



LIBHARY Michigan State University

This is to certify that the dissertation entitled

RIGIDITY AND SELF-ORGANIZATION OF RANDOM NETWORKS

presented by

Mykyta V. Chubynsky

has been accepted towards fulfillment of the requirements for the

Ph.D.	degree in	Physics
/	$\frac{1.7.7}{\text{Major Professors}}$	NL Signature

Date

MSU is an Affirmative Action/Equal Opportunity Institution

PLACE IN RETURN BOX to remove this checkout from your record. TO AVOID FINES return on or before date due. MAY BE RECALLED with earlier due date if requested.

.

.

DATE DUE	DATE DUE	DATE DUE
L DE C 1 9 2004		
_		

6/01 c:/CIRC/DateDue.p65-p.15

RIGIDITY AND SELF-ORGANIZATION OF RANDOM NETWORKS

By

Mykyta V. Chubynsky

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Physics & Astronomy

ABSTRACT

RIGIDITY AND SELF-ORGANIZATION OF RANDOM NETWORKS

By

Mykyta V. Chubynsky

In this dissertation, we consider various aspects of rigidity theory of random networks. First, we review previous results of rigidity theory, as well as its applications. Extending previous work on rigidity of two-dimensional Bethe lattices and related loopless networks, we construct the Bethe lattice rigidity theory of three-dimensional bond-bending networks modeling covalent glasses. We study how the location of the rigidity transition depends on the chemical order in the system. We then consider bond-bending networks with perfect chemical order and show that the rigidity transition in this case is always first order, even on regular networks with loops, unlike the second-order transition in randomly bond-diluted cases. Next, we study rigidity of central-force (non-bond-bending) networks in three dimensions that were not studied in detail previously due to the fact that a fast computer algorithm commonly used in rigidity studies is not, generally speaking, applicable to these systems. We show that this algorithm, while not exact, is very accurate in most cases and virtually exact in some cases and use it to study rigidity properties of diluted three-dimensional central-force networks. We also develop a slower algorithm for studying rigidity of such networks that is always exact. Finally, we consider a model of self-organization of networks that is meant to model certain aspects of non-randomness in glasses and their influences on rigidity. We show that in this model there are two separate phase transitions instead of one, the rigidity transition and the stress transition, with the intermediate phase between them being marginally rigid (with elastic moduli zero in the thermodynamic limit), but stressless. We consider a similar model of selforganization for conventional connectivity percolation, where the intermediate phase

also exists. We show numerically that in this model the conductivity of corresponding resistor networks depends linearly on bond concentration, just like in the mean-field theory of percolation, and prove this linearity for a closely related model. We finish by reviewing possible experimental evidence of the intermediate phase.

Acknowledgements

First of all, I would like to thank my advisor, Prof. Michael F. Thorpe. Needless to say, without him this work would have been impossible. I am also grateful to my other collaborators: Prof. D.J. Jacobs (Calif. State Northridge), Prof. W. Whiteley (York, Canada), Prof. J.C. Phillips (Rutgers) and Dr. A.J. Rader (Pittsburgh). Interactions with other members of Mike Thorpe's group, especially Ming Lei, as well as numerous visitors, were also important.

I am also grateful to other members of my guidance committee, Prof. N.O. Birge, Prof. P.M. Duxbury, Prof. S.D. Mahanti and Prof. C.-P. Yuan. Special thanks to Prof. Mark Dykman, who gave a lot of useful advice on both scientific and nonscientific matters throughout my stay at MSU.

The final copy of this dissertation was produced using a LATEX template created by former MSU students Robert Slater and Peter Peterson, who graciously made it available online.

Big thanks also to the former group secretary, Janet King, for secretarial support well above and beyond the call of duty.

While my education at MSU was important, my undergraduate institution, Kiev Shevchenko University (KSU) in Ukraine, deserves at least as much credit. I cannot say enough how grateful I am to my teachers at KSU, who have done a great job in very difficult conditions of a newly independent country. My undergraduate education saved me a lot of time and effort in graduate school.

Life is not only about doing research, and I was lucky to have met a few good friends and helpful people. Tatyana Sharpee has been a good friend and helped me a lot during my first year at MSU. Mikhail Katkov and Irma Kuljanishvili were always ready to help as well. Valentin Levashov had been my roommate for more than five years. And friendship with Tatiana Sevastianenko has been very enjoyable and made me a better person.

Last but not least, I thank my mother and grandmother for bringing me up and for their constant support and encouragement.

Table of Contents

L		OF TA	BLES	ix
LI	LIST OF FIGURES			x
1	Rev	view of	rigidity theory	1
	1.1	Introd	luction	1
	1.2	Study	ing rigidity	6
		1.2.1	Maxwell counting	7
		1.2.2	The pebble game algorithm for 2D networks	10
		1.2.3	Rigidity percolation in 2D	13
		1.2.4	Rigidity of 3D glass networks	17
		1.2.5	Covalent glass networks	20
		1.2.6	Body-bar representation	24
	1.3	Appli	cations of rigidity theory	28
		1.3.1	Network glasses	28
		1.3.2	Proteins	31
2	Rar	ndom l	Bond Model	33
	2.1	Rigidi	ty percolation on Bethe lattices and RBNs	35
	2.2	The ra	andom bond model of glass networks	41
		2.2.1	Chemical order	42

		2.2.2	Transfer matrix equations	49
		2.2.3	Floppy modes and locating the transition	54
		2.2.4	Networks with 2- and 4-coordinated sites	63
	2.3	Netwo	rks with dangling ends	63
		2.3.1	Networks without 1-2 bonds	65
		2.3.2	Randomly bonded 1-2-3 networks	66
	2.4	Conclu	usion	69
3	Che	emicall	y ordered networks	70
4	Rig	idity o	f general 3D networks	83
	4.1	Basic	notions of 3D rigidity	84
	4.2	Rigidi	ty analysis through network relaxation	94
		4.2.1	The algorithm	95
		4.2.2	An example: colloidal glass networks	103
	4.3	Pebble	e game for general 3D networks	109
		4.3.1	The algorithm	109
		4.3.2	Test for bond-bending networks	111
	4.4	Centra	al-force networks — first-order transition	115
		4.4.1	Topologically 3-dimensional networks	115
		4.4.2	Topologically 2-dimensional networks	120
	4.5	Prosp	ects for an exact algorithm	122

	4.6	Conclu	usion	124
5	Self	organ	ization	126
	5.1	Self-or	ganization in rigidity percolation	128
		5.1.1	Description of the model	128
		5.1.2	General properties	129
		5.1.3	Intermediate phase in 2D central-force networks	129
		5.1.4	Intermediate phase in 3D bond-bending networks	134
	5.2	Self-or	ganization in connectivity percolation	138
		5.2.1	The model	138
		5.2.2	The intermediate phase	140
	5.3	Condu	ictivity and elasticity	141
		5.3.1	Conductivity	141
		5.3.2	Mean-field behavior of conductivity	146
		5.3.3	Superconducting networks	153
		5.3.4	Elasticity	155
	5.4	Exper	imental evidence	156
	5.5	Conclu	usion	161
6	Con	clusio	n and outlook	162
B	BLI	OGRA	APHY	164

List of Tables

6.1	The summary of rigidity transitions in networks of various topologies	
	in both two and three dimensions. The top line shows the dimension-	
	ality of the space in which the network is embedded. The second line	
	specifies the topology of the network. The table shows the order of	
	each transition to our best knowledge. The entries discussed in this	
	dissertation are in red	162

List of Figures

1.1	(a) A floppy network with one internal floppy motion (shown with	
	arrows); (b) A rigid, but not stressed (isostatic) network; (c) A rigid	
	and stressed network. Images in this dissertation are presented in color.	4

- 1.2 An example of a graph, for which Maxwell counting is wrong. The black part is stressed and contains one redundant constraint.9
- 1.3 A demonstration of the pebble game. Independent (redundant) bonds are shown with solid (dashed) lines which are (are not) covered by a pebble. Large (filled, open) circles denote (anchored, free) pebbles. The two pebbles closest to a given site are tethered to that site. Small open circles denote pivots (sites belonging to more than one rigid cluster). Small filled circles are sites belonging to just one rigid cluster. Overconstrained bonds are shown with heavy red lines. Yellow regions are rigid bodies. The left panel shows how pebbles are reshuffled to free the fourth pebble at the end of the test constraint. Since the fourth pebble is found, the test constraint is independent and is covered by a pebble (right panel). There are four free pebbles left in the right panel, denoting four floppy modes. The figure is adapted from Ref. [5]. . . .
- 1.4 Two typical fragments of a triangular network, below (left panel) and above (right panel) the rigidity transition. Green points are pivots between several rigid clusters. Black bonds are stressed, red bonds are unstressed. Below the transition, the network is floppy overall, but there are a few rigid pieces having no pivots. Above the transition, the network is rigid and stressed, but there are a few floppy pockets. Adapted from Ref. [6].

