημα μας δίνει πειραματική πληροφορία όταν μέσα στο χρόνο του πειράματος το σύστημα επισκέπτεται ακόμα μικρότερα ποσοστά του φασικού χώρου<sup>2</sup>.

---

<sup>1</sup>Λ.χ. για το  $d = 2, L = 100$  πρότυπο Ising έχουμε  $2^{10000} \approx 10^{3010}$  καταστάσεις. Σε μια τυπική προσομοίωση παίρνουμε δείγμα από  $\approx 10^7$  καταστάσεις δηλ. ποσοστό  $\approx 10^{-3003}$  !!

<sup>2</sup>Ένα τυπικό αέριο με  $10^{22}$  μόρια σε δοχείο 1 λίτρου σε θερμοκρασία δωματίου και ατμοσφαιρική πίεση έχει μόρια που κινούνται με τυπικές ταχύτητες  $\approx 100ms^{-1}$  δηλ. τυπικό μήκος κύματος κατά de Broglie  $\approx 10^{-10}m$  δίνοντας  $\approx 10^{27}$  διαφορετικές καταστάσεις ανά μόριο. Συνολικά το σύστημα έχει  $(10^{27})^{10^{22}}$  καταστάσεις. Με ένα τυπικό ρυθμό από  $10^9$  κρούσεις ανά δευτερόλεπτο έχουμε  $\approx 10^{31}$  αλλαγές καταστάσεων ανά δευτερόλεπτο. Άρα θα χρειαστεί χρόνο περίπου  $10^{10^{23}}$  την ηλικία του σύμπαντος ώστε το σύστημα να επισκεφτεί όλες τις καταστάσεις [1].

## 2.1 ΔΕΙΓΜΑΤΟΛΗΨΙΑ

Ο κύριος στόχος μας είναι ο προσδιορισμός της μέσης τιμής  $\langle Q \rangle$  μιας φυσικής ποσότητας  $Q$  (λ.χ. ενέργειας, μαγνήτισης, συνάρτησης συσχετισμού) σε ένα στατιστικό σύστημα στην κανονική συλλογή

$$\langle Q \rangle = \sum_{\mu} p_{\mu} Q_{\mu} = \frac{\sum_{\mu} Q_{\mu} e^{-\beta E_{\mu}}}{\sum_{\mu} e^{-\beta E_{\mu}}}. \quad (2.1)$$

Για το λόγο αυτό επιλέγουμε δείγμα από καταστάσεις  $\{\mu_1, \mu_2, \dots, \mu_M\}$  οι οποίες κατανέμονται σύμφωνα με την κατανομή πιθανότητας  $P_{\mu}$  και ορίζουμε τον εκτιμητή (estimator)  $Q_M$  της  $\langle Q \rangle$

$$Q_M = \frac{\sum_{i=1}^M Q_{\mu_i} P_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}{\sum_{i=1}^M P_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}. \quad (2.2)$$

Ο παραπάνω τύπος γίνεται εύκολα κατανοητός αφού για μεγάλο δείγμα  $P_{\mu_i} \approx$  "Συχνότητα εύρεσης κατάστασης  $\mu_i$  στο δείγμα" και περιμένουμε ότι

$$\langle Q \rangle = \lim_{M \rightarrow \infty} Q_M. \quad (2.3)$$

Ο στόχος μας είναι η κατάλληλη επιλογή της κατανομής  $P_{\mu}$  έτσι ώστε η σύγκλιση (2.3) να γίνεται γρήγορα. Διακρίνουμε τις εξής περιπτώσεις:

### 2.1.1 ΑΠΛΗ ΔΕΙΓΜΑΤΟΛΗΨΙΑ

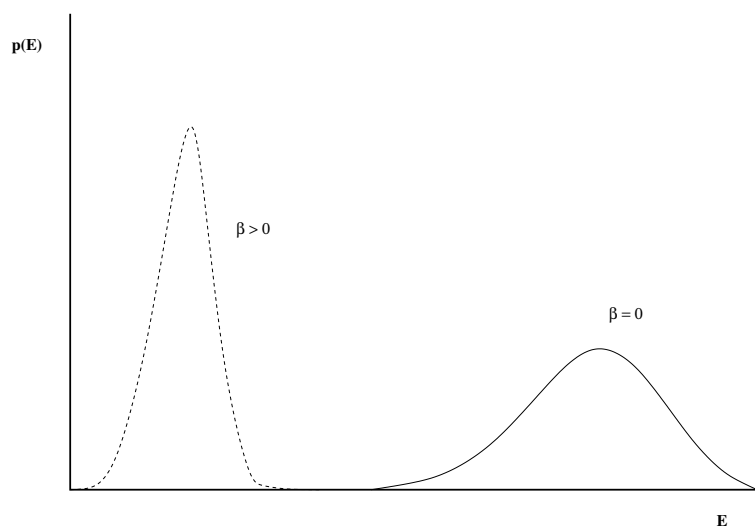
Διαλέγουμε  $P_{\mu} =$  σταθ. οπότε η (2.2) γίνεται

$$Q_M = \frac{\sum_{i=1}^M Q_{\mu_i} e^{-\beta E_{\mu_i}}}{\sum_{i=1}^M e^{-\beta E_{\mu_i}}}. \quad (2.4)$$

Το πρόβλημα με την επιλογή αυτή είναι ότι έχουμε πρόβλημα επικάλυψης του δείγματος με τις καταστάσεις που πραγματικά συνεισφέρουν στο άθροισμα (2.1). Όπως αναφέραμε και στην εισαγωγή, ο αριθμός των καταστάσεων που επιλέγεται σε μια προσομοίωση Μόντε Κάρλο στο δείγμα είναι ένα ελάχιστο ποσοστό του χώρου των καταστάσεων του συστήματος. Άρα η πιθανότητα να πάρουμε εκείνες τις καταστάσεις που δίνουν σημαντική συνεισφορά στο άθροισμα (2.1) είναι εν γένει πολύ μικρή. Ας πάρουμε για παράδειγμα την περίπτωση  $Q = E$  στο πρότυπο Ising. Σύμφωνα με τη σχέση (1.23) έχουμε

$$\langle E \rangle = \sum_E E p(E) \quad (2.5)$$

όπου  $p(E)$  είναι η πιθανότητα εμφάνισης ενέργειας  $E$  στο σύστημα. Η ποιοτική μορφή της φαίνεται στο Σχήμα 1.2. Από τις σχέσεις (1.30) και (1.31) έχουμε ότι  $\Delta E/E \sim 1/\sqrt{N}$  και  $\Delta E \sim 1/\beta$  οπότε για τις περιπτώσεις  $\beta = 0$  και  $\beta > 0$  παίρνουμε ποιοτικά τη συμπεριφορά που φαίνεται στο Σχήμα 2.1.1. Η κατανομή της απλής δειγ-



Σχήμα 2.1: Η πιθανότητα  $p(E)$  για το πρότυπο Ising για θερμοκρασίες  $\beta = 0$  και  $\beta > 0$ . Οι δύο κατανομές έχουν ελάχιστη επικάλυψη.

ματοληψίας αντιστοιχεί στην περίπτωση  $\beta = 0$  (βλ. σχέση (1.6)). Για να υπολογίσουμε με ακρίβεια το άθροισμα (2.5) για  $\beta > 0$  πρέπει να πάρουμε καλό δείγμα στην περιοχή όπου το γινόμενο  $E p_{\beta > 0}(E)$  είναι σχετικά σημαντικό. Όπως βλέπουμε και στο Σχήμα 2.1.1 η πιθανότητα να πάρουμε στο δείγμα κατάσταση με ενέργεια τέτοια στην περιοχή όπου είναι το  $E p_{\beta > 0}(E)$  είναι σχετικά σημαντικό είναι πολύ μικρή όταν η δειγματοληψία γίνεται με την κατανομή  $p_{\beta = 0}(E)$

Παρ' όλο που η συγκεκριμένη μέθοδος έχει το μειονέκτημα αυτό, μπορεί να φανεί χρήσιμη σε ορισμένες περιπτώσεις. Θα την εφαρμόσουμε λ.χ. αργότερα στην περίπτωση του τυχαίου περιπατητή. Παρατηρήστε επίσης ότι το δείγμα που παίρνουμε είναι ανεξάρτητο του  $\beta$  και χρησιμοποιώντας τη σχέση (2.4) υπολογίζουμε τις μέσες τιμές για κάθε  $\beta$ .

### 2.1.2 IMPORTANCE SAMPLING

Από ότι είδαμε παραπάνω ένα πολύ μικρό μέρος του χώρου των καταστάσεων δίνει σημαντική συνεισφορά στον υπολογισμό του

$\langle Q \rangle$ . Αν επιλέξουμε το δείγμα με πιθανότητα

$$P_\mu = p_\mu = \frac{e^{-\beta E_\mu}}{Z} \quad (2.6)$$

περιμένουμε να δειγματοληψήσουμε ακριβώς μέσα στον υπόχωρο αυτό. Ο υπολογισμός του εκτιμητή (2.2) γίνεται με τη σχέση

$$Q_M = \frac{\sum_{i=1}^M Q_{\mu_i} (e^{-\beta E_{\mu_i}})^{-1} e^{-\beta E_{\mu_i}}}{\sum_{i=1}^M (e^{-\beta E_{\mu_i}})^{-1} e^{-\beta E_{\mu_i}}} = \frac{1}{M} \sum_{i=1}^M Q_{\mu_i}. \quad (2.7)$$

Η παραπάνω δειγματοληψία λέγεται δειγματοληψία με κριτήριο σημαντικότητας (importance sampling) και είναι ο τρόπος ο οποίος χρησιμοποιείται για προσομοιώσεις στατιστικών συστημάτων στην κανονική συλλογή. Το δείγμα εξαρτάται από τη θερμοκρασία  $\beta$  και ο υπολογισμός των μέσων τιμών (2.7) απαιτεί καινούργια δειγματοληψία κάθε φορά σε αντίθεση με την (2.4). Αυτή η δυσκολία όμως, στις περισσότερες περιπτώσεις, είναι πολύ μικρότερη από το πρόβλημα της επικάλυψης που συζητήθηκε στην παράγραφο 2.1.1.

## 2.2 ΔΙΑΔΙΚΑΣΙΕΣ MARKOV

Για να πάρουμε ένα δείγμα το οποίο κατανέμεται σύμφωνα με την κατανομή  $P_\mu$  δεν αρκεί να το κάνουμε απ' ευθείας. Λ.χ. αν επιχειρήσουμε να δημιουργήσουμε δείγμα με  $P_\mu = \frac{e^{-\beta E_\mu}}{Z}$  επιλέγοντας τυχαία κατάσταση  $\mu$  και δεχόμεθα ή απορρίπτουμε την εισαγωγή στο δείγμα με πιθανότητα  $P_\mu$  θα έχουμε ελάχιστη πιθανότητα η κατάσταση να γίνει αποδεκτή στο δείγμα. Οπότε θα βρεθούμε στην ίδια δυσκολία που βρήκαμε στην περίπτωση της απλής δειγματοληψίας. Για το λόγο αυτό θα χρησιμοποιήσουμε μια διαδικασία Markov για τη δημιουργία του δείγματος. Αυτή είναι μια διαδικασία όπου δεδομένου του συστήματος σε μια κατάσταση  $\mu$  παράγει με στοχαστικό τρόπο μια νέα κατάσταση  $\nu$ . Έτσι δημιουργείται μια αλυσίδα καταστάσεων

$$\mu_0 \rightarrow \mu_1 \rightarrow \mu_2 \rightarrow \dots \rightarrow \mu_M \quad (2.8)$$

η οποία θα αποτελέσει το ζητούμενο δείγμα  $\{\mu_i\} \equiv \{\mu_0, \mu_1, \mu_2, \dots, \mu_M\}$ . Φανταζόμαστε ότι η επιλογή της κατάστασης  $\mu_i$  γίνεται στο "χρόνο"  $i$ . Η πιθανότητα μετάβασης  $P(\mu \rightarrow \nu)$  (transition probability) στην κατάσταση  $\nu$  όταν το σύστημα είναι στην κατάσταση  $\mu$  πρέπει να ικανοποιεί τις παρακάτω συνθήκες:

1. Είναι ανεξάρτητη του "χρόνου".

2. Εξαρτάται μόνο από τις καταστάσεις  $\mu, \nu$  και όχι από τη διαδρομή που κάναμε μέχρι την κατάσταση  $\mu$ .
3. Ικανοποιείται η σχέση

$$\sum_{\nu} P(\mu \rightarrow \nu) = 1. \quad (2.9)$$

Προσοχή, συνήθως  $P(\mu \rightarrow \mu) > 0$  και το σύστημα έχει πιθανότητα να παραμείνει στην ίδια κατάσταση.

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

Η προσομοίωση Μόντε Κάρλο με τον τρόπο αυτό γίνεται επιλέγοντας κατάλληλα μια αρχική κατάσταση  $\mu_0$  για το σύστημα και εφαρμόζοντας το παραπάνω αλγόριθμο. Η μεγαλύτερη προσπάθεια επικεντρώνεται στο προσδιορισμό των πιθανοτήτων μετάβασης  $P(\mu \rightarrow \mu)$  έτσι ώστε η σύγκλιση 4. να επιτυγχάνεται γρήγορα.

Σημαντική είναι και η επιλογή της αρχικής κατάστασης  $\mu_0$ . Αν αυτή δεν είναι μια τυπική κατάσταση του τελικού δείγματος θα πρέπει να περάσει κάποιος χρόνος μέχρι το σύστημα να βρεθεί σε "κατάσταση ισορροπίας" όπου πια η διαδικασία Markov δειγματοληπτεί μέσα στη σωστή κατανομή. Ο χρόνος που απαιτείται (thermalization time) μπορεί να γίνει σημαντικό μέρος της προσπάθειάς μας αν γίνει λάθος επιλογή της  $\mu_0$  ή/και των  $P(\mu \rightarrow \mu)$ .

Απαραίτητη προϋπόθεση για να πετύχουμε το δείγμα να ακολουθεί την ζητούμενη κατανομή σε μια τέτοια διαδικασία είναι να ικανοποιεί το κριτήριο της **εργοδικότητας**. Αυτό σημαίνει ότι από κάθε κατάσταση  $\mu$  που επιλέγουμε, κάθε άλλη δυνατή κατάσταση  $\nu$  είναι προσβάσιμη μέσω της διαδικασίας με ένα πεπερασμένο αριθμό από βήματα. Αν αυτή η κατάσταση δεν ικανοποιείται και υπάρχουν σημαντικές περιοχές του χώρου των καταστάσεων στις οποίες δεν μπορούμε να δειγματοληπτήσουμε δεν θα είναι δυνατόν να πετύχουμε τη ζητούμενη κατανομή. Στην πράξη, επειδή δεδομένης της  $\mu$  οι καταστάσεις  $\nu$  για τις οποίες  $P(\mu \rightarrow \nu) > 0$  είναι ελάχιστες, πρέπει να είμαστε ιδιαίτερα προσεκτικοί ώστε ο αλγόριθμος που επιλέγουμε να μην παραβιάζει τη συνθήκη της εργοδικότητας<sup>3</sup>.

<sup>3</sup>Στην πράξη υπάρχουν αλγόριθμοι για του οποίους έχουμε παραβίαση της εργοδικότητας αλλά επειδή οι καταστάσεις που δεν είναι προσβάσιμες αποτελούν σύνολο "μέτρου μηδέν" στο χώρο των καταστάσεων, η παραβίαση δεν επηρεάζει τα αποτελέσματά μας. Αντίθετα, υπάρχουν περιπτώσεις όπου η συνθήκη δεν παραβιάζεται αλλά η πιθανότητα να φτάσω σε κάποιες περιοχές του χώρου των καταστάσεων είναι στην πράξη απαγορευτικά μικρή. Αυτό για παράδειγμα μπορεί να συμβεί κοντά σε μια μετάβαση φάσης πρώτης τάξης όπου το σύστημα δυσκολεύεται να περάσει από καταστάσεις της μιας φάσης στην άλλη.

### 2.3 ΣΥΝΘΗΚΗ ΛΕΠΤΟΜΕΡΟΥΣ ΙΣΟΖΥΓΗΣΗΣ

Από την εξίσωση (1.2) μπορούμε εύκολα να καταλάβουμε ότι για να βρεθεί το σύστημα σε κατάσταση ισορροπίας στην κατανομή  $p_\mu$ , οι πιθανότητες μετάβασης πρέπει να ικανοποιούν τη σχέση

$$\sum_{\nu} p_\mu P(\mu \rightarrow \nu) = \sum_{\mu} p_\nu P(\nu \rightarrow \mu). \quad (2.10)$$

Αυτό σημαίνει ότι ο ρυθμός με τον οποίο το σύστημα μεταβαίνει από την κατάσταση  $\mu$  σε κάποια άλλη, είναι ίσος με το ρυθμό με τον οποίο το σύστημα μεταβαίνει στην κατάσταση  $\mu$  από κάποια άλλη. Προφανώς, η σχέση (2.9) μας δίνει

$$p_\mu = \sum_{\mu} p_\nu P(\nu \rightarrow \mu). \quad (2.11)$$

Η παραπάνω συνθήκη είναι αναγκαία αλλά δεν είναι ικανή (δες κεφάλαιο 2.2.3 του [1]). Μια ικανή, αλλά όχι αναγκαία συνθήκη είναι η συνθήκη λεπτομερούς ισοζύγησης (detailed balance condition) η οποία όταν ικανοποιείται από τις πιθανότητες μετάβασης τότε είναι δυνατόν να δείξει κανείς ότι το σύστημα αργά ή γρήγορα θα φτάσει σε κατάσταση θερμικής ισορροπίας.

$$p_\mu P(\mu \rightarrow \nu) = p_\nu P(\nu \rightarrow \mu) \quad (2.12)$$

Αθροίζοντας και τα δύο μέλη της (2.12) προκύπτει η συνθήκη ισορροπίας (2.10). Για την κατανομή της κανονικής συλλογής (1.6) έχουμε

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{p_\nu}{p_\mu} = e^{-\beta(E_\nu - E_\mu)}. \quad (2.13)$$

Μπορεί να δειχτεί ότι αν οι πιθανότητες μετάβασης ικανοποιούν τις παραπάνω συνθήκες τότε η κατανομή ισορροπίας του συστήματος θα είναι η κατανομή Boltzman (1.6). Το πρόγραμμα της προσομοίωσης Μόντε Κάρλο μπορεί να συνοψιστεί στα επόμενα βήματα:

1. Γράφουμε λογισμικό το οποίο κωδικοποιεί κατάλληλα επιλεγμένες πιθανότητες μετάβασης  $P(\mu \rightarrow \nu)$  που ικανοποιούν την (2.13)
2. Επιλέγουμε κατάλληλη αρχική κατάσταση  $\mu_0$ .
3. Αφήνουμε το σύστημα να εξελιχθεί μέχρι να προσεγγίσουμε την κατανομή Boltzman (1.6) (thermalization).

4. Συλλέγουμε δεδομένα για τις παρατηρήσιμες ποσότητες  $Q$  και υπολογίζουμε τον εκτιμητή  $Q_M$  με την (2.7).
5. Σταματάμε μόλις πετύχουμε την επιθυμητή ακρίβεια.

Η εξίσωση (2.13) έχει πολλές λύσεις. Το ποια θα επιλέξουμε εξαρτάται από την αποδοτικότητά τους σε ένα συγκεκριμένο πρόβλημα και πρέπει να εξεταστεί προσεκτικά κατά περίπτωση. Παραδείγματα τέτοιων επιλογών είναι:

$$P(\mu \rightarrow \nu) = A \cdot e^{-\frac{1}{2}\beta(E_\nu - E_\mu)}, \quad (2.14)$$

$$P(\mu \rightarrow \nu) = A \cdot \frac{e^{-\beta(E_\nu - E_\mu)}}{1 + e^{-\beta(E_\nu - E_\mu)}}, \quad (2.15)$$

$$P(\mu \rightarrow \nu) = A \cdot \begin{cases} e^{-\beta(E_\nu - E_\mu)} & E_\nu - E_\mu > 0 \\ 1 & E_\nu - E_\mu \leq 0 \end{cases}, \quad (2.16)$$

για κατάλληλα επιλεγμένες καταστάσεις  $\nu \neq \mu$  και

$$P(\mu \rightarrow \mu) = 1 - \sum_{\nu} P(\mu \rightarrow \nu) \quad (2.17)$$

ενώ  $P(\mu \rightarrow \nu') = 0$  για οποιαδήποτε άλλη κατάσταση  $\nu'$ . Οι σταθερές  $A$  πρέπει να επιλεγούν κατάλληλα έτσι ώστε

$$\sum_{\nu \neq \mu} P(\mu \rightarrow \nu) < 1 \quad (2.18)$$

για να έχει νόημα η (2.17).

Η σχέση (2.18) μας δίνει μεγάλη ελευθερία στην επιλογή των πιθανοτήτων μετάβασης. Στην πράξη οι  $P(\mu \rightarrow \nu)$  σπάνε σε δύο κομμάτια

$$P(\mu \rightarrow \nu) = g(\mu \rightarrow \nu) A(\mu \rightarrow \nu) \quad (2.19)$$

τα οποία αντιστοιχούν σε διακριτά βήματα στον αλγόριθμο.

Η πιθανότητα  $g(\mu \rightarrow \nu)$  είναι η **πιθανότητα επιλογής** (transition probability) της κατάστασης  $\nu$  όταν το σύστημα βρίσκεται στην κατάσταση  $\mu$ . Το πρώτο βήμα δηλαδή είναι να επιλέξουμε μια κατάσταση  $\nu \neq \mu$  με πιθανότητα  $g(\mu \rightarrow \nu)$ .

Το δεύτερο βήμα είναι να επιλέξουμε με πιθανότητα  $A(\mu \rightarrow \nu)$  αν το σύστημα θα μεταβεί στην κατάσταση  $\nu$ . Αν η απάντηση είναι όχι, τότε παραμένουμε στην κατάσταση  $\mu$ . Με τον τρόπο αυτό ικανοποιείται η σχέση (2.17). Οι πιθανότητες  $A(\mu \rightarrow \nu)$  ονομάζονται **λόγοι αποδοχής**.

Ο στόχος μας επικεντρώνεται στην εύρεση αλγόριθμου τέτοιου ώστε οι πιθανότητες επιλογής να δίνουν τους μέγιστους δυνατούς

λόγους αποδοχής για καταστάσεις  $\nu$  ασυσχέτιστες κατά το μέγιστο δυνατόν από την κατάσταση  $\mu$ . Ιδανική περίπτωση είναι να έχω  $A(\mu \rightarrow \nu) = 1$  για όλα τα  $\nu$  για τα οποία  $g(\mu \rightarrow \nu) > 0$ . Αυτό συμβαίνει για παράδειγμα στους cluster αλγόριθμους συστημάτων spin (λ.χ. πρότυπα Ising, Potts) όπως ο αλγόριθμος του Wolff που θα μελετήσουμε αργότερα.

## 2.4 ΠΡΟΒΛΗΜΑΤΑ

1. Αποδείξτε ότι οι σχέσεις (2.14)–(2.16) ικανοποιούν την (2.13).



## ΚΕΦΑΛΑΙΟ 3

# Ο ΤΥΧΑΙΟΣ ΠΕΡΙΠΑΤΗΤΗΣ

Στο κεφάλαιο αυτό θα μελετήσουμε τη διαδρομή που ακολουθεί ένας ... μεθυσμένος όταν αργά το βράδυ αποφασίσει να επιστρέψει σπίτι του από το σημείο που βρίσκεται. Λόγω της τύφλας του, τα βήματα που εκτελεί είναι ασυσχέτιστα μεταξύ τους και προς τυχαία διεύθυνση. Αυτά τα δυο χαρακτηριστικά θα ορίσουν το απλούστερο πρότυπο που θα μελετήσουμε. Τα πρότυπα αυτά και οι γενικεύσεις τους έχουν άμεση σχέση με τη συμπεριφορά φυσικών συστημάτων (κίνηση Brown, διάχυση, κίνηση impurities στο πλέγμα, ιδιότητες μακρών ελαστικών μακρομορίων σε μακροσκοπικές αποστάσεις, στοχαστικές διαδικασίες). Στη φυσική στοιχειωδών σωματιδίων παρουσιάζονται στη γεωμετρία των τυπικών διαδρομών των μποζονίων στα ολοκληρώματα διαδρομών του Feynman. Οι έννοιες που αναπτύσσονται από τη μελέτη του τυχαίου περιπατητή παίζουν σημαντικό ρόλο στην κατανόηση των στατιστικών ιδιοτήτων της ευκλείδειας κβαντικής θεωρίας πεδίου, θεωρίας τυχαίων (υπερ)επιφανειών, θεωρία χορδών κλπ.

Η γεωμετρία μιας τυπικής διαδρομής του απλού τυχαίου περιπατητή παύει να είναι κλασσική για δύο λόγους. Το μήκος της διαδρομής που διανύει ο περιπατητής είναι ανάλογος της **τετραγωνικής ρίζας** του χρόνου, με αποτέλεσμα να μην ισχύει η κλασσική σχέση  $r = vt$ . Ο δεύτερος λόγος είναι ότι η γεωμετρία του χώρου που πατάει ο περιπατητής (με τη στατιστική έννοια) έχει δομή fractal με διάσταση μεγαλύτερη της μονάδας<sup>1</sup>, δηλαδή δεν είναι ένα απλό μονοδιάστατο μονοπάτι. Παρόμοια φαινόμενα παρουσιάζονται στις τυχαίες επιφάνειες και κβαντικές θεωρίες πεδίου όπου η απόκλιση από την κλασσική συμπεριφορά μπορεί να γίνει κατανοητή με κατάλληλες γενικεύσεις των παραπάνω ιδεών.

Για περαιτέρω μελέτη παραπέμπουμε στα συγγράμματα [4, 6, 14, 15].

---

<sup>1</sup>Για την ακρίβεια η διάσταση Hausdorff  $d_H = 2$ .

### 3.1 ΤΥΧΑΙΕΣ ΔΙΑΔΡΟΜΕΣ

Υποθέτουμε ότι ένα σωματίο μπορεί να βρεθεί στις θέσεις ενός τετραγωνικού πλέγματος στις δύο διαστάσεις (στο επίπεδο). Σε κάποια χρονική στιγμή βρίσκεται σε μια από τις θέσεις του πλέγματος και αφού ισορροπήσει σε αυτή μπορεί να πηδήξει τυχαία σε μια γειτονική θέση όπου με τη σειρά του κάθετα για κάποιο χρόνο και ισορροπεί. Καθώς ισορροπεί, η ορμή που είχε και το βοήθησε να κάνει το πήδημα χάνεται οπότε χάνει τη μνήμη της θέσης που βρισκόταν προηγουμένως. Η διαδικασία αυτή επαναλαμβάνεται συνεχώς. Ο μηχανισμός του φαινομένου δεν θα μας απασχολήσει<sup>2</sup>, αναζητούμε ένα απλό φαινομενολογικό μοντέλο της διαδικασίας.

Υποθέτουμε ότι το σωματίο πηδάει με ίση πιθανότητα σε μια από τις πλησιέστερες θέσεις του πλέγματος κάθε φορά που περνάει σταθερός χρόνος  $\tau$ . Οι θέσεις του πλέγματος απέχουν μεταξύ τους απόσταση  $a$  (πλεγματική σταθερά). Το διάνυσμα που περιγράφει την μεταβολή της θέσης του σωματιδίου στο  $i$ -οστό πήδημα είναι μια τυχαία μεταβλητή  $\vec{\xi}_i$  μέτρου  $|\vec{\xi}_i| = a$ . Δηλαδή δεδομένης της θέσης  $\vec{r}_k$  του σωματιδίου τη χρονική στιγμή  $t_k = k\tau$ , η θέση του  $\vec{r}_{k+1}$  τη χρονική στιγμή  $t_{k+1} = (k+1)\tau = t_k + \tau$  θα είναι

$$\vec{r}_{k+1} = \vec{r}_k + \vec{\xi}_k \quad (3.1)$$

όπου

$$\vec{\xi}_k = \begin{cases} a\hat{x} & \text{με πιθανότητα } \frac{1}{4} \\ -a\hat{x} & \text{με πιθανότητα } \frac{1}{4} \\ a\hat{y} & \text{με πιθανότητα } \frac{1}{4} \\ -a\hat{y} & \text{με πιθανότητα } \frac{1}{4} \end{cases}. \quad (3.2)$$

Σύμφωνα με τα παραπάνω, οι τιμές των  $\vec{\xi}_i$  είναι ανεξάρτητες από την τιμή της θέσης που είχε προηγουμένως το σωματίο. Οπότε οι τιμές  $\vec{\xi}_i$  και  $\vec{\xi}_j$  είναι ασυσχέτιστες για  $i \neq j$  και ισχύει ότι

$$\langle \vec{\xi}_i \cdot \vec{\xi}_j \rangle = \langle \vec{\xi}_i \rangle \cdot \langle \vec{\xi}_j \rangle. \quad (3.3)$$

Οι τιμές που παίρνουν τα  $\vec{\xi}_i$  είναι ισοπίθανες οπότε επειδή οι θετικές και αρνητικές τιμές συμβαίνουν το ίδιο συχνά οι θετικοί και αρνητικοί όροι στον υπολογισμό της  $\langle \vec{\xi}_i \rangle$  αλληλοαναιρούνται και έχουμε

$$\langle \vec{\xi}_i \rangle = \vec{0}. \quad (3.4)$$

<sup>2</sup>Μπορεί λ.χ. να είναι θερμικά διεγερμένα ηχητικά κύματα που δίνουν στο σωματίο την απαραίτητη ενέργεια για το πήδημα, το κβαντικό φαινόμενο σήραγγας κλπ

οπότε  $\langle \vec{\xi}_i \cdot \vec{\xi}_j \rangle = 0$  για  $i \neq j$ . Επειδή το μήκος των διανυσμάτων είναι σταθερό  $|\vec{\xi}_i| = a$  παίρνουμε επομένως τη σχέση

$$\langle \vec{\xi}_i \cdot \vec{\xi}_j \rangle = a^2 \delta_{i,j} \quad (3.5)$$

Η πιθανότητα να εμφανιστεί μια διαδρομή  $C_N$  μήκους είναι<sup>3</sup>

$$p(C_N) = \frac{1}{z^N} \quad (3.6)$$

όπου  $z = 4$  ο αριθμός των πλησιεστέρων γειτόνων μιας πλεγματικής θέσης. Η πιθανότητα αυτή εξαρτάται από το μήκος της διαδρομής και όχι από τη γεωμετρία της. Αυτό προκύπτει από την προφανή σχέση  $p(C_{N+1}) = \frac{1}{z} p(C_N)$  μια και υπάρχουν ακριβώς  $z$  ισοπίθανες περιπτώσεις. Οπότε η συνάρτηση επιμερισμού είναι

$$Z_N = z^N. \quad (3.7)$$

και είναι ίση με τον αριθμό των διαφορετικών διαδρομών μήκους  $N$ . Λ.χ. αν μοντελοποιούμε μια πολυμερική αλυσίδα, η (3.7) δίνει τη συνάρτηση επιμερισμού της.

Μέσα σε χρόνο  $t = N\tau$  το σωματίο μετατοπίζεται κατά

$$\vec{R} = \sum_{i=1}^N \vec{\xi}_i. \quad (3.8)$$

Η μέση τιμή της μετατόπισης είναι 0

$$\langle \vec{R} \rangle = \sum_{i=1}^N \langle \vec{\xi}_i \rangle = \vec{0}. \quad (3.9)$$

Η μέση τιμή της μετατόπισης στο τετράγωνο είναι

$$\langle R^2 \rangle = \langle \vec{R} \cdot \vec{R} \rangle = \sum_{i,j=1}^N \langle \vec{\xi}_i \cdot \vec{\xi}_j \rangle = a^2 \sum_{i,j=1}^N \delta_{i,j} = a^2 N \quad (3.10)$$

Βγάζουμε λοιπόν το πολύ σημαντικό αποτέλεσμα ότι ο τυχαίος περιπατητής απομακρύνεται πολύ αργά από το αρχικό του σημείο

$$R_{rms} = \sqrt{\langle R^2 \rangle} = a\sqrt{N} \quad (3.11)$$

Για σωματίο με μη μηδενική μέση ταχύτητα (δες πρόβλημα) αναμένεται  $R_{rms} \propto N$ .

<sup>3</sup> Δηλ. μετά από χρόνο  $t = N\tau$ , όχι το φυσικό μήκος της καμπύλης. Μετράμε και τα πηδήματα πάνω στα links που έχει ήδη επισκεφτεί το σωματίο.

Η παραπάνω σχέση ορίζει έναν κρίσιμο εκθέτη  $\nu$

$$\langle R^2 \rangle \sim N^{2\nu}, \quad (3.12)$$

όπου το σύμβολο  $\sim$  σημαίνει ασυμπτωτική συμπεριφορά για  $N \rightarrow \infty$ . Για ένα κλασικό περιπατητή  $\nu = 1$  ενώ για τον τυχαίο περιπατητή  $\nu = \frac{1}{2}$ .

Παραλλαγές του Τυχαίου Περιπατητή (Random Walker - RW για συντομία) αποτελούν ο Μη Επιστρέφων Τυχαίος Περιπατητής (Non Reversal Random Walk - NRRW για συντομία) και ο Αυτοαποφεύγων Τυχαίος Περιπατητής (Self Avoiding Random Walk - SAW για συντομία). Ο NRRW ορίζεται όταν τα διανύσματα  $\vec{\xi}_i$  επιλέγονται ισοπίθανα αποκλείοντας την επιστροφή στο αρχικό σημείο. Ο SAW είναι ένας NRRW όπου κάθε φορά που ο περιπατητής επισκέπτεται μία πλεγματική θέση που έχει ήδη επισκεφτεί, το ... περπάτημα σταματάει. Επί πλέον είναι δυνατόν να προσθέσουμε, εκτός από την άπειρη απωστική ενέργεια για σημεία που συμπίπτουν, και μια ελκτική ενέργεια  $-\epsilon$  για κάθε ζευγάρι σημείων που ανήκουν στη διαδρομή και είναι πλησιέστεροι γείτονες. Κάθε επιτρεπόμενη διαδρομή θα ζυγίζεται τότε με πιθανότητα κατά Boltzmann σύμφωνα με την (1.6).

Για τον NRRW η σχέση (3.12) είναι ίδια με τον RW, δηλαδή  $\nu = \frac{1}{2}$ . Παρόλο που οι λεπτομέρειες των διαδρομών σε μικρές αποστάσεις είναι διαφορετικές, οι ιδιότητες τους σε μακροσκοπικές κλίμακες είναι παρόμοιες. Είναι περίπτωση συστημάτων που ανήκουν στην ίδια κλάση παγκοσμιότητας (universality class) σύμφωνα με τη συζήτηση στην ενότητα 1.5.

Δεν συμβαίνει το ίδιο και για τον SAW. Για το σύστημα αυτό δεν είναι δυνατόν να γίνουν αναλυτικοί υπολογισμοί αλλά εικάζεται ότι

$$\langle R^2 \rangle^{\text{SAW}} \sim N^{2\nu} \quad \nu = \frac{3}{4}. \quad (3.13)$$

Οι τυπικές διαδρομές στο πρότυπο αυτό είναι μεγαλύτερες από τον RW. Αν εισάγουμε την έλξη μεταξύ πλησιεστέρων γειτόνων σύμφωνα με τα παραπάνω, τότε υπάρχει κρίσιμη θερμοκρασία  $\beta_c$  τέτοια ώστε για θερμοκρασία  $\beta < \beta_c$  να έχουμε παρόμοια συμπεριφορά με την (3.13) ενώ για  $\beta > \beta_c$  οι διαδρομές να "συνθλίβονται" από την αλληλεπίδραση και να παίρνουμε  $\nu = 1/3 < \nu_{RW}$ . Για  $\beta = \beta_c$  έχουμε  $\nu = \frac{1}{2}$ . Για περισσότερες λεπτομέρειες παραπέμπουμε στο βιβλίο των Binder-Heermann [4].

Σύμφωνα με τη συζήτηση στο προηγούμενο κεφάλαιο, είμαστε έτοιμοι να προγραμματίσουμε τον τυχαίο περιπατητή. Ο αλγόριθμος είναι πολύ απλός, σημειώνουμε πως οι μετρήσεις γίνονται με απλή δειγματοληψία:

1. Επιλέγουμε αριθμό τυχαίων διαδρομών που θα παράγουμε.
2. Επιλέγουμε αριθμό βημάτων κάθε διαδρομής.
3. Επιλέγουμε αρχική θέση των διαδρομών
4. Σε κάθε βήμα μιας διαδρομής επιλέγουμε με ίση πιθανότητα κίνηση προς τα δεξιά, αριστερά, πάνω και κάτω.
5. Στο τέλος κάθε διαδρομής, μετράμε τις ιδιότητες που μας ενδιαφέρουν ( $\vec{R}$ ,  $R^2$ , κλπ).
6. Στο τέλος υπολογίζουμε τις μέσες τιμές και σφάλματα στις ποσότητες που μετρήσαμε.

Η μόνη έννοια που χρειάζεται να εξηγήσουμε πως θα προγραμματίσουμε είναι αυτή της επιλογής της “τυχαίας διεύθυνσης”.

## 3.2 (ΨΕΥΔΟ)ΤΥΧΑΙΟΙ ΑΡΙΘΜΟΙ

Η παραγωγή ψευδοτυχαίων αριθμών είναι στην καρδιά της προσομοίωσης Μόντε Κάρλο. Για να κατασκευάσουμε μια ακολουθία καταστάσεων με μια διαδικασία Markov χρειάζεται να επιλέξουμε την επόμενη κατάσταση με μια κατανομή πιθανότητας. Το κεφάλαιο αυτό θα μπορούσε να αποτελέσει ξεχωριστό βιβλίο γιατί τα σφάλματα που μπορεί να προκύψουν από λάθος επιλογή αλγόριθμου μπορεί να είναι σημαντικά και δύσκολο να ανακαλυφθούν.

Η παραγωγή των ψευδοτυχαίων αριθμών είναι κατά κανόνα ντετερμινιστική: Στη γεννήτρια τυχαίων αριθμών (δηλ. στον αλγόριθμο) δίνονται κάποιες αρχικές συνθήκες από τις οποίες η παραγωγή των αριθμών αφήνεται να εξελιχθεί στο “χρόνο”. Ο επόμενος αριθμός προκαθορίζεται από την κατάσταση της γεννήτριας εξ’ ου και η ντετερμινιστική εξέλιξη. Ίδιες αρχικές συνθήκες δίνουν την ίδια ακριβώς ακολουθία αριθμών. Η εξέλιξη αυτή όμως είναι χασοτική. Ελάχιστα διαφορετικές αρχικές συνθήκες φτιάχνουν ακολουθίες που αποκλίνουν εκθετικά στο χρόνο η μία από την άλλη. Για παρόμοιο λόγο, ο αριθμός που παράγεται σε κάθε βήμα θεωρείται πως είναι ασυσχέτιστος από τον προηγούμενο. Εδώ βρίσκεται και το αδύνατο σημείο που προκαλεί τα δυσκολότερα προβλήματα γιατί στις προβληματικές γεννήτριες βρίσκονται λεπτοί συσχετισμοί που είναι δύσκολο να προσδιοριστούν. Πραγματικά τυχαίοι αριθμοί (χρήσιμοι στην κρυπτογραφία) μπορούν να παραχθούν από ειδικές συσκευές που βασίζονται στο χρόνο διάσπασης ραδιενεργών υλικών.

Οι πιο δημοφιλείς γεννήτριες λόγω της απλότητάς τους είναι οι γεννήτριες modulo (D.H. Lehmer, 1951). Αυτές χρειάζονται μόνο έναν αριθμό  $x_0$  (seed) ως αρχική συνθήκη και από τον αριθμό  $x_{i-1}$  παράγουν τον  $x_i$  από τη σχέση

$$x_i = ax_{i-1} + c \pmod{m} \quad (3.14)$$

για κατάλληλα επιλεγμένες τιμές των  $a$ ,  $c$  και  $m$ . Η κατάσταση της γεννήτριας προσδιορίζεται πλήρως από ένα μόνο αριθμό, την τρέχουσα τιμή του  $x_i$ . Υπάρχει βιβλιογραφία σχετικά με τη σωστή επιλογή των  $a$ ,  $c$  και  $m$ , εδώ ας σημειώσουμε πως αυτή είναι διαφορετική για γλώσσα C ή FORTRAN και για συστήματα που είναι 32-bit ή 4-bit. Για λεπτομέρειες παραπέμπουμε στο ειδικό κεφάλαιο των Numerical Recipes [16].

Η τιμή του  $m$  καθορίζει την μέγιστη περίοδο της γεννήτριας. Είναι σαφές πως αν σε κάποια στιγμή της ακολουθίας πάρω τον ίδιο αριθμό που πήρα πριν από  $k$  βήματα, η γεννήτρια θα αρχίσει να παράγει ακριβώς την ίδια ακολουθία και αυτό θα συνεχίζεται περιοδικά. Αφού έχω  $m$  δυνατούς διαφορετικούς αριθμούς, η περίοδος είναι το πολύ  $m$ . Για κακή επιλογή του  $m$  η περίοδος θα είναι πολύ μικρότερη. Η τιμή του  $m$  όμως δεν μπορεί να είναι αυθαίρετα μεγάλη αφού σε οποιοδήποτε υπολογιστή υπάρχει ένας μέγιστος ακεραίος που καθορίζεται από τον αριθμό των bits που χρησιμοποιεί να τον αναπαραστήσει. Για ακεραίους 4-bytes (32-bits) χωρίς πρόσημο ο μέγιστος αριθμός είναι  $2^{32}$ . Καλή επιλογή των  $a$ ,  $c$  και  $m$  μπορεί να αποδειχθεί (βλ. Knuth[17]) ότι δίνει ακολουθία που είναι αναδιάταξη  $\{\pi_0, \pi_1, \dots, \pi_{m-1}\}$  των αριθμών  $0, 1, \dots, m-1$ . Μια τέτοια επιλογή δίνει αρκετά καλούς τυχαίους αριθμούς για τις περισσότερες εφαρμογές αλλά για σοβαρούς υπολογισμούς θα πρέπει να ανατρέξουμε σε μια προσεκτικά επιλεγμένη γεννήτρια τυχαίων αριθμών. Δίνεται επίσης η γενική συμβουλή να αποφεύγονται οι γεννήτριες που παρέχονται από το λειτουργικό σύστημα (λ.χ. `drand48()` στο UNIX/Linux). Για κώδικα ο αναγνώστης μπορεί να ανατρέξει στις αναφορές [1], [2], [16], [18]. Για portable προγράμματα προτείνονται οι γεννήτριες RANLUX [18] και η γεννήτρια των Marsaglia, Zaman and Tsang. Πρόγραμμα Fortran για την πρώτη διατίθεται από την ιστοσελίδα του μαθήματος ενώ για την τελευταία μπορεί να βρεθεί στο βιβλίο/site του Berg [2].

Για να γίνει κατανοητή η χρήση των γεννητριών αλλά και να αναδειχθεί ένα βασικό πρόβλημα των modulo γεννητριών παραθέτουμε παρακάτω τις συναρτήσεις `naiveran()` και `drandom()`. Η πρώτη είναι προβληματική και θα μελετήσουμε τους συσχετισμούς που κρύβονται στις ακολουθίες τυχαίων αριθμών που παράγει καθώς και την επίδραση τους στις προσομοιώσεις του τυχαίου περι-

πατητή. Η δεύτερη είναι αυτή που θα χρησιμοποιήσουμε για τους σκοπούς του μαθήματος.

### 3.2.1 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Στο σημείο αυτό μπορούμε να κάνουμε μερικές παρατηρήσεις σχετικά με τις μεθόδους προγραμματισμού και τις συμβάσεις που θα ακολουθήσουμε στο βιβλίο αυτό. Η γλώσσα προγραμματισμού που επιλέγεται είναι η C και το λειτουργικό σύστημα πάνω στο οποίο θα δουλέψουμε είναι το Unix (όπως λ.χ. το GNU/Linux). Ο αναγνώστης που δεν έχει καθόλου οικειότητα με τα παραπάνω θα πρέπει να "κολυπήσει στα βαθιά" ή να μελετήσει παράλληλα τα βασικά για τα παραπάνω. Στην ιστοσελίδα του μαθήματος δίνονται αναφορές σε ελεύθερα διαθέσιμα εγχειρίδια σε διάφορα επίπεδα παρουσίασης. Βασική χρήση του λειτουργικού είναι αρκετή για την εκτέλεση των βασικών διαδικασιών, αν και όπως θα φανεί στα επόμενα, η αποτελεσματική παραγωγή και ανάλυση μεγάλης ποσότητας δεδομένων απαιτεί τη συνδυαστική χρήση πολλών εργαλείων που διαθέτει το σύστημα. Το πλεονέκτημα της χρήσης των κατάλληλων εργαλείων GNU και γενικότερα του ελεύθερου λογισμικού είναι η υψηλή ποιότητά τους, η ευελιξία και συμβατότητα μεταξύ τους που επιτρέπει στο μελετητή να τα χρησιμοποιήσει εύκολα και αποδοτικά για να εκτελέσει σύνθετες εργασίες. Αναφέρουμε ενδεικτικά μερικά από τα προγράμματα που κρίνονται ιδιαίτερα χρήσιμα:

- **GNU C Compiler:** Ο μεταγλωττιστής (compiler) είναι το βασικό εργαλείο που μεταφράζει εντολές από μία ανώτερη γλώσσα προγραμματισμού σε "γλώσσα μηχανής", δηλ. σε ψηφιακές εντολές που μεταφέρονται στη μνήμη και εκτελούνται από τον επεξεργαστή του υπολογιστή. Είναι πολύ σημαντικό, ιδιαίτερα όταν οι πόροι του συστήματος είναι πολύτιμοι, η χρήση του μεταγλωττιστή να γίνεται με τον καλύτερο δυνατό τρόπο χρησιμοποιώντας όλες τις δυνατότητες βελτιστοποίησης (optimization) που αφορούν το πρόγραμμά μας. Οι καλοί μεταγλωττιστές συνήθως εκτελούν τη βελτιστοποίηση αυτόματα με την προσθήκη κατάλληλων "διακοπών" (switches, options) κατά την εκτέλεση της μεταγλώττισης. Πολλές φορές όμως η χρήση ιδιαίτερων διαδικασιών βελτιστοποίησης απαιτείται να ζητηθεί ρητά από τον χρήστη. Συνιστάται λοιπόν η καλή μελέτη των εγχειριδίων ενός μεταγλωττιστή προτού γίνει η χρήση του. Τονίζεται ότι αλλάζοντας πλατφόρμα hardware/software οι συνθήκες και ο καλύτερος διαθέσιμος μεταγλωττιστής συνήθως αλλάζουν

και απαιτείται να μελετάται με προσοχή το νέο περιβάλλον. Η επιλογή του GNU C Compiler έγινε λόγω της ελεύθερης διαθεσιμότητάς του στα περισσότερα λειτουργικά συστήματα και επεξεργαστές. Στα συστήματα Linux είναι πάντα προεγκατεστημένος αλλά μπορείτε να τον κατεβάσετε ελεύθερα και να τον εγκαταστήσετε όταν αυτό δεν συμβαίνει (ακόμα σε περιβάλλον Windows). Σε υπολογιστές με επεξεργαστή Intel συνιστάται η χρήση του χωρίς χρέωση διαθέσιμου (για Linux) μεταγλωττιστή της Intel.

- **GNU Make utility:** Πρόγραμμα το οποίο οργανώνει τη μεταγλώττιση μεγάλων προγραμμάτων. Στην περίπτωση αυτή μία ομάδα προγραμματιστών μοιράζει τα διάφορα κομμάτια του κώδικα σε πολλά αρχεία/καταλόγους έτσι ώστε να γίνεται δυνατός ο καταμερισμός της εργασίας αποτελεσματικά, καθώς και να είναι δυνατή η διόρθωση/συντήρηση του κώδικα χωρίζοντάς τον σε εννοιολογικά διαφορετικά τμήματα (modules) που είναι ανεξάρτητα μεταξύ τους. Το πρόγραμμα make μεταγλωττίζει μόνο τα κομμάτια εκείνα του κώδικα που έχουν ενημερωθεί σώζοντας πολύτιμο χρόνο, καθώς και κωδικοποιούν πολύπλοκες εντολές μεταγλώττισης και σύνδεσης με βιβλιοθήκες που θα ήταν πρακτικά αδύνατον να επαναλαμβάνουμε κάθε φορά. Το πρόγραμμα αυτό θα το χρησιμοποιήσουμε στο επόμενο κεφάλαιο.
- **emacs editor:** Χρήσιμος επεξεργαστής κειμένου (...και όχι μόνο), ιδιαίτερα για προγραμματιστές, ο οποίος “καταλαβαίνει” αρκετές γλώσσες προγραμματισμού και παρέχει εργαλεία για την αποτελεσματική επεξεργασία του κειμένου του προγράμματος. Ανάμεσα στα πλεονεκτήματά του είναι η εκπληκτική προσαρμοστικότητα του στις ανάγκες και επιθυμίες του χρήστη, η ολοκλήρωση των λειτουργιών του με το λειτουργικό σύστημα (αρχικά είχε σχεδιαστεί με την προοπτική να αποτελέσει ο ίδιος λειτουργικό σύστημα), η δυνατότητα επεξεργασίας αρχείων που βρίσκονται σε απομακρυσμένο υπολογιστή μέσω δικτύου, η διαθεσιμότητά του ελεύθερα σε πολλές πλατφόρμες κλπ.
- **shell scripting:** Μεγάλος αριθμός επαναλαμβανόμενων εντολών μπορούν να κωδικοποιηθούν σε “σενάρια φλοιού” (shell scripts)<sup>4</sup> από εντολές που είναι ίδιες με αυτές που ο χρήστης δίνει όταν αλληλεπιδρά με το λειτουργικό σύστημα.

<sup>4</sup>Τα σενάρια είναι προγράμματα που γράφονται με εντολές που διαβάζονται από άνθρωπο και δεν χρειάζονται ξεχωριστή μεταγλώττιση. Τέτοιες γλώσσες είναι μεταξύ άλλων η awk, perl, tcsh, bash, javascript, emacs lisp, python κλπ.



- **grep, sort:** Εργαλεία κειμένου με τα οποία ο χρήστης φιλτράρει και ταξινομεί δεδομένα. Αναφέρονται ειδικά λόγω της μεγάλης σημασίας τους στην εργασία μας.
- **awk:** Εξαιρετικά δυνατό εργαλείο για μεταποίηση δεδομένων από τη γραμμή δεδομένων. Πολλές χρήσιμες εργασίες γίνονται ευέλικτα με “προγράμματα μιας γραμμής” (one liners).
- **gnuplot (plotting-fitting):** Ευέλικτο πρόγραμμα για την αποτελεσματική εικονοποίηση των αποτελεσμάτων μας. Έχει το πλεονέκτημα της εύκολης χρήσης εργαλείων του φλοιού για φιλτράρισμα και επεξεργασία δεδομένων, εύκολη χρήση από τη γραμμή εντολών για την χρήση του από άλλα προγράμματα για την επεξεργασία πολλών γραφημάτων. Κατασκευάζει γραφήματα δημοσιεύσιμης ποιότητας. Επίσης έχει ενσωματωμένο πρόγραμμα για μη γραμμική προσαρμογή δεδομένων (non-linear fitting). Διατίθεται ελεύθερα.
- **redirection/piping:** Μια σημαντική συνιστώσα των συστημάτων Unix. Ο χρήστης μπορεί να ανακατευθύνει την έξοδο προγραμμάτων σε οποιοδήποτε αρχείο (συμπεριλαμβανομένων και “συσκευών” που στο Unix είναι αρχεία) ή/και στην είσοδο άλλων προγραμμάτων. Το τελευταίο χρησιμοποιείται για την κατασκευή χρήσιμων φίλτρων δεδομένων και θα τα χρησιμοποιήσουμε εκτενώς στα επόμενα.

Ο φοιτητής που δεν είναι εξοικειωμένος με τα παραπάνω προγράμματα/έννοιες καλείται τώρα ψάξει στα κατάλληλα εγχειρίδια και στο διαδίκτυο. Στην ιστοσελίδα του μαθήματος θα βρίσκεται αρκετό υλικό το οποίο είναι ελεύθερα διαθέσιμο.

Αξίζει να σημειωθούν μερικά ακόμα προγράμματα τα οποία αν και δεν θα τα χρησιμοποιήσουμε πολύ στην εισαγωγική αυτή μας προσέγγιση, ενδέχεται να φανούν χρήσιμα σε φοιτητές που θα εκτελέσουν πιο προχωρημένες εργασίες σε ανάλυση δεδομένων και προσομοίωση Μόντε Κάρλο. Οι διορθωτές προγραμμάτων (debuggers) `gdb` και η πολύ ωραία γραφική interface `ddd`, τα εργαλεία χρονομέτρησης `time` και `gprof` και η γλώσσα σεναρίων `perl`. Τα προγράμματα εμπορίου `Mathematica`, `Maple` και `Matlab` προσφέρουν στο μελετητή μεγάλες δυνατότητες στην ανάλυση των δεδομένων, ακόμα και για τη χρήση τους ως γλώσσα προγραμματισμού των προγραμμάτων παραγωγής των δεδομένων. Για εικονικές διαδραστικές εφαρμογές απλών προσομοιώσεων που δεν είναι αριθμητικά απαιτητικές συστήνουμε το πακέτο `opensourcephysics.org` που απαιτούν τη χρήση `Java`[5].

Η πρώτη εργασία που κάνει κάποιος που προγραμματίζει σε ένα καινούργιο περιβάλλον είναι η εκτέλεση ενός προγράμματος

“Hello World”. Αυτό το βήμα είναι σημαντικό παρά την απλότητά του. Μας εξοικειώνει με την επεξεργασία και διαχείριση αρχείων στο καινούργιο σύστημα, την χρήση του μεταγλωττιστή και τέλος την εκτέλεση των μεταγλωττισμένων προγραμμάτων. Το πρόγραμμα στο αρχείο `hello.c` κάνει ακριβώς αυτό.

```

/***** hello.c *****/
#include <stdlib.h>
#include <stdio.h>

int main(){

    printf("Hello World!\n");

}

```

Η μεταγλώττιση γίνεται με την εντολή<sup>5</sup>:

```
% cc hello.c -o hello
```

η οποία παράγει το αρχείο `hello` το οποίο είναι εκτελέσιμο. Η εντολή

```
% ./hello
```

παράγει στην έξοδο το μήνυμα `Hello World!` Οι γραμμές

```

#include <stdlib.h>
#include <stdio.h>

```

δεν είναι εντολές της γλώσσας C αλλά του προεπεξεργαστή (preprocessor) `cpp` που είναι ένας απλός επεξεργαστής κειμένου. Η εντολή `#include` (το `#` στην πρώτη στήλη) αντικαθιστά τη γραμμή αυτή με τα περιεχόμενα του αρχείου `stdlib.h` και `stdio.h` αντίστοιχα. Τα αρχεία αυτά είναι σε ειδικούς καταλόγους στο σύστημα (συνήθως στους καταλόγους `/usr/include`, `/include`) και αυτό υποδηλώνεται από τον εγκλεισμό του ονόματος του αρχείου σε `<...>`<sup>6</sup>. Οι κατάλογοι που ψάχνει ο `cpp` μπορούν να ρυθμιστούν με κατάλληλους διακόπτες (switches, options) `-I` στον μεταγλωττιστή. Ο προεπεξεργαστής καλείται από την εντολή μεταγλώττισης και δεν χρειάζεται

<sup>5</sup>Η εντολή `cc` στο σύστημα GNU/Linux αναφέρεται στον GNU C compiler `gcc`. Σε άλλα συστήματα ενδέχεται να αναφέρεται σε άλλο μεταγλωττιστή.

<sup>6</sup>Αν τα ονόματα των αρχείων ήταν εγκλεισμένα μεταξύ διπλών εισαγωγικών `"..."` τότε η διαδρομή αναφέρεται στη σχετική ή απόλυτη διαδρομή προς τον κατάλογο σύμφωνα με τις συμβάσεις του λειτουργικού.

να την εκτελέσουμε ξεχωριστά (αν και θα μπορούσαμε). Είναι σημαντικό να καταλάβουμε ότι ο προεπεξεργαστής δεν μεταγλωττίζει αλλά τροποποιεί το κείμενο του προγράμματος, και το τροποποιημένο αυτό κείμενο είναι που επεξεργάζεται ο μεταγλωττιστής. Τα περιεχόμενα των συγκεκριμένων αρχείων είναι δηλώσεις μεταβλητών και macros που χρησιμοποιούν οι βιβλιοθήκες συστήματος και εισόδου/εξόδου (λ.χ. δήλωση της συνάρτησης `printf()` η οποία τυπώνει στην καθιερωμένη έξοδο την ακολουθία χαρακτήρων `Hello World!`).

Η εκτέλεση εντολών στη γλώσσα C αρχίζει από τη συνάρτηση `main()` η οποία συνήθως καλεί άλλες συναρτήσεις για την εκτέλεση ενός πολύπλοκου προγράμματος. Τη συνάρτηση αυτή δεν χρειάζεται φυσικά να τη δηλώσουμε. Η συνάρτηση `printf()` παίρνει εδώ ως όρισμα μια ακολουθία χαρακτήρων την οποία τυπώνει στην καθιερωμένη έξοδο (`standard output` ή `stdout`). Προσέξτε πως για να τυπωθεί μια νέα γραμμή πρέπει να το δηλώσουμε ρητά με τον ειδικό χαρακτήρα `\n`. Κάθε εντολή στην γλώσσα C πρέπει να τελειώνει με το χαρακτήρα `;` και μπορεί να συνεχίζεται σε πολλές γραμμές. Ό,τι περικλείεται μεταξύ `/* ... */` είναι σχόλιο και αγνοείται στη μεταγλώττιση. Ό,τι περικλείεται μεταξύ `{ ... }` είναι ένα "scope" στο πρόγραμμα, στη συγκεκριμένη περίπτωση τα περιεχόμενα της συνάρτησης `main()`.

Είμαστε έτοιμοι τώρα να γράψουμε τον κώδικα για την μελέτη των γεννητριών ψευδοτυχαίων αριθμών που αναφέραμε στο προηγούμενο κεφάλαιο. Ο κώδικας για την `naiveran()` βρίσκεται στο αρχείο `random.c`

```

/***** random.c *****/
static int ibm= 13337;
double naiveran(){

    int mult, modulo;
    double rmodulo;

    mult    = 1277;
    modulo  = 131072 ; // equal to 2^17
    rmodulo = (double)modulo;
    ibm    *= mult;
    ibm    = ibm % modulo;
    return ibm/rmodulo;
}

```

Παρατηρούμε τις δηλώσεις `int` (ακέραιοι), `double` (πραγματικοί διπλής ακρίβειας - 8 byte) των μεταβλητών `mult`, `modulo`, `rmodulo` οι οποίες γίνονται μέσα στο scope της συνάρτησης `naiveran()`.

Οι μεταβλητές αυτές “ζουν” μόνο μέσα στο `score` της συνάρτησης και οι αρχικές τους τιμές δεν καθορίζονται από κάποιο `standard`<sup>7</sup>. Αυτό σημαίνει πως με το καινούργιο κάλεσμα της συνάρτησης οι τιμές τους θα έχουν ξεχαστεί. Αντίθετα η μεταβλητή `ibm` ορίζεται έξω από το `score` της συνάρτησης και η τιμή της παραμένει σε σταθερό σημείο της μνήμης. Η γραμμή:

```
modulo = 131072 ; // equal to 2^17
```

ορίζει ό,τι είναι δεξιά από το “//” να είναι σχόλιο. Αυτό το συντακτικό είναι μέρος όχι της C αλλά της C++, αλλά είναι αποδεκτό από τους περισσότερους μεταγλωττιστές. Η γραμμή

```
rmodulo = (double)modulo;
```

είναι παράδειγμα της μετατροπής μιας μεταβλητής ενός τύπου σε έναν άλλο μέσω ενός “cast”. Πιο συγκεκριμένα η μεταβλητή τύπου `int` μετατρέπεται σε τύπου `double` με το `cast (double)`. Η γραμμή

```
ibm *= mult;
```

είναι ισοδύναμη με την

```
ibm = ibm * mult;
```

Τέλος η γραμμή

```
return ibm/rmodulo;
```

επιστρέφει την τιμή `ibm/modulo` στη θέση του ονόματος της συνάρτησης στο καλών πρόγραμμα. Φυσικά το `control` του προγράμματος επιστρέφει στο σημείο που έγινε η κλήση της συνάρτησης. Ο τύπος του αποτελέσματος εξαρτάται από τη δήλωση της συνάρτησης `naiveran()` η οποία στη συγκεκριμένη περίπτωση είναι `double`.

Ο κώδικας για την `drandom()` βρίσκεται στο αρχείο `drandom.c`

```
/****** drandom.c *****/
#define a 16807
#define m 2147483647
#define q 127773
#define r 2836
#define conv (1.0/(m-1))
```

<sup>7</sup>Ορισμένοι μεταγλωττιστές τις αρχικοποιούν στη μηδενική τιμή, αλλά αυτό μπορεί να οδηγήσει σε μη μεταφερόμενο (non-portable) κώδικα και απρόσμενα bugs!

```
static int seed = 323412;
double drandom(){
    long l;

    l = seed/q;
    seed = a*(seed-q*l) - r*l;
    if(seed < 0) seed +=m;
    return conv*(seed-1);
}
```

Εδώ βλέπουμε την άλλη συχνά χρησιμοποιούμενη εντολή του προεπεξεργαστή `#define` η οποία ορίζει ένα `macro`. Στην πιο απλή μορφή του ορίζει την τιμή ενός *συμβόλου* την οποία αντικαθιστά σε οποιοδήποτε σημείο του κώδικα συναντήσει το σύμβολο. Έτσι η γραμμή

```
#define a 16807
```

θα αντικαταστήσει την τιμή 16807 οπουδήποτε συναντήσει το σύμβολο `a`. Η γραμμή

```
#define conv (1.0/(m-1))
```

θα αντικαταστήσει την παραπάνω παράσταση οπουδήποτε συναντήσει το σύμβολο `conv`. Προσοχή, ο προεπεξεργαστής δεν γνωρίζει τι αντικαθιστά: Οι σειρές που γίνονται οι πράξεις εξαρτάται από τη μορφή της τελικής εντολής που θα σχηματιστεί, οπότε είναι καλό να *περικλείουμε πάντα ένα macro με παρενθέσεις*, αλλιώς μας περιμένουν εκπλήξεις! Η γραμμή

```
if(seed < 0) seed +=m;
```

είναι περίπτωση `if statement`. Αν η εντολή που θα εκτελεστεί μετά είναι μόνο μια, δεν απαιτείται ο εγκλεισμός σε `{...}` αγκύλες.

Η ρουτίνα-οδηγός `main.c` δίνεται από

```
/****** main.c *****/
// Program to show the correlations in the naive random number generator
#include <stdio.h>
#include <stdlib.h>

double naiveran(),drandom();
int main(int argc, char **argv){
    int i,N;
    double L = 10000.0;
    if(argc == 1)
```

```

    N=1000;
else
    N=atoi(argv[1]);

for(i=1;i<=N;i++){
    printf("%d %d\n", (int)(L*naiveran()), (int)(L*naiveran()));
    // printf("%d %d\n", (int)(L*drandom()), (int)(L*drandom()));
    // printf("%d %d\n", (int)(L*drand48()), (int)(L*drand48()));
    if(i%1000==0){
        fflush(stdout);
        sleep(1);
    }
}
}
}

```

Βλέπουμε ότι αρχικά απαιτείται η δήλωση των προς χρήση συναρτήσεων `naiveran()` και `drandom()` ώστε να γνωρίζει ο μεταγλωττιστής τον τύπο τους. Παρατηρούμε τώρα ότι η `main()` δηλώνεται να έχει ορίσματα. Αυτά αναφέρονται στα ορίσματα που έχει η εντολή που εκτελεί το πρόγραμμα στην γραμμή εντολών. Η μεταβλητή `argc` τύπου `int` μετράει τον αριθμό των ορισμάτων συμπεριλαμβανομένου και του ονόματος του προγράμματος (δηλ. είναι ίση με 1 όταν δεν υπάρχει κανένα όρισμα). Η μεταβλητή `argv` είναι ένα διάνυσμα (array) από ακολουθίες χαρακτήρων (strings) που αποθηκεύει το περιεχόμενο των ορισμάτων ως "λέξεις". Η πρώτη λέξη είναι ο όνομα του προγράμματος, η δεύτερη το πρώτο όρισμα κ.ο.κ. Για το λόγο αυτό είναι ορισμένη σαν ένα pointer σε pointer από χαρακτήρες `char`. Θυμίζουμε στον αναγνώστη ότι η ακολουθία χαρακτήρων (string) στη C ορίζεται ως ένα array χαρακτήρων το τέλος της οποίας είναι ο χαρακτήρας `null = '\0'`<sup>8</sup>

Εκτελώντας τώρα τις εντολές:

```

% cc main.c random.c drandom.c -o naiveran
% naiveran 10000

```

μεταγλωττίζει το πρόγραμμα και το εκτελεί με όρισμα την ακολουθία χαρακτήρων "10000"={ '1', '0', '0', '0', '0', '\0' } (και όχι τον ακέραιο 10000!). Στην περίπτωση αυτή `argc = 2`, `argv[0] = "naiveran"`, `argv[1] = "10000"`. Για να μετατρέψουμε την ακολουθία χαρακτήρων "10000" στον ακέραιο 10000 χρησιμοποιούμε την συνάρτηση `atoi()`. Η γραμμή

```

    N=atoi(argv[1]);

```

<sup>8</sup>Ο `null` χαρακτήρας είναι διαφορετικός από τον '0' που έχει αριθμητική τιμή 48.

κάνει ακριβώς τη μετατροπή αυτή και αποθηκεύει τον ακέραιο 10000 στη μεταβλητή *N*. Στη συνέχεια βλέπουμε ένα απλό παράδειγμα κατασκευάσματος `if ... else` το οποίο σε απλά ελληνικά λέει ότι αν το πρόγραμμα εκτελεστεί χωρίς όρισμα (`argc = 1`) η μεταβλητή *N* θα έχει την τιμή 1000 ενώ σε κάθε άλλη περίπτωση την τιμή που δίνει ο χρήστης στο πρώτο όρισμα στη γραμμή εντολών<sup>9</sup>

Το `block for(i=1; i<=N; i++) ...` επαναλαμβάνεται *N* φορές: Η εντολή `i=1` εκτελείται μία φορά στην αρχή. Η συνθήκη `i<=N` ελέγχεται πριν εκτελεστεί το `block`. Αν η τιμή είναι `TRUE`<sup>10</sup> το `block` εκτελείται αλλιώς το πρόγραμμα συνεχίζεται. Η τελευταία εντολή `i++` εκτελείται στο τέλος του `block` και στη συγκεκριμένη περίπτωση αυξάνει την τιμή του *i*. Μέσα στο `block` εκτελείται κατ' αρχήν η εντολή

```
printf("%d %d\n", (int)(L*naiveran()), (int)(L*naiveran()));
```

η οποία τυπώνει στην καθιερωμένη έξοδο `stdout` σε κάθε γραμμή δύο τυχαίους ακεραίους από 0 ... (L-1). Αυτό γίνεται καλώντας τη `double` συνάρτηση  $0 \leq \text{naiveran}() < 1$ , πολλαπλασιάζοντας με `L=10000.0` και μετατρέποντας το αποτέλεσμα σε `int` με το `cast (int)` παίρνοντας απλά το ακέραιο μέρος.

Οι επόμενες γραμμές είναι σχόλια. Η διαφορά με την προηγούμενη γραμμή είναι η κλήση άλλης συνάρτησης για την παραγωγή των τυχαίων αριθμών. Η συνάρτηση `drandom()` είναι η γεννήτρια τυχαίων αριθμών που δεν έχει το πρόβλημα της `naiveran()`. Είναι η συνάρτηση που επιλέξαμε στο μάθημα αυτό για να χρησιμοποιήσουμε, λόγω της απλότητάς της και ταχύτητάς της. Η συνάρτηση `drand48()` είναι συνάρτηση που παρέχεται από το σύστημα και είναι ένας απλός modulo generator (δείτε τα `man pages`). Ο λόγος που γράψαμε τη `main()` με το τρόπο αυτό είναι γιατί αποσχολιάζοντας την κατάλληλη γραμμή θα επαναμεταγλωττίζουμε τον κώδικα ξανά ώστε να χρησιμοποιήσουμε διαφορετική γεννήτρια τυχαίων αριθμών. Οπότε μεταβάλλουμε τις παρακάτω γραμμές ως:

```
// printf("%d %d\n", (int)(L*naiveran()), (int)(L*naiveran()));
printf("%d %d\n", (int)(L*drandom()), (int)(L*drandom()));
// printf("%d %d\n", (int)(L*drand48()), (int)(L*drand48()));
```

και δίνουμε την εντολή

```
% cc main.c random.c drandom.c -o drandom
```

<sup>9</sup>Ο συνετός προγραμματιστής θα έκανε έλεγχο αν η τιμή της *N* είναι αποδεκτή.

<sup>10</sup>Στην C μια λογική έκφραση δεν είναι ξεχωριστός τύπος αλλά είναι τύπου `int` και είναι `FALSE` όταν είναι 0 και `TRUE` αν είναι μη-μηδενική.

για να πάρουμε το πρόγραμμα `drandom` και τις μεταβάλλουμε ως

```
// printf("%d %d\n", (int)(L*naiveran()), (int)(L*naiveran()));
// printf("%d %d\n", (int)(L*drandom()), (int)(L*drandom()));
printf("%d %d\n", (int)(L*drand48()), (int)(L*drand48()));
```

και δίνουμε την εντολή

```
% cc main.c random.c drandom.c -o drand48
```

για να πάρουμε το πρόγραμμα `drand48`.

Ο γραμμές

```
if(i%1000==0){
    fflush(stdout);
    sleep(1);
}
```

καλεί τις συναρτήσεις `fflush()`, `sleep()` κάθε φορά που το `i` είναι πολλαπλάσιο του 1000. Η πράξη `i%1000` μας δίνει το υπόλοιπο της διαιρέσεως του `i` με το 1000 (πράξη modulo). Η συνάρτηση `fflush()` αδειάζει τα buffers εξόδου στο αρχείο που δείχνει το όρισμά της (εδώ η καθιερωμένη έξοδος `stdout`). Ο λόγος είναι ότι οι συναρτήσεις `printf()` και συναφείς έχουν "buffered output", δηλ. δεν γράφουν κάθε bit ή byte αλλά σε μια περιοχή της μνήμης στην οποία έχουν γρήγορη πρόσβαση και μόλις τα buffers γεμίσουν<sup>11</sup> τότε μόνο γίνεται η μεταφορά των δεδομένων στο αρχείο (στο οποίο συνήθως έχουμε αργή πρόσβαση). Τέλος η συνάρτηση `sleep()` κάνει μία παύση για 1 δευτερόλεπτο (ή όσο ζητήσουμε). Με τον τρόπο αυτό πετυχαίνουμε να τυπώνονται τα ζευγάρια τυχαίων αριθμών ανά 1000 και να παρακολουθούμε το αποτέλεσμα γραφικά.

### 3.2.2 ΑΠΟΤΕΛΕΣΜΑΤΑ

Αφού μεταγλωττίσουμε το πρόγραμμα σύμφωνα με τα παραπάνω μπορούμε να μελετήσουμε το πρόβλημα που παρουσιάζει ο αλγόριθμος στην `naiveran()`. Ας αρχίσουμε τρέχοντας το πρόγραμμα με την εντολή

```
% naiveran 100000 > naiveran.out
% drandom 100000 > drandom.out
```

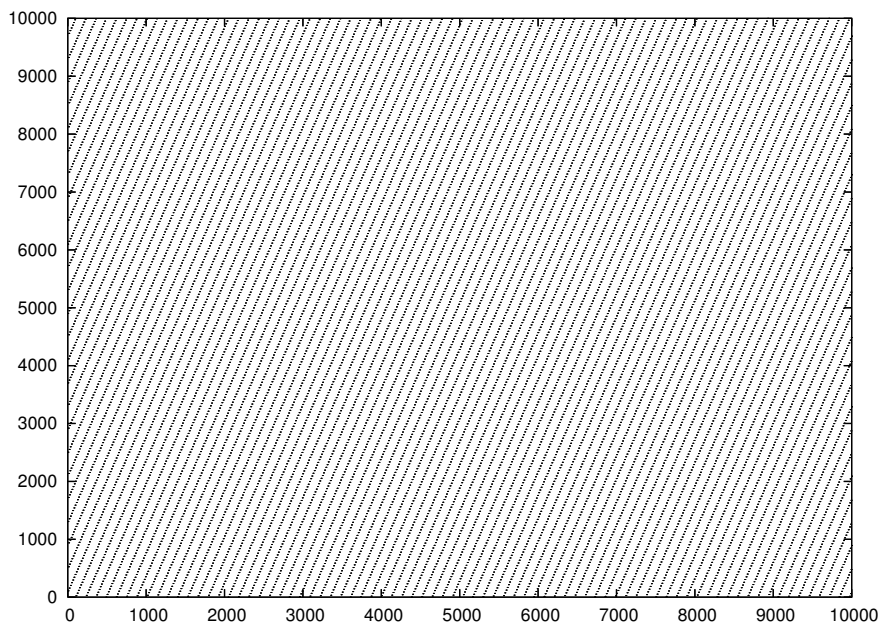
<sup>11</sup>Το μέγεθος του buffer εξαρτάται από το σύστημα και τη συσκευή στην οποία γράφουμε.



Με τις παραπάνω εντολές παράγουμε 100000 ζεύγη τυχαίων ακεραίων ομοιόμορφα κατανομημένων στο διάστημα  $[0, 10000)$  χρησιμοποιώντας την αντίστοιχη συνάρτηση. Για να δούμε γραφικά τα αποτελέσματά μας χρησιμοποιούμε το πρόγραμμα `gnuplot`<sup>12</sup>:

```
% gnuplot
gnuplot> plot "naiveran.out" using 1:2 with dots
gnuplot> plot "drandom.out" using 1:2 with dots
```

Οι παραπάνω εντολές παράγουν τα Σχήματα 3.2.2 και 3.2.2 στα οποία βλέπουμε καθαρά το συσχετισμό των ζευγών που παράγονται από την `naiveran()`.



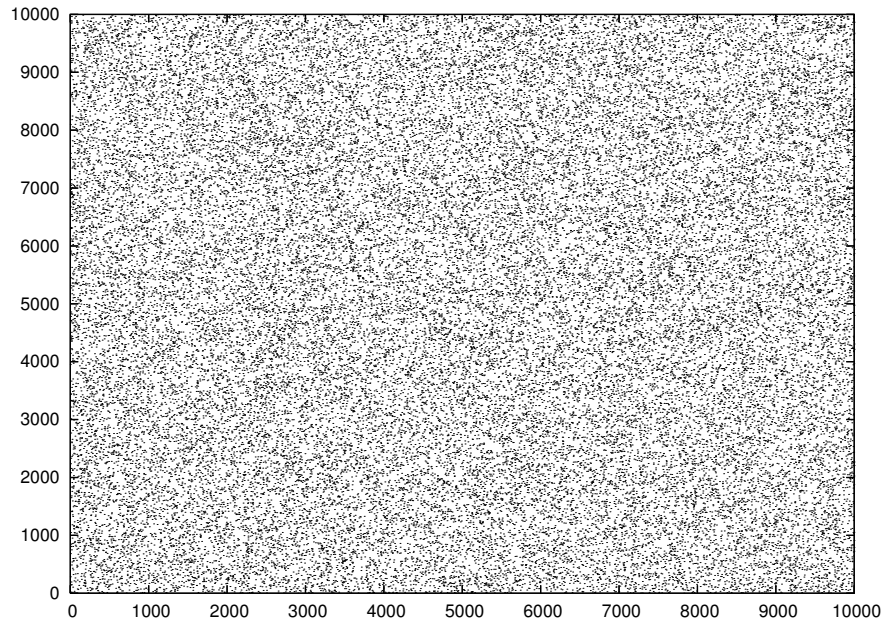
Σχήμα 3.1: Τα ζεύγη ψευδοτυχαίων αριθμών από τη συνάρτηση `naiveran()`. Φαίνεται καθαρά ο συσχετισμός των ζευγών που παράγονται αφού τα σημεία κάθονται πάνω σε εμφανές διακριτό πλέγμα.

Η ομοιόμορφη κατανομή των τυχαίων αριθμών μπορεί ναδειχθεί γραφικά φτιάχνοντας το ιστόγραμμα της σχετικής συχνότητας των τυχαίων αριθμών. Για να κάνουμε ιστογράμματα χρησιμοποιούμε το script `histogram` γραμμένο σε `awk`<sup>13</sup> ως εξής:

```
% histogram -h
```

<sup>12</sup>Το πρόγραμμα διαθέτει πολύ καλή on-line βοήθεια. Λ.χ. δώστε τις εντολές `help` ή `help plot`. Επισκεφτείτε την ιστοσελίδα `www.gnuplot.info` για εγχειρίδιο και `demos`.

<sup>13</sup>Βλ. συνοδευτικό λογισμικό στο αντίστοιχο κεφάλαιο



Σχήμα 3.2: Τα ζεύγη ψευδοτυχαίων αριθμών από τη συνάρτηση `drandom()`. Φαίνεται ότι τα σημεία αυτά κατανέμονται τυχαία σε σχέση με αυτά της `naiveran()`.

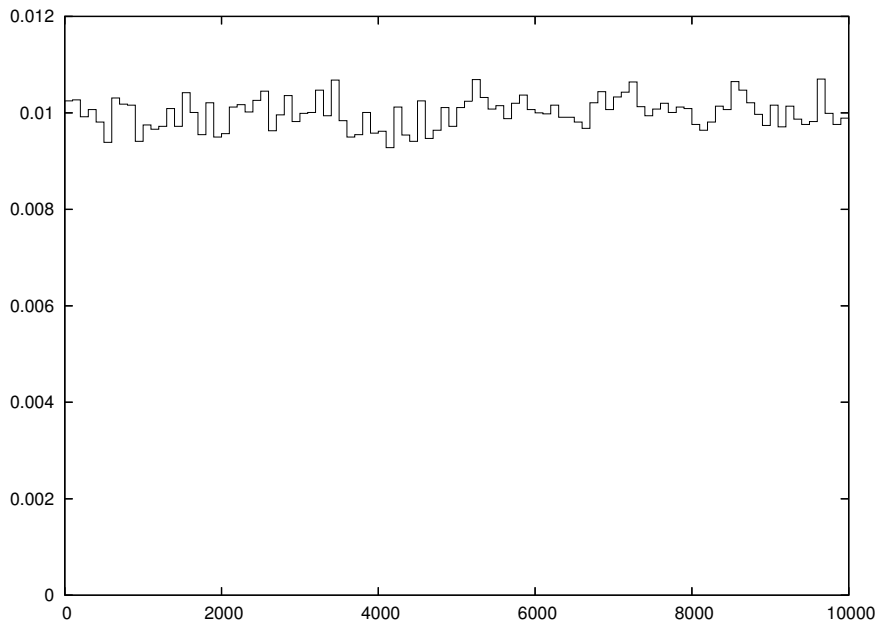
```
% histogram -v f=0.01 drandom.out > drandom.hst
% gnuplot
gnuplot> plot "drandom.hst" using 1:3 with histsteps
gnuplot> plot [0:10000][0:0.012] "drandom.hst" using 1:3 with histsteps
```

Τα αποτελέσματα φαίνονται στα Σχήματα 3.2.2 και 3.2.2. Στη συνέχεια μελετούμε τη διασπορά των μετρήσεων μας, οι οποίες είναι εμφανείς ήδη από το Σχήμα 3.2.2. Η διασπορά αυτή μειώνεται με το μέγεθος του δείγματος των τυχαίων αριθμών που συλλέγουμε. Αυτό φαίνεται στο ιστόγραμμα του Σχήματος 3.2.2. Για να δούμε ποσοτικά την εξάρτηση των διακυμάνσεων από το μέγεθος του δείγματος  $n$ , υπολογίζουμε την ποσότητα ( $x_i$  είναι η ακολουθία των τυχαίων αριθμών)

$$\sigma = \sqrt{\frac{1}{n-1} \left( \frac{1}{n} \sum_{i=1}^n x_i^2 - \left( \frac{1}{n} \sum_{i=1}^n x_i \right)^2 \right)} \quad (3.15)$$

την οποία αποδίδουμε γραφικά στο Σχήμα 3.2.2. Η σχέση

$$\ln \sigma \sim \frac{1}{2} \ln(n) \quad (3.16)$$



Σχήμα 3.3: Η σχετική συχνότητα κατανομής των ψευδοτυχαίων αριθμών που παράγονται από την συνάρτηση `drandom()`. Φαίνεται η ομοιόμορφη κατανομή και η διασπορά γύρω από τη μέση τιμή.

όπως φαίνεται από την ευθεία που προσαρμόζουμε στα σημεία δείχνει ότι

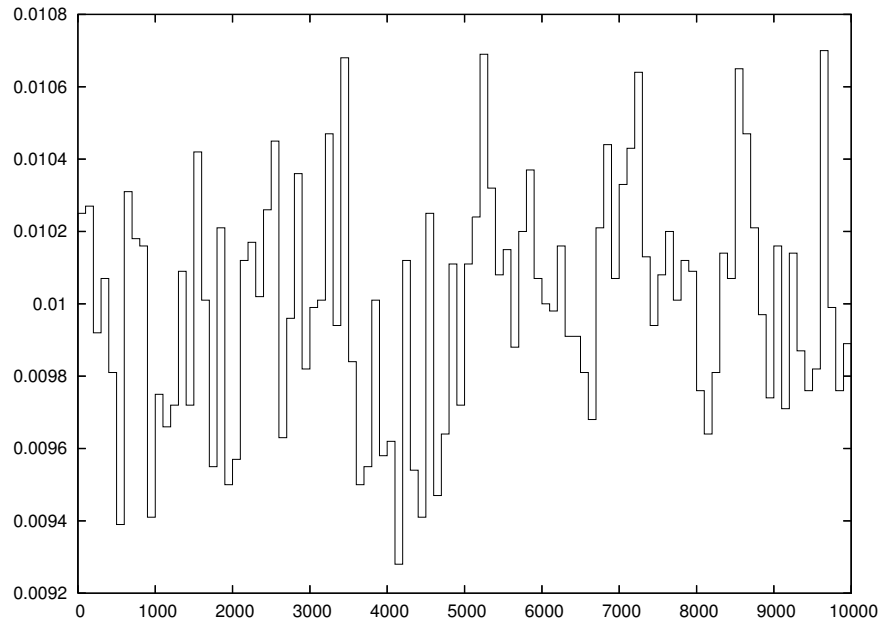
$$\sigma \sim \frac{1}{\sqrt{n}} \quad (3.17)$$

Θα κλείσουμε το κεφάλαιο αυτό αναφέροντας σύντομα την παραγωγή ψευδοτυχαίων αριθμών με κατανομή διαφορετική από την ομοιόμορφη. Για μια τυχαία κατανομή με συνάρτηση πυκνότητας πιθανότητας  $f(x)$  μπορούμε να πάρουμε ψευδοτυχαίους αριθμούς που ακολουθούν την  $f(x)$  από ομοιόμορφα κατανεμημένους αριθμούς στο διάστημα  $[0, 1)$ . Έστω η συνάρτηση κατανομής

$$0 \leq u = F(x) = \int_{-\infty}^x f(x') dx' \leq 1, \quad (3.18)$$

που δεν είναι άλλο από το εμβαδόν κάτω από την  $f(x)$  στο  $(-\infty, x]$  και ισούται με την πιθανότητα  $P(x' < x)$ . Αν επιλέξουμε το  $u$  ομοιόμορφα κατανεμημένο στο διάστημα  $[0, 1)$  τότε θα έχουμε  $P(u' < u) = u$ . Τότε το  $x = F^{-1}(u)$  είναι τέτοιο ώστε  $P(x' < x) = u = F(x)$ , άρα κατανέμεται σύμφωνα με την  $f(x)$ . Οπότε αν  $u_i$  είναι ομοιόμορφα κατανεμημένοι ψευδοτυχαίοι αριθμοί τότε οι

$$x_i = F^{-1}(u_i) \quad (3.19)$$



Σχήμα 3.4: Το προηγούμενο σχήμα σε μεγαλύτερη κλίμακα για να φανεί καθαρότερα η διασπορά.