12

- 1.6 The fractions of bonds in the percolating rigid cluster (red open circles) and in the percolating stressed region (black filled circles) for the diluted triangular net. Averaged over two realizations on a 400 × 400 lattice.
 16

1.7	An example of a bond-bending network. There is an angular constraint between every two central-force constraints.	18
1.8	A schematic drawing of a covalent glass network. There are atoms of valence (coordination) 2 (blue), 3 (green) and 4 (red).	21
1.9	The number of floppy modes per degree of freedom f for the diluted amorphous silicon network (red circles) and the diluted diamond lat- tice (green diamonds). The dashed black line is the Maxwell counting result, Eq. (1.8). The inset shows the second derivative of f , with the cusp at the transition clearly seen in both cases. Adapted from Ref. [6].	25
1.10	The fractions of sites in the percolating rigid cluster (open red circles) and in the percolating stressed region (filled black circles) for the di- luted diamond lattice. The results are averages over 11 realizations on networks of 125000 sites. The rounding near the transition is due to finite-size effects.	26
1.11	The correspondence between bond-bending networks (top) and body- bar networks (bottom). The bond-bending network has one internal motion - a rotation around a hinge. Likewise, a system of two bodies with 6 degrees of freedom each connected by 5 bars has one internal degree of freedom.	27
1.12	The frequency of a particular vibrational mode in a Ge-S glass measured by Raman scattering as a function of the mean coordination number. Different mean coordination numbers are obtained by varying the composition. A break is seen near $\langle r \rangle = 2.4$. Adapted from Ref. [26]	30
1.13	The rigid cluster decomposition for the HIV protease. Only the protein backbone is shown and not the side chains. There is one big rigid cluster (blue), but the flaps consist of many small clusters and thus are flexible and can easily move.	32
2.1	A random bond network. Sites are connected at random, regardless of the distance	34
2.2	A 3-fold coordinated tree. The root is red, the leaves are green	36

2.3	A tree connected to a busbar. Present bonds are solid lines, missing bonds are dotted lines. Sites in level 0 (green) are rigidly attached to the busbar.	37
2.4	Solutions of equation (2.2) for the probability that a site has zero de- grees of freedom, T_0 . There is always a trivial solution identically equal to zero (green). There may also be non-trivial solutions (red), stable (solid line) and unstable (dashed line). The black line shows the solu- tion that is actually realized in RBNs	39
2.5	A bond-bending tree. Present bonds are solid red lines, missing bonds are dotted red lines. There are angular constraints (blue dashed lines) between every pair of present bonds stemming from the same site	42
2.6	The allowed region (white) in the plane of two parameters, the mean coordination $\langle r \rangle$ and the fraction of 2-3 bonds among all bonds y_{23} characterizing chemical order. Line 1 is the left boundary [no 3-3 bonds; Eq. (2.13)] and line 2 is the right boundary [no 2-2 bonds; Eq. (2.14) of the allowed region. Line 3 corresponds to networks obtained by random dilution [Eq. (2.20)]. Line 4 corresponds to randomly bonded networks [Eq. (2.15)]. The upper corner of the allowed region represents perfectly chemically ordered networks; the bottom represents completely phase separated networks.	46
2.7	A tree of bodies and bars. Each bond consists of 5 bars; each site is a body having 6 degrees of freedom (when isolated). There are two types of bodies, 2- and 3-coordinated. In the representation we use there is always one bond pointing upwards from every site, regardless of its coordination.	50
2.8	A drawing illustrating how the equations for quantities $T_j^{(i)}$, (2.21) and (2.22), are constructed. Sites 1 and 3 are 2- and 3-coordinated, respectively; sites 2, 4 and 5 can have coordinations 2 or 3 and averaging over their possible coordinations should be done.	52
2.9	A drawing illustrating the site insertion procedure used to obtain the number of floppy modes and locate the transition. Site 1 is inserted next to a 2-coordinated site 2. Sites 3 and 4 can have coordination 2 or 3, and averaging over their possible coordinations should be done.	56

2.10	The number of floppy modes per degree of freedom f (thick red line)
	as a function of mean coordination $\langle r \rangle$ along the site insertion line 5
	in the phase diagram (Figure 2.11). This plot also illustrates how the
	rigidity transition can be located. Line 1 is the Maxwell counting line
	coinciding with the true number of floppy modes in the floppy region
	(below the transition). Line 2 is obtained by counting the floppy mode
	changes in the site insertion process (as described in the text). The
	transition is located at the point where these two lines intersect

58

59

60

- 2.11 The phase diagram for 2-3 random bond networks. The thick red line is the phase boundary separating the floppy and the rigid phases. The thin purple line is the spinodal line, where the non-trivial solution of equations (2.24) ceases to exist. Lines 1 to 4 are as in Figure 2.6. Line 5 is one of bond insertion lines given by Eq. (2.25).
- 2.12 Comparison between the theory and the pebble game simulation results for the first derivative of the number of floppy modes per degree of freedom with respect to the mean coordination for 2-3 networks obtained by random dilution. The solid lines represent the theory described in this chapter. Brown open circles are simulation results for a sample with 8000 sites, green solid triangles are for a sample with 32000 sites. The inset shows the blowup of the region near the rigidity transition, with the open black diamonds for a sample with 103823 sites (averaged over 30 realizations) and the filled green circles for a sample with 262144 sites (averaged over 20 realizations). Simulation and plot by A.J. Rader.
- 2.13 The fraction of sites in the infinite rigid cluster as a function of the mean coordination for randomly diluted 2-3 random bond networks. The solid lines represent the theory described here: the red line is the solution that is actually realized and the black line between the spin-odal point $\langle r \rangle_s$ and the transition point $\langle r \rangle_c$ is the "metastable" non-trivial solution where the trivial solution corresponding to the floppy phase is actually realized. The green dots are the pebble game simulation results averaged over 10 networks of 100000 sites.
- 2.14 The phase diagram for 2-4 networks. The thick red line is the phase boundary between the rigid (blue) and the floppy (yellow) phases. The thin purple line is the spinodal line. The black dashed line is the random bonding line.

 3.1 The fully 3-coordinated network of 1800 sites obtained in Ref. [50], as described in the text, and used in the simulations in this chapter 3.2 An illustration of the bond decoration process. The starting point is a fully 3-coordinated network (a). Bonds are chosen in random order and decorated with one site each (b), until all bonds are decorated (c). Then bonds of the original network start to be decorated with a second site (d) — bonds decorated with two sites are circled	35
 3.2 An illustration of the bond decoration process. The starting point is a fully 3-coordinated network (a). Bonds are chosen in random order and decorated with one site each (b), until all bonds are decorated (c). Then bonds of the original network start to be decorated with a second site (d) — bonds decorated with two sites are circled	71
3.3 The fractions of sites in both the percolating rigid cluster and the per- colating stressed region (going from 0 to 1 at the transition) for the chemically ordered networks obtained by the bond decoration proce- dure described in the text (red line), compared to the fractions of sites in the percolating rigid cluster (green open circles) and in the perco-	73
lating stressed region (green solid circles) for a network obtained by random bond dilution of the diamond lattice (as in Figure 1.10)	74
3.4 The number of floppy modes per degree of freedom for the chemically ordered networks obtained by decorating a network of 3 × 3 supercells shown in Figure 3.1 (red circles) and for the randomly bond diluted diamond lattice, as in Figure 1.9 (green diamonds). The black dashed line is the Maxwell counting result.	75
3.5 A set of panels illustrating finite size effects for chemically ordered networks. Moving along the horizontal axis corresponds to changing the number of decorating sites. The right border of the plot corresponds to a network with one atom decorating each bond, which is exactly $\langle r \rangle = 2.4$. Moving to the left corresponds to adding second sites to bonds one by one; the number of added sites is specified. Panel (a) shows the number of floppy modes F (red circles). The solid green line is Maxwell counting. Panels (b) and (c) show the fractions of sites in the percolating rigid cluster and in the percolating stressed region, respectively (red).	76
3.6 Illustration of a subnetwork with a dangling end. Adding a dangling end (green) to a subnetwork (red) increases the floppy mode count, so subnetworks with dangling ends need not be considered in the proof in this chapter	31