είναι μια ακολουθία ψευδοτυχαίων αριθμών που κατανέμεται σύμφωνα με την  $f(x)$ . Για παράδειγμα θεωρούμε την κατανομή κατά Cauchy

$$f(x) = \frac{1}{\pi} \frac{c}{c^2 + x^2} \quad c > 0. \quad (3.20)$$

Τότε

$$F(x) = \int_{-\infty}^x f(x') dx' = \frac{1}{2} + \frac{1}{\pi} \tan^{-1} \left( \frac{x}{c} \right). \quad (3.21)$$

Σύμφωνα με τα παραπάνω η γεννήτρια τυχαίων αριθμών θα δίνεται από τη σχέση

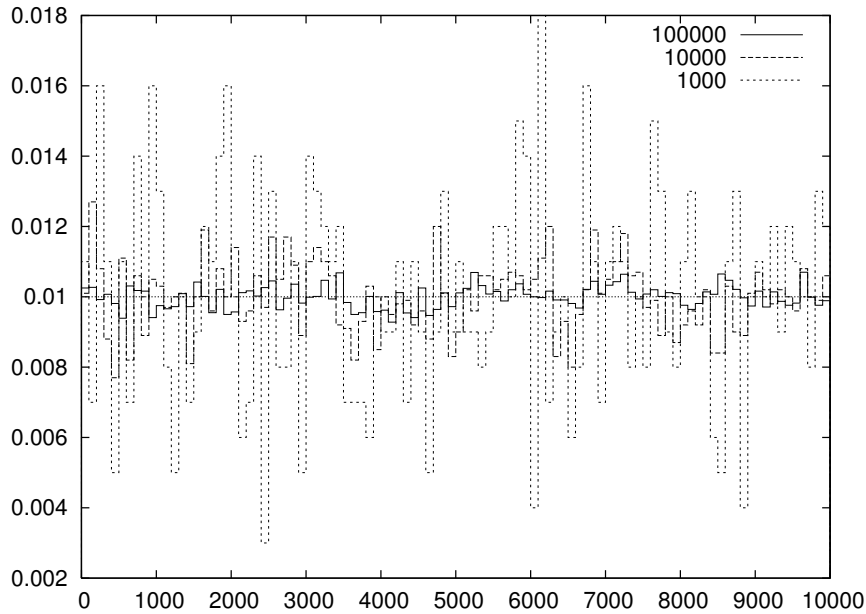
$$x_i = c \tan(\pi u_i - \pi/2) \quad (3.22)$$

ή ισοδύναμα (για πιο γρήγορο πρόγραμμα)

$$x_i = c \tan(2\pi u_i). \quad (3.23)$$

Περισσότερες εφαρμογές έχει η γεννήτρια Gaussian τυχαίων αριθμών. Η κατανομή αυτή δίνεται από τη συνάρτηση πυκνότητας πιθανότητας

$$g(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-x^2/(2\sigma^2)} \quad (3.24)$$



Σχήμα 3.5: Η σχετική συχνότητα κατανομής των ψευδοτυχαίων αριθμών που παράγονται από την συνάρτηση `drandom()` σαν συνάρτηση του μεγέθους του δείγματος των τυχαίων αριθμών  $n$  για  $n = 1000, 10000, 100000$ .

Η συνάρτηση κατανομής είναι η

$$G(x) = \int_{-\infty}^x g(x') dx' = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left( \frac{x}{\sqrt{2}\sigma} \right) \quad (3.25)$$

όπου η  $\operatorname{erf}(x) = \int_{-\infty}^x \exp\{-(x')^2\} dx'$  είναι η error function η οποία μπορεί να υπολογιστεί αριθμητικά καθώς και η αντίστροφή της. Αυτό θα οδηγούσε σε πολύ αργό αλγόριθμο για τη γεννήτριά μας οπότε χρησιμοποιούμε την πυκνότητα πιθανότητας  $\rho(x, y)$  δύο τυχαίων μεταβλητών  $x, y$

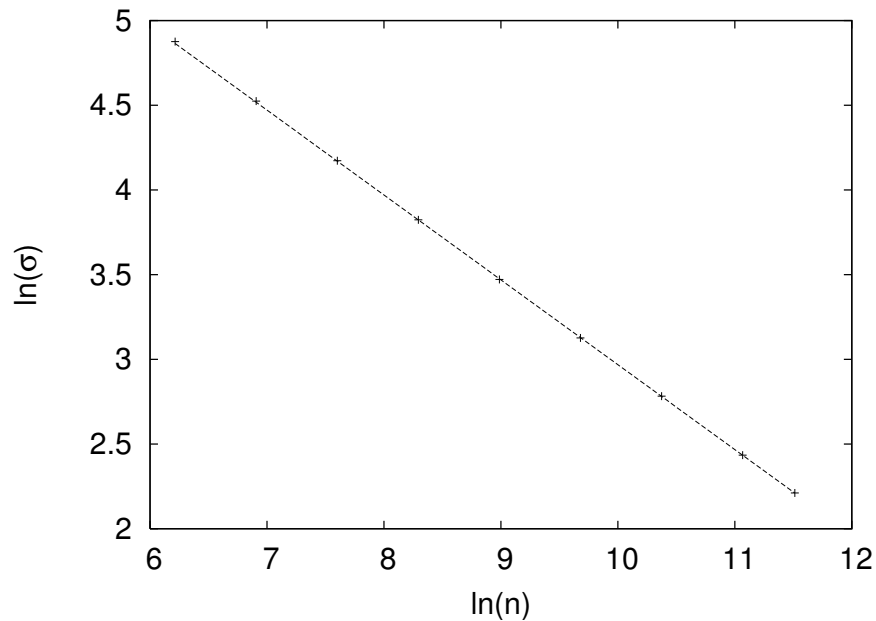
$$\rho(x, y) dx dy = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/(2\sigma^2)} \frac{1}{\sqrt{2\pi}\sigma} e^{-y^2/(2\sigma^2)} dx dy = \frac{1}{2\pi\sigma} e^{-r^2/(2\sigma^2)} r dr d\phi \quad (3.26)$$

όπου  $x = r \cos \phi$ ,  $y = r \sin \phi$ . Τότε έχουμε ότι

$$u = G(r) = \int_0^r \int_0^{2\pi} dr d\phi \rho(r, \phi) = 1 - e^{-r^2/(2\sigma^2)} \quad (3.27)$$

η οποία όταν αντιστραφεί θα μας δώσει

$$r = \sigma \sqrt{-2 \ln(1 - u)}. \quad (3.28)$$



Σχήμα 3.6: Η εξάρτηση της διασποράς (3.15) από το  $n$  για τη συνάρτηση `drandom()`.

Οπότε αρκεί να παράγουμε ακολουθία  $\{u_i\}$  ομοιόμορφα κατανεμημένων τυχαίων αριθμών και να πάρουμε:

$$r_i = \sigma \sqrt{-2 \ln(u_i)} \quad (3.29)$$

$$\phi_i = 2\pi u_{i+1} \quad (3.30)$$

$$x_i = r_i \cos \phi_i \quad (3.31)$$

$$x_{i+1} = r_i \sin \phi_i, \quad (3.32)$$

οπότε προκύπτει η ακολουθία  $\{x_i\}$  που είναι Gaussian τυχαίοι αριθμοί<sup>14</sup>. Ο προγραμματισμός της σχετικής συνάρτησης για  $\sigma = 1$  δίνεται παρακάτω:

```

/***** gausran() *****/
#define PI2 6.2831853071795864769
double gausran(){
    double r,phi;
    static int new = 1;
    static double x;
    if(new){
        new = 0;

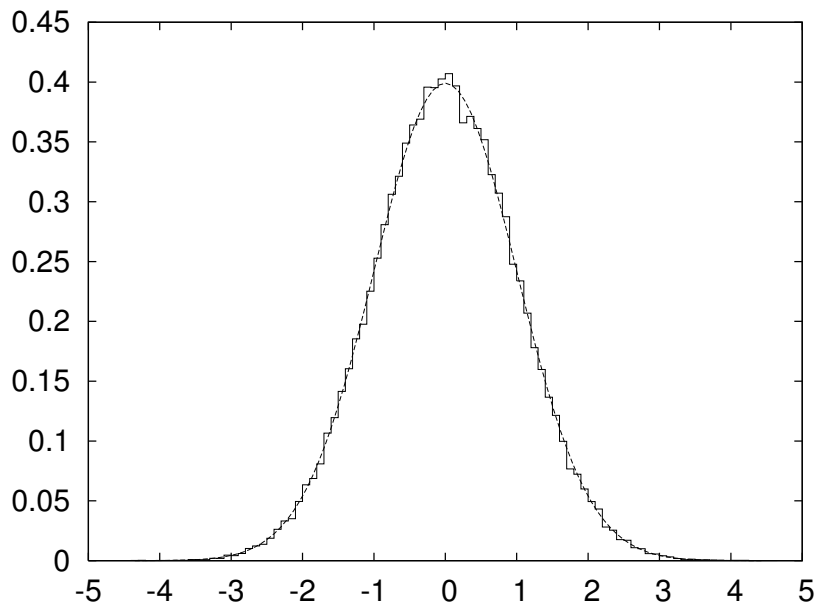
```

<sup>14</sup>Μπορεί να δειχθεί ότι τα  $x_i, x_{i+1}$  είναι στατιστικά ανεξάρτητα.

```

    phi = PI2*drandom();
    if( (r=drandom()) == 0.0) r=drandom();
    r = sqrt(-2.0 * log(r));
    x = r * cos(phi);
    return r * sin(phi);
}
else{
    new = 1;
    return x;
}
}

```



Σχήμα 3.7: Η σχετική συχνότητα κατανομής των ψευδοτυχαίων αριθμών που παράγονται από την συνάρτηση `gaussran()`. Το ιστόγραμμα υπερτίθεται στην γραφική παράσταση της (3.24).

Το αποτέλεσμα φαίνεται στο Σχήμα 3.2.2. Η μόνη ιδιομορφία στον προγραμματισμό είναι ότι σε κάθε εφαρμογή της (3.29) παράγουμε 2 τυχαίους αριθμούς ενώ η συνάρτηση επιστρέφει τον ένα. Για το λόγο αυτό η συνάρτηση διατηρεί μνήμη για το αν πρέπει να παράγει καινούργιο ζευγάρι (η “σημαία” (flag) `new`) και αν όχι την τιμή ενός από τους τυχαίους αριθμούς (μεταβλητή `x`). Οι τιμές των μεταβλητών αυτών παραμένουν στη μνήμη μετά από την επιστροφή από τη συνάρτηση δηλώνοντάς τις ως `static`:

```
static int    new = 1;
```

```
static double x;
```

Επίσης το πρόγραμμα ελέγχει το  $r$  να είναι μη μηδενικό. Αν είναι καλεί ξανά την `drandom()` η οποία δεν παράγει δύο μηδενικά στη σειρά:

```
if( (r=drandom()) == 0.0) r=drandom();
```

Η ανάλυση των αποτελεσμάτων αφήνεται για άσκηση. Δίνεται και συνοδευτικό λογισμικό.

### 3.3 ΜΕΛΕΤΗ ΤΟΥ ΤΥΧΑΙΟΥ ΠΕΡΙΠΑΤΗΤΗ

Αρχικά παραθέτουμε το πρόγραμμα που κωδικοποιεί τον αλγόριθμο που περιγράψαμε στο τέλος του υποκεφαλαίου 3.1. Το πρόγραμμα αποθηκεύεται στο αρχείο `rw.c`:

```

/***** rw.c *****/
#include <stdio.h>

#define Nwalk 1000
#define Nstep 100000

double drandom();
int main (){

    int    iwalk,istep,ir;
    double x,y;
    FILE   *fp,*fpR;

    fpR=fopen("dataR","w");
    for(iwalk=1;iwalk<=Nwalk;iwalk++){
        x=0.0;y=0.0;
        fp=fopen("data","w");
        for(istep=1;istep<=Nstep;istep++){
            ir=(int)(drandom()*4);
            switch(ir){
            case 0:
                x += 1.0;
                break;
            case 1:
                x -= 1.0;
                break;
            case 2:

```



```

        y += 1.0;
        break;
    case 3:
        y -= 1.0;
        break;
    }
    fprintf(fp,"%f %f\n",x,y);
} //istep
fclose (fp);
sleep (2 );
fprintf(fpR,"%f\n",x*x+y*y);
fflush (fpR);
} //iwalk
} //main

```

Στο πρόγραμμα αυτό αρχικά κωδικοποιούμε “σκληρά” το μήκος των διαδρομών  $N_{step}$  και τον αριθμό των διαδρομών  $N_{walk}$ . Οπότε για την αλλαγή των παραμέτρων αυτών απαιτείται επαναμεταγλωττισμός του προγράμματος. Τα αποτελέσματά μας τα αποθηκεύουμε στα αρχεία `dataR` και `data` στα οποία αποθηκεύουμε το τετράγωνο του μήκους της διαδρομής  $R^2$  και τις συντεταγμένες των σημείων  $(x, y)$  που επισκέπτεται ο περιπατητής αντίστοιχα σε κάθε διαδρομή. Τα αρχεία ανοίγουν για γράψιμο με τις εντολές

```

fpR=fopen("dataR", "w");
fp=fopen("data", "w");

```

που επιστρέφουν από ένα pointer σε FILE structure. Η οδηγία "w" στη συνάρτηση `fopen()` κάνει το pointer διαθέσιμο για γράψιμο από την αρχή του αρχείου, οπότε τα περιεχόμενα που μπορεί να υπήρχαν στα αρχεία αυτά χάνονται<sup>15</sup>. Στα αρχεία γράφουμε με τη συνάρτηση `fprintf()` η χρήση της οποίας είναι ανάλογη με αυτή της `printf()` ενώ τα κλείνουμε με την συνάρτηση `fclose()`. Για να κάνουμε τα περιεχόμενα των αρχείων άμεσα διαθέσιμα, αδειάζουμε τα buffers με τη συνάρτηση `fflush()`. Παρατηρούμε ότι το αρχείο `data` ανοίγει και μηδενίζεται στην αρχή κάθε διαδρομής, οπότε σε κάθε στιγμή περιέχει της συντεταγμένες μιας μόνο διαδρομής.

Στην αρχή κάθε διαδρομής μηδενίζουμε τη θέση του περιπατητή ενώ στο τέλος καταγράφουμε το  $(\text{μήκος})^2 = x*x+y*y$  της τελικής του θέσης στο αρχείο `dataR`:

<sup>15</sup>Η οδηγία "a" προσθέτει περιεχόμενα στο τέλος του αρχείου (append) ενώ η οδηγία "r" ανοίγει το αρχείο για διάβασμα. Για περισσότερες πληροφορίες δώστε την εντολή `% man fopen`.

```

for(iwalk=1;iwalk<=Nwalk;iwalk++){
    x=0.0;y=0.0;
    .... make one random walk ....
    fprintf(fpR,"%f\n",x*x+y*y);
    fflush (fpR);
} //iwalk

```

Η εντολή `fflush (fpR);` γράφει άμεσα τον αριθμό  $x*x+y*y$  στο αρχείο `dataR`.

Κάθε διαδρομή εκτελείται σε `Nstep` βήματα στο βρόγχο

```

for(istep=1;istep<=Nstep;istep++){
    ....
} //istep

```

όπου αφού επιλεγεί το τυχαίο διάνυσμα  $\vec{\xi}_{istep}$  αυτό προστίθεται στην εκάστοτε θέση  $\vec{r}_{istep} = (x, y)$ . Το  $\vec{\xi}_{istep}$  επιλέγεται στη γραμμή

```
ir=(int)(drandom()*4);
```

όπου ο `ir = 0, 1, 2, 3` λόγω του τελεστή (`cast`) (`int`) που απλά κόβει το δεκαδικό μέρος ενός `double`. Οι τιμές του `ir` αντιστοιχούν στις τέσσερις δυνατές τιμές του  $\vec{\xi}$ . Η ενημέρωση της θέσης του περιπατητή γίνεται με το `switch()` `construct` που ανάλογα με την τιμή του `ir` ενημερώνει τις συντεταγμένες που μεταβάλλονται στην αντίστοιχη διεύθυνση. Θυμίζουμε στον αναγνώστη τη χρήση των τελεστών `+=` (`-=`) που αυξάνουν (μειώνουν) το αριστερό μέλος κατά την ποσότητα που είναι στο δεξί μέλος.

Επίσης τονίζουμε στον αναγνώστη ότι στην απλή αυτή μορφή του προγράμματος παράγουμε πάντα την ίδια ακολουθία ψευδοτυχαίων αριθμών, άρα και ακολουθία τυχαίων διαδρομών, μια και η `drandom()` “σπέρνεται” πάντα με τον ίδια αρχική κατάσταση. Αφήνεται για άσκηση η μεταβολή του κώδικα ώστε να αρθεί ο περιορισμός αυτός.

Για τη μεταγλώττιση του κώδικα και εκτέλεση του προγράμματος δίνουμε τις εντολές

```

% cc rw.c drandom.c -o rw
% ./rw

```

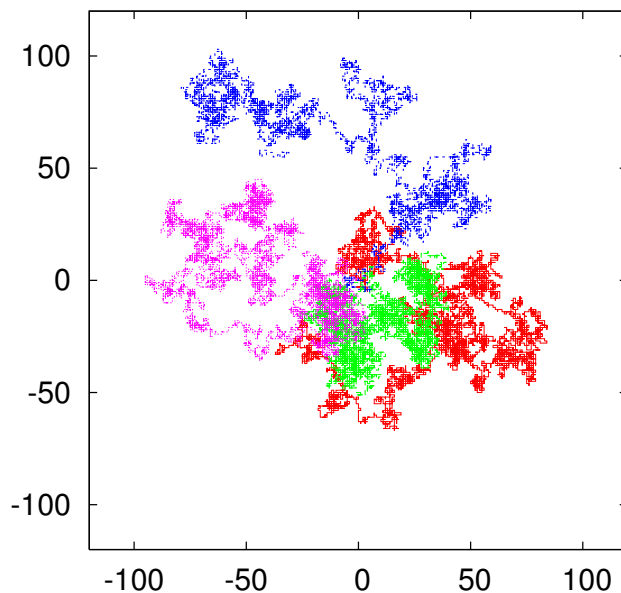
που μας δίνει το εκτελέσιμο αρχείο `rw` που τρέχει το πρόγραμμα. Λόγω της εντολής `sleep(2)`, το πρόγραμμα σταματάει για 2 δευτ. κάθε φορά που ολοκληρώνεται μια διαδρομή (αφαιρέστε την όταν θα θέλετε να παράγετε πολλές διαδρομές). Αυτό μας επιτρέπει να παρακολουθήσουμε γραφικά τις διαδρομές που παράγονται. Κατά τη διάρκεια που το πρόγραμμα τρέχει δώστε τις εντολές

```
% gnuplot
gnuplot> plot "data" with lines
```

και επαναλάβετε την τελευταία εντολή όσες φορές θέλετε. Για να γίνεται αυτή η διαδικασία αυτόματα χρησιμοποιήστε το script `eternal-rw` από το συνοδευτικό λογισμικό:

```
% ./rw &
% ./eternal-rw &
% killall rw eternal-rw gnuplot
```

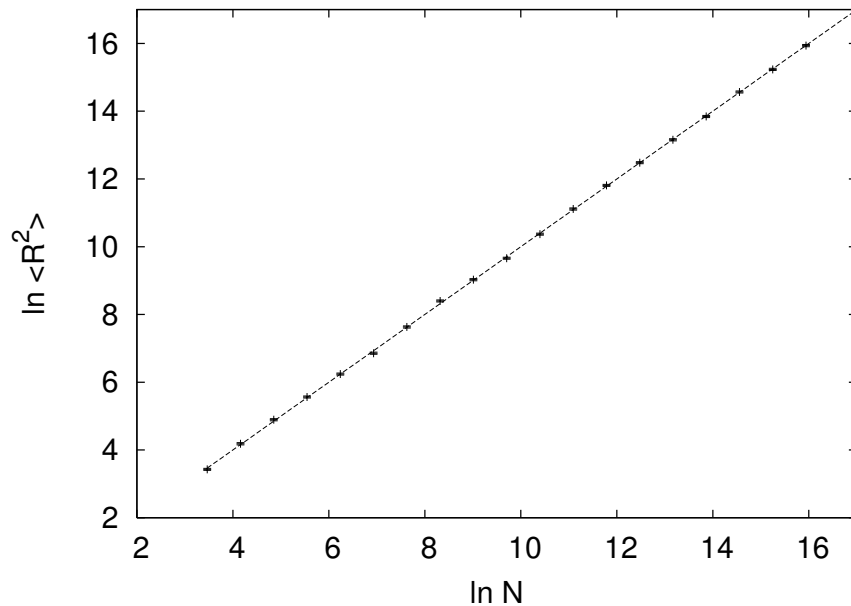
όπου την τελευταία εντολή τη δίνετε φυσικά για να τερματίσετε την εκτέλεση των προγραμμάτων.



Σχήμα 3.8: 4 τυπικές διαδρομές του τυχαίου περιπατητή για  $N = 10000$ .

Στο Σχήμα 3.3 παρατίθενται τυπικές διαδρομές από το τρέξιμο του παραπάνω προγράμματος. Στο Σχήμα 3.3 φαίνονται τα αποτελέσματα για την ποσότητα  $\langle R^2 \rangle$  για  $N = 10, \dots, 100000$  τα οποία επιβεβαιώνουν τη σχέση (3.10)  $\langle R^2 \rangle = N$ . Το σχήμα αυτό μπορεί να αναπαραχθεί ως εξής:

1. Θέτουμε τις `Nwalk` και `Nstep` στις επιθυμητές τιμές στο αρχείο `rw.c`. Αφαιρούμε την εντολή `sleep(2)`;
2. Μεταγλωττίζουμε τον κώδικα με την εντολή



Σχήμα 3.9: Αριθμητική επιβεβαίωση της σχέσης  $\langle R^2 \rangle = N$  για  $N = 10, \dots, 100000$ . Η ευθεία γραμμή είναι προσαρμογή της συνάρτησης  $y = ax$  με  $a = 0.9994(13)$  και  $\chi^2/(\text{dof}) = 0.25$

```
% cc rw.c drandom.c -o rw
```

3. Τρέχουμε το πρόγραμμα και αναλύουμε τα δεδομένα από το αρχείο dataR:

```
% ./rw
% awk '{av+=$1}END{print av/NR}' dataR
```

και σημειώνουμε τα αποτελέσματα σε ένα αρχείο r2.dat σε δύο στήλες με το μήκος των διαδρομών  $N$  το υπολογιζόμενο  $\langle R^2 \rangle$ . Το πρόγραμμα awk με την εντολή `{av+=$1}` προσθέτει την πρώτη στήλη κάθε γραμμής του αρχείου dataR στη μεταβλητή av ενώ αφού διαβάσει το αρχείο με την εντολή `END{print av/NR}` τυπώνει τη μεταβλητή av διαιρεμένη με τον αριθμό των γραμμών στο αρχείο NR = “Number of Records”, δηλ. το μέσο όρο της πρώτης στήλης του dataR.

4. Με τη μέθοδο των ελαχίστων τετραγώνων βρίσκουμε τη βέλτιστη ευθεία  $y = ax+b$  που περνάει από τα σημεία  $(\ln N, \ln \langle R^2 \rangle)$ .

## 5. Φτιάχνουμε το διάγραμμα με τις εντολές

```
% gnuplot
gnuplot> a = <your value>
gnuplot> b = <your value>
gnuplot> plot "r2.dat" using (log($1)):(log($2)), a*x+b
```

όπου <your value> είναι οι τιμές που υπολογίσατε με τη μέθοδο των ελαχίστων τετραγώνων.

Φυσικά τα παραπάνω αποτελέσματα δεν έχουν νόημα αν δεν γνωρίζουμε τα στατιστικά σφάλματα που υπεισέρχονται. Μια και πρόκειται για στατιστικές ποσότητες, η αναμενόμενη τιμή προσεγγίζεται στο όριο άπειρων μετρήσεων με μια ταχύτητα  $\sim 1/\sqrt{M}$  όπου  $M$  ο αριθμός των μετρήσεων. Μια και κάθε φορά παράγουμε στοχαστικά ανεξάρτητες μεταξύ τους διαδρομές (μια διαδρομή είναι ανεξάρτητη από τις προηγούμενες)<sup>16</sup>, το σφάλμα στις μετρήσεις μας θα δίνεται από τη σχέση (3.15), λ.χ.

$$\delta \langle R^2 \rangle = \sqrt{\frac{1}{M-1} \left( \frac{1}{M} \sum_{i=1}^M (R_i^2)^2 - \left( \frac{1}{M} \sum_{i=1}^M R_i^2 \right)^2 \right)} \quad (3.33)$$

Το σφάλμα υπολογίζεται εύκολα είτε προσθέτοντας τον υπολογισμό στο πρόγραμμα `rw.c` είτε κρατώντας το βασικό πρόγραμμα στη δυνατόν απλούστερη μορφή και αφήνοντας την (επαναλαμβανόμενη) "βρομοδουλειά" σε εξωτερικά βοηθητικά προγράμματα (utilities) όπως το παρακάτω αρχείο `average`:

```
#!/bin/awk -f
{
  av += $1;      # the sum of data
  er += $1*$1;  # $ the sum of squares of data
}
END{
  av /= NR;     # NR = "Number of Records" = number of lines
  er /= NR;
  # formula for error of uncorrelated measurements
  er = sqrt( (er - av*av)/(NR-1) );
  print av, "+/-", er;
}
```

<sup>16</sup> Αν υπάρχει στοχαστική εξάρτηση τότε αυτή πρέπει να ληφθεί υπόψη όπως θα δούμε στα επόμενα κεφάλαια.

Το παραπάνω αρχείο είναι παράδειγμα από script το οποίο ερμηνεύεται από κάποιο πρόγραμμα, στη συγκεκριμένη περίπτωση το πρόγραμμα `awk`. Αυτό γίνεται αντιληπτό από το λειτουργικό σύστημα από την πρώτη γραμμή `#!/bin/awk -f` που καλεί το πρόγραμμα όταν εκτελεστεί το αρχείο `average` ως μια οποιαδήποτε εντολή. Για να γίνει αυτό, αφού γράψουμε το αρχείο, του δίνουμε εκτελέσιμη πρόσβαση με την εντολή

```
% chmod a+x average
```

και στη συνέχεια το εκτελούμε λ.χ. πάνω στο αρχείο `dataR`

```
% ./average dataR
```

Οι εντολές που περιέχονται στο αρχείο είναι παρόμοιες με αυτές που θα μπορούσαμε να δώσουμε από τη γραμμή εντολών, όπως δείξαμε παραπάνω που υπολογίσαμε μόνο το μέσο όρο. Οι εντολές που έχουμε μεταξύ `{ ... }` εκτελούνται για κάθε Record (δηλ. εδώ για κάθε γραμμή του προς ανάλυση αρχείου) εκτός αυτές που είναι μεταξύ `END{ ... }` που εκτελούνται αφού διαβαστεί όλο το αρχείο<sup>17</sup>. Οπότε οι γραμμές

```
av += $1;      # the sum of data
er += $1*$1;  # $ the sum of squares of data
```

προσθέτουν στις μεταβλητές `av` και `er` την τιμή της πρώτης στήλης κάθε γραμμής του αρχείου `dataR` και του τετραγώνου αυτής αντίστοιχα. Οι γραμμές

```
av /= NR;     # NR = "Number of Records" = number of lines
er /= NR;
```

εκτελούνται αφού διαβαστεί όλο το αρχείο `dataR` και διαιρούν τις παραπάνω μεταβλητές με την προορισμένη μεταβλητή της `awk` `NR` η οποία σε κάθε στιγμή έχει τιμή ίση με τον αριθμό των γραμμών του αρχείου που έχει επεξεργαστεί το πρόγραμμα. Οι τελευταίες γραμμές του script κάνουν τον τελικό υπολογισμό του σφάλματος σύμφωνα με τη σχέση (3.33) και τυπώνουν το αποτέλεσμα.

Συνοψίζοντας προτείνουμε την παραπάνω σειρά εντολών για κάθε τιμή των `Nstep` που επιλέγουμε (για δεδομένο `Nwalk`):

```
% cc rw.c drandom.c -o rw
% ./rw
% ./average dataR
... collection of <R^2> data with errors in a file r2.dat ...
```

<sup>17</sup>Υπάρχει και η δυνατότητα εκτέλεσης εντολών πριν διαβαστεί το αρχείο (για λ.χ. αρχικοποίηση μεταβλητών) όταν αυτές είναι μεταξύ `BEGIN{ ... }`

```
% gnuplot
gnuplot> plot "dataR" with lines
gnuplot> quit
..after all values of Nwalk have been analyzed and a,b calculated..
% gnuplot
gnuplot> a = <your value>
gnuplot> b = <your value>
gnuplot> plot "r2.dat" using (log($1)):(log($2)):(($3/$2) with err, a*x+b
```

Φυσικά όπως γρήγορα θα φανεί στον επιμελή αναγνώστη, η δουλειά γίνεται γρήγορα πολλή, επαναλαμβανόμενη και ... κουραστική. Για τους ...τεμπέληδες<sup>18</sup> προτείνουμε να μελετήσουν τα σενάρια φλοιού (shell scripts) που θα βρουν στο συνοδευτικό λογισμικό (βλ. σχετικά το `rw1-ana1.csh` αφού όμως αποκτήσουν την εμπειρία της δουλειάς που προτείναμε παραπάνω. Περισσότερα για αυτά στα επόμενα κεφάλαια.

### 3.3.1 Η ΓΕΩΜΕΤΡΙΑ ΤΩΝ ΤΥΧΑΙΩΝ ΔΙΑΔΡΟΜΩΝ

Είμαστε τώρα σε θέση να αναλύσουμε με μεγαλύτερη λεπτομέρεια τη γεωμετρία μιας τυπικής διαδρομής του τυχαίου περιπατητή. Η πρώτη πληροφορία που έχουμε ήδη πάρει είναι η σχέση (3.11)  $\langle R^2 \rangle = N$ . Αυτή έχει ήδη ερμηνευτεί αν τη δούμε σαν σχέση μετατόπισης - χρόνου ότι ο τυχαίος παρατηρητής (υπό τη στατιστική έννοια) δεν κινείται με κλασική ταχύτητα. Τι σχήμα έχει η τυπική τροχιά που ακολουθεί? Είναι μια μονοδιάστατη κλασική τροχιά ή κάτι πιο περίπλοκο? Αν δεχτούμε ότι  $\langle R^2 \rangle$  είναι μια τυπική κλίμακα μήκους που καθορίζει την έκταση του σχήματος που "ζωγραφίζει" ο περιπατητής στο χώρο τότε η σχέση (3.11) υποδεικνύει ότι το μήκος της τροχιάς  $N$  αντιστοιχεί στο τυπικό "εμβαδόν" της περιοχής (αφού δίνει τα σημεία που επισκέπτεται ο παρατηρητής). Οπότε περιμένουμε ότι οι διαδρομές που ακολουθεί ο περιπατητής θα αντιστοιχούν σε μια διδιάστατη επιφάνεια και όχι σε μια μονοδιάστατη καμπύλη! Φυσικά ο πραγματικός τυχαίος περιπατητής δεν περιορίζεται να μετακινείται σε ένα πλέγμα (σε καθορισμένες δηλ. διευθύνσεις με καθορισμένου μήκους βήμα) όπου έχουμε πολλαπλές επισκέψεις των ίδιων σημείων και επικαλύψεις βημάτων, αλλά ελπίζουμε πως στο όριο μεγάλων διαδρομών ότι το μοντέλο μας θα πιάσει τα βασικά χαρακτηριστικά του πραγματικού τυχαίου περιπατητή όπως και όντως συμβαίνει.

Η τροχιά που ακολουθεί ο τυχαίος περιπατητής χαρακτηρίζεται από διάφορες κλίμακες μήκους/εμβαδού που γεννώνται δυνα-

<sup>18</sup>Κυκλοφορεί πως πρόκειται για ... αρετή για όσους δουλεύουν σε βαριούς υπολογισμούς

μικά. Η μία είναι το “εμβαδόν”  $A$  της περιοχής που επισκέπτεται ο περιπατητής. Η ποσότητα αυτή σχετίζεται όπως θα δούμε άμεσα με το μήκος της διαδρομής  $N$  και μπορεί μικροσκοπικά να οριστεί με διάφορους τρόπους. Η άλλη είναι η περίμετρος  $l_p$ , δηλ. τα σημεία εκείνα που βρίσκονται στο όριο της περιοχής που διαγράφει ο περιπατητής. Επίσης η μέση απόσταση σημείων  $l_{av}$  και η μέγιστη απόσταση  $l_{max}$  είναι άλλες κλίμακες μήκους που σχετίζονται με την έκταση της τυχαίας διαδρομής.

Αν πραγματικά έχουμε το μη κλασσικό αποτέλεσμα ότι η τυπική τυχαία διαδρομή είναι διδιάστατη, τότε πρέπει να μελετήσουμε σχέσεις βάθμισης της μορφής

$$A \sim l^d \quad (3.34)$$

όπου “ $\sim$ ” σημαίνει ασυμπτωτική σχέση στο όριο  $N \rightarrow \infty$ . Η διάσταση της τροχιάς  $d$  θα πρέπει να είναι 2 αν οι παραπάνω συλλογισμοί μας είναι σωστοί. Υπάρχει πάντα η δυνατότητα το  $d$  να μην είναι 2 ακόμα ούτε και ακέραιος, κάτι που συμβαίνει σε πιο γενικευμένα μοντέλα τυχαίων διαδρομών. Γενικά το σχήμα της τυχαίας διαδρομής δεν είναι ένα κλασσικό μονοδιάστατο σχήμα αλλά ένα fractal.

Για να μελετήσουμε την γεωμετρία της διαδρομής, χρειάζεται να ορίσουμε το χώρο μέσα στον οποίο κινείται ο περιπατητής. Αυτός θα είναι ένα τετραγωνικό πλέγμα με την τοπολογία του torus όπως στο Σχήμα 1.5. Το πλέγμα έχει  $L$  πλεγματικές θέσεις σε κάθε πλευρά και  $L^2$  συνολικά πλεγματικά σημεία. Η απόσταση  $a$  μεταξύ των πλεγματικών σημείων λαμβάνεται ως μονάδα μέτρησης μήκους ( $a = 1$ ) και οι θέσεις του περιπατητή καθορίζονται από δύο ακεραίους  $(x, y)$ , τις συντεταγμένες του διανύσματος  $\vec{r}$ . Μία τυχαία διαδρομή είναι μια ακολουθία από  $N$  σημεία του πλέγματος:

$$\mathcal{P} = \{\vec{r}_k | k = 1, \dots, N\} \quad (3.35)$$

Οι συνοριακές συνθήκες δεν παίζουν κανένα ρόλο αν πάρουμε  $L > N$ , αλλά στην αντίθετη περίπτωση ο τυχαίος περιπατητής μπορεί να φτάσει στα σύνορα του πλέγματος οπότε στην περίπτωση αυτή πρέπει να ορίζουμε τι συμβαίνει. Φυσικά αν ο τυχαίος περιπατητής περνά συχνά το σύνορο, τότε τα αποτελέσματά μας θα έχουν bias από finite size effects. Από τη σχέση  $\langle R^2 \rangle = N$  αναμένεται τα finite size effects να γίνονται ισχυρά για  $L \sim \sqrt{N}$ <sup>19</sup>.

<sup>19</sup>Για το λόγο αυτό μπορεί η εξοικονόμηση μνήμης/χρόνο CPU στο πρόγραμμα να είναι σημαντική για μεγάλες διαδρομές. Γενικά στους υπολογισμούς μας βρίσκουμε ότι για  $L \approx (5 - 6)\sqrt{N}$  είναι πρακτικά απίθανο να φτάσει ο παρατηρητής στο σύνορο του πλέγματος μια και η πιθανότητα να φτάσει ο παρατηρητής σε απόσταση  $r$  από την αρχή  $P(r) \propto 2\pi r e^{-r^2/(2\langle R^2 \rangle)}$ .



Από τη διαδρομή  $\mathcal{P}$  ορίζεται η συνάρτηση συχνότητας  $h(\vec{r})$  για κάθε σημείο του πλέγματος, που είναι ίση με τον αριθμό φορών που ο τυχαίος περιπατητής επισκέφτηκε το εν λόγω σημείο. Προφανώς

$$\sum_{\vec{r} \in \mathcal{P}} h(\vec{r}) = N \quad (3.36)$$

Είναι βολικό να ορίσουμε την περιοχή του πλέγματος την οποία έχει επισκεφτεί ο περιπατητής:

$$\mathcal{A} = \{\vec{r} \mid h(\vec{r}) > 0\} \quad (3.37)$$

Το εμβαδόν θα το ορίσουμε με τρεις τρόπους, που όπως θα φανεί είναι ισοδύναμοι για  $N \rightarrow \infty$ . Κοιτάζοντας το Σχήμα 3.3 ο αριθμός των σημείων που έχει επισκεφτεί ο περιπατητής μας δίνουν τον πρώτο ορισμό

$$A_1 = \left\langle \sum_{\vec{r}} H(h(\vec{r})) \right\rangle = \left\langle \sum_{\vec{r} \in \mathcal{A}} 1 \right\rangle = \langle N_{\mathcal{A}} \rangle \quad (3.38)$$

όπου  $H(x)$  η συνάρτηση Heavyside

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} . \quad (3.39)$$

Δηλ.  $A_1$  είναι η μέση τιμή του αριθμού των σημείων  $N_{\mathcal{A}}$  που περιέχονται στην περιοχή  $\mathcal{A}$  που επισκέφτηκε ο περιπατητής.

Μπορούμε να ορίσουμε το εμβαδόν να αποτελείται μόνο από εσωτερικά σημεία δηλ.

$$A_2 = \left\langle \sum_{\vec{r}} G_0(\vec{r}) \right\rangle \quad (3.40)$$

όπου η συνάρτηση  $G_0(\vec{r}) = 1$  όταν  $h(\vec{r}) > 0$  ( $\vec{r} \in \mathcal{A}$ ) και  $h(\vec{r}') > 0$  για κάθε πλησιέστερο γείτονα  $\vec{r}'$  του  $\vec{r}$  και 0 σε κάθε άλλη περίπτωση. Τέλος αν και όχι προφανές εξ αρχής ο "σωστότερος" ορισμός για το εμβαδόν είναι ο

$$A_0 = N \quad (3.41)$$

ακόμα και αν ο περιπατητής επισκέπτεται ένα σημείο περισσότερες από μια φορά. Ο λόγος είναι ότι ο τυχαίος περιπατητής όταν επιστρέφει στο ίδιο σημείο αυτό οφείλεται στο γεγονός ότι περιορίσαμε την κίνηση πάνω στο πλέγμα, ενώ αν ήταν ελεύθερος η πιθανότητα να επιστρέψει ακριβώς στο ίδιο σημείο θα ήταν μηδέν. Οπότε σύμφωνα με τους παραπάνω ορισμούς έχουμε μικροσκοπικά ότι

$$A_0 > A_1 > A_2 . \quad (3.42)$$

Αυτό όμως που θα βρούμε είναι ότι για  $N \rightarrow \infty$  οι τρεις ορισμοί είναι ασυμπτωτικά ισοδύναμοι. Αυτό είναι ένα παράδειγμα όπου μια ποσότητα είναι δυνατόν να οριστεί με μη μονοσήμαντο τρόπο στο πλέγμα (μικροσκοπικά) αλλά σε μεγάλες κλίμακες μήκους (οριζόμενες ανάλογα με το πρόβλημα) να είναι ισοδύναμες.

Το ίδιο θα βρούμε στην προσπάθειά μας να ορίσουμε μια κλίμακα μήκους που να ορίζει την “έκταση” της τυχαίας διαδρομής. Ο πρώτος ορισμός είναι η απόσταση της τελικής θέσης του περιπατητή

$$l_0 = \sqrt{\langle R^2 \rangle} \quad (3.43)$$

για την οποία έχουμε ήδη βρει ότι

$$A_0 = l_0^2 \quad (3.44)$$

Στη συνέχεια ορίζουμε το μήκος  $l_1$  να είναι η μέση περίμετρος  $\langle l_p \rangle$ , δηλ. η μέση τιμή του αριθμού των σημείων που βρίσκονται στο σύνορο της περιοχής που περπάτησε ο περιπατητής.

$$l_1 = \langle l_p \rangle = \left\langle \sum_{\vec{r}} G_1(\vec{r}) \right\rangle. \quad (3.45)$$

όπου η συνάρτηση  $G_1(\vec{r}) = 1$  όταν  $h(\vec{r}) > 0$  ( $\vec{r} \in \mathcal{A}$ ) και υπάρχει πλησιέστερος γείτονας  $\vec{r}'$  με  $h(\vec{r}') = 0$ , δηλ. που δεν έχει επισκεφτεί ο περιπατητής.

Μια άλλη κλίμακα μήκους ορίζεται από τη μέση απόσταση των σημείων  $l_{av}$  που ανήκουν στην περιοχή  $\mathcal{A}$  που επισκέφτηκε ο παρατηρητής. Για κάθε ζεύγος  $\vec{r}_i, \vec{r}_j \in \mathcal{A}$  υπολογίζουμε την απόσταση  $r_{ij}^2 = (\vec{r}_j - \vec{r}_i)^2$  και ορίζουμε

$$l_2 = \sqrt{\langle l_{av} \rangle} = \sqrt{\frac{1}{N_{ij}} \left\langle \sum_{\langle ij \rangle} r_{ij}^2 \right\rangle} \quad (3.46)$$

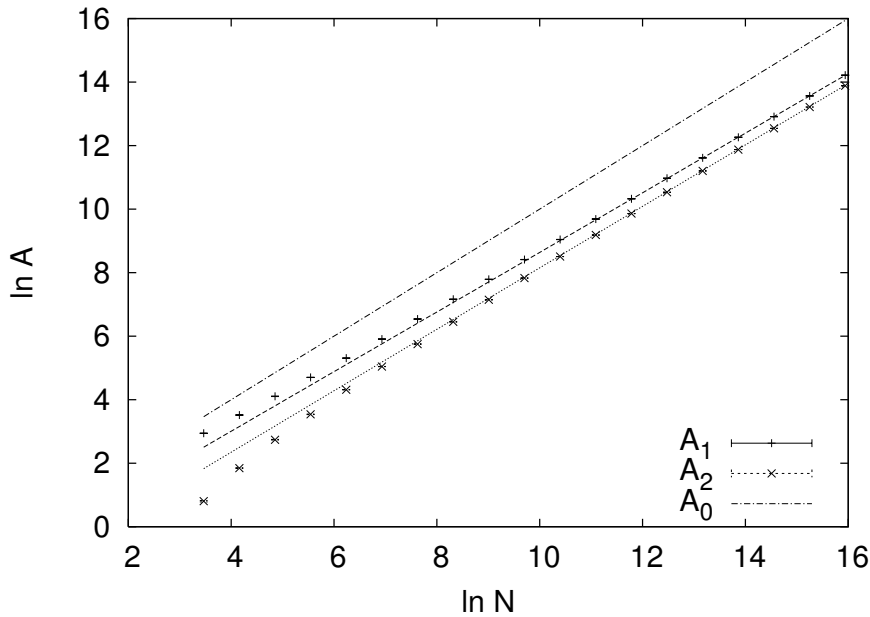
όπου  $N_{ij} = N_{\mathcal{A}}(N_{\mathcal{A}} - 1)/2$  ο αριθμός των ζευγών των σημείων στην  $\mathcal{A}$ .

Τέλος θα ορίσουμε και την κλίμακα μήκους  $l_3$  που ορίζεται από τη μέγιστη απόσταση  $l_{max}$  μεταξύ των σημείων της  $\mathcal{A}$ :

$$l_3 = \langle l_{max} \rangle = \left\langle \max_{\langle ij \rangle} \{r_{ij}\} \right\rangle \quad (3.47)$$

Αναζητούμε τώρα σχέσεις βάθμισης της μορφής

$$A_i \sim N^{a_i} \quad i = 1, 2, \quad (3.48)$$



Σχήμα 3.10: Αριθμητική επιβεβαίωση της σχέσης  $A_i = N^{a_i}$  για  $N = 32, \dots, 8388608$  και  $i = 1, 2$ . Βρίσκουμε ότι  $a_1 = 0.938(2)$ ,  $a_2 = 0.968(2)$ . Η ευθεία γραμμή αντιστοιχεί στο  $A_0 \equiv N$

και

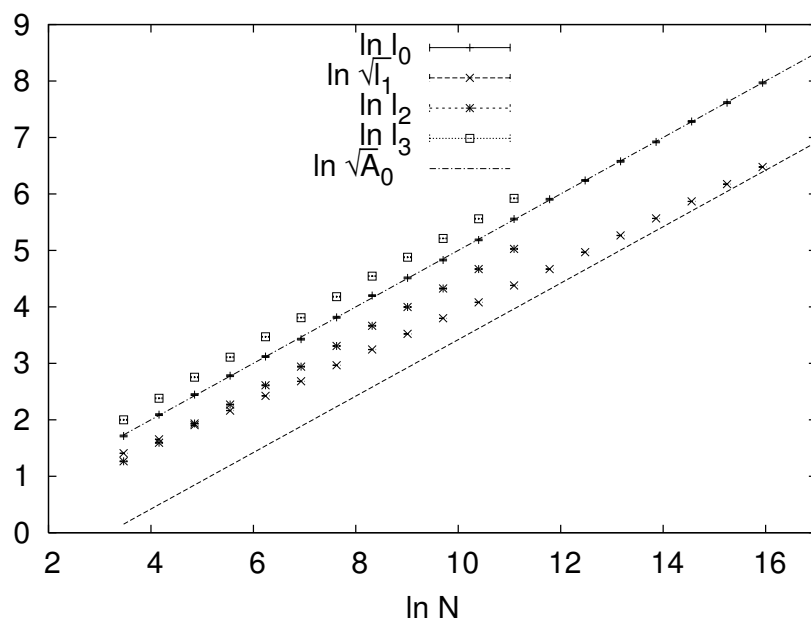
$$l_i \sim N^{b_i} \quad i = 1, 2, 3. \quad (3.49)$$

Αν οι προβλέψεις μας για την γεωμετρία της τυπικής διαδρομής του περιπατητή είναι σωστές τότε  $a_i = 1$  και  $b_i = 1/2$ .

Τα αποτελέσματά μας συνοψίζονται στα Σχήματα (3.3.1) και (3.3.1) καθώς και στον Πίνακα 3.3.1. Παρατηρούμε ότι οι σχέσεις βάθμισης γίνονται φανερές και για μικρές σχετικά διαδρομές  $N \gtrsim 100$ . Οι τιμές που παίρνουμε για τους εκθέτες  $a_i$  και  $b_i$  είναι συνεπείς με τις αναμένουσες εκτός από τον  $b_1$ . Το τελευταίο αυτό αποτέλεσμα υποδηλώνει ότι το περίγραμμα των τυχαίων διαδρομών δεν είναι ένα κλασσικό σχήμα αλλά ένα *fractal* με διάσταση ίδια με αυτή της περιοχής που “περικλείει” δηλ. της  $A$ !

Θα κλείσουμε το κεφάλαιο μελετώντας τις συναρτήσεις συχνότητας  $h(\vec{r})$ . Αυτές θα μας δώσουν ένα μέτρο της τάσης του περιπατητή να επιστρέφει στο ίδιο σημείο και είναι κατά τρόπο τινά μια συνάρτηση πυκνότητας των σημείων που επισκέπτεται ο περιπατητής, κατά την έννοια που είπαμε παραπάνω μια και περιορίζουμε το περιπατητή να κινείται πάνω στο τετραγωνικό πλέγμα.

Στο Σχήμα 3.3.1 βλέπουμε τη συνάρτηση  $h(\vec{r})$  για 4 διαφορε-



Σχήμα 3.11: Αριθμητική επιβεβαίωση της σχέσης  $l_i = N^{b_i}$  με  $N = 32, \dots, 8388608$  για  $i = 0, 1$  και  $N = 32, \dots, 65536$  για  $i = 2, 3$ . Βρίσκουμε ότι  $b_0 = 0.499(3)$ ,  $b_1 = 0.878(4)$ ,  $b_2 = 0.50(1)$ ,  $b_3 = 0.50(1)$ . Παρατηρούμε την ανώμαλη βάρθρωση για το  $l_1$  και τις μεγάλες επιδράσεις πεπερασμένου μεγέθους στις μετρήσεις. Οι ευθείες γραμμές αντιστοιχούν σε βάρθρωση  $\sim \sqrt{N}$

τικές διαδρομές του τυχαίου περιπατητή για  $N = 4096$ . Παρατηρούμε πως τα σημεία που τείνει να επισκέπτεται συχνά είναι συγκεντρωμένα αλλά οι περιοχές αυτές έχουν τυχαία κατανομή (πρβλ. Σχήμα 3.3). Στο Σχήμα 3.3.1 βλέπουμε τη συνάρτηση

$$h_{av}(\vec{r}) = \langle h(\vec{r}) \rangle, \quad (3.50)$$

η οποία δείχνει κυκλική συμμετρία, όπως και αναμένεται, με την περιοχή κοντά στην αρχή να είναι συχνότερα επισκέψιμη.

### 3.3.2 ΑΝΑΛΥΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Θα αναλύσουμε τώρα το πρόγραμμα που χρησιμοποιήσαμε για να πάρουμε τα αποτελέσματα της προηγούμενης παραγράφου.

Για να κάνουμε την ανάλυση της γεωμετρίας των τυχαίων διαδρομών πρέπει να αποθηκεύσουμε στη μνήμη τα σημεία που επισκέφτηκε ο περιπατητής. Επειδή στις μετρήσεις που κάνουμε δεν παίζει ρόλο η σειρά που τα επισκέφτηκε, είναι βολικό να αποθηκεύσουμε απλά τη συνάρτηση  $h(\vec{r})$ . Για το λόγο αυτό πρέπει να

$i$	$a_i$	$b_i$
0	1	0.49(1)
1	0.88(1)	0.76(2)
2	1.03(2)	0.49(1)
3		0.50(1)

Πίνακας 3.1: Αποτελέσματα για τους εκθέτες βάθμισης των σχέσεων  $A_i \sim N^{a_i}$  και  $l_i \sim N^{b_i}$ . Για διδιάστατες διαδρομές περιμένουμε  $a_i = 1$   $b_i = 1/2$ . Φαίνεται η ανώμαλη βάθμιση της περιμέτρου των διαδρομών  $l_1$  με  $b_1 \approx 3/4$ .

βρούμε ένα αποτελεσματικό τρόπο να κωδικοποιήσουμε τις συντεταγμένες των πλεγματικών θέσεων.

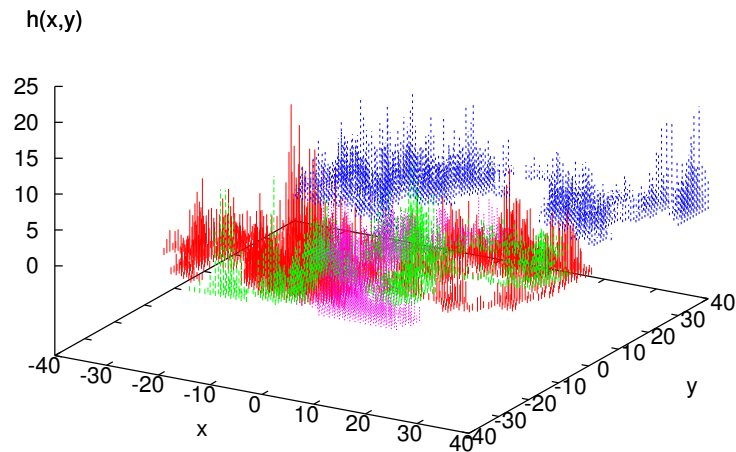
Αν το τετραγωνικό πλέγμα έχει  $L$  πλεγματικές θέσεις σε κάθε πλευρά του, η κάθε πλεγματική θέση μπορεί να κωδικοποιηθεί βάζοντας την αρχή των αξόνων στη μια γωνία του πλέγματος και θέτοντας  $\vec{r} = x\hat{i} + y\hat{j}$ ,  $x, y = 0, \dots, L - 1$ . Προσέξτε τη σύμβαση η μέτρηση να αρχίζει από το 0, μια και γράφουμε τα προγράμματά μας σε C. Στην περίπτωση αυτή η συνάρτηση  $h(\vec{r})$  αντιστοιχεί σε ένα `array h[L][L]` και η τιμή συνάρτησης στο  $\vec{r}$  είναι `h[x][y]` όπου `int x, y`; . Οι συνοριακές συνθήκες επιλέγονται περιοδικές και τότε οι συντεταγμένες  $(x, y)$  ορίζονται απλά `mod L` (το υπόλοιπο της διαίρεσης με  $L$ ). Αυτό φαίνεται στα Σχήματα 3.3.2 και 3.4.

Για να υπολογιστεί όμως μια θέση στο πλέγμα πρέπει να υπολογιστεί η θέση στη μνήμη  $x*L+y$ <sup>20</sup> που εμπεριέχει ένα πολλαπλασιασμό, ο οποίος είναι ακριβός σε χρόνο CPU. Αυτό σε πολλές περιπτώσεις κάνει αυτή τη δομή δεδομένων ασύμφορη<sup>21</sup>. Είναι τότε αποτελεσματικότερο να δίνεται η θέση κάθε σημείου του πλέγματος από ένα μόνο ακέραιο αριθμό  $r = 0, \dots, L^2 - 1$  όπως φαίνεται στο Σχήμα 3.3.2. Στην περίπτωση αυτή συμφέρει να πάρουμε τις συνοριακές συνθήκες ελικοειδείς όπως φαίνεται στο Σχήμα 3.3.2. Η τοπολογία παραμένει τοροειδής αλλά η γεωμετρία είναι διαφορετική όπως φαίνεται στο Σχήμα 3.4.

Μια και ο τυχαίος περιπατητής σε κάθε βήμα του πηδάει σε μια από τις 4 πλησιέστερα γειτονικές του θέσεις, πρέπει να προσδιορίσουμε πώς γίνεται αυτό αλγοριθμικά. Παρατηρούμε ότι αν ο περιπατητής βρίσκεται στη θέση  $r$ , οι πλησιέστεροι γείτονες στις διευθύνσεις  $\pm x$  είναι οι  $r \pm 1$ , ενώ στις διευθύνσεις  $\pm y$  οι  $r \pm L$ . Οι συνοριακές συνθήκες λαμβάνονται πολύ απλά υπ' όψη παρατηρώντας ότι φεύγουμε από το τεταρτημόριο που βρισκόμαστε όταν ο

<sup>20</sup>Προσοχή, η θέση στη μνήμη δεν είναι αριθμητικά ίση με τη θέση  $r$  στο `array h[r]` που αναφέρεται παρακάτω.

<sup>21</sup>Αυτό θα είναι σημαντικό στην περίπτωση του μοντέλου Ising που θα μελετήσουμε αργότερα.



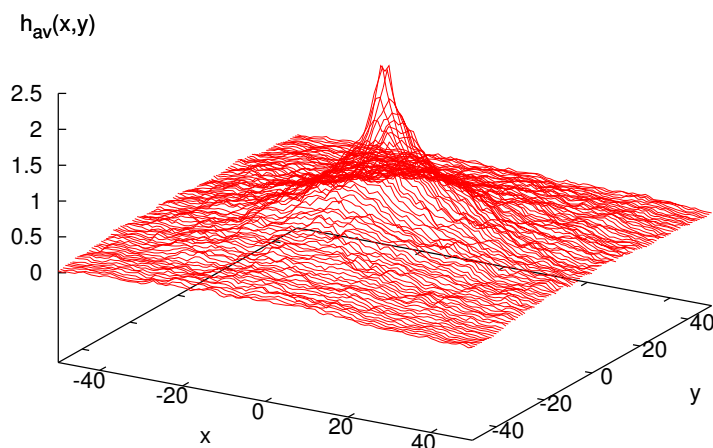
Σχήμα 3.12: Η συνάρτηση  $h(\vec{r})$  για 4 διαφορετικές διαδρομές του τυχαίου περιπατητή για  $N = 4096$ .

δείκτης  $r$  φεύγει από το επιτρεπτό διάστημα  $0, \dots, L^2 - 1$ . Η απλότητα που προσφέρουν οι ελικοειδείς συνοριακές συνθήκες είναι ότι επαναφέροντας τον  $r$  στο σωστό διάστημα προσθέτοντας ή αφαιρώντας ανάλογα  $L^2$ , βρίσκουμε το σωστό πλησιέστερο γείτονα! Η κίνηση στις 4 δυνατές διευθύνσεις αντιστοιχεί τότε στον απλό κώδικα<sup>22</sup>:

```
int r,L,L2;
L2 = L*L;
if(++r >= L2) r -= L2; // Move in +X direction
if(--r < 0) r += L2; // Move in -X direction
if(r+=L >= L2) r -= L2; // Move in +Y direction
if((r-=L) < 0) r += L2; // Move in -Y direction
```

Στο τελευταίο σημείο που πρέπει να δώσουμε προσοχή στον κώδικά μας είναι η αλλαγή συντεταγμένων  $r \rightarrow (x,y)$ . Αυτές θα μας χρειαστούν για τον υπολογισμό των αποστάσεων μεταξύ των πλεγματικών σημείων.

<sup>22</sup>Θυμίζουμε στον αναγνώστη ότι η έκφραση  $(++r)$  έχει τιμή  $r+1$  ενώ η  $(r++)$  έχει τιμή  $r$ . Και στις δύο περιπτώσεις φυσικά στο τέλος η μεταβλητή  $r$  έχει αυξηθεί κατά 1.



Σχήμα 3.13: Η συνάρτηση  $h_{av}(\vec{r})$  για  $N = 4096$ .

Για να ορίσουμε την κίνηση του περιπατητή στις συντεταγμένες αυτές είναι βολικό να περιοριστούμε στις περιπτώσεις όπου ο  $L$  είναι περιττός. Τότε η κίνηση του περιπατητή μπορεί να ξεκινήσει από το κέντρο του πλέγματος το οποίο βρίσκεται στη θέση  $x = [L^2/2]^{23}$  το οποίο θα αντιστοιχήσουμε στη θέση  $(0, 0)$  (βλ. Σχήμα 3.3.2). Στη συνέχεια παρατηρούμε ότι

$$r = y * L + x \quad (3.51)$$

για  $\text{int } x, y, L$ , οπότε η αντίστροφη σχέση  $r \rightarrow (x, y)$  δίνεται από την

$$y = r/L; x = r - y * L; x- = (L/2); y- = (L/2); . \quad (3.52)$$

Θυμίζουμε στον αναγνώστη ότι οι μεταβλητές  $r, x, y, L$  είναι τύπου  $\text{int}$  και ότι η διαίρεση δύο μεταβλητών τύπου  $\text{int}$  δίνει το ακέραιο μέρος του αποτελέσματος. Οπότε η έκφραση  $r/L$  δίνει πόσες σειρές διασχίζουμε στο Σχήμα 3.3.2 για να φτάσουμε στην πλεγματική θέση  $r$  ενώ η έκφραση  $r - y * L$  δίνει το υπόλοιπο της διαίρεσης, άρα τη στήλη που βρίσκεται η πλεγματική θέση  $r$ . Οι δύο τελευταίες πράξεις  $x- = (L/2); y- = (L/2);$  μετατοπίζουν τις συντεταγμένες έτσι ώστε η αρχή των αξόνων να είναι η θέση  $r = (L * L) / 2$ .

<sup>23</sup> $[x]$  είναι το ακέραιο μέρος του πραγματικού  $x$ .

0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
5	6	7	8	9	5	6	7	8	9	5	6	7	8	9
10	11	12	13	14	10	11	12	13	14	10	11	12	13	14
15	16	17	18	19	15	16	17	18	19	15	16	17	18	19
20	21	22	23	24	20	21	22	23	24	20	21	22	23	24
0	1	2	3	4	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	0	1	2	3	4
5	6	7	8	9	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	5	6	7	8	9
10	11	12	13	14	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	10	11	12	13	14
15	16	17	18	19	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	15	16	17	18	19
20	21	22	23	24	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	20	21	22	23	24
0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
5	6	7	8	9	5	6	7	8	9	5	6	7	8	9
10	11	12	13	14	10	11	12	13	14	10	11	12	13	14
15	16	17	18	19	15	16	17	18	19	15	16	17	18	19
20	21	22	23	24	20	21	22	23	24	20	21	22	23	24

Σχήμα 3.14: Τετραγωνικό πλέγμα με περιοδικές συνοριακές συνθήκες και  $L = 5$ . Η τοπολογία είναι τοροϊδής.

Είμαστε έτοιμοι τώρα να παραθέσουμε τον κώδικα που έχουμε στο αρχείο `rw2.c`. Παραλείπουμε μόνο τις συναρτήσεις `drandom()` και `init_random()`. Η πρώτη είναι ήδη γνωστή ενώ τη δεύτερη την αφήνουμε στον αναγνώστη να αρχίζει την `drandom()` κατά το δοκούν...

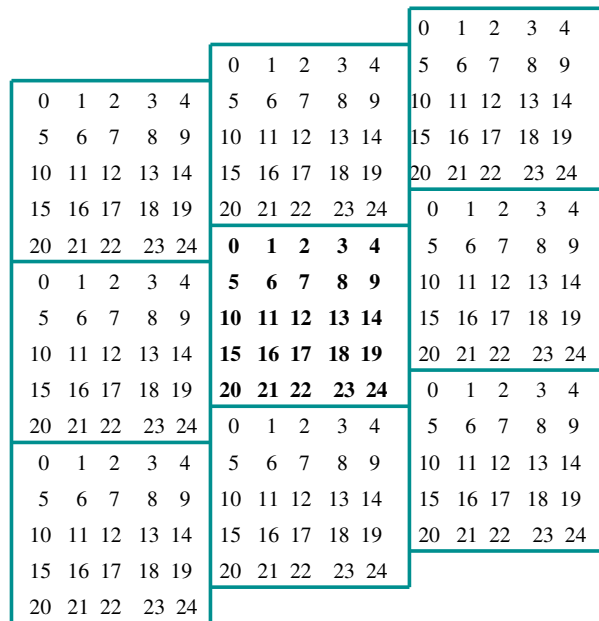
```

/***** rw2.c *****/
#include <stdlib.h>
#include <stdio.h>

/*
  Nwalk: Number of random walks generated.
  N      : Length of random walks
  L      : Linear size of lattice and L2 = L*L
*/
int  Nwalk, N, L, L2;
int  *h; /* The frequency function array h[L2]: How many times each
          point is visited by the random walker */
int  R; /*Final position of walker */
int  iwalk; /*Random walk counter */
int  *hav; /*Average h[r] */
void  init(int, char**),end();
void  generate_a_walk(),measure();

```





Σχήμα 3.15: Τετραγωνικό πλέγμα με ελικοειδείς συνοριακές συνθήκες και  $L = 5$ . Η τοπολογία είναι τοροϊδής.

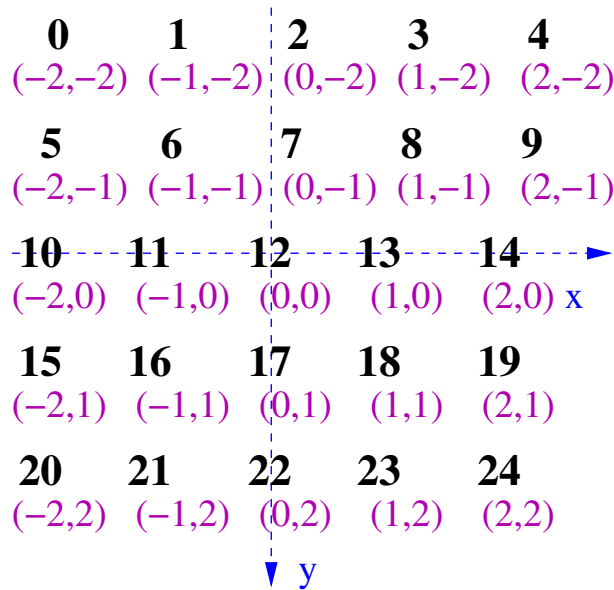
```

double measA1(); /*Area and length functions          */
void  measA2l1(double *,double *),measl2l3(double *,double *);
double drandom();
void  init_random();

/***** main  () *****/
int main(int argc, char **argv){
    init(argc,argv); /*initialize program*/
    for(iwalk=1;iwalk<=Nwalk;iwalk++){
        generate_a_walk();
        measure();
    }
    end();
} //main()

/***** init  () *****/
void init(int argc, char **argv){
    int i;
    if( argc != 4 ){

```



Σχήμα 3.16: Συντεταγμένες  $r = 0, \dots, L^2 - 1$  και συντεταγμένες  $(x, y)$  με  $x, y = -[L/2], \dots, [L/2]$  για περιττό  $L$ . Στο σχήμα  $L = 5$ .

```

fprintf(stderr, "Usage: %s <Nwalk> <N> <L>\n", basename(argv[0]));
exit(1);
}
Nwalk = atoi(argv[1]);
N      = atoi(argv[2]);
L      = atoi(argv[3]);
if( L%2 == 0 ) L++; //we accept odd L only
L2     = L*L;
printf("# Size of lattice: L= %d  L2= %d\n", L, L2);
printf("# Nwalk= %d random walks with Nstep= %d steps\n", Nwalk, N);
/*Allocate memory for the h[L2] array */
h      = (int *)malloc((size_t) (L2*sizeof(int)));
if( h   == NULL){fprintf(stderr, "rw2: Not enough memory!\n");exit(1);}
hav    = (int *)malloc((size_t) (L2*sizeof(int)));
if( hav == NULL){fprintf(stderr, "rw2: Not enough memory!\n");exit(1);}
for(i=0; i<L2; i++) {h[i] = 0; hav[i] = 0;} /*Init to zero*/
init_random(); /*Initialize random number generator*/
}

/***** generate_a_walk() *****/
void generate_a_walk(){
int i, ir, r, r0;

```

```

int boing; //count hits at the edge of lattice
int nret ; //count no. of comeback times
boing=0; nret=0;
for(i=0;i<L2;i++) h[i] = 0;
/*r is the position of the walker on the lattice*/
r = L2/2; /*We place walker at the center of lattice (L2 is odd!)*
r0= r;
for(i=0;i<N;i++){ /*N steps foreach walk*/
  ir=(int)(drandom()*4);
  switch(ir){
  case 0: // Move in +X direction
    if((++r )>=L2){r -= L2;boing++;};//carefull ++r not r++!
    h[r]++;
    break;
  case 1: // Move in -X direction
    if((--r )< 0 ){r += L2;boing++;};//carefull --r not r--!
    h[r]++;
    break;
  case 2: // Move in +Y direction
    if((r+=L)>=L2){r -= L2;boing++;}
    h[r]++;
    break;
  case 3: // Move in -Y direction
    if((r-=L)< 0 ){r += L2;boing++;}
    h[r]++;
    break;
  }//switch(ir)
  //printf("w %d %d %d\n",iwalk,i,r);//if you wish to print the path
  if(r == r0) nret++;
} //for(i=0;i<N;i++)
R = r;
printf("boing= %d  iwalk= %d\n",boing,iwalk);
printf("nret= %d\n",nret);
} //generate_a_walk
/***** measure() *****/
void measure(){
  int x,y; //translate r-position to x,y-position on the lattice
  int i,Lh;
  double xr,yr;
  double l0,l1,l2,l3,A0,A1,A2;
  double R2;

  Lh = L/2;
  /* Final position x,y,R2:          */

```

```

y = R/L ; x = R - y*L;
y -= Lh ; x -= Lh;//origin at the center of lattice
yr = (double)y; xr = (double)x;
R2 = xr*xr+yr*yr;
printf ("R %d %f %d %d %d\n",iwalk,R2,x,y,R);

/* Area and length measurements: */
A0= (double)N; l0 = R2;
A1 = measA1(); measA2l1(&A2,&l1); measl2l3(&l2,&l3);
printf("A1 %d %f %f %f %f %f %f %f\n",iwalk,A0,A1,A2,l0,l1,l2,l3);

/* Average histogram */
for(i=0;i<L2;i++){
    hav[i] += h[i];
    if(iwalk <= 10 ){
        y=i/L;x=i-y*L;x-=Lh;y-=Lh;//if you wish to print all hs'
        printf("hst %d %d %d %d\n",iwalk,x,y,h[i]);
    }//if(iwalk <= 10 )
} //for(i=0;i<L2;i++)
} //measure

/**** Area and length computation functions *****/
/**** A1 = # of sites that rw has visited *****/
double measA1(){
    int icount,i;
    icount=0;
    for(i=0;i<L2;i++)
        if(h[i] > 0 ) icount++;
    return (double)icount;
} //A1()

/* A2 = # of sites that are internal to region rw has visited */
/* l1 = # of boundary sites, i.e. at least one nn not visited */
void measA2l1(double *A2, double *l1){
    int icountA,icountl,i,nn,sum;
    icountA = 0; icountl = 0;
    for(i=0;i<L2;i++){
        sum = 0;
        if(h[i]>0){
            /*Visit 4 neighbors. If all visited then site is internal*/
            if((nn=i+1)>= L2) nn-=L2; if( h[nn] > 0 ) sum++ ;
            if((nn=i-1)< 0 ) nn+=L2; if( h[nn] > 0 ) sum++ ;
            if((nn=i+L)>= L2) nn-=L2; if( h[nn] > 0 ) sum++ ;
            if((nn=i-L)< 0 ) nn+=L2; if( h[nn] > 0 ) sum++ ;

```

```

        if( sum == 4      ) icountA++;
        else              icountl++;
    }//if(h[i]>0)
}//for(i=0;i<L2;i++)
*A2 = (double)icountA; *l1 = (double)icountl;
}//A2l1()

/*l2 is the average of the distances squared          */
/*l3 is the max distance squared between points      */
void measl2l3(double *l2, double *l3){
    int i, j;
    int xi, yi, xj, yj;
    double norm2;
    double xij, yij, rij;
    double len2, len3;

    len2=0.0;len3=0.0;norm2=0.0;
    for(i=0;i<L2;i++){
        if( h[i] > 0 ){
            yi=i/L;xi=i-yi*L;    //i-coordinates
            for(j=i+1;j<L2;j++){
if( h[j] > 0 ){
                yj=j/L; xj=j-yj*L;//j-coordinates
                xij = (double)(xj-xi); yij = (double)(yj-yi);
                rij = xij*xij + yij*yij;
                len2 += rij;
                norm2 += 1.0; //Count number of pairs
                if( rij > len3) len3 = rij;
            }//if( h[j] > 0 )
        }//for(j=i+1;j<L2;j++)
    }//if( h[i] > 0 )
}//for(i=0;i<L2;i++)
    *l2 = len2/norm2;
    *l3 = len3;
}//l2l3()

/***** end() *****/
void end(){
    int i,x,y,Lh;
    Lh=L/2;
    /*Print average histograms*/
    for(i=0;i<L2;i++){
        y=i/L;x=i-y*L;x-=Lh;y-=Lh;
        printf("hav %d %d %f\n",x,y,((double)hav[i])/((double)Nwalk));
    }
}

```

```

} //for(i=0;i<L2;i++)
} //end

```

Για να τρέξουμε το πρόγραμμα μεταγλωττίζουμε και τρέχουμε ως εξής:

```

% cc -O2 rw2.c -o rw2
% ./rw2 <Nwalk> <N> <L> >& rwalk2.dat &

```

όπου <Nwalk>, <N> και <L> είναι ο αριθμός των παραγομένων τυχαίων διαδρομών, το μήκος τους και η πλευρά του τετραγωνικού πλέγματος αντίστοιχα. Μερικές επεξηγήσεις πάνω σε τεχνικές λεπτομέρειες του παραπάνω προγράμματος:

Η δομή του προγράμματος είναι απλή. Η συνάρτηση `main()` καλεί αρχικά την `init()` που αρχικοποιεί τις απαραίτητες μεταβλητές καθώς την γεννήτρια των ψευδοτυχαίων αριθμών `drandom()`. Μετά καλεί `Nwalk` φορές τη συνάρτηση `generate_a_walk()` που παράγει μια τυχαία διαδρομή και τη συνάρτηση `measure()` που αναλαμβάνει τις μετρήσεις πάνω σε κάθε διαδρομή. Τέλος η συνάρτηση `end()` κάνει τις τελευταίες δουλειές πριν τον τερματισμό του προγράμματος.

Στην `init()` διαβάζουμε τις βασικές παραμέτρους `Nwalk`, `N`, `L`, διαβάζοντας τα ορίσματα `argv` της εντολής που τρέχει το πρόγραμμα και μετατρέποντάς τα σε ακεραίους με την `atoi()` όπως εξηγήσαμε στη σελ. 44. `Nwalk` είναι ο αριθμός των τυχαίων διαδρομών που θα παράγει το πρόγραμμα, `N` το μήκος τους `N` και `L` ο αριθμός των πλεγματικών θέσεων σε κάθε πλευρά του τετραγώνου που ορίζει το βασικό πλέγμα. Στη συνέχεια ζητάμε την απαιτούμενη μνήμη για τα arrays `h[L*L]` και `hav[L*L]` που αποθηκεύουν τη συνάρτηση  $h(\vec{r})$  και  $h_{av}(\vec{r}) = \langle h(\vec{r}) \rangle$  αντίστοιχα. Επειδή το μέγεθος αυτό δεν είναι γνωστό εκ των προτέρων, ζητούμε τη μνήμη δυναμικά χρησιμοποιώντας τη συνάρτηση `malloc()`.

```

int L, L2, *h, *hav;
L2   = L*L;
h    = (int *)malloc((size_t) (L2*sizeof(int)));
hav  = (int *)malloc((size_t) (L2*sizeof(int)));

```

Τα `h` και `hav` είναι `pointers` σε `int` και χρησιμοποιούμε τα αναγκαία casts `(int *)` για τις απαραίτητες μετατροπές. Το macro `sizeof()` δίνει τον αριθμό των bytes που καταλαμβάνει το όρισμά του, στην περίπτωση μας των `int`. Το όρισμα της `malloc()` είναι ο αριθμός των bytes που ζητάμε από τη μνήμη του υπολογιστή και η τιμή της ένα `pointer` στη θέση της μνήμης που μας δόθηκε.

Στη συνάρτηση `generate_a_walk()` επιλέγουμε την τυχαία διεύθυνση όπως στο πρόγραμμα `rw.c`, ενώ η αντίστοιχη κίνηση γίνεται

όπως αναφέρθηκε παραπάνω. Η μεταβλητή `boing` μετράει πόσες φορές ο τυχαίος περιπατητής περνάει το σύνορο του πλέγματος. Με αυτή ελέγχουμε τα `finite size effects`. Αναφέραμε παραπάνω ότι επιλέγοντας  $L \sim (5-6)\sqrt{N}$  για  $N_{\text{walk}} \approx 1000$ ,  $N \lesssim 10^5$  τα `finite size effects` είναι αμελητέα. Το array `h[r]` ενημερώνεται σε κάθε βήμα. Για κίνηση στην  $+x$  διεύθυνση έχουμε

```
case 0: // Move in +X direction
    if((++r )>=L2){r -= L2;boing++;}//carefull ++r not r++!
    h[r]++;
```

Η μεταβλητή `nret` μετράει πόσες φορές ο περιπατητής επιστρέφει στο αρχικό σημείο. Τέλος σημειώνουμε την τελική θέση του περιπατητή στη `global` μεταβλητή `R` για επεξεργασία από την `measure()`

Στη `measure()` μετράμε το τετράγωνο της απόστασης  $R^2$  που διανύει ο περιπατητής στο τέλος της διαδρομής. Για το λόγο αυτό υπολογίζουμε και τις συντεταγμένες  $(x, y)$  σύμφωνα με τον αλγόριθμο που δώσαμε παραπάνω:

```
int Lh, x, y;
double yr,xr, R2;
Lh = L/2;
y = R/L ; x = R - y*L; y -= Lh ; x -= Lh;
yr = (double)y; xr = (double)x;
R2 = xr*xr+yr*yr;
```

Η μετατροπή σε `double` γίνεται για να μην ξεπεράσουμε το άνω όριο των μεταβλητών `int`.

Στη συνέχεια υπολογίζουμε τις ποσότητες  $A_1$ ,  $A_2$ ,  $l_1$ ,  $l_2^2$  και  $l_3^2$ . Προσοχή στην ανάλυση για τα μήκη  $l_i$ , είναι όλα στο τετράγωνο εκτός από το  $l_1$  (αφήσαμε τις τετραγωνικές ρίζες για τα προγράμματα της ανάλυσης!). Αυτό γίνεται στις συναρτήσεις `measA1()`, `measA2l1()` και `measl2l3()`. Τέλος δεν ξεχνάμε τις απαραίτητες πράξεις για τον υπολογισμό της μέσης τιμής `hav[r]` της  $h(\vec{r})$ .

Στη συνάρτηση `measA1()` αρκεί να βρούμε πόσα σημεία έχει επισκεφτεί ο περιπατητής. Η συνθήκη είναι απλή, `h[i]>0`, οπότε αρκεί να μετρήσουμε τα σημεία αυτά:

```
for(i=0;i<L2;i++)
    if(h[i] > 0 ) icount++;
return (double)icount;
```

Στη συνάρτηση `measA2l1()` μετράμε τον αριθμό των εσωτερικών σημείων `A2` και συνοριακών σημείων `l1` στην περιοχή  $\mathcal{A}$ . Η συνθήκη για ένα σημείο να ανήκει στο εσωτερικό της  $\mathcal{A}$  είναι να ανήκει στην  $\mathcal{A}$ , `h[i]>0` και κάθε πλησιέστερος γείτονας `nn` να ανήκει

στην  $A$ ,  $h[nn]>0$ . Η συνθήκη για ένα σημείο να ανήκει στο σύνορο της  $A$  είναι να ανήκει στην  $A$  και να έχει λιγότερους από τέσσερις πλησιέστερους γείτονες να ανήκουν στην  $A$ . Οι πλησιέστεροι γείτονες βρίσκονται με τον ίδιο αλγόριθμο που υπολογίζουμε τα βήματα του περιπατητή και η μεταβλητή `sum` υπολογίζει τον αριθμό των πλησιεστέρων γειτόνων. Μια υπενθύμιση στον αναγνώστη: Η συνάρτηση `measA211()` καλείται από το πρόγραμμα ως

```
measA211(&A2,&l1);
```

για `double A2, l1`. Επειδή θέλουμε η `measA211()` να αλλάξει την τιμή των παραπάνω μεταβλητών και στη C οι συναρτήσεις περνούν τα ορίσματά τους “by value” και όχι “by reference” (όπως λ.χ. η Fortran) πρέπει μέσα στην `measA211()` να περάσουμε τα pointers προς τις παραπάνω μεταβλητές. Ο συμβολισμός `&A2` σημαίνει το pointer προς τη θέση μνήμης που βρίσκεται η μεταβλητή `A2`. Η `measA211()` παίρνει όρισμα τα pointers προς τις αντίστοιχες μεταβλητές και μεταβάλλει τις τιμές που έχουν αποθηκευτεί σε αυτές:

```
void measA211(double *A2, double *l1){
    ...
    *A2 = (double)icountA; *l1 = (double)icountl;
}
```

Με ανάλογο τρόπο στην `meas1213()` γίνεται και ο υπολογισμός (των τετραγώνων) των μέσων και μέγιστων αποστάσεων  $12,13$  μεταξύ των σημείων της  $A$ . Εδώ προσδιορίζονται τα ζεύγη των σημείων  $i, j$  που ανήκουν στην  $A$  ( $h[i]>0$  και  $h[j]>0$ ) από τα nested loops:

```
for(i=0;i<L2;i++){
    if( h[i] > 0 ){
        for(j=i+1;j<L2;j++){
            if( h[j] > 0 ){
                .... calculate distance squared ....
            }
        }
    }
}
```

όπου “.... calculate distance squared ....” υπολογίζει την απόσταση από τις συντεταγμένες των σημείων στο  $(x, y)$  σύστημα. Παρατηρούμε ότι για τη συνάρτηση αυτή και μόνο συμφέρει η δομή δεδομένων βασισμένη στο  $(x, y)$  σύστημα συντεταγμένων. Για τη σωστή κανονικοποίηση μετράμε τα ζεύγη των σημείων με



τη μεταβλητή `norm2` ενώ μετατρέπουμε τις συντεταγμένες σε τύπου `double` για να μην έχουμε προβλήματα υπερχείλισης ακεραίων (`overflow`).

Τα δεδομένα μας τυπώνονται στην καθιερωμένη έξοδο (`stdout`) και όχι σε ξεχωριστά αρχεία. Κάθε γραμμή `printf` αρχίζει με μία συγκεκριμένη ακολουθία χαρακτήρων έτσι ώστε με τη χρήση φίλτρων να παίρνουμε την πληροφορία που μας ενδιαφέρει. Οι εντολές εξόδου

```
printf("boing= %d iwalk= %d\n",boing,iwalk);
printf("nret= %d\n",nret);
printf("R %d %f %d %d %d\n",iwalk,R2,x,y,R);
printf("A1 %d %f %f %f %f %f %f %f\n",iwalk,A0,A1,A2,10,11,12,13);
```

παράγουν στην έξοδο (που επαναορίσαμε να είναι το αρχείο `rwalk2.dat`) γραμμές της μορφής

```
boing= 0 iwalk= 1
nret= 0
R 1 5.000 -2 1 15
A1 1 5.000 3.000 0.000 5.000 3.000 9.333 17.000
```

οπότε με τις εντολές

```
% grep ^boing rwalk2.dat
% grep ^nret rwalk2.dat
```

εξετάζουμε την εξέλιξη του αριθμού που ο τυχαίος περιπατητής περνάει το σύνορο του πλέγματος και του αριθμού των φορών `nret` που επιστρέφει στο σημείο από το οποίο ξεκίνησε.

Για να πάρουμε τις μέσες τιμές και τα σφάλματα των μετρήσιμων ποσοτήτων χρησιμοποιούμε εντολές της μορφής

```
% grep ^R rwalk2.dat | awk '{print $3}' | ./average
% grep ^A1 rwalk2.dat | awk '{print $7}' | ./average
```

όπου η πρώτη υπολογίζει το  $\langle R^2 \rangle \pm \delta \langle R^2 \rangle$  και η δεύτερη το  $l_1 \pm \delta l_1$  (Το `script average` το ορίσαμε στη σελ. 59). Για να δούμε τις αντίστοιχες χρονοσειρές:

```
% gnuplot
gnuplot> plot "<grep ^R rwalk2.dat | awk '{print $3}'" with lines
gnuplot> plot "<grep ^A1 rwalk2.dat | awk '{print $7}'" with lines
```

Στο σημείο αυτό ο προσεκτικός αναγνώστης έχει όλα τα εφόδια για να κάνει την άσκηση 10 της σελ. 85 την οποία πρέπει οπωσδήποτε να κάνει.

### 3.3.3 ΠΑΡΑΛΛΑΓΕΣ ΤΟΥ ΜΟΝΤΕΛΟΥ

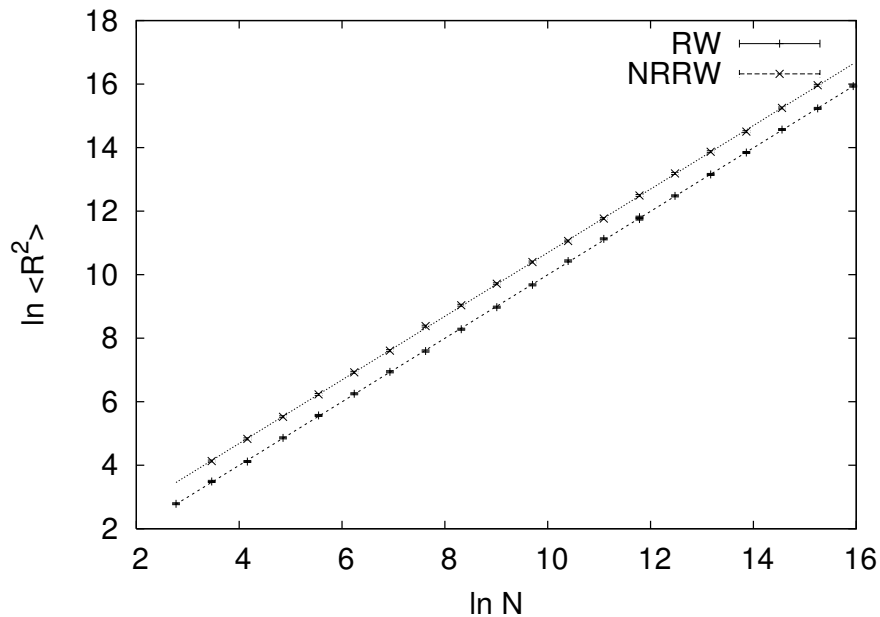
Στην παράγραφο αυτή θα συζητήσουμε τις αλλαγές που επιφέρουν μεταβολές στο μικροσκοπικό ορισμό του μοντέλου του τυχαίου περιπατητή (RW) στη συμπεριφορά που έχει για  $N \rightarrow \infty$ .