4.1	The double-banana graph. Two bananas (green and yellow) can rotate around the implied hinge (red dashed line). The 3D generalization of the Laman's theorem is wrong for this network	85
4.2	The double-banana graph from Figure 4.1 with one constraint in one of the bananas removed. The red subnetwork is a rigid cluster, but is only rigid due to the left banana and not by itself. However, inserting the implied hinge (green dashed line) explicitly makes the subnetwork rigid by itself.	87
4.3	The three-banana graph. Sites 1, 2 and 3 form a non-contiguous rigid cluster.	88
4.4	The double-banana graph with hinge rotation locked by a constraint (red).	90
4.5	Two examples of networks with pivots. Constraints and sites belonging to the same cluster have the same color; pivots shared between several clusters are black. Both examples are valid in 2D and in 3D	91
4.6	A network in 3D with a hinge (black).	91
4.7	Illustration of floppy mode counting for the double-banana graph. First, the implied hinge (red) is inserted explicitly. The first cluster (green plus the hinge) has 5 sites and 10 constraints, so Maxwell counting for it gives $5 \times 3 - 10 = 5$ floppy modes, and there must be 1 redundant constraint. Same for the second cluster (blue plus the hinge). Thus there are a total of two redundant constraints, and by adding them to the global Maxwell count for the whole network, the correct number of floppy modes is obtained.	93
4.8	Schematic illustration of the relaxation procedure. In the network of interest, the spring equilibrium lengths are chosen equal to the initial distances between sites they connect, so initially the network is unstrained (a). All sites are displaced at random (b) and then relaxed (c). The distances between sites 3 and 6 and between sites 4 and 5 in (c) differ from those in (a), so these pairs of sites are not mutually rigid. All other distances are preserved, so all other pairs of sites are mutually rigid. The actual procedure differs from this in that the problem is linearized, so all displacements are always effectively infinitesimal.	96

 $\mathbf{x}\mathbf{v}$

4.9	The logarithm of the sum over two realizations of the absolute value of the scalar product, Eq. (4.2), for all pairs of sites of a network with 216 sites. The gap between "zero" and "non-zero" values is clearly seen. Pairs with the values below the gap are mutually rigid, those with the	
	values above the gap are not.	98
4.10	Finding an implied hinge. Candidate endpoints of an implied hinge (A and B) have to be rigid and not directly connected. An arbitrary site C rigid with respect to both A and B is chosen. If there exists a site D also rigid with respect to both A and B, but not rigid with respect to C, then A-B is a hinge; otherwise it is not a hinge	100
4.11	A confocal microscope image of a colloidal glass [60]	105
4.12	A rigid tetrahedron with many hinges (black) and one non-hinge con- straint (red). Such rigid clusters do not show up well when the visual- ization scheme described here is used	106
4.13	Rigid cluster decomposition for a colloidal glass network obtained as described in the text for the time delay 252 s. The network appears almost flat, as it is rather thin, but is treated as three-dimensional. Bonds and sites belonging to just one cluster are colored, otherwise they remain black. Adjacent bonds/sites belonging to different rigid clusters have different colors. Implied hinges are shown with dashed black lines.	107
		100
4.14	Same as in Figure 4.13, but for the time delay 3240 s.	108
4.15	A schematic drawing of a trivial benign implied hinge. A shaded body is a rigid cluster. Three blue sites and two red bonds form a rigid cluster that, although not rigid by itself, is simple enough to be taken care of in the rigid cluster decomposition procedure. The implied hinge is the green dashed line	112
4.16	A configuration with a trivial benign hinge shared by two clusters (blue and green) that can arise in bond-bending networks. Central-force constraints are solid lines, angular constraints are dashed lines	113
4.17	A configuration with a malignant hinge shared by two clusters (blue and green) that can arise in bond-bending networks. Central-force con- straints are solid lines, angular constraints are dashed lines. Without the hinge, this is topologically equivalent to the double-banana graph in Dimme 4.1	110
	In r igure 4.1	113

4.18	The relative deviation of the floppy mode count obtained with con- straints inserted at random in the pebble game procedure from the correct result obtained with the proper order of insertion. Filled green circles are for networks of 1000 sites averaged over 100 realizations. Open red circles are for networks of 8000 sites averaged over 10 real- izations.	114
4.19	The number of floppy modes per degree of freedom in the top panel and the sizes of the percolating rigid cluster (green) and stressed region (red) in the bottom panel for the bond diluted central-force bcc net- works. Averages over 10 realizations on networks of 54000 sites. The dashed line in the top panel is Maxwell counting	118
4.20	Same as in Figure 4.19, for the fcc networks. Averages over 10 realiza- tions on networks of 62500 sites	119
4.21	The elastic moduli for the FCC network (figure adapted from [61]). Quantity p in the graph is the fraction of present bonds; thus $\langle r \rangle = 12p$.	120
4.22	A square lattice with diagonals used here in the study of a topologically 2D network in 3D.	121
4.23	The number of floppy modes per degree of freedom for the diluted square network with diagonals (as in Figure 4.22). Average over 10 networks with 40000 sites.	122
5.1	The number of floppy modes per degree of freedom for random and self-organized diluted triangular networks in 2D. The thresholds are shown with various symbols, as specified in the legend. The intermediate phase in the self-organized case is shaded. Note that the rigidity transition occurs at the same f in the random and self-organized cases. The number of floppy modes in the self-organized case is strictly linear and coincides with the Maxwell counting result all the way up to the stress transition at $\langle r \rangle = 2.4.$	132
5.2	The fractions of bonds in the percolating rigid cluster and percolating stressed region for random and self-organized diluted triangular networks in 2D. Averages over two realizations on 400×400 networks. The intermediate phase in the self-organized case is shaded	133

5.3	The number of floppy modes per degree of freedom for random and self- organized diluted diamond networks in 3D. The thresholds are shown with various symbols, as specified in the legend. The intermediate phase in the self-organized case is shaded. The rigidity thresholds are close in both cases, which is probably coincidental. The rigidity transitions definitely occur at different f , unlike the 2D CF case in Figure 5.1. The stress transition in the self-organized case is below the Maxwell counting threshold $\langle r \rangle = 2.4. \ldots \ldots \ldots \ldots \ldots$	136
5.4	The fractions of sites in the percolating rigid cluster and in the per- colating stressed region for self-organized diluted diamond networks in 3D. The intermediate phase is shaded. Circles are averages over 4 networks with 64000 sites, triangles are averages over 5 networks with 125000 sites. The dashed lines are the power law fit below the stress transition and for the guidance of the eye above. Note the break in the slope at the stress transition	137
5.5	The fractions of bonds in the percolating rigid cluster (blue dots) and in the percolating "stressed" region (green dots) in the connectivity self-organization model for the diluted square lattice. Ten realizations on the 400×400 lattice without averaging. The intermediate phase is shaded	141
5.6	The conducting backbone (black) in the busbar geometry (open bound- aries with busbars) in the intermediate phase. Bonds in the percolating cluster, but not in the backbone are red. Bonds not in the percolating cluster are green.	143
5.7	The conducting backbone in the source-sink geometry in the interme- diate phase. Bonds between the source (sink) and the respective edge of the network are treated on equal footing with regular lattice bonds and can be present or absent. The backbone (black) is a single path between the source and the sink. Bonds in the percolating cluster, but not in the backbone, are red. Bonds not in the percolating cluster are green.	144
5.8	The conductivity for the random (green circles) and self-organized (red circles) diluted square networks. Averages over 25 realizations on the 200×200 lattice. The mean field result (the black straight line reaching 1 at $\langle r \rangle = 4$ is also shown. The threshold for the self-organized model is always at $\langle r \rangle = 2$. The threshold for the random model is also at $\langle r \rangle = 2$ in this case, which is a peculiarity of the square lattice.	145

5.9	The conductivity for the random (green circles) and self-organized (red circles) diluted cubic networks. Averages over 25 realizations on the $30 \times 30 \times 30$ lattice. The mean field result (the black straight line reaching 1 at $\langle r \rangle = 6$ is also shown. The thresholds now do not coincide (unlike in Figure 5.8 for the square lattice).	146
5.10	A schematic diagram showing relations between trees and networks and between thickets and networks. The left column of dots denotes the set of all possible trees on the full lattice; the right column is the set of all thickets; the middle column is the set of all spanning networks with a certain number of bonds. The connections in the left part (between trees and networks) show what networks can be built by adding bonds to a tree, or, conversely, what trees are subgraphs of a given network. The connections in the right part show similar relations between thickets and networks. The ratio of the numbers of connections in the left part and in the right part is proportional to the conductivity of a network with a certain number of bonds in the thermodynamic limit, as discussed in the text	149
5.11	Conductivity for the restricted dilution model described in the text. Average over 25 realizations on the 100×100 square lattice. There is an "intermediate phase" of a different type, with the spanning cluster, but zero conductivity. The straight line is the mean-field result, from which there is obvious deviation.	152
5.12	Conductivity for the superconductor (blue dots) and resistor (green dots) networks in the self-organized model on the square lattice. The conductivity of resistor networks is not exactly zero in the intermediate phase, since open boundaries with the busbars were used, unlike in Figure 5.8, where periodic boundary conditions were used and the conductivity was exactly zero in the intermediate phase. This non-zero conductivity in the intermediate phase is a finite-size effect. All results are averages over ten square 100×100 networks	154
5.13	An elastic self-organized network in the intermediate phase with the applied stress backbone (black) shown. Bonds in the percolating cluster, but not in the backbone, are red. Bonds not in the percolating cluster are green.	157
5.14	The elastic modulus C_{11} for the random and self-organized diluted square networks. Averages over 10 realizations on the 100×100 lattice. The intermediate phase is shaded. The inset shows the blowup of the critical region. Deviations from the mean-field line (black) are clearly seen in both the random and the self-organized cases.	158

5.15	(a) Frequencies of two Raman modes corresponding to the motion of	
	corner-sharing tetrahedra in $Si_x Se_{1-x}$ shown as a function of mean	
	coordination; (b) Non-reversing heat variation in $Si_x Se_{1-x}$ (red filled	
	circles) and $Ge_{0.25}S_{0.75-x}I_x$ (green open circles) glasses. Both panels	
	are adapted from Ref. [83]	160

Chapter 1: Review of rigidity theory

1.1 Introduction

In 1929, almost immediately after quantum mechanics was discovered, P.A.M. Dirac pointed out that the "underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known ..." [1]. Recently, in 1999, another Nobel laureate, Robert Laughlin, gave a talk [2] starting with a slide jokingly titled "Theory of Everything" containing the Schroedinger equation for a system of electrons and nuclei and then listing many random things (from rocks to rockets and from hippos to humans) that can indeed be explained, at least in principle, by such a theory (one thing on the list was "anthrax" - which makes one wonder if Laughlin could foresee that this would become a hot issue in just over two years, but discussing that would lead us too far astray). Of course, by itself, as Laughlin put it, this "theory of everything wasn't worth anything," as the Schroedinger's equation is too hard to solve for all but the simplest systems. Nevertheless, in the 70 years since Dirac's assertion huge advances have been made and some standard methods (Hartree-Fock, density functional theory, Car-Parrinello, pseudopotentials) and approximations (Born-Oppenheimer, local density approximation) have been developed [3] that, together with the remarkable increase in computational power, have made studying properties of many condensed matter systems from first principles not only feasible, but almost routine. Yet, there are limitations, so condensed matter physics is not yet dead. Problems arise, for instance, for strongly correlated systems, where some of the usual approaches do not work or do not answer relevant questions, but also for systems

that have to be considered on too long length and/or time scales. In these cases, one needs to introduce and study simpler models that capture essential features of interest. In particular, this is why statistical-mechanical models, such as the Ising model, as well as various percolation models, are of interest and importance.

This dissertation deals with one of such simplified approaches based on *rigidity* theory. The model considered is that of a set of sites and constraints; we will call such a set a network. Sites are point-like objects. A constraint between two sites is a statement that the distance between these sites should take a certain value; whenever this distance is different, there is an associated energy cost. A constraint can thus be thought of as an elastic spring between a pair of sites; this spring should have a "natural" length corresponding to its being "undeformed", but whether the spring is harmonic or not is not important. It is assumed that interactions between sites not connected by a constraint do not exist. When this is considered as a model of a real condensed matter system, this is clearly a simplification, since in fact, all atoms in any such system interact. These interactions, however, are not all of equal strength: if, for example, atoms are covalently bonded, atoms connected by a bond (first neighbors), as well as second neighbors, interact much more strongly than other pairs of atoms, so our approach may turn out to be a reasonable one.

What are some of the relevant questions that one can address in connection with our model?

• First of all, we are interested in finding *rigid clusters*, which are sets of sites that are mutually rigid (i.e., cannot move with respect to each other without an energy cost). Consider the networks shown in Figure 1.1 (assuming that the problem is two-dimensional, so sites can only move in the plane). In Figure 1.1,(a) there is a possible zero-energy motion shown schematically by arrows; thus site A is not rigid with respect to site C and site B is not rigid with respect to site D and {A,B}, {B,C}, {C,D}, and {A,D} are separate rigid

clusters. In Figs. 1,(b) and (c), all sites are mutually rigid, so the only rigid cluster is {A,B,C,D}. The concept of rigid clusters is similar to the concept of clusters in ordinary percolation, which are sets of sites connected to each other; the notable difference, however, is that a site can belong to several rigid clusters at once. In general, we will see many parallels between *rigidity* and *connectivity* problems, although the former are more complex than the latter.

- Secondly, while in Figure 1.1, (a) and (b), all constraints can have their undeformed lengths, in Figure 1.1,(c) this is not the case in fact, all springs will be deformed, unless their natural lengths happen to take very special values. Thus, all constraints in Figure 1.1,(c) will be *stressed* and in general, for any network we can pose a problem of finding what constraints are stressed. A network or its part that is rigid but *not* stressed (as in Figure 1.1, (b)) is called *isostatic*.
- Thirdly, if we consider small displacements from equilibrium (harmonic approximation), we can ask how many motions (or normal modes) have zero frequency. In d dimensions, there are always at least d(d + 1)/2 zero-frequency motions corresponding to rigid-body translations and rotations of the network. Thus for d = 2 the number of zero-frequency motions is at least 3. In Figure 1.1,(b) and (c) there are no other zero-frequency motions; in Figure 1.1,(a) there is one more motion shown with arrows, thus the total number of zero-frequency motions is 4.

One important class of rigidity problems are *rigidity percolation* problems. In usual *connectivity percolation* [4], one considers a regular lattice (such as the square lattice in 2D) and, removing bonds or sites at random, asks a question at what concentration of bonds/sites there is a path between opposite boundaries, or, equivalently, a percolating cluster (infinite in the thermodynamic limit). Further,



Figure 1.1: (a) A floppy network with one internal floppy motion (shown with arrows); (b) A rigid, but not stressed (isostatic) network; (c) A rigid and stressed network. Images in this dissertation are presented in color.

one can be interested in the statistics of clusters, the size of the infinite cluster, etc. The connectivity percolation transition is *second order*, which means that the fraction of bonds/sites in the percolating cluster changes continuously from zero when the number of bonds/sites is increased beyond the transition point. We can ask similar questions for rigidity, by changing the number of constraints and asking if there is an infinite rigid cluster in the network. In fact, connectivity is a particular case of rigidity, when sites are only allowed to move in one dimension: in this case, whenever two sites are connected, they are rigidly connected. In situations considered initially (randomly diluted networks in 2D [5], as well as networks with all second-neighbor constraints present, so-called *bond-bending networks* [6]), the rigidity percolation transition was second order likewise; however, there are some models, in which the transition is *first order*, some of which will be considered here. One thing to notice is that besides the percolating rigid cluster, we can also ask a question whether stressed regions in the network percolate (*stress percolation*).

Often rigidity and stress percolation happen at the same concentration of constraints, however, there are exceptions, as we will see.

In this dissertation, we describe results of studies of rigidity of several different classes of networks. In Chapter 2, we discuss the problem of rigidity of trees (or Bethe lattices) and related random bond models. The big advantage of this problem is that, as often is the case with Bethe lattices, it is solvable and one does not have to perform numerical simulations, with their inherent finite size effects. Having in mind practical applications to network glasses, we consider covalent bond-bending networks with prescribed fractions of sites having different *coordination numbers* (the coordination number of a site is the number of bonds stemming out of that site), as well as varying degrees of *chemical order* expressed as probabilities of bonds connecting sites with certain coordination numbers. In Chapter 3, we consider networks with perfect chemical order, but disordered otherwise, and prove some very general results for such networks, most importantly, the fact that the rigidity transition in them is a very sharp first-order transition. In Chapter 4 we describe some preliminary, but very interesting and encouraging, results on *central-force* networks (without second-neighbor constraints) in 3 dimensions also exhibiting the first-order rigidity transition, unlike the previously considered bond-bending networks (with second-neighbor constraints). Finally, in Chapter 5, we introduce our model of network self-organization and show that this self-organization leads to presence of two separate rigidity and stress percolation transitions and an intermediate phase between them that is rigid but not stressed. Throughout this work, we emphasize theory, although real-life applications are mentioned.

In the rest of this chapter, we review previously obtained results of rigidity theory, as well as its applications. This is not intended as a comprehensive review; rather, the purpose is to introduce the concepts used in the following chapters and give the reader an idea of how the theory can be applied.

 $\mathbf{5}$

1.2 Studying rigidity

In this section we review various methods that were devised to solve the rigidity problem posed in the introduction.