Η πρώτη ερώτηση που δημιουργείται είναι ποιες οι αλλαγές που επιφέρει η απαίτηση ο περιπατητής να μην επιστρέφει άμεσα στο σημείο από το οποίο ξεκίνησε. Αυτό οδηγεί στο μοντέλο του μη επιστρέφοντος τυχαίου περιπατητή (NRRW, Nonreversal Random Walking). Στο πρότυπο αυτό το επόμενο βήμα του περιπατητή μπορεί να είναι προς οποιαδήποτε διεύθυνση με ίση πιθανότητα εκτός από αυτή από την οποία ήρθε. Το κίνητρο για τον ορισμό αυτό είναι ότι έτσι μπορούμε να αποφύγουμε τις επικαλύψεις διαδρομών, κάτι που οφείλεται στο γεγονός ότι περιορίζουμε τον περιπατητή στο πλέγμα και που μπορεί να εισάγει τεχνητές επιδράσεις στις μετρήσεις που δεν θα υπήρχαν στο ρεαλιστικό περιπατητή (που κινείται δηλ. προς οποιαδήποτε διεύθυνση). Παρ' όλα αυτά όπως θα φανεί το πρότυπο αυτό δεν έχει καμιά ουσιαστική διαφορά από αυτό που ήδη μελετήσαμε. "Ουσιαστική" σημαίνει πως σε μακροσκοπικές κλίμακες οι RW και NRRW έχουν τις ίδιες ιδιότητες. Το αποτέλεσμα του περιορισμού στον NRRW είναι απλά οι ιδιότητες αυτές να εμφανίζονται σε ελαφρά μεγαλύτερες κλίμακες μήκους. Για να γίνουμε πιο συγκεκριμένοι, ορίζοντας με τον ίδιο τρόπο τους εκθέτες  $a_i$  και  $b_i$  στις σχέσεις (3.48) και (3.49) βρίσκουμε ότι έχουν τις ίδιες τιμές και στα δύο πρότυπα RW και NRRW. Αυτό φαίνεται αρχικά στο Σχήμα 3.3.3 όπου βλέπουμε ότι και για τα δύο μοντέλα

$$\langle R^2 \rangle \sim aN \quad (3.53)$$

όπου  $a = 1$  για τον RW και  $a = 1.98(4)$  για τον NRRW. Το γεγονός ότι  $a > 1$  για τον NRRW σημαίνει ότι ο τελευταίος "απλώνεται" περισσότερο από τον RW αλλά ότι οι διαδρομή που ακολουθεί είναι όμοια με αυτή του RW σε μεγαλύτερη όμως κλίμακα. Αυτό φαίνεται και στα Σχήματα 3.3.3 και 3.3.3 που είναι όμοια με τα Σχήματα 3.3.1 και 3.3.1 για τον RW.

Η κωδικοποίηση του NRRW γίνεται χωρίς δυσκολία με απλή μετατροπή του προγράμματος `rw.c`. Απλά χρειάζεται σε κάθε βήμα να κρατάμε στη μνήμη τη διεύθυνση από την οποία προήλθε ο περιπατητής. Για απλότητα οι τέσσερις διευθύνσεις  $+x$ ,  $+y$ ,  $-x$  και  $-y$  κωδικοποιούνται στους ακεραίους  $0, \dots, 3$  αντίστοιχα. Αν ο περιπατητής προέρχεται από τη διεύθυνση `dir` τότε το επόμενο βήμα γίνεται σε μία από τις τρεις διευθύνσεις `dir-1`, `dir`, `dir+1` όπου η πράξη  $\pm$  γίνεται  $\text{mod } 4$ . Η μεταβολή γίνεται στην συνάρτηση `generate_a_walk()` ως εξής:

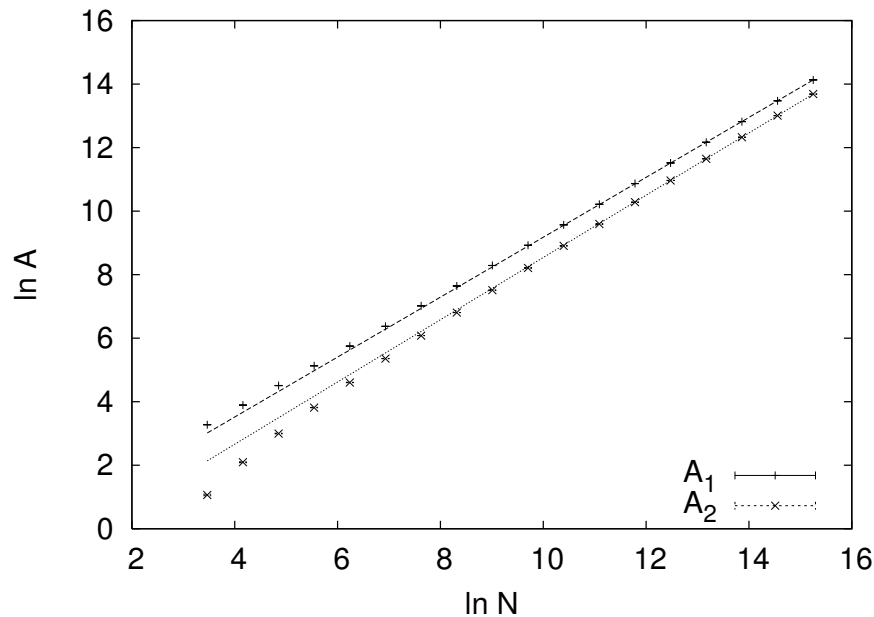


Σχήμα 3.17: Μελέτη της σχέσης  $\langle R^2 \rangle \sim aN$  για τα πρότυπα RW και NRRW. Παρατηρούμε ότι για το πρότυπο NRRW η μόνη διαφορά είναι στον αριθμητικό συντελεστή  $a = 1.98(4)$  ο οποίος είναι 1 για τον RW.

```

r = L2/2; r0= r;
/*dir is the stored direction of the previous step*/
dir=(int)(drandom()*4);
for(i=0;i<N;i++){
  /* 3 possible choices: (dir-1)%4,(dir)%4,(dir+1)%4 */
  /* ir always >0 for % operation, we add 4 */
  ir=dir+3+(int)(drandom()*3);
  ir %= 4;
  dir = ir;//Stored for next move
  //The four directions are now stored counterclockwise:
  switch(ir){
  case 0: // Move in +X direction
    if((++r )>=L2){r -= L2;boing++;}//carefull ++r not r++!
    h[r]++;
    break;
  case 2: // Move in -X direction
    if((--r )< 0 ){r += L2;boing++;}//carefull --r not r--!
    h[r]++;
    break;
  case 1: // Move in +Y direction

```



Σχήμα 3.18: Αριθμητική επιβεβαίωση της σχέσης  $A_i = N^{a_i}$  για  $N = 32, \dots, 4194304$  και  $i = 1, 2$  για τον NRRW. Βρίσκουμε ότι  $a_1 = 0.943(2)$ ,  $a_2 = 0.981(4)$ .

```

    if((r+=L)>=L2){r -= L2;boing++;}
    h[r]++;
    break;
case 3: // Move in -Y direction
    if((r-=L)< 0 ){r += L2;boing++;}
    h[r]++;
    break;
} //switch(ir)

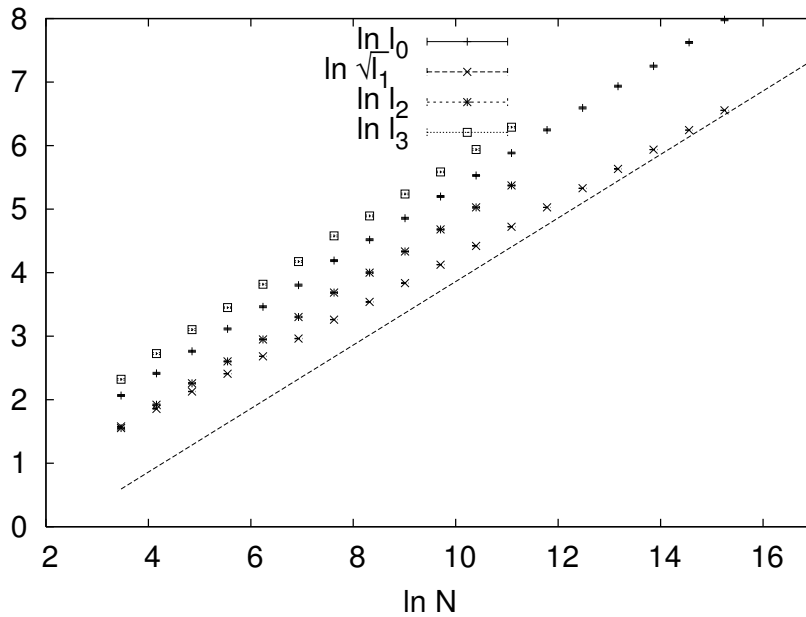
```

Οι μόνες μεταβολές που κάναμε αφορούν την αποθήκευση της διεύθυνσης στη μεταβλητή `int dir`, την αντιστοιχία των ακεραιών  $0, \dots, 3$  στις διευθύνσεις με φορά αντίθετη της φοράς κίνησης των δεικτών του ρολογιού και την επιλογή της διεύθυνσης κίνησης `ir` να είναι οι `dir-1`, `dir`, `dir+1`. Ειδικά το τελευταίο το πετυχαίνουμε με την πράξη

```
ir = dir - 1 + ran + 4, ran = 0,1,2
```

και κατόπιν παίρνοντας το υπόλοιπο της διαίρεσης με το 4 με την πράξη `mod 4`. Το 4 το προσθέτουμε αφού  $4 \bmod 4 = 0$  και θέλουμε το `ir` να είναι θετικό όταν κάνουμε την πράξη `ir = ir % 4`.

Θα κλείσουμε την ενότητα αυτή με μια αναφορά στον "Αυτο-αποφεύγων Τυχαίος Περιπατητής (Self Avoiding Random Walk -



Σχήμα 3.19: Αριθμητική επιβεβαίωση της σχέσης  $l_i = N^{b_i}$  για  $N = 32, \dots, 4194304$  και  $i = 0, 1$  και για  $N = 32, \dots, 65536$  και  $i = 2, 3$  για τον NRRW. Βρίσκουμε ότι  $b_0 = 0.500(2)$ ,  $b_2 = 0.500(3)$ ,  $b_3 = 0.503(3)$ .

SAW). Τέτοιες διαδρομές μπορούμε να δημιουργήσουμε ξεκινώντας από ένα σημείο του πλέγματος και σταματώντας όποτε ο SAW επισκέπτεται μια πλεγματική θέση στην οποία έχει ήδη βρεθεί. Τέτοιες διαδρομές θυμίζουν την κανονική συλλογή ενός συστήματος που προσομοιάζει μια λ.χ. πρωτεΐνη σε αρκετά μεγάλη θερμοκρασία έτσι ώστε να μπορέσουμε να αγνοήσουμε την έλξη μεταξύ των ατόμων της. Το πρόβλημα με την προσέγγιση αυτή είναι ότι οι διαδρομές με μεγάλο μήκος  $N$  είναι πιθανοθεωρητικά εκθετικά καταπιεσμένες και έτσι δεν μπορούμε να τις μελετήσουμε για μεγάλο  $N$ . Πράγματι μπορεί ναδειχθεί ότι ο αριθμός των διαδρομών μήκους  $N$  ασυμπτωτικά για  $N \rightarrow \infty$  έχει την συμπεριφορά

$$Z_N^{\text{SAW}} \sim N^{\gamma-1} z_{\text{eff}}^N \quad (3.54)$$

ενώ ο αντίστοιχος αριθμός των διαδρομών μήκους  $N$  για τον RW είναι

$$Z_N^{\text{RW}} = z^N \quad (3.55)$$

όπου  $z = 4$  ο αριθμός πλησιεστέρων γειτόνων στο πλέγμα. Ο εκθέτης  $\gamma$  είναι ένας κρίσιμος εκθέτης ενώ ο  $z_{\text{eff}} < z - 1$  είναι μη ακέραιος. Η πιθανότητα μια RW διαδρομή να είναι SAW είναι ο

λόγος των παραπάνω ποσοτήτων:

$$p_N = \frac{Z_N^{\text{SAW}}}{Z_N^{\text{RW}}} \sim N^{\gamma-1} \left( \frac{z^{\text{eff}}}{z} \right)^N = \exp \left( -N \ln \frac{z^{\text{eff}}}{z} + (\gamma - 1) \ln N \right), \quad (3.56)$$

η οποία μειώνεται εκθετικά με το  $N$ . Για να επιτύχουμε στατιστικά ικανοποιητική συλλογή δεδομένων, η προσπάθειά μας αυξάνει εκθετικά με το  $N$ . Το μέγιστο  $N$  που μπορεί να μελετήσουμε με τον τρόπο αυτό είναι  $\sim 100$ . Το πρόγραμμα `saw2.c` μπορεί να χρησιμοποιηθεί για την προσομοίωση του μοντέλου.

### 3.4 ΠΡΟΒΛΗΜΑΤΑ

1. Εκτελέστε την εντολή:

```
% cpp hello.c > hello1.c
% less hello1.c
```

Εξετάστε τα περιεχόμενα του αρχείου `hello1.c` και συγκρίνετέ τα με τα περιεχόμενα των αρχείων `/usr/include/stdlib.h` `/usr/include/stdio.h`. Τι ακριβώς συνέβη?

2. Αναπαράγετε τα αποτελέσματα που οδηγούν στα Σχήματα 3.2.2 και 3.2.2 και φτιάξτε τα αντίστοιχα σχήματα. (Συνοδευτικό λογισμικό: `study-ran-fluct`, `plot`)
3. Από το συνοδευτικό λογισμικό εκτελέστε τις εντολές<sup>24</sup>:

```
% ./naiveran 100000 > data &
% ./eternal-ran
```

και στη συνέχεια

```
% ./drandom 100000 > data &
% ./eternal-ran
```

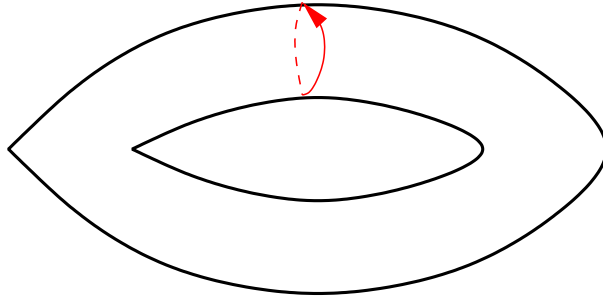
4. Κωδικοποιήστε την παραγωγή τυχαίων αριθμών με κατανομή Gaussian και Cauchy ( $c = 1$ ). Φτιάξτε ιστογράμματα κατανομής των αριθμών. Δείξτε ότι οι σχέσεις (3.22) και (3.23) δίνουν τις ίδιες κατανομές. (Συνοδευτικό λογισμικό: `gausran.c`)

<sup>24</sup>Μπορείτε να πατήσετε `Ctrl-c` για να σταματήσει το πρόγραμμα `eternal-ran`.

5. Αποδείξτε ότι αν η μέση τιμή των διανυσμάτων  $\langle \vec{\xi}_i \rangle = \vec{v}\tau$  τότε  $\langle \vec{R} \rangle = \vec{v}\tau N$  και έχουμε γραμμική σχέση μεταξύ μετατόπισης-μήκους τροχιάς. Η ποσότητα  $\vec{v}$  είναι η μέση ταχύτητα του σωματιδίου. Υπολογίστε το  $\langle R^2 \rangle$  για μεγάλες τιμές του  $N$ .
6. Εξετάστε αριθμητικά τις παραπάνω σχέσεις θέτοντας στον κώδικά σας την πιθανότητα στην πρώτη γραμμή της (3.2) ίση με  $1/2$  και τις υπόλοιπες  $1/6$ . Υπολογίστε τις ποσότητες  $\langle (\xi_i)_x \rangle$ ,  $\langle (\xi_i)_y \rangle$  και από αυτές τη μέση ταχύτητα του σωματιδίου. Εξετάστε τις σχέσεις  $\langle R^2 \rangle \sim N^\alpha$  και  $\langle R_x \rangle \sim N^{2a_x}$   $\langle R_y \rangle \sim N^{2a_y}$ . Ποια η σχέση μεταξύ  $a$ ,  $a_x$  και  $a_y$ ?
7. Μεταβάλετε το πρόγραμμα `rw.c` ώστε ο χρήστης να παρέχει στο πρόγραμμα τον `seed` στην `drandom()` που επιθυμεί. Μπορείτε να κάνετε το `seed` να καθορίζεται από τη χρονική στιγμή που ξεκινάει το πρόγραμμα? (Υποδ.: Χρησιμοποιήστε την εντολή `seed = (long)time((time_t *) 0);`. Για καλύτερη αρχικοποίηση δείτε στο συνοδευτικό λογισμικό πως μπορείτε να χρησιμοποιήσετε τον αριθμό διαδικασίας (process ID) με τη συνάρτηση `getpid()`)
8. Για τον τυχαίο περιπατητή γνωρίζουμε ότι  $\langle \vec{R} \rangle = \vec{0}$ . Υπολογίστε τις μέσες τιμές  $\langle x \rangle$  και  $\langle y \rangle$  για  $N = 100, 100000$ . Είναι πραγματικά μηδέν? Γιατί? Πως εξαρτάται το αποτέλεσμά σας από τον αριθμό των μετρήσεων?
9. Υπολογίστε τη μέση τιμή του αριθμού των φορών που ο τυχαίος περιπατητής επιστρέφει στο αρχικό σημείο σαν συνάρτηση του  $N$ . Τι συμβαίνει για  $N \rightarrow \infty$  και γιατί?
10. Να αναπαράγετε τα Σχήματα 3.3, 3.3.1 και 3.3.1 καθώς και τον Πίνακα 3.3.1 (Συνοδευτικό λογισμικό).
11. Χρονομετρήστε το πρόγραμμα `rw2.c` για  $N = 32, 64, 128, 256, 512, 1024, 2048$  και  $L \approx (5 - 6)\sqrt{N}$ . Στη συνέχεια επαναχρονομετρήστε το αφαιρώντας διαδοχικά από το πρόγραμμα τις συναρτήσεις `measA1()`, `measA211()` και `meas1213()`. Μελετήστε γραφικά τη σχέση του χρόνου CPU με τα  $N$  και  $L$  αντίστοιχα (σε log-log plot). Προβλέψτε για πόσο περίπου θα τρέξει το πρόγραμμα για  $N = 4096, 8192, 16384$  και μετά επιβεβαιώστε την πρόβλεψή σας. Για πόσο περιμένετε να τρέξει το πρόγραμμά σας για  $N = 131072$ ?
12. Να αναπαράγετε τα Σχήματα 3.3.3, 3.3.3 και 3.3.3 καθώς και τον Πίνακα 3.3.1 (Συνοδευτικό λογισμικό).

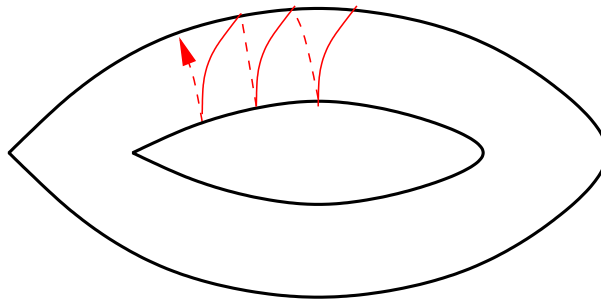
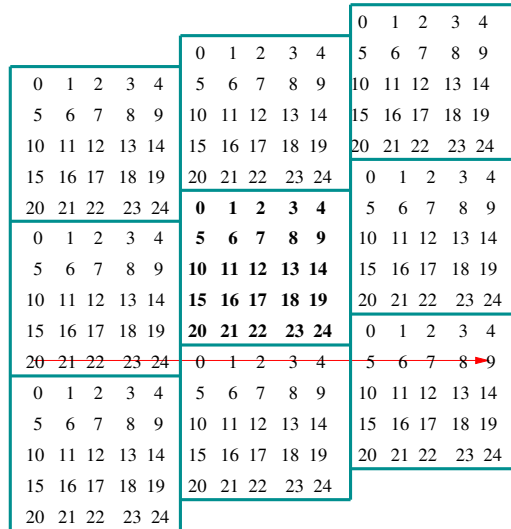
13. Να γράψετε κώδικα για τον SAW. Να υπολογίσετε τον εκθέτη  $\nu$  της σχέσης  $\langle R^2 \rangle \sim N^{2\nu}$  και να τον συγκρίνετε με την αναμενόμενη τιμή του.

0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
5	6	7	8	9	5	6	7	8	9	5	6	7	8	9
10	11	12	13	14	10	11	12	13	14	10	11	12	13	14
15	16	17	18	19	15	16	17	18	19	15	16	17	18	19
20	21	22	23	24	20	21	22	23	24	20	21	22	23	24
0	1	2	3	4	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	0	1	2	3	4
5	6	7	8	9	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	5	6	7	8	9
10	11	12	13	14	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	10	11	12	13	14
15	16	17	18	19	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	15	16	17	18	19
20	21	22	23	24	<del>20</del>	<del>21</del>	<del>22</del>	<del>23</del>	<del>24</del>	<del>20</del>	<del>21</del>	<del>22</del>	<del>23</del>	<del>24</del>
0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
5	6	7	8	9	5	6	7	8	9	5	6	7	8	9
10	11	12	13	14	10	11	12	13	14	10	11	12	13	14
15	16	17	18	19	15	16	17	18	19	15	16	17	18	19
20	21	22	23	24	20	21	22	23	24	20	21	22	23	24



Σχήμα 3.20: Οριζόντια κίνηση στο τετραγωνικό πλέγμα με περιοδικές συνοριακές συνθήκες και  $L = 5$ . Η τροχιά είναι κύκλος.





Σχήμα 3.21: Οριζόντια κίνηση στο τετραγωνικό πλέγμα με ελικοειδείς συνοριακές συνθήκες και  $L = 5$ . Η τροχιά είναι σπείρα.



## ΚΕΦΑΛΑΙΟ 4

# ΠΡΟΣΟΜΟΙΩΣΗ ΤΟΥ ΠΡΟΤΥΠΟΥ ISING

Στο Κεφάλαιο αυτό θα γίνει εισαγωγή των βασικών μεθόδων Μόντε Κάρλο για την προσομοίωση του πρότυπου Ising ορισμένου στο διδιάστατο τετραγωνικό πλέγμα. Αρχικά η μελέτη θα γίνει με τον αλγόριθμο Metropolis όπου θα μελετηθεί η διαδικασία που το σύστημα φέρνεται στην κατάσταση θερμικής ισορροπίας (thermalization) καθώς και η διαδικασία με την οποία παίρνουμε στατιστικά ανεξάρτητες μετρήσεις μελετώντας τους χρόνους αυτοσυσχετισμού (autocorrelation times) του συστήματος. Η κατανόηση των βασικών εννοιών είναι θεμελιώδης, αφού τα παραπάνω είναι οι κύριες τεχνικές δυσκολίες που αντιμετωπίζουμε όταν προσπαθούμε να μελετήσουμε συστήματα με μεγάλο αριθμό βαθμών ελευθερίας.

Η “καρδιά” των προγραμμάτων που θα χρησιμοποιήσουμε, είναι δανεισμένη από το βιβλίο των Newman και Barkema [1].

### 4.1 ΑΛΓΟΡΙΘΜΟΣ METROPOLIS

Το πρότυπο Ising ορίστηκε στην ενότητα 1.5. Ειδικά εδώ θεωρούμε την περίπτωση του τετραγωνικού πλέγματος με  $L$  πλεγματικές θέσεις σε κάθε πλευρά, έτσι ώστε  $N = L \times L = L^2$  είναι ο συνολικός αριθμός πλεγματικών θέσεων και  $N_l = 2N$  ο συνολικός αριθμός των “δεσμών” μεταξύ πλησιεστέρων γειτόνων. Η τελευταία σχέση ισχύει επειδή επιλέγουμε ελικοειδείς συνοριακές συνθήκες στο πλέγμα (βλ. Ενότητα 3.3.2, Σχήμα 3.3.2). Σε κάθε πλεγματική θέση  $i$  έχουμε ένα βαθμό ελευθερίας, το “σπιν”  $s_i$  το οποίο παίρνει δύο τιμές, έστω  $\pm 1$ . Θεωρούμε το σύστημα για την περίπτωση μηδενικού μαγνητικού πεδίου  $B = 0$  οπότε η Hamiltonian

του συστήματος θα δίνεται από τη σχέση<sup>1</sup>

$$H = - \sum_{\langle ij \rangle} s_i s_j . \quad (4.1)$$

Θυμίζουμε πως  $\sum_{\langle ij \rangle}$  είναι άθροισμα πάνω στους  $\langle i, j \rangle$ , δηλ. πάνω στα ζεύγη γειτονικών πλεγματικών θέσεων.  $\sum_{\langle ij \rangle} = (1/2) \sum_{i=1}^N \sum_{j=1}^N$  αφού στη δεύτερη περίπτωση κάθε δεσμός αθροίζεται δύο φορές, μια από κάθε πλεγματική θέση που βρίσκεται στα άκρα του. Η συνάρτηση επιμερισμού είναι

$$Z = \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} \dots \sum_{s_N=\pm 1} e^{-\beta H[\{s_i\}]} \equiv \sum_{\{s_i\}} e^{\beta \sum_{\langle ij \rangle} s_i s_j} . \quad (4.2)$$

Στόχος μας είναι η δημιουργία ενός δείγματος του συστήματος που κατανέμεται σύμφωνα με την κατανομή Boltzman (1.6) και θα το δημιουργήσουμε φτιάχνοντας μια αλυσίδα Markov σύμφωνα με αυτά που ειπώθηκαν στην παράγραφο 2.2. Η δειγματοληψία θα γίνει σύμφωνα με την (2.6) και οι μέσες τιμές θα υπολογίζονται από το δείγμα από την (2.7). Επιλέγοντας σε κάθε βήμα της διαδικασίας την επόμενη κατάσταση του συστήματος σύμφωνα με την (2.13), το δείγμα μας θα βρίσκεται προσεγγιστικά στην επιθυμητή κατανομή για αρκετά μεγάλο πλήθος στοιχείων ή ισοδύναμα αριθμό βημάτων ("χρόνο") στην διαδικασία Markov.

Έστω ότι το σύστημά μας βρίσκεται αρχικά στην κατάσταση  $\mu^2$ . Η πιθανότητα να βρεθεί στο επόμενο βήμα της διαδικασίας στην κατάσταση  $\nu$ , σύμφωνα με την (2.19) είναι

$$P(\mu \rightarrow \nu) = g(\mu \rightarrow \nu) A(\mu \rightarrow \nu) , \quad (4.3)$$

όπου  $g(\mu \rightarrow \nu)$  είναι η **πιθανότητα επιλογής** της κατάστασης  $\nu$  όταν το σύστημα βρίσκεται στην κατάσταση  $\mu$  και  $A(\mu \rightarrow \nu)$  ο **λόγος αποδοχής**, δηλ. η πιθανότητα τότε το σύστημα να μεταβεί στη νέα κατάσταση. Αν ικανοποιείται η συνθήκη λεπτομερούς ισοζύγησης (2.13)

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{g(\mu \rightarrow \nu) A(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu) A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)} \quad (4.4)$$

τότε η κατανομή στο δείγμα μας θα συγκλίνει στην (1.6)  $p_\mu = e^{-\beta E_\mu} / Z$ . Για να είναι οι πιθανότητες  $P(\mu \rightarrow \nu)$  αρκετά μεγάλες ώστε το σύστημα να αλλάζει συχνά κατάσταση (τουλάχιστο >50%)

<sup>1</sup>Η σταθερά  $J = 1$  με κατάλληλη επιλογή μονάδων για τα  $s_i$ .

<sup>2</sup>Η κατάσταση  $\mu$  εδώ ορίζεται από μία συγκεκριμένη διάταξη των  $N$  σπινς  $\{s_i\}_{i=1 \dots N}$

οι μεταβολές στην ενέργεια  $E_\nu - E_\mu$  δεν πρέπει να είναι πολύ μεγάλες. Θα πρέπει το γινόμενο με τη θερμοκρασία να είναι αριθμός της τάξης της μονάδας. Ένας τρόπος να το πετύχουμε αυτό είναι να εφαρμόσουμε έναν αλγόριθμο που σε κάθε βήμα θα μεταβάλλει την τιμή ενός από τα σπιν του πλέγματος  $s_i = \pm 1 \rightarrow s'_i \mp 1$ . Επειδή η ενέργεια (4.1) είναι τοπική ποσότητα, η μεταβολή θα είναι αριθμός σχετικά μικρός. Πιο συγκεκριμένα, αν κάθε πλεγματική θέση έχει  $z = 4$  πλησιέστερους γείτονες, με μια τέτοια μεταβολή το πρόσημο του όρου  $s_i s_j$  αλλάζει για  $z$  όρους της (4.1), αυτούς που έχουν στο ένα άκρο τους τη θέση  $i$ . Η μεταβολή της ενέργειας είναι  $\pm 2$  για κάθε δεσμό. Αν η κατάσταση  $\mu$  ορίζεται να είναι η  $\{s_1, \dots, s_i, \dots, s_N\}$  και η κατάσταση  $\nu$  είναι η  $\{s_1, \dots, s'_i, \dots, s_N\}$  (δηλ. με όλα τα σπιν ίδια, εκτός από το  $s_i$  το οποίο αλλάζει πρόσημο), τότε η μεταβολή της ενέργειας θα είναι

$$|\Delta E| \leq 2z \Leftrightarrow E_\mu - 2z \leq E_\nu \leq E_\mu + 2z. \quad (4.5)$$

Αν επιλέξουμε τη θέση  $i$  τυχαία τότε

$$g(\mu \rightarrow \nu) = g(\nu \rightarrow \mu) = \begin{cases} \frac{1}{N} & \text{για } (\mu, \nu) \text{ διαφέρουν ως προς ένα σπιν flip} \\ 0 & \text{αλλιώς} \end{cases}, \quad (4.6)$$

ο αλγόριθμος είναι εργοδικός. Θα πρέπει τότε να ισχύει

$$\frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)}. \quad (4.7)$$

Ένας απλός τρόπος να ικανοποιήσουμε τη συνθήκη αυτή είναι να πάρουμε τη σχέση (2.14)

$$A(\mu \rightarrow \nu) = A_0 \cdot e^{-\frac{1}{2}\beta(E_\nu - E_\mu)}. \quad (4.8)$$

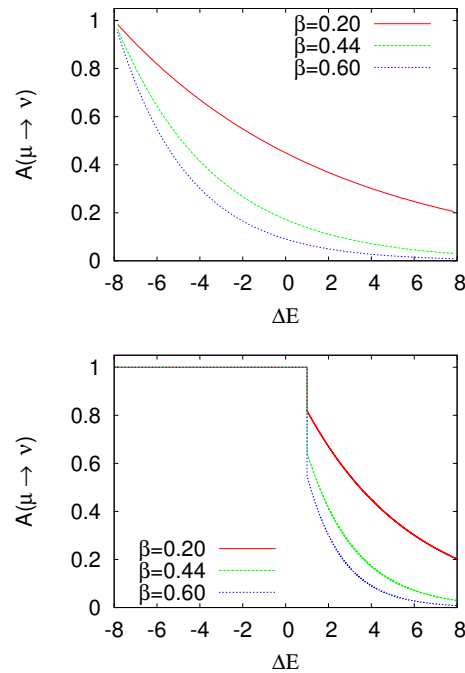
Για να μεγιστοποιήσουμε τους λόγους αποδοχής, αφού θα πρέπει να ισχύει  $A(\mu \rightarrow \nu) \leq 1$ ,  $|\Delta E| \leq 2z$ , παίρνουμε  $A_0 = e^{-\beta z}$ . Άρα

$$A(\mu \rightarrow \nu) = e^{-\frac{1}{2}\beta(E_\nu - E_\mu + 2z)}. \quad (4.9)$$

Στο Σχήμα 4.1 φαίνεται η εξάρτηση του  $A(\mu \rightarrow \nu)$  από τη μεταβολή της ενέργειας για διάφορα  $\beta$ . Παρατηρούμε ότι η πιθανότητα αυτή γίνεται μικρή, ακόμα και για μηδενική μεταβολή της ενέργειας, με αποτέλεσμα η αποδοτικότητα της επιλογής αυτής να είναι μικρή.

Είναι πολύ αποδοτικότερο να χρησιμοποιήσουμε τον αλγόριθμο των Nicolas Metropolis *et. al.* 1953 [19] ο οποίος δίνεται από την (2.16)

$$A(\mu \rightarrow \nu) = \begin{cases} e^{-\beta(E_\nu - E_\mu)} & E_\nu - E_\mu > 0 \\ 1 & E_\nu - E_\mu \leq 0 \end{cases}. \quad (4.10)$$



Σχήμα 4.1: Λόγοι αποδοχής  $A(\mu \rightarrow \nu)$  για το διδιάστατο πρότυπο Ising στο τετραγωνικό πλέγμα για τη σχέση (4.9) (αριστερά) και για τον αλγόριθμο Metropolis (δεξιά) σαν συνάρτηση της μεταβολής της ενέργειας  $\Delta E = E_\nu - E_\mu$ . Φαίνεται η υπεροχή του τελευταίου, αφού οι λόγοι αποδοχής είναι πολύ μεγαλύτεροι.

Σύμφωνα με την παραπάνω σχέση, το σύστημα πάντα δέχεται την αλλαγή μιας κατάστασης όταν αυτή ελαττώνει την ενέργεια, ενώ όταν η αλλαγή οδηγεί σε αύξηση της ενέργειας αυτή γίνεται δεκτή με πιθανότητα μικρότερη της μονάδας. Όπως φαίνεται και από το Σχήμα 4.1, το σύστημα δέχεται αλλαγές καταστάσεων πολύ συχνότερα από την προηγούμενη επιλογή μας. Ο αλγόριθμος αυτός έχει πολύ γενικότερη εφαρμογή από την περίπτωση που μελετάμε και έχει ευρεία χρήση σε ένα μεγάλο φάσμα προβλημάτων λόγω της γενικότητας, απλότητας και αποδοτικότητάς του. Να σημειώσουμε πως η επιλογή μας να αλλάζουμε την κατάσταση  $\mu \rightarrow \nu$  αλλάζοντας την τιμή ενός μόνο σπιν δεν είναι περιορισμός της μεθόδου αυτής και θα μπορούσε να εφαρμοστεί για οποιαδήποτε αλλαγή καταστάσεων.

## 4.2 ΣΧΕΔΙΑΣΜΟΣ ΚΩΔΙΚΑ

Για το σχεδιασμό του κώδικα, είναι αρχικά σημαντικό να σχεδιάσουμε τη δομή των δεδομένων. Οι βαθμοί ελευθερίας μας είναι τα σπιν  $s_i = \pm 1$  ορισμένα στις  $N$  πλεγματικές θέσεις. Επιλέγοντας ελικοειδείς συνοριακές συνθήκες για τετραγωνικό πλέγμα πλευράς  $L$  και  $N=L*L$  πλεγματικών θέσεων, σύμφωνα με όσα ειπώθηκαν στην ενότητα 3.3.2 και φαίνονται στα Σχήματα 3.3.2, 3.4, η θέση πάνω στο πλέγμα μπορεί να προσδιοριστεί από έναν ακέραιο  $r=0, \dots, N-1$  ενώ οι πλησιέστεροι γείτονες από τους ακέραιους  $nn$  από

```
if((nn=r+1)>= N) nn -= N //+X direction
if((nn=r-1)< 0) nn += N //-X direction
if((nn=r+L)>= N) nn -= N //+Y direction
if((nn=r-L)< 0) nn += N //-Y direction
```

Οι διατάξεις των σπιν μπορούν να αποθηκευτούν σε ακέραιο array `int s[N]` με επιτρεπτές τιμές  $\pm 1$ .

Η επιλογή των ελικοειδών (ή τοροειδών/περιοδικών) συνοριακών συνθηκών είναι σημαντική για τον σημαντικό περιορισμό των φαινομένων πεπερασμένου μεγέθους του συστήματός μας. Όλες οι πλεγματικές θέσεις έχουν τον ίδιο αριθμό πλησιεστέρων γειτόνων και την ίδια τοπική γεωμετρία. Το σύστημα είναι συμμετρικό ως προς τις μετατοπίσεις και όλα τα σπιν ισοδύναμα. Αυτό δε θα συνέβαινε αν λ.χ. επιλέγαμε το τετράγωνο να έχει σύνορο όπου τα σπιν θα έπαιρναν λ.χ. καθορισμένες ή ελεύθερες τιμές.

Η δυναμική της Μόντε Κάρλο εξέλιξης καθορίζεται από την αρχική κατάσταση του συστήματος και τον αλγόριθμο του Metropolis. Κατάλληλη επιλογή αρχικών συνθηκών μπορεί να είναι εξαιρετικά σημαντική για προσομοιώσεις συστημάτων με τη μέθοδο Μόντε Κάρλο. Από αυτή θα εξαρτηθεί το πόσο γρήγορα, ακόμα και το αν, το σύστημα θα βρεθεί σε κατάσταση θερμικής ισορροπίας. Στην περίπτωση μας αυτό δε θα είναι ιδιαίτερα σημαντικό πρόβλημα αλλά θα το μελετήσουμε λόγω της σημασίας του σε άλλα προβλήματα. Εδώ η αρχική κατάσταση μπορεί να επιλεγεί να είναι “παγωμένη” ( $\beta = \infty$  - όλα τα σπιν να έχουν την ίδια τιμή) ή “θερμή” ( $\beta = 0$  - τα σπιν να έχουν τιμή  $\pm 1$  με ίση πιθανότητα  $1/2$ ). Για μεγάλα πλέγματα όπου η εύρεση της κατάστασης θερμικής ισορροπίας μπορεί να απαιτεί αρκετά βήματα, επιλέγουμε τη σταδιακή μεταβολή της θερμοκρασίας από  $\beta = 0$  ή  $\beta = \infty$  στις επιθυμητές τιμές. Σε κάθε βήμα μεταβολής της θερμοκρασίας χρησιμοποιούμε τη διάταξη σπιν από την προηγούμενη προσομοίωση έτσι ώστε να βρισκόμαστε γρήγορα στη νέα κατάσταση θερμικής ισορροπίας.

Ξεκινώντας από διαφορετικές αρχικές συνθήκες και συγκρίνοντας τα αποτελέσματά μας ελέγχουμε την ορθότητα τους, δηλ. το αν το σύστημά μας έχει πετύχει τη θερμική ισορροπία και αν η μέθοδός μας πάσχει από προβλήματα εργοδικότητας.

Εξετάζουμε τώρα τα βήματα της διαδικασίας Markov που παράγεται από τον αλγόριθμο Metropolis. Έστω ότι το σύστημά μας είναι στην κατάσταση  $\mu = \{s_1^\mu, \dots, s_k^\mu, \dots, s_N^\mu\}$  και εξετάζουμε τη μετάβαση στη νέα κατάσταση  $\nu = \{s_1^\nu, \dots, s_k^\nu, \dots, s_N^\nu\}$  που προκύπτει από την αλλαγή της τιμής του σπιν  $s_k^\nu = -s_k^\mu$  (spin flip) ενώ τα υπόλοιπα σπιν παραμένουν τα ίδια  $s_j^\nu = s_j^\mu \forall j \neq k$ .

Η μεταβολή της ενέργειας θα είναι

$$E_\nu - E_\mu = \left(-\sum_{\langle ij \rangle} s_i^\nu s_j^\nu\right) - \left(-\sum_{\langle ij \rangle} s_i^\mu s_j^\mu\right) \quad (4.11)$$

$$= -\sum_{\langle ik \rangle} s_i^\mu (s_k^\nu - s_k^\mu) \quad (4.12)$$

$$= 2\sum_{\langle ik \rangle} s_i^\mu s_k^\mu \quad (4.13)$$

$$= 2s_k^\mu \left(\sum_{\langle ik \rangle} s_i^\mu\right), \quad (4.14)$$

όπου στη δεύτερη σειρά απαλείψαμε τους κοινούς όρους των δύο αθροισμάτων και στην τρίτη χρησιμοποιήσαμε τη σχέση  $s_k^\nu - s_k^\mu = -2s_k^\mu$  που μπορείτε εύκολα να αποδείξετε εξετάζοντας τις δύο περιπτώσεις  $s_k^\mu = \pm 1$ .

Ο αλγόριθμος Metropolis εφαρμόζεται τώρα εύκολα. Υπολογίζοντας απλά το άθροισμα στην παρένθεση της τελευταίας σειράς της (4.11) παίρνουμε τη μεταβολή της ενέργειας  $E_\nu - E_\mu$ . Αν  $E_\nu - E_\mu \leq 0$  δεχόμαστε τη κατάσταση  $\nu$  να είναι η νέα κατάσταση του συστήματος. Αν  $E_\nu - E_\mu > 0$  τότε  $A(\mu \rightarrow \nu) = e^{-\beta(E_\nu - E_\mu)} < 1$ . Επιλέγουμε έναν ομοιόμορφα κατανομημένο τυχαίο αριθμό  $0 \leq x < 1$ . Η πιθανότητα να έχουμε  $x < A(\mu \rightarrow \nu)$  είναι  $A(\mu \rightarrow \nu)$ . Άρα αν  $x < A(\mu \rightarrow \nu)$  δεχόμαστε τη μεταβολή. Αν  $A(\mu \rightarrow \nu) < x$  η μεταβολή απορρίπτεται και παραμένουμε στην κατάσταση  $\mu$ .

Μια μικρή τεχνική παρατήρηση στο σχεδιασμό του προγράμματος είναι ότι οι δυνατές τιμές του αθροίσματος  $\left(\sum_{\langle ik \rangle} s_i^\mu\right) = -4, -2, 0, 2, 4$ . Οι τιμές που υπεισέρχονται στον υπολογισμό της  $A(\mu \rightarrow \nu)$  είναι μόνο αυτές που αυξάνουν την ενέργεια, δηλ. οι δύο τιμές 2, 4. Οι πέντε αυτές τιμές για τα  $A(\mu \rightarrow \nu)$  μπορούν να αποθηκευτούν στη μνήμη του υπολογιστή και να αποφύγουμε την ακριβή σε χρόνο επανάληψη υπολογισμού των εκθετικών  $e^{-\beta(E_\nu - E_\mu)}$ .



Τέλος το πρόγραμμά μας θα πρέπει να υπολογίζει τις φυσικές ποσότητες που θα μετρήσουμε κατά τη μελέτη του συστήματός μας. Αυτές να είναι η ενέργεια (4.1)

$$= - \sum_{\langle ij \rangle} s_i s_j, \quad (4.15)$$

και η μαγνήτιση

$$M = \left| \sum_i s_i \right|. \quad (4.16)$$

Η απόλυτη τιμή στην τελευταία σχέση μπαίνει λόγω της συμμετρίας ανάκλασης των σπιν του συστήματος (ομάδα  $Z_2$ ). Επειδή είναι ισοπίθανο να έχουμε δύο διατάξεις με όλα τα σπιν τους να έχουν αντίθετες τις τιμές τους σε κάθε πλεγματική θέση, η μέση τιμή  $\langle \sum_i s_i \rangle$  είναι πάντα 0. Για αρκετά μεγάλο αριθμό μετρήσεων όσες φορές το σύστημά μας θα είναι σε διάταξη με  $\sum_i s_i = M_1$  θα είναι και σε διάταξη με  $\sum_i s_i = -M_1$ . Στην πράξη, όπως θα δούμε, η μετάβαση από την κατάσταση με μαγνήτιση  $M_1$  στην κατάσταση με μαγνήτιση  $-M_1$  μπορεί να χρειάζεται μεγάλο αριθμό βημάτων Μόντε Κάρλο και αυτό να μην παρατηρείται στην προσομοίωσή μας. Στο όριο  $N \rightarrow \infty$  (θερμοδυναμικό όριο) το σύστημα επιλέγει τη μία από τις δύο καταστάσεις (αυθόρμητο σπάσιμο συμμετρίας).

Οι μετρήσεις της ενέργειας και της μαγνήτισης μπορούν να γίνονται με δύο τρόπους. Στον πρώτο ενημερώνουμε την τιμή της ενέργειας και της μαγνήτισης σε κάθε βήμα Μόντε Κάρλο. Αυτό γίνεται εύκολα γιατί η μεταβολή των τιμών των αθροισμάτων στις (4.15) και (4.16) εξαρτάται μόνο από την τιμή του σπιν  $s_k^\mu$  που αλλάζει και των τιμών του σπιν των πλησιεστέρων γειτόνων. Τη μεταβολή της ενέργειας την υπολογίζουμε έτσι κι αλλιώς από τη σχέση (4.11) ενώ η μεταβολή του αθροίσματος στην (4.16) δίνεται απλά από την

$$\sum_i s_i^\nu - \sum_i s_i^\mu = s_k^\nu - s_k^\mu = -2s_k^\mu \quad (4.17)$$

Ο δεύτερος τρόπος είναι να υπολογίζουμε τα αθροίσματα (4.15) και (4.16) κάθε φορά που θέλουμε να πάρουμε μια μέτρηση. Ποια μέθοδος είναι πιο συμφέρουσα εξαρτάται από το πόσο συχνά παίρνουμε μια στατιστικά ανεξάρτητη μέτρηση (χρόνο αυτοσυσχετισμού, δεξ παρακάτω) και από το λόγο αποδοχής των βημάτων Μόντε Κάρλο. Λ.χ. για τον υπολογισμό της μαγνήτισης, αν ο μέσος λόγος αποδοχής είναι  $\bar{A}$ , τότε με τον πρώτο τρόπο έχω  $\bar{A}N$  προσθέσεις ανά Μόντε Κάρλο βήματα (1 sweep) ενώ με τον δεύτερο έχω  $N$  προσθέσεις ανά μέτρηση.

Είναι συνηθισμένο να κανονικοποιούμε την ενέργεια ανά δεσμό

$$\langle e \rangle = \frac{1}{N_l} \langle E \rangle = \frac{1}{2N} \langle E \rangle, \quad (4.18)$$

η οποία παίρνει τιμή από -1 (θεμελιώδης κατάσταση, όλοι οι  $2N$  δεσμοί έχουν τιμή -1) μέχρι +1, ενώ τη μαγνήτιση ανά πλεγματική θέση

$$\langle m \rangle = \frac{1}{N} \langle M \rangle, \quad (4.19)$$

η οποία παίρνει τιμή από 0 (πλήρης αταξία -  $\beta = 0$ ) έως 1 (θεμελιώδης κατάσταση -  $\beta = \infty$ ). Για το λόγο αυτό, η μαγνήτιση είναι μια παράμετρος τάξης (order parameter), δηλ. μια φυσική ποσότητα της οποίας η μέση τιμή μας υποδεικνύει τότε το σύστημα βρίσκεται σε φάση τάξης (ordered phase) ή σε φάση αταξίας (disordered phase).

Από τις ποσότητες αυτές υπολογίζουμε τις διακυμάνσεις της ενέργειας από την ειδική θερμότητα

$$c = \beta^2 N \langle (e - \langle e \rangle)^2 \rangle = \beta^2 N (\langle e^2 \rangle - \langle e \rangle^2), \quad (4.20)$$

και της μαγνήτισης από τη μαγνητική επιδεκτικότητα

$$\chi = \beta N \langle (m - \langle m \rangle)^2 \rangle = \beta N (\langle m^2 \rangle - \langle m \rangle^2). \quad (4.21)$$

Για να εκτιμήσουμε πόσα δεδομένα πρέπει να συλλέξουμε για μια καλή μέτρηση των παραπάνω ποσοτήτων, ανατρέχουμε στο αποτέλεσμα της στατιστικής που μας λέει ότι αν έχουμε  $n$  ανεξάρτητες μετρήσεις, τότε το στατιστικό σφάλμα ελαττώνεται σαν  $\sim 1/\sqrt{n}$ . Το πρόβλημα είναι τότε έχουμε ανεξάρτητες μετρήσεις και θα το αντιμετωπίσουμε παρακάτω.

### 4.2.1 Ο ΚΩΔΙΚΑΣ

Στην παράγραφο αυτή θα παρουσιάσουμε και αναλύσουμε τον κώδικα που θα χρησιμοποιήσουμε στην προσομοίωση του πρώτου Ising. Αυτόν το παίρνουμε από το βιβλίο των Newman και Barkema [1]. Θα ακολουθήσουμε τη φιλοσοφία του modular programming όπου θα τοποθετήσουμε σε διαφορετικά αρχεία ξεχωριστές ενότητες του κώδικα. Αυτό κάνει ευκολότερη την ανάπτυξη, έλεγχο και συντήρηση του κώδικα από έναν ή και περισσότερους προγραμματιστές.

Αρχικά σε ένα αρχείο-επικεφαλίδα (header file) τοποθετούμε τους ορισμούς που είναι κοινοί για ένα ή περισσότερα αρχεία. Το αρχείο αυτό, εδώ το ονομάζουμε `include.h`, θα συμπεριλαμβάνεται σε κάθε αρχείο που θα το χρησιμοποιεί με μια εντολή του preprocessor:

```
#include "include.h"
```

Τα περιεχόμενα του αρχείου είναι:

```

/***** include.h *****/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define L 12
#define N (L*L)
#define XNN 1
#define YNN L

int    s[N];
double beta;
double prob[5];

int    E(), M();
double drandom();
long seed;
void init(int ), met(), measure();

```

Χρησιμοποιούμε τον `preprocessor` για να ορίσουμε συμβολικά το μέγεθος του συστήματος  $L$  (εδώ είναι 12) και το  $N$  το οποίο είναι το  $L \cdot L^3$ . Το array  $s[N]$  είναι η βασική θέση στη μνήμη που χρησιμοποιούμε για την τρέχουσα διάταξη των σπιν του συστήματος. Θυμίζουμε πως οι επιτρεπτές τιμές του δείκτη του array είναι  $s[0]$ ,  $s[1]$ , ...,  $s[N-1]$ .

Η μεταβλητή  $\beta$  είναι η θερμοκρασία του συστήματος και στην  $\text{prob}[5]$  τοποθετούμε τις δυνατές τιμές των πιθανοτήτων  $A(\mu \rightarrow \nu)$  όπως εξηγήσαμε στη σελίδα 94. Οι υπόλοιπες δηλώσεις αφορούν τις συναρτήσεις  $E()$ ,  $M()$  που υπολογίζουν την ενέργεια και τη μαγνήτιση του συστήματος, την  $\text{init}(\text{int } \text{start})$  που αρχικοποιεί την προσομοίωση, την  $\text{met}()$  που κάνει ένα Μόντε Κάρλο sweep ( $N$  βήματα Metropolis) καθώς και τη  $\text{measure}()$  που κάνει τις μετρήσεις της ενέργειας και της μαγνήτισης.

Η  $\text{drandom}()$  είναι αυτή που θα χρησιμοποιήσουμε για γεννήτρια ψευδοτυχαίων αριθμών και ο σπόρος  $\text{seed}$  είναι ορατός από όλο το πρόγραμμα, έτσι ώστε να μπορούμε να τον θέσουμε από διαφορετικές συναρτήσεις όποτε το επιθυμούμε.

Τέλος οι παράμετροι  $XNN$  και  $YNN$  είναι αυτές που μας οδηγούν στους πλησιέστερους γείτονες μιας πλεγματικής θέσης σύμφωνα

---

<sup>3</sup> Παρατηρήστε τις παρενθέσεις. Γιατί μπαίνουν?

με όσα είπαμε στην αρχή της ενότητας 4.2. Για παράδειγμα  $i+xNN$  είναι ο πλησιέστερος γείτονας της (εσωτερικής) θέσης  $i$  κατά την  $+x$  διεύθυνση και  $i-yNN$  είναι ο πλησιέστερος γείτονας κατά την  $-y$  διεύθυνση.

Το πρόγραμμά μας οδηγείται από την συνάρτηση `main()`:

```

/***** main.c *****/
#include "include.h"

int main(int argc, char **argv){

    int start=0;
    int isweep,nsweep = 100;

    beta=0.21;seed=9873;start=1;
    init(start);
    for(isweep=0;isweep<nsweep;isweep++){
        met();
        measure();
    }
}

```

Από εδώ καταλαβαίνουμε τη δομή της προσομοίωσης: Αφού καθορίσουμε τις βασικές παραμέτρους της προσομοίωσης (`start=0` αρχίζει το σύστημα από μία ψυχρή διάταξη ενώ `start=1` από θερμή, η θερμοκρασία τίθεται στην (θερμή) τιμή `beta=0.21`, ο σπόρος της `drandom()` `seed=9873` και των αριθμό των sweeps να είναι `nsweep=100`) καλούμε την `init(start)` να αρχικοποιήσει την προσομοίωση και μετά για `nsweeps` φορές καλούμε την `met()` να κάνει ένα sweep και την `measure()` να κάνει μια μέτρηση.

Προχωρώντας ένα ακόμα επίπεδο παρακάτω στη δομή του προγράμματος εξετάζουμε τη συνάρτηση `init(start)`:

```

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

/* start= 0  cold
   1  hot */
void init(int start){
    int i;
    /*Initialize probabilities*/
    for(i=2;i<5;i+=2) prob[i] = exp(-2*beta*i);
    /*Initial configuration*/
    switch(start){
        case 0:/*cold start*/

```

```

    for(i=0;i<N;i++) s[i]=1;
    break;
case 1:/*hot start*/
    for(i=0;i<N;i++){
        if(drandom() < 0.5)
            s[i] = 1;
        else
            s[i] = -1;
    }
    break;
default:
    printf("start= %d not valid. Exiting...\n",start);
    exit(1);
    break;
}
}

```

Η δομή της συνάρτησης είναι πολύ απλή. Αρχικοποιεί το array prob[5] με τις τιμές των πιθανοτήτων  $A(\mu \rightarrow \nu) = e^{-\beta(E_\nu - E_\mu)} = e^{-2\beta s_k^\mu (\sum_{\langle ik \rangle} s_i^\mu)}$ . Οι πιθανότητες αυτές μας ενδιαφέρουν όταν  $s_k^\mu (\sum_{\langle ik \rangle} s_i^\mu) > 0$  και οι δυνατές τιμές που μπορεί τότε να πάρει είναι 2 και 4. Αυτές είναι και οι τιμές που αποθηκεύουμε και θυμόμαστε ότι ο δείκτης του array είναι η έκφραση  $s_k^\mu (\sum_{\langle ik \rangle} s_i^\mu)$  όταν αυτή είναι θετική.

Στη συνέχεια καθορίζουμε την αρχική διάταξη των σπιν ανάλογα με την τιμή που παίρνει ο ακέραιος start. Το switch block μας επιτρέπει να προσθέσουμε στο μέλλον και άλλες δυνατότητες αν τυχόν αναβαθμίσουμε τον κώδικα. Αν start=0 οι τιμές όλων των σπιν τίθενται ίσες με 1, ενώ αν start=1 θέτουμε τις τιμές των σπιν με ίση πιθανότητα  $\pm 1$ . Η πιθανότητα  $\text{drandom}() < 0.5$  είναι  $1/2$  οπότε στην περίπτωση αυτή  $s[i]=1$  αλλιώς (πιθανότητα  $1 - 1/2 = 1/2$ )  $s[i]=-1$ .

Η καρδιά του προγράμματος είναι η συνάρτηση met(). Αυτή εκτελεί N Metropolis βήματα. Επισκέπτεται N φορές μία τυχαία πλεγματική θέση και θέτει το ερώτημα αν θα μεταβάλλει την τιμή του σπιν (spin flip). Αυτό γίνεται σύμφωνα με τον αλγόριθμο Metropolis υπολογίζοντας τη μεταβολή στην ενέργεια πριν και μετά την αλλαγή του σπιν σύμφωνα με τη σχέση (4.11):

```

/***** 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 random site i
  /*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/2*/
  delta = sum*s[i];
  /*flip*/
  if(delta <=0 )           {s[i] = -s[i];} //accept
  else if(drandom(<prob[delta]){s[i] = -s[i];} //accept
} //sweep ends
}

```

### Η γραμμή

```
i=N*drandom();//pick a random site i
```

επιλέγει ισοπίθανα μια πλεγματική θέση  $i=0, \dots, N-1$ . Το δεξί μέλος αφού υπολογιστεί ως `double` (γινόμενο `int` επί `double`) μετατρέπεται πάλι σε `int` κόβοντας το κομμάτι μετά την υποδιαστολή. Εδώ είναι σημαντικό να μην παρουσιαστεί ποτέ η τιμή  $i=N$  που μπορεί να γίνει αν `drandom()`=1.0. Θα μπορούσαμε εδώ να προσθέσουμε μια γραμμή ελέγχου αλλά εμπιστευόμαστε τον προγραμματιστή της `drandom()`.

Στη συνέχεια υπολογίζουμε το άθροισμα  $(\sum_{\langle ik \rangle} s_i^\mu)$  της (4.11). Για το λόγο αυτό πρέπει να βρούμε τους πλησιέστερους γείτονες της θέσης  $i$  και αυτό γίνεται με τις γραμμές

```

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];

```

### Η εντολή λ.χ.

```
if((nn=i+XNN)>= N) nn -= N;
```

υπολογίζει τον γείτονα στην  $+x$  διεύθυνση. Το `if(...)` μπαίνει για την περίπτωση που ξεπεράσουμε τα όρια των τιμών  $0, \dots, N-1$ , οπότε επαναφέρουμε την τιμή αφαιρώντας  $N$ . Εδώ βλέπουμε την

απλότητα των ελικοειδών συνοριακών συνθηκών, και είναι ακριβώς ο λόγος που τις επιλέξαμε. Αφού βρεθεί ο γείτονας, προσθέτουμε την τιμή του σπιν του στην ακέραια μεταβλητή *sum*. Επαναλαμβάνουμε τη διαδικασία για όλους τους 4 πλησιέστερους γείτονες.

Στη συνέχεια στην μεταβλητή *delta* θέτουμε το γινόμενο (4.11)  $s_k^\mu \left( \sum_{\langle ik \rangle} s_i^\mu \right)$ . Αν αυτό είναι αρνητικό, τότε η μεταβολή της ενέργειας είναι αρνητική και η αλλαγή του σπιν γίνεται δεκτή. Αν όχι, τότε εφαρμόζουμε το κριτήριο (4.10) χρησιμοποιώντας την τιμή του *array prob[delta]* σύμφωνα με τον τρόπο που το κατασκευάσαμε στην *init(start)*. Η πιθανότητα του ομοιόμορφα κατανεμημένου στο διάστημα  $[0, 1)$  αριθμού *drandom()* < *prob[delta]* είναι *prob[delta]* και στην περίπτωση αυτή η αλλαγή γίνεται δεκτή. Σε κάθε άλλη περίπτωση η αλλαγή δε γίνεται δεκτή και το σπιν *s[i]* διατηρεί την παλιά του τιμή.

Αφού η διαδικασία επαναληφθεί *N* φορές, η συνάρτηση επιστρέφει στη *main()*.

Μετά από κάθε Metropolis sweep έχουμε επιλέξει να κάνουμε μια μέτρηση της ενέργειας και της μαγνήτισης. Εδώ επιλέγουμε μινιμαλιστική προσέγγιση του κεντρικού κώδικα και υπολογίζουμε μόνο τα αναγκαία. Αφήνουμε λ.χ. στη διαδικασία της ανάλυσης τον υπολογισμό των μέσων τιμών, στατιστικών σφαλμάτων, της ειδικής θερμότητας και μαγνητικής επιδεκτικότητας. Δίνουμε στην ανάλυση τη μέγιστη δυνατή πληροφορία που θέλουμε να παρέχουμε και αφήνουμε στη φάση εκείνη να αποφασιστεί αν και πώς θα υπολογιστεί μια φυσική ποσότητα. Αυτό (αν φυσικά δεν έχουμε πρόβλημα αποθηκευτικού χώρου) είναι επιβεβλημένο αν η παραγωγή των δεδομένων είναι "ακριβή" και η ανάλυση "φτηνή". Ένας άλλος λόγος που παρέχουμε στην ανάλυση την εκάστοτε τιμή της ενέργειας και μαγνήτισης είναι ώστε να γίνει δυνατή η παρακολούθηση της προσομοίωσης και να εντοπιστούν προβλήματα όπως το αν το σύστημα έφτασε σε θερμική ισορροπία, παρουσιάζονται μεγάλοι αυτοσυσχετισμοί ή ακόμα και ο εντοπισμός σφαλμάτων στο πρόγραμμα. Η συνάρτηση των μετρήσεων υπολογίζει κάθε φορά που καλείται τη συνολική ενέργεια και μαγνήτιση (χωρίς την απόλυτη τιμή) καλώντας τις ξεχωριστές συναρτήσεις *E()* και *M()*. Αυτές απλά εφαρμόζουν τις σχέσεις (4.15) και (4.16) (χωρίς την απόλυτη τιμή).

```

/***** 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;
}

```

Τέλος η γνωστή μας `drandom()` παρατίθεται για πληρότητα, όπως προσαρμόζεται για τις ανάγκες του προγράμματος.

```

/***** 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;

    return conv*(seed-1);
}

```

Η μεταγλώττιση του κώδικα γίνεται με την εντολή

```
% cc -O2 main.c init.c measure.c met.c drandom.c -o is -lm
```



η οποία παράγει το εκτελέσιμο αρχείο `is` το οποίο μπορούμε τώρα να τρέξουμε:

```
% ./is > out.dat
% less out.dat
-52 10
-48 40
-64 44
-92 64
.....
```

Η έξοδος του προγράμματος είναι σε δύο στήλες, στην πρώτη τυπώνεται η συνολική ενέργεια και στη δεύτερη η μαγνήτιση (χωρίς απόλυτη τιμή). Για να φτιάξουμε τις λεγόμενες “χρονοσειρές” των μετρήσεών μας μπορούμε να δώσουμε τις εντολές

```
% gnuplot
gnuplot> plot "out.dat" using 1 with lines
gnuplot> plot "out.dat" using 2 with lines
gnuplot> plot "out.dat" using (($2>0)?$2:-$2) with lines
```

Στην τελευταία γραμμή φτιάχνουμε το γράφημα της  $M$  παίρνοντας την απόλυτη τιμή του αθροίσματος  $\sum_i s_i$ . Ζητάμε από το `gnuplot` να μας κάνει τη γραφική παράσταση παίρνοντας τη δεύτερη στήλη `$2` αν αυτή είναι θετική (`$2>0`) αλλιώς το αντίθετό της `-$2`.

#### 4.2.2 ΒΕΛΤΙΩΣΗ ΤΟΥ INTERFACE

Στην ενότητα αυτή θα κάνουμε βελτιώσεις στον κώδικα, κυρίως στο επίπεδο αλληλεπίδρασης με το χρήστη (user interface). Το πρόγραμμα στη μορφή που του δώσαμε στην προηγούμενη ενότητα έχει προγραμματισμένες (hard coded) τις διάφορες παραμέτρους της προσομοίωσης και κάθε φορά που θέλουμε να τις αλλάξουμε πρέπει να τις κωδικοποιήσουμε στο αντίστοιχο αρχείο και να επαναμεταγλωττίσουμε τον κώδικα. Τέλος η άλλη αλλαγή αφορά στο να αποθηκεύουμε την τελευταία διάταξη στην οποία βρίσκεται το σύστημα και να την διαβάζουμε για αρχική διάταξη όταν ξεκινάμε μια καινούργια προσομοίωση.

Σίγουρα οι παράμετροι `beta`, `nsweeps`, `seed` κλπ είναι εύκολο να διαβάζονται από το πρόγραμμα χωρίς να κάνουμε κάτι το ιδιαίτερο. Αυτό που θα κάνουμε και χρειάζεται κάποια επεξήγηση είναι ότι στο νέο πρόγραμμα θα θέσουμε και το μέγεθος του συστήματος `L` να παρέχεται διαδραστικά από το χρήστη. Η διαφορά με τις προηγούμενες παραμέτρους, είναι ότι το `L` (άρα και το `N`) καθορίζει την ποσότητα μνήμης που θα ζητήσει το πρόγραμμα από το

λειτουργικό σύστημα για να τρέξει, μέσω του μεγέθους του array `s[N]`. Χρειαζόμαστε η γλώσσα προγραμματισμού που επιλέγουμε να μας παρέχει τη δυνατότητα να ζητάμε τη στιγμή που το πρόγραμμα τρέχει την απαραίτητη μνήμη για τα arrays που χρησιμοποιεί. Η γλώσσα C έχει τη συνάρτηση `malloc()` για τη λειτουργία αυτή<sup>4</sup>

Ας εξηγήσουμε στο σημείο αυτό πώς θα το κάνουμε. Καταρχήν το array `s[N]` δεν μπορεί να οριστεί εκ των προτέρων όπως είπαμε, αφού το `N` δεν είναι γνωστό. Αντί αυτού, το `N` γίνεται μια μεταβλητή τύπου `int` και ζητείται η τιμή της να δοθεί από το χρήστη. Όσο για το `s`, αυτό πια γίνεται ένα *pointer* σε `int`:

```
int L, N, *s;
```

Αφού η τιμή το `N` γίνει γνωστή, το πρόγραμμα καλεί την `malloc()`

```
s = (int *)malloc((size_t) (N*sizeof(int)));
```

Η `malloc()` παίρνει σαν όρισμα τον αριθμό των bytes που ζητάει ο χρήστης και επιστρέφει ένα *pointer* στη μνήμη με την αιτούμενη μνήμη (αν αποτύχει αυτό είναι το `NULL pointer`). Εμείς θέλουμε να αποθηκεύσουμε `N` ακέραιους άρα θέλουμε  $(N) \times$  (αρ. bytes που πιάνει στη μνήμη κάθε ακέραιος) bytes. Στο λειτουργικό που έχουμε, κάθε ακέραιος πιάνει 4 bytes μνήμης αλλά εμείς προτιμούμε αυτό να μας το δώσει η συνάρτηση `sizeof()` (εδώ `sizeof(int)=4`). Με τον τρόπο αυτό, το πρόγραμμά μας θα τρέχει σωστά και σε άλλες πλατφόρμες (portability). Το cast `(size_t)` χρειάζεται γιατί όπως μαθαίνουμε από το εγχειρίδιο της `malloc` το όρισμά της πρέπει να είναι αυτού του τύπου. Το cast `(int *)` στο *pointer* που επιστρέφει η `malloc()` χρειάζεται, αλλιώς θα πάρουμε *pointer* σε `void`. Δείτε το εγχειρίδιο της

```
% man malloc
```

Όταν η μνήμη που ζητήσαμε δε μας είναι πια αναγκαία, μπορούμε να την ελευθερώσουμε με την συνάρτηση `free()`

```
free(s);
```

Σε μας δεν είναι τώρα αναγκαίο αφού χρειαζόμαστε το `s` μέχρι το τέλος, αλλά το αναφέρουμε για πληρότητα. Όταν ξεχνάμε να το κάνουμε, το πρόγραμμά μας έχει "memory leaks" και μπορεί να δημιουργήσει προβλήματα στον εαυτό του και τους ... άλλους!

Περίληπτικά αναφέρουμε ότι οι άλλες αλλαγές που αξίζει να σημειωθούν είναι η αποθήκευση και διάβασμα της τελικής διάταξης του συστήματος και το πέρασμα των τιμών των παραμέτρων

<sup>4</sup>Με τη Fortran 77 δεν έχουμε τη δυνατότητα αυτή.

διαδραστικά μέσω unix options. Αυτές θα συζητηθούν στη μελέτη των μεταβολών στα αρχεία `init.c` και `end.c`.

Αρχίζουμε τη μελέτη του κώδικα από το αρχείο επικεφαλίδα (header file) `include.h`.

```

/***** 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();

```

Οι αλλαγές είναι λίγες αλλά θεμελιώδεις: Οι παράμετροι `L`, `N`, `XNN`, `YNN` δεν περνούν μέσω του preprocessor με εντολές `#define` αλλά ορίζονται σαν μεταβλητές που θα περάσει ο χρήστης όταν τρέχει το πρόγραμμα. Ορίζεται το pointer `int *s` για την διάταξη των σπιν. Οι υπόλοιπες αλλαγές αφορούν τον ορισμό συναρτήσεων που θα συζητήσουμε παρακάτω.

Προχωράμε στη `main()`:

```

/***** main.c *****/
#include "include.h"

int main(int argc, char **argv){

    int isweep;

    init(argc, argv);
    for(isweep=0; isweep<nsweep; isweep++){
        met();
    }
}

```

```

    measure();
}
simend();
}

```

Οι αλλαγές είναι λίγες: Βλέπουμε ότι η `init(argc,argv)` καλείται με ορίσματα τις `argc`, `argv` (βλ. σελίδα 44) έτσι ώστε να γίνει εκεί η επεξεργασία των δεδομένων που περνά ο χρήστης από τη γραμμή εντολών. Οι τιμές των παραμέτρων `nsweep`, `beta`, `start`, `seed` δεν ορίζονται εδώ. Και τέλος καλείται μια καινούργια συνάρτηση, η `simend()` (`simulation end`) η οποία εκτελεί όσες εντολές χρειάζονται για να κλείσει το πρόγραμμα. Εδώ η πιο σημαντική θα είναι η καταγραφή της τελευταίας διάταξης των spin σε ένα αρχείο.

Στην `init(argc,argv)` γίνονται οι αρχικοποιήσεις όλων των αναγκαίων δεδομένων για την προσομοίωση:

```

/***** 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     oseed=-1,OL=-1;
    FILE   *fin;

    /*define parameters from options:*/
    L=-1;beta=-1.;nsweep=-1;start=-1;seed=-1;
    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);
/*Initial Configuration: cold (0), hot (1) and old (2)*/
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 %lf %*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;
}

```

```
}

```

Ορισμένες παρατηρήσεις: Στις γραμμές

```
L=-1;beta=-1.;nsweep=-1;start=-1;seed=-1;
get_the_options(argc,argv);
```

ορίζουμε τις παραμέτρους να έχουν “παράξενες τιμές”, έτσι ώστε να ελέγξουμε αν αυτές έχουν τεθεί από το χρήστη από τη γραμμή εντολών. Η συνάρτηση `get_the_options(argc,argv)` κάνει ακριβώς αυτή τη δουλειά και θα τη δούμε λεπτομερώς παρακάτω. Μετά το κάλεσμά της υποθέτουμε πως οι σχετικές παράμετροι (πρέπει) να έχουν τεθεί στις σωστές τους τιμές. Στις επόμενες σειρές κάνουμε μερικούς απλούς ελέγχους αν αυτό όντως έχει γίνει σωστά. Αν όχι, τότε με το κάλεσμα της συνάρτησης `locerr("Error message.")` το πρόγραμμα σταματάει αφού τυπώσει στο `stderr` το μήνυμα “Error message.”.

Αν αρχίζουμε την προσομοίωση από μία καινούργια διάταξη σπιν (`start=0,1`) ο χρήστης πρέπει οπωσδήποτε να δώσει το μέγεθος του συστήματος `L`, το σπόρο `seed` της γεννήτριας ψευδοτυχαίων αριθμών και τη θερμοκρασία `beta`. Δεδομένων αυτών υπολογίζουμε τις ποσότητες `N`, `XNN`, `YNN` και ζητάμε τη μνήμη για το array `s`:

```
s = (int *)malloc((size_t) (N*sizeof(int)));
if (!s) locerr("allocation failure in s[N]");
```

Η δεύτερη γραμμή ελέγχει αν όντως ο πυρήνας του λειτουργικού διέθεσε τη ζητούμενη μνήμη. Αν όχι τότε το `s=NULL` και η παράσταση `(!s)` είναι μη μηδενική, άρα το `if(!s) ... ; statement` εκτελείται.

Στη συνέχεια ελέγχουμε αν τέθηκαν οι μεταβλητές `start`, `nsweep` και αν όχι το πρόγραμμα σταματάει. Μετά, όπως και πριν, υπολογίζεται το array `prob[5]` και καθορίζεται η αρχική διάταξη των σπιν για “cold” (`start=0`) ή “hot” (`start=1`) `start`. Η διαφορά τώρα έγκειται στην ανάγνωση ενός αρχείου που περιέχει μια παλιά αποθηκευμένη διάταξη σπιν (`start=2`).

Για την ανάγνωση του αρχείου με την παλιά διάταξη των σπιν υποθέτουμε πως το όνομα του είναι `conf` και έχει τη μορφή:

```
# Configuration of 2d Ising model on square lattice. Parameters (N=Lx*Ly) and s [N]
Lx= 12 Ly= 12 beta= 0.440000 seed= 235860773
-1
1
1
-1
.....
```

Δηλ. η πρώτη γραμμή είναι ένα ενημερωτικό μήνυμα, η δεύτερη έχει τις τιμές των παραμέτρων `L`, `beta`, `seed` που καθορίζουν πλήρως τη φυσική πληροφορία για τη διάταξη. Το φορμά είναι

```
"string" L "string" L "string" beta "string" seed
```

όπου για το πρόγραμμα το περιεχόμενο των "string" μπορεί να αγνοηθεί. Ακολουθούν οι τιμές των σπιν, μία ανά σειρά. Παρατηρούμε ότι η πληροφορία για τον `seed` μας επιτρέπει να ξεκινήσουμε τη γεννήτρια ψευδοτυχαίων αριθμών από το σημείο ακριβώς που είχε σταματήσει. Έτσι μπορούμε μια προσομοίωση που κρατάει για πολύ χρόνο να την κόψουμε σε μικρότερα κομμάτια χωρίς αυτό να έχει καμιά επίδραση στο δείγμα διατάξεων που θα μετρήσουμε. Αν είχαμε πιο πολύπλοκη γεννήτρια, τότε θα έπρεπε να αποθηκεύσουμε περισσότερη πληροφορία που να απεικονίζει την ακριβή κατάσταση της γεννήτριας.

Αρχικά ανοίγουμε το αρχείο `conf` με τη συνάρτηση `fopen()`. Αν αυτό αποτύχει (λ.χ. το αρχείο δεν υπάρχει), σταματάμε το πρόγραμμα:

```
if(!(fin=fopen("conf","r"))){locerr("Configuration file conf not found");
```

Η γραμμή του ενημερωτικού μηνύματος διαβάζεται σε ένα "buffer" `buf` (βοηθητικό χώρο μνήμης) και απλά αγνοείται

```
buf=(char *)malloc((size_t) (len*sizeof(char)));
getline(&buf,&len,fin);
```

Στην πρώτη γραμμή ζητείται ο χώρος στη μνήμη για την `buf`. Στη δεύτερη γραμμή διαβάζουμε μια γραμμή από το FILE pointer `fin` (που ανοίξαμε με την προηγούμενη εντολή και "δείχνει" στο `conf`) και την αποθηκεύουμε στο `buf`. Αυτό γίνεται με τη συνάρτηση `getline()` η οποία για ασφάλεια μας ζητάει πόσα το πολύ bytes θα διαβάσει από τη γραμμή με τη μεταβλητή `len` (εμείς θέσαμε παραπάνω την τιμή της σε 1024)<sup>5</sup>

Στη συνέχεια διαβάζουμε τη γραμμή με τις φυσικές πληροφορίες για τη διάταξη.

```
fscanf(fin,"%s %d %s %d %s %lf %s %ld",&OL,&OL,&obeta,&oseed);
```

Προσέξτε πώς κάνουμε την συνάρτηση `fscanf()` να αγνοήσει τα σχόλια με τα "%s". Οι τιμές των μεταβλητών `L`, `beta`, `seed` της προηγούμενης διάταξης αποθηκεύονται σε προσωρινό χώρο `OL`, `obeta`, `oseed` ώστε να γίνει η επεξεργασία τους: δε θέλουμε πάντα

<sup>5</sup> Διαβάζοντας με άλλες συναρτήσεις (λ.χ. την `fscanf()`) μπορούμε να εισάγουμε `buffer overflow` προβλήματα που κάνουν το πρόγραμμά μας ευάλωτο από κακόβουλους χρήστες.

να είναι αυτές οι τιμές που θα χρησιμοποιήσουμε στην προσομοίωσή μας. Φυσικά αυτό δεν ισχύει για το μέγεθος του συστήματος, οπότε αν δεν διαβάσαμε διάταξη σωστού μεγέθους σταματάμε το πρόγραμμα:

```
if( L < 0 )L=0L;
if( L != 0L )locerr("Given L different from the one read from conf");
```

Στην πρώτη γραμμή, αν ο χρήστης δεν έχει επιβάλλει κάποιο  $L$  χρησιμοποιούμε αυτό της παλιάς διάταξης. Το ίδιο ισχύει και για τις  $seed$ ,  $beta$  αλλά εδώ επιτρέπουμε στο χρήστη να χρησιμοποιήσει νέες τιμές αν το επιθυμεί (λ.χ. προσομοίωση σε νέα θερμοκρασία ή ξεκίνημα της γεννήτριας ψευδοτυχαίων αριθμών από καινούργιο σημείο).

Στη συνέχεια, αφού ζητήσουμε όπως και πριν χώρο μνήμης για το  $s$  διαβάζουμε τη διάταξη των σπιν, ελέγχοντας όμως δυο πιθανές πηγές σφάλματος: Το αρχείο  $conf$  να είναι "φθαρμένο" (corrupted) ή να περιέχει μη αποδεκτές τιμές για τα σπιν (δηλ.  $\neq \pm 1$ ).

```
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);
```

Στη γραμμή

```
if( fscanf(fin,"%d",&s[i]) != 1 || (s[i] != 1 && s[i] != -1))
```

η συνάρτηση  $fscanf(fin,"%d",&s[i])$  έχει τιμή 1 κάθε φορά που διαβάζει μια τιμή των σπιν το οποίο απαιτούμε να συμβαίνει όσο διαβάζουμε τα  $N$  σπινς. Η δεύτερη απαίτηση ( $s[i] != 1 \ \&\& \ s[i] != -1$ ) είναι τα σπιν να έχουν απαραίτητα τις τιμές  $\pm 1$ . Τέλος κλείνουμε το αρχείο  $conf$  με τη συνάρτηση  $fclose()$  και τυπώνουμε ένα πληροφοριακό μήνυμα στο  $stdout$  με τη συνάρτηση  $simmessage(stdout)$ .

Στο τέλος της προσομοίωσης καλούμε την συνάρτηση  $simmend()$  για το τελικό "σκούπισμα". Τον κώδικά της τον αποθηκεύουμε στο αρχείο  $end.c$ .

```
/****** 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.");
fprintf(fout," 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);
}

```

Με τη συνάρτηση `rename()` μετονομάζουμε το αρχείο της παλιάς διάταξης (αν υπάρχει) σε `conf.old`. Στη συνέχεια με την `fout = fopen("conf","w")` ανοίγουμε αρχείο `conf` για γράψιμο ("w"). Τυπώνουμε τα δεδομένα σύμφωνα με τις συμβάσεις που αναφέραμε παραπάνω για το φoρμά του αρχείου μιας διάταξης των σπιν.

Τέλος παραθέτουμε και τον κώδικα για τις συναρτήσεις `get_the_options()` η οποία διαβάζει τις παραμέτρους μέσω `unix options`, `simmesssage()` που τυπώνει πληροφορίες που καθορίζουν τις συνθήκες της προσομοίωσης και `locerr()` που σταματάει το πρόγραμμα τυπώνοντας ένα μήνυμα στο `stderr`. Η επιλογή να περάσουμε τις παραμέτρους μέσω `unix options` έχει τα πλεονεκτήματα ότι τις περνάμε προαιρετικά και με όποια σειρά θέλουμε. Ας δούμε κατ' αρχήν πώς δουλεύουν. Έστω ότι το εκτελέσιμο αρχείο του προγράμματος είναι το `is`. Η εντολή

```
% ./is -L 10 -b 0.44 -s 1 -S 5342 -n 1000
```

θα τρέξει το πρόγραμμα θέτοντας  $L=10$  (`-L 10`),  $\beta=0.44$  (`-b 0.44`),  $\text{start}=1$  (`-s 1`),  $\text{seed}=5342$  (`-S 5342`) και  $\text{nsweep}=1000$  (`-n 1000`). Τα `-L`, `-b`, `-s`, `-S`, `-n` είναι τα `options` ή `switches` και μπορούμε να τα βάλουμε με όποια σειρά θέλουμε. Οι τιμές που ακολουθούν είναι οι τιμές που θα περάσουν στις τιμές των αντίστοιχων μεταβλητών. Υπάρχουν και `options` που δεν απαιτούν ορίσματα, αλλά μεταβάλλουν τον τρόπο που λειτουργεί το πρόγραμμα (θυμηθείτε λ.χ. πώς το `-l` αλλάζει τον τρόπο λειτουργίας της εντολής `ls` αν την εκτελέσουμε ως `ls -l`). Στην περίπτωσή μας, το `option -h` (`help`) τυπώνει μια σύντομη βοήθεια χρήσεως για τον χρήστη και σταματάει το πρόγραμμα:

```
% ./is -h
```

```
Usage: is [options]
-L: Lattice length (N=L*L)
-b: beta (options beta overrides the one in config)
-s: start (0 cold, 1 hot, 2 old config.)
-S: seed (options seed overrides the one in config)
-n: number of sweeps and measurements of E and M
```

Monte Carlo simulation of 2d Ising Model. Metropolis is used by default. Using the options, the parameters of the simulations must be set for a new run (start=0,1). If start=2, a configuration is read from the file conf.

Έτσι μπορούμε να θυμόμαστε τον τρόπο χρήσης του προγράμματος.

Ας δούμε πώς το προγραμματίζουμε:

```
/****** 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 "hL:b:s:S:n:"
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 '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;
    default:
        errflg++;
    }/*switch*/
    if(errflg) usage(argv);
}/*while...*/
}/*get_the_options()*/
/* ----- usage() ----- */
void usage(char **argv){
    /*Careful: New lines end with \n\ : No space after last backslacsh
       indicates line 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\
Monte Carlo simulation of 2d Ising Model. Metropolis is used by \n\
default. 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()*/
/* ----- locerr() ----- */
void locerr( char *errmes ){
    fprintf(stderr,"%s: %s Exiting...\n",prog,errmes);
    exit(1);
}
/* ----- simmessage() ----- */
/*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()*/

```

Καλώντας την συνάρτηση `get_the_options(argc,argv)`, της περνάμε τα ορίσματα της γραμμής εντολών, συμπεριλαμβανομένων και των options. Η συνάρτηση `getopt(argc, argv, OPTARGS)` παίρνει τα ορίσματα της γραμμής εντολών και επεξεργάζεται εκείνα που είναι options. Ποιοι χαρακτήρες είναι options καθορίζονται από την ακολουθία χαρακτήρων `OPTARGS` την οποία ορίζουμε στη γραμμή

```
#define OPTARGS "hL:b:s:S:n:"
```

να έχει την τιμή `"hL:b:s:S:n:"`. Παρατηρήστε ότι εκεί έχουμε βάλει όλους τους χαρακτήρες που αναφέραμε παραπάνω ως options. Οι χαρακτήρες `L`, `b`, `s`, `S`, `n` ακολουθούνται από το χαρακτήρα `:"` που υποδεικνύει ότι είναι options που παίρνουν όρισμα. Αντίθετα ο `h` δεν παίρνει. Παρατηρήστε τώρα, πώς με το `switch(c)` block καθορίζουμε τις δράσεις του προγράμματος κάθε φορά που περνά ένα option. Λ.χ. η τιμή της μεταβλητής `L` τίθεται από το case

```

case 'L':
    L = atoi(optarg);
    break;

```

με τη βοήθεια φυσικά της `atoi()` όπως είχαμε περιγράψει στη σελίδα 44.

Η συνάρτηση `usage(argv)` είναι αυτή που τυπώνει το σύντομο μήνυμα βοήθειας στο χρήστη (παρατηρήστε πώς καλείται όταν περνάμε το option `-h`). Η `locerr(errmes)` τυπώνει ένα μήνυμα σφάλματος και καλεί την `exit(1)` για να σταματήσει το πρόγραμμα με non-zero error status (1) (ένδειξη μη ομαλής λειτουργίας του προγράμματος).

Τέλος η `simmessage()` παίζει εξαιρετικά σημαντικό ρόλο στην αποθήκευση των δεδομένων. Τοποθετεί “ετικέτα” στα αποτελέσματα μας που χωρίς αυτή θα ήταν όσο επικίνδυνα όσο και ένα φάρμακο χωρίς ετικέτα: Μπορεί να το πάρεις και να μην πάθεις τίποτα αλλά μπορεί και .... Η πληροφορία που βάζουμε για τα δεδομένα μας πρέπει να είναι τόσο σαφής ώστε να επιτρέπει στο χρήστη των δεδομένων να έχει όλη τη σχετική φυσική πληροφορία. Παρατηρήστε πώς παίρνουμε το χρόνο που ξεκινάει το πρόγραμμα με τις συναρτήσεις `time()` και `ctime()` και πώς αντιγράφουμε μεταβλητές περιβάλλοντος (`environment variables`) με τις συναρτήσεις `getenv()` και `strcpy()`. Δείτε τα σχετικά εγχειρίδια (`man 2 time`, `man 3 ctime`, `man getenv`). Η πληροφορία σχετικά με τον υπολογιστή που τρέχουμε μπορεί να έχει σημασία όταν μετράμε τις αποδόσεις του προγράμματος αλλά ακόμα και την ορθότητα των αποτελεσμάτων<sup>6</sup>!

Για να μεταγλωττίσουμε τον κώδικα χρησιμοποιούμε τη γνωστή εντολή

```
% cc -O2 main.c init.c met.c measure.c end.c options.c drandom.c \
  -o is -lm
```

και τρέχουμε το πρόγραμμα όπως αναφέραμε παραπάνω. Λ. χ.

```
% /usr/bin/time ./is -L 10 -b 0.44 -s 1 -S 5342 -n 10000 >& out.dat &
```

Όπου προσθήσαμε και την εντολή `time` για τη μέτρηση των αποδόσεων του προγράμματος.

Ένα χρήσιμο εργαλείο για τη μεταγλώττιση του προγράμματος μας είναι το πρόγραμμα `make`. Θα μπορούσε να γραφτεί ολόκληρο κεφάλαιο για το πρόγραμμα αυτό, παραπέμπουμε τον ενδιαφερόμενο αναγνώστη στις σχετικές `info pages` (εντολή: `info make`). Ας αρκεστούμε να πούμε πως αν στον κατάλογο που βρίσκεστε υπάρχει ένα αρχείο με όνομα `Makefile` με περιεχόμενα<sup>7</sup>:

```
# ##### Makefile #####
OBSJ = main.o init.o met.o measure.o end.o options.o drandom.o
```

<sup>6</sup>Υπάρχουν περιπτώσεις που `bugs` εξαρτώνται από μεταγλωττιστές - είτε από σφάλμα δικό μας (λ.χ. αρχικοποίηση τιμών μεταβλητών) είτε από `bugs` του μεταγλωττιστή (λ.χ. σφάλματα από βελτιστοποίηση είναι τα συχνότερα), από το λειτουργικό σύστημα ή βιβλιοθήκες που χρησιμοποιούμε. Γενική συμβουλή είναι να ελέγχετε πάντα ως προς την ορθότητα τα αποτελέσματά σας σε ένα νέο περιβάλλον.

<sup>7</sup>Προσοχή, ένα από τα ... βίτσια του `make` είναι ότι οι εκτελέσιμες εντολές (`$(CC)`), `/bin/rm` στο παράδειγμα) βρίσκονται σε μια γραμμή που αρχίζει από `TAB` και όχι από κενό ή άλλο χαρακτήρα. Άρα στο παραπάνω παράδειγμα `Makefile` ο κενός χώρος που φαίνεται στις γραμμές αυτές είναι ένα `TAB` και όχι 8 κενό χαρακτήρες

```

CFLAGS = -O2
LIBS    = -lm

is: $(OBJS)
$(CC) $(CFLAGS) $^ -o $@ $(LIBS)

$(OBJS): include.h

clean:
/bin/rm -f *.o is core*

τότε η εντολή

% make

```

είναι αρκετή για να “φτιάξει” το εκτελέσιμο αρχείο `is`. Τι κερδίσαμε? Για να δείτε, κάντε μια μετατροπή σε ένα από τα αρχεία του κώδικα, λ.χ. στο `main.c` και εκτελέστε την εντολή `make` ξανά. Θα δείτε πως μόνο το αρχείο που μεταβάλατε ξαναμεταγλωττίζεται και όχι τα άλλα. Θέτοντας κανόνες εξάρτησης (dependencies) μπορείτε να πετύχετε ακόμα πιο πολύπλοκες μεταγλωττίσεις. Παρατηρήστε επίσης ότι με τη χρήση μεταβλητών μπορούμε να κωδικοποιήσουμε τη σύνδεση με βιβλιοθήκες (λ.χ. η μεταβλητή `LIBS` παραπάνω), επίπεδα βελτιστοποίησης, μεταγλώττιση υπό συνθήκες (αρχιτεκτονική, μεταγλωττιστής κλπ) και άλλα ... πολλά! Ενθαρρύνουμε τον φοιτητή με ενδιαφέρον στον προγραμματισμό να επενδύσει λίγο χρόνο να μάθει περισσότερα.

### 4.3 EN ΑΡΧΗ ΗΝ ΨΥΧΟΣ (Η ... ΖΕΣΤΗ)

Στην ενότητα αυτή θα συζητήσουμε τα τεχνικά προβλήματα που συναντά μια προσομοίωση μέχρι να φέρουμε το σύστημα σε θερμική ισορροπία. Το πρόβλημα αυτό μπορεί να είναι από αμελητέο μέχρι και απαγορευτικό για την προσομοίωση κάποιων συστημάτων. Είναι όμως το πρώτο πρόβλημα που μας απασχολεί από τη στιγμή που έχουμε ένα πρόγραμμα που (νομίζουμε ότι) κάνει σωστούς υπολογισμούς. Έτσι παρ’ όλο που όπως ήδη αναφέραμε προηγουμένως για το διδιάστατο πρότυπο Ising δεν είναι ιδιαίτερα σημαντικό πρόβλημα, θα το μελετήσουμε λόγω της σπουδαιότητάς του για άλλα προβλήματα.

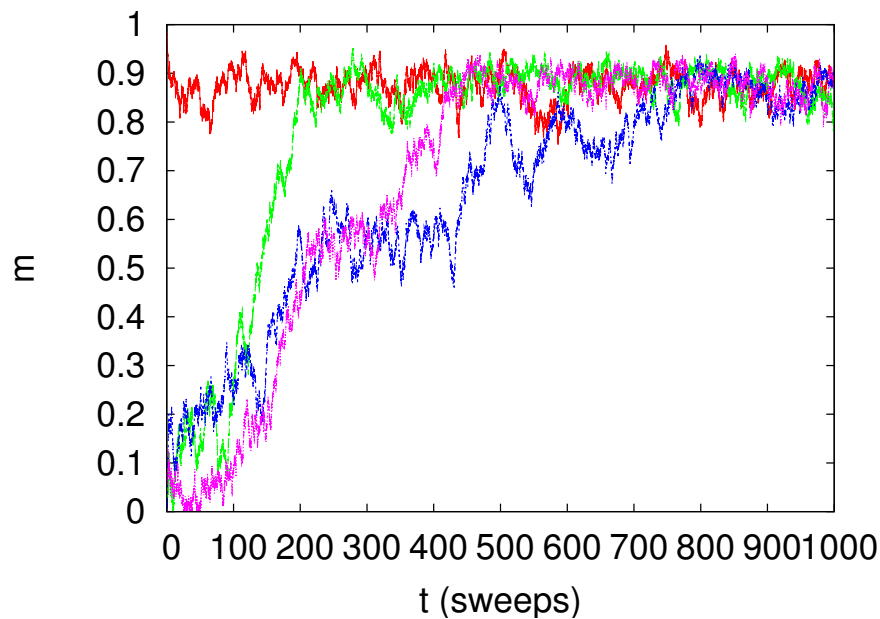
Όπως ήδη είδαμε, όταν ξεκινούμε μια προσομοίωση χρειάζεται να τοποθετήσουμε το σύστημά μας σε μια αρχική κατάσταση έτσι ώστε με τον αλγόριθμο που έχουμε επιλέξει να ξεκινήσουμε μια

διαδικασία Markov για την παραγωγή του δείγματός μας. Γνωρίζουμε όμως από την ενότητα 1.2 πως το σύστημά μας όταν είναι σε θερμική ισορροπία με μια δεδομένη θερμοκρασία, τότε βρίσκεται κυρίως σε καταστάσεις με ενέργεια που πολύ λίγο διαφέρει από μια μέση τιμή, ενώ καταστάσεις με ενέργεια που διαφέρει κατά πολύ τις επισκέπτεται πολύ σπάνια<sup>8</sup>. Έτσι αν η αρχική κατάσταση που θα επιλέξουμε δεν είναι μια τυπική κατάσταση θα πρέπει το σύστημά μας να κάνει έναν τυχαίο ... περίπατο στο χώρο των καταστάσεων μέχρι να “βρει” την περιοχή των καταστάσεων εκείνη που αντιστοιχεί στην θερμοκρασία που έχουμε επιλέξει. Αυτή είναι η διαδικασία της εύρεσης της κατάστασης θερμικής ισορροπίας (thermalization).

Εδώ παρουσιάζονται δύο προβλήματα: Το πρώτο είναι η κατάλληλη επιλογή της αρχικής κατάστασης και το δεύτερο είναι να εντοπίσουμε πότε το σύστημά μας έχει βρεθεί σε κατάσταση θερμικής ισορροπίας. Ως προς το πρώτο, έχουμε ήδη δει πως η αρχική κατάσταση για το πρότυπο Ising μπορεί να επιλεγεί να είναι (α) παγωμένη (β) ... καυτή (γ) σε μια άλλη - πιθανώς γειτονική - θερμοκρασία από μια προηγούμενη προσομοίωση. Είναι φυσικά κατανοητό πως αν επιλέξουμε την αρχική κατάσταση να είναι λ.χ. καυτή για να προσομοιώσουμε το σύστημα σε μια χαμηλή θερμοκρασία ότι θα αργήσει περισσότερο να βρεθεί σε θερμική ισορροπία από ότι αν επιλέγαμε την αρχική κατάσταση να είναι παγωμένη (ή σε μια κοντινή θερμοκρασία). Αυτό φαίνεται στο Σχήμα 4.2 όπου η λανθασμένη επιλογή κάνει τη διαδικασία αρκετά δυσκολότερη. Η δυσκολία της διαδικασίας εξαρτάται από τη φυσική ποσότητα, τη θερμοκρασία και το μέγεθος του συστήματος. Συνοπτικά, ευκολότερα η ενέργεια βρίσκεται στη θερμική περιοχή παρά η μαγνήτιση. Μια τοπική ποσότητα (local quantity, fast mode) μελετάται ευκολότερα από μια που έχει μη τοπικές εξαρτήσεις (non--local quantity, slow mode). Η κατάσταση θερμικής ισορροπίας βρίσκεται ευκολότερα (αν φυσικά αρχίσουμε από την κατάλληλη αρχική κατάσταση) όταν είμαστε μακριά από την κρίσιμη θερμοκρασία. Γενικά αναμένεται ότι είναι ευκολότερο να φέρουμε σε θερμική ισορροπία ένα μικρό σύστημα παρά ένα μεγάλο.

Το δεύτερο πρόβλημα είναι να εντοπίσουμε πότε το σύστημά μας έχει βρεθεί σε κατάσταση θερμικής ισορροπίας και φυσικά να αγνοήσουμε όλες τις μετρήσεις που έχουμε κάνει πριν. Από το σχήμα 4.2 βλέπουμε πως η στιγμή αυτή, ακόμα και για την ίδια αρχική κατάσταση, μπορεί να έχει τιμή με μεγάλες διακυμάνσεις. Ο

<sup>8</sup>Φυσικά αυτό γίνεται εντονότερο όσο το σύστημά μας έχει μεγαλύτερο μέγεθος.

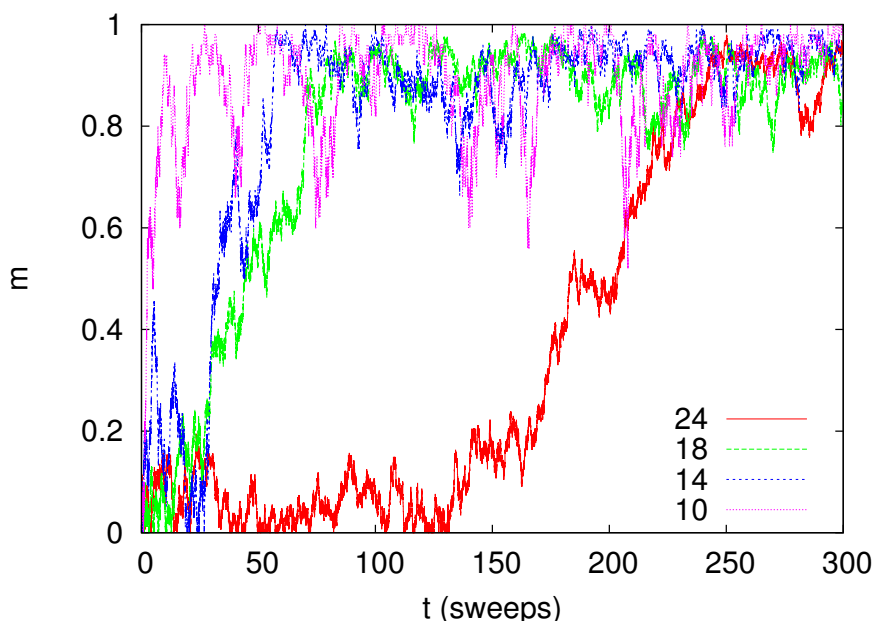


Σχήμα 4.2: Μαγνήτιση ανά πλεγματική θέση για το πρότυπο Ising με  $L = 40$ ,  $\beta = 0.48$ . Φαίνεται η διαδικασία εύρεσης θερμικής ισορροπίας για μια αρχική κατάσταση παγωμένη και τρεις αρχικές καταστάσεις καυτές. Επαναλαμβάνοντας τη διαδικασία αυτή για τη δεύτερη περίπτωση φαίνεται πως για να είμαστε σίγουροι πως το σύστημα είναι σε κατάσταση θερμικής ισορροπίας χρειαζόμαστε περισσότερα από  $\sim 1000$  sweeps.

πιο απλός τρόπος είναι, αν φυσικά δεν έχει απαγορευτικό κόστος, να ξεκινήσουμε αρκετές προσομοιώσεις με διαφορετικές αρχικές καταστάσεις ή/και από την ίδια αρχική κατάσταση αλλά ακολουθώντας διαφορετική στοχαστική ... πορεία και να παρακολουθήσουμε την εξέλιξη στον (Μόντε Κάρλο) χρόνο των προς μέτρηση φυσικών ποσοτήτων. Όταν όλα τα μονοπάτια συναντηθούν, τότε έχουμε πετύχει θερμική ισορροπία και μπορούμε να ξεκινήσουμε τις μετρήσεις μας.

Πιο συστηματικά όμως αποκτούμε πεποίθηση ότι όντως έχουμε πετύχει θερμική ισορροπία όταν τα αποτελέσματά μας έχουν σταθερή τιμή (μέσα στα όρια του στατιστικού σφάλματος) καθώς αποκόπτουμε από το δείγμα μας όλο και μεγαλύτερο αριθμό από μετρήσεις. Η διαδικασία φαίνεται στο σχήμα 4.5 όπου για τις μετρήσεις του πάνω σχήματος αποκόπτουμε διαδοχικά 0, 20, 50, 100, 200, 400, 800, 1600, 3200 και 6400 αρχικές μετρήσεις μέχρι η μέση τιμή της μαγνήτισης να πάρει μια σταθερή τιμή μέσα στα όρια του στατιστικού σφάλματος.

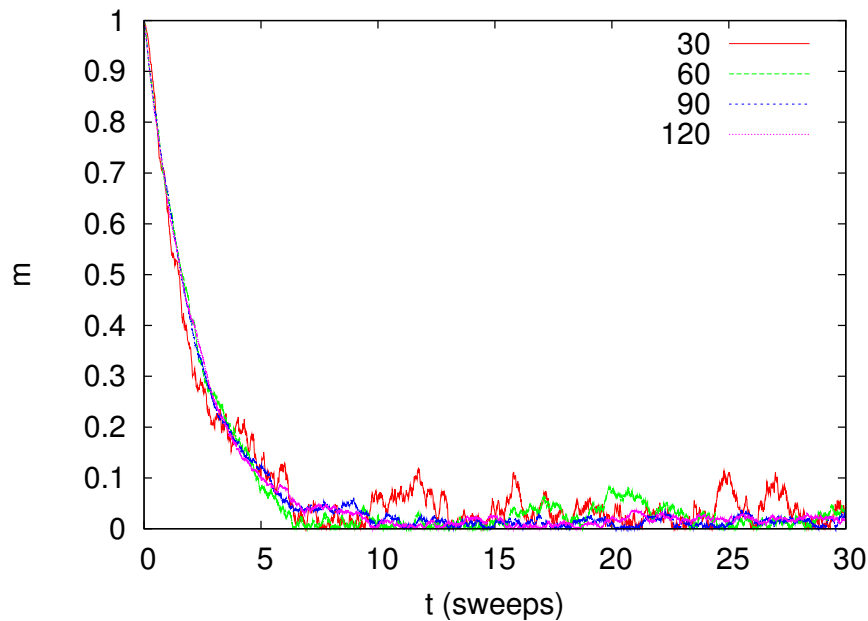




Σχήμα 4.3: Μαγνήτιση ανά πλεγματική θέση για το πρότυπο Ising με  $L = 10, 14, 18, 24$  και  $\beta = 0.50$ . Φαίνεται η διαδικασία εύρεσης θερμικής ισορροπίας για μια αρχική κατάσταση καυτή και ότι η δυσκολία αυξάνει με το μέγεθος του συστήματος.

#### 4.4 ΤΟ ΖΗΤΟΥΜΕΝΟ, Η ΑΝΕΞΑΡΤΗΣΙΑ

Για να αποκτήσουμε ένα δείγμα από στατιστικά ανεξάρτητες μετρήσεις θα έπρεπε μέσα από μια διαδικασία Markov το επόμενο βήμα να μην έχει στατιστικό συσχετισμό με το προηγούμενο. Είναι όμως φανερό πως με τους αλγόριθμους που χρησιμοποιούμε αυτό δεν είναι δυνατόν. Στον αλγόριθμο Metropolis το επόμενο βήμα διαφέρει από το προηγούμενο το πολύ κατά την τιμή ενός σπιν. Άρα η επόμενη διάταξη των σπιν είναι στατιστικά πολύ ισχυρά συσχετισμένη με την προηγούμενη. Στην καλύτερη περίπτωση θα περίμενε κανείς να πάρει μία πρακτικά στατιστικά ανεξάρτητη διάταξη όταν κάνει 1 βήμα Metropolis ανά πλεγματική θέση, αυτό που ονομάζεται ένα “sweep” του πλέγματος. Ενώ αυτό όντως συμβαίνει στην πράξη για πολλές περιπτώσεις, λ.χ. στο πρότυπο Ising για θερμοκρασίες αρκετά διαφορετικές από την κρίσιμη  $\beta_c$ , στις περισσότερες ενδιαφέρουσες περιπτώσεις χρειαζόμαστε πολύ μεγαλύτερο αριθμό από sweeps για να πάρουμε μια στατιστικά ανεξάρτητη διάταξη. Αυτό γίνεται λ.χ. όταν το μήκος συσχετισμού  $\xi$  (1.48) γίνεται πολύ μεγαλύτερο από μια πλεγματική θέση, κάτι



Σχήμα 4.4: Μαγνήτιση ανά πλεγματική θέση για το πρότυπο Ising με  $L = 30, 60, 90, 120$  και  $\beta = 0.20$ . Φαίνεται η διαδικασία εύρεσης θερμικής ισορροπίας για μια αρχική κατάσταση παγωμένη και ότι εδώ η δυσκολία δεν αυξάνει με το μέγεθος του συστήματος.

που συμβαίνει στην κρίσιμη περιοχή μιας συνεχούς μετάβασης φάσης. Αυτό μπορεί να φανεί στα σχήματα 4.6. Καθώς πλησιάζουμε την κρίσιμη θερμοκρασία από τη θερμή περιοχή όπου τα σπιν είναι σε αταξία, έχουμε τη δημιουργία μεγάλων περιοχών (clusters) από όμοια σπιν. Καταλαβαίνουμε ότι για να έχουμε δύο ασυσχέτιστες διατάξεις σπιν στην περίπτωση αυτή, πρέπει το μέγεθος, το σχήμα και τη θέση αυτών των clusters να είναι ασυσχέτιστα. Για να γίνει κάτι τέτοιο, πρέπει να καταστραφούν τα υπάρχοντα clusters και να δημιουργηθούν κάπου αλλού. Με ένα single-flip αλγόριθμο όπως ο Metropolis η διαδικασία αυτή θα παίρνει αρκετό χρόνο.

Για την ποσοτική μελέτη του αυτοσυσχετισμού των διατάξεων σπιν που παράγονται στην προσομοίωσή μας χρησιμοποιούμε τη συνάρτηση αυτοσυσχετισμού (autocorrelation function). Έστω μια φυσική ποσότητα  $\mathcal{O}$  (λ.χ. ενέργεια, μαγνήτιση, μαγνητική επιδεκτικότητα κλπ) και  $\mathcal{O}(t)$  η τιμή της μετά από Μόντε Κάρλο "χρόνο"  $t$ . Οι μονάδες του  $t$  μπορεί να είναι sweeps ή (υπο)πολλαπλάσια των sweeps. Η συνάρτηση αυτοσυσχετισμού  $\rho_{\mathcal{O}}(t)$  της φυσικής πο-

σότητας  $\mathcal{O}$  ορίζεται για κάθε χρόνο  $t$  να είναι η

$$\rho_{\mathcal{O}}(t) = \frac{\langle (\mathcal{O}(t') - \langle \mathcal{O} \rangle)(\mathcal{O}(t'+t) - \langle \mathcal{O} \rangle) \rangle_{t'}}{\langle (\mathcal{O} - \langle \mathcal{O} \rangle)^2 \rangle}, \quad (4.22)$$

όπου  $\langle \dots \rangle_{t'}$  είναι μία μέση τιμή παίρνοντας ως αρχική τιμή όλα τα στοιχεία του δείγματος με  $t' < t_{\max} - t$ . Η κανονικοποίηση που βάζουμε στον παρονομαστή είναι τέτοια ώστε  $\rho_{\mathcal{O}}(0) = 1$ .

Ο ορισμός θυμίζει την συνάρτηση συσχετισμού των σπιν στο χώρο (βλ. εξ. (1.47)). Η συζήτηση για την σημασία του ορισμού είναι παράλληλη με αυτή της ενότητας 1.4. Με λίγα λόγια, όταν η τιμή μιας ποσότητας μετά από χρόνο  $t$  είναι ισχυρά συσχετισμένη με την αρχική, τότε το γινόμενο στον αριθμητή της (4.22) θα είναι σχεδόν πάντοτε θετικό, με αποτέλεσμα η μέση τιμή να είναι θετική. Όταν ο συσχετισμός είναι ασθενής, τότε το γινόμενο θα είναι πότε θετικό και πότε αρνητικό στο δείγμα, έτσι ώστε ο μέσος όρος να είναι μικρός. Αν υπάρχει αντισυσχετισμός τότε το κλάσμα είναι αρνητικό. Αρνητικές τιμές για την  $\rho_{\mathcal{O}}(t)$  οφείλονται στο πεπερασμένο μέγεθος του δείγματος και πρέπει να απορρίπτονται.

Η συνάρτηση  $\rho_{\mathcal{O}}(t)$  θα ελαττώνεται ασυμπτωτικά με εκθετικό τρόπο

$$\rho_{\mathcal{O}}(t) \sim e^{-t/\tau_{\mathcal{O}}}. \quad (4.23)$$

Η κλίμακα χρόνου  $\tau_{\mathcal{O}}$  χαρακτηρίζει τον αποσυσχετισμό των μετρήσεων της  $\mathcal{O}$  και ονομάζεται χρόνος αυτοσυσχετισμού της  $\mathcal{O}$ . Μετά από χρόνο  $2t$  η  $\rho_{\mathcal{O}}(t)$  έχει πέσει στο  $1/e^2 \approx 14\%$  της αρχικής της τιμής. Συμβατικά θεωρούμε ότι τότε έχουμε μία ανεξάρτητη μέτρηση της  $\mathcal{O}$ <sup>9</sup>. Άρα αν έχουμε  $t_{\max}$  μετρήσεις, ο αριθμός των ανεξάρτητων μετρήσεων της  $\mathcal{O}$  θα είναι

$$n_{\mathcal{O}} = \frac{t_{\max}}{2\tau_{\mathcal{O}}}. \quad (4.24)$$

Αν μία μέτρηση κοστίζει σε χρόνο, τότε από τα παραπάνω βλέπουμε ότι θα τη μετράμε σε διαστήματα χρόνου  $\sim \tau_{\mathcal{O}}$ . Αν όμως το κόστος της μέτρησης δεν είναι μεγάλο τότε μετράμε συχνότερα. Αυτό συμβαίνει γιατί υπάρχει στατιστική πληροφορία και στις μετρήσεις που είναι ελαφρά συσχετισμένες. Φυσικά όταν κάνουμε μέτρηση του  $\tau_{\mathcal{O}}$ , θα πρέπει να μετράμε και για  $t \ll \tau_{\mathcal{O}}$ .

Ένα παράδειγμα φαίνεται στο σχήμα 4.7 για την περίπτωση της μαγνήτισης ( $\mathcal{O} = m$ ). Υπολογίζουμε τη συνάρτηση  $\rho_m(t)$  και

<sup>9</sup>Οι χρόνοι αυτοσυσχετισμού μπορεί να είναι πολύ διαφορετικοί για διαφορετικές φυσικές ποσότητες. Δεν χρειάζεται να έχουμε πλήρως αποσυσχετισμένη διάταξη του συστήματος για μια ανεξάρτητη μέτρηση μια ποσότητας με μικρό  $\tau$ .

βλέπουμε ότι η προσαρμογή στην (4.23) είναι πολύ καλή με  $\tau_m = 235 \pm 3$  sweeps. Η μέτρηση έγινε σε ένα δείγμα από  $10^6$  μετρήσεις με 1 μέτρηση/sweep. Άρα οι ανεξάρτητες μετρήσεις μας ήταν στην πραγματικότητα  $\approx 10^6 / (2 \times 235) \approx 2128$ .

Ένας καλύτερος εκτιμητής του χρόνου αυτοσυσχετισμού είναι ο ολοκληρωμένος χρόνος αυτοσυσχετισμού  $\tau_{\text{int},\mathcal{O}}$  (integrated autocorrelation time). Ο ορισμός του προκύπτει από την υπόθεση (4.23) από όπου προκύπτει

$$\tau_{\text{int},\mathcal{O}} = \int_0^{+\infty} dt \rho_{\mathcal{O}}(t) \sim \int_0^{+\infty} dt e^{-t/\tau_{\mathcal{O}}} = \tau_{\mathcal{O}} \quad (4.25)$$

Προφανώς  $\tau_{\text{int},\mathcal{O}} \sim \tau_{\mathcal{O}}$ , αλλά στον υπολογισμό υπεισέρχονται συστηματικά σφάλματα, γι' αυτό οι δύο ορισμοί μπορεί να διαφέρουν λίγο στην πράξη. Επίσης τα στατιστικά σφάλματα, ιδιαίτερα από τη συμπεριφορά της  $\rho_{\mathcal{O}}(t)$  για μεγάλους χρόνους  $t \gg \tau_{\mathcal{O}}$  επιδρούν με διαφορετικούς τρόπους στους δύο ορισμούς. Αν το δείγμα μας είναι καλό, τότε θα έχουμε καλύτερη συμφωνία. Επίσης, επειδή το δείγμα μας δεν είναι άπειρο πρέπει να επιλέξουμε ένα  $t_{\text{max}}$  όπου θα κόψουμε την ολοκλήρωση. Για μικρούς χρόνους δε θα έχουμε ακριβώς τη συμπεριφορά (4.23) και αυτό θα προσθέσει στο συστηματικό σφάλμα. Αλλά παρόμοια προβλήματα έχουμε και όταν προσπαθούμε να υπολογίσουμε τον  $\tau_{\mathcal{O}}$  με προσαρμογή των δεδομένων: Πρέπει να γίνει εκτίμηση των ορίων στο διάστημα του χρόνου που θα γίνει η προσαρμογή που θα αντιστοιχεί στην (4.23), κάτι το οποίο αποτελεί ... τέχνη! Όταν θέλουμε μία γρήγορη εκτίμηση του χρόνου αυτοσυσχετισμού αγνοούμε τα συστηματικά σφάλματα για μικρούς χρόνους και υπολογίζουμε τον

$$\tau_{\text{int},\mathcal{O}}(t_{\text{max}}) = \int_0^{t_{\text{max}}} dt \rho_{\mathcal{O}}(t). \quad (4.26)$$

και κάνουμε τη γραφική παράσταση  $\tau_{\text{int},\mathcal{O}}(t_{\text{max}})$ . Όταν παρατηρήσουμε πως η τιμή του  $\tau_{\text{int},\mathcal{O}}(t_{\text{max}})$  σταθεροποιείται για κάποιο διάστημα του  $t_{\text{max}}$ , παίρνουμε αυτή την τιμή ως εκτιμητή του  $\tau_{\text{int},\mathcal{O}}$ . Έχουμε έτσι ένα απλό τρόπο

Η διαδικασία αυτή φαίνεται στο σχήμα 4.8 όπου χρησιμοποιήσαμε τις ίδιες μετρήσεις με αυτές του σχήματος 4.7. Το αποτέλεσμα είναι  $\tau_{\text{int},m} = 217(3)$  sweeps που είναι μικρότερο από αυτό που υπολογίσαμε προηγουμένως. Η διαφορά αυτή συχνά δεν είναι σημαντική. Συνήθως αυτό που μας ενδιαφέρει είναι η εξάρτηση του χρόνου αυτοσυσχετισμού από τις παραμέτρους του συστήματος, εδώ το μέγεθος  $L$  και τη θερμοκρασία  $\beta$ . Και οι δύο εκτιμητές θα μας δώσουν το ίδιο αποτέλεσμα για την κύρια (ασυμπτωτική) συμπεριφορά για αρκετά μεγάλα  $L$  και για  $\beta$  αρκετά κοντά στην

κρίσιμη θερμοκρασία  $\beta_c$ . Συνήθως ο υπολογισμός του  $\tau_{\text{int},\mathcal{O}}$  προτιμάται.

Ο χρόνος αυτοσυσχετισμού δεν είναι σοβαρό πρόβλημα όταν το σύστημα βρίσκεται μακριά από την κρίσιμη περιοχή θερμοκρασιών. Στο σχήμα 4.9 βλέπουμε πως αυτός είναι μόνο μερικά sweeps και ότι αυτό είναι ανεξάρτητο από το μέγεθος του συστήματος  $L$ . Όταν όμως πλησιάζουμε την κρίσιμη περιοχή τότε ο χρόνος αυτοσυσχετισμού μεγαλώνει. Στην κρίσιμη περιοχή η αύξηση αυτή παρουσιάζει βάρθμηση (scaling). Για τον αλγόριθμο του Metropolis αυτό σημαίνει ότι για μεγάλα  $L$  έχουμε

$$\tau \sim L^z, \quad (4.27)$$

όπου για την περίπτωση της μαγνήτισης έχουμε  $z = 2.1665 \pm 0.0012$  [20]. Αυτή η τιμή είναι αρκετά μεγάλη και κάνει τον αλγόριθμο αρκετά ακριβό. Σημαίνει πως ο χρόνος για την προσομοίωση του συστήματος για δεδομένο αριθμό ανεξάρτητων διατάξεων σπιν θα αυξάνει σαν

$$t_{\text{CPU}} \sim L^{d+z} \approx L^{4.17}. \quad (4.28)$$

Σε επόμενα κεφάλαια θα συζητήσουμε με περισσότερη λεπτομέρεια τη σχέση βάρθμησης (4.27) καθώς και καλύτερους αλγόριθμους που βελτιώνουν δραστικά τη συμπεριφορά αυτή.

## 4.5 ΑΝΑΖΗΤΩΝΤΑΣ ΤΟ ... ΣΦΑΛΜΑ

Σε μια τυχαία μεταβλητή, όπως είναι μία παρατηρήσιμη ποσότητα σε μία προσομοίωση Μόντε Κάρλο, η μέση τιμή της εκτιμώμενη από ένα δείγμα δε δίνει καμιά πληροφορία για την αξία της μέτρησης. Η γνώση της πραγματικής στατιστικής κατανομής δίνει όλη τη σχετική πληροφορία, αλλά στην πράξη συνήθως αρκούμαστε από τη γνώση του "σφάλματος" της μέτρησης. Για να ορίσουμε αυτό που θα ονομάζουμε στατιστικό σφάλμα στο βιβλίο αυτό, θα κάνουμε την πολύ απλή αλλά πρακτική υπόθεση ότι η κατανομή των μετρήσεων είναι Gaussian. Αυτό γενικά είναι μια πολύ καλή προσέγγιση όταν έχουμε ανεξάρτητες μετρήσεις σε ένα πρότυπο στατιστικής φυσικής. Η αρχή του στατιστικού σφάλματος είναι θερμικές διακυμάνσεις γύρω από τη μέση τιμή που όπως έχουμε συζητήσει στην ενότητα 1.2 (βλ. και σχέση (1.29)) έχουν συνήθως κατά πολύ καλή προσέγγιση Gaussian κατανομή. Χαρακτηριστικό των στατιστικών σφαλμάτων είναι ότι εξαλείφονται αυξάνοντας τον αριθμό των μετρήσεων. Αυτό γίνεται σχετικά σιγά αφού μειώνονται αντιστρόφως ανάλογα με την τετραγωνική ρίζα του μεγέθους το δείγματος.

Μια άλλη κατηγορία σφαλμάτων είναι τα *συστηματικά* σφάλματα τα οποία συνήθως δεν έχουν ... συστηματικό τρόπο να τα εκτιμήσουμε. Αυτά στην περίπτωση μας μπορούν να έχουν απλή αιτία (λ.χ. να μην είναι το σύστημα σε θερμική ισορροπία) ή αρκετά δυσκολότερη να εντοπιστεί (λ.χ. μία κακή γεννήτρια ψευδοτυχαίων αριθμών).

Τέλος στην περίπτωση που κάποιος χρησιμοποιεί ένα διακριτό πρότυπο (όπως το Ising) για να προσομοιώσει ένα συνεχές μοντέλο, τότε στους εκτιμητές υπεισέρχονται σφάλματα λόγω της διακριτοποίησης και του πεπερασμένου μεγέθους του συστήματος. Αυτά τα σφάλματα μειώνονται με την προσομοίωση μεγαλύτερων συστημάτων και αντιμετωπίζονται συστηματικά με μεθόδους όπως λ.χ. τη βάθμιση πεπερασμένου μεγέθους (finite size scaling) όπου οι τιμές για πεπερασμένο μέγεθος συστήματος μας δίνουν την τιμή για το απείρωσ μεγάλο σύστημα. Αυτό θα είναι αντικείμενο της μελέτης μας σε επόμενο κεφάλαιο.

#### 4.5.1 ΣΦΑΛΜΑΤΑ ΑΝΕΞΑΡΤΗΤΩΝ ΜΕΤΡΗΣΕΩΝ

Σύμφωνα με την υπόθεση ότι η πηγή των σφαλμάτων στις μετρήσεις μας είναι οι θερμικές διακυμάνσεις προκύπτει ότι η πραγματική τιμή εκτιμάται από τη μέση τιμή του δείγματος και ότι το σφάλμα θα είναι το σφάλμα της μέσης τιμής. Έτσι αν έχουμε ένα δείγμα από  $n$  μετρήσεις  $\mathcal{O}_0, \mathcal{O}_1, \dots, \mathcal{O}_{n-1}$

$$\langle \mathcal{O} \rangle = \frac{1}{n} \sum_{i=0}^{n-1} \mathcal{O}_i \quad (4.29)$$

$$(\delta \mathcal{O})^2 \equiv \sigma_{\mathcal{O}}^2 = \frac{1}{n-1} \left\{ \frac{1}{n} \sum_{i=0}^{n-1} (\mathcal{O}_i - \langle \mathcal{O} \rangle)^2 \right\} = \frac{1}{n-1} (\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2) \quad (4.30)$$

Οι παραπάνω σχέσεις υποθέτουν πως το δείγμα μας αποτελείται από ανεξάρτητες μετρήσεις. Αυτό όπως είδαμε δε συμβαίνει σε μια προσομοίωση Μόντε Κάρλο λόγω του αυτοσυσχετισμού των μετρήσεων. Είδαμε πως αν ο χρόνος αυτοσυσχετισμού, μετρημένος σε μονάδες "αριθμός μετρήσεων", είναι  $\tau_{\mathcal{O}}$ , σύμφωνα με την 4.24 θα έχουμε  $n_{\mathcal{O}} = n/(2\tau_{\mathcal{O}})$  ανεξάρτητες μετρήσεις. Στην περίπτωση αυτή μπορεί να δειχθεί ότι το πραγματικό στατιστικό σφάλμα στη μέτρηση της  $\mathcal{O}$  είναι [21] (δείτε επίσης Κεφ. 4.1 του [2])

$$(\delta \mathcal{O})^2 = \frac{1 + 2\tau_{\mathcal{O}}}{n-1} (\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2). \quad (4.31)$$

Αν  $\tau_{\mathcal{O}} \ll 1$  τότε παίρνουμε την (4.30). Αν  $\tau_{\mathcal{O}} \gg 1$  τότε παίρνουμε

$$(\delta\mathcal{O})^2 \approx \frac{2\tau_{\mathcal{O}}}{n-1} \left( \langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2 \right) \quad (4.32)$$

$$\approx \frac{1}{(n/2\tau_{\mathcal{O}})} \left( \langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2 \right) \quad (4.33)$$

$$\approx \frac{1}{n_{\mathcal{O}} - 1} \left( \langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2 \right) \quad (4.34)$$

που δεν είναι παρά η (4.30) για  $n_{\mathcal{O}}$  ανεξάρτητες μετρήσεις (υποθέσαμε ότι  $1 \ll n_{\mathcal{O}} \ll n$ ). Η παραπάνω σχέση δικαιολογεί το γεγονός ότι θεωρήσαμε τις μετρήσεις πρακτικά ανεξάρτητες μετά από χρόνο  $\sim 2\tau_{\mathcal{O}}$ .

Οι παραπάνω σχέσεις μας δίνουν την ελευθερία να επιλέξουμε όπως μας βολεύει το πόσο συχνά θα παίρνουμε μετρήσεις πάνω στο σύστημα. Αλλά έχουν μερικές δυσκολίες στην εφαρμογή τους. Η πρώτη είναι πως πρέπει να κάνουμε τη μέτρηση του χρόνου αυτοσυσχετισμού με τις γνωστές δυσκολίες που περιγράψαμε στην ενότητα 4.4. Η δεύτερη είναι ότι δεν ξέρουμε τα αποτελέσματα που θα έχει στον υπολογισμό του σφάλματος η πραγματική κατανομή των μετρήσεων, ιδιαίτερα αν η ποσότητα που μετράμε δεν είναι μια απλή τοπική ποσότητα όπως λ.χ. η ενέργεια. Ας πάρουμε για παράδειγμα τη μαγνητική επιδεκτικότητα (4.21). Αυτό προϋποθέτει τη μέτρηση των  $\langle m \rangle$  και  $\langle m^2 \rangle$  (ή ακόμα καλύτερα την ποσότητα  $(m_i - \langle m \rangle)$  πάνω σε κάθε διάταξη  $i$ ) και δεν μπορεί να οριστεί πάνω σε μία μόνο διάταξη των σπιν. Αν υπολογίσουμε τις παραπάνω ποσότητες αφού συγκεντρώσουμε το δείγμα μπαίνει το ερώτημα πώς θα υπολογιστεί το σφάλμα  $\delta\chi$ . Αυτό δεν είναι απλή συνάρτηση των  $\delta\langle m \rangle$  και  $\delta\langle m^2 \rangle$  λόγω του συσχετισμού των δύο ποσοτήτων (έτσι λ.χ. ο γνωστός από τα εισαγωγικά εργαστήρια φυσικής τύπος διάδοσης των σφαλμάτων δεν ισχύει).

#### 4.5.2 Ο ΣΟΥΓΙΑΣ ΕΙΝΑΙ ΧΡΗΣΙΜΟΣ

Η πιο απλή λύση που βρίσκει κανείς στα προβλήματα που αναφέραμε στην προηγούμενη υποενότητα είναι να χωρίσει κανείς το δείγμα του σε blocks ή bins. Το όνομα προέρχεται από την εικόνα ότι τις  $n$  μετρήσεις μας τις βάζουμε σε  $n_b$  "καλάθια" και κάθε καλάθι το χρησιμοποιούμε ως μία ανεξάρτητη μέτρηση. Αυτό θα είναι σωστό αν ο αριθμός των μετρήσεων σε κάθε καλάθι είναι  $b = (n/n_b) \gg \tau_{\mathcal{O}}$ . Αν  $\mathcal{O}_i^b$   $i = 0, \dots, n_b - 1$  είναι οι μέσες τιμές της  $\mathcal{O}$  στο καλάθι  $i$ , τότε το σφάλμα θα δίνεται από την (4.30)

$$(\delta\mathcal{O})^2 = \frac{1}{n_b - 1} \left\{ \frac{1}{n_b} \sum_{i=0}^{n_b-1} (\mathcal{O}_i^b - \langle \mathcal{O}^b \rangle)^2 \right\} \quad (4.35)$$

Αυτή είναι η μέθοδος binning ή blocking και είναι απλή στη χρήση της. Παρατηρήστε ότι ποσότητες όπως η μαγνητική επιδεκτικότητα υπολογίζονται σε κάθε καλάθι σαν να ήταν ένα ανεξάρτητο δείγμα και το σφάλμα είναι τότε εύκολο να υπολογιστεί από την (4.35). Όπως είπαμε, για να είναι δυνατή η μέθοδος αυτή θα πρέπει κάθε καλάθι να είναι στατιστικά ανεξάρτητο από το άλλο. Αν δεν είναι τότε το σφάλμα θα είναι υποεκτιμημένο κατά τον παράγοντα  $2\tau_{\mathcal{O}}/(n_b - 1)$  της σχέσης (4.31). Τα καλάθια είναι στην πράξη στατιστικά ανεξάρτητα όταν  $b \sim 2\tau_{\mathcal{O}}$ . Αν δεν γνωρίζουμε τον  $\tau_{\mathcal{O}}$  υπολογίζουμε επανειλημμένα το σφάλμα (4.35) μειώνοντας διαδοχικά τον αριθμό των καλάθιων  $n_b$ . Όταν το σφάλμα παύει να αυξάνεται και πάρει σταθερή τιμή, τότε θεωρούμε πως η μέθοδος συγκλίνει στη σωστή τιμή.

Η μέθοδος όμως που θα επιλέξουμε στις προσομοιώσεις μας είναι η μέθοδος του ... σουγιά (jackknife method). Είναι πιο σταθερή και δίνει πιο αξιόπιστα αποτελέσματα. Η βασική ιδέα είναι ίδια με τη μέθοδο binning με τη μόνη διαφορά ότι τα “καλάθια” φτιάχνονται με διαφορετικό τρόπο και η σχέση (4.35) ελαφρά τροποποιείται. Χωρίζουμε τα δεδομένα μας σε  $n_b$  bins τα οποία περιέχουν από  $b = n - (n/n_b)$  στοιχεία ως εξής: Το bin  $j$  προκύπτει από τα αρχικά δεδομένα  $\mathcal{O}_0, \dots, \mathcal{O}_{n-1}$  αν διαγράψουμε τα περιεχόμενα του  $j$ -καλαθιού της μεθόδου binning. Η διαδικασία φαίνεται σχηματικά στο σχήμα 4.13. Σε κάθε bin υπολογίζουμε τη μέση τιμή της  $\mathcal{O}$ . Προκύπτουν τότε οι μέσες τιμές  $\mathcal{O}_0^b, \mathcal{O}_1^b, \dots, \mathcal{O}_{n_b-1}^b$ . Τότε το σφάλμα στην μέτρηση της  $\mathcal{O}$  είναι

$$(\delta\mathcal{O})^2 = \sum_{j=0}^{n_b-1} \left( \mathcal{O}_j^b - \langle \mathcal{O}^b \rangle \right)^2 = n_b \left( \langle (\mathcal{O}^b)^2 \rangle - \langle \mathcal{O}^b \rangle^2 \right). \quad (4.36)$$

Όπως και στη μέθοδο binning ο αριθμός των bins καθορίζεται μεταβάλλοντάς τον και προσδιορίζοντας εκείνες τις τιμές για τις οποίες η τιμή του σφάλματος σταθεροποιείται

Για περισσότερες λεπτομέρειες και αποδείξεις των ισχυρισμών στην υποενότητα αυτή παραπέμπουμε τον αναγνώστη στο βιβλίο του Berg [2]. Στο παράρτημα Β'.1 δίνουμε πρόγραμμα για τη χρήση της μεθόδου και σχετικά παραδείγματα.

Η μέθοδος jackknife είναι η μέθοδος υπολογισμού των σφαλμάτων που προτιμάται από την πλειοψηφία της επιστημονικής κοινότητας και αυτή θα ακολουθήσουμε στο βιβλίο αυτό.

### 4.5.3 ΟΤΑΝ ΔΕΝΕΙΣ ΤΙΣ ΜΠΟΤΕΣ ΣΟΥ...

Μια άλλη μέθοδος, χρήσιμη στην εκτίμηση των σφαλμάτων, είναι η μέθοδος bootstrap. Έστω ότι έχουμε  $n$  ανεξάρτητες μετρή-



σεις. Από αυτές δημιουργούμε  $n_S$  τυχαία δείγματα από αυτές: Με ομοιόμορφη πιθανότητα διαλέγουμε μία μέτρηση από τις  $n$  μετρήσεις. Αυτό το κάνουμε  $n$  φορές από το ίδιο σύνολο των  $n$  μετρήσεων - δηλ. χωρίς να αφαιρούμε τη μέτρηση που διαλέξαμε από αυτές που θα διαλέξουμε την επόμενη φορά και έτσι φτιάχνουμε ένα δείγμα. Αυτό σημαίνει ότι το δείγμα θα περιέχει  $\sim 1-1/e \approx 63\%$  ίδιες μετρήσεις. Σε κάθε δείγμα  $i = 0, \dots, n_S - 1$  υπολογίζουμε την ποσότητα  $\mathcal{O}_i^S$  και από αυτές τη μέση τιμή

$$\langle \mathcal{O}^S \rangle = \frac{1}{n_S} \sum_{i=0}^{n_S-1} \mathcal{O}_i^S, \quad (4.37)$$

και την

$$\langle (\mathcal{O}^S)^2 \rangle = \frac{1}{n_S} \sum_{i=0}^{n_S-1} (\mathcal{O}_i^S)^2. \quad (4.38)$$

Η εκτίμηση για το σφάλμα στη μέτρηση του  $\langle \mathcal{O} \rangle$  είναι τότε

$$(\delta \mathcal{O})^2 = \langle (\mathcal{O}^S)^2 \rangle - \langle \mathcal{O}^S \rangle^2. \quad (4.39)$$

Πρέπει να τονίσουμε ότι ο παραπάνω τύπος δίνει το σφάλμα για ανεξάρτητες μετρήσεις. Αν έχουμε μη αμελητέους χρόνους αυτοσυσχετισμού, πρέπει να χρησιμοποιήσουμε τη διόρθωση που δίνει η σχέση

$$(\delta \mathcal{O})^2 = (1 + 2\tau_{\mathcal{O}}) \left( \langle (\mathcal{O}^S)^2 \rangle - \langle \mathcal{O}^S \rangle^2 \right) \quad (4.40)$$

Στο παράρτημα Β'.2 δείχνουμε πώς να χρησιμοποιήσουμε τη μέθοδο binning για να υπολογίσουμε το πραγματικό σφάλμα  $\delta \mathcal{O}$  χωρίς a priori γνώση του  $\tau_{\mathcal{O}}$ .

Για περισσότερες λεπτομέρειες παραπέμπουμε τον αναγνώστη στα άρθρα του Bradley Efron [22]. Στο παράρτημα Β'.2 δίνουμε πρόγραμμα για τη χρήση της μεθόδου και σχετικά παραδείγματα.

## 4.6 ΠΡΟΒΛΗΜΑΤΑ

1. Αποδείξτε ότι η (4.10) ικανοποιεί τη συνθήκη λεπτομερούς ισοζύγησης.
2. Γράψτε πρόγραμμα που θα σας τυπώνει την ποσότητα μνήμης σε bytes που πιάνουν οι ακόλουθοι τύποι μεταβλητών: char, int, long, ulong, real, double, size\_t
3. Μεταβάλετε τον πρόγραμμα έτσι ώστε να μετράει τον μέσο λόγο αποδοχής των βημάτων Metropolis. Δηλ. μετρήστε τον

αριθμό των δεκτών αλλαγών στα σπιν σε σχέση με τον αριθμό των επιχειρούμενων αλλαγών. Εξετάστε τη σχέση του αριθμού αυτού από τη θερμοκρασία και το μέγεθος του συστήματος.

4. Προσθέστε τον κατάλληλο κώδικα ώστε όταν ο χρήστης στέλνει σήμα από το λειτουργικό σύστημα (signal) το πρόγραμμα να τυπώνει την τρέχουσα διάταξη των σπιν του συστήματος. Μπορείτε αυτό να το κάνετε για τα INT και USR1 signals (στην πρώτη περίπτωση το πρόγραμμα να σταματάει, στη δεύτερη να συνεχίζει). Δείτε σχετικά:

```
% kill -1
% man kill
% man 2 signal
% man sigaction
```

5. Αναπαράγετε τα σχήματα 4.7 και υπολογίστε τον  $\tau_m$ . Επαναλάβετε το ίδιο για την ενέργεια και υπολογίστε τον  $\tau_e$ . Συγκρίνετε τις τιμές που βγάλατε με τις τιμές των  $\tau_{int,m}$  και  $\tau_{int,e}$ .
6. Αναπαράγετε τα σχήματα 4.9 και επαναλάβετε τον υπολογισμό σας για την ενέργεια.
7. Αναπαράγετε τα σχήματα 4.10. Επαναλάβετε το ίδιο και για την ενέργεια. Στη συνέχεια κάντε τα ανάλογα σχήματα για τους  $\tau_{int,m}$  και  $\tau_{int,e}$  σαν συνάρτηση του  $t_{max}$  (βλ. σχήμα 4.8)
8. Αναπαράγετε τα σχήματα 4.11 και 4.12. Επαναλάβετε το ίδιο και για την ενέργεια. Στη συνέχεια κάντε τα ανάλογα σχήματα για τους  $\tau_{int,m}$  και  $\tau_{int,e}$  σαν συνάρτηση του  $t_{max}$  (βλ. σχήμα 4.8)
9. Κάνετε τις απαραίτητες μετατροπές στον κώδικα ώστε να προσομοιώσετε το πρότυπο Ising παρουσία εξωτερικού μαγνητικού πεδίου  $B$  (βλ. εξ. (1.51)). Υπολογίστε την μαγνητισμό  $m(\beta, B)$  για  $L = 32$  και  $\beta = 0.2, 0.4, 0.6, 0.8, 1.0$  στο ενδιαφέρον φάσμα των θερμοκρασιών. Παρατηρήστε της διαδικασίας εύρεσης της θερμικής ισορροπίας καθώς αυξάνει το  $B$  από διαφορετικές αρχικές καταστάσεις: Παγωμένη με σπιν παράλληλα στο  $B$ , παγωμένη με σπιν αντιπαράλληλα στο  $B$ , καυτή και την προηγούμενη κατάσταση σε γειτονική θερμοκρασία. Μελετήστε πως μεταβάλλεται η θερμοκρασία στην οποία έχουμε μετάβαση από μη μαγνητισμένη σε μαγνητισμένη κατάσταση καθώς αυξάνει το  $B$ .

10. Φαινόμενο υστέρησης: Στην προηγούμενη άσκηση το σύστημα για  $B \neq 0$  παρουσιάζει μετάβαση φάση πρώτης τάξης, δηλ. ασυνέχεια στην τιμή της παραμέτρου τάξης. Εδώ αυτή είναι η μαγνήτιση σαν συνάρτηση του  $B$ . Παρουσιάζεται στο σύστημα αυτό το φαινόμενο της υστέρησης. Για να το δείτε θέστε  $L = 32$  και  $\beta = 0.55$ .

(α') Φέρτε το σύστημα σε θερμική ισορροπία για  $B = 0$ .

(β') Προσομοιώστε το σύστημα για  $B = 0.2$  χρησιμοποιώντας ως αρχική κατάσταση αυτή του προηγούμενου βήματος. Κάντε 100 sweeps και υπολογίστε την  $\langle m \rangle$ .

(γ') Συνεχίστε την ίδια διαδικασία αυξάνοντας το μαγνητικό πεδίο κατά  $\delta B = 0.2$  κάθε φορά. Σταματήστε όταν  $\langle m \rangle \approx 0.95$ .

(δ') Από την τελευταία διάταξη του προηγούμενου βήματος αρχίστε να ελαττώνετε διαδοχικά το μαγν. πεδίο κατά  $\delta B = -0.2$  μέχρι  $\langle m \rangle \approx -0.95$ .

(ε') Επαναλάβετε ξανά από την τελευταία διάταξη του συστήματος αυξάνοντας το μαγν. πεδίο ξανά κατά  $\delta B = 0.2$  μέχρι  $\langle m \rangle \approx 0.95$ .

Τι παρατηρείτε? Φτιάξτε μια κλειστή καμπύλη στο επίπεδο  $(B, m)$ ? Ποιες είναι οι τιμές της μαγνήτισης για  $B = 0$ ? Γιατί είναι διαφορετικές?

Τα συστήματα που παρουσιάζουν μετάβαση πρώτης τάξης, παρουσιάζουν καταστάσεις που είναι τοπικά ελάχιστα της ελεύθερης ενέργειας αλλά υπάρχει μόνο ένα πραγματικό ελάχιστο. Αυτό φαίνεται στο σχήμα 1.2 όπου απεικονίζονται οι δύο καταστάσεις να έχουν την ίδια ελεύθερη ενέργεια. Αυτό συμβαίνει ακριβώς στο κρίσιμο σημείο. Όταν φύγουμε από αυτό, τότε το ένα ελάχιστο είναι το πραγματικό, ενώ το άλλο έχει μεγαλύτερη ελεύθερη ενέργεια. Όταν το σύστημα βρεθεί σε αυτή την κατάσταση, τότε χρειάζεται μεγάλο χρόνο για να μπορέσει να υπερπηδήσει το φράγμα ελεύθερης ενέργειας και να βρει την πραγματική θεμελιώδη κατάσταση. Μία τέτοια κατάσταση ονομάζεται μετασταθής. Σε μια προσομοίωση Μόντε Κάρλο μια τέτοια περίπτωση παρουσιάζει μεγάλη δυσκολία, αφού μπορούμε να θεωρήσουμε κατά λάθος ότι βρισκόμαστε στη θεμελιώδη κατάσταση. Εσείς επαναλάβετε τις προσομοιώσεις που κάνατε παραπάνω, αλλά κάντε αυτή τη φορά 100,000 sweeps. Παρατηρήστε τη χρονοσειρά της μαγνήτισης και παρατηρήστε τις μεταπτώσεις από τη μια μετασταθή κατάσταση στην άλλη. Κάνετε το ιστόγραμμα

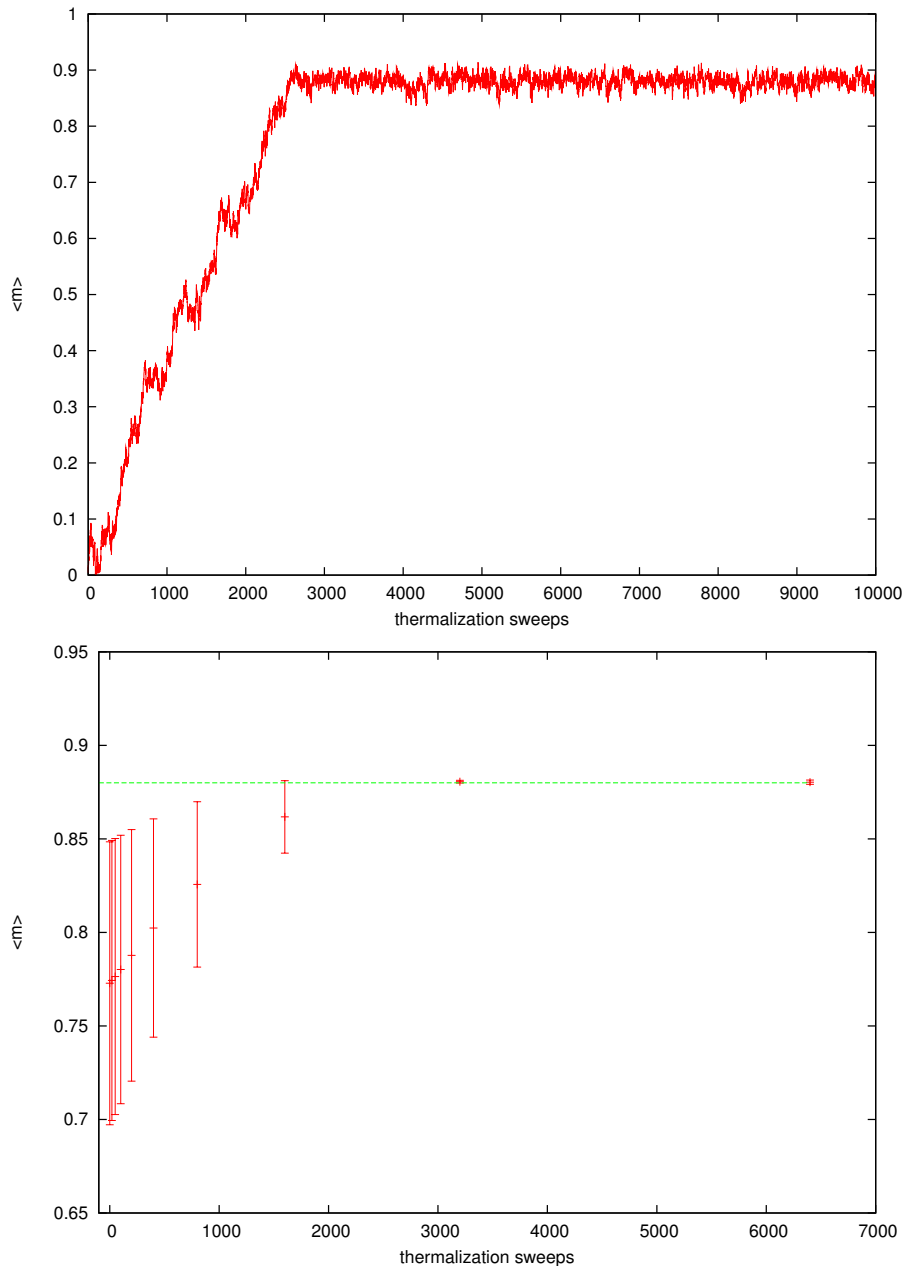
των τιμών της μαγνήτισης και εντοπίστε τη μετασταθή κατάσταση. Πώς μεταβάλλεται το ιστόγραμμα καθώς αυξάνεται το  $B$ ?

11. Γράψτε κώδικα για το πρότυπο Ising στις 2 διαστάσεις αλλά σε ένα πλέγμα τριγωνικό. Η κύρια διαφορά είναι ότι ο αριθμός των πλησιεστέρων γειτόνων είναι  $z = 6$  αντί για  $z = 4$ . Για τη δομή του πλέγματος θα βρείτε χρήσιμη πληροφορία στο κεφ. 13.1.2 των Newman και Barkema (ειδικά από το Σχήμα 13.4). Υπολογίστε ξανά τη μεταβολή της ενέργειας σε κάθε spin flip για το βήμα Metropolis. Από τα μέγιστα της μαγνητικής επιδεκτικότητας και της ειδικής θερμότητας εξετάστε αν η κρίσιμη θερμοκρασία του συστήματος είναι πράγματι  $\beta_c \approx 0.274653072$ . Παρατηρήστε ότι ενώ η τιμή της  $\beta_c$  αλλάζει σε σχέση με το τετραγωνικό πλέγμα, οι κρίσιμοι εκθέτες παραμένουν οι ίδιοι (παγκοσμιότητα).
12. Γράψτε κώδικα για το πρότυπο Ising στις τρεις διαστάσεις πάνω σε κυβικό πλέγμα. Χρησιμοποιήστε ελικοειδείς συνοριακές συνθήκες και τότε θα χρειαστούν πολύ λίγες μετατροπές στον κώδικα (μαζί με τα  $XNN=1$  και  $YNN=L$  θα έχετε και  $ZNN=L*L$ ).
13. Προσομοιώστε το αντισιδηρομαγνητικό πρότυπο Ising. Στον κώδικα που έχετε αρκεί να πάρετε αρνητικές θερμοκρασίες. Ποια είναι η θεμελιώδης κατάσταση?

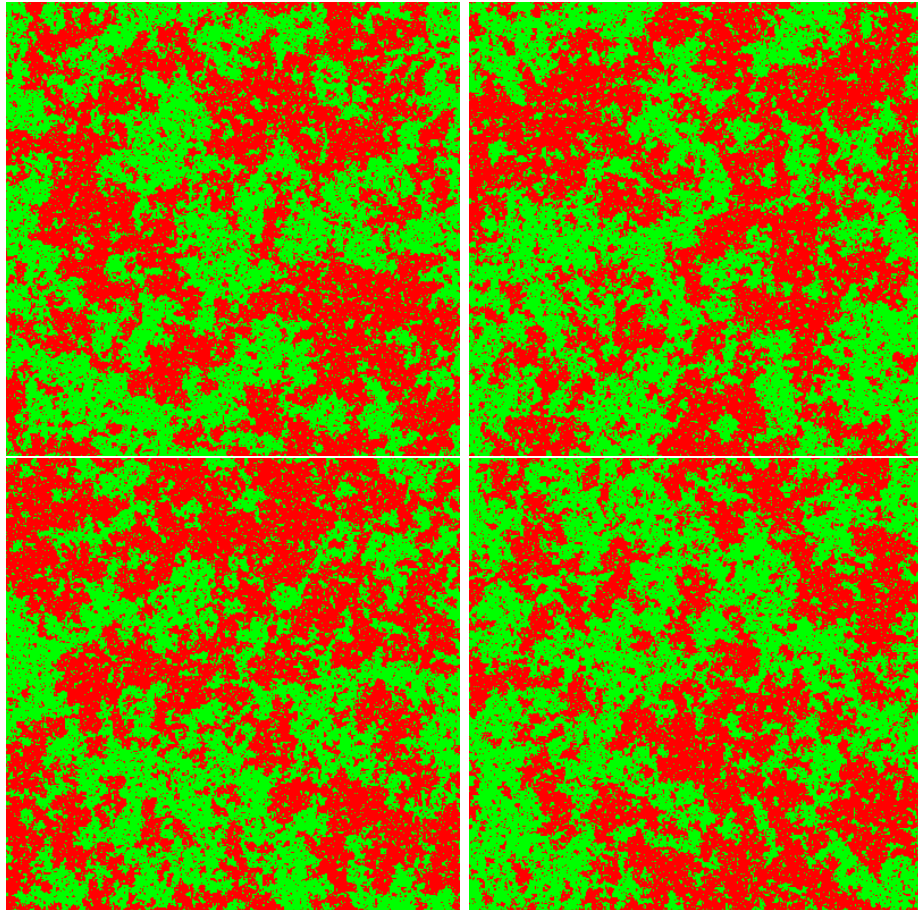
Ορίστε την staggered magnetization  $m_s$  να είναι η μαγνήτιση ανά πλεγματική θέση ενός υποπλέγματος το οποίο περιέχει τα πλεγματικά σημεία με περιττή  $x$  και  $y$  συντεταγμένη. Θέστε  $L = 32$  και υπολογίστε την εξάρτηση της ενέργειας,  $m_s$ , ειδικής θερμότητας και μαγνητικής επιδεκτικότητας  $\chi$  και της staggered μαγνητικής επιδεκτικότητας  $\chi_s = \beta N/4 \langle (m_s - \langle m_s \rangle)^2 \rangle$ .

Η  $\chi$  παρουσιάζει ένα μέγιστο στην περιοχή  $\beta \approx 0.4407$ . Υπολογίστε την τιμή της στη θερμοκρασία αυτή για  $L = 32 - 120$ . Δείξτε ότι η τιμή αυτή δεν αποκλίνει για  $L \rightarrow \infty$ , άρα δεν έχουμε ένδειξη μετάβασης φάσης από την  $\chi$ .

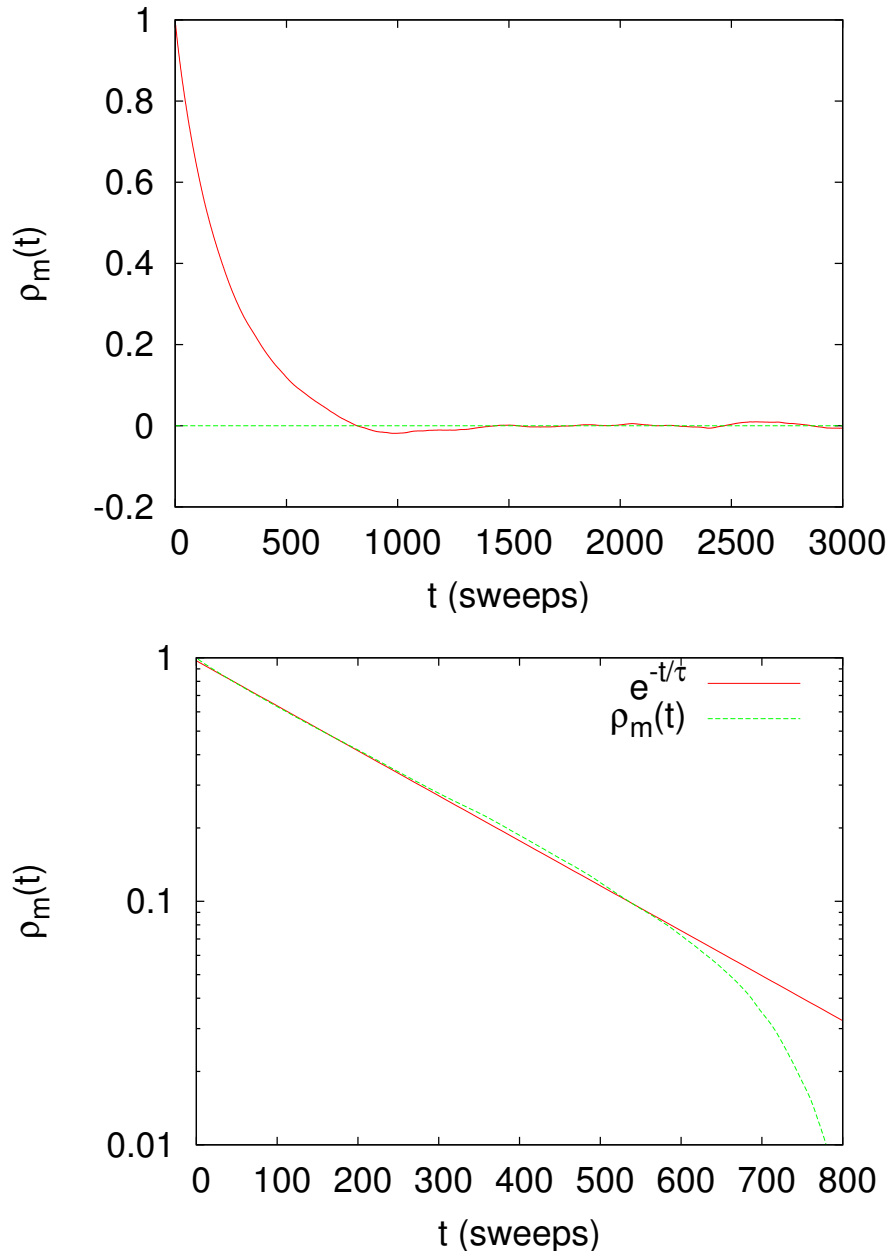
Επαναλάβετε για την  $\chi_s$ . Τι συμπεραίνετε? Συγκρίνετε τη συμπεριφορά της  $\langle m_s \rangle$  για το αντισιδηρομαγνητικό πρότυπο Ising με την  $\langle m \rangle$  του σιδηρομαγνητικού.



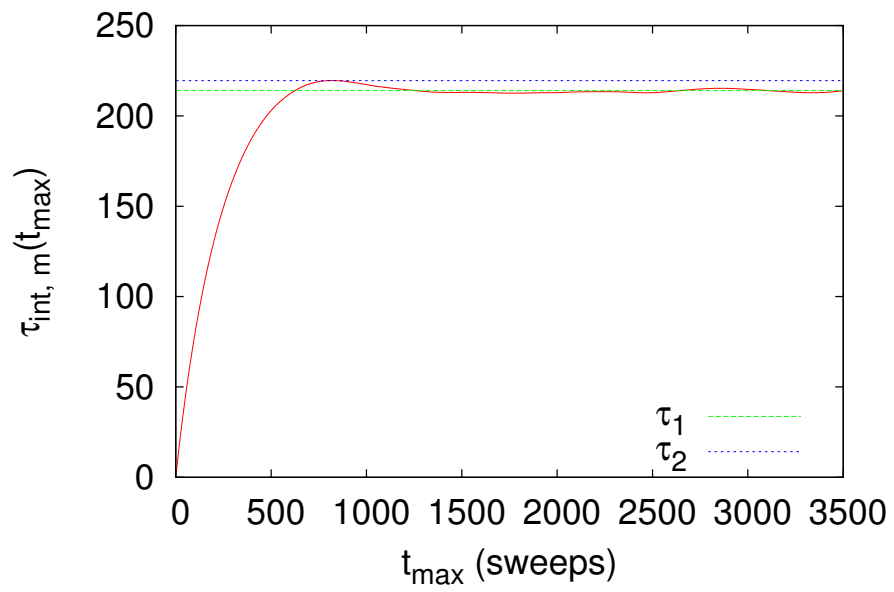
Σχήμα 4.5: Μαγνήτιση ανά πλεγματική θέση για το πρότυπο Ising με  $L = 100$  και  $\beta = 0.48$ . Φαίνεται η διαδικασία εύρεσης θερμικής ισορροπίας για μια αρχική κατάσταση καυτή. Στη συνέχεια, στο κάτω σχήμα, υπολογίζουμε τη μέση τιμή  $\langle m \rangle$  αποκόπτοντας “thermalization sweeps” μετρήσεις και τις δείχνουμε στο δεύτερο σχήμα μαζί με τα αντίστοιχα σφάλματα. Παρατηρούμε ότι όταν αποκόψουμε ικανό αριθμό από αρχικές μετρήσεις, σε συμφωνία με το πάνω σχήμα, το αποτέλεσμα σταθεροποιείται ( $\langle m \rangle = 0.880(1)$ ) και βρίσκουμε πως το σύστημα είναι σε θερμική ισορροπία.



Σχήμα 4.6: Διατάξεις spin για πρότυπο Ising με  $L = 400$ ,  $\beta = 0.4292$  μετά από 4000, 9000, 12000 και 45000 sweeps αντίστοιχα Παρατηρούμε τη δημιουργία μεγάλων clusters από όμοια spin τα οποία είναι δύσκολο να καταστρέψουμε και να δημιουργήσουμε αλλού με έναν single-flip αλγόριθμο όπως ο Metropolis. Για το λόγο αυτό οι χρόνοι αυτοσυσχετισμού είναι μεγάλοι.

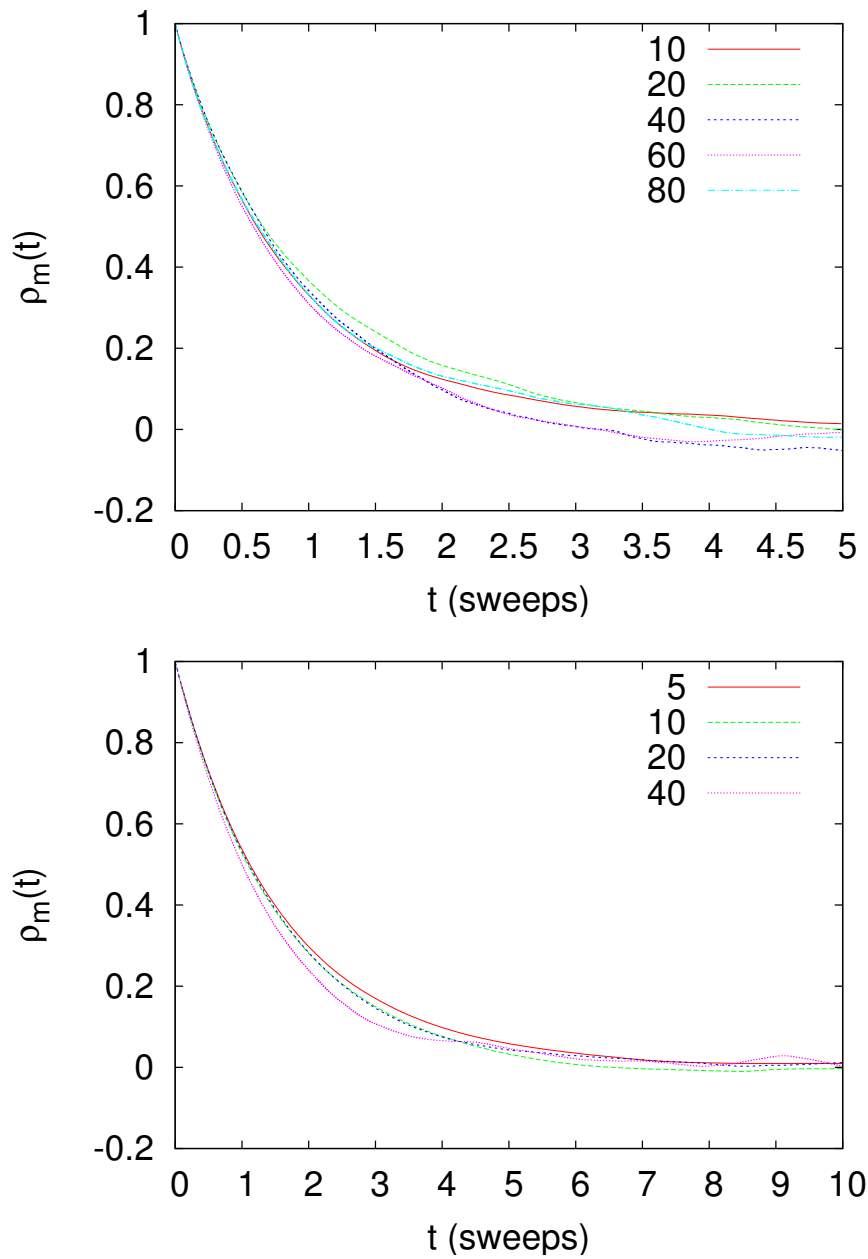


Σχήμα 4.7: Η συνάρτηση αυτοσυσχετισμού της μαγνήτισης  $\rho_m(t)$  για το πρότυπο Ising και  $L = 100$ ,  $\beta = 0.42$ . Φαίνεται η εκθετική πτώση της και ότι  $\tau_m \approx 200$  sweeps. Φαίνονται και τα φαινόμενα πεπερασμένου μεγέθους το δείγματος (1,000,000 μετρήσεις) όταν η  $\rho$  αρχίζει να έχει διακυμάνσεις γύρω από το 0. Στο δεύτερο σχήμα γίνεται προσαρμογή στην εκθετική πτώση  $Ce^{-t/\tau}$  (βλ. εξ. (4.23)) με  $\tau = 235(3)$  sweeps.

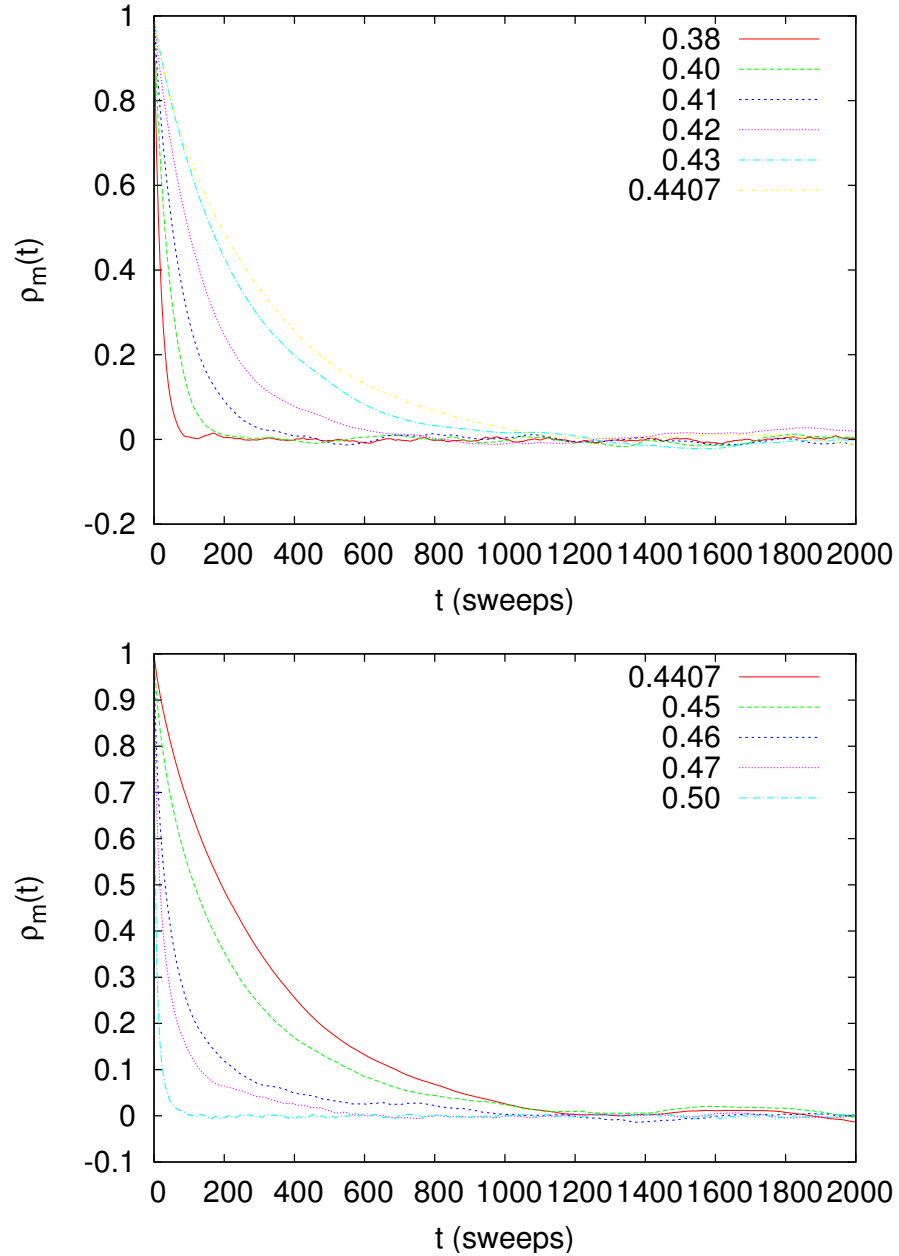


Σχήμα 4.8: Υπολογισμός του ολοκληρωμένου χρόνου αυτοσυσχετισμού της μαγνήτισης (integrated autocorrelation time) για τα δεδομένα του σχήματος 4.7. Παρατηρούμε ένα πλατό στις τιμές του  $\tau_{int, m}$  για  $\tau_1 = 214(1)$ sweeps αλλά και ένα μέγιστο για  $\tau_2 \approx 219.5$ sweeps. Η πτώση των τιμών από  $\tau_1$  σε  $\tau_2$  οφείλεται σε αρνητικές τιμές της  $\rho_m(t)$  λόγω θορύβου του δείγματος των μετρήσεων. Άρα θα πάρουμε  $\tau_{int, m} = 217(3)$ sweeps να είναι το αποτέλεσμά μας.

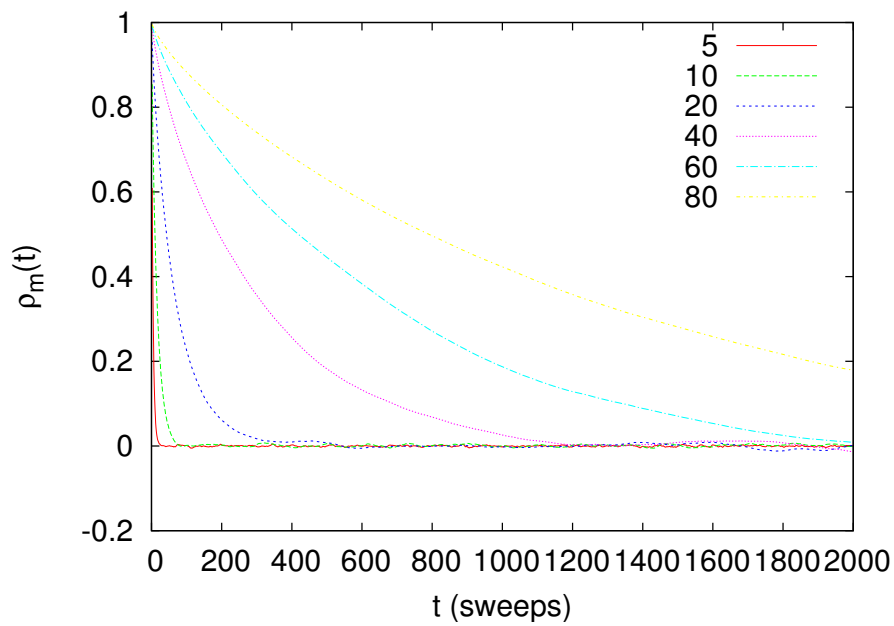




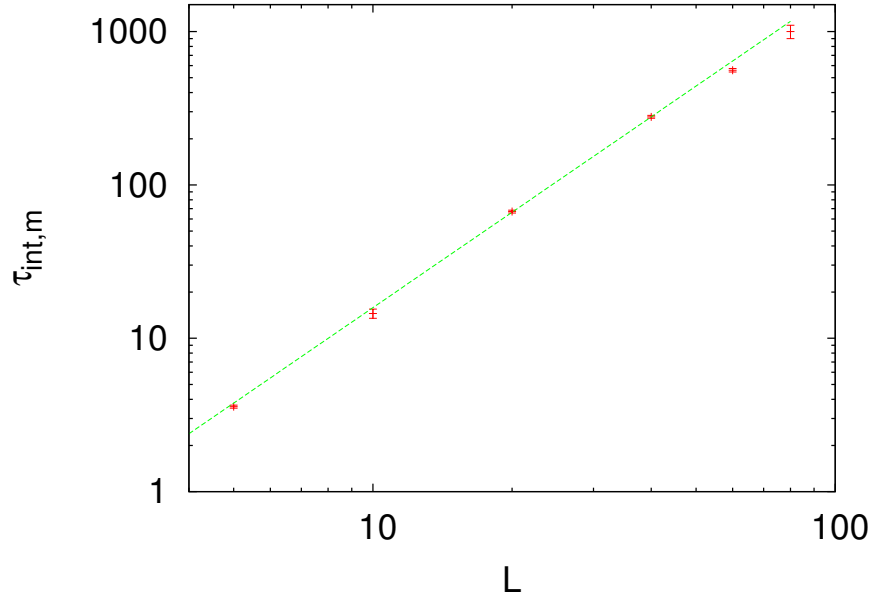
Σχήμα 4.9: Η συνάρτηση αυτοσυσχετισμού της μαγνήτισης για το πρότυπο Ising σε υψηλή θερμοκρασία  $\beta = 0.20$  για  $L = 10, 20, 40, 60, 80$  (επάνω σχήμα) και σε χαμηλή θερμοκρασία  $\beta = 0.65$  για  $L = 5, 10, 20, 40$  (κάτω σχήμα). Ο χρόνος αυτοσυσχετισμού σε sweeps είναι ανεξάρτητος του  $L$ .



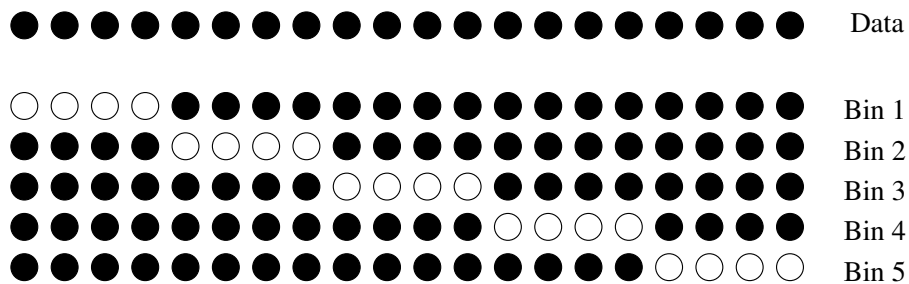
Σχήμα 4.10: Η συνάρτηση αυτοσυσχετισμού της μαγνήτισης για το πρότυπο Ising για  $L = 40$ . Στο επάνω σχήμα φαίνεται πώς αυξάνει ο χρόνος αυτοσυσχετισμού όταν πλησιάζουμε την κρίσιμη θερμοκρασία από τη θερμή φάση (φάση αταξίας) ενώ στο κάτω πώς αυξάνει ενώ πλησιάζουμε από την ψυχρή φάση (φάση τάξης).



Σχήμα 4.11: Η συνάρτηση αυτοσυσχετισμού της μαγνήτισης για το πρότυπο Ising για  $\beta = 0.4407 \approx \beta_c$  για διαφορετικά  $L$ . Παρατηρούμε την μεγάλη αύξηση του χρόνου αυτοσυσχετισμού με το μέγεθος του συστήματος στην κρίσιμη περιοχή.



Σχήμα 4.12: Ο ολοκληρωμένος χρόνος αυτοσυσχετισμού  $\tau_{\text{int},m}$  για  $\beta = \beta_c$  σε διάγραμμα λογαριθμικής κλίμακας. Η συνεχής καμπύλη είναι προσαρμογή στη συνάρτηση  $0.136(10)L^{2.067(21)}$ . Η αναμενόμενη τιμή του εκθέτη είναι  $2.1665(12)$  και η διαφορά οφείλεται στα σχετικά μικρά μεγέθη του συστήματος.



Σχήμα 4.13: Η μέθοδος jackknife για ένα δείγμα από  $n = 20$  μετρήσεις. Χωρίζουμε τα δεδομένα σε  $n_b = 5$  bins το οποίο το καθένα έχει  $b = n - (n/n_b) = 20 - 4 = 16$  μετρήσεις (οι γεμάτοι δίσκοι). Σε κάθε δείγμα υπολογίζουμε την μέση τιμή  $\mathcal{O}_i^b$  και από αυτές το σφάλμα  $\delta\mathcal{O} = \sqrt{n_b (\langle (\mathcal{O}^b)^2 \rangle - \langle \mathcal{O}^b \rangle^2)}$ .