First of all, while the exact penalty associated with violating each constraint in the network need not be specified, we can always for simplicity think of constraints as harmonic springs with spring constants assigned arbitrarily. Then our system is linear and we can write down its dynamical matrix. We can then find, at least in principle, the eigenvalues and eigenvectors of the dynamical matrix. Zero eigenvalues correspond to zero-frequency modes, or motions that cost no energy (also known as *floppy modes*). The corresponding eigenvectors describe the motions themselves, and from them it is possible to deduce what pairs of sites are rigid and thus find rigid clusters. It is also possible to find what bonds are stressed by relaxing the network of springs and seeing what springs remain strained after relaxation. This solves the rigidity problem in principle. But this solution is not quite satisfactory. Finding eigenvalues and eigenvectors is not particularly fast computationally, and the computational cost rises quite fast with the system size — as the cube of the number of degrees of freedom. Another problem is that, as with any numerical procedure dealing with real (non-integer) numbers, there are round-off errors, so that the eigenvalues that are supposed to be zero are actually slightly different from zero, so zero-frequency motions can be confused with those having very small but non-zero frequency. Perhaps as a minor (or maybe not so minor) point, there is also feeling of dissatisfaction from the fact that to solve the problem, we had to input a lot of irrelevant information, such as (arbitrary) values of spring constants, whereas just the geometry of the network (moreover, as it turns out, just its topology) should be enough. Clearly, we need methods that are faster and hopefully integer (i.e., do not deal with real numbers) and perhaps geometrical in origin.

1.2.1 Maxwell counting

A very simple, yet remarkably accurate method of finding the number of floppy modes was proposed by none other than J.C. Maxwell of the electromagnetic theory and Maxwell-Boltzmann distribution fame, in 1864 [7]. The method is known as *constraint counting* or *Maxwell counting*.

Suppose for now that there are no stressed constraints in the network, so that all constraints can be satisfied simultaneously. In the linear approximation each constraint is a linear relation between the system's coordinates. Then the number of floppy modes is the dimensionality of the space of solutions of the system of equations formed by these relations. Each new relation decreases the dimensionality of the space of solutions (i.e., the number of floppy modes) by one, *if* it is linearly independent of the rest of the relations; otherwise, it does not change the number of floppy modes. In the first case, the corresponding constraint is *independent*; in the second case it is *redundant*. Redundant constraints do not change the number of floppy modes or the configuration of rigid clusters. At the same time, they introduce stress in the network, giving rise to stressed, or overconstrained regions, since normally the linear systems with linearly dependent equations have no solutions, unless their right-hand sides are special (which in our case means special, non-generic, constraint lengths), and therefore all constraints cannot be accommodated simultaneously without deforming. In Figure 1.1,(b), all five constraints are independent; in Figure 1.1,(c), the sixth constraint is redundant.

Now, the assumption of Maxwell counting is that all constraints are independent, i.e., there are no redundant constraints. Then each constraint reduces the number of floppy modes by one, and since the number of floppy modes in a network of N sites in d dimensions without constraints is dN, the number of floppy

modes is

$$F = dN - N_c, \tag{1.1}$$

where N_c is the number of constraints. Equation (1.1) gives the number of floppy modes in the Maxwell counting approximation. Introducing the number of floppy modes per degree of freedom, f = F/dN, and the number of constraints per degree of freedom $n_c = N_c/dN$,

$$f = 1 - n_c.$$
 (1.2)

Clearly Maxwell counting is wrong, e.g., for the network in Figure 1.1,(c), since there is a redundant constraint in the network. In this case, however, it is not really dangerous, since Maxwell counting would give the number of floppy modes F = 2, which is clearly absurd, since any network in 2D (except that consisting of just a single isolated site) has at least 3 floppy modes corresponding to rigid body motions. More dangerous is the case in Figure 1.2, where the true number of floppy modes is 4, but Maxwell counting gives F = 3, and the error is not obvious. Despite these failures, for big random networks Maxwell counting is usually remarkably good, as we will see in the next subsection, where we compare Maxwell counting with exact results.

If the number of redundant constraints N_r is somehow known, we can use the fact that redundant constraints do not change the number of floppy modes and write *exactly*

$$F = dN - N_c + N_r. \tag{1.3}$$

Of course, this equation is useless so far, since there is no straightforward way of



Figure 1.2: An example of a graph, for which Maxwell counting is wrong. The black part is stressed and contains one redundant constraint.

finding the number of redundant constraints, short of actually solving the eigenproblem. The only useful conclusion from it at this point is that Maxwell counting gives a *lower bound* for F, since N_r is non-negative.

We now apply Maxwell counting to central-force random bond-diluted networks in 2D.

Suppose we start with a regular network in 2D (such as the triangular net) with all sites bonded to the same number z of other sites (z = 6 for the triangular net). Each bond in the network represents a constraint. We retain the fraction p of the bonds chosen at random and remove the rest. For convenience, we can introduce the mean coordination $\langle r \rangle$ as the average number of bonds stemming from a site: $\langle r \rangle = zp$. Then, if the number of sites is N, the total number of constraints is $N_c = \langle r \rangle N/2$ and the number of constraints per degree of freedom is $n_c = \langle r \rangle/4$, and Maxwell counting [Eq. (1.2)] gives

$$f = 1 - \frac{\langle r \rangle}{4}.\tag{1.4}$$

We note that for $\langle r \rangle < 4$, f > 0, so there is a macroscopic number of floppy modes in the network, so one can assume that the network is floppy as a whole. On the other hand, for $\langle r \rangle > 4$, f < 0, which obviously cannot be true, so definitely there are redundant constraints and stress and the network is probably rigid as a whole, with the true number of floppy modes close to zero. If these considerations are at least roughly true, there must be a *rigidity percolation transition* somewhere close to $\langle r \rangle = 4$. We will check these results in the next subsection after describing an exact algorithm for studying rigidity.

1.2.2 The pebble game algorithm for 2D networks

While Maxwell counting is remarkably accurate (as we will see), it is still not perfect and in particular, being of "mean-field" nature, it fails to adequately describe the critical region near the rigidity percolation transition. A real breakthrough in studying rigidity percolation came in 1995, when an *exact* algorithm, *the pebble game*, was proposed [5, 8]. The algorithm is integer and it also is very fast, with the computational effort scaling linearly with the network size, except in the critical region, where the scaling is roughly $\propto N^{1.15}$. Thus very big networks (up to a few million sites) can be analyzed routinely.

Consider the network in Figure 1.2. As we said, Maxwell counting is wrong for it, as there is one redundant constraint that Maxwell counting by itself done for the whole network cannot detect. But, if constraint counting is done *for every subnetwork* (subgraph) of the network, it will be seen that the black piece in Figure 1.2 apparently has 2 floppy modes, and thus the error (i.e., the presence of a redundant constraint) will be detected. It turns out that this is always the case, i.e., by doing constraint counting for all subnetworks, the presence of redundant constraints can *always* be detected. This is the essence of a theorem formulated and proved by Laman in 1970 [9]:

Theorem: A generic network in two dimensions does not have redundant constraints, if and only if no subnetwork containing n sites $(n \ge 2)$ and b bonds violates $b \le 2n - 3$.

The idea now is to build the network up one constraint at a time. After placing

each constraint, Laman's theorem is applied and thus the newly added constraint is tested for redundancy. All previously determined redundant constraints are ignored in the process, so that if redundancy is detected, it can only be due to the constraint placed last. By counting redundant constraints, we can determine the exact number of floppy modes using Eq. (1.3). This is the idea of the pebble game algorithm.

The pebble game is conducted in the following way. Each site in the network has two pebbles tethered to it that represent the site's two degrees of freedom. A pebble is either free or anchored to one of the constraints stemming from the site to which it is tethered. Each independent constraint should be covered by one pebble (belonging to one of its end sites) — this indicates that an independent constraint subtracts one floppy mode. Then the number of free pebbles is equal to the number of floppy modes. Each new constraint added to the network should be tested for independence. For this, freeing four pebbles is attempted at the ends of the newly inserted constraint (two pebbles at each end). The number of pebbles is equal to the number of degrees of freedom and the number of anchored pebbles is equal to the number of independent constraints. Then, if four pebbles can be freed, this means that the number of independent constraints in all subgraphs including the new constraint (not counting the new constraint itself) is less than the number of degrees of freedom in the subgraph by at least four, and thus, according to Laman's theorem, the constraint is independent and should be covered by one of the four freed pebbles. If only three pebbles can be freed (at least three can always be freed), the new constraint is redundant and is not covered. In the process of freeing pebbles, an anchored pebble can be freed anywhere in the network, if the constraint that it covered is covered immediately from the opposite end, so that covered constraints remain covered all the time. Thus free pebbles can "propagate" through the network. This process is illustrated in Figure 1.3. Since each bond is covered at just one end, the network becomes a *directed graph*, with "arrows" at each bond



Figure 1.3: A demonstration of the pebble game. Independent (redundant) bonds are shown with solid (dashed) lines which are (are not) covered by a pebble. Large (filled, open) circles denote (anchored, free) pebbles. The two pebbles closest to a given site are tethered to that site. Small open circles denote pivots (sites belonging to more than one rigid cluster). Small filled circles are sites belonging to just one rigid cluster. Overconstrained bonds are shown with heavy red lines. Yellow regions are rigid bodies. The left panel shows how pebbles are reshuffled to free the fourth pebble at the end of the test constraint. Since the fourth pebble is found, the test constraint is independent and is covered by a pebble (right panel). There are four free pebbles left in the right panel, denoting four floppy modes. The figure is adapted from Ref. [5].

pointing in the direction of the covering pebble, and this specifies the paths in which free pebbles can propagate, so search for pebbles that can propagate to a given destination is simplified.

If the new constraint is redundant, the region over which the (failed) search for the fourth pebble was conducted is recorded and it represents the *stressed* (*overconstrained*) region created by the constraint. Stressed regions are sets of constraints that are stressed and deform concurrently when any constraint in the set undergoes deformation. If the new stressed region overlaps with any previously determined stressed region(s) and the overlap is two sites or more, the stressed regions are merged. Obviously, all stressed bonds can be found in this way. Note that every stressed region contains a certain number of redundant constraints, but exactly what constraints in the region are deemed redundant and thus are not covered by a pebble is arbitrary and depends on the order of insertion of constraints.

Once big stressed regions appear in the network, these regions undergo the *triangularization procedure* that consists in connecting two sites in the, region to all the rest and placing pebbles on these connections. This procedure shortens free pebble searches significantly. The results are not affected, since the regions are stressed and any constraint placed in them is redundant anyway.

At any stage, the network can also be decomposed into rigid clusters. A constraint not yet assigned to any cluster is picked and three pebbles are freed at its ends (the fourth pebble covers the constraint itself). Then the search for a free pebble starts from the constraint's ends outwards with the three free pebbles locked in place. The region to where any free pebble cannot move coincides with the rigid cluster to which the constraint belongs. Note that in 2D a constraint always belongs to just one rigid cluster, but a site can be shared between several rigid clusters, in which case it is a *pivot*. Several pivots are indicated in Figure 1.3 (see also Figure 4.5 in Chapter 4).

Typical results of the pebble game analysis for a piece of a randomly diluted triangular net are shown in Figure 1.4. The picture shows pivots that are shared between different rigid clusters and separate them from each other and also indicates stressed bonds.

1.2.3 Rigidity percolation in 2D

In this subsection, we review some results obtained for central-force generic rigidity percolation on the diluted triangular net obtained using the pebble game algorithm. A more detailed account can be found in Ref. [10].

We first compare the number of floppy modes to the Maxwell counting result,


Figure 1.4: Two typical fragments of a triangular network, below (left panel) and above (right panel) the rigidity transition. Green points are pivots between several rigid clusters. Black bonds are stressed, red bonds are unstressed. Below the transition, the network is floppy overall, but there are a few rigid pieces having no pivots. Above the transition, the network is rigid and stressed, but there are a few floppy pockets. Adapted from Ref. [6].

Eq. (1.4). This is shown in Figure 1.5. The exact value of f is indeed very close to the Maxwell value f_M far enough below the Maxwell counting estimate for the rigidity transition $\langle r \rangle_c^M = 4$, but then starts to deviate significantly and does not reach zero (until full coordination, $\langle r \rangle = 6$, is reached).

We now study rigidity percolation. Just as in usual connectivity percolation, we can choose as the relevant order parameter the size of the percolating rigid cluster (more specifically, we use the fraction of constraints in the percolating rigid cluster P_{∞}^{r}). Besides, as we might also be interested in *stress percolation*, we also use another order parameter, the fraction of constraints in the percolating stressed region P_{∞}^{s} . These two order parameters, as it turns out, become zero at the same point, so the rigidity and stress percolation thresholds coincide (Figure 1.6). This should be compared with the self-organization model (Chapter 5), where these thresholds do not coincide and there is an *intermediate phase* between them. Using



Figure 1.5: The number of floppy modes per degree of freedom f for the diluted triangular net. The black dashed line is the Maxwell counting result, Eq. (1.4). The inset shows the second derivative of f, with the data points for three different network sizes (lengths are indicated). The cusp at the rigidity transition is clearly seen. The inset is adapted from Ref. [5].



Figure 1.6: The fractions of bonds in the percolating rigid cluster (red open circles) and in the percolating stressed region (black filled circles) for the diluted triangular net. Averaged over two realizations on a 400×400 lattice.

finite-size scaling, the position of the threshold was found to be $\langle r \rangle = 3.961 \pm 0.002$. This is amazingly close to the Maxwell counting prediction $\langle r \rangle = 4$. As the order parameters seem to change continuously (rather than jump) at the threshold, the phase transition is apparently *second order*. It has been suggested by Duxbury and co-workers that the transition might be weakly first order; while we think this is unlikely, it cannot be completely ruled out at the present time.

The fraction of floppy modes f looks quite smooth, but the second derivative of it with respect to $\langle r \rangle$ (shown in the inset), in fact, exhibits a *cusp* at the rigidity threshold (Figure 1.5) This is similar to the behavior of specific heat at second-order thermal phase transitions. As the specific heat is minus the second derivative of the free energy with respect to the temperature, it was suggested that minus the number of floppy modes plays the role of the free energy in the rigidity percolation problem. This should be compared to connectivity percolation, which was mapped onto a particular case of the Potts model [11], so that the partition function can be written, and the free energy turns out to be the negative number of clusters; since the connectivity problem is equivalent to the 1D rigidity problem (see section 1.1), the number of floppy modes for connectivity is equal to the number of clusters. Thus analogy is clear, however, no similar mapping that would allow to write down the partition function and the free energy was proposed for the rigidity problem so far; the only positive indication is the proof for randomly bond diluted central-force networks [12] that -f has the right convexity properties for it to be the free energy, in other words, that the second derivative is non-negative.

Assuming that the transition is indeed second order and using the definitions of the order parameter and the analog of the specific heat, the standard set of critical exponents were evaluated, and the conclusion is that the rigidity transition is in a different universality class than connectivity.

1.2.4 Rigidity of 3D glass networks

Unfortunately, Laman's theorem cannot be generalized for arbitrary 3D networks. The necessary part is easily generalizable: indeed, if b > dn - d(d+1)/2for at least one subnetwork with $n \ge d$, there definitely are redundant constraints. The sufficient part, however, does not generalize. A detailed discussion of this can be found in Chapter 4; in that chapter, Figure 4.1 shows an example of a network that violates the straightforward generalization of Laman's theorem.

At the same time, Laman's theorem and with it, the pebble game, can be generalized for a particular class of 3D networks. Imagine an arbitrary network specified by a set of *bonds*, with constraints on both the bond *lengths* and *angles*. Or, equivalently, there are usual length constraints between first neighbors (i.e., sites directly connected by a bond) and also between all second neighbors, as illustrated by Figure 1.7. Constraints between first neighbors are commonly called



Figure 1.7: An example of a bond-bending network. There is an angular constraint between every two central-force constraints.

central-force constraints, while constraints between second neighbors are called *bond-bending* or *angular* constraints, since they fix angles between bonds. Such networks themselves are called *bond-bending networks*, and for them, Laman's theorem is generalizable in the form of the following statement [13, 14]:

A generic bond-bending network in 3D does not have a redundant constraint, if and only if no subset of the network containing $n \ge 2$ sites and b constraints violates $b \le 3n - 6$.

This is a part of the *molecular framework conjecture*; the full conjecture is in essence the statement that the 3D pebble game algorithm described below is exact for such a network. While the statement is only a conjecture and is not proved yet, no counterexamples were ever found in many years of extensive studies, so we find it reasonable to believe that the statement is correct and use it for analyzing rigidity of 3D bond-bending networks. Note, by the way, that rigidity of 2D bond-bending networks is equivalent to connectivity: whenever two sites are connected by a path of bonds, in the corresponding bond-bending network these two sites are mutually rigid.

The pebble game procedure for 3D bond-bending networks is similar to the 2D pebble game described in subsection 1.2.2, albeit more complicated [14, 15]. The

first obvious difference is that there are *three* pebbles per site, since each site now has three degrees of freedom. A very important feature is that the proper order of inserting constraints should be obeyed: a CF constraint is placed first, and all of the induced angular constraints (i.e., those constraints locking angles between the just inserted CF constraint and other CF constraints already in the network) should be placed immediately afterwards. This is done in order to ensure that the network is as close to being bond-bending at every step as possible. Failure to follow this rule leads to errors, as discussed in subsection 4.3.2.