## ΚΕΦΑΛΑΙΟ 5

# CLUSTER ALGORITHMS

Ο αλγόριθμος Metropolis που αναπτύχθηκε στο προηγούμενο κεφάλαιο πάσχει από κρίσιμη επιβράδυνση (critical slowing down) στην κρίσιμη περιοχή αλλαγής φάσης. Στο κεφάλαιο αυτό θα αναλύσουμε τους λόγους εμφάνισης και τις δυσκολίες που παρουσιάζει το φαινόμενο αυτό στις προσομοιώσεις Monte Carlo. Θα παρουσιάσουμε άλλους αλγόριθμους (ειδικά τον αλγόριθμο του Wolff) που σχεδόν εξαλείφουν το φαινόμενο. Οι αλγόριθμοι αυτοί βασίζονται στη μερική κατανόηση της δυναμικής των μοντέλων που μελετούμε και έχουν ειδική εφαρμογή σε αντίθεση με τον αλγόριθμο Metropolis που εφαρμόζεται σε όλα τα συστήματα στατιστικής φυσικής και έχει γενική εφαρμογή.

### 5.1 ΚΡΙΣΙΜΗ ΕΠΙΒΡΑΔΥΝΣΗ

Όπως συζητήσαμε στην ενότητα 4.4 η προσομοίωση του πρότυπου Ising στην κρίσιμη περιοχή αλλαγής φάσης με τον αλγόριθμο Metropolis παρουσιάζει μεγάλη αύξηση του χρόνου αυτοσυσχετισμού που δίνεται από τη σχέση βάθμισης

$$\tau \sim \xi^z \quad (5.1)$$

Στην ψευδοκρίσιμη περιοχή το μήκος συσχετισμού γίνεται  $\xi \sim L$  και παίρνουμε τη σχέση (4.27)  $\tau \sim L^z$ . Όταν  $z > 0$  έχουμε το φαινόμενο της κρίσιμης επιβράδυνσης στην προσομοίωση.

Εδώ αξίζει να αναφέρουμε ότι το φαινόμενο αυτό είναι που ουσιαστικά περιορίζει τη δυνατότητα που έχουμε να προσομοιώσουμε συστήματα με πολύ μεγάλο  $L$  από την άποψη του διαθέσιμου  $t_{\text{CPU}}$ <sup>1</sup>. Όταν μετράμε μια τοπική ποσότητα, όπως η μαγνή-

<sup>1</sup>Φυσικά το μέγεθος της μνήμης είναι ο άλλος σημαντικός παράγοντας που θα βάλει φραγμό στις προσομοιώσιμες τιμές του  $L$ .

τιση ανά πλεγματική θέση  $\langle m \rangle$ , η αύξηση του  $L$  αυξάνει το χρόνο<sup>2</sup>  $t_{\text{CPU}} \sim L^d$ , αλλά ταυτόχρονα οι μετρήσεις που παίρνουμε για την  $\langle m \rangle$  αυξάνει με τον ίδιο ρυθμό. Άρα αν σε κάθε sweep του πλέγματος παίρναμε μία στατιστικά ανεξάρτητη διάταξη των σπιν τότε δε θα είχαμε επί πλέον κόστος για μια δεδομένης ακρίβειας μέτρηση αφού με την αύξηση του  $L$  το αυξημένο κόστος παραγωγής της διάταξης θα αντισταθμιζόταν από τα περισσότερα σπιν πάνω στα οποία θα μετρήσουμε την  $\langle m \rangle$  και θα είχαμε για το χρόνο συλλογής δεδομένης στατιστικής για τη μαγνήτιση ανά πλεγματική θέση  $t_{\text{CPU}}^{(m)} \sim L^0 = 1$ . Οπότε θα μας συνέφερε η μέτρηση στο μεγαλύτερο δυνατόν  $L$  ώστε να ελαχιστοποιήσουμε τα φαινόμενα επίδρασης του πεπερασμένου μεγέθους του πλέγματος.

Η παρουσία κρίσιμης επιβράδυνσης όμως προσθέτει το κόστος παραγωγής ανεξάρτητων μετρήσεων και παίρνουμε  $t_{\text{CPU}}^{(m)} \sim L^z$  με αποτέλεσμα για κάποια τιμή του  $L$  να γίνεται απαγορευτικά ακριβό να προσομοιώσουμε μεγαλύτερο σύστημα. Στη συγκεκριμένη περίπτωση που μελετάμε, το πρότυπο Ising στο επίπεδο με τον αλγόριθμο Metropolis  $z \approx 2.17$  και το πρόβλημα αυτό παρουσιάζεται πολύ γρήγορα όπως είδαμε στο προηγούμενο κεφάλαιο. Είναι ιδιαίτερα ενδιαφέρον να αναζητηθούν αλγόριθμοι που να ελαχιστοποιούν το φαινόμενο και στην περίπτωση του πρότυπου Ising αλλά και γενικότερων μοντέλων με σπιν η λύση είναι αρκετά απλή. Όπως θα δούμε όμως, η λύση προκύπτει από τη βαθύτερη κατανόηση της δυναμικής που προκαλεί το φαινόμενο της κρίσιμης επιβράδυνσης και αφορά μόνο τα πρότυπα εκείνα με παρόμοια δυναμική δομή<sup>3</sup>.

Ο λόγος που παρουσιάζεται το φαινόμενο της κρίσιμης επιβράδυνσης είναι η αύξηση του μήκους συσχετισμού  $\xi$  στην κρίσιμη περιοχή. Καθώς πλησιάζουμε την κρίσιμη θερμοκρασία  $\beta \rightarrow \beta_c^-$  από τη φάση αταξίας των σπιν, δημιουργούνται στο πλέγμα μεγάλες περιοχές όπου τα σπιν δείχνουν προς την ίδια κατεύθυνση. Η επικράτηση των περιοχών αυτών είναι που τελικά θα οδηγήσει το σύστημα στη φάση τάξης με μη μηδενική μαγνήτιση ανά πλεγματική θέση στο θερμοδυναμικό όριο. Ο αλγόριθμος Metropolis επιχειρεί την αλλαγή των σπιν ένα-ένα κάτι το οποίο είναι δύσκολο ειδικά για σπιν μέσα στις περιοχές αυτές. Για ένα σπιν με τέσσερις γείτονες με σπιν στην ίδια κατεύθυνση με αυτό, η αλλαγή γίνεται δεκτή με πιθανότητα  $\sim e^{-8\beta_c} \approx 0.029$  η οποία είναι πολύ μικρή. Προκύπτει ότι τα περισσότερα σπιν στα οποία η αλλαγή γίνεται

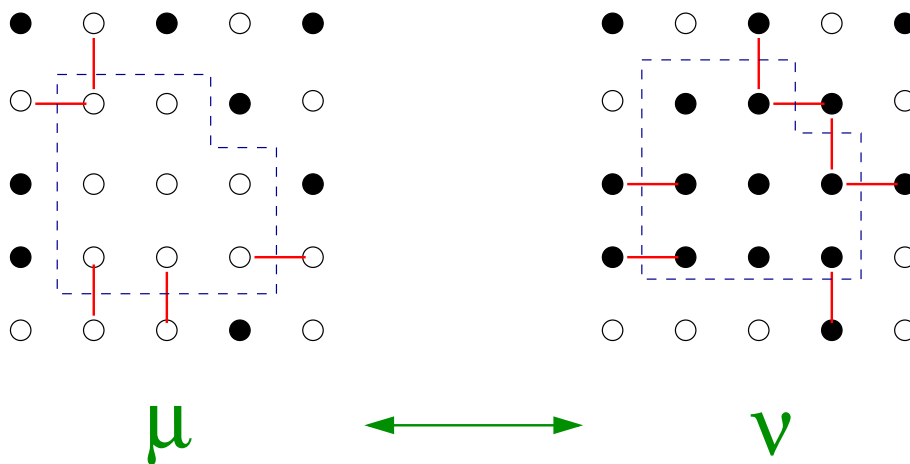
<sup>2</sup> $d = 2$  στην περίπτωση του πρότυπου Ising στο επίπεδο, αλλά το επιχείρημα ισχύει και για περισσότερες διαστάσεις  $d > 2$ .

<sup>3</sup>Ενώ αντίθετα ο αλγόριθμος Metropolis έχει γενική εφαρμογή στα περισσότερα συστήματα της στατιστικής φυσικής.

δεκτή βρίσκονται στα όρια των περιοχών αυτών και η αλλαγή από μία διάταξη σε μια άλλη όπου οι περιοχές αυτές βρίσκονται σε ανεξάρτητες θέσεις και έχουν ανεξάρτητα σχήματα απαιτεί μεγάλο χρόνο: Μια περιοχή δημιουργείται/καταστρέφεται (στοχαστικά φυσικά) μετακινώντας το σύνορό της.

## 5.2 Ο ΑΛΓΟΡΙΘΜΟΣ ΤΟΥ WOLFF

Η ιδέα για τη βελτίωση του φαινομένου της κρίσιμης επιβράδυνσης είναι να επινοηθούν αλγόριθμοι όπου θα μπορεί κανείς σε ένα βήμα να αλλάξει την τιμή των σπιν σε περιοχές συγκρίσιμες με τις μεγάλες περιοχές όμοιων σπιν. Οι περιοχές αυτές από κατάλληλα επιλεγμένα όμοια σπιν ονομάζονται τα clusters του αλγόριθμου. Το πρώτο βήμα έγινε με τον αλγόριθμο Swendsen-Wang [23] ενώ αργότερα παρουσιάστηκε και ο αλγόριθμος του Uli Wolff [24].



Σχήμα 5.1: Οι δύο διατάξεις των σπιν που προκύπτουν από την αλλαγή των σπιν ενός Wolff cluster. Φαίνονται οι δεσμοί που δημιουργούνται/καταστρέφονται κατά τη μετάβαση οι οποίοι βρίσκονται στο σύνορο του cluster.

Από την ιδέα που παρουσιάσαμε προκύπτει ότι η διαδικασία κατασκευής των clusters πρέπει να είναι στοχαστική και να εξαρτάται από τη θερμοκρασία. Σε ψηλές θερμοκρασίες  $\beta \ll \beta_c$  πρέπει να ευνοείται η κατασκευή μικρών clusters με μέγεθος ίσο με μερικές πλεγματικές σταθερές ενώ σε χαμηλές θερμοκρασίες  $\beta \gg \beta_c$  να ευνοείται η κατασκευή μεγάλων clusters με μέγεθος  $\sim L$ .

Η βασική ιδέα στον αλγόριθμο του Wolff είναι να επιλέξουμε ένα σπιν-“σπόρο” (γεννήτορα, seed) και να κατασκευάσουμε ένα

cluster γύρω από αυτόν. Σε κάθε βήμα προσθέτουμε μέλη στο cluster με πιθανότητα  $P_{\text{add}} = P_{\text{add}}(\beta)$ . Με κατάλληλα επιλεγμένη  $P_{\text{add}}(\beta)$  το cluster θα αναπτύσσεται με τον αναμενόμενο τρόπο και θα ικανοποιείται η συνθήκη λεπτομερούς ισοζύγησης (2.12). Η σχέση φαίνεται γραφικά στο Σχήμα 5.1. Στην κατάσταση  $\mu$  το cluster περιγράφεται από τη διακεκομμένη γραμμή. Η νέα κατάσταση  $\nu$  είναι αυτή όπου όλα τα σπιν του cluster αντιστρέφονται, ενώ όλα τα σπιν στις πλεγματικές θέσεις εκτός του cluster μένουν τα ίδια.

Η κατάλληλη επιλογή της  $P_{\text{add}}$  θα μας δώσει τη σχέση (2.13)

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)}. \quad (5.2)$$

Η κρίσιμη παρατήρηση είναι πως η μεταβολή της ενέργειας το δεξί μέλος της (5.2) οφείλεται στο σπάσιμο/δημιουργία δεσμών στο *σύνоро* του cluster αφού στο εσωτερικό του δεν μεταβάλλονται οι δεσμοί από όμοια σπιν ενώ εκτός του cluster δεν επέρχεται καμιά μεταβολή. Αυτό μπορείτε να το δείτε στο απλό παράδειγμα του σχήματος 5.1. Με κατάλληλη επιλογή πιθανότητας επιλογής  $g(\mu \rightarrow \nu)$  της κατάστασης  $\nu$  και του λόγου αποδοχής  $A(\mu \rightarrow \nu)$  έτσι ώστε

$$P(\mu \rightarrow \nu) = g(\mu \rightarrow \nu) A(\mu \rightarrow \nu), \quad (5.3)$$

θα πετύχουμε να ισχύει η (5.2) και να μεγιστοποιηθούν οι λόγοι αποδοχής. Μάλιστα στην περίπτωσή μας θα βρούμε  $A(\mu \rightarrow \nu) = 1!$

Η πιθανότητα επιλογής  $g(\mu \rightarrow \nu)$  είναι η πιθανότητα να κατασκευάσουμε το συγκεκριμένο cluster και σπάει σε 3 όρους από ανεξάρτητες επιλογές:

$$g(\mu \rightarrow \nu) = p_{\text{seed}} \times p_{\text{yes}}^{\text{int}} \times p_{\text{no}}^{\text{border}} \quad (5.4)$$

Ο πρώτος όρος είναι η πιθανότητα να ξεκινήσουμε το cluster από το συγκεκριμένο seed. Επιλέγοντας με ομοιόμορφη πιθανότητα μια πλεγματική θέση έχουμε

$$p_{\text{seed}} = \frac{1}{N}. \quad (5.5)$$

Στη συνέχεια το cluster αναπτύσσεται γύρω από το γεννήτορά του. Ο δεύτερος όρος  $p_{\text{yes}}^{\text{int}}$  είναι η πιθανότητα να ενσωματώσουμε όλα τα μέλη του cluster, δηλ. η πιθανότητα να πούμε “ναι” όταν με πιθανότητα  $P_{\text{add}}$  τα προσθέτουμε από το γείτονά τους που βρίσκεται ήδη στο cluster. Η πιθανότητα αυτή είναι πολύπλοκη σχέση του μεγέθους και σχήματος του cluster αλλά ευτυχώς δεν είναι



σημαντική στον υπολογισμό μας. Ο λόγος είναι ότι κατά τη μετάβαση  $\nu \rightarrow \mu$  ο αντίστοιχος όρος είναι ακριβώς ο ίδιος αφού τα δύο cluster είναι ακριβώς τα ίδια (μόνο έχουν αντίθετα σπιν)!

$$p_{\text{yes}}^{\text{int}}(\mu \rightarrow \nu) = p_{\text{yes}}^{\text{int}}(\nu \rightarrow \mu) \equiv C_{\mu\nu}. \quad (5.6)$$

Ο τρίτος όρος είναι ο πιο ενδιαφέρων. Το cluster σταματάει την ανάπτυξη του όταν στις θέσεις πάνω στο σύνορο του πούμε “όχι” στην ενσωμάτωση των γειτονικών όμοιων σπιν (τα αντίθετα σπιν προφανώς δεν ενσωματώνονται). Αν  $P_{\text{add}}$  είναι η πιθανότητα να πούμε “ναι” στην ενσωμάτωση, η πιθανότητα να πούμε “όχι” είναι  $1 - P_{\text{add}}$ . Ας υποθέσουμε πως τα γειτονικά με το cluster όμοια σπιν στην κατάσταση  $\mu$  είναι  $m$  ενώ στην κατάσταση  $\nu$  είναι  $n$ . Λ.χ. στο σχήμα 5.1  $m = 5$  και  $n = 7$ . Άρα η πιθανότητα να σταματήσουμε το cluster της κατάστασης  $\mu$  στο σύνορό που φαίνεται στο σχήμα είναι να πούμε “όχι”  $m$  φορές, κάτι που γίνεται με πιθανότητα  $(1 - P_{\text{add}})^m$ :

$$p_{\text{no}}^{\text{border}}(\mu \rightarrow \nu) = (1 - P_{\text{add}})^m. \quad (5.7)$$

Ανάλογα, το cluster στην κατάσταση  $\nu$  σταματάει στο ίδιο σύνορο με πιθανότητα

$$p_{\text{no}}^{\text{border}}(\nu \rightarrow \mu) = (1 - P_{\text{add}})^n. \quad (5.8)$$

Τελικά παίρνουμε

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{\frac{1}{N} C_{\mu\nu} (1 - P_{\text{add}})^m A(\mu \rightarrow \nu)}{\frac{1}{N} C_{\mu\nu} (1 - P_{\text{add}})^n A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)}. \quad (5.9)$$

Το δεξί μέλος της παραπάνω εξίσωσης εξαρτάται πάλι μόνο από τους δεσμούς στο όριο του cluster. Οι εσωτερικοί δεσμοί δε συνεισφέρουν, οπότε η μεταβολή στην ενέργεια οφείλεται μόνο στη δημιουργία/καταστροφή των δεσμών στο όριο. Κάθε δεσμός που δημιουργείται στο όριο κατά τη μετάβαση  $\mu \rightarrow \nu$  μειώνει την ενέργεια κατά 2 ενώ κάθε δεσμός που σπάει την αυξάνει κατά 2:

$$E_\nu - E_\mu = (-2n) - (-2m) = 2(m - n). \quad (5.10)$$

Τελικά παίρνουμε

$$(1 - P_{\text{add}})^{m-n} \frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = e^{-2\beta(\mu - \nu)} \Rightarrow \frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = \left[ e^{2\beta} (1 - P_{\text{add}}) \right]^{n-m}. \quad (5.11)$$

Από την παραπάνω σχέση βλέπουμε ότι αν επιλέξουμε

$$1 - P_{\text{add}} = e^{-2\beta} \Rightarrow P_{\text{add}} = 1 - e^{-2\beta}, \quad (5.12)$$

μπορούμε επίσης να επιλέξουμε

$$A(\mu \rightarrow \nu) = A(\nu \rightarrow \mu) = 1! \quad (5.13)$$

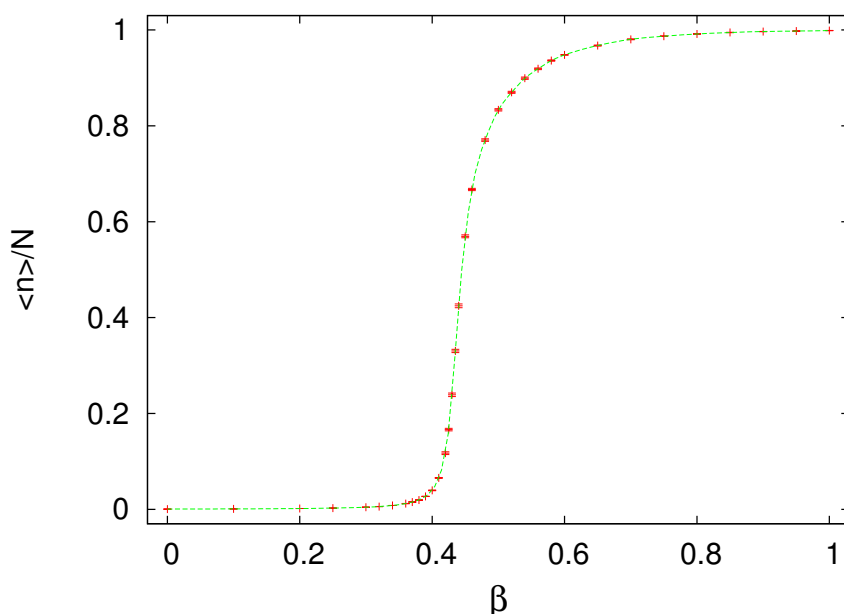
Άρα μπορούμε να έχουμε τη συνθήκη (5.2) να ισχύει φτιάχνοντας το cluster με την  $P_{\text{add}}$  της (5.12), αλλάζοντας τα σπιν και κάνοντας πάντα τη νέα κατάσταση που προκύπτει δεκτή στο επόμενο βήμα του Μόντε Κάρλο.

Συνοψίζοντας, ένα βήμα του αλγόριθμου Wolff γίνεται ως εξής:

1. Επιλέγουμε με ομοιόμορφη πιθανότητα  $p_{\text{seed}} = \frac{1}{N}$  μία πλεγματική θέση που θα είναι ο γεννήτορας (seed) του cluster. Είναι το πρώτο νέο μέλος του cluster.
2. Επαγωγικά: Για κάθε νέο μέλος του cluster επισκεπτόμαστε τους πλησιέστερους γείτονές του που δεν ανήκουν ήδη στο cluster. Αν το σπιν τους είναι ίδιο με αυτό των μελών του cluster τους προσθέτουμε στα “νέα μέλη” με πιθανότητα  $P_{\text{add}}$ . Το αρχικό σπιν είναι πια “παλιό μέλος”.
3. Όταν τελειώσουν τα “νέα μέλη” η διαδικασία ανάπτυξης του cluster σταματάει.
4. Αλλάζουμε την τιμή του σπιν κάθε μέλους του cluster.

Ο αλγόριθμος είναι εργοδικός αφού από μια οποιαδήποτε κατάσταση μπορούμε να πάμε σε μία άλλη με μια σειρά από clusters μεγέθους 1 (ισοδύναμο με single-flip). Η πιθανότητα  $P_{\text{add}}$  εξαρτάται από τη θερμοκρασία  $\beta$  και είναι μικρή για μεγάλες θερμοκρασίες ( $\beta \ll \beta_c$ ) και σχεδόν 1 όταν η θερμοκρασία τείνει στο μηδέν ( $\beta \gg \beta_c$ ). Άρα στην πρώτη περίπτωση ο αλγόριθμος φτιάχνει πολύ μικρά clusters (μεγέθους 1 για  $\beta = 0$ ) ενώ μεγάλα στη δεύτερη. Άρα στις μεγάλες θερμοκρασίες έχουμε σχεδόν τυχαία spin flips όπως και στον αλγόριθμο Metropolis ενώ στις μικρές θερμοκρασίες (μεγάλο  $\beta$ ) ο αλγόριθμος έχει μεγάλη πιθανότητα να αλλάξει το cluster που επικρατεί στο πλέγμα. Αυτό φαίνεται καθαρά στο Σχήμα 5.2 όπου υπολογίζουμε το ποσοστό του πλέγματος που καλύπτεται από το τυπικό Wolff cluster  $\langle n \rangle / N$  σαν συνάρτηση της θερμοκρασίας. Για μικρά  $\beta$ ,  $\langle n \rangle / N \rightarrow 1/N$  ενώ για μεγάλα  $\langle n \rangle / N \rightarrow 1$ .

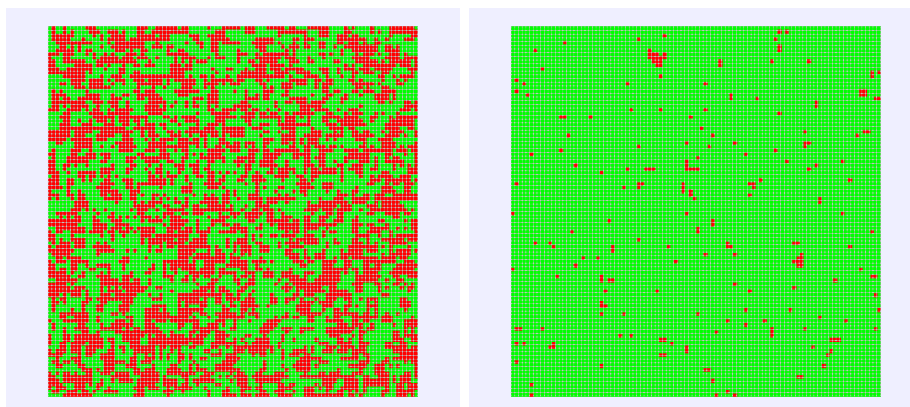
Στο Σχήμα 5.3 βλέπουμε τυπικές διατάξεις των σπιν σε μικρές και μεγάλες θερμοκρασίες. Για μικρά  $\beta$  ο αλγόριθμος Wolff επιλέγει τυχαία πλεγματική θέση, φτιάχνει (συνήθως) ένα μικρό cluster και του αλλάζει την τιμή των σπιν. Ο αλγόριθμος Metropolis επιλέγει τυχαία πλεγματική θέση και με μεγάλη πιθανότητα της αλλάζει το σπιν. Το αποτέλεσμα είναι η (σχεδόν) τυχαία κατανομή των σπιν όπως αναμένεται στη φάση αταξίας. Για μεγάλα  $\beta$  οι



Σχήμα 5.2: Η εξάρτηση του μεγέθους των Wolff clusters από τη θερμοκρασία. Δείχνεται το μέσο μέγεθος ενός Wolff cluster σε σχέση με μέγεθος του πλέγματος. Σε μεγάλες θερμοκρασίες  $\beta \ll \beta_c$  αυτό είναι  $\sim 1/N$  ενώ όταν η θερμοκρασία γίνεται πολύ μικρή  $\beta \gg \beta_c$  γίνεται  $\sim 1$ . Τα δεδομένα είναι για το πρότυπο Ising στο τετραγωνικό πλέγμα για  $L = 40$ .

διατάξεις των σπιν “παγώνουν” με μικρές θερμικές διακυμάνσεις σκορπισμένες στο πλέγμα. Ο αλγόριθμος του Wolff τυπικά επιλέγει για seed μία πλεγματική θέση στο cluster που επικρατεί στο πλέγμα και το νέο cluster είναι με μεγάλη πιθανότητα σχεδόν όλο το προηγούμενο. Με την αλλαγή της τιμής των σπιν, όλες οι θερμικές διακυμάνσεις των αντίθετων σπιν ενσωματώνονται στο νέο cluster. Νέες θερμικές διακυμάνσεις προκύπτουν από τα (λίγα και σκόρπια) spin που δεν ενσωματώθηκαν στο νέο cluster. Ο αλγόριθμος Metropolis επιλέγει τυχαίες θέσεις στο πλέγμα και αλλάζει με μικρή πιθανότητα τα σπιν μέσα στο μεγάλο cluster ενώ εξαφανίζει με πιθανότητα 1 τις θερμικές διακυμάνσεις. Οι δύο αλγόριθμοι λειτουργούν με όμοια αποτελεσματικότητα.

Στο Σχήμα 5.3 βλέπουμε τυπικές διατάξεις των σπιν στην ψευδοκρίσιμη περιοχή. Αυτές κυριαρχούνται από cluster μεγάλου μεγέθους με τυχαίο σχήμα, μέγεθος και κατανομή θέσης στο πλέγμα. Ο αλγόριθμος Wolff φτιάχνει με μεγάλη πιθανότητα μεγάλα clusters με αποτέλεσμα μεγάλα cluster να καταστρέφονται και να δημιουργούνται σε χρόνο ανάλογο με το μέγεθος των cluster (στο Σχήμα 5.2 βλέπουμε ότι  $\langle n \rangle / N \approx 0.5$ ). Αντιθέτως ο αλγόριθμος Metropolis



Σχήμα 5.3: Μία τυπική διάταξη των σπιν στη φάση αταξίας (αριστερά,  $\beta = 0.25$ ) και τάξης (δεξιά,  $\beta = 0.5556$ ) για το πρότυπο Ising στο τετραγωνικό πλέγμα για  $L = 100$ .

αλλάζει τα cluster κυρίως μεταβάλλοντας το σχήμα τους στο σύνορο με αποτέλεσμα να χρειάζονται πολλά sweeps για την καταστροφή μιας διάταξης από spin clusters και δημιουργία μιας νέας και στατιστικά ανεξάρτητης από την προηγούμενη. Αναμένεται λοιπόν δραστική μείωση του χρόνου αυτοσυσχετισμού στην ψευδοκρίσιμη περιοχή με τη χρήση του αλγόριθμου του Wolff.

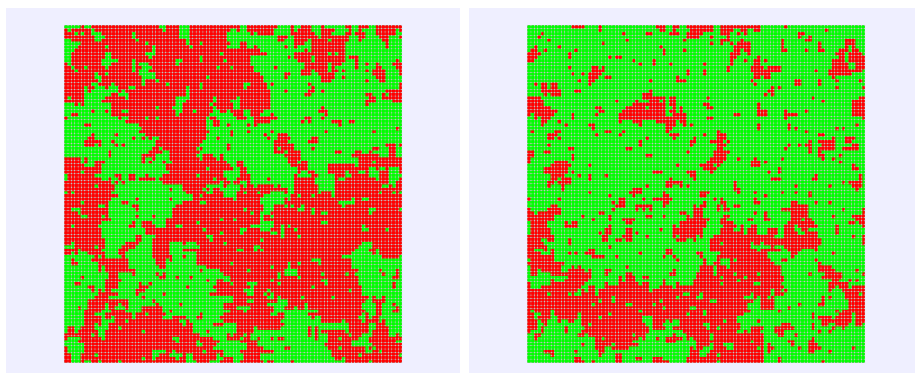
Η μέση τιμή του μεγέθους των Wolff clusters είναι μια δυναμική ποσότητα. Για να το δούμε αυτό καθαρά θα δείξουμε ότι στη φάση αταξίας ( $\beta < \beta_c$ ) ισχύει

$$\chi = \beta \langle n \rangle . \quad (5.14)$$

Παρουσιάζουμε την απόδειξη από τους Newmann και Barkema [1]: Αντί να κατασκευάσουμε ένα Wolff cluster, επισκεπτόμαστε κάθε ζευγάρι από πλεγματικές θέσεις και αν έχουν το ίδιο σπιν τότε δημιουργούμε μεταξύ τους ένα δεσμό με πιθανότητα  $P_{\text{add}} = 1 - e^{-2\beta}$ . Όταν τελειώσει η διαδικασία το πλέγμα έχει χωριστεί σε  $N_c$  clusters κάθε ένα από τα οποία έχει  $n_i$  μέλη και σπιν  $S_i$ . Επιλέγουμε τυχαία μία πλεγματική θέση και αλλάζουμε το σπιν του cluster στο οποίο ανήκει. Καταστρέφουμε τους δεσμούς και επαναλαμβάνουμε τη διαδικασία<sup>4</sup>. Η συνολική μαγνήτιση τότε είναι:

$$M = \sum_{i=1}^{N_c} S_i n_i , \quad (5.15)$$

<sup>4</sup> Αυτό είναι ισοδύναμο ως προς το αποτέλεσμα με τον αλγόριθμο Wolff, φυσικά όχι με τον πιο ... αποδοτικό τρόπο.



Σχήμα 5.4: Δύο τυπικές διατάξεις των σπιν στην (ψευδο)κρίσιμη περιοχή ( $\beta = 0.4348$ ) για το πρότυπο Ising στο τετραγωνικό πλέγμα για  $L = 100$ . Οι δύο διατάξεις “απέχουν” 5000 sweeps χρησιμοποιώντας τον αλγόριθμο Metropolis.

και

$$\langle M^2 \rangle = \left\langle \left( \sum_{i=1}^{N_c} S_i n_i \right) \left( \sum_{j=1}^{N_c} S_j n_j \right) \right\rangle = \left\langle \sum_{i \neq j} S_i S_j n_i n_j \right\rangle + \left\langle \sum_i S_i^2 n_i^2 \right\rangle. \quad (5.16)$$

Οι τιμές  $S_i = \pm 1$  είναι ισοπίθανες λόγω της συμμετρίας του μοντέλου άρα ο πρώτος όρος είναι μηδέν. Επειδή  $S_i^2 = 1$  παίρνουμε τελικά

$$\langle m^2 \rangle = \frac{1}{N^2} \langle M^2 \rangle = \frac{1}{N^2} \left\langle \sum_i n_i^2 \right\rangle. \quad (5.17)$$

Στον αλγόριθμο Wolff η κατασκευή ενός cluster ισοδυναμεί με την επιλογή ενός από τα cluster που φτιάξαμε παραπάνω. Σύμφωνα με τα παραπάνω (τυχαία επιλογή πλεγματικής θέσης και αλλαγή του σπιν στο cluster που ανήκει) η πιθανότητα επιλογής του cluster  $i$  είναι

$$p_i = \frac{n_i}{N}, \quad (5.18)$$

άρα η μέση τιμή των Wolff clusters θα είναι

$$\langle n \rangle = \left\langle \sum_i p_i n_i \right\rangle = \left\langle \sum_i \frac{n_i}{N} n_i \right\rangle = N \langle m^2 \rangle. \quad (5.19)$$

Χρησιμοποιώντας τη Σχέση (5.17) και το γεγονός ότι για  $\beta < \beta_c$

έχουμε  $\langle m \rangle = 0^5$  παίρνουμε τελικά

$$\chi = \beta N (\langle m^2 \rangle - \langle m \rangle^2) = \beta \langle n \rangle . \quad (5.20)$$

### 5.3 ΣΧΕΔΙΑΣΜΟΣ ΚΩΔΙΚΑ

Στην παράγραφο αυτή θα σχεδιάσουμε την υλοποίηση του αλγόριθμου Wolff όπως αυτός παρουσιάστηκε στη σελίδα 144. Για να αναπτύξουμε ένα cluster γύρω από το seed που επιλέγεται θα χρειαστεί να χρησιμοποιηθεί μια βοηθητική θέση στη μνήμη (buffer) από την οποία θα ανασύρουμε τα νέα μέλη του cluster για να εξετάσουμε αν θα προσθέσουμε στο cluster τους πλησιέστερους γείτονές τους. Κάθε πλεγματική θέση που θα προσθέτουμε θα αποθηκεύεται στο buffer για να εξεταστεί με τη σειρά του αργότερα.

Υπάρχουν δύο τέτοιες δομές δεδομένων που αρκούν για τη δουλειά που θέλουμε να κάνουμε. Η πρώτη είναι το stack (ή LIFO: last in – first out) και η δεύτερη το queue (ή FIFO: first in – first out). Και οι δύο είναι arrays στα οποία η διαφορά είναι το πώς ανασύρουμε τα δεδομένα από αυτά. Αυτό γίνεται ακριβώς όπως λέει και το όνομά τους. Στο stack όταν ανασύρουμε ένα στοιχείο, παίρνουμε αυτό που είχαμε βάλει τελευταίο. Στο queue όταν ανασύρουμε ένα στοιχείο, παίρνουμε αυτό που είχαμε βάλει πρώτο.

Το stack υλοποιείται από ένα μονοδιάστατο array  $stack[N]$  στο οποίο “σπρώχνουμε” (push) μια νέα τιμή και “αντλούμε” (pop) μια τιμή που θέλουμε να χρησιμοποιήσουμε. Για τη χρήση αυτή χρειαζόμαστε έναν ακέραιο  $m$  ίσο με τα ενεργά στοιχεία στο stack. Το στοιχείο που θα αντλήσουμε ανά πάσα στιγμή είναι το  $stack[m-1]$ . Για να σπρώξουμε ένα στοιχείο  $e$  στο stack ακολουθούμε τα εξής βήματα:

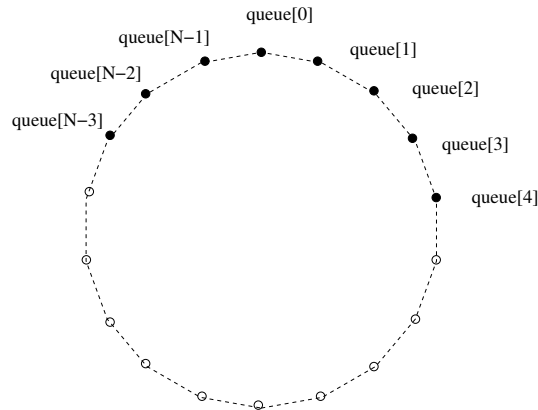
1. Ελέγχουμε αν το stack είναι γεμάτο ( $m < N$ )
2. Θέτουμε  $stack[m] = e$
3. Αυξάνουμε το  $m$  κατά 1

Αντίθετα, για να τραβήξουμε μια τιμή και να την αποθηκεύσουμε στη μεταβλητή  $e$  κάνουμε τα εξής:

1. Ελέγχουμε αν το stack είναι άδειο ( $m > 0$ )

<sup>5</sup>Αυτό ισχύει στο θερμοδυναμικό όριο, για πεπερασμένο πλέγμα οι δύο ποσότητες θα διαφέρουν κατά τον παράγοντα  $\langle m \rangle > 0$ , η διαφορά αυτή όμως εξαφανίζεται καθώς το μέγεθος του πλέγματος απειρίζεται.

2. Μειώνουμε το  $m$  κατά 1
3. Θέτουμε  $e = \text{stack}[m]$



Σχήμα 5.5: Η τοπολογία των δεδομένων στο queue. Στο εικονιζόμενο array έχουμε 8 στοιχεία αποθηκευμένα στο  $\text{queue}[N-3] \dots \text{queue}[4]$ . Έχουμε  $m=5$ ,  $n=N-3$ ,  $m-n = 8 \bmod N$ . Ένα στοιχείο προστίθεται στο  $\text{queue}[m]=\text{queue}[5]$  και ένα στοιχείο ανασύρεται διαβάζοντας το  $\text{queue}[n]=\text{queue}[N-3]$ .

Η υλοποίηση του queue γίνεται διαφορετικά. Η τοπολογία της διάταξης των στοιχείων είναι κυκλική αντί για γραμμική όπως φαίνεται στο Σχήμα 5.5. Χρησιμοποιούμε ένα array  $\text{queue}[N]$  και δύο ακέραιους  $m$ ,  $n$  να δείχνουν στην αρχή και τέλος του buffer. Η αρχή των δεδομένων είναι το στοιχείο  $\text{queue}[m-1]$  ενώ το τέλος το  $\text{queue}[n]$ . Όταν το queue είναι άδειο  $m=n$  ενώ το ίδιο ισχύει και όταν είναι γεμάτο. Άρα θα χρειαστούμε και μια μεταβλητή  $flag$  την οποία αρχικά θα της δώσουμε την τιμή 0 (άδειο queue)<sup>6</sup>. Ο αριθμός  $(m-n) \bmod N$  είναι ο αριθμός των αποθηκευμένων στοιχείων<sup>7</sup>. Όταν πρόκειται το queue γεμίζει θα παίρνει την τιμή 1. Άρα για να προσθέσουμε ένα στοιχείο  $e$  στο queue ακολουθούμε τα εξής βήματα:

1. Ελέγχουμε αν το stack είναι γεμάτο ( $flag=1$ )
2. Θέτουμε  $\text{queue}[m] = e$
3. Αυξάνουμε το  $m$  κατά  $1 \bmod N$
4. Αν  $m=n$  θέτουμε  $flag=1$

<sup>6</sup>Αν επιλέξουμε να αποθηκεύουμε το πολύ  $N-1$  στοιχεία στο  $\text{queue}[N]$  τότε ο αλγόριθμος απλοποιείται (άσκηση).

<sup>7</sup>Εκτός αν  $m=n$  οπότε είναι 0 ή  $N$  ανάλογα με την τιμή της  $flag$ .

Για να τραβήξουμε μια τιμή και να την αποθηκεύσουμε στη μεταβλητή  $e$  κάνουμε τα εξής:

1. Ελέγχουμε αν το `stack` είναι άδειο (`flag=0`)
2. Θέτουμε  $e = \text{queue}[n]$
3. Αυξάνουμε το  $n$  κατά  $1 \bmod N$
4. Αν  $m=n$  θέτουμε `flag=0`

Συνοψίζοντας την αλγοριθμική διαδικασία κατασκευής ενός Wolff cluster στην περίπτωση του πρότυπου Ising θα κάνουμε τα εξής:

1. Επιλέγουμε ένα `seed` από το πλέγμα με πιθανότητα  $1/N$ .
2. Ελέγχουμε τους πλησιέστερους γείτονές του. Αν έχουν το ίδιο σπιν, τους προσθέτουμε στο `cluster` με πιθανότητα  $P_{\text{add}} = 1 - e^{-2\beta}$ . Τα σπιν που προσθέτουμε ("νέα μέλη") αποθηκεύονται στο `array stack[N]` σύμφωνα με τα παραπάνω.
3. Ανασύρουμε μία πλεγματική θέση από το `stack[N]`. Αν είναι άδειο προχωρούμε στο επόμενο βήμα. Ελέγχουμε τους πλησιέστερους γείτονές της. Αν δεν είναι ήδη στο `cluster` και έχουν το ίδιο σπιν, τους προσθέτουμε στο `cluster` με πιθανότητα  $P_{\text{add}}$ .
4. Σημειώνουμε το μέγεθος του `cluster` και αλλάζουμε το σπιν του.

Η επιλογή `stack` ή `queue` βρίσκεται πως δεν έχει ουσιαστική διαφορά. Τα αποτελέσματα είναι ίδια και δεν υπάρχουν σημαντικές διαφορές στην ταχύτητα. Η διαφορά είναι μόνο στον τρόπο που αναπτύσσονται τα `clusters` (για `stack` το `cluster` αναπτύσσεται σπειροειδώς γύρω από το `seed` ενώ για `queue` το `cluster` αναπτύσσεται πρώτα σε μία διεύθυνση και μετά στην άλλη). Ο προσεκτικός προγραμματιστής θα δοκιμάσει και τις δύο ώστε να ελέγξει τα αποτελέσματά του για λάθη στον κώδικα. Πιο σημαντικός λόγος είναι να ελέγξει την ποιότητα της γεννήτριας τυχαίων αριθμών αφού ο αλγόριθμος του Wolff είναι ιδιαίτερα ευαίσθητος στις ατέλειές τους.

### 5.3.1 Ο ΚΩΔΙΚΑΣ

Στην παράγραφο αυτή θα παρουσιάσουμε και θα αναλύσουμε τον κώδικα που θα χρησιμοποιήσουμε στην προσομοίωση του πρότυπου Ising με τον αλγόριθμο Wolff. Αυτόν το παίρνουμε από το βιβλίο των Newman και Barkema [1].



Η καρδιά του αλγόριθμου βρίσκεται στη συνάρτηση `wolff()` την οποία αποθηκεύουμε στο αρχείο `wolff.c`. Κάθε κλήση της συνάρτησης από το κυρίως πρόγραμμα κατασκευάζει ένα Wolff cluster, του αλλάζει το σπιν και σημειώνει το μέγεθός του.

Στο πρώτο μέρος της συνάρτησης δημιουργούμε το buffer τύπου `stack stack[N]` το οποίο θα χρησιμοποιήσουμε στην κατασκευή του cluster. Αυτό το κάνουμε δυναμικά με τη χρήση της συνάρτησης `malloc()` και είμαστε προσεκτικοί στο τέλος να ελευθερώσουμε τη ζητούμενη μνήμη με τη συνάρτηση `free()` (αλλιώς θα έχουμε "memory leak" και η μνήμη που ζητά το πρόγραμμα θα αυξηθεί ανεξέλεγκτα).

```
stack = (int *)malloc((size_t)(N*sizeof(int)));
if(!stack)locerr("allocation failure for stack in wolff()");
....
free(stack);
```

Παρατηρούμε πως αν το σύστημα αδυνατεί να μας δώσει τη ζητούμενη μνήμη, τότε θέτει `stack=NULL` και η `locerr()` σταματάει το πρόγραμμα.

Στη συνέχεια αρχικοποιούμε το cluster με την επιλογή του `seed`.

```
i=N*drandom();
stack[0]=i;
sp=1;
oldspin = s[i];
newspin = -s[i];
s[i] = newspin;
ncluster = 1;
```

`i` είναι το `seed` και τοποθετείται αμέσως στο cluster (`stack[0]=1`). Η μεταβλητή `sp` είναι ο αριθμός των στοιχείων του `stack` και αρχικά τίθεται ίση με 1. Η `ncluster` μετράει τον αριθμό των πλεγματικών θέσεων στο cluster και τίθεται αρχικά ίση με 1. `oldspin=s[i]` είναι η παλιά τιμή του σπιν του cluster και `newspin=-oldspin` η νέα. Η τιμή του σπιν του νέου μέλους του cluster αλλάζεται *αμέσως* (`s[i]=newspin`)! Αυτό βοηθάει στην πιο αποτελεσματική εφαρμογή του αλγόριθμου. Ελέγχοντας αν το σπιν ενός πλησιέστερου γείτονα είναι ίσο με `oldspin` ελέγχει *ταυτόχρονα* αν το σπιν του είναι ίδιο με του cluster και αν ήδη έχει ήδη μπει στο cluster από προηγούμενο έλεγχο!

Ο βρόχος επανάληψης πάνω στα νέα μέλη του cluster συνοψίζεται παρακάτω:

```
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++;
        }
    .... check other three nearest neighbors ....
}/*while(sp)*/

```

Ο βρόχος `while(sp)` εκτελείται όσο  $sp > 0$ , δηλ. όσο το `stack` είναι γεμάτο και έχουμε νέα μέλη το `cluster`. Η μεταβλητή `current` είναι η τρέχουσα πλεγματική θέση που τραβάμε από το `stack` για να κοιτάξουμε τους πλησιέστερους γείτονές της ( $current = stack[--sp]$ )<sup>8</sup>. Η γραμμή `if((nn=current+XNN)>=N) nn -= N;` επιλέγει τον πλησιέστερο γείτονα προς τα δεξιά και τον αποθηκεύει στη μεταβλητή `nn`. Αν το σπιν του `nn` το `s[nn]` είναι ίσο με `oldspin` τότε ο γείτονας αυτός έχει σπιν ίδιο με αυτό του `cluster` και δεν έχει ήδη μπει στο `cluster` (γιατί τότε θα είχαμε αλλάξει το σπιν του). Η μεταβλητή `padd` είναι ίση με  $P_{add}$  (αυτό τίθεται στην `init()`) και αν `drandom() < padd` (το οποίο συμβαίνει με πιθανότητα  $P_{add}$ ) προσθέτουμε τον `nn` στο `cluster`: Τον προσθέτουμε στο `stack` (`stack[sp++] = nn`)<sup>9</sup>, αλλάζουμε το σπιν (`s[nn] = newspin`) και αυξάνουμε το μέγεθος του `cluster` κατά 1 (`ncluster++`). Επαναλαμβάνουμε ακριβώς τα ίδια και για τους υπόλοιπους πλησιέστερους γείτονες. Ολόκληρος ο κώδικας παρατίθεται παρακάτω για τη διευκόλυνση του αναγνώστη:

```

/***** wolf.c *****/
#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)));

```

<sup>8</sup>Η τιμή της παράστασης `--sp` προκύπτει αφού μειωθεί κατά 1, όπως πρέπει στον αλγόριθμο του `stack`.

<sup>9</sup>Η τιμή της παράστασης `sp++` προκύπτει πριν από την αύξηση του `sp` κατά 1, όπως πρέπει στον αλγόριθμο του `stack`.

```

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()*/

```

Για να συνδέσουμε τη συνάρτηση με το υπόλοιπο πρόγραμμα ώστε να έχουμε κατασκευή ενός cluster ανά “sweep” μεταβάλλουμε τη `main()` ως εξής:

```

/***** 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();
}

```

Ορίσαμε τη (global) μεταβλητή `algorithm` ώστε ο χρήστης να μπορεί να ελέγχει τον αλγόριθμο που θα χρησιμοποιήσει (Wolff ή Metropolis)<sup>10</sup>. Μένει να οριστεί η (global) μεταβλητή  $p_{add} \equiv P_{add} = 1 - e^{-2\beta}$ , κάτι το οποίο κάνουμε στην `init()`. Στο αρχείο `include.h` προσθέτουμε τις γραμμές

```

double padd;
int    algorithm;

```

ΚΑΙ ΣΤΟ `init.c`

```

algorithm=0;//default is metropolis, 1 is wolff
get_the_options(argc,argv);
padd = 1.0 - exp(-2*beta);

```

ενώ μεταβάλλουμε το αρχείο `options.c` ώστε να παίρνει ένα διακόπτη `-w` όπου θα θέτει `algorithm=1` ώστε το πρόγραμμα να λειτουργεί τον αλγόριθμο του Wolff:

<sup>10</sup>Μεταβάλετε την παραπάνω συνάρτηση ώστε η `wolff()` να καλείται μέχρι φτιάξει clusters με συνολικό μέγεθος τουλάχιστον `N`.

```
#define OPTARGS "?h1234567890.L:b:s:S:n:w"
void get_the_options(int argc, char **argv){
    .....
    while (!errflg && (c = getopt(argc, argv, OPTARGS)) != -1){
        switch(c){
            case 'w':
                algorithm = 1;
                break;
            .....
        }
    }
}
```

Προσθέτουμε τη σχετική πληροφορία στο μήνυμα βοήθειας `usage()` και `simmessage()` και ... είμαστε έτοιμοι! Για τη μεταγλώττιση χρησιμοποιούμε το `Makefile`

```
# ##### Makefile #####
OBJS = main.o init.o wolff.o met.o measure.o end.o options.o drandom.o
CFLAGS = -O2
LIBS = -lm

is: $(OBJS)
$(CC) $(CFLAGS) $^ -o $@ $(LIBS)

$(OBJS): include.h
```

και με την εντολές

```
> make
> ./is -h
Usage: is [options]
    -L: Lattice length (N=L*L)
    -b: beta (options beta overrides the one in config)
    -s: start (0 cold, 1 hot, 2 old config.)
    -S: seed (options seed overrides the one in config)
    -n: number of sweeps and measurements of E and M
    -w: use wolff algorithm for the updates
    .....
> ./is -L 20 -b 0.44 -s 1 -S 34235322 -n 5000 -w > outL20b0.44
```

μεταγλωττίζουμε, υπενθυμίζουμε στον εαυτό μας τον τρόπο χρήσης του εκτελέσιμου αρχείου `is` και κάνουμε ένα δοκιμαστικό τρέξιμο για  $L = 40$ ,  $\beta = 0.44$  φτιάχνοντας 5000 clusters ξεκινώντας από μία "θερμή" διάταξη των σπιν αποθηκεύοντας τα δεδομένα στο αρχείο `outL20b0.44`.

## 5.4 ΣΥΛΛΟΓΗ ΔΕΔΟΜΕΝΩΝ (και shell scripting)

Για να πάρουμε τα αποτελέσματά μας επιλέγουμε αρχικά το μέγεθος ενός πλέγματος που θέλουμε να προσομοιώσουμε και στη συνέχεια τις θερμοκρασίες που θα προσομοιώσουμε. Η δουλειά γίνεται γρήγορα βαρετά επαναλαμβανόμενη, κουραστική και τότε είναι που υπεισέρχονται απρόσκλητοι επισκέπτες τα ... ζουζούνια! Η τεμπελιά στην περίπτωση αυτή είναι αρετή και αξίζει να μάθουμε μερικές τεχνικές που κάνουν τη ζωή μας ευκολότερη αλλά και τα αποτελέσματά μας πιο αξιόπιστα (...συνήθως...). Στη βοήθειά μας έρχεται ο φλοιός (Shell) και τα εργαλεία του. Ο φλοιός μας επιτρέπει να εκτελέσουμε μια σειρά εντολών με το ίδιο ακριβώς συντακτικό που θα χρησιμοποιούσαμε στη γραμμή εντολών γράφοντάς τις σε ένα απλό αρχείο κειμένου. Ένα απλό παράδειγμα είναι η σειρά εντολών που γράφουμε στο αρχείο run1:

```
# ##### run1 #####
./is -L 20 -b 0.10 -s 1 -n 5000 -w -S 3423 > outL20b0.10
./is -L 20 -b 0.20 -s 2 -n 5000 -w > outL20b0.20
./is -L 20 -b 0.30 -s 2 -n 5000 -w > outL20b0.30
./is -L 20 -b 0.40 -s 2 -n 5000 -w > outL20b0.40
./is -L 20 -b 0.42 -s 2 -n 5000 -w > outL20b0.42
./is -L 20 -b 0.44 -s 2 -n 5000 -w > outL20b0.44
./is -L 20 -b 0.46 -s 2 -n 5000 -w > outL20b0.46
./is -L 20 -b 0.48 -s 2 -n 5000 -w > outL20b0.48
./is -L 20 -b 0.50 -s 2 -n 5000 -w > outL20b0.50
./is -L 20 -b 0.60 -s 2 -n 5000 -w > outL20b0.60
./is -L 20 -b 0.70 -s 2 -n 5000 -w > outL20b0.70
```

Η πρώτη γραμμή που αρχίζει από το χαρακτήρα # είναι σχόλιο και θα αγνοηθεί από το φλοιό. Η δεύτερη ξεκινάει από την αρχή μια προσομοίωση από “καυτή” διάταξη σπιν (-s 1) για πλέγμα με L=20 (-L 20) και θερμοκρασία  $\beta = 0.10$  (-b 0.10). Αρχικό seed καθορίζεται ο αριθμός 3423 (-S 3423) και κάνουμε μετρήσεις πάνω σε 5000 Wolff clusters (-n 5000 -w). Τα αποτελέσματα ανακατευθύνονται από το stdin στο αρχείο outL20b0.10 (> outL20b0.10).

Οι επόμενες δέκα γραμμές συνεχίζουν την προσομοίωση στις θερμοκρασίες  $\beta = 0.20 - 0.70$ . Κάθε προσομοίωση συνεχίζει από τη διάταξη σπιν που αποθήκευσε το πρόγραμμα στο αρχείο conf στην προηγούμενη προσομοίωση.

Μόλις φτιάξαμε ένα (πρωτόγονο είναι αλήθεια) σενάριο φλοιού (shell script)! Για να τρέξουν οι εντολές αυτές πρέπει να κάνουμε το αρχείο εκτελέσιμο, δίνοντάς του (την πρώτη φορά μόνο) άδεια πρόσβασης x

```
> chmod a+x run1
```

και στη συνέχεια να το εκτελέσουμε καλώντας το

```
> ./run1
```

Δεν τα πήγαμε και άσχημα... Αλλά μπορούμε και καλύτερα! Αντί να βάλουμε μία γραμμή για κάθε προσομοίωση, μπορούμε να χρησιμοποιήσουμε τις προγραμματιστικές δυνατότητες του φλοιού. Ας δούμε πώς. Στο αρχείο run2 γράφουμε:

```
#!/bin/tcsh -f
# ##### run2 #####
set L      = 20
set betas  = (0.10 0.20 0.30 0.40 0.42 0.44 0.46 0.48 0.50 0.60 0.70)
set start  = "-s 1 -S 3423"
set nsweeps = 5000

foreach beta ($betas)
  echo "L= $L beta= $beta"
  ./is -L $L -b $beta -n $nsweeps -w $start > outL${L}b${beta}
  set start = "-s 2"
end
```

Η πρώτη γραμμή καλεί το φλοιό tcsh να ερμηνεύσει το σενάριο. Αυτό δε χρειαζόταν στο run1 γιατί οποιοσδήποτε φλοιός μπορούσε να ερμηνεύσει τις εντολές που είχαμε γράψει. Εδώ όμως βάζουμε συντακτικό που είναι κατανοητό μόνο από τον συγκεκριμένο φλοιό. Το ζεύγος χαρακτήρων #! πρέπει να είναι οι δύο πρώτοι χαρακτήρες του αρχείου.

Η δεύτερη γραμμή είναι όπως και πριν σχόλιο.

Η τρίτη γραμμή ορίζει μία μεταβλητή φλοιού με το όνομα L. Η τιμή της μεταβλητής είναι "20" (string, όχι αριθμός ακέραιος) και έχουμε πρόσβαση σε αυτή βάζοντας ένα \$ μπροστά από το όνομά της. Δηλ. όπου γράφουμε στο σενάριο \$L (ή \${L}) ο φλοιός θα αντικαθιστά το string 20. Για παράδειγμα όταν γράφουμε στο σενάριο outL\${L}b ο φλοιός θα φτιάχνει το string outL20b.

Η τέταρτη γραμμή ορίζει ένα array με το όνομα betas. Πρόσβαση στα στοιχεία του array παρέχεται από το συντακτικό \$betas[αριθμός], όπου "αριθμός" είναι το στοιχείο του array αρχίζοντας από το 1. Στο παραπάνω παράδειγμα \$betas[1]= 0.10, \$betas[2]= 0.20, ..., \$betas[11]= 0.70. Ειδικά η μεταβλητή \$#betas=11 είναι ο αριθμός των στοιχείων του array. Όταν γράφουμε \$betas ο φλοιός αντικαθιστά όλες τις τιμές του array<sup>11</sup>.

<sup>11</sup>Δοκιμάστε την εντολή: echo \$betas[3] \$#betas \$betas

Η πέμπτη γραμμή ορίζει τη μεταβλητή `start` να έχει την τιμή `$start` ίση με το string `"-s 1 -S 3423"`. Τα εισαγωγικά μπήκαν επειδή οι λέξεις χωρίζονται με κενά και ορίζουν έτσι επακριβώς το string. Οτιδήποτε γράφουμε από το χαρακτήρα `#` και μετά είναι σχόλιο και ο φλοιός το αγνοεί.

Η εντολή `foreach` είναι ένας τρόπος να τρέξουμε ένα βρόχο στο φλοιό `tcsh`. Οι επαναλαμβανόμενες εντολές αρχίζουν από την επόμενη γραμμή και τελειώνουν στη γραμμή με την εντολή `end`. Οι επαναλήψεις γίνονται μία για κάθε τιμή του `array` που βάζουμε ανάμεσα στις παρενθέσεις. Εδώ θα γίνει μία φορά για κάθε τιμή της μεταβλητής `$betas`. Κάθε φορά που γίνεται μια επανάληψη, η μεταβλητή βρόχου, το όνομα της οποίας μπαίνει αμέσως μετά τη λέξη `foreach` αντικαθίσταται από το επόμενο στοιχείο του `array`. Άρα ο βρόχος που γράψαμε θα εκτελεστεί μία φορά για κάθε τιμή του `betas` και η μεταβλητή `beta` θα παίρνει διαδοχικά τις τιμές 0.10, 0.20, ... , 0.70. Οι επόμενες τρεις γραμμές είναι οι εντολές που θα εκτελεστούν έντεκα φορές. Η `echo` "αντηχεί" τα ορίσματά της και μας πληροφορεί για την τρέχουσα τιμή των παραμέτρων της προσομοίωσης που τρέχει (πολύ χρήσιμο, ειδικά αν οι προσομοιώσεις θέλουν αρκετό χρόνο). Η `./is` τρέχει το πρόγραμμα, κάθε φορά με διαφορετική τιμή για τη `beta`. Προσέξτε πώς το αρχείο στο οποίο ανακατευθύνουμε τα δεδομένα αλλάζει όνομα κάθε φορά που αλλάζει η τιμή της `beta`. Άρα στο τέλος θα έχουμε τα δεδομένα μας στα αρχεία `outL20b0.10`, `outL20b0.20`, ..., `outL20b0.70`. Η τρίτη εντολή αλλάζει τον τρόπο που αρχίζει η προσομοίωση, από θερμή διάταξη σπιν στο να διαβάσει τη διάταξη από το αρχείο `conf`. Την πρώτη φορά που τρέχει ο βρόχος η τιμή της `start` είναι `"-s 1 -S 3423"` (θερμή διάταξη, `seed` ίσο με 3423) ενώ στις επόμενες η τιμή της είναι `"-s 2"` (old configuration).

Όταν λοιπόν θέλουμε να τρέξουμε για μια άλλη τιμή του `L` δεν έχουμε παρά να αλλάξουμε την τιμή της αντίστοιχης μεταβλητής και να τρέξουμε πάλι το ίδιο `script`. Ή μήπως όχι... Χτύπησε πάλι η τεμπελιά:

```
#!/bin/tcsh -f

set Ls      = (10 20 40)
set betas   = (0.10 0.20 0.30 0.40 0.42 0.44 0.46 0.48 0.50 0.60 0.70)
set nsweeps = 5000

foreach L    ($Ls )
  set start  = "-s 1 -S 3423"
  foreach beta ($betas)
    echo "L= $L beta= $beta"
```



```
./is -L $L -b $beta -n $nsweeps -w $start > outL${L}b${beta}
set start = "-s 2"
end
end
```

Δεν έχουμε παρά να προσθέσουμε ένα βρόχο για κάθε τιμή του  $L$  ορίζοντας τώρα τη μεταβλητή-array  $L_s$  και βάζοντας όσα  $L$  τραβάει η ψυχή μας... Προσέξτε μόνο πού μετακινήσαμε τον ορισμό της μεταβλητής  $start$  (γιατί;)

## 5.5 ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ

Η έλεγχος στα δεδομένα μας είναι να εξετάσουμε γραφικά την εξέλιξή τους στον Μόντε Κάρλο χρόνο (time histories). Από εκεί θα κάνουμε μια πρώτη διάγνωση για τυχών προβλήματα και λάθη και θα αποκτήσουμε μία αίσθηση για τους χρόνους εύρεσης της κατάστασης θερμικής ισορροπίας (thermalization) και των χρόνων αυτοσυσχετισμού. Από τα αρχεία που παράγαμε στην προηγούμενη παράγραφο μπορούμε να το καταφέρουμε εύκολα με τη βοήθεια του `gnuplot`. Για παράδειγμα οι εντολές μέσα από το `gnuplot`<sup>12</sup>:

```
gnuplot> plot "<grep -v '#' outL40b0.44" using 1 \
with lines title "E"
gnuplot> plot "<grep -v '#' outL40b0.44" using (abs($2)) \
with lines title "|M|"
gnuplot> plot "<awk '/#clu/{print $2}' outL40b0.44" using 1 \
with lines title "n"
```

μας δίνουν με απλό τρόπο τα time histories της ενέργειας, της απόλυτης τιμής της μαγνήτισης και του μεγέθους των cluster που κατασκευάζονται από τον αλγόριθμο του Wolff για  $L = 40$ ,  $\beta = 0.44$ <sup>13</sup>.

Για να υπολογίσουμε τις τιμές της μέσης ενέργειας ανά link  $\langle e \rangle = \frac{1}{2N} \langle E \rangle$  και της (απόλυτης τιμής της) μαγνήτισης ανά πλεγματική θέση  $\langle m \rangle = \frac{1}{N} \langle M \rangle$  μπορούμε να χρησιμοποιήσουμε το πρόγραμμα για `jaccknife` που περιγράψαμε στο Παράρτημα Β'. Αυτό θα το βρείτε και στα `Tools` στο συνοδευτικό λογισμικό στο αρχείο `jack.c`. Το μεταγλωττίζουμε στο εκτελέσιμο αρχείο `jack` με την εντολή

<sup>12</sup>Σας θυμίζουμε πως η εντολή `plot` στο `gnuplot` δέχεται αντί για το όνομα ενός αρχείου δεδομένων, το `stdout` μιας εντολής `command` με το συντακτικό `plot "<command"`.

<sup>13</sup>Οι μακριές γραμμές που δε χωράνε στο κείμενο, συνεχίζουν στην επόμενη γραμμή μετά το χαρακτήρα `\`.

```
> cc -O2 jack.c -o jack -lm
```

Η μέση τιμή  $\langle e \rangle$  μπορεί τότε να υπολογιστεί με την εντολή

```
> grep -v # outL40b0.44 | awk -v L=40 'NR>500{print $1/(2*L*L)}' \
| ./jack
```

όπου με την `awk` ορίσαμε τη μεταβλητή  $L=40$  και τυπώσαμε την πρώτη στήλη διαιρεμένη με  $2N = 2L^2$ . Με τη συνθήκη `NR>500` τυπώνουμε αφού περάσουν οι πρώτες 500 γραμμές, απορρίπτοντας τις πρώτες μετρήσεις για να είμαστε σίγουροι για την εύρεση της θερμικής ισορροπίας. Το αποτέλεσμα τυπώνεται στο `stdout` ως εξής:

```
# NDAT = 4500 data. JACK = 10 groups
# <o>, chi= (<o^2>-<o>^2)
# <o> +/- err                chi +/- err
-0.71091166666 0.0024162628283 0.0015719190590 7.819205433e-05
```

Οι γραμμές που αρχίζουν με `#` είναι σχόλια του προγράμματος και μας εξηγούν τα αποτελέσματα. Οι δύο πρώτοι αριθμοί είναι η  $\langle e \rangle$  και το σφάλμα της ενώ ο τρίτος και ο τέταρτος οι διακυμάνσεις της ενέργειας  $\langle e^2 \rangle - \langle e \rangle^2$  και το σφάλμα της. Πολλαπλασιάζοντας την τελευταία με  $\beta^2 N$  παίρνουμε την ειδική θερμότητα  $c$  και το σφάλμα της σύμφωνα με την εξίσωση (4.20). Μπορούμε να τυπώσουμε τις πληροφορίες αυτές βολικά για να τις αποθηκεύσουμε σε ένα αρχείο με την εντολή:

```
> set L = 40; set b = 0.44 ;                \
grep -v # outL${L}b${b}                    | \
awk -v L=${L} 'NR>500{print $1/(2*L*L)}' | \
./jack | grep -v #                          | \
awk -v L=${L} -v b=${b} '{print "e",L,b,$1,$2,b*b*L*L*$3,b*b*L*L*$4}'
```

Γιατί το κάναμε αυτό; Όχι λόγω ... μαζοχισμού! Η παραπάνω είναι 3 εντολές που δίνονται σε μία γραμμή στη γραμμή εντολών (φυσικά εμείς τη σπάμε για να φαίνεται στο κείμενο με τη γνωστή σύμβαση της `\`). Ανακαλώντας την και αλλάζοντας μόνο την τιμή της  $L$  ή/και της  $b$  παίρνουμε χωρίς καινούργιο κόπο τα αποτελέσματα για διαφορετική τιμή του  $L$  ή/και  $\beta$ . Επίσης το shell script που θα φτιάξουμε σε λίγο θα μας φαίνεται λιγότερο ακατανόητο... Το αποτέλεσμα είναι

```
e 40 0.42 -0.619523333 0.00189807 0.311391 0.0228302
```

που δίνει  $\langle e \rangle = -0.6195(19)$  και  $c = 0.311(23)$ .

Η εντολή για τον υπολογισμό της μαγνήτισης δεν παρουσιάζει τώρα ιδιαίτερη δυσκολία. Το μόνο που έχουμε να κάνουμε είναι να

υπολογίσουμε την απόλυτη τιμή της δεύτερης στήλης του αρχείου των δεδομένων για τις γραμμές που δεν αρχίζουν από #

```
> set L = 40 ; set b = 0.42 ; \
grep -v # outL${L}b${b} | \
awk -v L=$L 'NR>500{m=($2>0)?$2:-$2;print m/(L*L)}' | \
./jack | grep -v # | \
awk -v L=$L -v b=$b '{print "m",L,b,$1,$2,b*L*L*$3,b*L*L*$4}'
```

Η απόλυτη τιμή υπολογίζεται από την έκφραση  $(\$2>0)?\$2:-\$2$ , αποθηκεύεται στη μεταβλητή  $m$  και τυπώνεται αφού διαιρεθεί με  $N = L^2$ . Το αποτέλεσμα είναι

```
m 40 0.44 0.6250527778 0.00900370 21.8345 1.39975
```

που δίνει  $\langle m \rangle = 0.6251(90)$  και  $\chi = 21.8(14)$ .

Παρόμοια δίνουμε την εντολή για τον υπολογισμό του  $\langle n \rangle / N$ :

```
> set L = 40 ; set b = 0.44 ; \
grep '#clu' outL${L}b${b} | \
awk -v L=$L 'NR>500{print $2/(L*L)}' | \
./jack | grep -v # | \
awk -v L=$L -v b=$b '{print "n",L,b,$1,$2}'
```

Το αποτέλεσμα είναι

```
n 40 0.44 0.4257476389 0.01302602
```

που δίνει  $\langle n \rangle / N = 0.426(13)$ .

Να τα βάλουμε όλα μαζί στο script; Όπως τα δουλέψαμε αυτό είναι τετριμμένο τώρα. Αρκεί να προσθέσουμε τις παραπάνω εντολές μετά από κάθε εντολή προσομοίωσης και όλα θα γίνουν ... μόνα τους. Τα γράφουμε όλα μαζί στο script run3:

```
#!/bin/tcsh -f
set Ls      = (10 20 40 60 80 100)
set betas   = (0.00 0.10 0.20 0.25 0.30 0.34 0.38 \
               0.40 0.42 0.43 0.44 0.45 0.46 0.48 \
               0.48 0.50 0.55 0.60 0.65 0.70 0.80 )
set nsweeps = 100000

foreach L   ($Ls )
  set start = "-s 1 -S 3423"
  foreach beta ($betas)
    ./is -L $L -b $beta -n $nsweeps -w $start > outL${L}b${beta}
    set start = "-s 2"
    # Calculate <e> = <E>/(2N and c=beta^2*N*(<e^2>-<e>^2):
```

```

grep -v '#' outL${L}b${beta} | \
awk -v L=$L 'NR>500{print $1/(2*L*L)}' | \
./jack | grep -v '#' | \
awk -v L=$L -v b=$beta '{print "e",L,b,$1,$2,b*b*L*L*$3,b*b*L*L*$4}'
# Calculate <m> = <|M|>/N and chi=beta*N*(<m^2>-<m>^2)
grep -v '#' outL${L}b${beta} | \
awk -v L=$L 'NR>500{m=($2>0)?$2:-$2;print m/(L*L)}' | \
./jack | grep -v '#' | \
awk -v L=$L -v b=$beta '{print "m",L,b,$1,$2,b*L*L*$3,b*L*L*$4}'
end
end

```

Το τρέχουμε με την εντολή

```
> ./run3 > out &
```

και αφού τελειώσει μπορούμε να δούμε τα αποτελέσματα με το `gnuplot`<sup>14</sup>:

```

set xlabel "beta"
set ylabel "<m>"
plot "<grep '^m 10 ' out" using 3:4:5 with errorbars title " 10"
replot "<grep '^m 20 ' out" using 3:4:5 with errorbars title " 20"
replot "<grep '^m 40 ' out" using 3:4:5 with errorbars title " 40"
replot "<grep '^m 60 ' out" using 3:4:5 with errorbars title " 60"
replot "<grep '^m 80 ' out" using 3:4:5 with errorbars title " 80"
replot "<grep '^m 100 ' out" using 3:4:5 with errorbars title "100"

```

δίνει τη γραφική παράσταση για τη μαγνήτιση

```

set ylabel "chi"
set log y
plot "<grep '^m 10 ' out" using 3:6:7 with errorbars title " 10"
replot "<grep '^m 20 ' out" using 3:6:7 with errorbars title " 20"
replot "<grep '^m 40 ' out" using 3:6:7 with errorbars title " 40"
replot "<grep '^m 60 ' out" using 3:6:7 with errorbars title " 60"
replot "<grep '^m 80 ' out" using 3:6:7 with errorbars title " 80"
replot "<grep '^m 100 ' out" using 3:6:7 with errorbars title "100"

```

δίνει τη γραφική παράσταση της μαγνητικής επιδεκτικότητας

```

set ylabel "<e>"
plot "<grep '^e 10 ' out" using 3:4:5 with errorbars title " 10"
replot "<grep '^e 20 ' out" using 3:4:5 with errorbars title " 20"

```

<sup>14</sup>Πρόκειται για εντολές που δίνουμε μέσα στο `gnuplot` παρόλο που δεν ακολουθούμε τη συνηθισμένη σύμβαση να προτάσσουμε το prompt `gnuplot>`

```
replot "<grep '^e 40 ' out" using 3:4:5 with errorbars title " 40"
replot "<grep '^e 60 ' out" using 3:4:5 with errorbars title " 60"
replot "<grep '^e 80 ' out" using 3:4:5 with errorbars title " 80"
replot "<grep '^e 100 ' out" using 3:4:5 with errorbars title "100"
```

δίνει τη γραφική παράσταση της ενέργειας

```
set ylabel "c"
plot "<grep '^e 10 ' out" using 3:6:7 with errorbars title " 10"
replot "<grep '^e 20 ' out" using 3:6:7 with errorbars title " 20"
replot "<grep '^e 40 ' out" using 3:6:7 with errorbars title " 40"
replot "<grep '^e 60 ' out" using 3:6:7 with errorbars title " 60"
replot "<grep '^e 80 ' out" using 3:6:7 with errorbars title " 80"
replot "<grep '^e 100 ' out" using 3:6:7 with errorbars title "100"
```

δίνει τη γραφική παράσταση της ειδικής θερμότητας και

```
set ylabel "<n>/N"
plot "<grep '^n 10 ' out" using 3:4:5 with errorbars title " 10"
replot "<grep '^n 20 ' out" using 3:4:5 with errorbars title " 20"
replot "<grep '^n 40 ' out" using 3:4:5 with errorbars title " 40"
replot "<grep '^n 60 ' out" using 3:4:5 with errorbars title " 60"
replot "<grep '^n 80 ' out" using 3:4:5 with errorbars title " 80"
replot "<grep '^n 100 ' out" using 3:4:5 with errorbars title "100"
```

δίνει τη γραφική παράσταση της  $\langle n \rangle / N$ .

## 5.6 ΧΡΟΝΟΙ ΑΥΤΟΣΥΣΧΕΤΙΣΜΟΥ

Στην περίπτωση του αλγόριθμου του Metropolis ορίσαμε τη “μονάδα του χρόνου” στη κατασκευή του δείγματος στην προσομοίωση Μόντε Κάρλο όταν κατά μέσο όρο έχουμε  $N$  απόπειρες αλλαγής των σπιν του πλέγματος ή ισοδύναμα την κατά μέσο όρο απόπειρα μιας αλλαγής ανά σπιν του πλέγματος. Τη μονάδα αυτή την ονομάσαμε ένα sweep του πλέγματος και το μέγεθος του δείγματος το μετράγαμε στη μονάδα αυτή.

Στην περίπτωση του αλγόριθμου του Wolff έχουμε την επί πλέον πολυπλοκότητα που εισάγει το μεταβλητό μέγεθος των Wolff clusters με τη θερμοκρασία. Άρα η θέσπιση του βήματος του αλγόριθμου να είναι η κατασκευή ενός cluster δε δίνει ένα καλό ορισμό του sweep. Για πολύ χαμηλές θερμοκρασίες  $\beta \gg \beta_c$  η κατασκευή ενός cluster είναι σχεδόν ισοδύναμη με μία αλλαγή σπιν ανά πλεγματική θέση<sup>15</sup>. Αν κρατήσουμε αυτό να είναι η μονάδα μέτρησης

<sup>15</sup>Αν ο μέσος λόγος αποδοχής στον αλγόριθμο Metropolis για  $\beta \gg \beta_c$  είναι  $a$  τότε ένα Wolff cluster ισοδυναμεί κατά μέσο όρο με  $1/a$  Metropolis sweeps.

του χρόνου στην προσομοίωση, τότε ένα τέτοιο sweep θα είναι ίσο με

$$(1 \text{ sweep}) = \frac{N}{\langle n \rangle} (\text{Wolff cluster updates}) \quad (5.21)$$

Για μεγάλες θερμοκρασίες  $\beta \ll \beta_c$  χρειαζόμαστε περίπου  $N$  Wolff clusters για να έχουμε ένα sweep του πλέγματος. Ανάλογα θα ορίσουμε και τους χρόνους αυτοσυσχετισμού. Για μια φυσική ποσότητα  $\mathcal{O}$  ο χρόνος αυτοσυσχετισμού σε μονάδες κατασκευής Wolff cluster θα συμβολίζεται με  $\tau_{\mathcal{O}}^W$  ενώ ο χρόνος αυτοσυσχετισμού  $\tau_{\mathcal{O}}$  που θα χρησιμοποιήσουμε για τον έλεγχο της απόδοσης του αλγόριθμου θα είναι σε μονάδες sweeps και θα ισχύει

$$\tau_{\mathcal{O}} = \tau_{\mathcal{O}}^W \frac{\langle n \rangle}{N}. \quad (5.22)$$

Αρχικά πραγματοποιούμε προσομοιώσεις για  $L = 10, 20, 40, 60, 80$  και 100 στη θερμοκρασία  $\beta = 0.4407$ . Κατασκευάζουμε  $5 \times 10^6$  clusters. Τα αποτελέσματα αποθηκεύονται σε αρχεία με όνομα `outL${L}b0.4407`. Στη συνέχεια πραγματοποιούμε προσομοιώσεις με τον αλγόριθμο Metropolis και πραγματοποιούμε  $10 \times 10^6$  sweeps. Τα αποτελέσματα αποθηκεύονται σε αρχεία με όνομα `outL${L}b0.4407met`. Το παρακάτω σενάριο φλοιού κάνει τα πράγματα πιο ... ξεκούραστα

```
#!/bin/tcsh -f
set Ls      = (10 20 40 60 80 100)
set beta    = 0.4407
set nsweeps = 5000000
set start   = "-s 1 -S 3423"
# Wolf cluster algorithm:
foreach L   ($Ls)
  ./is -L $L -b $beta -n $nsweeps $start -w> outL${L}b${beta}
  # Mean cluster size <n>/N
  grep '#clu' outL${L}b${beta} | \
  awk -v L=$L 'NR>10000{print $2/(L*L)}' | \
  ./jack -d $nsweeps | grep -v '#' | \
  awk -v L=$L -v b=$beta '{print "n",L,b,$1,$2}'
end
# Metropolis algorithm
set nsweeps = 10000000
foreach L   ($Ls)
  ./is -L $L -b $beta -n $nsweeps $start > outL${L}b${beta}met
end
```

Αντιγράφουμε το αρχείο `autoc.c` από τον κατάλογο `Tools` του συνοδευτικού λογισμικού και το μεταγλωττίζουμε

```
> cc -O3 autoc.c -o autoc -lm
```

Με το παρακάτω σενάριο φλοιού μετράμε τις συναρτήσεις αυτοσυσχετισμού της μαγνήτισης  $\rho_m(t)$

```
#!/bin/tcsh -f
set Ls = (10 20 40 60 80 100)
set b = 0.4407
# Wolff
set tmax = 1000
set ndata = 5000000
foreach L ($Ls)
  set f = outL${L}b${b}
  grep -v '#' $f | \
  awk -v L=$L 'BEGIN{N=L*L}NR>100000{print ($2>0)?($2/N):(-$2/N)}' |\
  ./autoc -t $tmax -n $ndata> $f.rhom
end
# Metropolis
set tmax = 8000
set ndata = 10000000
foreach L ($Ls)
  set f = outL${L}b${b}met
  grep -v '#' $f | \
  awk -v L=$L 'BEGIN{N=L*L}NR>100000{print ($2>0)?($2/N):(-$2/N)}' |\
  ./autoc -t $tmax -n $ndata> $f.rhom
end
```

Ο μέγιστος χρόνος στη μέτρηση της  $\rho_m(t)$  είναι φυσικά πολύ μεγαλύτερος για τον αλγόριθμο Metropolis (μεταβλητή  $t_{max}$ ). Επίσης πετάμε τις πρώτες 100000 μετρήσεις. Τα αποτελέσματα τα βρίσκουμε σε αρχεία των οποίων οι καταλήξεις είναι `.rhom`. Στη συνέχεια προσαρμόζουμε την  $\rho_m(t)$  στη συνάρτηση (A'.4) με τρεις χρόνους αυτοσυσχετισμού όπως περιγράφεται στο Παράρτημα A'. Τα αποτελέσματά μας δίνονται στον Πίνακα 5.6<sup>16</sup>

Από τη σχέση (5.1) αναμένεται να έχουμε  $\tau_m \sim L^z$  όπου  $z$  ο δυναμικός κρίσιμος εκθέτης. Ο  $z$  υπολογίζεται εύκολα σε μία συνεδρία του gnuplot

```
gnuplot> tau(x) = c*x**z
gnuplot> fit tau(x) "autoc.dat" u 1:2:3 via c,z
gnuplot> plot "autoc.dat" u 1:2:3 w e title "Wolff steps", tau(x)
```

<sup>16</sup>Παρατηρούμε μια διαφορά στις μετρήσεις μας για το χρόνο αυτοσυσχετισμού για τον αλγόριθμο Metropolis σε σχέση με τις αριθμητικές τιμές που παρουσιάσαμε στο Παράρτημα A'. Η διαφορά είναι ο πενταπλασιασμός της στατιστικής μας και δείχνει πως το πραγματικό σφάλμα στον προσδιορισμό των  $\tau$  εμπεριέχει συστηματικά σφάλματα που δεν έχουμε λάβει υπ' όψη μας.

$L$	$\tau_m^W$	$\langle n \rangle / N$	$\tau_m$	$\tau_{m, \text{Metropolis}}$
10	2.18(2)	0.6124(2)	1.33(1)	16.1(1)
20	3.48(5)	0.5159(1)	1.80(3)	70.7(4)
40	5.10(6)	0.4342(2)	2.21(3)	330(6)
60	6.12(6)	0.3927(2)	2.40(2)	795(5)
80	7.33(7)	0.3653(3)	2.68(3)	1740(150)
100	8.36(6)	0.3457(1)	2.89(2)	2660(170)

Πίνακας 5.1: Οι χρόνοι αυτοσυσχετισμού της μαγνήτισης που προκύπτουν από τις προσομοιώσεις και την ανάλυση που περιγράφεται στο κείμενο. Στη 2η στήλη είναι οι χρόνοι αυτοσυσχετισμού  $\tau_m^W$  του αλγόριθμου Wolff σε μονάδα "χρόνου" την κατασκευή ενός Wolff cluster. Η 3η στήλη έχει το μέσο μέγεθος των Wolff clusters σαν ποσοστό του μεγέθους του πλέγματος  $\langle n \rangle / N$ . Η 4η στήλη έχει το  $\tau_m = \tau_m^W \langle n \rangle / N$ , το χρόνο αυτοσυσχετισμού της μαγνήτισης σε ένα Wolff sweep σύμφωνα με την εξίσωση (5.22). Και τέλος η 5η στήλη έχει τους χρόνους αυτοσυσχετισμού της μαγνήτισης για τον αλγόριθμο Metropolis για σύγκριση.

```
gnuplot> fit tau(x) "autoc.dat" u 1:6:7 via c,z
gnuplot> plot "autoc.dat" u 1:6:7 w e title "Wolff sweeps", tau(x)
gnuplot> fit tau(x) "autoc.dat" u 1:8:9 via c,z
gnuplot> plot "autoc.dat" u 1:8:9 w e title "Metropolis", tau(x)
```

όπου υπολογίζουμε το  $z$  για τον αλγόριθμο Wolff σε Wolff steps, Wolff sweeps και για τον αλγόριθμο Metropolis σε Metropolis sweeps. Τα αποτελέσματα είναι

$$\tau_m^W \sim L^z, \quad z = 0.54 \pm 0.02 \quad (5.23)$$

$$\tau_m \sim L^z, \quad z = 0.29 \pm 0.02 \quad (5.24)$$

$$\tau_{m, \text{Metropolis}} \sim L^z, \quad z = 2.21 \pm 0.02 \quad (5.25)$$

Οι αντίστοιχες γραφικές παραστάσεις δίνονται στα Σχήματα 5.9-5.11. Οι τιμές που αναμένονται από τη βιβλιογραφία είναι 0.50(1), 0.25(1) και 2.167(1) [1, 20, 27]. Καλύτερα αποτελέσματα μπορούμε να πάρουμε αυξάνοντας τη στατιστική και το μέγεθος του πλέγματος, κάτι που το αφήνουμε σαν άσκηση στον αναγνώστη.