Similar to the 2D pebble game, after inserting a constraint, we attempt to collect the maximum number of pebbles (six in 3D) at its ends. Unlike in the 2D case, this is *always* possible, since a rigid body has six degrees of freedom (for comparison, in 2D a rigid body has three degrees of freedom, so only three free pebbles out of four are guaranteed). So, obviously, the testing for redundancy is not over yet. What we need to check, according to the 3D generalization of Laman's theorem, is that every subnetwork including the new constraint and consisting of at least 3 sites has the number of constraints fewer than the number of degrees of freedom by at least six — then the new constraint is independent. This is true, if the seventh pebble can be freed at each site directly connected to both ends of the newly inserted constraint in turns, while keeping the six already freed pebbles free. In fact, fewer checks are needed [16]: for a CF constraint, only those sites connected to one of the ends by a CF constraint need to be checked; for an angular constraint, only the vertex of the angle has to be checked. If any of the checks fail, the constraint is redundant and the region of the failed pebble search is recorded as the stressed region and merged with the previously determined stressed regions if needed. Similarly to triangularization in the 2D case, *tetrahedralization* is performed to speed things up.

Just as in 2D, rigid cluster decomposition can also be done. Unlike in 2D, a

constraint can be shared between two rigid clusters (be a *hinge*, around which the two clusters rotate), so instead of starting with a single constraint and mapping its cluster, we start with an *angle* between two bonds. We free a total of six pebbles at the three sites forming the angle and then find the region where the seventh pebble cannot be found, and this region is the rigid cluster.

1.2.5 Covalent glass networks

The 3D rigidity theory can be applied to *network glasses*, which are non-metallic glasses with covalent bonding between atoms. On the microscopic level, these glasses are modeled as the *continuous random network* [17]. In this model, the structure is viewed as a network of bonds between atoms that is topologically disordered (i.e., cannot be continuously deformed into an ordered crystal) and, as implied by the word "continuous", has no macro- or mesoscopic voids and thus is about as dense as the corresponding crystal (see Figure 1.8). This has become a well-established model mostly through extensive diffraction studies [18].

The most important forces between atoms in glass networks are nearest-neighbor bond-stretching and angular bond-bending forces. Other, weaker forces (dihedral, van der Waals, etc.) can often be neglected. What is obtained then is a bond-bending network whose rigidity can be studied using methods described here; in particular, luckily, the 3D pebble game can be applied.

Following Thorpe [19], we consider chalcogenide compounds of the type $Ge_x As_y Se_{1-x-y}$, where Ge stands for any atom that has valence four and thus bonded to four other atoms (has coordination number 4), As stands for any atom with coordination 3 and Se any atom with coordination 2. Varying x and y, we can change the average number of constraints per atom and thus the rigidity properties of the network.

We first do Maxwell counting for the networks under consideration. First, we



Figure 1.8: A schematic drawing of a covalent glass network. There are atoms of valence (coordination) 2 (blue), 3 (green) and 4 (red).

count all constraints. The number of bond-stretching constraints is simply the number of bonds, which is $\langle r \rangle N/2$, where $\langle r \rangle$ is mean coordination (now defined in terms of the number of bonds and not the total number of constraints). For a site of coordination r, the total number of bond-bending constraints is r(r-1)/2, which gives 1 for r = 2, 3 for r = 3 and 6 for r = 4. However, for r = 4, one of these constraints is always redundant, and it is reasonable to exclude it; so the number of remaining angular constraints for r = 2, 3 and 4 is 1, 3 and 5, respectively, or 2r - 3. Then, the total number of constraints is

$$N_c = \sum_{r=2}^{4} n_r \left[r/2 + (2r - 3) \right], \qquad (1.5)$$

where n_r is the number of sites of coordination r, or, using $\langle r \rangle = \sum_{r=2}^4 n_r r/N$,

$$N_c = N\left(\frac{5}{2}\langle r \rangle - 3\right). \tag{1.6}$$

The number of constraints per degree of freedom is

$$n_c = \frac{N_c}{3N} = \frac{5}{6} \langle r \rangle - 1, \qquad (1.7)$$

and, according to Eq. (1.2), the Maxwell counting result is

$$f = 2 - \frac{5}{6} \langle r \rangle. \tag{1.8}$$

Just as in the 2D case, we associate the rigidity transition with the point where f becomes zero; thus the Maxwell counting prediction for the rigidity threshold is $\langle r \rangle_c^M = 2.4.$

Note that f in the Maxwell counting approximation only depends on $\langle r \rangle = 2 + 2x + y$ (x and y are the parameters from the chemical formula $Ge_x As_y Se_{1-x-y}$ and not on x and y separately. Likewise, the mean coordination at the threshold is constant. This is because the number of angular constraints associated with an atom depends linearly on the coordination of the atom. We note that if there are also onefold-coordinated atoms (such as halogen atoms) in the network, the linear formula for the number of angular constraints (2r - 3) no longer applies, giving an absurd result -1 (there are, in fact, no angular constraints, so the answer should have been zero). Because of this, the threshold obtained by Maxwell counting is no longer a constant, but depends on the concentration of 1-coordinated sites n_1 [20]:

$$\langle r \rangle_c^M = 2.4 - 0.4n_1. \tag{1.9}$$

We subject this equation to some scrutiny in Chapter 2.

We now return to the case when there are no 1-coordinated sites and describe the pebble game analysis [6]. The networks were constructed by starting from a network with all sites 4-coordinated and then decreasing the mean coordination by selecting bonds at random and removing each selected bond, *provided that* no 1-coordinated sites are created. This ensures that the network always consists of sites of coordination 2, 3 and 4 only. The advantage of this procedure is that the whole sequence of networks can be analyzed in a single pebble game run, by starting from the network with the lowest coordination and then placing bonds in the order reverse to that in which they were removed, thus recreating the whole sequence, and running the pebble game concurrently. Of course, the results should be averaged over several realizations.

As the starting fully 4-coordinated network, the amorphous silicon network constructed using the Wooten-Winer-Weaire (WWW) procedure [21, 22] was used. The WWW method starts with the crystalline diamond lattice and amorphizes it by repeatedly applying a bond-switching procedure that changes two neighboring 6-fold rings into a 5-fold and a 7-fold rings or vice versa, while keeping the coordination of all atoms at 4 all the time. After the initial amorphization, relaxation is performed by carrying out the same bond-switching procedure, but accepting or rejecting each bond switching with the appropriate Boltzmann weight.

Besides the amorphous silicon network, the crystalline diamond lattice was also used for rigidity studies. The latter is topologically ordered, but becomes disordered very fast upon dilution and the results for the two starting networks do not differ much.

The results of the pebble game study are presented in Figures 1.9 and 1.10. Just as in the 2D central-force case, the rigidity and stress thresholds coincide and the transition is second order. The threshold is located at $\langle r \rangle = 2.375 \pm 0.003$ and $\langle r \rangle = 2.385 \pm 0.003$ for the diamond lattice and the amorphous silicon network, respectively, which is, remarkably, within about 1% of the Maxwell counting result $\langle r \rangle = 2.4$.

1.2.6 Body-bar representation

It turns out that 3D bond-bending networks can be represented as equivalent central-force *body-bar* networks [23, 24]. In this representation, each site with coordination higher than 2 of the original 3D network is represented as a *body* having 6 degrees of freedom and every bond (i.e., every nearest-neighbor constraint) is represented as a bunch of 5 *bars* (i.e., 5 constraints). This correspondence is illustrated in Figure 1.11. Sites of coordination 1 should be represented as having 5 degrees of freedom (since, say, a *dimer* consisting of two connected 1-coordinated sites should have no internal degrees of freedom) and isolated sites as having 3 degrees of freedom.

The body-bar representation has several important advantages over the original



Figure 1.9: The number of floppy modes per degree of freedom f for the diluted amorphous silicon network (red circles) and the diluted diamond lattice (green diamonds). The dashed black line is the Maxwell counting result, Eq. (1.8). The inset shows the second derivative of f, with the cusp at the transition clearly seen in both cases. Adapted from Ref. [6].



Figure 1.10: The fractions of sites in the percolating rigid cluster (open red circles) and in the percolating stressed region (filled black circles) for the diluted diamond lattice. The results are averages over 11 realizations on networks of 125000 sites. The rounding near the transition is due to finite-size effects.