Αξίζει τον κόπο να σημειώσουμε τη σχέση μεταξύ των δυναμικών εκθετών των εξισώσεων (5.23) και (5.24). Από την εξίσωση (5.20)  $\chi = \beta \langle n \rangle$  και από την εξίσωση (1.59)  $\chi \sim |t|^{-\gamma}$ , (1.55)  $\xi \sim |t|^{-\nu}$  και στην ψευδοκρίσιμη περιοχή  $\xi \sim L$  παίρνουμε

$$\tau_m = \tau_m^W \frac{\langle n \rangle}{L^2} \sim L^{z^W} \frac{L^{\gamma/\nu}}{L^2} = L^{z^W + \gamma/\nu - 2} \quad (5.26)$$

όπου υποθέσαμε  $\tau_m^W \sim L^{z^W}$ ,  $z^W \equiv 0.54(2)$  και  $\tau_m \sim L^z$ . Οπότε παίρνουμε

$$z = z^W + \frac{\gamma}{\nu} - 2. \quad (5.27)$$



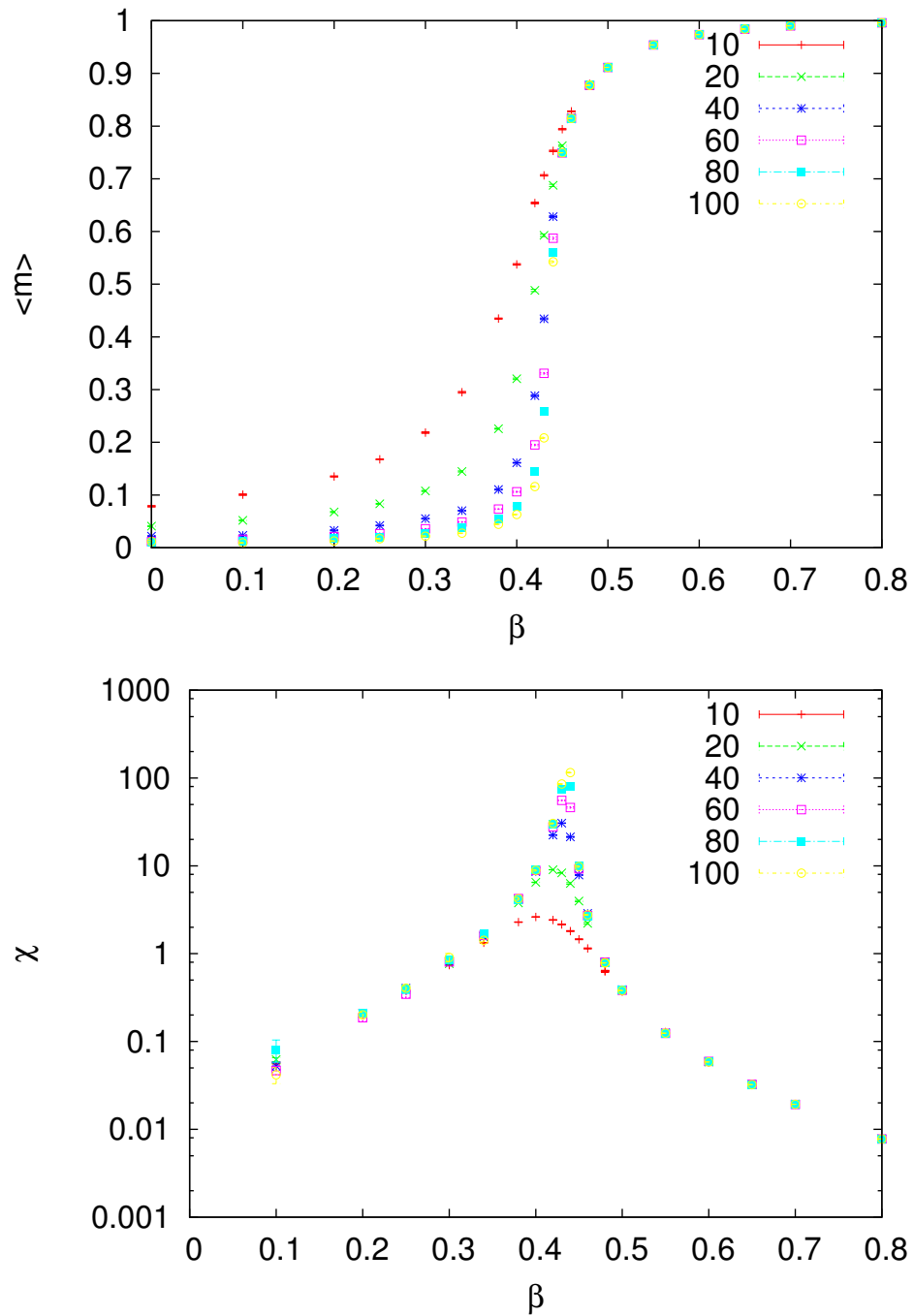
Χρησιμοποιώντας τις τιμές (1.61)  $\gamma = 7/4$ ,  $\nu = 1$  παίρνουμε

$$z = z^W - \frac{1}{4}, \quad (5.28)$$

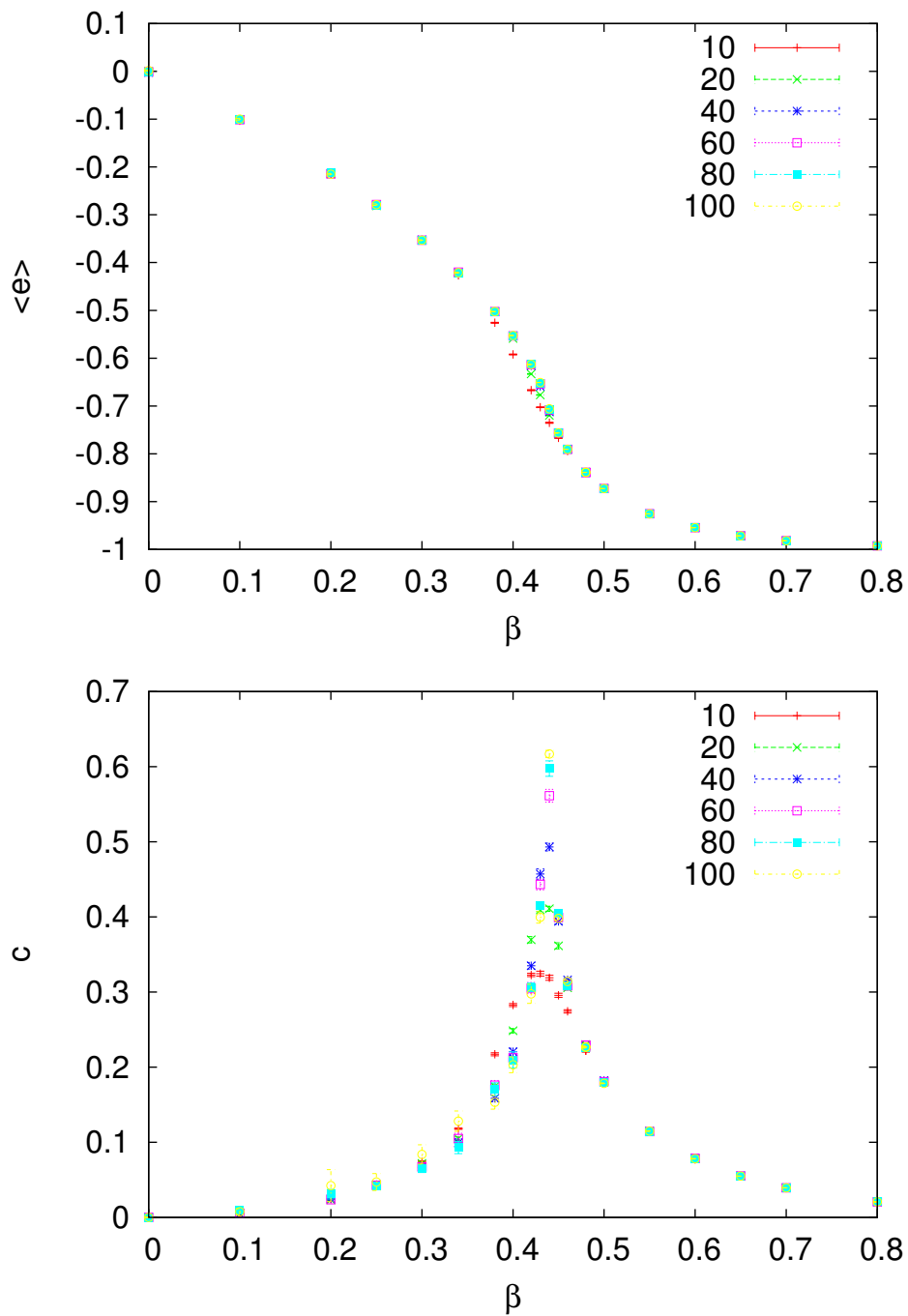
που ισχύει για τις τιμές που υπολογίσαμε, μέσα στα όρια του σφάλματος, καθώς και για τις τιμές της βιβλιογραφίας.

## 5.7 ΠΡΟΒΛΗΜΑΤΑ

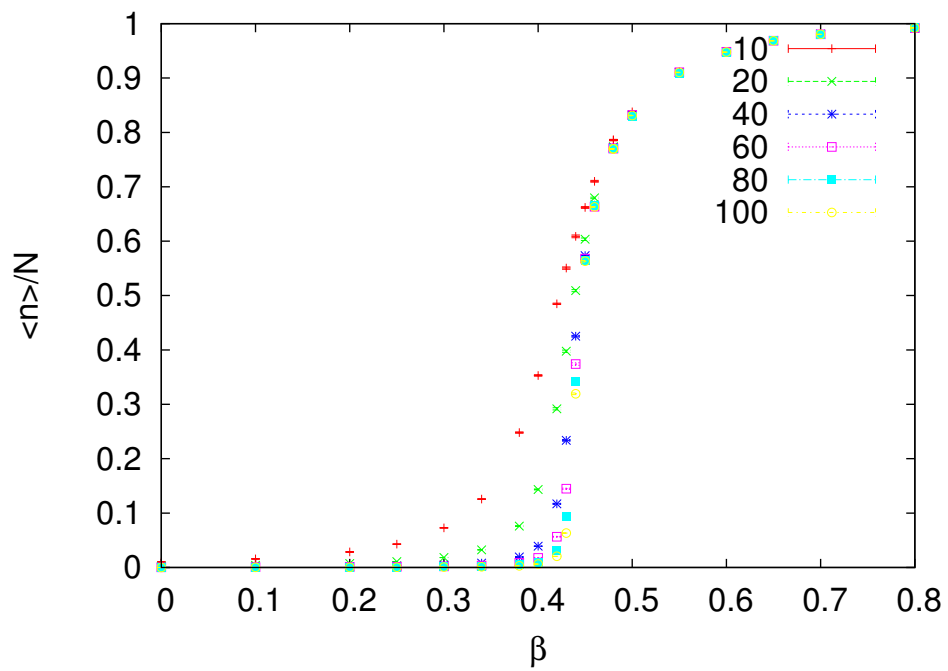
1. Υπολογίστε το μέσο λόγο αποδοχής  $a$  στον αλγόριθμο Metropolis σαν συνάρτηση της θερμοκρασίας για  $L = 10, 40, 100$ . Στις ίδιες θερμοκρασίες υπολογίστε για τον αλγόριθμο Wolff το μέσο μέγεθος των κατασκευαζόμενων clusters  $\langle n \rangle$ . Στη συνέχεια υπολογίστε τον αριθμό των Wolff clusters που ισοδυναμούν με ένα Metropolis sweep για κάθε θερμοκρασία που προσομοιώσατε. Παραστήστε γραφικά τις παραπάνω ποσότητες.
2. Στα σχήματα 5.6–5.8 προσθέστε ανάλογα σημεία για  $L = 50, 120, 140, 160, 180, 200$ .
3. Στα σχήματα 5.9–5.10 προσθέστε ανάλογα σημεία για  $L = 50, 90, 130, 150, 190, 250$ . Επαναυπολογίστε τον κρίσιμο εκθέτη  $z$  χρησιμοποιώντας και τα δικά σας δεδομένα.
4. Στο σχήμα 5.11 προσθέστε ανάλογα σημεία για  $L = 30, 50, 70, 90$ . Επαναυπολογίστε τον κρίσιμο εκθέτη  $z$  χρησιμοποιώντας και τα δικά σας δεδομένα.



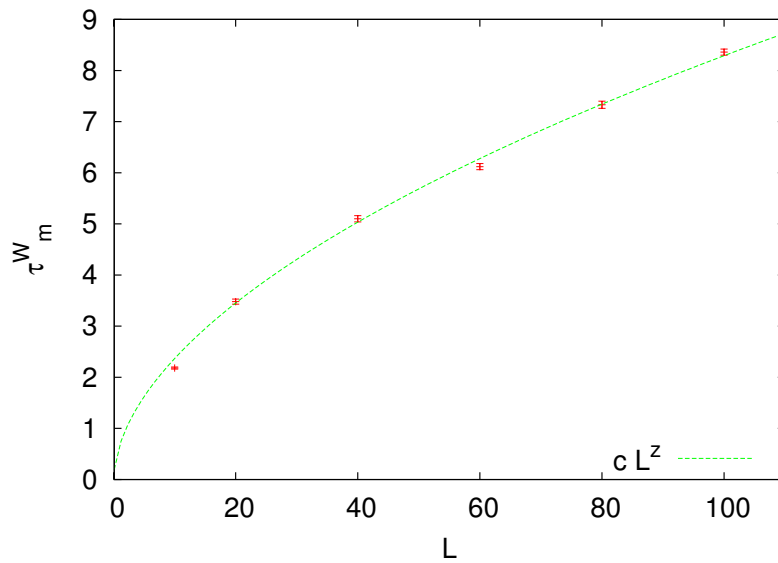
Σχήμα 5.6: Τα αποτελέσματα από τις προσομοιώσεις που γίνονται από το σε-  
νάριο φλοιού run3. Η  $\langle m \rangle$  δίνεται στο επάνω σχήμα όπου φαίνεται καθαρά η μεί-  
ωση της μαγνήτισης σε ψηλές θερμοκρασίες  $\beta \ll \beta_c$  σαν  $1/L$ . Η μαγνητική επιδε-  
κτικότητα  $\chi$  αναπαρίσταται γραφικά στο κάτω σχήμα. Μακριά από την κρίσιμη  
περιοχή είναι σχεδόν ανεξάρτητη από το μέγεθος του πλέγματος, ενώ στην ψευ-  
δοκρίσιμη περιοχή έχουμε γρήγορη αύξηση όπως αναμένεται από την εξίσωση  
(1.59).



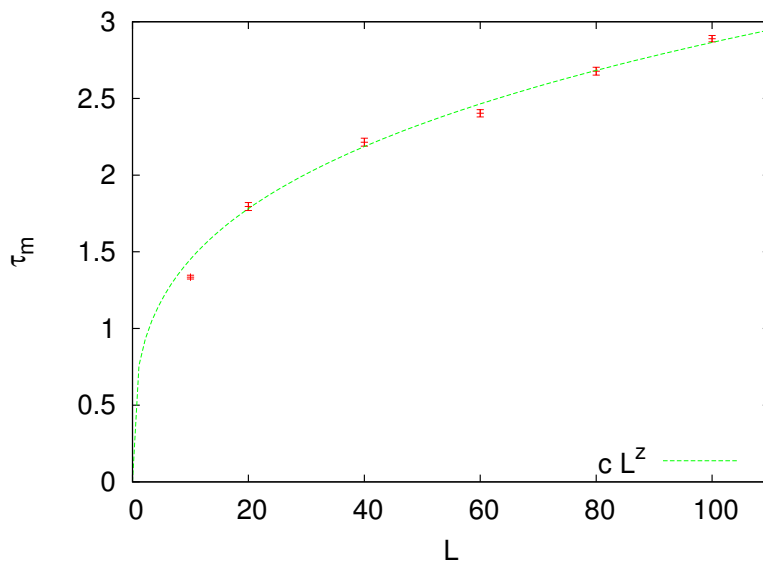
Σχήμα 5.7: Τα αποτελέσματα από τις προσομοιώσεις που γίνονται από το σε-  
νάριο φλοιού `run3`. Η  $\langle e \rangle$  δίνεται στο επάνω σχήμα. Η ειδική θερμότητα  $c$  αναπαρί-  
σταται γραφικά στο κάτω σχήμα. Μακριά από την κρίσιμη περιοχή είναι σχεδόν  
ανεξάρτητη από το μέγεθος του πλέγματος, ενώ στην ψευδοκρίσιμη περιοχή  
έχουμε αύξηση όπως αναμένεται από την εξίσωση (1.57).



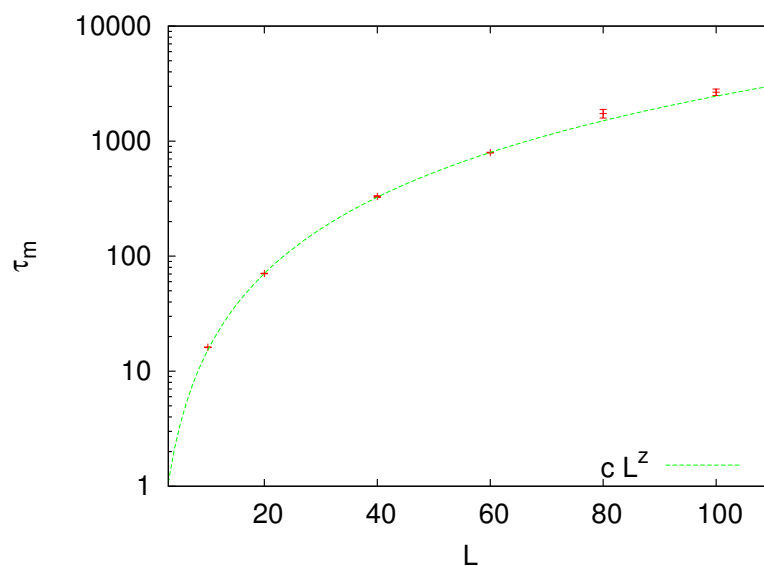
Σχήμα 5.8: Τα αποτελέσματα από τις προσομοιώσεις που γίνονται από το σε-  
νάριο φλοιού `run3`. Δίνεται η γραφική παράσταση του μέσου μεγέθους των Wolff  
clusters σαν ποσοστό του συνολικού πλέγματος  $\langle n \rangle / N$ .



Σχήμα 5.9: Οι χρόνοι αυτοσυσχετισμού  $\tau_m^W$  της μαγνήτισης για τον αλγόριθμο του Wolff για  $\beta = 0.4407$ . Η μονάδα του χρόνου είναι ένα Wolff cluster. Ο δυναμικός κρίσιμος εκθέτης βρίσκεται με την προσαρμογή της συνάρτησης  $cL^z$  από όπου προκύπτει  $z = 0.54(2)$ .



Σχήμα 5.10: Οι χρόνοι αυτοσυσχετισμού  $\tau_m$  της μαγνήτισης για τον αλγόριθμο του Wolff για  $\beta = 0.4407$ . Η μονάδα του χρόνου είναι ένα Wolff sweep. Ο δυναμικός κρίσιμος εκθέτης βρίσκεται με την προσαρμογή της συνάρτησης  $cL^z$  από όπου προκύπτει  $z = 0.29(2)$ .



Σχήμα 5.11: Οι χρόνοι αυτοσυσχετισμού  $\tau_{m,\text{Metropolis}}$  της μαγνήτισης για τον αλγόριθμο Metropolis για  $\beta = 0.4407$ . Η μονάδα του χρόνου είναι ένα Metropolis sweep. Ο δυναμικός κρίσιμος εκθέτης βρίσκεται με την προσαρμογή της συνάρτησης  $cL^z$  από όπου προκύπτει  $z = 2.21(2)$ .

## ΚΕΦΑΛΑΙΟ 6

# ΚΡΙΣΙΜΟΙ ΕΚΘΕΤΕΣ

### 6.1 ΕΙΣΑΓΩΓΗ

Ήδη τονίσαμε σε προηγούμενα κεφάλαια πως ένα φυσικό σύστημα το οποίο εμφανίζει μία συνεχή μετάβαση φάσης κατά την οποία καθώς  $\beta \rightarrow \beta_c$  ή ισοδύναμα καθώς η αδιάστατη ή ανηγμένη θερμοκρασία<sup>1</sup>

$$t \equiv \frac{\beta_c - \beta}{\beta_c} \rightarrow 0, \quad (6.1)$$

το μήκος συσχετισμού στο θερμοδυναμικό όριο  $\xi \equiv \xi(\beta, L = \infty)$  απειρίζεται σύμφωνα με τη σχέση

$$\xi \sim |t|^{-\nu} \quad (\nu = 1 \text{ για } 2\text{d-Ising}), \quad (6.2)$$

βρίσκεται σε μία κλάση παγκοσμιότητας (universality class) η οποία χαρακτηρίζεται από κρίσιμους εκθέτες. Οι κρίσιμοι εκθέτες περιγράφουν την κύρια μη αναλυτική συμπεριφορά των φυσικών ποσοτήτων που παρουσιάζεται ασυμπτωτικά στο θερμοδυναμικό όριο<sup>2</sup>  $L \rightarrow \infty$  όταν  $t \rightarrow 0$ . Αυτό που χαρακτηρίζει μία κλάση παγκοσμιότητας είναι ότι παρόλο που αποτελείται από συστήματα που δίνονται από μικροσκοπικές Χαμιλτονιανές διαφορετικές μεταξύ τους, αυτά έχουν ίδιους κρίσιμους εκθέτες. Έτσι λ.χ. στο πρότυπο Ising αν αλλάξουμε την αλληλεπίδραση πλησιεστέρων γειτόνων προσθέτοντας όρους από πιο απομακρυσμένες πλεγματικές θέσεις (πάντα όμως σε απόσταση  $\ll \xi$ ) ή αν αλλάξουμε την τοπολογία του πλέγματος από τετραγωνικό σε τριγωνικό, εξαγωνικό κλπ θα παραμείνουμε στην ίδια κλάση παγκοσμιότητας. Έτσι παρουσιάζεται η πρακτική ευκολία ένα πολύπλοκο φυσικό σύστημα να

<sup>1</sup>Συνήθως στη βιβλιογραφία ορίζεται  $t = (T - T_c)/T_c$  αλλά για  $t \ll 1$  οι δύο ορισμοί είναι σχεδόν ισοδύναμοι (διαφέρουν κατά ένα όρο  $\sim t^2$ ).

<sup>2</sup>Προσοχή: πρώτα  $L \rightarrow \infty$  και μετά  $t \rightarrow 0$ .

$L$	$\gamma(t < 0)$	$\gamma(t > 0)$
40	1.7598(44)	1.730(17)
60	1.7455(24)	1.691(14)
80	1.7409(21)	1.737(12)
100	1.7420(24)	1.7226(75)
120	1.7390(15)	1.7725(69)
140	1.7390(23)	1.7354(72)
160	1.7387(10)	1.746(17)
200	1.7380(11)	1.759(15)
500	1.7335(8)	1.7485(83)

Πίνακας 6.1: Υπολογισμός του κρίσιμου εκθέτη  $\gamma$  από την προσαρμογή των δεδομένων των Σχημάτων 6.1 και 6.2. Στην πρώτη στήλη προσαρμόζονται τα δεδομένα για  $\beta > \beta_c(t < 0)$  και στη δεύτερη τα δεδομένα για  $\beta < \beta_c(t > 0)$ . Στις παρενθέσεις δίνονται τα στατιστικά σφάλματα των προσαρμογών και όχι τα συστηματικά που προέρχονται από την επιλογή του διαστήματος της προσαρμογής. Η αναμενόμενη τιμή είναι  $\gamma = 7/4$ .

μπορούμε να το μελετήσουμε στην περιοχή της μετάβασης φάσης με το απλούστερο δυνατό μοντέλο που ανήκει στην ίδια κλάση παγκοσμιοτήτας. Για παράδειγμα η μετάβαση φάσης υγρό/ατμός κοντά στο τρικρίσιμο σημείο ανήκει στην ίδια κλάση παγκοσμιοτήτας με το πρότυπο Ising στο επίπεδο.

Στο πρότυπο Ising είδαμε στο Κεφάλαιο 1 τους κρίσιμους εκθέτες

$$\chi \sim |t|^{-\gamma}, \quad \gamma = 7/4, \quad (6.3)$$

$$c \sim |t|^{-\alpha}, \quad \alpha = 0 \quad \text{και} \quad (6.4)$$

$$\langle m \rangle \sim |t|^\beta \quad t < 0, \quad \beta = 1/8. \quad (6.5)$$

Η παραπάνω συμπεριφορά παρουσιάζεται μόνο όταν  $L \rightarrow \infty$ . Για πεπερασμένο πλέγμα όλες οι ποσότητες είναι αναλυτικές αφού υπολογίζονται από την αναλυτική συνάρτηση επιμερισμού  $Z$  που δίνεται από την (1.53). Στο πεπερασμένο πλέγμα όσο  $1 \ll \xi \ll L$  το μοντέλο συμπεριφέρεται (προσεγγιστικά) όπως το άπειρο σύστημα ενώ καθώς  $\beta \approx \beta_c$  και το  $\xi \sim L$  παρουσιάζονται φαινόμενα επίδρασης πεπερασμένου μεγέθους (finite size effects). Οι διακυμάνσεις  $\chi$ ,  $c$  στο πεπερασμένο πλέγμα παρουσιάζουν μέγιστο για μια ψευδοκρίσιμη θερμοκρασία  $\beta_c(L)$  η οποία φυσικά<sup>3</sup>

$$\lim_{L \rightarrow \infty} \beta_c(L) = \beta_c. \quad (6.6)$$

<sup>3</sup>Κάθε φυσική ποσότητα μπορεί να έχει (ελαφρά) διαφορετική ψευδοκρίσιμη θερμοκρασία οπότε μπορεί να γράφουμε  $\beta_c^x(L)$ ,  $\beta_c^y(L)$ .



$L$	$\beta(t < 0)$	$\beta_+(t > 0)$
40	0.1101(7)	0.1122(29)
60	0.1129(5)	0.1102(19)
80	0.1147(5)	0.1118(21)
100	0.1175(3)	0.1170(11)
120	0.1167(4)	0.1172(16)
140	0.1190(2)	0.1187(19)
160	0.1191(4)	0.1134(20)
200	0.1205(10)	0.1138(24)
500	0.1221(2)	0.1294(50)

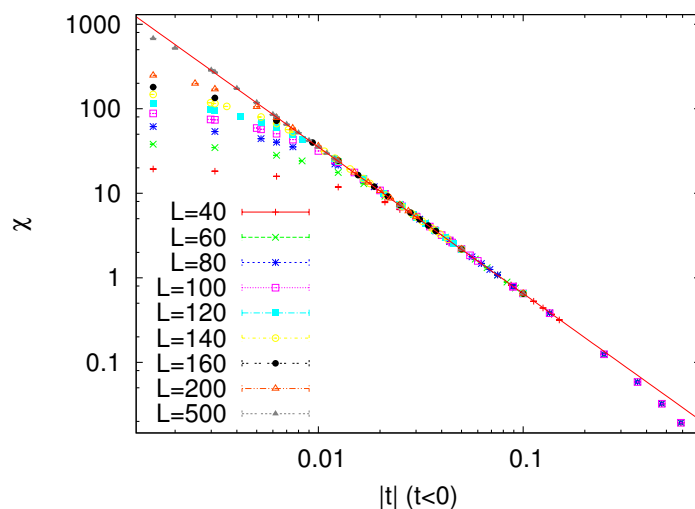
Πίνακας 6.2: Υπολογισμός του κρίσιμου εκθέτη  $\beta$  από την προσαρμογή των δεδομένων των Σχημάτων 6.3. Στην πρώτη στήλη προσαρμόζονται τα δεδομένα για  $\beta > \beta_c(t < 0)$  και στη δεύτερη τα δεδομένα για  $\beta < \beta_c(t > 0)$ . Στις παρενθέσεις δίνονται τα στατιστικά σφάλματα των προσαρμογών και όχι τα συστηματικά που προέρχονται από την επιλογή του διαστήματος της προσαρμογής. Η αναμενόμενη τιμή είναι  $\beta = \beta_+ = 1/8$ .

$L$	$\gamma/\nu$	$\beta/\nu$
40-100	1.754(1)	0.1253(1)
140-1000	1.740(2)	0.1239(3)
40-1000	1.749(1)	0.1246(1)

Πίνακας 6.3: Οι κρίσιμοι εκθέτες  $\gamma/\nu$  και  $\beta/\nu$  που δίνονται από τη μελέτη βάθμισης μεγέθους των σχέσεων (6.7) και (6.9). Στην πρώτη στήλη δίνονται τα διαστήματα στις τιμές του  $L$  που θεωρήθηκαν στην προσαρμογή των τιμών των  $\chi(\beta_c, L)$  και  $\langle m \rangle(\beta_c, L)$  σε συναρτήσεις της μορφής  $aL^g$  και δίπλα οι αντίστοιχες τιμές για (την απόλυτη τιμή) των εκθετών  $g$  που προκύπτουν.

$L$	$\beta_c(L)$	$\chi_{\max}$	$\beta'_c(L)$	$c_{\max}$
40	0.4308(4)	30.68(4)	0.437(1)	0.5000(20)
60	0.4342(2)	62.5(1)	0.4382(7)	0.5515(15)
80	0.4357(2)	103.5(1)	0.4388(5)	0.5865(12)
100	0.4368(1)	153.3(2)	0.4396(2)	0.6154(18)
120	0.4375(1)	210.9(2)	0.4396(4)	0.6373(20)
140	0.43793(13)	276.2(4)	0.4397(5)	0.6554(18)
160	0.4382(1)	349.0(5)	0.4398(4)	0.6718(25)
200	0.43870(7)	516.3(7)	0.4399(2)	0.6974(17)
500	0.43988(4)	2558(5)	0.44038(8)	0.7953(25)
1000	0.44028(4)	8544(10)	0.44054(8)	0.8542(36)

Πίνακας 6.4: Ο υπολογισμός των ψευδοκρίσιμων θερμοκρασιών  $\beta_c(L)$  και  $\beta'_c(L)$  που προκύπτουν από τον υπολογισμό της θέσης των μέγιστων της μαγνητικής επιδεκτικότητας  $\chi_{\max}$  και της ειδικής θερμότητας  $c_{\max}$  αντίστοιχα. Δίνονται και οι αντίστοιχες τιμές των μέγιστων.



Σχήμα 6.1: Οι συναρτήσεις της μαγνητικής επιδεκτικότητας  $\chi(t, L)$  στην περιοχή θερμοκρασιών που παρουσιάζεται βάρθμιση σύμφωνα με την σχέση (6.3). Η ευθεία γραμμή είναι προσαρμογή στη σχέση αυτή για το μεγαλύτερο πλέγμα. Παρατηρήστε την ελάττωση των finite size effects καθώς το  $L$  μεγαλώνει και το διάστημα προσαρμογής επεκτείνεται σε ολοένα μικρότερα  $|t|$ . Τα δεδομένα είναι για θερμοκρασίες  $\beta > \beta_c(t < 0)$  όπου το κρίσιμο σημείο προσεγγίζεται από τη φάση τάξης.

Θυμίζουμε στον αναγνώστη ότι για το πρότυπο Ising στο τετραγωνικό πλέγμα έχουμε  $\beta_c = \log(1 + \sqrt{2})/2$ .

Για  $\beta \approx \beta_c(L)$ , λόγω της (6.2), έχουμε  $\xi(t, L) \sim L \implies |t| \sim L^{-1/\nu}$  οπότε οι σχέσεις (6.3)–(6.5) γίνονται

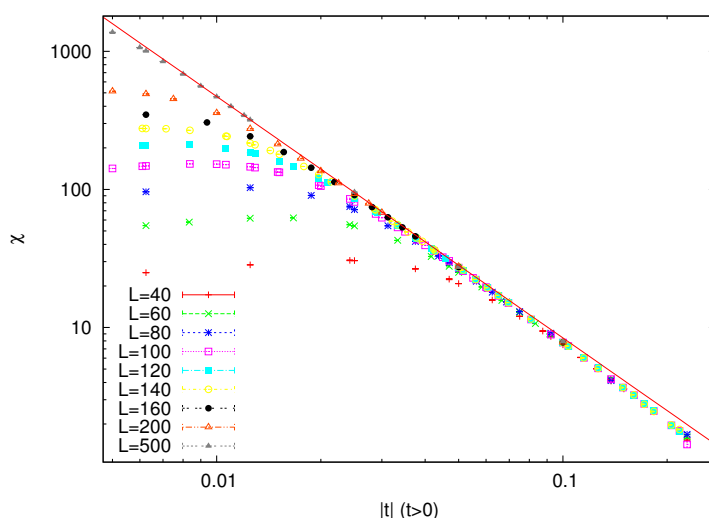
$$\chi \sim L^{\gamma/\nu}, \quad (6.7)$$

$$c \sim L^{\alpha/\nu} \quad \text{και} \quad (6.8)$$

$$m \sim L^{-\beta/\nu}. \quad (6.9)$$

Τονίζουμε ότι οι παραπάνω σχέσεις αφορούν όλη την ψευδοκρίσιμη περιοχή, οπότε στην πράξη μπορούμε να πάρουμε είτε τις τιμές των ποσοτήτων όταν  $\beta = \beta_c(L)$  είτε όταν  $\beta = \beta_c$ <sup>4</sup>. Στις επόμενες παραγράφους θα δείξουμε πώς να υπολογίσουμε τους κρίσιμους εκθέτες από τις σχέσεις (6.3)–(6.5) και (6.7)–(6.9).

<sup>4</sup>Στο όριο  $L \rightarrow \infty$  η διαφορά δεν έχει σημασία. Η ταχύτητα σύγκλισης στις ασυμπτωτικές αυτές σχέσεις μπορεί να διαφέρει.

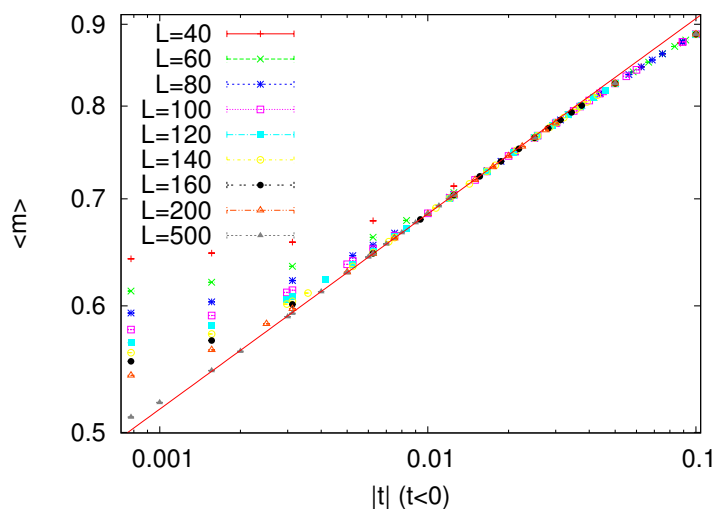


Σχήμα 6.2: Οι συναρτήσεις της μαγνητικής επιδεκτικότητας  $\chi(t, L)$  στην περιοχή θερμοκρασιών που παρουσιάζεται βάθμιση σύμφωνα με την σχέση (6.3). Η ευθεία γραμμή είναι προσαρμογή στη σχέση αυτή για το μεγαλύτερο πλέγμα. Παρατηρήστε την ελάττωση των finite size effects καθώς το  $L$  μεγαλώνει και το διάστημα προσαρμογής επεκτείνεται σε ολοένα μικρότερα  $|t|$ . Τα δεδομένα είναι για θερμοκρασίες  $\beta < \beta_c(t > 0)$  όπου το κρίσιμο σημείο προσεγγίζεται από τη φάση αταξίας. Τα finite size effects είναι μεγαλύτερα από την περίπτωση  $t < 0$  λόγω της επίδρασης των μεγάλων διακυμάνσεων στο ψευδοκρίσιμο σημείο  $\beta_c(L) < \beta_c$ .

## 6.2 ΒΑΘΜΙΣΗ ΘΕΡΜΟΚΡΑΣΙΑΣ

Στην παράγραφο αυτή θα ελέγξουμε το βαθμό στον οποίο οι σχέσεις (6.3)–(6.5) μπορούν να χρησιμοποιηθούν για τον υπολογισμό των κρίσιμων εκθετών  $\gamma$ ,  $\alpha$  και  $\beta$ . Το συμπέρασμα που θα βγάλουμε είναι πως οι σχέσεις αυτές δεν είναι η καλύτερη επιλογή για τον εν λόγω υπολογισμό και ότι θα πρέπει να αποφεύγονται αν είναι δυνατόν<sup>5</sup>. Για να δούμε καθαρή βάθμιση, θα πρέπει να προσομοιώσουμε σε πολύ μικρά  $t \ll 1$  και αρκετά μεγάλο  $L$  λόγω των μεγάλων φαινομένων επίδρασης πεπερασμένου μεγέθους (finite size effects). Επίσης υπάρχουν παραδείγματα που οι τιμές που παίρνει κανείς εξαρτώνται ισχυρά από το διάστημα προσαρμογής των δεδομένων με αποτέλεσμα να έχουμε μεγάλα συστηματικά σφάλματα από τη διαδικασία προσαρμογής (fitting)

<sup>5</sup> Παρατηρήστε πως στο συγκεκριμένο μοντέλο η κρίσιμη θερμοκρασία μας είναι επακριβώς γνωστή. Όταν δεν την γνωρίζουμε, τότε υπεισέρχονται επί πλέον συστηματικά σφάλματα που δυσχεραίνουν τον υπολογισμό. Για τον αριθμητικό υπολογισμό της κρίσιμης θερμοκρασίας θα μιλήσουμε σε επόμενη παράγραφο.



Σχήμα 6.3: Οι συναρτήσεις της μαγνήτισης  $\langle m \rangle(t, L)$  στην περιοχή θερμοκρασιών που παρουσιάζεται βάρθιση σύμφωνα με την σχέση (6.5). Η ευθεία γραμμή είναι προσαρμογή στη σχέση αυτή για το μεγαλύτερο πλέγμα. Παρατηρήστε την ελάττωση των finite size effects καθώς το  $L$  μεγαλώνει και το διάστημα προσαρμογής επεκτείνεται σε ολοένα μικρότερα  $|t|$ . Τα δεδομένα είναι για θερμοκρασίες  $\beta > \beta_c(t < 0)$  όπου το κρίσιμο σημείο προσεγγίζεται από τη φάση τάξης.

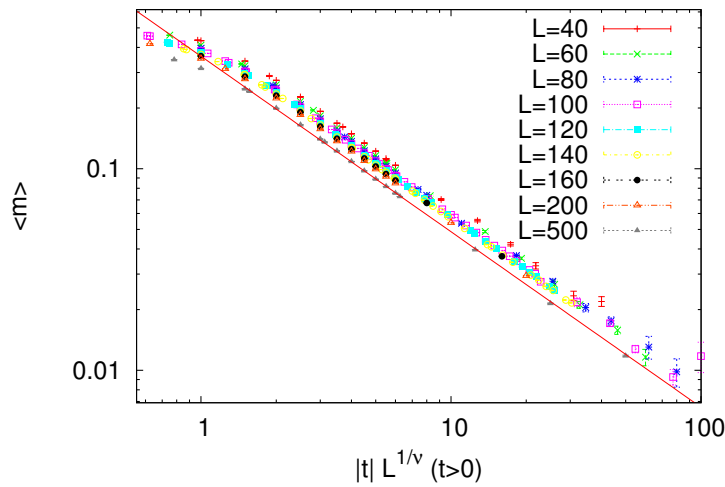
ή ακόμα και λάθος αποτελέσματα<sup>6</sup>

Για το λόγο αυτό προσομοίωσαμε το πρότυπο Ising για  $L = 40, 60, 80, 100, 120, 140, 160, 200$  και  $500$ . Διαλέξαμε θερμοκρασίες που αντιστοιχούν σε αρκετά μικρά  $t$  γύρω από την κρίσιμη θερμοκρασία. Στο συνοδευτικό λογισμικό θα βρείτε τα shell scripts που χρησιμοποιήθηκαν για το λόγο αυτό.

Αρχικά υπολογίζουμε τον εκθέτη  $\gamma$  από τη σχέση (6.3). Προσαρμόζουμε στην κατάλληλη περιοχή του  $|t|$  την τιμή της  $\chi(t)$  για κάθε τιμή του  $L$  στη συνάρτηση  $a|t|^{-\gamma}$  που έχει δύο ανεξάρτητες παραμέτρους  $\gamma, a$ . Η επιλογή του διαστήματος γίνεται με τον εντοπισμό γραμμικής συμπεριφοράς της  $\chi(t)$  σε διάγραμμα με λογαριθμική κλίμακα και στους δύο άξονες<sup>7</sup>. Παρατηρούμε ότι για μεγάλο  $|t|$  έχουμε απόκλιση από τη γραμμική συμπεριφορά ενώ για πολύ μικρό  $|t|$  παρουσιάζονται σφάλματα από τα φαινόμενα επίδρασης πεπερασμένου μεγέθους όταν  $\xi \approx L$ . Καθώς το  $L$  μεγαλώνει τα φαινόμενα επίδρασης πεπερασμένου μεγέθους μειώνονται και τα

<sup>6</sup>Στο [1] αναφέρεται πως στο random field Ising model παρουσιάζεται ψευδο-βάρθιση για ένα διάστημα του  $t$  ενώ για πολύ μικρότερο  $t$  έχουμε crossover σε διαφορετική βάρθιση που δίνει το σωστό εκθέτη. Δείτε επίσης τα [28], [29].

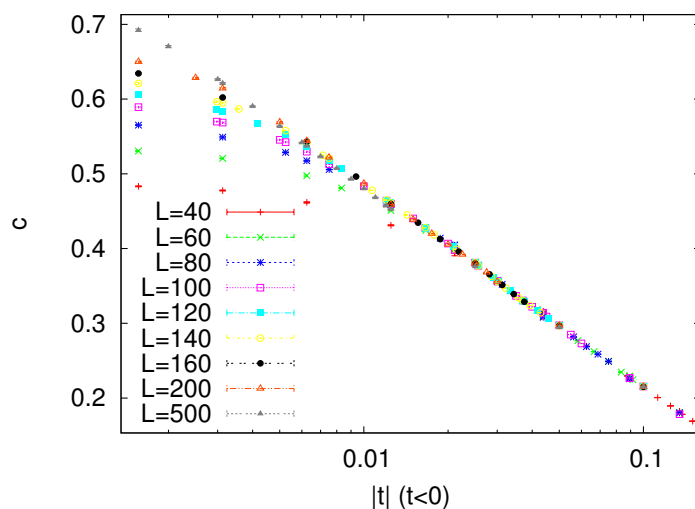
<sup>7</sup>Η προσαρμογή μπορεί να γίνει και με γραμμική προσαρμογή σε ευθεία των σημείων  $(\log |t_i|, \log \chi(t_i))$  με απλή μέθοδο ελαχίστων τετραγώνων.



Σχήμα 6.4: Οι συναρτήσεις της μαγνήτισης  $\langle m \rangle(t, L)$  στην περιοχή θερμοκρασιών που παρουσιάζεται βύθιση σύμφωνα με την σχέση (6.12). Η ευθεία γραμμή είναι προσαρμογή στη σχέση αυτή για το μεγαλύτερο πλέγμα. Παρατηρήστε την ελάττωση των finite size effects καθώς το  $L$  μεγαλώνει και το διάστημα προσαρμογής επεκτείνεται σε ολοένα μικρότερα  $|t|$ . Τα δεδομένα είναι για θερμοκρασίες  $\beta < \beta_c(t > 0)$  όπου το κρίσιμο σημείο προσεγγίζεται από τη φάση αταξίας.

δεδομένα πλησιάζουν κοντύτερα προς την ασυμπτωτική συμπεριφορά  $|t|^{-\gamma}$  για ολοένα και μικρότερο  $|t|$ . Τα αποτελέσματα είναι καθαρότερα για  $\beta > \beta_c(t < 0)$  γιατί για  $t > 0$  οι διακυμάνσεις είναι ισχυρότερες κοντά στην ψευδοκρίσιμη θερμοκρασία  $\beta_c(L) < \beta_c$  και έχουμε μεγαλύτερα φαινόμενα επίδρασης πεπερασμένου μεγέθους.

Στον πίνακα 6.1 σημειώνουμε τα αποτελέσματά μας για τον εκθέτη  $\gamma$  για όλες τις τιμές του  $L$  που μετρήσαμε. Τα σφάλματα που σημειώνονται στον πίνακα είναι τα απλά στατιστικά σφάλματα της προσαρμογής που έδωσε το αναφερόμενο αποτέλεσμα και είναι πολύ μικρότερο από το συστηματικό σφάλμα της προσαρμογής. Το τελευταίο προέρχεται κυρίως από την εκτίμηση του διαστήματος θερμοκρασιών  $t$  που θα συμπεριλάβουμε στην προσαρμογή. Μία συστηματική αντιμετώπιση της εκτίμησης των σφαλμάτων αυτών είναι να αλλάζουμε το διάστημα αυτό και να συμπεριλάβουμε στις δυνατές τιμές αυτές που προκύπτουν από όλες τις αποδεκτές προσαρμογές στη συνάρτηση που επιλέξαμε. Καμιά φορά αυτό δίνει υπερεκτιμημένο σφάλμα και έγκειται στην "τέχνη" και εμπειρία μας να αποφασίσουμε τις τιμές που θα κρατήσουμε. Για παράδειγμα, από τα Σχήματα 6.1 και 6.2 παρατηρούμε ότι το διάστημα της προσαρμογής γίνεται πιο ευδιάκριτο



Σχήμα 6.5: Οι συναρτήσεις της ειδικής θερμότητας  $c(t, L)$  στην περιοχή θερμοκρασιών που παρουσιάζεται βάθμιση σύμφωνα με την σχέση (6.16). Παρατηρήστε ότι μόνο ο άξονας  $|t|$  είναι σε λογαριθμική κλίμακα. Τα δεδομένα είναι για θερμοκρασίες  $\beta > \beta_c(t < 0)$  όπου το κρίσιμο σημείο προσεγγίζεται από τη φάση τάξης.

μελετώντας την  $\chi(t)$  για ολοένα και μεγαλύτερα  $L$ . Καθώς το  $L$  μεγαλώνει, τα σημεία πλησιάζουν προς την ασυμπτωτική καμπύλη ολοένα και περισσότερο. Επίσης οι λογικές τιμές για τις παραμέτρους που υπεισέρχονται στην προσαρμογή μπορεί να αποτελέσει και αυτό ένα κριτήριο αποδοχής ή όχι του αποτελέσματος μιας προσαρμογής (λ.χ. να έχει η σταθερά  $a$  τιμές  $\sim 1$ ). Λαμβάνοντας υπόψη τα παραπάνω μπορούμε να αναφέρουμε τα αποτελέσματα

$$\gamma = 1.74 \pm 0.02 \quad (t < 0), \quad (6.10)$$

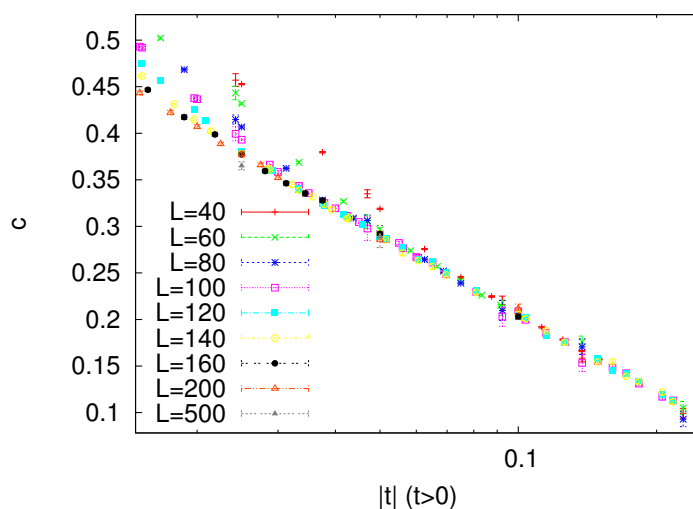
$$\gamma = 1.73 \pm 0.04 \quad (t > 0). \quad (6.11)$$

Στη συνέχεια υπολογίζουμε τον κρίσιμο εκθέτη  $\beta$  από την εξίσωση (6.5). Η σχέση αυτή ισχύει καθώς προσεγγίζουμε το κρίσιμο σημείο από την ψυχρή περιοχή  $\beta > \beta_c$  ή  $t < 0$ . Στο θερμοδυναμικό όριο η μαγνήτιση είναι παντού μηδέν για κάθε  $\beta < \beta_c$ . Παρόλα αυτά, στο πεπερασμένο πλέγμα  $\langle m \rangle > 0$  και είναι λογικό να περιμένουμε μία σχέση βάθμισης της μορφής

$$\langle m \rangle \sim |t|^{\beta_+ - 1}, \quad t > 0. \quad (6.12)$$

όπου ο ορισμός του  $\beta_+$  γίνεται έτσι γιατί όπως θα δούμε

$$\beta_+ = \beta = 1/8. \quad (6.13)$$



Σχήμα 6.6: Οι συναρτήσεις της ειδικής θερμότητας  $c(t, L)$  στην περιοχή θερμοκρασιών που παρουσιάζεται βαθμίσωση σύμφωνα με την σχέση (6.16). Παρατηρήστε ότι μόνο ο άξονας  $|t|$  είναι σε λογαριθμική κλίμακα. Τα δεδομένα είναι για θερμοκρασίες  $\beta < \beta_c(t > 0)$  όπου το κρίσιμο σημείο προσεγγίζεται από τη φάση αταξίας.

Ακολουθούμε τη διαδικασία προσαρμογής των δεδομένων που περιγράψαμε παραπάνω. Τα αποτελέσματα για τους εκθέτες  $\beta$  και  $\beta_+$  καταγράφονται στον Πίνακα 6.1. Λαμβάνοντας υπ' όψη τα συστηματικά σφάλματα που περιγράψαμε παραπάνω βρίσκουμε ότι

$$\beta = 0.121 \pm 0.003 \quad t < 0, \quad (6.14)$$

$$\beta_+ = 0.120 \pm 0.007 \quad t < 0, \quad (6.15)$$

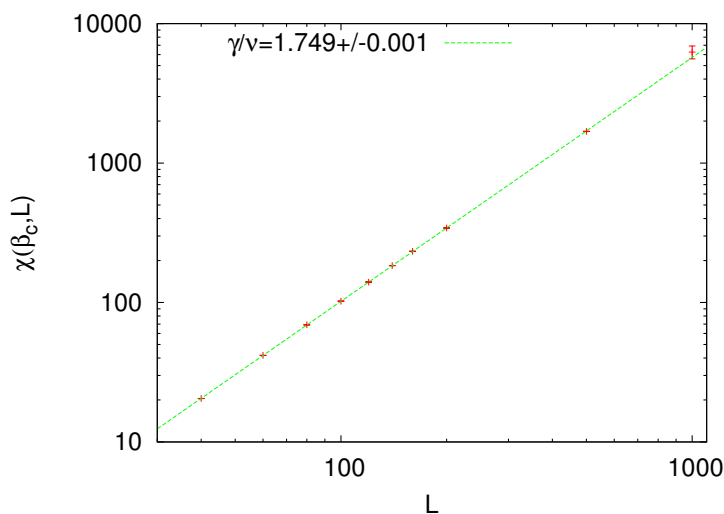
που πρέπει να συγκριθεί με την αναμενόμενη τιμή  $\beta = \beta_+ = 1/8$ .

Η περίπτωση του εκθέτη  $\alpha$  χρειάζεται ιδιαίτερη προσοχή. Η τιμή που αναμένεται είναι  $\alpha = 0$ . Αυτό δε σημαίνει ότι  $c \sim$  σταθ. αλλά ότι<sup>8</sup>

$$c \sim |\log |t||. \quad (6.16)$$

Στην περίπτωση αυτή βρίσκουμε ότι τα δεδομένα προσαρμόζονται καλύτερα στην παραπάνω σχέση παρά σε μια σχέση δύναμης όπως στην περίπτωση της μαγνήτισης και τις μαγνητικής επιδεικτικότητας. Αυτό μπορεί να φανεί εποπτικά κάνοντας τη γραφική παράσταση σε λογαριθμική κλίμακα μόνο κατά τον άξονα  $|t|$  και να συγκρίνουμε με την ανάλογη γραφική παράσταση όπου και οι

<sup>8</sup> Αυτό δεν αποκλείει η κύρια συμπεριφορά να είναι μια δύναμη του λογάριθμου ή λογάριθμος του λογάριθμου κλπ. Αυτό θα πρέπει να μελετηθεί με προσοχή.



Σχήμα 6.7: Οι τιμές  $\chi(\beta_c, L)$  στην κρίσιμη θερμοκρασία για διαφορετικές τιμές του  $L$ . Οι κλίμακες στους άξονες είναι λογαριθμικές και η γραμμική σχέση των δεδομένων που προκύπτει, αποδεικνύει τη σχέση δύναμης  $\chi(\beta_c, L) = cL^g$ . Με προσαρμογή των δεδομένων στην συνάρτηση αυτή, ο υπολογισμός του εκθέτη  $g$  δίνει τον κρίσιμο εκθέτη  $\gamma/\nu$ , όπως προκύπτει από τη σχέση (6.7).

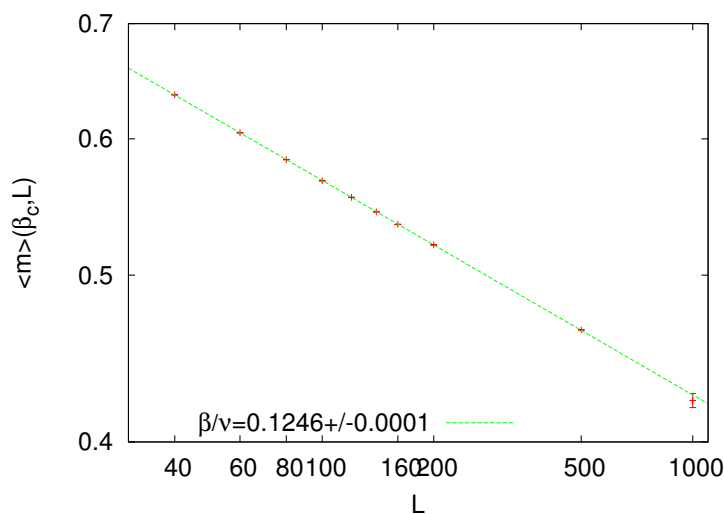
δύο άξονες είναι σε λογαριθμική κλίμακα. Βρίσκουμε ότι τα δεδομένα μας ακολουθούν μια ευθεία στην πρώτη περίπτωση ενώ στη δεύτερη όχι. Μια προσεκτική μελέτη θα συγκρίνει την ποιότητα των προσαρμογών στις δύο συναρτήσεις και θα επιλέξει το καλύτερο μοντέλο. Το αφήνουμε σαν άσκηση στον αναγνώστη.

### 6.3 ΒΑΘΜΙΣΗ ΠΕΠΕΡΑΣΜΕΝΟΥ ΜΕΓΕΘΟΥΣ

Στην παράγραφο αυτή θα υπολογίσουμε τους κρίσιμους εκθέτες με τη βοήθεια των σχέσεων (6.7)-(6.9), δηλ. με τη βάθμιση που παρουσιάζουν ασυμπτωτικά οι  $\chi(\beta = \beta_c, L)$ ,  $c(\beta = \beta_c, L)$  και  $\langle m \rangle(\beta = \beta_c, L)$  καθώς το μέγεθος του συστήματος αυξάνει. Η διεθνής ορολογία για τη μέθοδο αυτή του υπολογισμού είναι “finite size scaling”.

Η σχέση (6.7) δίνει τον εκθέτη  $\gamma/\nu$ . Για τον υπολογισμό του, χρησιμοποιούμε τις τιμές τις μαγνητικής επιδεκτικότητας στη γνωστή θερμοκρασία  $\beta_c$  για διαφορετικές τιμές του  $L$ . Με προσαρμογή των τιμών  $\chi(\beta_c, L)$  στη συνάρτηση  $aL^g$  υπολογίζουμε τις σταθερές  $a$  και  $g$  και συγκρίνουμε την τελευταία με την αναμενόμενη τιμή  $\gamma/\nu = 7/4 = 1.75$ . Στη διαδικασία των προσαρμογών πρέπει να αποφασίσουμε ποιες τιμές του  $L$  θα χρησιμοποιήσουμε.





Σχήμα 6.8: Οι τιμές  $\langle m \rangle(\beta_c, L)$  στην κρίσιμη θερμοκρασία για διαφορετικές τιμές του  $L$ . Οι κλίμακες στους άξονες είναι λογαριθμικές και η γραμμική σχέση των δεδομένων που προκύπτει, αποδεικνύει τη σχέση δύναμης  $\langle m \rangle(\beta_c, L) = cL^g$ . Με προσαρμογή των δεδομένων στην συνάρτηση αυτή, ο υπολογισμός του εκθέτη  $g$  δίνει τον κρίσιμο εκθέτη  $\beta/\nu$ , όπως προκύπτει από τη σχέση (6.9).

Φυσικά το πρώτο κριτήριο είναι οι προσαρμογές να δίνουν αποδεκτό  $\chi^2/\text{dof} \lesssim 1$  και το σφάλμα στις σταθερές  $g$  και  $a$  να είναι μικρό, αλλά παρατηρούμε ότι και ο περιορισμός αυτός δεν είναι αρκετός: Στον πίνακα 6.3 βλέπουμε μικρές μεταβολές στις τιμές του εκθέτη  $\gamma/\nu$  που παίρνουμε για διαφορετικά διαστήματα προσαρμογής. Οι μεταβολές αυτές μας δίνουν ένα μέτρο του συστηματικού σφάλματος στον προσδιορισμό τους. Στο πρόβλημα 5 σας ζητείται να δοκιμάσετε τον υπολογισμό μόνοι σας. Στον πίνακα 6.3 δίνονται τα αποτελέσματα και στο σχήμα 6.7 την αντίστοιχη γραφική παράσταση. Το τελικό αποτέλεσμα, λαμβάνοντας υπόψη το συστηματικό σφάλμα είναι:

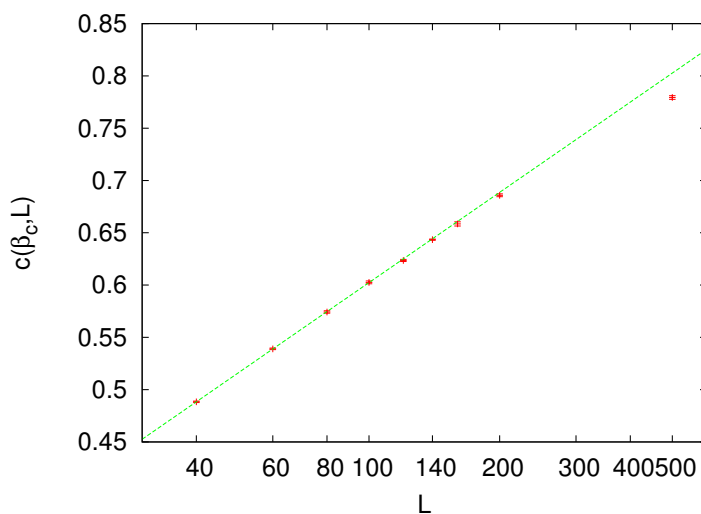
$$\frac{\gamma}{\nu} = 1.748 \pm 0.005. \quad (6.17)$$

Η σχέση (6.9) δίνει τον εκθέτη  $\beta/\nu$ . Για τον υπολογισμό του, ακολουθούμε ανάλογη διαδικασία για τις τιμές της μαγνήτισης  $\langle m \rangle(\beta_c, L)$  στην κρίσιμη θερμοκρασία. Το αποτέλεσμα είναι

$$\frac{\beta}{\nu} = 0.1245 \pm 0.0006. \quad (6.18)$$

Η σχέση (6.9) δίνει τον εκθέτη  $\alpha/\nu$ . Αλλά η αναμενόμενη τιμή  $\alpha = 0$  οδηγεί, σε αναλογία με τη σχέση (6.16) στη σχέση

$$c(\beta_c, L) \sim \log L. \quad (6.19)$$

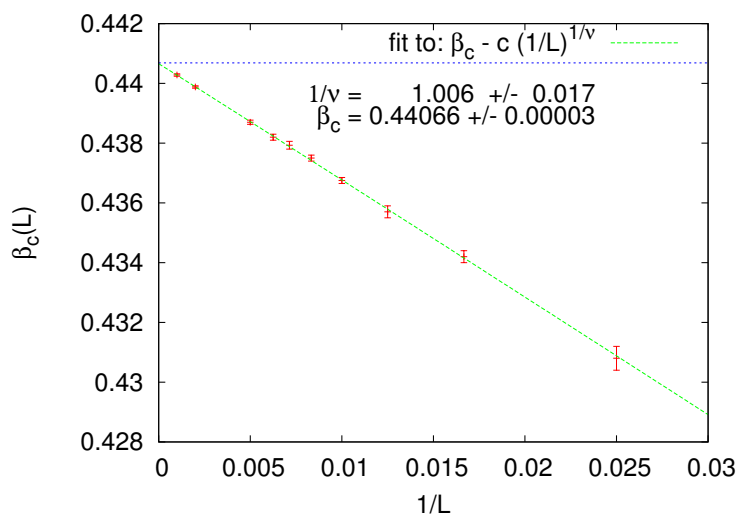


Σχήμα 6.9: Οι τιμές  $c(\beta_c, L)$  στην κρίσιμη θερμοκρασία για διαφορετικές τιμές του  $L$ . Η κλίμακα στον οριζόντιο άξονα είναι λογαριθμική, ενώ στον κάθετο δεν είναι. Αυτό αποδεικνύει τη σχέση  $c(\beta_c, L) = c \log L$  το οποίο αποδεικνύεται και με την προσαρμογή των δεδομένων στην αντίστοιχη συνάρτηση. Αυτό είναι συμβατό με το γνωστό αποτέλεσμα  $\alpha = 0$  στη σχέση (6.8).

Η σχέση αυτή δείχνεται στο σχήμα 6.9. Ο κάθετος άξονας δεν είναι σε λογαριθμική κλίμακα ενώ ο οριζόντιος είναι. Η γραμμική σχέση των δεδομένων στο σχήμα δείχνει ότι είναι συμβατά με την εξίσωση (6.19). Στο πρόβλημα 5 θα πρέπει να δείξετε αν η προσαρμογή των δεδομένων στο λογάριθμο είναι καλύτερη από την προσαρμογή σε μια συνάρτηση δύναμης της μορφής  $cL^a + b$  και να μελετήσετε τις δυσκολίες που προκύπτουν στη μελέτη αυτή. Βελτιώνοντας τα δεδομένα του πίνακα 6.6 με μεγαλύτερη στατιστική (ελάττωση στατιστικού σφάλματος) και μετρήσεις σε μεγαλύτερα  $L$  τα αποτελέσματα μπορούν να γίνουν ακόμα καθαρότερα.

Παρατηρούμε ότι με τη μέθοδο βάθμισης πεπερασμένου μεγέθους (finite size scaling) οι τιμές που προκύπτουν για τους κρίσιμους εκθέτες προκύπτουν ευκολότερα από τη μελέτη της βάθμισης ως προς τη θερμοκρασία που μελετήσαμε στην παράγραφο 6.2. Οι τιμές που μετράμε ακολουθούν την ασυμπτωτική σχέση βάθμισης που δίνουν οι σχέσεις (6.7)–(6.9) με πολύ μικρότερη επίδραση των φαινομένων επίδρασης πεπερασμένου μεγέθους<sup>9</sup>. Μπο-

<sup>9</sup>Θυμίζουμε στον αναγνώστη πως στους υπολογισμούς που δώσανε τις παραπάνω τιμές είχαμε μεγαλύτερη αστάθεια στις προσαρμογές ως προς την επιλογή διαστήματος των τιμών των δεδομένων. Στην επόμενη παράγραφο θα δούμε ότι όταν η ακριβής τιμή της κρίσιμης θερμοκρασίας μας είναι άγνωστη, η υπεροχή

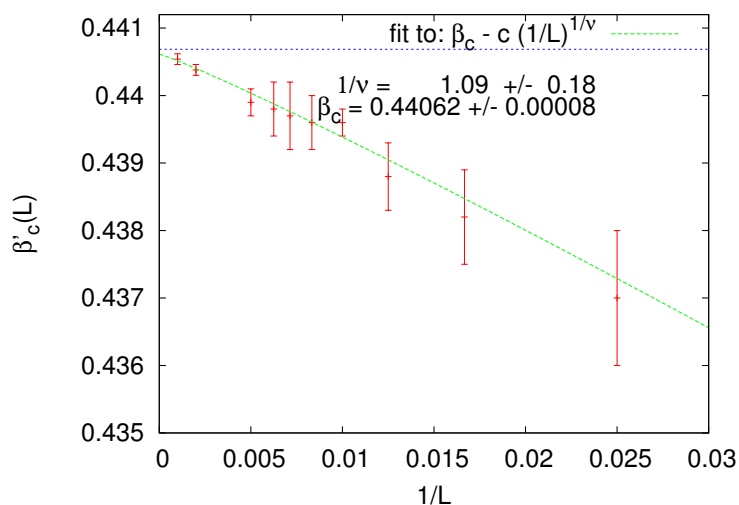


Σχήμα 6.10: Προσδιορισμός της κρίσιμης θερμοκρασίας  $\beta_c$  και του κρίσιμου εκθέτη  $\nu$  από τη σχέση (6.22). Χρησιμοποιώντας τις τιμές των ψευδοκρίσιμων θερμοκρασιών  $\beta_c(L)$  από τον πίνακα 6.4 προσαρμόζουμε τα δεδομένα σε μια συνάρτηση  $a - c(1/L)^b$ . Από τις τιμές που προκύπτουν για τις παραμέτρους  $a$ ,  $b$  και  $c$  υπολογίζουμε τα  $\beta_c = a$ ,  $1/\nu = b$ . Η οριζόντια γραμμή είναι η ακριβής τιμή  $\beta_c = \log(1 + \sqrt{2})/2 = 0.44069\dots$

ρούμε με τη μελέτη πολύ μικρότερου συστήματος να πάρουμε τιμές που έχουν μεγαλύτερη ακρίβεια και προκύπτουν καθαρότερα από αυτές που δώσαμε στις εξισώσεις (6.10), (6.11), (6.14), (6.15).

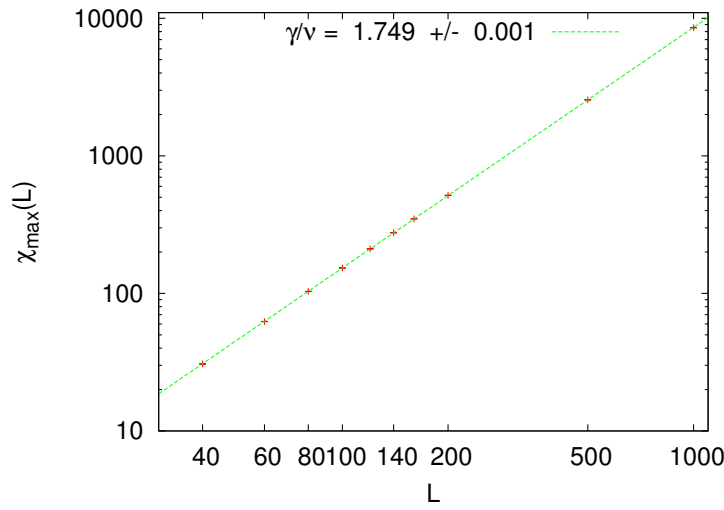
## 6.4 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΗΣ $\beta_c$

Στην παράγραφο 6.3 συζητήσαμε τη βάρθρωση των φυσικών ποσοτήτων σαν συνάρτηση της ανηγμένης θερμοκρασίας  $t$  και του μεγέθους του συστήματος  $L$ . Τη συζήτηση βοήθησε σημαντικά η γνώση της ακριβούς κρίσιμης θερμοκρασίας  $\beta_c = \log(1 + \sqrt{2})/2$  η οποία είναι γνωστή από τους αναλυτικούς υπολογισμούς και ειδικότερα από τη λύση του Onsager [12]. Δυστυχώς στις περισσότερες ενδιαφέρουσες περιπτώσεις η κρίσιμη παράμετρος δεν είναι γνωστή. Στην περίπτωση αυτή η ανάλυση γίνεται δυσκολότερη, αφενός μεν γιατί η κρίσιμη θερμοκρασία πρέπει να υπολογιστεί και αφετέρου διότι η πεπερασμένη ακρίβεια υπολογισμού της (άρα και στον υπολογισμό της  $t$ ) εισάγει σημαντικά συστηματικά σφάλματα στον υπολογισμό των κρίσιμων εκθετών. Επίσης η μη εκ των του finite size scaling είναι ακόμα μεγαλύτερη.



Σχήμα 6.11: Προσδιορισμός της κρίσιμης θερμοκρασίας  $\beta_c$  και του κρίσιμου εκθέτη  $\nu$  από τη σχέση (6.22). Χρησιμοποιώντας τις τιμές των ψευδοκρίσιμων θερμοκρασιών  $\beta'_c(L)$  από τον πίνακα 6.4 προσαρμόζουμε τα δεδομένα σε μια συνάρτηση  $a - c(1/L)^b$ . Από τις τιμές που προκύπτουν για τις παραμέτρους  $a$ ,  $b$  και  $c$  υπολογίζουμε τα  $\beta_c = a$ ,  $1/\nu = b$ . Η οριζόντια γραμμή είναι η ακριβής τιμή  $\beta_c = \log(1 + \sqrt{2})/2 = 0.44069\dots$

προτέρων γνώση της  $\beta_c$  κάνει την επιλογή των τιμών που θα επιλέξουμε στις σχέσεις βάθμισης (6.7)–(6.9) να πρέπει να προσδιοριστεί ακριβώς ώστε να έχουμε γρήγορη σύγκλιση στην ασυμπτωτική συμπεριφορά και ελαχιστοποίηση των συστηματικών σφαλμάτων. Εδώ η κατανόηση της αιτίας της βάθμισης των φυσικών ποσοτήτων μπορεί να βοηθήσει. Ο λόγος που οδηγεί την ψευδοκρίσιμη συμπεριφορά ενός πεπερασμένου συστήματος που παρουσιάζει συνεχή μετάβαση φάσης στο θερμοδυναμικό όριο είναι η αύξηση του μήκους συσχετισμού  $\xi$  σε μέγεθος που συγκρίνεται με το μέγεθος του συστήματος  $L$ . Περαιτέρω αύξηση “πνίγεται” από το πεπερασμένο μέγεθος του πλέγματος και παρουσιάζονται τα φαινόμενα επίδρασης πεπερασμένου μεγέθους (finite size effects) που ορίζουν την ψευδοκρίσιμη περιοχή. Στην περιοχή αυτή ποσότητες που απειρίζονται στο θερμοδυναμικό όριο (ή παρουσιάζουν μη αναλυτική συμπεριφορά, λ.χ. απότομη αλλαγή της τιμής της παραγωγού) έχουν πεπερασμένη, μέγιστη τιμή (ή πλησιάζουν προς τη μη αναλυτική συμπεριφορά). Αυτή, καθώς το μέγεθος του συστήματος μεγαλώνει, εκτείνεται σε μια όλο και πιο στενή περιοχή γύρω από την κρίσιμη θερμοκρασία. Οποιαδήποτε τιμή και αν πά-



Σχήμα 6.12: Υπολογισμός του κρίσιμου εκθέτη  $\gamma/\nu$  από τα μέγιστα της μαγνητικής επιδεκτικότητας, εφαρμόζοντας την ασυμπτωτική σχέση (6.9). Οι τιμές  $\chi_{\max}(L)$  λαμβάνονται από τον πίνακα (6.4) και προσαρμόζονται σε μια συνάρτηση της μορφής  $aL^b$ . Από το αποτέλεσμα της προσαρμογής παίρνουμε  $\gamma/\nu = 1.749(1)$ .

ρούμε στην περιοχή αυτή με συστηματικό τρόπο<sup>10</sup> για αρκετά μεγάλο  $L$  θα πάρουμε τις ασυμπτωτικές συμπεριφορές που δίνονται από τις σχέσεις (6.7)-(6.9). Μία επιλογή που δίνει καλά αποτελέσματα είναι να υπολογίσουμε τις ποσότητες στην ψευδοκρίσιμη θερμοκρασία  $\beta_c(L)$ . Αυτή ορίζεται από την τιμή που παρουσιάζεται το μέγιστο των διακυμάνσεων της παραμέτρου τάξης που εδώ είναι η μαγνήτιση  $\langle m \rangle$

$$\chi(\beta_c(L), L) \equiv \chi_{\max}(L). \quad (6.20)$$

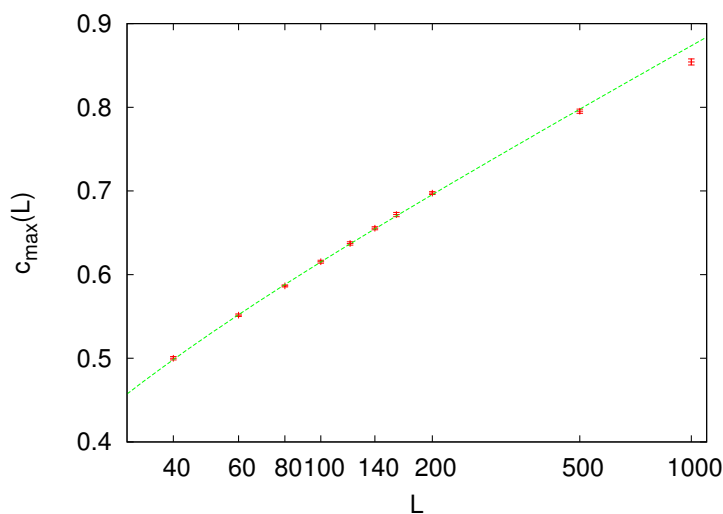
Στην περίπτωση αυτή στο αριστερό μέλος των εξισώσεων (6.7)-(6.9) είναι οι τιμές για  $\beta = \beta_c(L)$ .

Ο ορισμός της  $\beta_c(L)$  μπορεί να μην είναι μοναδικός. Θα μπορούσε λ.χ. κανείς να χρησιμοποιήσει το μέγιστο της ειδικής θερμότητας

$$c(\beta'_c(L), L) \equiv c_{\max}(L). \quad (6.21)$$

που ορίζει μια διαφορετική  $\beta'_c(L)$ . Προφανώς  $\lim_{L \rightarrow \infty} \beta'_c(L) = \lim_{L \rightarrow \infty} \beta_c(L) = \beta_c$  οπότε και οι δύο επιλογές θα δώσουν για αρκετά μεγάλα  $L$  τα ίδια αποτελέσματα. Δεν συμβαίνει όμως το ίδιο γρήγορα και

<sup>10</sup>Όπως λ.χ. κάναμε στην παράγραφο 6.3 όπου υπολογίσαμε κάθε ποσότητα ακριβώς στην  $\beta_c$



Σχήμα 6.13: Υπολογισμός του κρίσιμου εκθέτη  $\alpha/\nu$  από τα μέγιστα της μαγνητικής επιδεκτικότητας, εφαρμόζοντας την ασυμπτωτική σχέση (6.8). Οι τιμές  $c_{\max}(L)$  λαμβάνονται από τον πίνακα (6.4) και προσαρμόζονται σε μια συνάρτηση της μορφής  $a \log L + b - c/L$ . Παίρνουμε  $a = 0.107(3)$ ,  $b = 0.13(1)$  και  $c = 1.2(3)$  με  $\chi^2/\text{dof} = 0.9$  για προσαρμογή των σημείων  $L = 40, \dots, 500$ . Αν αποπειραθούμε προσαρμογή κάτω από τις ίδιες συνθήκες σε συνάρτηση της μορφής  $aL^d + b - c/L$  παίρνουμε  $d = 0.004(97)$ , δηλ. εκθέτη συμβατό με την τιμή μηδέν και “περίεργες” τιμές για τις σταθερές  $a, b$ . Συμπεραίνουμε ότι τα δεδομένα μας είναι συμβατά με τον εκθέτη  $\alpha/\nu$  να είναι μηδέν.

για τις δύο επιλογές, οπότε είναι συνήθως ευνοϊκή μία από αυτές (εδώ η  $\beta_c(L)$ ).

Το πρώτο βήμα είναι ο υπολογισμός του  $\beta_c$ . Όταν είμαστε στην ψευδοκρίσιμη περιοχή έχουμε  $\xi \approx L$ , οπότε η βασική σχέση (6.2) δίνει

$$|t| = \left| \frac{\beta_c - \beta_c(L)}{\beta_c} \right| \sim \xi^{-\frac{1}{\nu}} \sim L^{-\frac{1}{\nu}} \Rightarrow \beta_c(L) = \beta_c - \frac{c}{L^{\frac{1}{\nu}}}. \quad (6.22)$$

Η διαδικασία είναι απλή, ως προς την αρχή της: Μετράμε αρχικά τη μαγνητική επιδεκτικότητα. Για κάθε  $L$  προσδιορίζουμε την ψευδοκρίσιμη περιοχή και υπολογίζουμε τη θέση  $\beta_c(L)$  και την τιμή του μέγιστου  $\chi_{\max}$ . Για να γίνει αυτό μπορεί να χρειαστεί να πάρουμε αρκετές μετρήσεις γύρω από τη  $\beta_c(L)$ . Ιδιαίτερη προσοχή δίνουμε στο να έχουμε αρκετές ανεξάρτητες μετρήσεις μετρώοντας το χρόνο αυτοσυσχετισμού (αυτός αυξάνει με το μέγεθος  $L$  σαν  $\tau \sim L^z$ ) και φυσικά να έχουμε φτάσει στην κατάσταση θερμικής ισορροπίας. Χρησιμοποιούμε τη σχέση (6.22) και προσαρμόζουμε τα δεδομένα μας σε μια συνάρτηση της μορφής  $a - c/L^b$

και από τις τιμές που προκύπτουν για τις σταθερές  $a, b$  και  $c$  υπολογίζουμε τις  $\beta_c = a, \nu = 1/b$ . Σε περίπτωση που γνωρίζουμε μια από τις σταθερές  $\beta_c, \nu$  κρατάμε στην προσαρμογή τις αντίστοιχες παραμέτρους  $a, b$  σταθερές και υπολογίζουμε την άλλη.

Τα αποτελέσματα δείχνονται γραφικά στο σχήμα 6.10 όπου γίνεται η γραφική παράσταση των δεδομένων του πίνακα 6.4. Το αποτέλεσμα είναι:

$$\begin{aligned}\beta_c &= 0.44066 \pm 0.00003 \\ \frac{1}{\nu} &= 1.006 \pm 0.017.\end{aligned}$$

Αυτό μπορεί να συγκριθεί με τις γνωστές τιμές  $\beta_c = \log(1+\sqrt{2})/2 \approx 0.44069$  και  $1/\nu = 1$ .

Η διαδικασία επαναλαμβάνεται για τις ψευδοκρίσιμες θερμοκρασίες  $\beta'_c(L)$  όπου προκύπτουν από τα μέγιστα της ειδικής θερμότητας  $c_{\max}$ . Τα αποτελέσματα δείχνονται γραφικά στο σχήμα 6.11. Το αποτέλεσμα είναι:

$$\begin{aligned}\beta_c &= 0.44062 \pm 0.00008 \\ \frac{1}{\nu} &= 1.09 \pm 0.18.\end{aligned}$$

Παρατηρούμε από το σχήμα 6.11 και από τα παραπάνω αποτελέσματα ότι η μέτρηση από τη μελέτη της ειδικής θερμότητας δίνει αποτελέσματα συμβατά με τις εξισώσεις (6.23) αλλά με πολύ μικρότερη ακρίβεια και καθαρότητα. Τα μέγιστα της ειδικής θερμότητας είναι πιο ασαφή από αυτά της μαγνητικής επιδεκτικότητας.

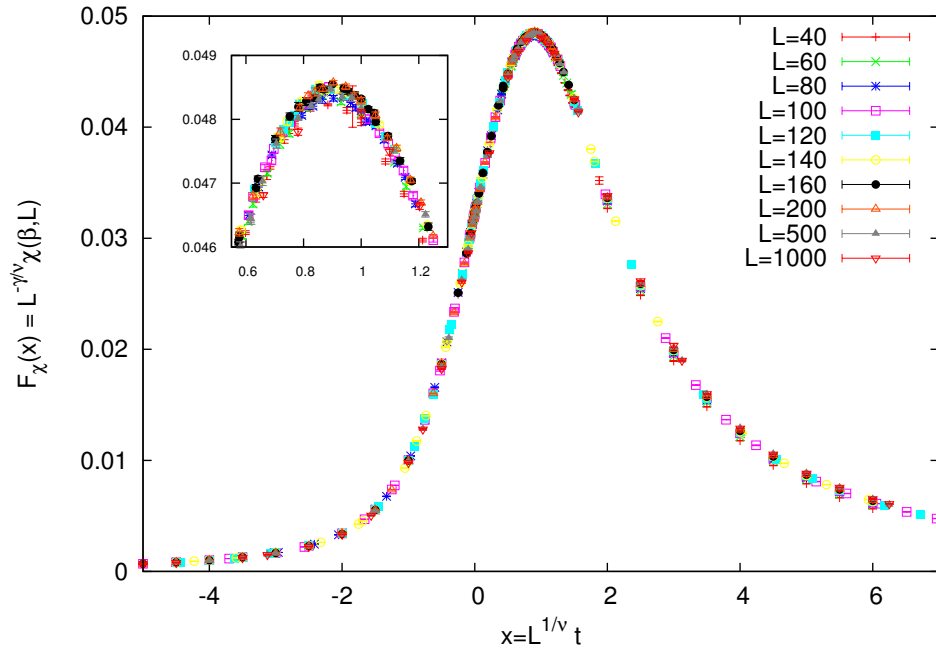
Από τα μέγιστα της μαγνητικής επιδεκτικότητας  $\chi_{\max}(L)$  υπολογίζουμε τον εκθέτη  $\gamma/\nu$ . Οι τιμές τους δίνονται στον πίνακα 6.4. Σύμφωνα με την ασυμπτωτική σχέση (6.9) προσαρμόζουμε τα δεδομένα σε μια συνάρτηση της μορφής  $aL^b$  με  $a$  και  $b$  προσδιοριστές παραμέτρους. Βρίσκουμε πολύ καλή βάρθμιση, άρα τα δεδομένα μας είναι μέσα στην ασυμπτωτική περιοχή. Το αποτέλεσμα είναι

$$\frac{\gamma}{\nu} = 1.749 \pm 0.001, \quad (6.23)$$

που είναι σε άριστη συμφωνία με την αναμενόμενη τιμή  $7/4$ .

Από τα μέγιστα της ειδικής θερμότητας υπολογίζουμε τον εκθέτη  $\alpha/\nu$ . Λόγω του ότι  $\alpha = 0$  η μορφή της ασυμπτωτικής σχέσης αναμένεται να δίνεται από την (6.19). Βρίσκουμε ότι τα δεδομένα μας δεν προσαρμόζονται καλά σε μια απλή συνάρτηση της μορφής  $a \log L$  και είναι πιθανό να παίρνουμε συνεισφορές από όρους που μηδενίζονται στο θερμοδυναμικό όριο, αλλά που δίνουν πεπερασμένη και μη αμελητέα συνεισφορά για πεπερασμένο  $L$ . Πράγματι προσαρμόζοντας τα δεδομένα σε μια συνάρτηση της μορφής

$a \log L + b - c/L$  παίρνουμε πολύ καλή προσαρμογή των δεδομένων<sup>11</sup>. Η απόπειρα προσαρμογής σε συνάρτηση δύναμης της μορφής  $aL^d + b - c/L$  δεν είναι επιτυχής<sup>12</sup> και η οποία δίνει έτσι και αλλιώς εκθέτη συμβατό με τιμή μηδέν. Τα αποτελέσματα δείχνονται γραφικά στο σχήμα 6.13



Σχήμα 6.14: “Κατάρρευση” (collapse) των καμπύλων  $\chi(\beta, L)$  για διαφορετικές τιμές του  $L$  σύμφωνα με τη σχέση (6.31). Για το σχήμα έχουν χρησιμοποιηθεί οι γνωστές τιμές  $\beta_c = \ln(1 + \sqrt{2})/2$ ,  $\nu = 1$  και  $\gamma/\nu = 7/4$ .

## 6.5 ΜΕΛΕΤΗ ΒΑΘΜΙΣΗΣ ΜΕ ... ΚΑΤΑΡΡΕΥΣΗ

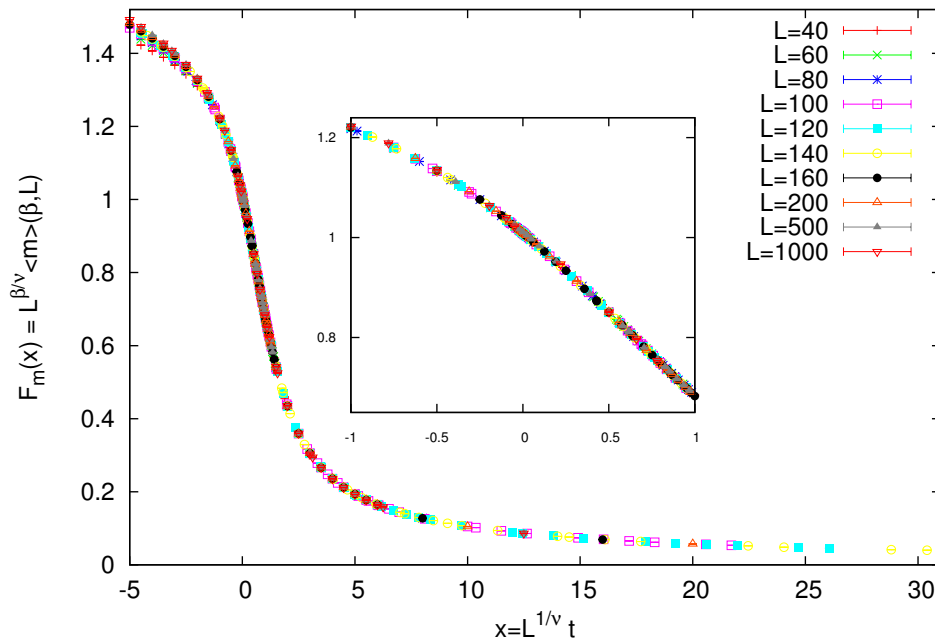
Οι σχέσεις βάθμισης (6.3)–(6.9) προέρχονται από τη δυναμική εμφάνιση μιας μοναδικής κλίμακας μήκους που απειρίζεται στην κρίσιμη περιοχή, του μήκους συσχετισμού  $\xi$ <sup>13</sup>. Η εμφάνιση αυτής

<sup>11</sup>Τα αποτελέσματά μας δικαιώνονται και από τους αναλυτικούς υπολογισμούς στο [30] όπου αποδεικνύεται ότι οι διορθωτικοί όροι είναι ακέραιες δυνάμεις του  $1/L$ :  $c = a \log L + \sum_{k=0}^{\infty} c_k/L^k$ .

<sup>12</sup>Τα  $a$  και  $b$  δεν προσδιορίζονται μέσα στο σφάλμα, ουσιαστικά ο ένας όρος αναιρεί τον άλλο και μένει μόνο ο τρίτος, το  $c$ .

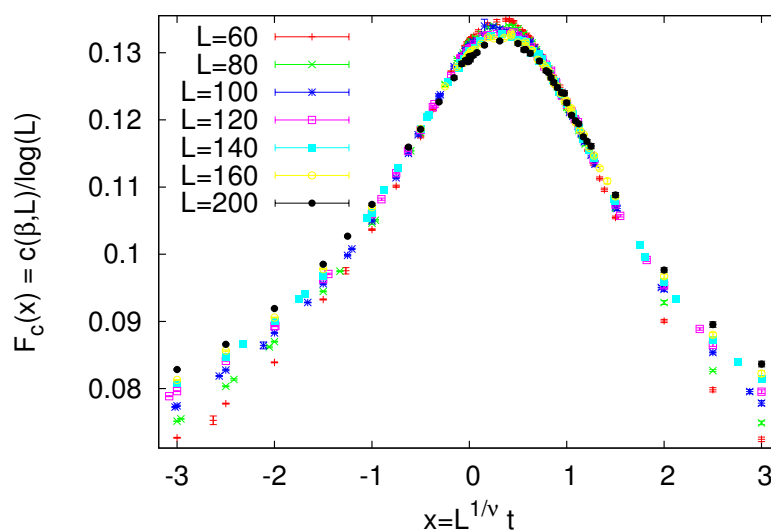
<sup>13</sup>Προσοχή: εδώ το  $\xi = \xi(t)$  είναι το μήκος συσχετισμού που έχει το άπειρο σύστημα σε θερμοκρασία  $t$  και όχι το μήκος συσχετισμού που μετράμε στο πεπερασμένο σύστημα.





Σχήμα 6.15: “Κατάρρευση” (collapse) των καμπύλων  $\langle m \rangle(\beta, L)$  για διαφορετικές τιμές του  $L$  σύμφωνα με τη σχέση (6.32). Για το σχήμα έχουν χρησιμοποιηθεί οι γνωστές τιμές  $\beta_c = \ln(1 + \sqrt{2})/2$ ,  $\nu = 1$  και  $\beta/\nu = 1/8$ .

της κλίμακας που απειρίζεται καθώς πλησιάζουμε το κρίσιμο σημείο σαν  $\xi \sim |t|^{-\nu}$  οδηγεί στην παγκοσμιότητα της κρίσιμης συμπεριφοράς όλων των συστημάτων που ανήκουν στην ίδια κλάση παγκοσμιότητας. Αν πάρουμε για παράδειγμα τη μαγνητική επιδεκτικότητα  $\chi(\beta, L)$ , οι τιμές τις εξαρτώνται ξεχωριστά από τη θερμοκρασία  $\beta$ , το μέγεθος του συστήματος  $L$  και φυσικά από τις λεπτομέρειες του συστήματος. Η εμφάνιση παγκόσμιας συμπεριφοράς για ολόκληρη την κλάση ισοδυναμίας που ανήκει το μοντέλο που μελετάμε μας οδηγεί στην *υπόθεση* ότι η μαγνητική επιδεκτικότητα στην περιοχή αυτή εξαρτάται *μόνο* από το μήκος συσχετισμού  $\xi$ . Καθώς στο πεπερασμένο σύστημα πλησιάζουμε την κρίσιμη θερμοκρασία τα φαινόμενα κυριαρχούνται από την “κατάπνιξη” των διακυμάνσεων που προκύπτουν όταν  $\xi \sim L$ . Άρα οι κλίμακες μήκους που καθορίζουν την κύρια συμπεριφορά των συναρτήσεων που περιγράφουν τη βάρθρωση  $\chi \sim \xi^{\gamma/\nu}$  είναι οι  $\xi$  και  $L$ , άρα η *αδιάστατη* μεταβλητή  $L/\xi$  είναι η μοναδική μεταβλητή που υπεισέρχεται στις συναρτήσεις βάρθρωσης. Για να πάρουμε την σχέση βάρθρωσης  $\chi \sim \xi^{\gamma/\nu}$  αρκεί να υποθέσουμε ότι κοντά στην κρί-



Σχήμα 6.16: “Κατάρρευση” (collapse) των καμπύλων  $c(\beta, L)$  για διαφορετικές τιμές του  $L$  σύμφωνα με τη σχέση (6.33). Για το σχήμα έχουν χρησιμοποιηθεί οι γνωστές τιμές  $\beta_c = \ln(1 + \sqrt{2})/2$  και  $\nu = 1$ .

σιμη περιοχή<sup>14</sup>

$$\chi = \xi^{\gamma/\nu} F_{\chi}^{(0)}(L/\xi), \quad (6.24)$$

όπου  $F_{\chi}^{(0)}(z)$  είναι μια συνάρτηση μιας μεταβλητής με τις ιδιότητες

$$F_{\chi}^{(0)}(z) = \text{σταθ.} \quad z \gg 1, \quad (6.25)$$

και

$$F_{\chi}^{(0)}(z) \sim z^{\gamma/\nu} \quad z \rightarrow 0. \quad (6.26)$$

Πράγματι όταν  $1 \ll \xi \ll L$  ( $z \gg 1$ ) η μαγνητική επιδεκτικότητα παίρνει τιμές πολύ κοντά στις τιμές του άπειρου συστήματος (θερμοδυναμικό όριο) και από την (6.25) παίρνουμε  $\chi \sim \xi^{\gamma/\nu}$ . Καθώς  $\xi \sim L$  εμφανίζονται τα φαινόμενα επίδρασης πεπερασμένου μεγέθους και από την (6.26) παίρνουμε  $\chi \sim \xi^{\gamma/\nu} (L/\xi)^{\gamma/\nu} = L^{\gamma/\nu}$ . Η τελευταία δεν είναι άλλη από τη σχέση (6.7) για τα μέγιστα της μαγνητικής επιδεκτικότητας στο πεπερασμένο σύστημα που μελετήσαμε στο σχήμα 6.12. Άρα η συνάρτηση  $F_{\chi}^{(0)}(z)$  περιγράφει τον τρόπο που “αποκόπτεται” η μαγνητική επιδεκτικότητα από τα φαινόμενα επίδρασης πεπερασμένου μεγέθους.

Η συνάρτηση  $F_{\chi}^{(0)}(z)$  μπορεί να υπολογιστεί από τις μετρήσεις που κάνουμε στην προσομοίωση Μόντε Κάρλο. Επειδή το μήκος

<sup>14</sup>Για περισσότερες λεπτομέρειες παραπέμπουμε στο Παράρτημα Γ'.

συσχετισμού δεν υπολογίζεται άμεσα, αλλά υπεισέρχεται έμμεσα στις μετρήσεις μας, είναι πιο βολικό να ορίσουμε την αδιάστατη μεταβλητή βάθμισης να είναι η

$$x = L^{1/\nu}t, \quad (6.27)$$

όπου  $|x| \sim (L/\xi)^{1/\nu}$  αφού<sup>15</sup>  $\xi \sim |t|^{-\nu}$ . Ορίζοντας τότε  $F_\chi(x) \propto x^{-\gamma} F_\chi^{(0)}(x^\nu)$  έτσι ώστε η σχέση (6.24) να γίνει

$$\chi = L^{\gamma/\nu} F_\chi(x) = L^{\gamma/\nu} F_\chi(L^{1/\nu}t). \quad (6.28)$$

Οι ασυμπτωτικές ιδιότητες της συνάρτησης βάθμισης  $F_\chi(x)$  προκύπτουν από τις σχέσεις (6.25) και (6.26). Για  $x = L^{1/\nu}t \gg 1$  για την  $F_\chi^{(0)}(x^\nu)$  ισχύει η (6.25) και παίρνουμε  $F_\chi^{(0)}(x^\nu) = \text{σταθ.}$  Από τον ορισμό  $F_\chi(x) = x^{-\gamma} F_\chi^{(0)}(x^\nu)$  παίρνουμε  $F_\chi(x) \sim x^{-\gamma} = (L/\xi)^{-\gamma/\nu}$  και επιβεβαιώνουμε την ιδιότητα βάθμισης της μαγνητικής επιδεκτικότητας στο θερμοδυναμικό όριο  $\chi \sim L^{\gamma/\nu} F_\chi(x) \sim L^{\gamma/\nu} (L/\xi)^{-\gamma/\nu} = \xi^{\gamma/\nu}$ . Άρα

$$F_\chi(x) \sim x^{-\gamma} \quad x \gg 1. \quad (6.29)$$

Όταν  $x \rightarrow 0$  ισχύει η (6.26) και  $F_\chi^{(0)}(x^\nu) \sim (x^\nu)^{\gamma/\nu} = x^\gamma$ . Τότε έχουμε  $F_\chi(x) \propto x^{-\gamma} F_\chi^{(0)}(x^\nu) \sim x^{-\gamma} x^\gamma = \text{σταθ.}$  Επιβεβαιώνουμε ότι όταν επικρατούν τα φαινόμενα επίδρασης πεπερασμένου μεγέθους ( $x \rightarrow 0$ ) παίρνουμε  $\chi = L^{\gamma/\nu} F_\chi(x) \sim L^{\gamma/\nu}$  όπως και παραπάνω. Άρα

$$F_\chi(x) \sim \text{σταθ.} \quad |x| \ll 1. \quad (6.30)$$

Αντιστρέφοντας τη σχέση (6.28) μπορούμε να δούμε πώς είναι δυνατόν να υπολογιστεί η συνάρτηση βάθμισης από τις μετρήσεις της μαγνητικής επιδεκτικότητας

$$F_\chi(L^{1/\nu}t) = L^{-\gamma/\nu} \chi(\beta, L), \quad (6.31)$$

όπου η  $\chi(\beta, L)$  μετριέται σε θερμοκρασίες κοντά στην κρίσιμη περιοχή για διαφορετικές τιμές του  $L$ . Όταν η σχέση (6.31) ισχύει, τότε όλες οι μετρήσεις μας θα πέφτουν πάνω στην ίδια καμπύλη που δίνει τη συνάρτηση  $F_\chi(x)$  ανεξάρτητα από το μέγεθος του συστήματος  $L$ ! Φυσικά μικρές αποκλίσεις αναμένονται λόγω φαινομένων επίδρασης πεπερασμένου μεγέθους, ιδιαίτερα για μικρά  $L$ . Αλλά όπως θα δούμε, η σύγκλιση γίνεται στην πράξη αρκετά γρήγορα.

Το μεγάλο κέρδος από την προσέγγιση αυτή στις μετρήσεις μας είναι ότι μπορούμε να χρησιμοποιήσουμε την παραπάνω σχέση

<sup>15</sup> Αγνοούμε την απόλυτη τιμή στον ορισμό της  $x$  ώστε να έχουμε ενιαίο φορμαλισμό για τις περιοχές πάνω και κάτω από την κρίσιμη θερμοκρασία

για να προσδιορίσουμε την κρίσιμη θερμοκρασία  $\beta_c$ , τον εκθέτη  $\nu$  καθώς και το λόγο  $\gamma/\nu$  ταυτόχρονα! Παρατηρήστε ότι για να ισχύει η (6.31) πρέπει να προσδιορίσουμε την μεταβλητή  $x = L^{1/\nu}t$  για την οποία είναι αναγκαίο να γνωρίζουμε την  $\beta_c$  ( $t = (\beta_c - \beta)/\beta_c$ ) και τον εκθέτη  $\nu$ . Για τον υπολογισμό της  $F_\chi$  χρειάζεται να γνωρίζουμε τον εκθέτη  $\gamma/\nu$  που παρουσιάζεται στο δεξί μέλος της (6.31). Η σχέση (6.31) βρίσκεται να είναι πολύ ευαίσθητα εξαρτημένη από τις παραμέτρους  $\beta_c$ ,  $\nu$  και  $\gamma/\nu$  και έτσι έχουμε μια εξαιρετική και ακριβή μέθοδο για τον υπολογισμό τους.

Ο υπολογισμός γίνεται εύκολα. Θέτουμε δοκιμαστικές τιμές για τις παραμέτρους  $(\beta_c, \nu, \gamma/\nu)$ . Από τα δεδομένα μας χρησιμοποιούμε τις τιμές  $L$ ,  $\beta$ ,  $\beta_c$  και  $\nu$  για να υπολογίσουμε τη μεταβλητή βάθμισης  $x = L^{1/\nu}t = L^{1/\nu}(\beta_c - \beta)/\beta_c$ . Αυτή την αντιστοιχούμε στην τιμή της μαγνητικής επιδεκτικότητας  $\chi(\beta, L)$  από την οποία, μαζί με την τιμή του  $\gamma/\nu$ , υπολογίζουμε την  $F_\chi = \chi(\beta, L)/L^{\gamma/\nu}$ . Κάνουμε τη γραφική παράσταση των σημείων  $(x_i, F_\chi(x_i))$  και επικεντρωνόμαστε στο διάστημα κοντά στην κρίσιμη περιοχή ( $t \approx 0$ ). Αλλάζουμε τις τιμές των παραμέτρων  $(\beta_c, \nu, \gamma/\nu)$  μέχρι να πετύχουμε βέλτιστη σύμπτωση των καμπύλων που προκύπτουν για τα διαφορετικά  $L$  που μελετάμε. Μόλις το πετύχουμε, οι τιμές των  $(\beta_c, \nu, \gamma/\nu)$  που αντιστοιχούν στη βέλτιστη σύμπτωση είναι και οι ζητούμενες.

Η “κατάρρευση” (collapse) των καμπύλων που προκύπτουν από τα σημεία  $(L_i^{1/\nu}(\beta_c - \beta_i)/\beta_c, L_i^{-\gamma/\nu}\chi(\beta_i, L_i))$  για διαφορετικά  $L$  είναι η αποδοτικότερη μέθοδος εκμετάλλευσης των ιδιοτήτων βάθμισης στην κρίσιμη περιοχή. Στο σχήμα 6.14 δείχνουμε τη συνάρτηση  $F_\chi(x)$  για τις γνωστές τιμές των παραμέτρων  $(\beta_c, \nu, \gamma/\nu) = (\ln(1 + \sqrt{2})/2, 1, 7/4)$ . Μικρές μεταβολές από τις τιμές αυτές κάνουν εμφανή την διαφορά αυτή. Μεταβάλλοντας μια από τις παραμέτρους και παρατηρώντας την αποστασιοποίηση των καμπύλων μεταξύ διαφορετικών  $L$ , μπορούμε να πάρουμε μια πρόχειρη εκτίμηση της ελάχιστης ακρίβειας της μεθόδου

$$\begin{aligned}\beta_c &= 0.44069 \pm 0.00001 \\ \nu &= 1.00 \pm 0.01 \\ \frac{\gamma}{\nu} &= 1.750 \pm 0.002,\end{aligned}$$

Παρατηρούμε ότι με αυτή την πρόχειρη μέθοδο πετύχαμε ακρίβεια συγκρίσιμη με αυτή των προηγούμενων (συστηματικών) υπολογισμών μας!

Ανάλογη διαδικασία μπορεί να ακολουθηθεί και για άλλες υπολογιζόμενες ποσότητες που παρουσιάζουν βάθμιση, όπως είναι η ειδική θερμότητα και η μαγνήτιση. Οι σχέσεις (6.8) και (6.9) γε-

νικούνται σύμφωνα με τα παραπάνω για τη μαγνήτιση<sup>16</sup>

$$\langle m \rangle (\beta, L) = L^{-\beta/\nu} F_m(L^{1/\nu} t), \quad (6.32)$$

και για την ειδική θερμότητα σε

$$c(\beta, L) = L^{\alpha/\nu} F_c(L^{1/\nu} t) = \log(L) F_c(L^{1/\nu} t), \quad (6.33)$$

αφού  $\alpha = 0$ . Τα αποτελέσματα δείχνονται στα σχήματα 6.15 και 6.16 αντίστοιχα.

Για διευκόλυνση το αναγνώστη παραθέτουμε ένα πρόγραμμα gnuplot για την εύκολη κατασκευή σχημάτων όπως αυτά που δείχνονται στα σχήματα 6.14–6.16. Αν υποθέσουμε ότι τα δεδομένα μας είναι σε ένα αρχείο all τοποθετημένα ως εξής<sup>17</sup>:

```
# #####
# e L beta <e> +/- err c +/- err
# m L beta <m> +/- err chi +/- err
# n L beta <n>/N +/- err
# -----
....
e 1000 0.462721 -0.79839031 7.506e-07 0.290266 0.00027
m 1000 0.462721 0.82648701 1.384e-06 2.137 0.00179
....
```

όπου γραμμές που αρχίζουν με m έχουν τις μετρήσεις  $(L, \beta, \langle m \rangle, \delta \langle m \rangle, \chi, \delta \chi)$  ενώ αυτές που αρχίζουν από e τις μετρήσεις  $(L, \beta, \langle e \rangle, \delta \langle e \rangle, c, \delta c)$ . Οι εντολές που πρέπει να προγραμματίσουμε γράφονται σε ένα αρχείο scale\_gamma.gpl:

```
# Usage:
# Ls = "40 60 80 100 120 140 160 200 500 1000"
# bc = bcc; nu = 1 ; gnu = 1.75; load "scale_gamma.gpl";
# Ls: the values of L used in te collapse
# bc: the critical temperature used in the calculation of t=(beta_c-beta)/beta_c
# nu: the exponent used in the calculation of x=L^{1/nu} t
# gnu: the exponent used in the calculation of F_chi = L^{-gnu} chi(beta,L)

#the exact critical temperature (use bc=bcc is you wish):
bcc = 0.5*log(1.0+sqrt(2.0));
NLs = words(Ls); # The number of lattice sizes used in the calculation
```

<sup>16</sup>Προσοχή στο αριστερό μέλος  $\beta$  είναι η θερμοκρασία ενώ στο δεξί ο εκθέτης που ορίστηκε στη σχέση βάθμισης (6.5).

<sup>17</sup>Το αρχείο μπορεί να βρεθεί στο συνοδευτικό λογισμικό στον κατάλογο ch06/all.

```

LL(i) = word (Ls,i);# Returns the i_th lattice size
cplot(i) = sprintf("\
  <grep 'm %s ' all|\
  sort -k 3,3g|\
  awk -v L=%s -v bc=%f -v nu=%f -v gnu=%f \
  '{print L^(1.0/nu)*(bc-$3)/bc,L^(-gnu)*$6,L^(-gnu)*$7}'\
  ",LL(i),LL(i),bc,nu,gnu);

set macros
set term wxt enhanced

set title sprintf("b_c= %f nu= %f g/n= %f",bc,nu,gnu)
set xlabel "x=L^{1/nu} t"
set ylabel "F(x) = L^{-g/n} chi({/Symbol b},L)"

plot for[i=1:NLs] cplot(i) u 1:2:3 w e t sprintf("L=%s",LL(i))

```

Για τη χρήση του προγράμματος, ξεκινάμε το gnuplot και δίνουμε τις εντολές

```

gnuplot> Ls = "40 60 80 100 120 140 160 200 500 1000"
gnuplot> bc = 0.4406868; nu = 1 ; gnu = 1.75;
gnuplot> load "scale_gamma.gpl"

```

Οι δύο πρώτες εντολές καθορίζουν τις παραμέτρους του διαγράμματος. Η μεταβλητή  $L_s$  περιέχει τα μεγέθη του πλέγματος, την ... κατάρρευση των οποίων θέλουμε να μελετήσουμε, κάθε  $L$  χωρισμένο από το άλλο από ένα ή περισσότερα κενά. Η μεταβλητές  $bc$ ,  $nu$ ,  $gnu$  είναι οι παράμετροι  $\beta_c$ ,  $\nu$ ,  $\gamma/\nu$  που θα χρησιμοποιηθούν στη βάρθρωση (6.31). Η τρίτη εντολή καλεί το πρόγραμμά μας και φτιάχνει το διάγραμμα. Αν δεν έχουμε διαλέξει τις επιθυμητές παραμέτρους, τις επαναορίζουμε και... επαναλαμβάνουμε.

Για να καταλάβετε πώς λειτουργεί το πρόγραμμα δείτε την τεκμηρίωση του gnuplot<sup>18</sup>. Θα σταθούμε μόνο στην κατασκευή του φίλτρου που μας δίνει τα σημεία του γραφήματος σωστά κανονικοποιημένα. Το gnuplot το βλέπει σαν μια ακολουθία χαρακτήρων (string) `cplot(i)` που είναι διαφορετική για κάθε  $L$  (ο δείκτης  $i$  αντιστοιχεί στην  $i$ -λέξη στη μεταβλητή  $L_s$ ). Το string αυτό τοποθετείται στην εντολή `plot` για κάθε  $i$  και έχει τη μορφή "`< grep ... L^(-gnu)*$7`". Οι εκάστοτε τιμές περνούν στο string αυτό μέσω της συνάρτησης `sprintf()` η οποία κάθε φορά καλείται με διαφορετικό  $i$ . Την κύρια δουλειά την κάνει η `awk` που πολλαπλασιάζει

<sup>18</sup>Από την online τεκμηρίωση, μπορείτε να χρησιμοποιήσετε τις εντολές `help word`, `help words`, `help macros`, `help sprintf`, `help plot iteration`

την κάθε μέτρηση (στήλες 6 και 7: \$6, \$7 είναι η  $\chi$  και το σφάλμα της  $\delta\chi$ ) με τον σωστό παράγοντα. Η συνιστώσα του φίλτρου με το `grep` επιλέγει τις γραμμές από το αρχείο `a11` που έχουν τη μαγνητική επιδεκτικότητα για το δεδομένο μέγεθος πλέγματος  $L$ . Η συνιστώσα του φίλτρου με την εντολή `sort` ταξινομεί τα δεδομένα που παίρνει με σειρά αύξουσας θερμοκρασίας (στήλη 3: `-k3,3g`)

Το κύριο ερώτημα είναι πώς θα συστηματοποιηθεί η παραπάνω μέθοδος; Πώς κρίνουμε ότι μια κατάρρευση είναι καλή και πότε πάυει να είναι; Πώς θα εκτιμήσουμε τα σφάλματα στις παραμέτρους που προσπαθούμε να προσδιορίσουμε; Για το τελευταίο ερώτημα προσπαθήσαμε να δώσουμε παραπάνω μια πρόχειρη μέθοδο για την εκτίμηση των σφαλμάτων. Μια άλλη μέθοδος είναι να εφαρμόσουμε μια μέθοδο τύπου binning ή bootstrap. Στην πρώτη περίπτωση χωρίζουμε τα δεδομένα μας σε  $n_b$  bins και εργαζόμαστε ανεξάρτητα σε κάθε ένα από αυτά. Κάθε bin θα μας δώσει μια βέλτιστη ομάδα παραμέτρων  $(\beta_c, \nu, \gamma/\nu)$  την οποία θα χρησιμοποιήσουμε σαν μια ανεξάρτητη μέτρηση. Το σφάλμα τότε θα υπολογιστεί από το γνωστό τύπο (4.30) (για  $n = n_b$ ).

Για να μελετήσουμε ποσοτικά την ποιότητα της κατάρρευσης ορίζουμε μία ποσότητα που θυμίζει το  $\chi^2/\text{dof}$  που ορίζεται σε μια κοινή προσαρμογή δεδομένων όπως παρουσιάστηκε στο Παράρτημα Α'. Αυτή θα πρέπει να είναι μεγαλύτερη όταν αυξάνεται η απόσταση μεταξύ των καμπύλων που προκύπτουν από την κατάρρευση των δεδομένων για κάθε  $L$ . Έστω ότι οι μετρήσεις μας αποτελούνται από  $N_L$  σετ μετρήσεων για  $L = L_1, L_2, \dots, L_{N_L}$ . Αφού επιλέξουμε ένα σετ παραμέτρων  $\mathbf{p} \equiv (\beta_c, \nu, \gamma/\nu)$  και ένα διάστημα  $\Delta x \equiv [x_{\min}, x_{\max}]$  υπολογίζουμε τα σετ των δεδομένων  $\{(x_{i,k}, F_\chi(x_{i,k}; \mathbf{p}, L_i))\}_{k=1, \dots, n_i}$  που προκύπτουν από τις μετρήσεις μας. Αυτά αποτελούνται από τα  $n_i$  σημεία στο σετ δεδομένων με  $L = L_i$  των οποίων τα  $x_k$  βρίσκονται μέσα στο διάστημα  $\Delta x$ . Για κάθε σημείο  $x_k$  υπολογίζουμε την προκύπτουσα συνάρτηση βάρθμισης  $F_\chi(x_{i,k}; \mathbf{p}, L_i) = L_i^{-\gamma/\nu} \chi(\beta_{i,k}, L_i)$  η οποία εξαρτάται από τις επιλεγθείσες παραμέτρους  $\mathbf{b}$  και το επιλεγμένο μέγεθος  $L_i$ . Στη συνέχεια επιλέγουμε ένα τρόπο παρεμβολής (interpolation) μεταξύ των παραπάνω σημείων η οποία ορίζει συνεχείς συναρτήσεις παρεμβολής  $F_\chi(x; \mathbf{p}, L_i)$ <sup>19</sup> ώστε να έχουμε μια καλή εκτίμηση της συνάρτησης βάρθμισης μεταξύ των διακριτών σημείων που αποτελούν τα δεδομένα μας. Τότε κάθε σημείο στο σετ δεδομένων  $\{(x_{i,k}, F_\chi(x_{i,k}; \mathbf{p}, L_i))\}_i$  έχει δεδομένη απόσταση από κάθε άλλο σετ δεδομένων  $j$  ( $j = 1, \dots, N_L$  και  $j \neq i$ ) ίση με<sup>20</sup>

<sup>19</sup> Αυτή είναι ίσως και η πιο σημαντική επιλογή που θα κάνουμε. Πολυωνυμική προσαρμογή (συμπ. και γραμμικής), παρεμβολή τύπου `cspline` ή γενίκευσης της, `multihistogramming` είναι δυνατές επιλογές. Πιο απαλλαγμένη από συστηματικά σφάλματα είναι η τελευταία.

<sup>20</sup> Εδώ φαίνεται η ανάγκη της παρεμβολής αφού η τιμή  $x_{i,k}$  μπορεί να μην υπάρ-

$|F_\chi(x_{i,k}; \mathbf{p}, L_i) - F_\chi(x_{i,k}; \mathbf{p}, L_j)|$ . Ορίζεται τότε η ποσότητα

$$\chi^2(\mathbf{p}; \Delta x) = \frac{1}{N_L(N_L - 1)n_{\text{points}}} \times \sum_{i=1}^{N_L} \sum_{\{j=1, j \neq i\}}^{N_L} \sum_{k=1}^{n_i} \frac{(F_\chi(x_{i,k}; \mathbf{p}, L_i) - F_\chi(x_{i,k}; \mathbf{p}, L_j))^2}{(\delta F_\chi(x_{i,k}; \mathbf{p}, L_i))^2} \quad (6.34)$$

όπου  $n_{\text{points}} = \sum_{i=1}^{N_L} n_i$  είναι ο αριθμός των όρων στο άθροισμα. Η κανονικοποίηση με  $N_L(N_L - 1)$  γίνεται γιατί αυτός είναι ο αριθμός που οι καμπύλες συνδυάζονται ανά ζεύγη μεταξύ τους. Κάθε όρος ζυγίζεται με το σφάλμα του  $\delta F_\chi(x_{i,k}; \mathbf{p}, L_i)$  (που προέρχεται από το  $\delta\chi$ ) ώστε τα σημεία με μεγάλο σφάλμα να έχουν μικρότερη συνεισφορά από αυτά με μεγαλύτερο. Ο ορισμός αυτός χρησιμοποιήθηκε στο [31] αλλά ο αναγνώστης μπορεί να δει και εναλλακτικές προσεγγίσεις στα [1], [29].

Το  $\chi^2(\mathbf{p}; \Delta x)$  εξαρτάται από τις παραμέτρους  $\mathbf{p}$  και το διάστημα  $\Delta x$ . Γίνεται κατ' αρχήν μια ελαχιστοποίηση του ως προς τις παραμέτρους  $\mathbf{p}$  κρατώντας το  $\Delta x$  σταθερό. Το ελάχιστο δίνεται από τις τιμές των παραμέτρων  $\mathbf{p}_{\min}$  που είναι και η εκτίμηση των τιμών που θέλουμε να υπολογίσουμε. Για τον υπολογισμό των σφαλμάτων  $\delta\mathbf{p}$  η πιο συστηματική διαδικασία είναι η μέθοδος binning όπως αναφέραμε και παραπάνω στη σελίδα 197. Εναλλακτικά, αν υποθέσουμε μια κατανομή των μετρήσεων τύπου  $\chi^2$  και το αποτέλεσμα μας δίνει  $\chi^2 \lesssim 1$ , τότε τα διαστήματα των παραμέτρων μέσα στα όρια του σφάλματος είναι αυτά για τα οποία  $\chi^2 \lesssim 2$ .

Τα αποτελέσματα εξαρτώνται από το διάστημα  $\Delta x$ . Συνήθως [29] αυτό επιλέγεται να έχει το κέντρο του στο μέγιστο της  $F_\chi(x)$  έτσι ώστε  $\Delta x = [x_{\max} - \delta x, x_{\max} + \delta x]$ . Αν το  $\delta x$  είναι μεγαλύτερο από όσο πρέπει, τότε το  $\chi^2(\mathbf{p}_{\min}; \Delta x)$  είναι μεγάλο γιατί δεν έχουμε καλή βάρθρωση. Αν είναι πολύ μικρό, τότε τα σφάλματα  $\delta\mathbf{p}$  θα είναι μεγάλα. Παίρνοντας το όριο  $\delta x \rightarrow 0$  μελετάμε τη σύγκλιση των τιμών  $\mathbf{p}_{\min}$  (βλ. σχήμα 8.7, σελ 238 στο [1] καθώς και το [29]) από όπου καταλήγουμε στις τελικές τιμές του υπολογισμού μας.

## 6.6 BINDER CUMULANT ΚΑΙ ... ΟΧΙ ΜΟΝΟ

Μέχρι τώρα μελετήσαμε τις διακυμάνσεις των φυσικών ποσοτήτων από τον υπολογισμό cumulant<sup>21</sup> δεύτερης τάξης (μαγνη-

χει στο σετ δεδομένων  $j$ .

<sup>21</sup>

<http://en.wikipedia.org/wiki/Cumulant>,

<http://mathworld.wolfram.com/Cumulant-GeneratingFunction.html>



τική επιδεκτικότητα και ειδική θερμότητα). Σημαντική πληροφορία για τους κρίσιμους εκθέτες και την κρίσιμη θερμοκρασία μπορεί να ληφθεί από τη μελέτη των cumulant ανωτέρας τάξης. Ο πιο γνωστός, ο Binder cumulant, είναι τέταρτης τάξης<sup>22</sup> και παίρνει το όνομα του Kurt Binder που τον μελέτησε πρώτος [32, 33]

$$U = 1 - \frac{\langle m^4 \rangle}{3 \langle m^2 \rangle^2}. \quad (6.35)$$

Στο παράρτημα Γ' παρουσιάζονται με λεπτομέρεια οι ιδιότητές του. Αυτές που μας αφορούν άμεσα είναι πως σε μια συνεχή μετάβαση φάσης

$$U = \begin{cases} 0 & \beta \ll \beta_c \\ U^* & \beta = \beta_c \\ \frac{2}{3} & \beta \gg \beta_c \end{cases}, \quad (6.36)$$

όπου  $U^* = 0.610690(1)$  για το πρότυπο Ising στο τετραγωνικό πλέγμα [33]. Η τιμή  $U = 0$  ταυτίζεται με αυτή της γκαουσιανής κατανομής ενώ  $U = 2/3$  αντιστοιχεί περίπου με την γκαουσιανή κατανομή μικρής διασποράς γύρω από δύο συμμετρικές τιμές  $\pm \langle m \rangle$  (βλ. ασκήσεις 1 και 2 στο Παράρτημα Γ').

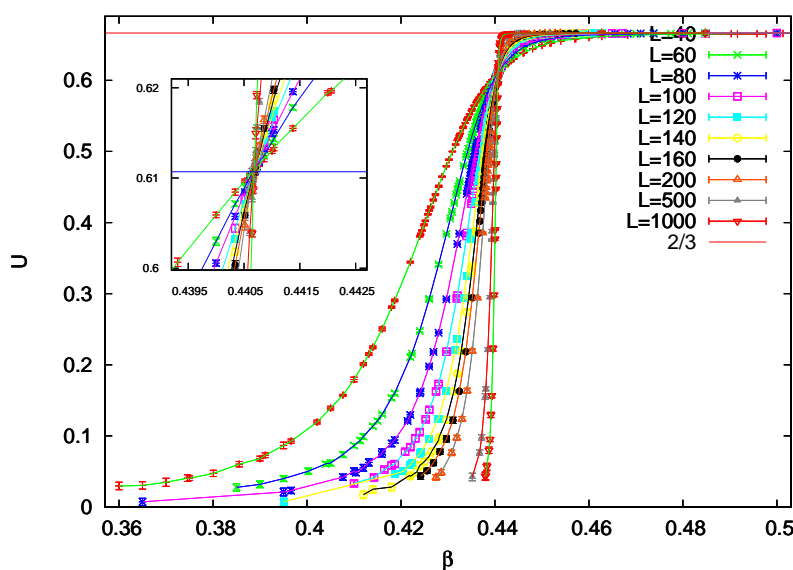
Στην πράξη βρίσκεται πως οι διορθώσεις στην τιμή  $U^*$  από φαινόμενα επίδρασης πεπερασμένου μεγέθους είναι πολύ μικρές, οπότε ο υπολογισμός του  $U(\beta, L)$  δίνει με πολύ καλή ακρίβεια την κρίσιμη θερμοκρασία  $\beta_c$ . Το σημείο τομής των  $U(\beta, L)$  στο σημείο  $(\beta_c, U^*)$  για διαφορετικά  $L$  θα δίνει μια πολύ καλή εκτίμηση της  $\beta_c$ .

Στο Σχήμα 6.17 δείχνουμε τις μετρήσεις μας για το  $U(\beta, L)$ . Φαίνεται η εντυπωσιακή τομή των καμπύλων στο κρίσιμο σημείο  $(\beta_c, U^*)$ . Στον πίνακα 6.5 κάνουμε μια απόπειρα για συστηματικό υπολογισμό της  $\beta_c$  υπολογίζοντας την κρίσιμη θερμοκρασία από την τομή των  $U(\beta, L)$  για τρεις τιμές του  $L$ . Λαμβάνοντας υπόψη όλες τις μετρήσεις για  $L = 100 - 1000$  το αποτέλεσμα που παίρνουμε είναι

$$\beta_c = 0.440678(9) \quad U^* = 0.6107(4), \quad (6.37)$$

που είναι σε πολύ καλή συμφωνία με τις αναμενόμενες τιμές  $\beta_c = 0.44068679 \dots$ ,  $U^* = 0.610690(1)$ . Παρατηρούμε ότι στον υπολογισμό του  $U^*$  ενώ το συστηματικό σφάλμα λόγω πεπερασμένου μεγέθους μειώνεται καθώς αυξάνουμε το  $L$ , αυξάνεται το στατιστικό λόγω της αύξησης της κλίσης της καμπύλης  $U(\beta, L)$  κοντά στο σημείο  $\beta = \beta_c$ . Ο προσδιορισμός όμως της  $\beta_c$  γίνεται με αυξανόμενη ακρίβεια.

<sup>22</sup>Στη στατιστική το cumulant 4ης τάξης μιας τυχαίας μεταβλητής  $x$  ισούται με  $\kappa_4 = \langle (x - \langle x \rangle)^4 \rangle - 3 \langle (x - \langle x \rangle)^2 \rangle^2$  και  $\kappa_2 = \langle (x - \langle x \rangle)^2 \rangle$ , οπότε  $U = -\kappa_4/3\kappa_2^2$  για  $x = m$  και  $\langle m \rangle = 0$ .



Σχήμα 6.17: Ο Binder cumulant για διαφορετικές θερμοκρασίες και πλεγματικά μεγέθη. Στο ένθετο μεγεθύνεται η κρίσιμη περιοχή. Η οριζόντια γραμμή είναι η αναμενόμενη τιμή  $U^* = 0.610690(1)$  [33].

Οι σχέσεις βάθμισης πεπερασμένου μεγέθους μπορούν να χρησιμοποιηθούν και για τον Binder cumulant προκειμένου να προσδιοριστεί και έτσι η κρίσιμη θερμοκρασία και ο κρίσιμος εκθέτης  $1/\nu$ . Από τις σχέσεις (Γ'.17) (Γ'.47) του Παραρτήματος Γ' περιμένουμε ότι η βάθμιση του  $U$  δίνεται από

$$U = F_U(x) = F_U(L^{1/\nu}t). \quad (6.38)$$

Αυτό επιβεβαιώνεται εντυπωσιακά στο Σχήμα 6.18. Από την τιμή  $F_U(x=0)$  παίρνουμε  $U^* = 0.6107(4)$  σε συμφωνία με την (6.37).

Ο απευθείας υπολογισμός των κρίσιμων εκθετών, ιδιαίτερα του  $1/\nu$ , είναι αρκετά δύσκολος σε ένα γενικό υπό μελέτη σύστημα. Οπότε είναι θεμιτό να πάρει κανείς επί πλέον εκτιμήσεις από ποσότητες που είναι εύκολο να μετρηθούν και έχουν γνωστές και καλές ιδιότητες βάθμισης. Παρακάτω παραθέτουμε μερικές από αυτές<sup>23</sup>. Χαρακτηριστικό τους είναι ότι δίνονται από το συσχετισμό της τιμής της ενέργειας με την τιμή (κάποιας δύναμης της) μαγνήτισης.

Η παράγωγος του Binder Cumulant είναι

$$D_U = \frac{\partial U}{\partial \beta} = \frac{\langle m^4 E \rangle \langle m^2 \rangle + \langle m^4 \rangle (\langle m^2 \rangle \langle E \rangle - 2 \langle m^2 E \rangle)}{3 \langle m^2 \rangle^3} \quad (6.39)$$

<sup>23</sup>Ιδιαίτερη επιτυχία είχαν στη μελέτη του 3d πρότυπου Ising [35].

L			$\beta_c$	$U^*$
40	60	80	0.44069(4)	0.6109(5)
60	80	100	0.44069(4)	0.6108(7)
80	100	120	0.44068(4)	0.6108(7)
100	120	140	0.44069(4)	0.6108(11)
120	140	160	0.44069(4)	0.6109(20)
140	160	200	0.44067(3)	0.6102(12)
160	200	500	0.44067(2)	0.6105(10)
200	500	1000	0.44068(1)	0.6106(9)

Πίνακας 6.5: Υπολογισμός της κρίσιμης θερμοκρασίας  $\beta_c$  και της τιμής  $U^*$  από την τομή των καμπύλων  $U(\beta, L)$  που δείχνονται στο Σχήμα 6.17. Για τον υπολογισμό χρησιμοποιούνται κάθε φορά τρεις τιμές του  $L$ . Οι αναμενόμενες τιμές από τη θεωρία και τη βιβλιογραφία [33] είναι  $\beta_c = 0.44068679\dots$ ,  $U^* = 0.610690(1)$ .

Η βάθμισή του δίνεται από τη σχέση (Γ'.48)

$$D_U = L^{1/\nu} F_{D_U}(x) = L^{1/\nu} F_{D_U}(L^{1/\nu} t). \quad (6.40)$$

η οποία αναπαρίσταται στο Σχήμα 6.19. Παρατηρούμε ότι η  $D_U$  ορίζει και αυτή μια ψευδοκρίσιμη περιοχή από το μέγιστο των αντίστοιχων διακυμάνσεων. Η βάθμιση και θέση του μεγίστου θα μπορούσε να μας δώσει τον κρίσιμο εκθέτη  $1/\nu$  σύμφωνα με την ανάλυση που κάναμε στα Σχήματα 6.10 και 6.12 για τη μαγνητική επιδεκτικότητα.

Επίσης μπορεί να φανεί χρήσιμη η μελέτη συναρτήσεων συσχέτισμού της μορφής

$$D_{\ln m^n} = \frac{\partial \ln \langle m^n \rangle}{\partial \beta} = \langle E \rangle - \frac{\langle E m^n \rangle}{\langle m^n \rangle}, \quad (6.41)$$

των οποίων η βάθμιση δίνεται από την εξίσωση (Γ'.54) του Παραρτήματος Γ'

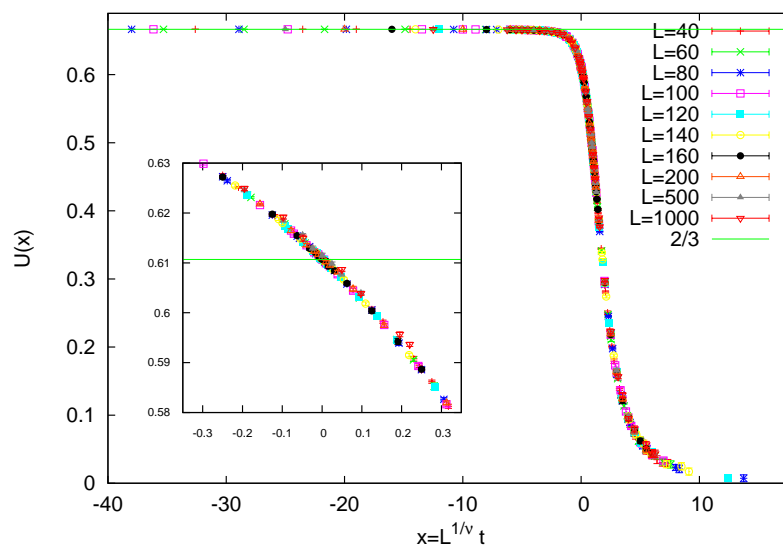
$$D_{\ln m^n} = L^{1/\nu} F_{D_{\ln m^n}}(x) = L^{1/\nu} F_{D_{\ln m^n}}(L^{1/\nu} t). \quad (6.42)$$

Ειδικότερα μας ενδιαφέρει η περίπτωση  $n = 1$

$$D_{\ln |m|} = \frac{\partial \ln \langle |m| \rangle}{\partial \beta} = \langle E \rangle - \frac{\langle E |m| \rangle}{\langle |m| \rangle}, \quad (6.43)$$

και  $n = 2$

$$D_{\ln m^2} = \frac{\partial \ln \langle m^2 \rangle}{\partial \beta} = \langle E \rangle - \frac{\langle E m^2 \rangle}{\langle m^2 \rangle}. \quad (6.44)$$



Σχήμα 6.18: Βάθμιση του Binder cumulant για  $1/\nu = 1$  χρησιμοποιώντας την ακριβή τιμή της  $\beta_c$  στην ανηγμένη θερμοκρασία  $t = (\beta_c - \beta)/\beta_c$ . Στο ένθετο μεγεθύνεται η κρίσιμη περιοχή. Η οριζόντια γραμμή είναι η αναμενόμενη τιμή  $U^* = 0.610690(1)$  [33].

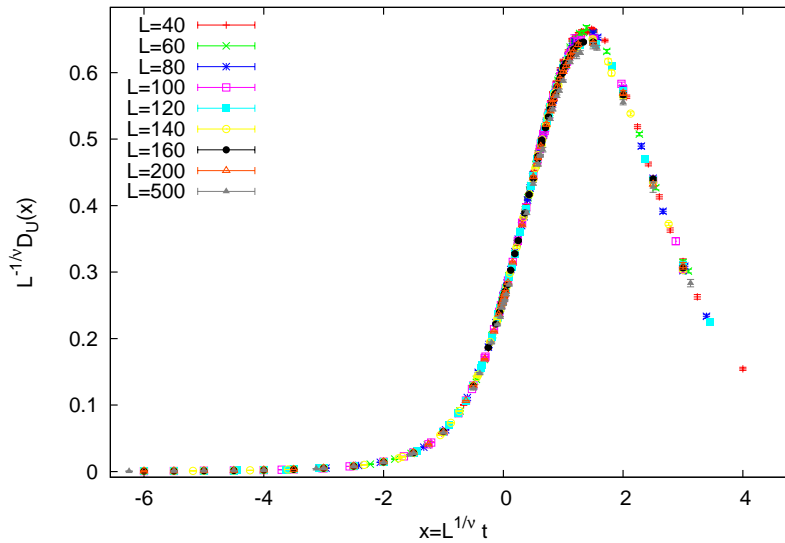
Τέλος αναφέρουμε το Energy Cumulant  $V$

$$V = 1 - \frac{\langle e^4 \rangle}{3 \langle e^2 \rangle^2}. \quad (6.45)$$

Στην εργασία [36] δείχνεται ότι σε μια μετάβαση φάσης δεύτερης τάξης στο κρίσιμο σημείο  $V^* = 2/3$  ενώ σε μια μετάβαση φάσης πρώτης τάξης παίρνουμε μη τετριμμένη τιμή. Έτσι αυτή η ποσότητα μπορεί να χρησιμοποιηθεί ως παράμετρος διαφοροποίησης μιας μετάβασης φάσης 1ης τάξης. Αυτό επιβεβαιώνεται στο Σχήμα 6.22. Τα ελάχιστα των καμπύλων  $V(\beta, L)$  τείνουν προς την κρίσιμη θερμοκρασία σύμφωνα με την σχέση (6.22).

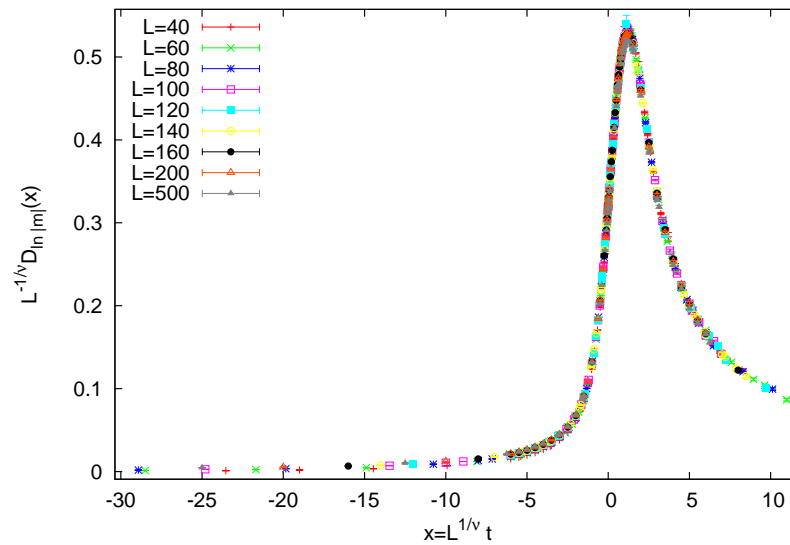
## 6.7 ΠΡΟΒΛΗΜΑΤΑ

Τα δεδομένα που θα χρειαστείτε για τους υπολογισμούς σας θα τα βρείτε στο συνοδευτικό λογισμικό στα αρχεία ch06/all και ch06/all.em. Μπορείτε φυσικά να συλλέξετε και τα δικά σας δεδομένα.



Σχήμα 6.19: Βάθμιση της παραγώγου του Binder cumulant  $D_U$  (βλ. εξίσωση (6.39)) για  $1/\nu = 1$  χρησιμοποιώντας την ακριβή τιμή της  $\beta_c$  στην ανηγμένη θερμοκρασία  $t = (\beta_c - \beta)/\beta_c$ .

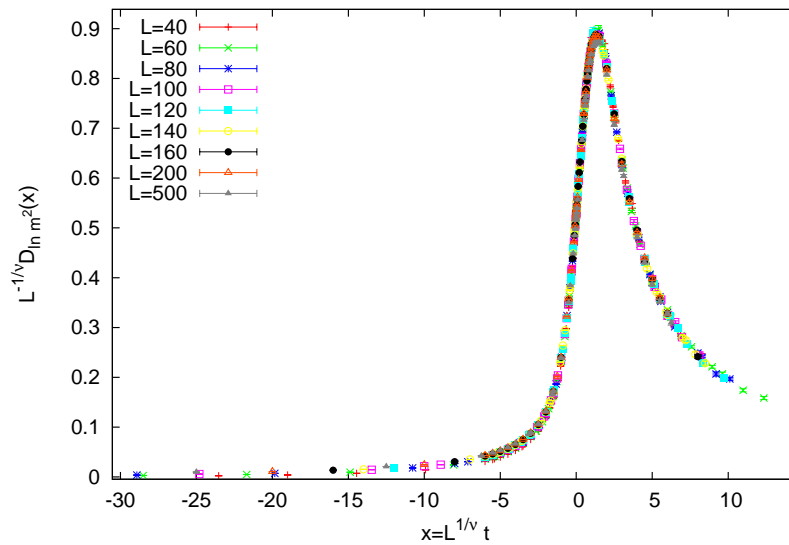
1. Εκτελέστε προσομοιώσεις με σκοπό να προσδιορίσετε τα μέγιστα της μαγνητικής επιδεκτικότητας και της ειδικής θερμότητας με ακρίβεια ανάλογη με αυτή που παρουσιάζεται στον πίνακα 6.4 για  $L = 10, 20, 40, 80, 100$ . Υπολογίστε τις αντίστοιχες ψευδοκρίσιμες θερμοκρασίες. Σε κάθε περίπτωση να κάνετε εκτίμηση των σφαλμάτων.
2. Υπολογίστε τις ασυμπτωτικές σχέσεις που δίνουν τον κρίσιμο εκθέτη  $\gamma/\nu$  προσαρμόζοντας τα δεδομένα για  $t = (T - T_c)/T_c$  αντί για  $t = (\beta_c - \beta)/\beta_c$  όπως στο Σχήμα 6.11. Σχολιάστε τις διαφορές.
3. Εκτελέστε τις προσαρμογές που οδηγούν στα αποτελέσματα (6.10), (6.11), (6.14) και (6.15)
4. Μελετήστε τη βάθμιση της ειδικής θερμότητας σαν συνάρτηση της θερμοκρασίας. Συγκρίνετε την ποιότητα των προσαρμογών στις συναρτήσεις  $a \log |t|$  και  $a |t|^\alpha$  μετρώντας το  $\chi^2/\text{dof}$  όπως περιγράφεται στο Παράρτημα Α' μετά τη Σελ. 213.
5. Δίνεται ο πίνακας 6.6 των μετρήσεων των τιμών των  $\chi(\beta_c, L)$ ,  $\langle m \rangle(\beta_c, L)$  και  $c(\beta_c, L)$ . Κάντε τις απαραίτητες προσαρμογές



Σχήμα 6.20: Βάθμιση της  $D_{\ln|m|}$  (βλ. εξίσωση (6.43)) για  $1/\nu = 1$  χρησιμοποιώντας την ακριβή τιμή της  $\beta_c$  στην ανηγμένη θερμοκρασία  $t = (\beta_c - \beta)/\beta_c$ .

ώστε να πάρετε τους εκθέτες  $\gamma/\nu$ ,  $\beta/\nu$  και  $\alpha$  όπως περιγράφεται στο κείμενο. Ειδικά για τον  $\alpha$  να δοκιμάσετε προσαρμογή σε δύναμη και σε λογάριθμο και να συγκρίνετε τα αποτελέσματα σύμφωνα με το κείμενο.

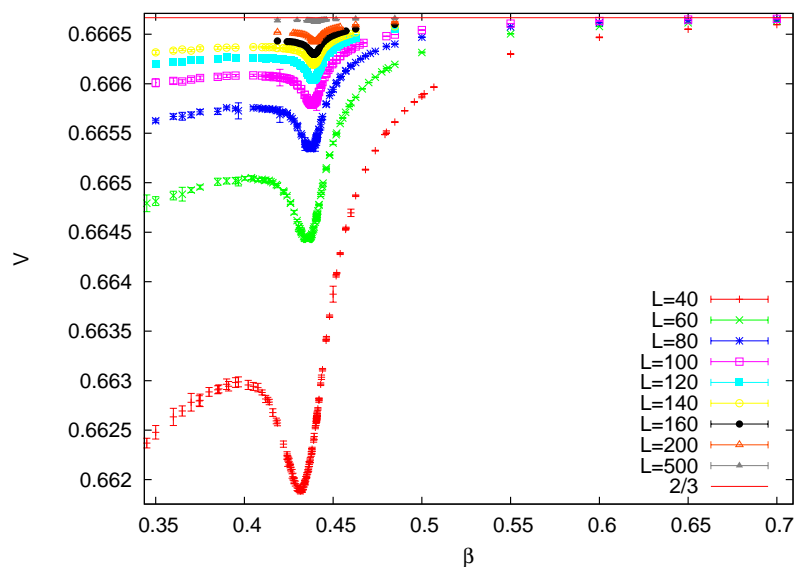
6. Δίνεται ο πίνακας 6.4 των μετρήσεων των τιμών των  $L$ ,  $\beta_c(L)$ ,  $\chi_{\max}$ ,  $\beta'_c(L)$  και  $c_{\max}$ . Κάντε τις απαραίτητες προσαρμογές ώστε να πάρετε τους εκθέτες  $1/\nu$ ,  $\gamma/\nu$ ,  $\alpha/\nu$  και την κρίσιμη θερμοκρασία  $\beta_c$  όπως περιγράφεται στο κείμενο. Ειδικά για τον  $\alpha$  να δοκιμάσετε προσαρμογή σε δύναμη και σε λογάριθμο και να συγκρίνετε τα αποτελέσματα σύμφωνα με το κείμενο.
7. Αναπαράγετε τις “καταρρεύσεις” των σχημάτων 6.14-6.16. Χρησιμοποιήστε τα δεδομένα από το αρχείο a11 από το συνοδευτικό λογισμικό, το οποίο περιγράφεται στο κείμενο. Καθορίστε αρχικά τις γνωστές παραμέτρους και υπολογίστε τις συναρτήσεις βάθμισης  $F_{\chi,m,c}$ . Μεταβάλετε κάθε παράμετρο χωριστά μέχρι η “κατάρρευση” να μην είναι ικανοποιητική και πάρετε τη μεταβολή της παραμέτρου να είναι το σφάλμα προσδιορισμού της. Προσδιορίστε το διάστημα της μεταβλητής  $x = L^{1/\nu} t$  όπου η βάθμιση είναι ικανοποιητική. Επαναλάβετε τον υπολογισμό σας, κάνοντας τις απαραίτητες μετρή-



Σχήμα 6.21: Βάθμιση της  $D_{\ln m^2}$  (βλ. εξίσωση (6.44)) για  $1/\nu = 1$  χρησιμοποιώντας την ακριβή τιμή της  $\beta_c$  στην ανηγμένη θερμοκρασία  $t = (\beta_c - \beta)/\beta_c$ .

σεις για  $L = 10, 20$ , και κρατώντας τις μετρήσεις μόνο για  $L = 10, 20, 40, 80, 120$ . Συγκρίνετε τα αποτελέσματα που θα πάρετε με τα προηγούμενα.

8. Αποδείξτε ότι για οποιαδήποτε παρατηρήσιμη ποσότητα  $\mathcal{O}$  ισχύει  $\partial \langle \mathcal{O} \rangle / \partial \beta = - \langle E \mathcal{O} \rangle + \langle \mathcal{O} \rangle \langle E \rangle = - \langle (E - \langle E \rangle)(\mathcal{O} - \langle \mathcal{O} \rangle) \rangle$ . Από τη σχέση αυτή υπολογίστε την παράγωγο του Binder Cumulant  $D_U$  και αποδείξτε τη σχέση (6.39).
9. Από το μέγιστο της παραγώγου του Binder Cumulant  $D_U$ , υπολογίστε τον κρίσιμο εκθέτη  $1/\nu$  σύμφωνα με την ανάλυση που κάναμε στα Σχήματα 6.10 και 6.12 για τη μαγνητική επιδεκτικότητα.



Σχήμα 6.22: Ο Energy Cumulant που ορίζεται στην εξίσωση (6.45). Καθώς αυξάνεται το μέγεθος του συστήματος, η τιμή του τείνει στην τιμή  $2/3$  όπως αναμένεται σε μια μετάβαση φάσης 2ης τάξης. Η θέση των ελαχίστων τείνουν στην κρίσιμη θερμοκρασία σαν  $L^{-1/\nu}$

$L$	$\chi(\beta_c, L)$		$\langle m \rangle(\beta_c, L)$		$c(\beta_c, L)$	
40	20.50	0.02	0.6364	0.0001	0.4883	0.0007
60	41.78	0.08	0.6049	0.0002	0.5390	0.0008
80	69.15	0.09	0.5835	0.0001	0.5743	0.0012
100	102.21	0.25	0.5673	0.0002	0.6026	0.0014
120	140.18	0.11	0.5548	0.0001	0.6235	0.0010
140	183.95	0.33	0.5442	0.0002	0.6434	0.0006
160	232.93	0.55	0.5351	0.0001	0.6584	0.0020
200	342.13	0.72	0.5206	0.0001	0.6858	0.0014
500	1687.2	4.4	0.4647	0.0002	0.7794	0.0018
1000	6245	664	0.4228	0.0040	-	-

Πίνακας 6.6: Ο πίνακας των τιμών  $\chi(\beta_c, L)$ ,  $\langle m \rangle(\beta_c, L)$  και  $c(\beta_c, L)$  στην κρίσιμη θερμοκρασία για διαφορετικά  $L$  που δίνεται για την άσκηση 5.



**Μέρος II**  
**ΠΑΡΑΡΤΗΜΑΤΑ**



## ΠΑΡΑΡΤΗΜΑ Α'

# ΣΥΝΑΡΤΗΣΗ ΚΑΙ ΧΡΟΝΟΣ ΑΥΤΟΣΥΣΧΕΤΙΣΜΟΥ

Στο παράρτημα αυτό θα δείξουμε τις τεχνικές λεπτομέρειες του υπολογισμού της συνάρτησης αυτοσυσχετισμού (4.22) και του χρόνου αυτοσυσχετισμού (4.23) και (4.25). Σε ένα δείγμα από  $n$  μετρήσεις  $\mathcal{O}(0), \mathcal{O}(1), \dots, \mathcal{O}(n-1)$  θα χρησιμοποιήσουμε ως εκτιμητή της σχέσης (4.22) την

$$\rho_{\mathcal{O}}(t) = \frac{1}{\rho_0} \frac{1}{n-t} \sum_{t'=0}^{n-1-t} (\mathcal{O}(t') - \langle \mathcal{O} \rangle_0)(\mathcal{O}(t'+t) - \langle \mathcal{O} \rangle_t) \quad (\text{A'.1})$$

όπου οι μέσες τιμές ορίζονται από τις σχέσεις<sup>1</sup>

$$\langle \mathcal{O} \rangle_0 \equiv \frac{1}{n} \sum_{t'=0}^{n-1} \mathcal{O}(t') \quad \langle \mathcal{O} \rangle_t \equiv \frac{1}{n-t} \sum_{t'=0}^{n-1-t} \mathcal{O}(t'+t). \quad (\text{A'.2})$$

Η σταθερά  $\rho_0$  επιλέγεται έτσι ώστε  $\rho_{\mathcal{O}}(0) = 1$

Το πρόγραμμα για τον υπολογισμό της συνάρτησης (4.22) και του χρόνου (4.25) είναι απλό να γραφτεί. Παραθέτουμε το αρχείο `autoc.c` από το συνοδευτικό λογισμικό. Διαβάστε τα ένθετα σχόλια για επεξηγήσεις των σημαντικών βημάτων.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
```

<sup>1</sup>Θα μπορούσαμε να πάρουμε  $\langle \mathcal{O} \rangle_0 = \langle \mathcal{O} \rangle_t = (1/n) \sum_{t'=0}^{n-1} \mathcal{O}(t')$  χωρίς μεγάλη διαφορά. Με τον ορισμό που χρησιμοποιούμε έχουμε ελαφρά μικρότερα φαινόμενα πεπερασμένου μεγέθους του δείγματος. Επίσης η επιλογή μας (A'.1) αντί για την  $\rho_{\mathcal{O}}(t) \propto (1/(n-t)) \sum_{t'=0}^{n-1-t} \mathcal{O}(t')\mathcal{O}(t'+t) - \langle \mathcal{O} \rangle_0 \langle \mathcal{O} \rangle_t$  μειώνει τα σφάλματα στρογγυλοποίησης.

## 210ΠΑΡΑΡΤΗΜΑ Α'. ΣΥΝΑΡΤΗΣΗ ΚΑΙ ΧΡΟΝΟΣ ΑΥΤΟΣΥΣΧΕΤΙΣΜΟΥ

```
#include <unistd.h>
#include <libgen.h>

#define STRLEN 200
void get_the_options(int ,char **),usage(char **),locerr(char *);
char prog[STRLEN];
int NMAX,tmax;

double rho(double *,int, int);
int main(int argc, char **argv){
    double *r,*tau,*x,norm;
    int i,ndat=0,t,tcut;
    /*Default values for max number of data and maximum time for rho and tau*/
    /*Use -n and -t options to change them. */
    NMAX=2000000;tmax=1000;//NMAX=2000000 requires ~ 2e6*8=16MB of memory
    strcpy(prog,(char *)basename(argv[0]));//program name stored in prog
    get_the_options(argc,argv);
    /*Allocate the array where we store the input data:*/
    x = (double *)malloc( NMAX * sizeof(double)); //data
    /*Read data without exceeding NMAX as long as it is provided from stdin*/
    while( (ndat < NMAX) && (scanf("%lf",&x[ndat++]) == 1));
    if( ndat >= NMAX)
        fprintf(stderr,"# %s: Warning: Read ndat= %d and reached the limit %d\n",
            prog,ndat,NMAX);
    /*we decrease tmax if it is comparable or larger to number of data.*/
    if(tmax > (ndat/10)) tmax = ndat/10;
    /*r[t] stores the values of autocorrelation function rho(t)*/
    r = (double *)malloc( tmax * sizeof(double)); //autocorrelation function
    for(t=0;t<tmax;t++) r[t] = rho(x,ndat,t); //rho(t)
    norm=1.0/r[0];for(t=0;t<tmax;t++)r[t] *= norm; //normalize r[0]=1
    /*tau[t] stores integrated autocorrelation times with tcut=t*/
    tau = (double *)malloc( tmax * sizeof(double)); //tau_int
    for(tcut=0;tcut<tmax;tcut++){
        tau[tcut] = 0.0;
        for(t=0;t<=tcut;t++) tau[tcut] += r[t];//sum of r[t]
    }
    /* Output: an informational header and the results.*/
    printf("# =====\n");
    printf("# Autoc function rho and integrated autoc time tau \n");
    printf("# ndat= %d tmax= %d\n",ndat,tmax);
    printf("# t rho(t) tau(tcut=t) \n");
    printf("# =====\n");
    /*Print the results in three columns:*/
    for(t=0;t<tmax;t++)
```

```

    printf("%d  %.15lg  %.15lg\n",t,r[t],tau[t]);
}/*main*/

double rho(double *x,int ndat,int t){
    int n,t0;
    double xav0=0.0,xavt=0.0,r=0.0;
    n=ndat-t;
    if(n<1)locerr("rho: n<1");
    /*Calculate the two averages: xav0= <x>_0 and xavt= <x>_t */
    for(t0=0;t0<n;t0++){
        xav0 += x[t0];
        xavt += x[t0+t];
    }
    xav0/=n;xavt/=n; //normalize the averages to number of data
    /*Calculate the t-correlations:*/
    for(t0=0;t0<n;t0++){
        r += (x[t0]-xav0)*(x[t0+t]-xavt);
    }
    r/=n; //normalize the averages to number of data
    return r;
}

void locerr( char *errmes ){
    fprintf(stderr,"%s: %s Exiting...\n",prog,errmes);
    exit(1);
}

#define OPTARGS "?h1234567890.t:n:"
void get_the_options(int argc,char **argv){

    int c,errflg = 0;
    while (!errflg && (c = getopt(argc, argv, OPTARGS)) != -1){
        switch(c){
            case 'n':
                NMAX = atoi(optarg);
                break;
            case 't':
                tmax = atoi(optarg);
                break;
            case 'h':
                errflg++;/*call usage*/
                break;
            default:
                errflg++;
        }/*switch*/
        if(errflg) usage(argv);
    }/*while...*/
}

```

## 212 ΠΑΡΑΡΤΗΜΑ Α'. ΣΥΝΑΡΤΗΣΗ ΚΑΙ ΧΡΟΝΟΣ ΑΥΤΟΣΥΣΧΕΤΙΣΜΟΥ

```
}/*get_the_options()*/  
void usage(char **argv){  
    fprintf(stderr, "\n\  
Usage: %s [-t <maxtime>] [-n <ndata>]\n\  
    Reads data from stdin (one column) and computes autocorrelation\  
    function and integrated autocorrelation time.\n", prog);  
    exit(1);  
}/*usage()*/
```

Για το μεταγλωττισμό του προγράμματος χρησιμοποιούμε την εντολή

```
% cc -O2 autoc.c -o autoc
```

και αν έχουμε τα δεδομένα μας σε μία στήλη στο αρχείο data υπολογίζουμε τη συνάρτηση αυτοσυσχετισμού και τον ολοκληρωμένο χρόνο αυτοσυσχετισμού με την εντολή

```
% cat data | ./autoc > data.rho
```

όπου τα αποτελέσματα είναι σε τρεις στήλες. Η πρώτη είναι ο χρόνος  $t$ , η δεύτερη η  $\rho(t)$  και η τρίτη η  $\tau_{\text{int},\rho}(t)$  (σχέση (4.26)). Τα αντίστοιχα σχήματα φτιάχνονται με τις εντολές

```
% gnuplot  
gnuplot> plot "data.rho" using 1:2 with lines  
gnuplot> plot "data.rho" using 1:3 with lines
```

Αν θέλουμε να αλλάξουμε το μέγιστο αριθμό δεδομένων  $N_{\text{MAX}}$  (λ.χ. αν έχουμε περισσότερα δεδομένα ή αν θέλουμε να μειώσουμε τη μνήμη που δεσμεύει το πρόγραμμα) ή το μέγιστο χρόνο  $T_{\text{MAX}}$  στη συνάρτηση αυτοσυσχετισμού χρησιμοποιούμε τα options  $-n$  και  $-t$  αντίστοιχα:

```
% cat data | autoc -n 10000000 -t 20000 > data.rho
```

Για την απευθείας εισαγωγή των δεδομένων δοκιμάστε την εντολή:

```
% gnuplot  
gnuplot> plot "<./is -L 20 -b 0.4407 -s 1 -S 345 -n 400000|\n    grep -v '#'|awk '{print ($2>0)?$2:-$2;}' |\n    autoc -t 500" using 1:2 with lines
```

Η παραπάνω gnuplot εντολή είναι σπασμένη σε 3 γραμμές για να φαίνεται στη σελίδα. Για να τη χρησιμοποιήσετε ενώστε τις γραμμές και αφαιρέστε τις  $\backslash$ .

Στη συνέχεια παραθέτουμε το script autoc\_L το οποίο κάνει τους απαραίτητους υπολογισμούς για το σχήμα 4.11.

```
#!/bin/tcsh -f

set nmeas = 2100000
set Ls    = (5 10 20 40 60 80)
set beta  = 0.4407
set tmax  = 2000
foreach L ($Ls)
  set N    = `awk -v L=$L 'BEGIN{print L*L}'`
  set rand = `perl -e 'srand();print int(3000000*rand()+1);`
  set out  = outL${L}b${beta}
  echo "Running L${L}b${beta}"
  ./is -L $L -b $beta -s 1 -S $rand -n $nmeas > $out
  echo "Autocorrelations L${L}b${beta}"
  grep -v '#' $out | \
    awk -v N=$N 'NR>100000{print ($2>0)?($2/N):(-$2/N)}'| \
    autoc -t $tmax > $out.rhom
end
```

Στη συνέχεια κατασκευάζουμε το σχήμα με το gnuplot:

```
% gnuplot
gnuplot> plot "outL5b0.4407.rhom" using 1:2 with lines title "5"
gnuplot> replot "outL10b0.4407.rhom" using 1:2 with lines title "10"
gnuplot> replot "outL20b0.4407.rhom" using 1:2 with lines title "20"
gnuplot> replot "outL40b0.4407.rhom" using 1:2 with lines title "40"
gnuplot> replot "outL60b0.4407.rhom" using 1:2 with lines title "60"
gnuplot> replot "outL80b0.4407.rhom" using 1:2 with lines title "80"
```

Με παρόμοιο τρόπο φτιάχνουμε και τα σχήματα 4.10.

Για τον υπολογισμό του  $\tau_m$  εργαζόμαστε ως εξής:

```
% gnuplot
gnuplot> f(x) = c * exp(-x/t)
gnuplot> set log y
gnuplot> plot [:1000] "outL40b0.4407.rhom" using 1:2 with lines
gnuplot> c = 1 ; t = 300
gnuplot> fit [150:650] f(x) "outL40b0.4407.rhom" using 1:2 via c,t
gnuplot> plot [:1000] "outL40b0.4407.rhom" using 1:2 with lines,f(x)
gnuplot> plot [:] "outL40b0.4407.rhom" using 1:3 with lines
```

όπου με την τελευταία γραμμή συγκρίνουμε με την τιμή του  $\tau_{int,m}$ . Η εντολή fit είναι ενδεικτική. Πρέπει να δοκιμαστούν διάφορες τιμές για τα όρια που εδώ επιλέχθηκαν να είναι [150:650]. Στην πραγματικότητα αυτό που κάναμε ήταν αφού με την πρώτη εντολή plot είδαμε περίπου πότε η συνάρτηση έχει την αναμενόμενη ασυμπτωτική συμπεριφορά (τη συνάρτηση  $f(x) = c \exp(-x/t)$ ) μεταβά-

## 214ΠΑΡΑΡΤΗΜΑ Α'. ΣΥΝΑΡΤΗΣΗ ΚΑΙ ΧΡΟΝΟΣ ΑΥΤΟΣΥΣΧΕΤΙΣΜΟΥ

λαμε τα άνω και κάτω όρια έτσι ώστε η τιμή για τον  $\tau_m$  να σταθεροποιηθεί ενώ το  $\chi^2/\text{dof}$  της προσαρμογής να είναι ελάχιστο<sup>3</sup>. Το  $\chi^2/\text{dof}$  της προσαρμογής το διαβάζουμε από το αποτέλεσμα της εντολής fit

```

.....
degrees of freedom      (FIT_NDF)                      : 449
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.000939201
variance of residuals (reduced chisquare) = WSSR/ndf  : 8.82099e-07

```

Final set of parameters	Asymptotic Standard Error		
=====	=====		
c	= 0.925371	+/- 0.0003773	(0.04078%)
t	= 285.736	+/- 0.1141	(0.03995%)
.....			

από τη γραμμή "variance of residuals". Από τις επόμενες γραμμές διαβάζουμε τις τιμές των παραμέτρων με τα σφάλματά τους<sup>4</sup> και συμπεραίνουμε ότι  $\tau_m = 285.7 \pm 0.1$ .

Πρέπει να τονιστεί ότι το σφάλμα αυτό είναι το στατιστικό σφάλμα της προσαρμογής για τα δεδομένα όρια που επιλέξαμε και όχι το πραγματικό σφάλμα. Συνήθως, όπως και εδώ, τα μεγαλύτερα σφάλματα στην προσαρμογή των δεδομένων σε μία συνάρτηση είναι συστηματικά που οφείλονται (εδώ) κυρίως στην επιλογή των ορίων<sup>5</sup>. Δοκιμάζοντας διάφορες τιμές μέχρι να διπλασιαστεί η τιμή του  $\chi^2/\text{dof}$  βρίσκουμε  $\tau_m = 285(2)$ .

<sup>2</sup> Αν έχουμε τα δεδομένα  $\{(x_i, y_i)\}, i = 1, \dots, n$  με σφάλμα  $\delta y_i$  τα οποία θέλουμε να προσαρμόσουμε στην  $f(x; c, t) = c e^{-x/t}$ , τότε το  $\chi^2(c, t) = \sum_{i=1}^n (y_i - f(x_i; c, t))^2 / \delta y_i^2$ . Το  $\chi^2/\text{dof}$  είναι κανονικοποιημένο στον αριθμό των βαθμών ελευθερίας (dof = degrees of freedom =  $n - 2$ ) που είναι ο αριθμός των σημείων στην προσαρμογή  $n$  μείον των αριθμό των ελεύθερων παραμέτρων της προσαρμογής (εδώ οι  $c, t$  είναι 2)

<sup>3</sup> Αποδεκτά  $\chi^2/\text{dof} \sim 1$  αλλά επειδή εδώ δεν παρέχουμε τα σφάλματα στην τιμή της συνάρτησης αυτοσυσχετισμού, το  $\chi^2/\text{dof}$  δεν είναι σωστά κανονικοποιημένο. Το πρόγραμμα θέτει  $\delta y_i = 1 \forall i$ .

<sup>4</sup> Στις παρενθέσεις είναι το επίπεδο εμπιστοσύνης (confidence level). Αυτό ορίζεται να είναι η πιθανότητα οι τιμές των παραμέτρων να είναι μέσα στο διάστημα που ορίζει το σφάλμα. Αυτό υποθέτει ότι έχουμε το σωστό στατιστικό βάρος σε κάθε σημείο που δίνεται από το σφάλμα - στο παράδειγμα δεν το έχουμε γι' αυτό και το επίπεδο είναι τόσο χαμηλό. Μια τιμή κάτω από 5% είναι πολύ χαμηλή και υποδηλώνει ότι το μοντέλο έχει ανάγκη από διορθώσεις. Οι αριθμητικές τιμές υποθέτουν Gaussian κατανομή των μετρήσεων και αν η υπόθεση αυτή δεν ισχύει πρέπει να ληφθούν απλά σαν μια ποιοτική ένδειξη.

<sup>5</sup> Για ένα προσεκτικό υπολογισμό μιας ποσότητας που προκύπτει από προσαρμογή δεδομένων, δοκιμάζουμε και άλλες συναρτήσεις που μπορεί να περιέχουν διορθώσεις της ασυμπτωτικής συμπεριφοράς.



Στη συγκεκριμένη περίπτωση το μεγαλύτερο συστηματικό σφάλμα προέρχεται από την παράλειψη της επίδρασης και άλλων χρόνων αυτοσυσχετισμού του συστήματος. Λάβαμε υπόψη μόνο την κύρια συνεισφορά από τον μεγαλύτερο χρόνο αυτοσυσχετισμού προσαρμόζοντας τα δεδομένα στη συνάρτηση

$$f(t) = c e^{-t/\tau}. \quad (\text{A'.3})$$

Στην πραγματικότητα λαμβάνοντας υπ' όψη και μικρότερους χρόνους αυτοσυσχετισμού περιμένουμε  $\rho_m(t) \sim a_1 e^{-t/\tau_1} + a_2 e^{-t/\tau_2} + \dots$ . Βρίσκουμε πως τα δεδομένα μας προσαρμόζονται με πολύ σταθερό τρόπο στη συνάρτηση

$$h(x) = a_1 e^{-x/\tau_1} + a_2 e^{-x/\tau_2} + a_3 e^{-x/\tau_3}. \quad (\text{A'.4})$$

Όπως φαίνεται και στο σχήμα A'.1 πετυχαίνουμε άριστη προσαρμογή για πολύ μικρούς χρόνους και τα αποτελέσματά μας για τον κύριο χρόνο αυτοσυσχετισμού  $\tau_m \equiv \tau_1 = 286.3(3)$ . Οι δευτερεύοντες χρόνοι αυτοσυσχετισμού είναι  $\tau_2 = 57(3)$ ,  $\tau_3 = 10.5(8)$  οι οποίοι είναι αρκετά μικρότεροι από τον  $\tau_1$ .

Για διευκόλυνση του αναγνώστη παραθέτουμε τις βασικές εντολές για την αναφερόμενη ανάλυση:

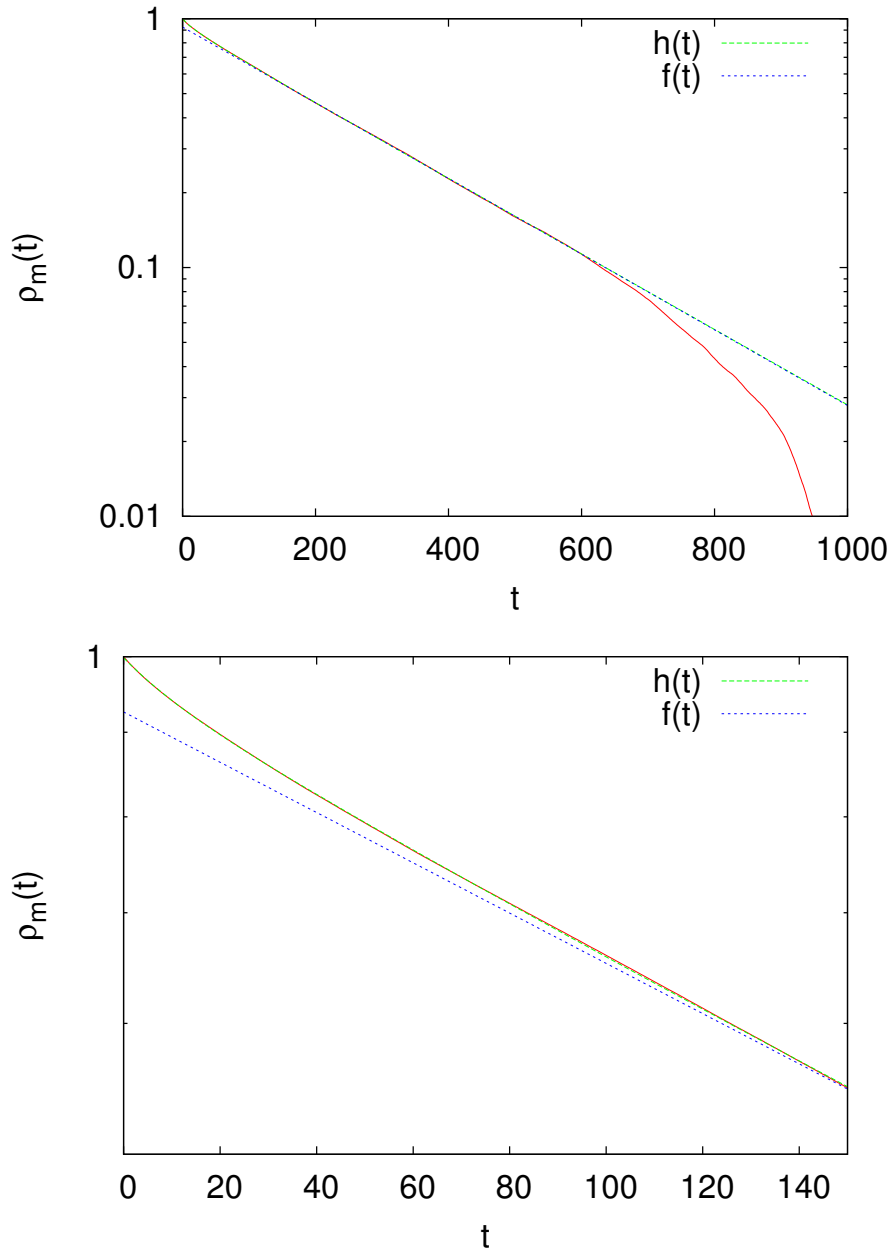
```
gnuplot> h(x) = a1 * exp(-x/t1) + a2 * exp(-x/t2) + a3 * exp(-x/t3)
gnuplot> a1 = 1; t1 = 285; a2 = 0.04; t2 = 56; a3 = 0.03; t3 = 10
gnuplot> fit [1:600] h(x) "outL40b0.4407.rhom" \
    using 1:2 via a1,t1,a2,t2,a3,t3
```

...

Final set of parameters		Asymptotic	Standard Error
=====		=====	=====
a1	= 0.922111	+/- 0.001046	(0.1135%)
t1	= 286.325	+/- 0.2354	(0.08221%)
a2	= 0.0462523	+/- 0.001219	(2.635%)
t2	= 56.6783	+/- 2.824	(4.982%)
a3	= 0.0300761	+/- 0.001558	(5.18%)
t3	= 10.5227	+/- 0.8382	(7.965%)

```
gnuplot> plot [:150] [0.5:] "outL40b0.4407.rhom" using 1:2 \
    with lines notit,h(x) ,f(x)
gnuplot> plot [:1000] [0.01:] "outL40b0.4407.rhom" using 1:2 \
    with lines notit,h(x) ,f(x)
```

216 ΠΑΡΑΡΤΗΜΑ Α'. ΣΥΝΑΡΤΗΣΗ ΚΑΙ ΧΡΟΝΟΣ ΑΥΤΟΣΥΣΧΕΤΙΣΜΟΥ



Σχήμα Α'.1: Προσαρμογή της συνάρτησης αυτοσυσχετισμού  $\rho_m(t)$  στις συναρτήσεις  $f(t) = ce^{-t/\tau}$  και  $h(t) = a_1e^{-t/\tau_1} + a_2e^{-t/\tau_2} + a_3e^{-t/\tau_3}$ . Για μεγάλους χρόνους  $f(t) \approx h(t)$ , αλλά όπως φαίνεται στο κάτω σχήμα η  $h(t)$  είναι αναγκαία για την προσαρμογή των μικρών χρόνων. Η επιλογή αυτή δίνει σταθερή τιμή για τον κύριο χρόνο αυτοσυσχετισμού  $\tau_m = \tau = \tau_1$ . Οι τιμές των παραμέτρων δίνονται στο κείμενο. Οι κάθετοι άξονες και των δύο σχημάτων είναι σε λογαριθμική κλίμακα.

## ΠΑΡΑΡΤΗΜΑ Β'

# ΥΠΟΛΟΓΙΣΜΟΣ ΣΦΑΛΜΑΤΩΝ

### Β'.1 ΜΕΘΟΔΟΣ JACKKNIFE

Εδώ δίνουμε το πρόγραμμα που υπολογίζει τα σφάλματα με τη μέθοδο jackknife που περιγράψαμε στην ενότητα 4.5.2. Στο σχήμα 4.13 φαίνεται πώς χωρίζουμε τα δεδομένα μας σε bins. Σε κάθε bin υπολογίζουμε τη μέση τιμή της ποσότητας  $\mathcal{O}$  που ενδιαφερόμαστε και στη συνέχεια από τη σχέση (4.36) τα σφάλματα. Για την εφαρμογή της μεθόδου παραθέτουμε πρόγραμμα από το αρχείο jack.c (θα το βρείτε στον υποκατάλογο Tools/ στο συνοδευτικό λογισμικό) το οποίο υπολογίζει τις ποσότητες  $\langle \mathcal{O} \rangle$ ,  $\delta \mathcal{O}$ ,  $\chi \equiv \langle (\mathcal{O} - \langle \mathcal{O} \rangle)^2 \rangle$  και  $\delta \chi$ .

```
/****** jack .c *****  
Simple jackknife program for computation of jackknife errors of data.  
The jackknife function can be incerted to any program.  
*/  
#include <stdlib.h>  
#include <stdio.h>  
#include <string.h>  
#include <math.h>  

```

```

void jackknife(int,int,double *,double *,double *,double *,double *);
void get_the_options(int ,char **),usage(char **),locerr(char *);
int main(int argc,char **argv){
    int    ndat=0;
    double 0,d0,chi,dchi;
    double *x;
    strcpy(prog,(char *)basename(argv[0]));
    maxdat=-1;JACK=10;
    get_the_options(argc,argv);
    if( maxdat <= 0 ) maxdat = MAXDAT;
    x = (double *) calloc((size_t) maxdat,(size_t) sizeof(double));
    /*Read data from stdin as long as it fits the array x[] .
       ndat counts the number of data.*/
    while( scanf("%lg",&x[ndat])==1 ){
        ndat++;
        if(ndat == maxdat){//we give user a warning if we ignore some data
            fprintf(stderr,"# %s: Warning: Read ndat= %d. Reached the limit %d\n",
                prog,ndat,maxdat);
            break;          //break the while loop
        }
    }
    jackknife(ndat,JACK,x,&0,&d0,&chi,&dchi);//calculate....
    printf("# NDATA = %d data. JACK = %d groups\n",ndat,JACK);//...and print
    printf("# <o>, chi= (<o^2>-<o>^2)\n");
    printf("# <o> +/- err                chi +/- err\n");
    printf("%.15lg %.15lg %.15lg %.15lg\n",0,d0,chi,dchi);
}
/*****
/* jackknife routine: you can use it in any of your programs...*/
*****/
void jackknife(int ndat, int jack, double *x,
    double *av0, double *er0, double *avchi, double *erchi){

    int i,j,binw,bin;
    double *0,*chi;
    0 = (double *) calloc((size_t) jack ,(size_t) sizeof(double));
    chi = (double *) calloc((size_t) jack ,(size_t) sizeof(double));
    binw = ndat/jack;
    if(binw<1)locerr("jackknife: binw < 1");
    ndat = binw * jack; //only full bins considered
    /* average value */
    for(i=0;i<ndat;i++)
        for(j=0;j<jack;j++){
            if( (i/binw) != j ){ //then we add this data point to the bin

```

```

        0 [j] += x[i];
    }
}
for(j=0;j<jack;j++) 0 [j] /= (ndat-binw); //average in each bin
/* susceptibility: */
for(i=0;i<ndat;i++)
    for(j=0;j<jack;j++){
        if( (i/binw) != j ){ //then we add this data point to the bin
            chi[j] += (x[i]-0[j])*(x[i]-0[j]);
        }
    }
for(j=0;j<jack;j++) chi[j] /= (ndat-binw); //average in each bin
/* Compute averages: */
*av0 = 0.0; *avchi = 0.0;
for(j=0;j<jack;j++){
    *av0 += 0[j]; *avchi += chi[j];
}
*av0 /= jack; *avchi /= jack;
/* Compute errors: */
*er0 = 0.0; *erchi = 0.0;
for(j=0;j<jack;j++){
    *er0 += (0 [j]-*av0 )*(0 [j]-*av0 );
    *erchi += (chi[j]-*avchi)*(chi[j]-*avchi);
}
*er0 = sqrt(*er0); *erchi = sqrt(*erchi);
free(0);free(chi); //release memory back to the system
}
/*****
void locerr( char *errmes ){
    fprintf(stderr,"%s: %s Exiting...\n",prog,errmes);
    exit(1);
}

#define OPTARGS "?hj:d:"
void get_the_options(int argc,char **argv){

    int c,errflg = 0;
    while (!errflg && (c = getopt(argc, argv, OPTARGS)) != -1){
        switch(c){
            case 'j':
                JACK = atoi(optarg);
                break;
            case 'd':
                maxdat = atoi(optarg);

```

```

        break;
    case 'h':
        errflg++;/*call usage*/
        break;
    default:
        errflg++;
    }/*switch*/
    if(errflg) usage(argv);
}/*while...*/
}/*get_the_options()*/

void usage(char **argv){
    fprintf(stderr,"\
Usage: %s [options] <file>\n\
        -j : No. jack groups Def. 10\n\
        -d : Give the maximum number of data points read.Def. %d\n\
Computes <o>, chi= (<o^2>-<o>^2)\n\
Data is in one column from stdin.\n",prog,MAXDAT);
    exit(1);
}/*usage()*/

```

Για τη μεταγλώττιση χρησιμοποιούμε την εντολή

```
% cc -O2 jack.c -o jack -lm
```

Αν τα δεδομένα μας είναι σε μία στήλη στο αρχείο data τότε η εντολή για να κάνουμε τον υπολογισμό των σφαλμάτων με 50 jack bins είναι:

```
% cat data | jack -j 50
```

Το πρόγραμμα δέχεται το πολύ 1,000,000 μετρήσεις (από το macro που ορίσαμε MAXDAT να είναι το 1000000). Αν έχουμε παραπάνω πρέπει να χρησιμοποιήσουμε το διακόπτη -d 2000000 για 2,000,000 μετρήσεις λ.χ. Η επιλογή να διαβάζουμε από το stdin γίνεται για να μπορεί το πρόγραμμα να χρησιμοποιηθεί σε συνδυασμό με φίλτρα που μπορεί να χρησιμοποιήσουμε για την ανάλυση των δεδομένων. Για παράδειγμα, για την ανάλυση της μαγνήτισης με το πρόγραμμα που έχουμε γράψει για το πρότυπο Ising μπορούμε να χρησιμοποιήσουμε την εντολή:

```
% is -L 20 -b 0.4407 -s 1 -S 342 -n 2000000 | grep -v # | \
awk -v L=20 '{print ($2>0)?($2/(L*L)):(-$2/(L*L))}' | \
jack -j 50 -d 2000000 | grep -v # | \
awk -v b=0.4407 -v L=20 '{print $1,$2,b*L*L*$3,b*L*L*$4}'

```

Η παραπάνω εντολή είναι μία γραμμή αν αφαιρέσουμε τις τελευταίες ανάποδα κάθετες γραμμές (backslash - '\'). Για πληρότητα της την εξηγήσουμε: Η πρώτη γραμμή τρέχει το πρόγραμμα `is` για το πρότυπο `Ising` για  $N = L \times L = 20 \times 20$  πλεγματικές θέσεις (`-L 20`) και  $\beta = 0.4407$  (`-b 0.4407`). Αρχίζει από "καυτή" διάταξη σπιν (`-s 1`) και κάνει 2,000,000 μετρήσεις (`-n 2000000`). Φιλτράρουμε με την εντολή `grep -v` τα σχόλια από την έξοδο του προγράμματος. Στη δεύτερη γραμμή καλούμε την `awk` ορίζοντας την μεταβλητή  $L=20$  να τυπώσει την απόλυτη τιμή της δεύτερης στήλης διαιρεμένης με των αριθμό των πλεγματικών θέσεων  $L*L$ . Στην τρίτη γραμμή γίνεται ο υπολογισμός της μέσης τιμής  $\langle m \rangle$  και  $\langle (m - \langle m \rangle)^2 \rangle$  με τα σφάλματά τους από το πρόγραμμα `jack`. Από τα αποτελέσματα αφαιρούνται τα σχόλια με την εντολή `grep -v`. Η τέταρτη γραμμή γίνεται απλά για τον υπολογισμό της μαγνητικής επιδεκτικότητας (4.21) όπου πρέπει να πολλαπλασιάσουμε με τον παράγοντα  $\beta N = \beta L^2$  τις διακυμάνσεις  $\langle (m - \langle m \rangle)^2 \rangle$  και το σφάλμα τους για να πάρουμε την  $\chi$ .

## B'.2 ΜΕΘΟΔΟΣ BOOTSTRAP

Στην υποενότητα αυτή παραθέτουμε το πρόγραμμα για τον υπολογισμό των σφαλμάτων με τη μέθοδο `bootstrap` σύμφωνα με όσα είπαμε στην ενότητα 4.5.3. Το πρόγραμμα κωδικοποιείται στο αρχείο `boot.c`:

```

/*****                               boot.c                               *****/
Simple program for computation of errors of data using bootstrap.
*/
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <unistd.h>
#include <libgen.h>
#include <time.h>

#define MAXDAT 1000000
#define STRLEN 200
char prog[STRLEN];
int SAMPLES,maxdat;

void bootstrap(int,int,double *,double *,double *,double *,double *);
void get_the_options(int ,char **),usage(char **),locerr(char *);

```

```

double drandom();
void init_random();
int main(int argc, char **argv){
    int ndat=0;
    double O,dO,chi,dchi;
    double *x;
    strcpy(prog,(char *)basename(argv[0]));
    maxdat=-1;SAMPLES=1000;
    get_the_options(argc,argv);init_random();
    if( maxdat <= 0 ) maxdat = MAXDAT;
    x = (double *) calloc((size_t) maxdat,(size_t) sizeof(double));
    /*Read data from stdin as long as it fits the array x[] .
    ndat counts the number of data.*/
    while( scanf("%lg",&x[ndat])==1 ){
        ndat++;
        if(ndat == maxdat){//we give user a warning if we ignore some data
            fprintf(stderr,"# %s: Warning: Read ndat= %d. Reached the limit %d\n",
                prog,ndat,maxdat);
            break; //break the while loop
        }
    }
    bootstrap(ndat,SAMPLES,x,&O,&dO,&chi,&dchi);//calculate...
    printf("# NDAT = %d data. SAMPLES = %d groups\n",ndat,SAMPLES);
    printf("# <o>, chi= (<o^2>-<o>^2)\n");
    printf("# <o> +/- err chi +/- err\n");
    printf("%.15lg %.15lg %.15lg %.15lg\n",O,dO,chi,dchi);
}
/*****
/* bootstrap routine: you can use it in any of your programs...*/
*****/
void bootstrap(int ndat, int samples, double *x,
    double *avO, double *erO, double *avchi, double *erchi){

    int i,j,k,binw,bin;
    double *O,*O2,*chi,xs;
    O = (double *) calloc((size_t) samples ,(size_t) sizeof(double));
    O2 = (double *) calloc((size_t) samples ,(size_t) sizeof(double));
    chi = (double *) calloc((size_t) samples ,(size_t) sizeof(double));
    for(j=0;j<samples;j++){
        for(i=0;i<ndat;i++){
            k = (int)(ndat*drandom());
            O [j] += x[k];
            O2[j] += x[k]*x[k];
        }
    }
}

```



```

    0[j] /=ndat;02[j] /=ndat;
    chi[j] = 02[j] - 0[j]*0[j];
}
/* Compute averages: */
*av0=0.0;*avchi=0.0;
for(j=0;j<samples;j++){
    *av0 += 0[j]; *avchi += chi[j];
}
*av0 /= samples;*avchi /= samples;
/* Compute errors: */
*er0 = 0.0; *erchi = 0.0;
for(j=0;j<samples;j++){
    *er0 += (0 [j]-*av0 )*(0 [j]-*av0 );
    *erchi += (chi[j]-*avchi)*(chi[j]-*avchi);
}
*er0 /= samples ; *erchi /= samples;
*er0 = sqrt(*er0); *erchi = sqrt(*erchi);
/* Compute the real av0 */
*av0 = 0.0;
for(i=0;i<ndat;i++) *av0 += x[i];
*av0 /= ndat;
free(0);free(chi);//release memory back to the system
}
/*****
void locerr( char *errmes ){
    fprintf(stderr,"%s: %s Exiting...\n",prog,errmes);
    exit(1);
}

#define OPTARGS "?hs:d:"
void get_the_options(int argc,char **argv){

    int c,errflg = 0;
    while (!errflg && (c = getopt(argc, argv, OPTARGS)) != -1){
        switch(c){
            case 's':
                SAMPLES = atoi(optarg);
                break;
            case 'd':
                maxdat = atoi(optarg);
                break;
            case 'h':
                errflg++;/*call usage*/
                break;

```

```

        default:
            errflg++;
        }/*switch*/
        if(errflg) usage(argv);
    }/*while...*/
}/*get_the_options()*/

void usage(char **argv){
    fprintf(stderr,"\
Usage: %s [options] <file>\n\
        -s : No. samples Def. 1000\n\
        -d : Give the maximum number of data points read.Def. %d\n\
Computes <o>, chi= (<o^2>-<o>^2)\n\
Data is in one column from stdin.\n",prog,MAXDAT);
    exit(1);
}/*usage()*/
/***** drandom() *****/
#define a 16807
#define m 2147483647
#define q 127773
#define r 2836
#define conv (1.0/(m-1))
#include <sys/types.h>
#include <unistd.h>
long seed;
double drandom(){
    long l;
    l = seed/q;
    seed = a*(seed-q*l) - r*l;
    if(seed < 0) seed +=m;
    return conv*(seed-1);
}
#include <unistd.h>
#include <sys/types.h>
/* Scramble time using the pid number:*/
void init_random(){
    long pid;
    seed = (long) time( (time_t) 0);
    pid = (long) getpid(); //Process ID number
    seed = seed ^ (pid + (pid << 15));
    printf("# Initiated drandom() with seed= %d\n",seed);
}

```

Για τη μεταγλώττιση χρησιμοποιούμε την εντολή

```
% cc -O2 boot.c -o boot -lm
```

Αν τα δεδομένα μας είναι σε μία στήλη στο αρχείο `data` τότε η εντολή για να κάνουμε τον υπολογισμό των σφαλμάτων με 500 samples είναι:

```
% cat data | boot -s 500
```

Το πρόγραμμα δέχεται το πολύ 1,000,000 μετρήσεις (από το `macro` που ορίσαμε `MAXDAT` να είναι το 1000000) όπως και στο πρόγραμμα `jack`. Αν έχουμε παραπάνω πρέπει να χρησιμοποιήσουμε το διακόπτη `-d 2000000` για 2,000,000 μετρήσεις λ.χ. Για παράδειγμα, για την ανάλυση της μαγνήτισης με το πρόγραμμα που έχουμε γράψει για το πρότυπο Ising μπορούμε να χρησιμοποιήσουμε την εντολή:

```
% is -L 20 -b 0.4407 -s 1 -S 342 -n 2000000 | grep -v # | \
  awk -v L=20 '{print ($2>0)?($2/(L*L)):(-$2/(L*L))}' | \
  boot -s 1000 -d 2000000 | grep -v # | \
  awk -v b=0.4407 -v L=20 '{print $1,$2,b*L*L*$3,b*L*L*$4}'
```

Η παραπάνω εντολή είναι μία γραμμή αν αφαιρέσουμε τις τελευταίες ανάποδα κάθετες γραμμές (backslash - '\').

## B'.3 Η ΣΥΓΚΡΙΣΗ ΤΩΝ ΜΕΘΟΔΩΝ

Στην υποενότητα αυτή θα χρησιμοποιήσουμε και θα συγκρίνουμε τις μεθόδους που αναφέραμε στις προηγούμενες ενότητες για τον υπολογισμό των σφαλμάτων. Θα χρησιμοποιήσουμε τη σχέση (4.31), τη μέθοδο `jackknife` (4.36) και τη μέθοδο `bootstrap` (4.40). Για να φανούν οι διαφορές θα χρησιμοποιήσουμε δεδομένα με ισχυρό αυτοσυσχετισμό. Επιλέγουμε στο πρότυπο Ising τον αλγόριθμο Metropolis για  $L = 40$ ,  $\beta = 0.4407 \approx \beta_c$ . Ειδικότερα θα μελετήσουμε τη μαγνήτιση ανά πλεγματική θέση (4.19). Κάνουμε 1,000,000 μετρήσεις με τις εντολές:

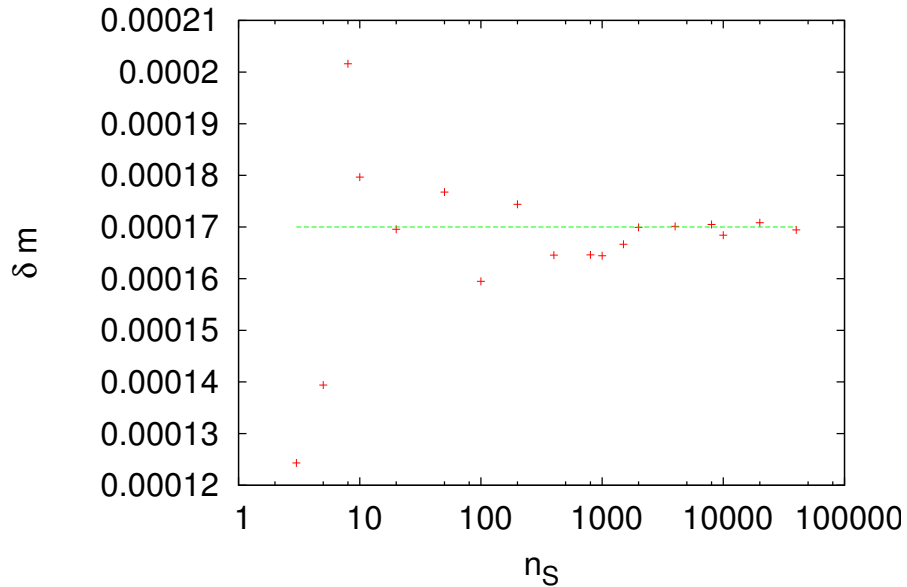
```
% ./is -L 40 -b 0.4407 -s 1 -S 5434365 -n 1000000 > outL40b0.4407.dat &
% grep -v # outL40b0.4407.dat | \
  awk -v L=40 '{if($2<0){$2=-$2};print $2/(L*L)}' > outL40b0.4407.m
% cat outL40b0.4407.m | autoc -t 10000 -n 1000000 > outL40b0.4407.rhom
```

Στις παραπάνω 3 εντολές, η δεύτερη έχει σπάσει σε δύο γραμμές. Το αρχείο `outL40b0.4407.m` έχει σε μία στήλη τις μετρήσεις της μαγνήτισης ενώ το αρχείο `outL40b0.4407.rhom` τη συνάρτηση αυτοσυσχετισμού και τον ολοκληρωμένο χρόνο αυτοσυσχετισμού.

Με το `gnuplot` μελετούμε το χρόνο αυτοσυσχετισμού όπως περιγράψαμε από τη σελίδα 213 και μετά. Παίρνουμε  $\tau_m = 286.3(3)$ . Υπολογίζουμε και τον ολοκληρωμένο χρόνο αυτοσυσχετισμού με αποτέλεσμα  $\tau_{\text{int},m} = 254(1)$ .

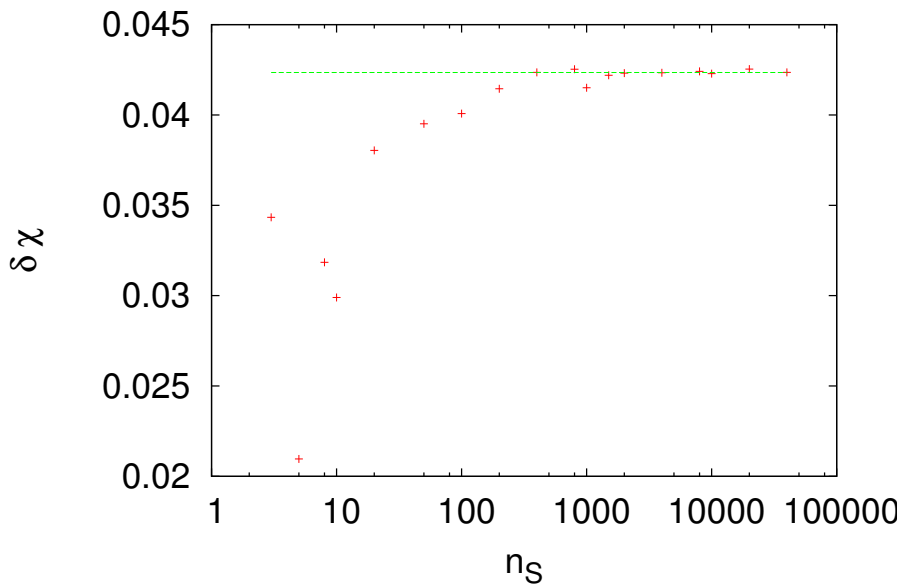
Η μέση τιμή  $\langle m \rangle = 0.638682$ . Η εφαρμογή της σχέσης (4.30) (που προϋποθέτει στατιστικά ανεξάρτητες μετρήσεις) δίνει το (υποεκτιμημένο) σφάλμα στατιστικής απόκλισης  $\delta_c m = 0.00017$ . Χρησιμοποιώντας τη σχέση (4.31) έχουμε  $\delta m = \sqrt{1 + 2\tau\delta_c m} \approx 0.004$ . Τη μαγνητική επιδεκτικότητα δεν είναι δυνατόν να την υπολογίσουμε με τον τρόπο αυτό. Άρα δίνουμε το αποτέλεσμα

$$\langle m \rangle = 0.639 \pm 0.004 \equiv 0.639(4) \quad (\text{B'.1})$$



Σχήμα Β'.1: Το σφάλμα  $\delta m$  της μαγνήτισης υπολογισμένο με τη μέθοδο bootstrap σαν συνάρτηση του αριθμού των δειγμάτων  $n_S$ . Παρατηρούμε πολύ γρήγορη σύγκλιση στην τιμή του σφάλματος που παίρνουμε από τη σχέση (4.30)  $\delta_c m = 0.00017$

Για τον υπολογισμό της μαγνητικής επιδεκτικότητας είναι αναγκαίο να χρησιμοποιήσουμε μία από τις μεθόδους jackknife ή bootstrap. Η δεύτερη εφαρμόζεται αρχικά με μεταβλητό αριθμό δειγμάτων  $n_S$  έτσι ώστε να προσδιορίσουμε τον βέλτιστο αριθμό από αυτά για να χρησιμοποιήσουμε στην ανάλυσή μας. Στο σχήμα Β'.1 φαίνονται τα αποτελέσματα για τη μαγνήτιση. Παρατηρούμε πολύ γρήγορη σύγκλιση στο σφάλμα της στατιστικής απόκλισης  $\delta_c m =$



Σχήμα B'.2: Το σφάλμα  $\delta\chi$  της μαγνητικής επιδεκτικότητας υπολογισμένο με τη μέθοδο bootstrap σαν συνάρτηση του αριθμού των δειγμάτων  $n_S$ . Παρατηρούμε σύγκλιση για  $n_S > 1000$  στην τιμή  $\delta_c\chi = 0.0435$ .

0.00017 για μικρό αριθμό δειγμάτων. Θα μπορούσαμε με ασφάλεια για την ανάλυση των δεδομένων της μαγνήτισης να επιλέγαμε  $n_S = 100$ . Για την περίπτωση της μαγνητικής επιδεκτικότητας η σύγκλιση είναι πιο αργή, αλλά πάλι θα μπορούσαμε να πάρουμε  $n_S = 500$ . Παίρνουμε  $\chi = 20.39$  και για το σφάλμα  $\delta_c\chi = 0.0435$ . Η τιμή του σφάλματος όμως αφορά ανεξάρτητες μετρήσεις, κάτι που δε συμβαίνει στην περίπτωση μας. Η τιμή αυτή θα πρέπει να διορθωθεί από τον παράγοντα  $\sqrt{1 + 2\tau_m}$  και να δώσει  $\delta\chi = 1$ . Άρα

$$\chi = 20 \pm 1 \equiv 20(1) \quad (\text{B'.2})$$

Αξίζει να παρατηρηθεί ότι το σφάλμα είναι αρκετά μεγάλο, κάτι που δείχνει την πρακτική δυσκολία που φέρνουν οι μεγάλοι χρόνοι αυτοσυσχετισμού (στην πραγματικότητα έχουμε μόνο  $n/(2\tau_m) \approx 1,000,000/(2 \times 286) \approx 1750$  ανεξάρτητες μετρήσεις). Επίσης σημειώνουμε ότι για τον υπολογισμό του σφάλματος με τον τρόπο αυτό είναι αναγκαία η γνώση του  $\tau_m$ .

Πιο ασφαλής, γρήγορη και ευσταθής μέθοδος είναι η μέθοδος jackknife. Εδώ δε θα χρειαστεί η εκ των προτέρων γνώση του  $\tau_m$ . Για τον προσδιορισμό του σφάλματος αρκεί να μελετήσουμε τα δεδομένα με μεταβλητό αριθμό από jackknife bins  $n_b$ . Στο σχήμα

Β'.3 φαίνονται τα αποτελέσματά μας για τη μαγνήτιση. Όταν τα jackknife bins  $n_b = n$  τότε τα δείγματά μας αποτελούνται από όλες τις μετρήσεις εκτός από ένα στοιχείο. Τότε το σφάλμα θα είναι το ίδιο με το σφάλμα της στατιστικής απόκλισης και θα είναι υποεκτιμημένο κατά το γνωστό παράγοντα  $\sqrt{1+2\tau_m}$ . Αυτό φαίνεται στο σχήμα Β'.3 όπου παρατηρείται μια πολύ αργή σύγκλιση στην τιμή  $\delta_c m = 0.00017$ . Η επίδραση των αυτοσυσχετισμών εξαφανίζεται καθώς από το δείγμα μας αφαιρούμε (bin width)  $\approx 2\tau_m$  μετρήσεις. Αυτό γίνεται όταν  $n_b \approx n/(\text{bin width}) = n/(2\tau_m) = 1,000,000/572 \approx 1750$ . Φυσικά αυτή η εκτίμηση δίνει την τάξη μεγέθους και μια προσεκτική μελέτη είναι αναγκαία για τον προσδιορισμό του σωστού αριθμού  $n_b$ . Στο σχήμα Β'.3 βλέπουμε ότι το σφάλμα συγκλίνει για  $100 < n_b < 800$  στην τιμή  $\delta m = 0.0036$  που είναι αρκετά κοντά στην τιμή  $\sqrt{1+2\tau_m}\delta_c m \approx 0.004$  (και ακόμα κοντότερα αν χρησιμοποιήσουμε τον ολοκληρωμένο χρόνο αυτοσυσχετισμού  $\tau_{\text{int},m} = 254$ ). Επίσης παρατηρούμε ότι χοντρική εκτίμηση του σφάλματος μπορούμε να πάρουμε για πολύ μικρό αριθμό  $n_b \approx 20-40$  κάτι που μπορεί να χρησιμοποιηθεί για γρήγορους υπολογισμούς.

Παρόμοια αποτελέσματα παίρνουμε και για την μαγνητική επιδεκτικότητα  $\chi$  όπου το σφάλμα συγκλίνει στην τιμή  $\delta\chi = 0.86$  σε συμφωνία με τις προηγούμενες εκτιμήσεις μας. Για  $n_b \rightarrow n$  το σφάλμα συγκλίνει στο (υποεκτιμημένο) σφάλμα  $\delta_c\chi = 0.0421$ .

Γεννιέται το ερώτημα αν ή μέθοδος bootstrap μπορεί να χρησιμοποιηθεί με ανάλογο τρόπο με αυτό της jackknife για τον προσδιορισμό του πραγματικού σφάλματος  $\delta m, \delta\chi$  χωρίς τη γνώση του  $\tau_m$ . Η απάντηση είναι ναι. Θα χρησιμοποιήσουμε τη διαίσθησή μας από τη μέθοδο jackknife και binning. Χωρίζουμε τα δεδομένα μας σε  $n_b$  bins των οποίων το εύρος (bin width)  $= n/n_b$ . Κάθε bin θεωρείται ως μια ανεξάρτητη μέτρηση που είναι η μέση τιμή της παρατηρήσιμης ποσότητας μέσα στο bin αυτό. Παίρνουμε  $n_S = 1000$  και εφαρμόζουμε τη μέθοδο bootstrap στο δείγμα των  $n_b$  μετρήσεων. Στα σχήματα Β'.5 και Β'.6 δείχνουμε τα αποτελέσματά μας σαν συνάρτηση του (bin width). Όταν αυτό γίνει (bin width)  $\approx 2\tau_m$  οι μετρήσεις μας θα είναι πρακτικά ανεξάρτητες και θα πρέπει να πάρουμε το πραγματικό σφάλμα. Στο σχήμα Β'.5 δείχνουμε τα αποτελέσματα για τη μαγνήτιση όπου για bin width=1 έχουμε το αποτέλεσμα  $\delta_c m = 0.00017$ . Όταν το εύρος γίνει  $> 2\tau_m$  το σφάλμα γίνεται  $\delta m = 0.0036$ , το οποίο το διαβάζουμε στο plateau για  $1100 < (\text{bin width}) < 16000$ , σε συμφωνία με τη μέθοδο jackknife. Στο σχήμα Β'.5 δείχνουμε τα αποτελέσματα για τη μαγνητική επιδεκτικότητα. Πάλι για bin width=1 έχουμε το αποτέλεσμα  $\delta_c\chi = 0.0421$ . Όταν το εύρος γίνει  $> 2\tau_m$  το σφάλμα γίνεται  $\delta\chi = 0.615$ , το οποίο το διαβάζουμε στο plateau για  $500 < (\text{bin width}) < 1000$  που δεν

είναι πολύ διαφορετικό από το αποτέλεσμα που πήραμε με τη μέθοδο jackknife.

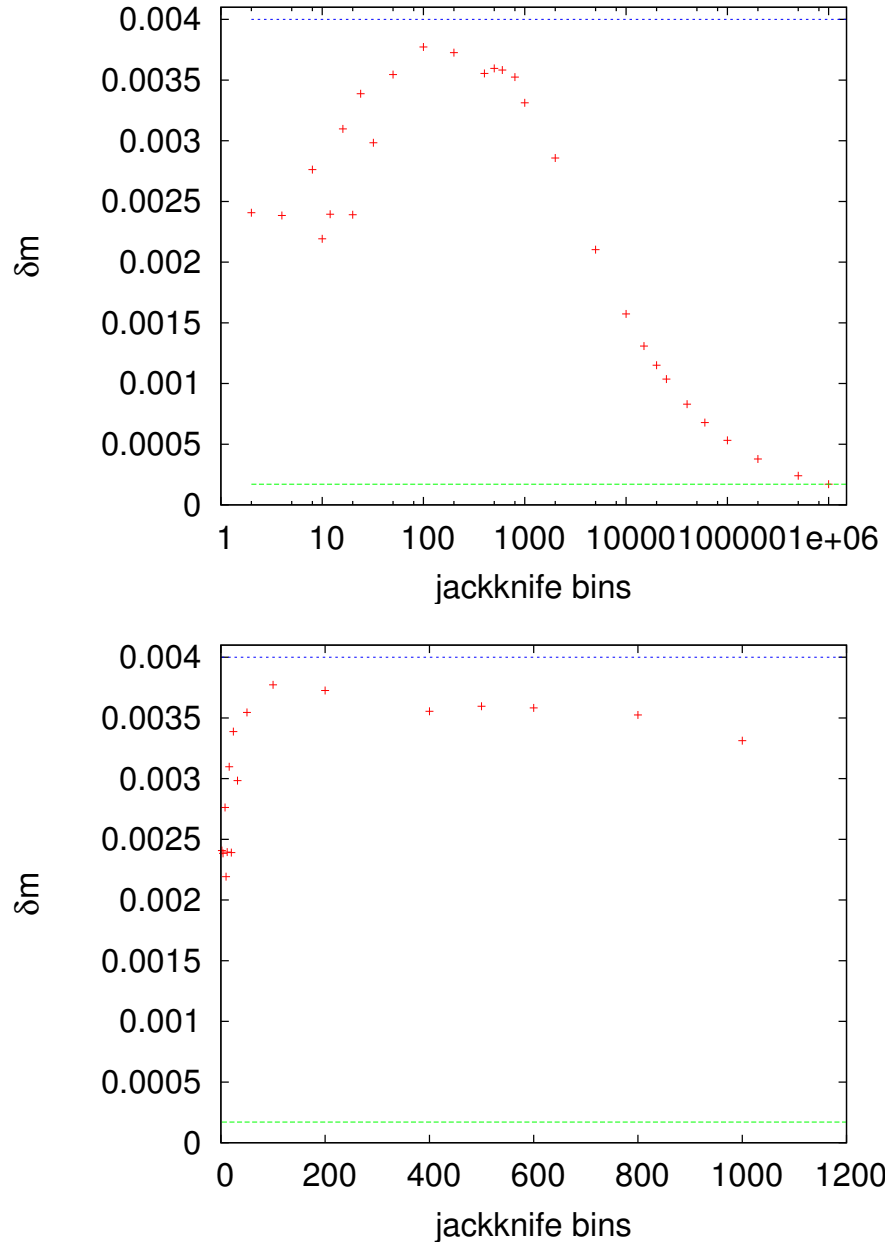
Κλείνουμε την υποενότητα αυτή με μια σημαντική παρατήρηση. Στην παρουσίασή μας υπολογίσαμε πρώτα το χρόνο αυτοσυσχετισμού  $\tau_m$  από τη συνάρτηση αυτοσυσχετισμού (σχήμα A'.1) και στη συνέχεια προσδιορίσαμε τότε οι μέθοδοι που χρησιμοποιήσαμε δώσαν το (υποεκτιμημένο) σφάλμα στατιστικής απόκλισης των μετρήσεων και τότε το πραγματικό που λαμβάνει υπόψη τους αυτοσυσχετισμούς. Στην πράξη όμως οι μέθοδοι αυτοί αποτελούν μία ανεξάρτητη μέθοδο υπολογισμού του  $\tau_m$  από τις σχέσεις  $\delta O / \delta_c O = \sqrt{1 + 2\tau_c}$ . Έτσι από τα παραπάνω συνάγεται ότι

$$\tau_m = \frac{1}{2} \left( \left( \frac{\delta m}{\delta_c m} \right)^2 - 1 \right) = \frac{1}{2} \left( \left( \frac{\delta \chi}{\delta_c \chi} \right)^2 - 1 \right) \dots \quad (B'.3)$$

σχέσεις που μπορούν να χρησιμοποιηθούν για τον έλεγχο της αυτοσυνέπειας της ανάλυσης των δεδομένων μας. Φυσικά θα υπενθυμίσουμε ότι ένα σύστημα έχει πολλούς χρόνους αυτοσυσχετισμού και διαφορετικές παρατηρήσιμες ποσότητες μπορεί η συμπεριφορά τους να καθορίζεται από διαφορετικούς χρόνους αυτοσυσχετισμού (fast modes, slow modes).

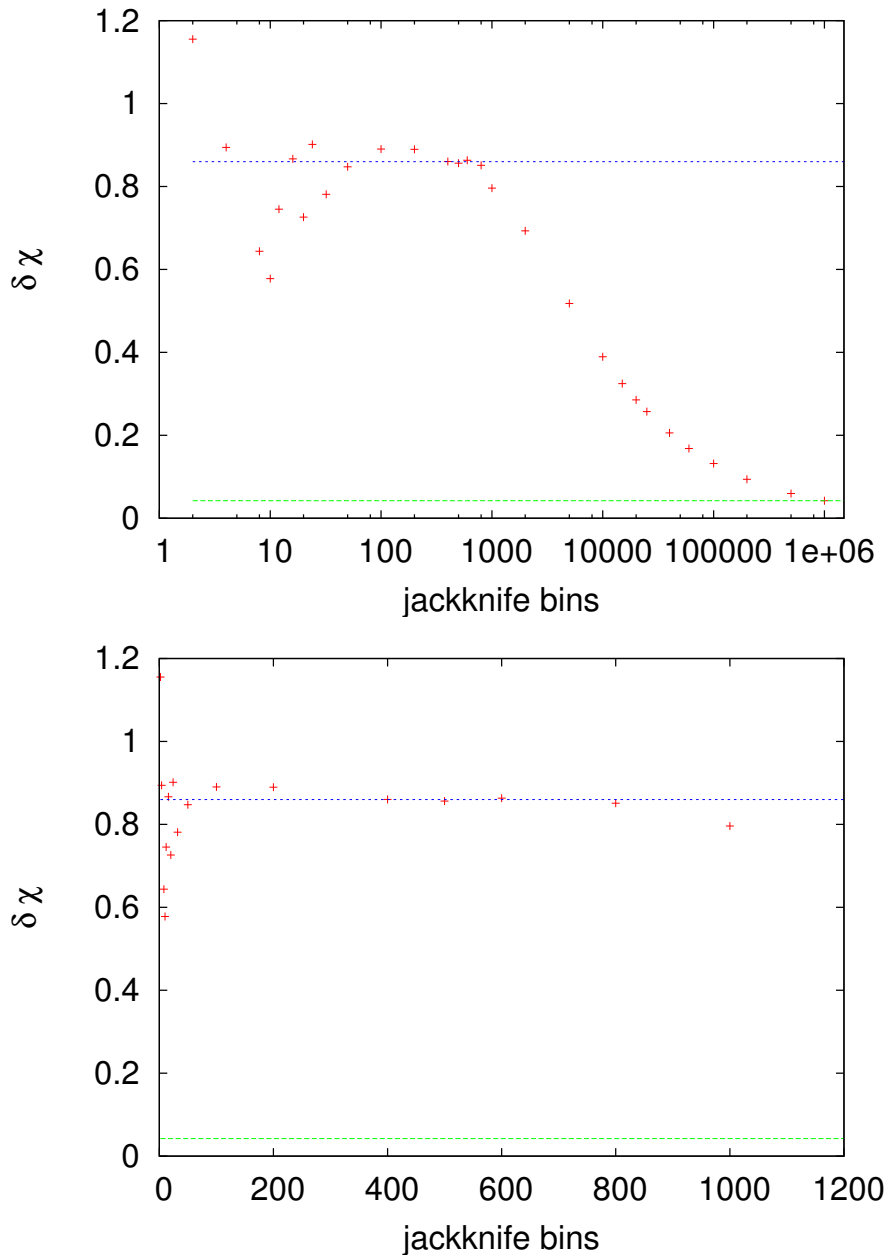
## B'.4 ΠΡΟΒΛΗΜΑΤΑ

1. Μετατρέψτε το πρόγραμμα `boot.c` έτσι ώστε να κάνει binning στα δεδομένα. Αναπαράγετε τα σχήματα B'.5 και B'.6. (Λύση: Το πρόγραμμα `boot_bin.c` στον υποκατάλογο `Tools/` του συνοδευτικού λογισμικού)

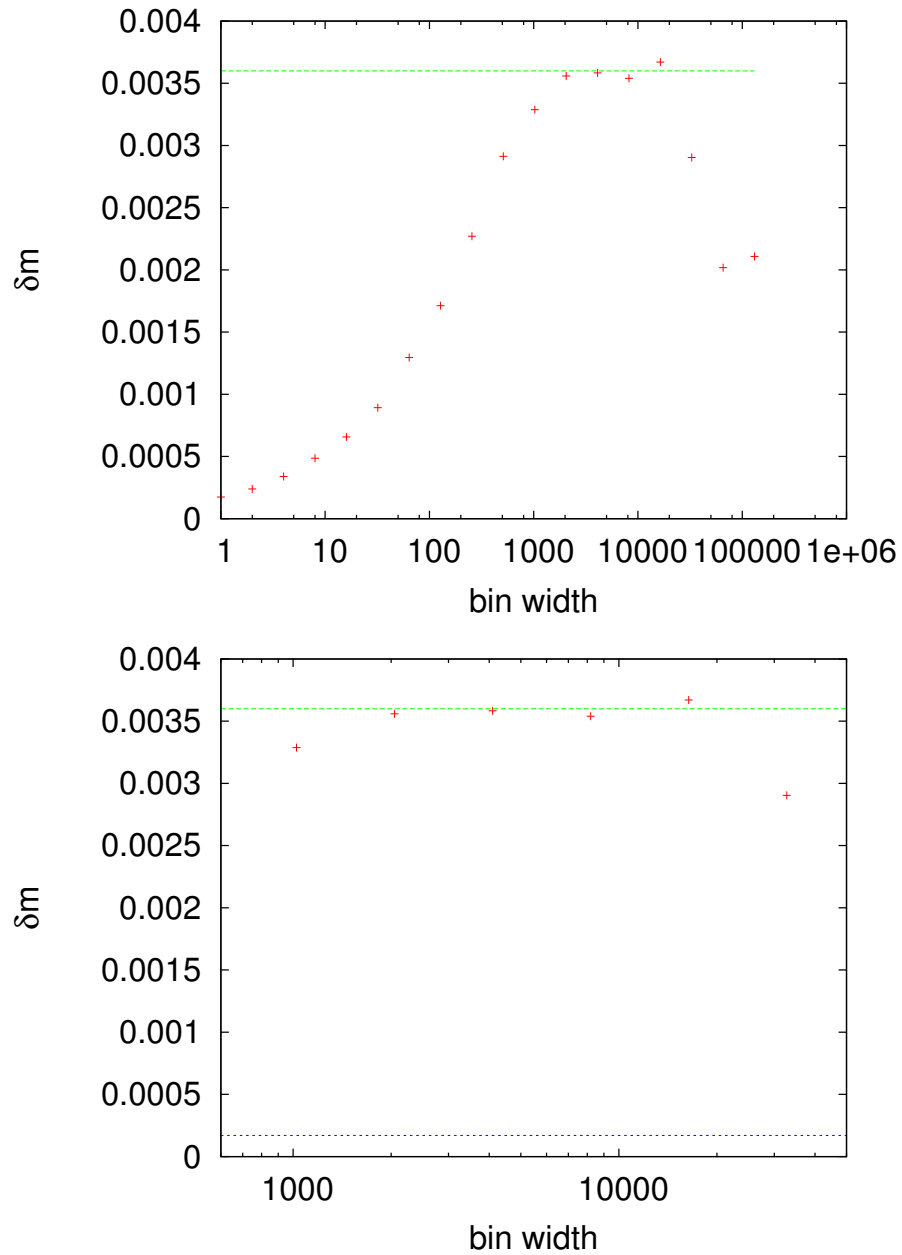


Σχήμα Β'.3: Το σφάλμα  $\delta m$  της μαγνήτισης υπολογισμένο με τη μέθοδο jackknife σαν συνάρτηση του αριθμού των δειγμάτων jackknife bins  $n_b$ . Παρατηρούμε σύγκλιση για  $100 < n_b < 800$  στην τιμή  $\delta m = 0.0036$ . Στο πάνω σχήμα φαίνεται καθαρά ότι καθώς φτάνουμε στο όριο να αφαιρούμε μόνο ένα στοιχείο από τα δεδομένα ( $n_b = n$ ) το σφάλμα πλησιάζει στην τιμή που υπολογίζεται από τη σχέση (4.30)  $\delta_c m = 0.00017$ . Οι οριζόντιες γραμμές αντιστοιχούν στις τιμές  $\delta_c m$  και  $\sqrt{1 + 2\tau_m}\delta_c m \approx 0.004$  όπου  $\tau_m = 286.3$ . Ο λόγος  $\delta m/\delta_c m \approx \sqrt{1 + 2\tau_m}$ .

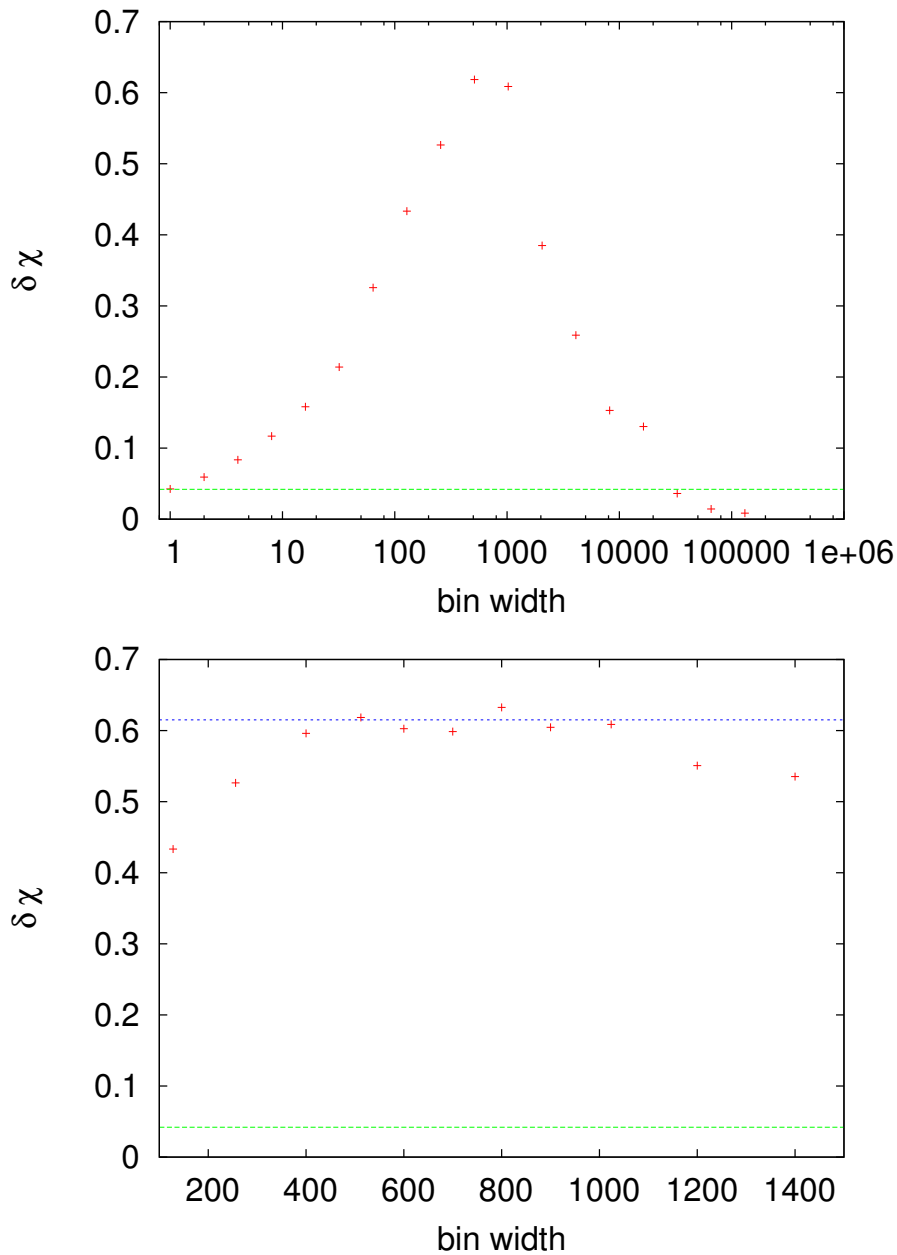




Σχήμα B'.4: Το σφάλμα  $\delta\chi$  της μαγνήτισης υπολογισμένο με τη μέθοδο jackknife σαν συνάρτηση του αριθμού των δειγμάτων jackknife bins  $n_b$ . Παρατηρούμε σύγκλιση για  $100 < n_b < 800$  στην τιμή  $\delta\chi = 0.86$ . Στο πάνω σχήμα φαίνεται καθαρά ότι καθώς φτάνουμε στο όριο να αφαιρούμε μόνο ένα στοιχείο από τα δεδομένα ( $n_b = n$ ) το σφάλμα πλησιάζει στην τιμή που θα παίρναμε αν θεωρούσαμε τις μετρήσεις ανεξάρτητες  $\delta_{c\chi} = 0.0421$ . Η τιμή αυτή είναι πολύ κοντά στην τιμή που πήραμε με τη μέθοδο bootstrap. Οι τιμές αυτές φαίνονται στα σχήματα στις δύο οριζόντιες γραμμές. Ο λόγος  $\delta\chi/\delta_{c\chi} \approx \sqrt{1 + 2\tau_m}$ .



Σχήμα Β'.5: Το σφάλμα  $\delta m$  της μαγνήτισης υπολογισμένο με τη μέθοδο bootstrap. Παίρνουμε  $n_S = 1000$  και χωρίζουμε τα δεδομένα μας σε bins των οποίων το εύρος bin width φαίνεται στον οριζόντιο άξονα. Κάθε bin θεωρείται ως μια ανεξάρτητη μέτρηση. Για bin width=1 έχουμε το αποτέλεσμα  $\delta_{cm} = 0.00017$ . Όταν το εύρος γίνει  $> 2\tau_m$  το σφάλμα γίνεται  $\delta m = 0.0036$ , το οποίο το διαβάζουμε στο plateau για  $1100 < \text{bin width} < 16000$ , σε συμφωνία με τη μέθοδο jackknife. Αυτό φαίνεται καθαρά στο κάτω σχήμα. Οι οριζόντιες γραμμές αντιστοιχούν στις τιμές των  $\delta_{cm}$  και  $\delta m$  που αναφέρθηκαν παραπάνω.



Σχήμα Β'.6: Το σφάλμα  $\delta\chi$  της μαγνητικής επιδεικτικότητας υπολογισμένο με τη μέθοδο bootstrap. Παίρνουμε  $n_S = 1000$  και χωρίζουμε τα δεδομένα μας σε bins των οποίων το εύρος bin width φαίνεται στον οριζόντιο άξονα. Κάθε bin θεωρείται ως μια ανεξάρτητη μέτρηση. Για bin width=1 έχουμε το αποτέλεσμα  $\delta_{c\chi} = 0.0421$ . Όταν το εύρος γίνει  $> 2\tau_m$  το σφάλμα γίνεται  $\delta\chi = 0.615$ , το οποίο το διαβάζουμε στο plateau για  $500 < \text{bin width} < 1000$  που δεν είναι πολύ διαφορετικό από το αποτέλεσμα που πήραμε με τη μέθοδο jackknife. Αυτό φαίνεται καθαρά στο κάτω σχήμα. Οι οριζόντιες γραμμές αντιστοιχούν στις τιμές των  $\delta_{c\chi}$  και  $\delta\chi$  που αναφέρθηκαν παραπάνω.



## ΠΑΡΑΡΤΗΜΑ Γ'

# ΒΑΘΜΙΣΗ ΚΑΙ ΚΡΙΣΙΜΟΙ ΕΚΘΕΤΕΣ

### Γ'.1 BINDER CUMULANT

Στο κεφάλαιο 6 μελετήσαμε αριθμητικά τις ιδιότητες βάθμισης της ποσότητας

$$U = 1 - \frac{\langle m^4 \rangle}{3 \langle m^2 \rangle^2}, \quad (\Gamma'.1)$$

γνωστής στη βιβλιογραφία ως Binder Cumulant. Εδώ χρησιμοποιώντας βασικές ιδιότητες βάθμισης ενός συστήματος κοντά στην κρίσιμη θερμοκρασία θα υπολογίσουμε τις ιδιότητες βάθμισης της ποσότητας αυτής και των παραγώγων της. Για περισσότερες λεπτομέρειες παραπέμπουμε τον αναγνώστη στα [33], [3]

Οι τιμές του  $U$  είναι απλές και τετριμμένες σε δύο περιπτώσεις: Όταν έχουμε γκαουσιανή κατανομή της μαγνήτισης, κάτι το οποίο συμβαίνει όταν είμαστε στη φάση αταξίας (ψηλή θερμοκρασία)  $U = 0$  και όταν είμαστε στη φάση τάξης (χαμηλή θερμοκρασία)  $U = 2/3$ . Η απόδειξη είναι πολύ εύκολη και τη σκιαγραφούμε στις ασκήσεις 1 και 2.

Όταν πλησιάζουμε την κρίσιμη θερμοκρασία μιας συνεχούς μετάβασης φάσης, το σύστημα παρουσιάζει ιδιότητες βάθμισης λόγω του απειριζόμενου μήκους συσχετισμού  $\xi$ , όπως περιγράψαμε στο κεφάλαιο 6. Ας υποθέσουμε ότι πλησιάζουμε από τις ψηλές θερμοκρασίες και η συνάρτηση κατανομής της μαγνήτισης ανά πλεγματική θέση  $s$  (όχι τις απόλυτης τιμής) είναι της μορφής

$$P(L, s) = L^{d/2} \left( \frac{\beta}{2\pi\chi} \right) e^{-s^2 \frac{L^2\beta}{2\chi}} \quad (\Gamma'.2)$$

πού είναι μια γκαουσιανή με τυπική απόκλιση  $\sigma = \chi/(\beta L^d)$ ,  $\chi =$

$\beta L^d \langle s^2 \rangle$  και προσωρινά έχουμε υποθέσει ότι το σύστημα είναι πάνω σε  $d$ -διάστατο υπερκυβικό πλέγμα με ακμή υπερκύβου  $L$ .

Όταν πλησιάζουμε την κρίσιμη θερμοκρασία, η βάθμιση παρουσιάζεται στη συνάρτηση κατανομής  $P(L, s)$  σύμφωνα με τη σχέση

$$P(L, s) = L^x p_0 \tilde{P}(aL^y s, \frac{L}{\xi}). \quad (\Gamma'.3)$$

Για να υπολογίσουμε τους εκθέτες στην παραπάνω σχέση, εφαρμόζουμε κατ' αρχήν τη συνθήκη κανονικοποίησης της πιθανότητας

$$\begin{aligned} 1 &= \int_{\infty}^{+\infty} ds P(L, s) \\ &= L^x p_0 \int_{\infty}^{+\infty} ds \tilde{P}(aL^y s, \frac{L}{\xi}) \\ &= L^x p_0 \frac{1}{aL^y} \int_{\infty}^{+\infty} dz \tilde{P}(z, \frac{L}{\xi}) \\ &= L^{x-y} p_0 \frac{1}{a} \int_{\infty}^{+\infty} dz \tilde{P}(z, \frac{L}{\xi}), \end{aligned} \quad (\Gamma'.4)$$

όπου θέσαμε  $z = aL^y s$ . Για να είναι μονάδα το αριστερό μέλος πρέπει  $x = y$ ,

$$C_0 \equiv \int_{\infty}^{+\infty} dz \tilde{P}(z, \frac{L}{\xi}) < \infty \quad (\Gamma'.5)$$

και  $p_0 = a/C_0$ . Τελικά παίρνουμε

$$P(L, s) = \frac{a}{C_0} L^y \tilde{P}(aL^y s, \frac{L}{\xi}). \quad (\Gamma'.6)$$

Οι ροπές της κατανομής των σπιν  $\langle s^k \rangle$  θα είναι τότε

$$\begin{aligned} \langle s^k \rangle &= \int_{\infty}^{+\infty} ds s^k P(L, s) \\ &= \frac{a}{C_0} L^y \int_{\infty}^{+\infty} ds s^k \tilde{P}(aL^y s, \frac{L}{\xi}) \\ &= \frac{a}{C_0} L^y \frac{1}{a^{k+1} L^{(k+1)y}} \int_{\infty}^{+\infty} dz z^k \tilde{P}(z, \frac{L}{\xi}) \\ &= \frac{1}{C_0 a^k L^{ky}} F_k \left( \frac{L}{\xi} \right). \end{aligned} \quad (\Gamma'.7)$$

Όταν πάρουμε το θερμοδυναμικό όριο και μετά πλησιάσουμε την κρίσιμη θερμοκρασία, τότε έχουμε αποκλίνον μήκος συσχετισμού  $\xi \rightarrow \infty$  με  $L/\xi \rightarrow \infty$ . Στην περιοχή  $\beta < \beta_c$  ( $\langle m \rangle = 0$ ) έχουμε  $\chi = \beta L^d \langle s^2 \rangle$  και από τη σχέση

$$\left. \begin{aligned} \chi &= \chi_+ t^{-\gamma} \\ \xi &= \xi_+ t^{-\nu} \end{aligned} \right\} \implies \chi = \frac{\chi_+}{\xi_+^{\gamma/\nu}} \xi^{\gamma/\nu} \sim \xi^{\gamma/\nu} \quad (\Gamma'.8)$$

παίρνουμε

$$\langle s^2 \rangle = \beta^{-1} L^{-d} \chi = \frac{\chi^+}{\beta L^d \xi_+^{\gamma/\nu}} \xi^{\gamma/\nu} \sim \xi^{\gamma/\nu}. \quad (\Gamma'.9)$$

Στις παραπάνω σχέσεις εισάγαμε τα “παγκόσμια” πλάτη (universal amplitudes)  $\chi_+$  και  $\xi_+$  τα οποία είναι παγκόσμιες σταθερές (ίδιες στην ίδια κλάση παγκοσμιότητας) και ορίζονται από τη σχέση (Γ'.8). Στο όριο αυτό, για έχει συνεπή βάθμιση η σχέση (Γ'.7) για  $k = 2$  στο δεξί και αριστερό μέλος, παίρνουμε (πρβλ. εξ. (6.29))

$$F_k \left( \frac{L}{\xi} \right) \sim \left( \frac{L}{\xi} \right)^{-\gamma/\nu} \quad \text{για} \quad \frac{L}{\xi} \gg 1. \quad (\Gamma'.10)$$

Για να βρούμε τη βάθμιση ως προς  $L$  αντικαθιστούμε τις παραπάνω σχέσεις στην (Γ'.7) για  $k = 2$  παίρνουμε

$$\frac{\chi^+}{\beta L^d \xi_+^{\gamma/\nu}} \xi^{\gamma/\nu} \sim \frac{1}{C_0 a^2 L^{2y}} \left( \frac{L}{\xi} \right)^{\gamma/\nu} \quad (\Gamma'.11)$$

από την οποία παίρνουμε

$$L^{-d} \sim L^{-2y} L^{-\gamma/\nu} \implies d = 2y + \frac{\gamma}{\nu} \implies y = \frac{d\nu - \gamma}{2\nu} = \frac{\beta}{\nu}, \quad (\Gamma'.12)$$

όπου χρησιμοποιήσαμε τη γνωστή<sup>1</sup> σχέση υπερβάθμισης (hyperscaling relation)

$$d\nu = \gamma + 2\beta. \quad (\Gamma'.13)$$

Καταλήγουμε στις σχέσεις που ισχύουν στην περιοχή αταξίας  $\beta < \beta_c$

$$P(L, s) = \frac{a}{C_0} L^{\beta/\nu} \tilde{P}(aL^{\beta/\nu} s, \frac{L}{\xi}) \quad (\Gamma'.14)$$

$$\langle s^2 \rangle = \frac{L^{-2\beta/\nu}}{C_0 a^2} F_2 \left( \frac{L}{\xi} \right) \quad (\Gamma'.15)$$

$$\langle s^4 \rangle = \frac{L^{-4\beta/\nu}}{C_0 a^4} F_4 \left( \frac{L}{\xi} \right) \quad (\Gamma'.16)$$

Από τη σχέση (Γ'.1) βρίσκουμε ότι η κρίσιμη συμπεριφορά του Binder cumulant είναι

$$U \sim 1 - \frac{\frac{L^{-4\beta/\nu}}{C_0 a^4} F_4 \left( \frac{L}{\xi} \right)}{3 \frac{L^{-4\beta/\nu}}{C_0^2 a^4} F_2 \left( \frac{L}{\xi} \right)^2} = 1 - \frac{C_0^2}{3} \frac{F_4 \left( \frac{L}{\xi} \right)}{F_2 \left( \frac{L}{\xi} \right)^2} \quad (\Gamma'.17)$$

<sup>1</sup>Δείτε λ.χ. [34], σχέσεις 3.35, 3.36, 3.53.

Στην ψευδοκρίσιμη περιοχή παίρνουμε το όριο  $L \rightarrow \infty$  με  $\xi \sim L$ . Οι σχέση (6.9) δίνει  $\langle s^k \rangle \sim L^{-k\beta/\nu}$ , οπότε για να είναι συνεπείς οι σχέσεις (Γ'.15), (Γ'.16) στο όριο αυτό θα πρέπει

$$F_k \left( \frac{L}{\xi} \right) \sim F_k(\infty) = \text{σταθ.} < \infty, \quad (\text{Γ'.18})$$

οπότε παίρνουμε

$$U \sim U^* = 1 - \frac{C_0^2 F_4(\infty)}{3 F_2(\infty)^2} = \text{σταθ.} \quad (\text{Γ'.19})$$

Άρα στην ψευδοκρίσιμη περιοχή, ο Binder cumulant παίρνει σχεδόν σταθερή τιμή  $U^*$  για όλες τις τιμές του  $L$ . Η τιμή αυτή έχει βρεθεί ότι εξαρτάται από τις συνοριακές συνθήκες και από την ανισοτροπικότητα των αλληλεπιδράσεων. Για τετραγωνικό πλέγμα στο πρότυπο Ising έχουμε [33] (Kamieniarz+Blöte)

$$U^* = 0.610690(1) \quad (\text{Γ'.20})$$

## Γ'.2 ΒΑΘΜΙΣΗ

Η βασική υπόθεση για τη βάρθρωση των θερμοδυναμικών ποσοτήτων στην περιοχή μιας συνεχούς μετάβασης φάσης κάτω από μια αλλαγή κλίμακας μήκους στο πλέγμα όπου το αδιάστατο μήκος συσχετισμού αλλάζει ως (δες λ.χ. κεφ. 3 στο [34])

$$\xi \rightarrow \frac{\xi}{b}. \quad (\text{Γ'.21})$$

όπου  $b$  είναι ο παράγοντας αλλαγής της κλίμακας μήκους, είναι ότι η ελεύθερη ενέργεια του συστήματος αλλάζει ως<sup>2</sup>

$$f(t, h) = b^{-d} f(tb^{yt}, hb^{yh}), \quad (\text{Γ'.22})$$

όπου  $t$  η ανηγμένη θερμοκρασία και  $h$  το εξωτερικό μαγνητικό πεδίο. Η παραπάνω σχέση συμπυκνώνει την υπόθεση βάρθρωσης (scaling hypothesis), και είναι μια σχέση ανάλογη με αυτή της (Γ'.3). Η σχέση αυτή μπορεί να γίνει κατανοητή μέσω της προσέγγισης της ομάδας επανακανονικοποίησης και η βασική υπόθεση είναι η εμφάνιση μιας μοναδικής δυναμικής κλίμακας μήκους που απειρίζεται καθώς πλησιάζουμε το κρίσιμο σημείο και από την οποία εξαρτώνται όλες οι φυσικές ποσότητες. Τα ορίσματα  $tb^{yt}$  και  $hb^{yh}$  δίνουν την αλλαγή που πρέπει να γίνει στις “σταθερές σύζευξης”  $t$  και  $h$  κάτω από την αλλαγή κλίμακας μήκους ώστε να ισχύει η ισότητα.

<sup>2</sup>Για την ακρίβεια το *ιδιάζον* (singular) κομμάτι της ελεύθερης ενέργειας



Εφαρμόζοντας την παραπάνω σχέση  $n$  φορές παίρνουμε

$$f(t, h) = b^{-nd} f(tb^{ny_t}, hb^{ny_h}). \quad (\Gamma'.23)$$

Αν την εφαρμόσουμε ένα μεγάλο αριθμό από φορές έτσι ώστε  $n \rightarrow \infty, t \rightarrow 0$  κρατώντας το γινόμενο  $tb^{ny_t} = t_0 = \mathcal{O}(1)$ , τότε αντικαθιστώντας τη σχέση που προκύπτει  $b^n \sim t^{-1/y_t}$  παίρνουμε

$$\begin{aligned} f(t, h) &= t^{d/y_t} f(t_0, ht^{-y_h/y_t}) \\ &\equiv t^{d/y_t} \Psi(ht^{-y_h/y_t}) \\ &= t^{2-\alpha} \Psi(ht^{-y_h/y_t}), \end{aligned} \quad (\Gamma'.24)$$

όπου στη δεύτερη γραμμή ορίσαμε τη συνάρτηση βάθμισης  $\Psi(z)$  και στην τρίτη ορίσαμε τον κρίσιμο εκθέτη

$$\alpha = 2 - \frac{d}{y_t}. \quad (\Gamma'.25)$$

Εφαρμόζοντας την παραπάνω υπόθεση στη σχέση (Γ'.21) για το μήκος συσχετισμού

$$\xi(t, h) = b^{-1} \xi(tb^{y_t}, hb^{y_h}) = \dots = b^{-n} \xi(tb^{ny_t}, hb^{ny_h}), \quad (\Gamma'.26)$$

και παίρνοντας το όριο  $n \rightarrow \infty, t \rightarrow 0$  κρατώντας το γινόμενο  $tb^{ny_t} \sim \mathcal{O}(1)$ , το αριστερό μέλος θα έχει μια πεπερασμένη τιμή, έστω  $\xi_0 < \infty$  ενώ το δεξί θα μας δώσει

$$\xi_0 = t^{1/y_t} \xi(t_0, hb^{-y_h/y_t}). \quad (\Gamma'.27)$$

Θεωρώντας την περίπτωση  $h = 0$  και συγκρίνοντας με τη γνωστή σχέση  $\xi \sim t^{-\nu}$  (εξίσωση (6.4)) που όρισε τον κρίσιμο εκθέτη  $\nu$  παίρνουμε

$$\xi = \xi_0 t^{-1/y_t} \implies \nu = \frac{1}{y_t}. \quad (\Gamma'.28)$$

Παραγωγίζοντας τη σχέση (Γ'.24) ως προς τη θερμοκρασία παίρνουμε

$$\frac{\partial f}{\partial t} \sim t^{1-\alpha} \Psi(ht^{-y_h/y_t}) + t^{2-\alpha} ht^{-y_h/y_t-1} \Psi'(ht^{-y_h/y_t}). \quad (\Gamma'.29)$$

Στην παραπάνω σχέση, και σε όλες που ακολουθούν που έχουν το σύμβολο  $\sim$  θα αμελούμε σταθερές που είναι άσχετες με τις ιδιότητες βάθμισης.

Παραγωγίζοντας την παραπάνω σχέση άλλη μια φορά, και θέτοντας  $h = 0$  παίρνουμε την ειδική θερμότητα

$$c \sim \frac{\partial^2 f}{\partial t^2} \sim t^{-\alpha} \Psi(0). \quad (\Gamma'.30)$$

Άρα ο κρίσιμος εκθέτης  $\alpha$  δεν είναι άλλος από τον κρίσιμο εκθέτη της ειδικής θερμότητας που ορίσαμε στην σχέση (6.4).

Η μαγνητική επιδεκτικότητα παίρνεται με ανάλογο τρόπο παραγωγίζοντας την (Γ'.24) ως προς  $h$

$$\frac{\partial f}{\partial h} \sim t^{d/y_t} t^{-y_h/y_t} \Psi'(ht^{-y_h/y_t}) \sim t^{\nu d - \nu y_h} \Psi'(ht^{-\nu y_h}) \quad (\Gamma'.31)$$

Παραγωγίζοντας την παραπάνω σχέση άλλη μια φορά, και θέτοντας  $h = 0$  παίρνουμε την μαγνητική επιδεκτικότητα

$$\chi \sim \frac{\partial^2 f}{\partial h^2} \sim t^{\nu d - 2\nu y_h} \Psi'(0), \quad (\Gamma'.32)$$

και συγκρίνοντας με τη σχέση (6.3)  $\chi \sim t^{-\gamma}$  παίρνουμε

$$\gamma = 2\nu y_h - \nu d \Leftrightarrow y_h = \frac{1}{2}(d + \frac{\gamma}{\nu}) = d - \frac{\beta}{\nu} = \frac{\beta + \gamma}{\nu} \quad (\Gamma'.33)$$

όπου στις τελευταίες δύο σχέσεις χρησιμοποιήσαμε τις “σχέσεις υπερβάθμισης” (hyperscaling relations)

$$\nu d = \gamma + 2\beta. \quad (\Gamma'.34)$$

### Γ'.3 ΒΑΘΜΙΣΗ ΠΕΠΕΡΑΣΜΕΝΟΥ ΜΕΓΕΘΟΥΣ

Ας επαναλάβουμε την ανάλυση της προηγούμενης παραγράφου στην περίπτωση που το σύστημά μας έχει πεπερασμένο μέγεθος. Για το λόγο αυτό, θα υποθέσουμε ότι το σύστημα έχει τους βαθμούς ελευθερίας του πάνω σε ένα πλέγμα με γραμμική διάσταση  $l = La$  (ο όγκος  $V = l^d$ ,  $d$  ο αριθμός των διαστάσεων), όπου το  $L$  είναι ο (αδιάστατος) αριθμός των πλεγματικών θέσεων και  $a$  είναι το μήκος που δίνει την πλεγματική σταθερά. Θα υποθέσουμε ότι παίρνουμε το όριο  $L \rightarrow \infty$  με  $a \rightarrow 0$  κρατώντας το μήκος  $l$  σταθερό. Αλλάζοντας την κλίμακα του  $L$  μειώνοντας τις πλεγματικές θέσεις κατά ένα παράγοντα

$$L \rightarrow \frac{L}{b} \Leftrightarrow L^{-1} \rightarrow bL^{-1}, \quad (\Gamma'.35)$$

έτσι ώστε να μειωθεί

$$a \rightarrow ba \quad (\Gamma'.36)$$

η σχέση (Γ'.22) γενικεύεται στην

$$f(t, h, L^{-1}) = b^{-nd} f(tb^{ny_t}, hL^{ny_h}, b^n L^{-1}). \quad (\Gamma'.37)$$

Παίρνοντας πάλι το όριο (προσέγγιση κρίσιμου σημείου)  $tb^{ny_t} = t_0 < \infty \implies b^n \sim t^{-1/y_t}$  η παραπάνω σχέση γίνεται

$$f(t, h, L^{-1}) = t^{d/y_t} f(t_0, ht^{-y_n/y_t}, t^{-1/y_t} L^{-1}) = t^{d/y_t} \Psi(ht^{-y_n/y_t}, t^{-1/y_t} L^{-1}). \quad (\Gamma'.38)$$

Παραγωγίζοντας όπως και στην προηγούμενη παράγραφο και θέτοντας  $h = 0$  παίρνουμε

$$\chi(t, L^{-1}) = \left. \frac{\partial^2 f}{\partial h^2} \right|_{h=0} = t^{-\gamma} \phi_2(L^{-1}t^{-\nu}) = t^{-\gamma} \phi_2\left(\frac{\xi}{L}\right). \quad (\Gamma'.39)$$

όπου θέσαμε τις τιμές  $y_t = 1/\nu$ ,  $\phi_2(x) = \Psi^{(0,2)}(x, 0) = \partial^2 \Psi(x, z)/\partial z^2|_{z=0}$ .

Το θερμοδυναμικό όριο το παίρνουμε για  $L \gg \xi$  όπου  $\phi_2(\frac{\xi}{L}) \rightarrow \phi_2(0) < \infty$ , οπότε παίρνουμε τη γνωστή σχέση  $\chi \sim t^{-\gamma}$ .

Αλλά όταν το  $L$  συγκρίνεται με το  $\xi$  επικρατούν τα φαινόμενα επίδρασης πεπερασμένου μεγέθους. Αυτά “πνίγουν” τις μεγάλες διακυμάνσεις και η μαγνητική επιδεκτικότητα παρουσιάζει ένα μέγιστο στη θερμοκρασία crossover  $t_X \equiv (\beta_c - \beta_c(L))/\beta_c$  όπου  $t_X \sim L^{-1/\nu}$ , όπου η τελευταία σχέση ισχύει γιατί  $L \sim \xi \sim t^{-\nu}$ . Παίρνουμε τότε

$$\chi_{\max} \sim t_X^{-\gamma} \phi_2(L^{-1}t_X^{-\nu}) \sim L^{\gamma/\nu} \phi_2(L^{-1}L) \sim L^{\gamma/\nu} \phi_2(1) \sim L^{\gamma/\nu}. \quad (\Gamma'.40)$$

Άρα στην περιοχή του μεγίστου θα πάρουμε μια συναρτησιακή σχέση της μορφής

$$\chi(t, L^{-1}) = L^{\gamma/\nu} F_\chi(L^{1/\nu}t), \quad (\Gamma'.41)$$

που δεν είναι άλλη από την εξίσωση (6.31). Η συνάρτηση  $F_\chi(x)$  είναι αναλυτική ως προς το όρισμά της  $x = L^{1/\nu}t$ , αφού για το πεπερασμένο σύστημα η  $\chi(t, L^{-1})$  είναι αναλυτική συνάρτηση της θερμοκρασίας<sup>3</sup>. Στο θερμοδυναμικό όριο ( $L \rightarrow \infty$ ,  $t$  πεπερασμένο άρα  $x \rightarrow \infty$ )

$$F_\chi(x) \sim x^{-\gamma} \quad x \gg 1, \quad (\Gamma'.42)$$

ώστε  $\chi(t, L^{-1}) = L^{\gamma/\nu} F_\chi(L^{1/\nu}t) \sim L^{\gamma/\nu} (L^{1/\nu}t)^{-\gamma} \sim t^{-\gamma}$  και κοντά στο ψευδοκρίσιμο σημείο

$$F_\chi(x) = F_{\chi,0} + F_{\chi,1}x + F_{\chi,2}x^2 + \dots \quad x \ll 1 \quad (\Gamma'.43)$$

οπότε περιμένουμε να έχουμε για  $L^{1/\nu}t \ll 1$

$$\chi(t, L^{-1}) = L^{\gamma/\nu} \left(1 + \chi_1 L^{1/\nu}t + \chi_2 L^{2/\nu}t^2 + \dots\right) \quad (\Gamma'.44)$$

Η παραπάνω σχέσεις μας οδηγούν στα συμπεράσματα:

<sup>3</sup> Προκύπτει από τον ορισμό της συνάρτησης επιμερισμού η οποία είναι αναλυτική συνάρτηση της θερμοκρασίας στην ψευδοκρίσιμη περιοχή. Άρα η  $x = L^{1/\nu}t$  είναι η μεταβλητή βάθμισης και όχι μια δύναμη αυτής λ.χ. η  $\tilde{x}$  που χρησιμοποιείται στις σχέσεις (6.25) και (6.26).

- Το ψευδοκρίσιμο σημείο μετατοπίζεται σαν  $\sim L^{-1/\nu}$  (σχέση (6.22))
- Το ύψος της κορυφής της μαγνητικής επιδεκτικότητας αυξάνει σαν  $\chi_{\max} \sim L^{\gamma/\nu}$
- Η κατεύθυνση μετατόπισης της κορυφής της μαγνητικής επιδεκτικότητας εξαρτάται από τις συνοριακές συνθήκες:
  - Περιοδικές συνθήκες καταπιέζουν τα αποτελέσματα των διακυμάνσεων, αφού τα wave vectors περιορίζονται από  $\frac{2\pi}{L}n$ . Αυτό αυξάνει την ψευδοκρίσιμη θερμοκρασία  $T_c(L)$  ( $\beta_c(L) < \beta_c \implies c > 0$  στην (6.22)).
  - Ελεύθερες συνοριακές συνθήκες οδηγούν σε ελεύθερες διακυμάνσεις στο σύνορο οι οποίες μειώνουν την ψευδοκρίσιμη θερμοκρασία  $T_c(L)$  ( $\beta_c(L) > \beta_c \implies c < 0$  στην (6.22))
  - Παγωμένα σπιν στο σύνορο οδηγούν σε αυξημένη τάξη στο σύστημα. Αυτό αυξάνει την ψευδοκρίσιμη θερμοκρασία  $T_c(L)$  ( $\beta_c(L) < \beta_c \implies c > 0$  στην (6.22)).

Καταλήγουμε ότι η  $F_\chi(L^{1/\nu}t)$  εξαρτάται από τις συνοριακές συνθήκες και τη γεωμετρία του πλέγματος.

Με ανάλογο τρόπο παίρνουμε

$$\langle m^k \rangle \sim L^{-d} \frac{\partial^k f}{\partial h^k} \sim L^{-d} t^{d/y_t - ky_h/y_t} \phi_k(L^{-1}t^{-\nu}) \sim L^{-d} t^{\nu d - k\nu y_h} \phi_k\left(\frac{\xi}{L}\right), \quad (\Gamma'.45)$$

και με ανάλογα επιχειρήματα που μας οδήγησαν στην εξίσωση (Γ'.41) παίρνουμε

$$\langle m^k \rangle \sim L^{-d} L^{-d+ky_h} F_k(L^{1/\nu}t) \sim L^{k\frac{\beta+\gamma}{\nu}} F_k(L^{1/\nu}t). \quad (\Gamma'.46)$$

Για το Binder cumulant θα έχουμε από τα παραπάνω

$$U = 1 - \frac{\langle m^4 \rangle}{3 \langle m^2 \rangle^2} \sim 1 - \frac{L^{4y_h} F_4(L^{1/\nu}t)}{3(L^{2y_h} F_2(L^{1/\nu}t))^2} \sim U_* + U_1(L^{1/\nu}t) + U_2(L^{1/\nu}t)^2 + \dots, \quad (\Gamma'.47)$$

όπου στην τελευταία ισότητα αναπτύξαμε τις αναλυτικές συναρτήσεις  $F_{2,4}(L^{1/\nu}t)$  για μικρά  $L^{1/\nu}t$ . Βλέπουμε τότε ότι

$$\frac{\partial U}{\partial \beta} \sim \partial_t U \sim L^{1/\nu}. \quad (\Gamma'.48)$$

Παραγωγίζοντας την (Γ'.38) ως προς τη θερμοκρασία παίρνουμε

$$\begin{aligned}
\frac{\partial f}{\partial t} &\sim t^{d/y_t-1} \Psi(ht^{y_h/y_t}, L^{-1}t^{-1/y_t}) \\
&\quad + t^{d/y_t} (ht^{y_h/y_t-1}) \Psi^{(1,0)}(ht^{y_h/y_t}, L^{-1}t^{-1/y_t}) \\
&\quad + t^{d/y_t} L^{-1}t^{-1/y_t-1} \Psi^{(0,1)}(ht^{y_h/y_t}, L^{-1}t^{-1/y_t}) \\
&\sim t^{\nu d-1} \Psi(ht^{\nu y_h}, L^{-1}t^{-\nu}) \\
&\quad + ht^{\nu d+\nu y_h-1} \Psi^{(1,0)}(ht^{\nu y_h}, L^{-1}t^{-\nu}) \\
&\quad + L^{-1}t^{\nu d-1-\nu} \Psi^{(0,1)}(ht^{\nu y_h}, L^{-1}t^{-\nu}), \tag{Γ'.49}
\end{aligned}$$

όπου υιοθετήσαμε το συμβολισμό  $\Psi^{(n,m)}(x, z) = \partial^{n+m} \Psi(x, z) / \partial x^n \partial z^m$ . Ο όρος που είναι ανάλογος του  $h$  εξαφανίζεται όταν θέσουμε  $h = 0$ . Στην ψευδοκρίσιμη περιοχή με  $t_X \sim L^{-1/\nu}$  ο πρώτος και ο τρίτος όρος είναι ίδιας τάξης ως προς  $L$  και παίρνουμε

$$\left. \frac{\partial f}{\partial t} \right|_{h=0} = L^{-d+\frac{1}{\nu}} F^1(L^{1/\nu}t), \tag{Γ'.50}$$

και με διαδοχικές παραγωγίσεις

$$\left. \frac{\partial^k f}{\partial t^k} \right|_{h=0} = L^{-d+\frac{k}{\nu}} F^k(L^{1/\nu}t). \tag{Γ'.51}$$

Οι παράγωγοι

$$\left. \frac{\partial^2 f}{\partial t \partial h} \right|_{h=0} = L^{-d+y_h+\frac{1}{\nu}} F_1^1(L^{1/\nu}t) = L^{\frac{1-\beta}{\nu}} F_1^1(L^{1/\nu}t), \tag{Γ'.52}$$

$$\left. \frac{\partial^{1+k} f}{\partial t \partial h^k} \right|_{h=0} = L^{-d+ky_h+\frac{1}{\nu}} F_k^1(L^{1/\nu}t). \tag{Γ'.53}$$

Ειδικότερα

$$\frac{\langle E m^k \rangle}{\langle m^k \rangle} = \frac{\left. \frac{\partial^{1+k} f}{\partial t \partial h^k} \right|_{h=0}}{\left. \frac{\partial^k f}{\partial h^k} \right|_{h=0}} \sim \frac{L^{-d+ky_h+\frac{1}{\nu}}}{L^{-d+ky_h}} \sim L^{1/\nu} \tag{Γ'.54}$$

$$\frac{\langle e^4 \rangle}{\langle e^2 \rangle^2} = \frac{L^{-d} \left. \frac{\partial^4 f}{\partial t^4} \right|_{h=0}}{\left( L^{-d} \left. \frac{\partial^2 f}{\partial t^2} \right|_{h=0} \right)^2} \sim \frac{L^{-\frac{4}{\nu}}}{(L^{-\frac{2}{\nu}})^2} \sim \text{σταθ.} \tag{Γ'.55}$$

## Γ'.4 ΠΡΟΒΛΗΜΑΤΑ

1. Δείξτε ότι για μία γκαουσιανή κατανομή  $f(x) = ae^{-x^2/2\sigma^2}$  έχουμε  $\langle x^2 \rangle = \sigma^2$  και  $\langle x^4 \rangle = 3\sigma^4$ . Συμπεράνετε ότι  $1 - \langle x^2 \rangle / (3 \langle x^4 \rangle) = 0$ .

2. Θεωρήστε την κατανομή που δίνεται από την πυκνότητα πιθανότητας

$$f(x) = a \left( e^{-\frac{(x-m)^2}{2\sigma^2}} + e^{-\frac{(x+m)^2}{2\sigma^2}} \right)$$

Κάντε τη γραφική της παράσταση και εξηγήστε γιατί μοιάζει με την κατανομή της μαγνήτισης στη φάση χαμηλής θερμοκρασίας  $\beta \gg \beta_c$ . Δείξτε ότι  $\langle x^4 \rangle = m^4 + 6m^2\sigma^2 + 3\sigma^4$  και  $\langle x^2 \rangle = m^2 + \sigma^2$ . Ερμηνεύστε τα αποτελέσματα, δηλ. τι εκφράζει η κάθε αναμενόμενη τιμή. Δείξτε ότι για  $\sigma \ll m$  παίρνουμε  $U \approx 2/3$ . Βεβαιωθείτε ότι η προσέγγιση που κάνατε αφορά το σύστημα που μελετάμε στη χαμηλή θερμοκρασία.

3. Υπολογίστε την παράγωγο  $\partial U / \partial \beta$  σαν συνάρτηση των  $\langle em^4 \rangle, \langle em^2 \rangle, \langle m^4 \rangle, \langle m^2 \rangle$ . Εφαρμόστε τα επιχειρήματα βάθμισης πεπερασμένου μεγέθους και αποδείξτε την εξίσωση (Γ'.48).

## ΠΑΡΑΡΤΗΜΑ Δ'

# ΚΡΙΣΙΜΟΙ ΕΚΘΕΤΕΣ

### Δ'.1 ΟΡΙΣΜΟΙ

$$\begin{aligned} \alpha: c &\sim t^{-\alpha}, & c_{\max} &\sim L^{\alpha/\nu}, & c(t, L) &= L^{\alpha/\nu} F^2(L^{1/\nu} t) \\ \beta: m &\sim t^\beta, & m &\sim L^{-\beta/\nu}, & m(t, L) &= L^{-\beta/\nu} F_1(L^{1/\nu} t) \\ \gamma: \chi &\sim t^\gamma, & \chi_{\max} &\sim L^{\gamma/\nu}, & \chi(t, L) &= L^{\gamma/\nu} F_2(L^{1/\nu} t) \\ \nu: \xi &\sim t^{-\nu}, & \xi &\sim L, \\ \delta: M &\sim h^{1/\delta} \\ z: \tau &\sim \xi^z \end{aligned} \quad (\Delta'.1)$$

μαζί με τη σχέση βάρθμησης που ορίζει τους εκθέτες  $y_t, y_h$

$$f(t, h) = t^{d/y_t} \Psi(ht^{y_h/y_t}) \quad (\Delta'.2)$$

και την σχέση που ορίζει τον εκθέτη  $\eta$  από τη συνάρτηση συσχέτισμού δύο σημείων  $G(\mathbf{r}, t) = \langle \mathbf{s}(\mathbf{r}) \cdot \mathbf{s}(\mathbf{0}) \rangle$  όπου στο κρίσιμο σημείο  $t = 0$  έχουμε

$$G(\mathbf{r}, t = 0) \sim \frac{1}{r^{d-2+\eta}}. \quad (\Delta'.3)$$

### Δ'.2 ΣΧΕΣΕΙΣ

Από τους ορισμούς και τις σχέσεις υπερβάρθμησης (hyperscaling relations) έχουμε

$$\begin{aligned} \alpha + 2\beta + \gamma &= 2 \\ \gamma + 2\beta &= \nu d \\ 2 - \nu d &= \alpha \\ \alpha + \beta(1 + \delta) &= 2 \\ \nu(2 - \eta) &= \gamma \end{aligned} \quad (\Delta'.4)$$

Μοντέλο	$\nu$	$\alpha$	$\beta$	$\gamma$	$\delta$	$\eta$	$y_t$	$y_h$
q=0 Potts (2d) [26]	$\infty$	$-\infty$	$\frac{1}{6}$	$\infty$	$\infty$	0	0	2
q=1 Potts (2d) [26]	$\frac{4}{3}$	$-\frac{2}{3}$	$\frac{5}{36}$	$2\frac{7}{18}$	$18\frac{1}{5}$	$\frac{5}{24}$	$\frac{3}{4}$	$\frac{91}{48}$
Ising (2d) [26]	1	0	$\frac{1}{8}$	$\frac{7}{4}$	15	$\frac{1}{4}$	1	$\frac{15}{8}$
q=3 Potts (2d) [26]	$\frac{5}{6}$	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{13}{9}$	14	$\frac{4}{15}$	$\frac{6}{5}$	$\frac{28}{15}$
q=4 Potts (2d) [26, 38]	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{12}$	$\frac{7}{6}$	15	$\frac{1}{4}$	$\frac{3}{2}$	$\frac{15}{8}$
classical (4d) [37]	$\frac{1}{2}$	0	$\frac{1}{2}$	1	3	0	2	3
Spherical (3d) [37]	1	-1	$\frac{5}{2}$	2	5	0	1	$\frac{5}{2}$
Ising (3d) [37]	—	$\frac{1}{8}$	$\frac{5}{16}$	$\frac{5}{4}$	5	—	—	—
Ising (3d) [41]	0.631	0.108(5)	0.327(4)	1.237(4)	4.77(5)	0.039	—	—
Heisenberg (3d)[39]	0.70	-0.1	0.36	1.4	5	0.03	—	—
XY (3d) [40]	0.663	—	—	1.327(8)	—	—	—	—
AF q=3 Potts (3d)[42]	0.66	-0.011	0.351	1.309	4.73	—	—	—

Πίνακας Δ'.1: Τιμές για τους κρίσιμους εκθέτες για τα μοντέλα που αναφέρονται στην πρώτη στήλη. Όπου οι τιμές είναι με υποδιαστολή είναι προσεγγιστικές, οι υπόλοιπες ακριβώς υπολογισμένες. Στις προσεγγιστικές τιμές δεν εφαρμόζουμε σχέσεις υπερβάθμισης αλλά αναφέρουμε απλά τις τιμές της βιβλιογραφίας. Οι τιμές για το πρότυπο 3d Ising από την [37] είναι εικασία. Για 3d Ising δεξ επίσης [6] p. 244. 3d XY και 3d AF q=3 Potts εικάζεται ότι ανήκουν στην ίδια κλάση παγκοσμιότητας. Wu for q=1 Potts (2d) reports  $y_h = 51/48$  and for q=4 (2d)  $\eta = 1/2$ , not compatible with hyperscaling relations - assume typos.

$$y_t = \frac{1}{\nu} = \frac{d}{2 - \alpha} \quad y_h = \frac{\beta + \gamma}{\nu} = \frac{1}{2} \left( d + \frac{\gamma}{\nu} \right) = d - \frac{\beta}{\nu} \quad (\Delta'.5)$$

$$\alpha = 2 - \frac{d}{y_t} \quad \beta = \frac{d - y_h}{y_t} \quad \gamma = \frac{2y_h - d}{y_t} \quad \delta = \frac{y_h}{d - y_h} \quad (\Delta'.6)$$

$$\eta = d + 2 - 2y_h \Leftrightarrow d - 2 + \eta = 2(d - y_h) \quad (\Delta'.7)$$

### Δ'.3 ΠΡΟΒΛΗΜΑΤΑ

1. Από τις σχέσεις (Δ'.4) και τις  $y_t = 1/\nu$ ,  $\gamma = (2y_h - d)/y_t$  να δείξετε τις υπόλοιπες ισότητες στις σχέσεις (Δ'.5) και (Δ'.6).