Figure 1.11: The correspondence between bond-bending networks (top) and body-bar networks (bottom). The bond-bending network has one internal motion - a rotation around a hinge. Likewise, a system of two bodies with 6 degrees of freedom each connected by 5 bars has one internal degree of freedom.

one. For example, Laman's theorem can also be generalized to body-bar networks, and the corresponding "6-dimensional" version of the pebble game is more straightforward than the 3D pebble game described above: in fact, it is totally analogous to the 2D pebble game procedure, the only obvious differences being that there are now 6, 5 or 3 pebbles tethered to each site (depending on the site's coordination) instead of 2 pebbles in the 2D case, and that 5 pebbles are placed on a bond instead of one. Also, the fact that there are now only nearest-neighbor connections makes many considerations simpler, and we take advantage of this in Chapters 2 and 3.

1.3 Applications of rigidity theory

1.3.1 Network glasses

Initially interest in applications of rigidity theory has been present mostly in the engineering community. Engineers are interested in mechanical stability of various structures, and rigidity theory certainly is relevant. Recently, however, the real boost to development of rigidity theory came from those interested in applications at the microscopic level. Historically, the first such application has been to covalent network glasses.

The first application of the constraint counting idea to glasses dates back to 1979. J.C. Phillips [25] was interested in what makes a particular material a good or a bad *glassformer*. Essentially all materials can be made amorphous if cooled from the melt fast enough, so the atoms do not have enough time to find the correct positions and arrange themselves into a crystal. However, some materials (most metals, for example) are very bad glassformers: they almost always crystallize upon cooling and only when cooled very fast can they freeze into a disordered amorphous solid. On the other hand, there are very good glassformers (such as the ordinary

window glass) that almost always form amorphous solids and have to be cooled extremely slowly to crystallize. All obvious explanations for what causes different glassforming ability failed to account for all cases. For example, one might think that materials with simpler crystal structures are worse glassformers, as their structures are easy for the atoms to arrange into. But there are very complex materials that easily crystallize; on the other hand, the window glass is mostly just the silicon oxide, SiO_2 , and the structure of its crystalline counterpart, quartz, is not particularly complex. Phillips noted that in a particular class of materials, namely, those with predominantly covalent bonding, the glassforming ability is best for those materials that have the number of constraints (defined as in subsection 1.2.5, i.e., first-neighbor central-force and second-neighbor angular constraints are counted) approximately balances the number of degrees of freedom of all the constituent atoms. The explanation he gave is that when the number of constraints is bigger, the network is overconstrained, so, when disordered, it has too high energy that is reduced by ordering; on the other hand, when the number of constraints is smaller, the network is flexible, so it can easily rearrange into a crystal without having to overcome large potential energy barriers; in both cases, crystals are easily formed, and only when there are neither too many nor too few constraints, glasses are formed instead.

Note that the difference between the number of degrees of freedom and the number of constraints that plays the pivotal role in Phillips' theory is the number of floppy modes in the Maxwell counting approximation; this difference, as we now know, becomes zero close to the rigidity percolation threshold. Thus we might say that according to Phillips' ideas, best glassformers are at or near the percolation threshold. This was noticed in 1983 by M.F. Thorpe [19], who introduced the concept of rigidity percolation and proposed that it should show up as a threshold in various mechanical and related properties when they are measured as a function



Figure 1.12: The frequency of a particular vibrational mode in a Ge-S glass measured by Raman scattering as a function of the mean coordination number. Different mean coordination numbers are obtained by varying the composition. A break is seen near $\langle r \rangle = 2.4$. Adapted from Ref. [26].

of the number of constraints or the mean coordination by varying the chemical composition.

Note that the Phillips' proposal deals with an optimum of a quantity, whereas Thorpe's suggestion is about a threshold. Optima are generally expected to be rather robust against presence of factors not taken into account by a theory, such as weaker interactions (van der Waals forces, etc.) in this case. Indeed, extrema in many physical quantities (not just the glass-forming ability) have been observed close to the rigidity percolation point $\langle r \rangle = 2.4$ [27, 28, 29, 30]. Thresholds, on the other hand, are more easily smeared away and have been harder to observe. Nevertheless, they have been seen, say, in Raman scattering measurements (Figure 1.12) [26].

1.3.2 Proteins

More recently, another interesting and potentially very promising application of rigidity has emerged. Protein molecules, from the physical point of view, are covalent networks, like glasses, just much smaller in size, usually thousands of atoms. Proteins play an extremely important role in functioning of all live organisms. They perform a huge variety of functions, from simple mechanical reinforcement to taking part in and catalyzing chemical reactions. In many of these functions, mechanical properties of proteins play a big role. To understand proteins' functions and control them, say, to cure diseases, it is very important to know what motions of what parts of a protein molecule are possible. This is an information that rigidity theory can give. For example, in Figure 1.13, the *rigid cluster* decomposition of a protein called HIV protease [31] is shown. Different adjacent rigid clusters have different colors. It is seen that much of the molecule is a big rigid cluster; however, flaps at the top of the picture consist of many small clusters and thus are flexible. These flaps can easily move and thus the protein performs its function that consists in cutting the DNA of the HIV virus that causes AIDS and thus multiplying the virus. Some anti-AIDS drugs work by binding to the protein and locking the flaps. This can be analyzed using rigidity theory.

Another application is to the protein folding problem [32]. Proteins are synthesized as a polymer chain that then folds itself into a complex three-dimensional structure unique for each protein. The essence of the protein folding problem is prediction of the structure of the protein, as well as the pathway in the configuration space that the protein takes to reach this final structure. Rigidity theory can help in predicting the pathway: simply speaking, the most rigid parts of the protein fold first, although the actual approach is more complicated (see [33] for details).



Figure 1.13: The rigid cluster decomposition for the HIV protease. Only the protein backbone is shown and not the side chains. There is one big rigid cluster (blue), but the flaps consist of many small clusters and thus are flexible and can easily move.

Chapter 2: Random Bond Model

The pebble game algorithm described in Chapter 1 can in principle be used to analyze rigidity of any network in 2D and any bond-bending network in 3D. Of course, just as with any numerical procedure, these networks have to be finite and in practice, there is always a limit on the size of those networks that can be analyzed in reasonable time. So, if our purpose is studying rigidity in the thermodynamic limit, there are always unavoidable finite-size effects, which can bring about controversies even in questions as basic as the order of the rigidity transition (see the discussion of the 2D rigidity transition in Chapter 1). It is therefore of interest that there is a class of networks, rigidity of which can be studied analytically in the thermodynamic limit. These are so-called *random-bond networks* (RBNs).

An example of an RBN is shown in Figure 2.1 In an RBN, sites placed randomly in space are connected at random regardless of Euclidean distances between them. Because of this property, there are virtually no finite-size loops in these networks, although there are loops of typical size $O(\ln N)$ (where N is the number of sites) [34]. If one starts with one site of an infinite RBN and then includes its first, second, etc. neighbors (in fact, any finite number of coordination shells), one ends up with a *tree*. In other words, *locally* an RBN is a tree, where the word "locally" should be understood in terms of *chemical* distances (along bonds), rather than Euclidean distances. It is not surprising then that for many calculations, we will be able to consider RBNs as infinite trees (or *Bethe lattices*); ultimately, however, the fact that there are infinite loops in RBNs (that distinguish them from Bethe lattices) is very important. We will refer to the RBN model of rigidity as the it random bond model (RBM).

At this point, a few words about terminology are in order, as there is some confusion in the literature. In this dissertation, a network with infinite loops (as



Figure 2.1: A random bond network. Sites are connected at random, regardless of the distance.

schematically depicted in Figure 2.1) is called an RBN; an infinite tree without any loops is called a Bethe lattice. Beware that in some papers, RBNs are called Bethe lattices and trees (both finite and infinite) are called Cayley trees (the term that we will not use here).

Studying various problems on Bethe lattices has a long history. The term "Bethe lattice" itself derives from the Bethe-Peierls approximation in the theory of substitutional alloys [35, 36] that was later shown to be exact [37] on Bethe lattices. Since then, a great variety of problems were considered. These include: connectivity percolation [38], the Ising model (equivalent to the original alloy problem) and Potts models in general [39, 40], tight-binding Hamiltonians [41], spin glasses [42], the random field Ising model [43, 44, 45], and others. A good review of early applications is in Ref. [46].

Rigidity percolation on bond- and site-diluted Bethe lattices and RBNs was first studied by Duxbury, Thorpe and others [47, 48, 12]. In this chapter, we apply their method to 3D bond-bending networks; having in mind applications to covalent glasses, instead of bond- or site-diluted networks we consider RBNs having prescribed fractions of sites of different coordinations, as well as varying degrees of chemical order. In section 2.1, we briefly review the method, as applied previously to bond-diluted Bethe lattices. In section 2.2, we describe our approach pointing out where it differs from previous studies, write down the basic equations and solve these equations obtaining rigidity phase diagrams. In section 2.3, we show how presence of 1-fold coordinated sites in the network can be easily taken into account. We end with conclusions in section 2.4.

Most results presented in this chapter were published in Ref. [6].

2.1 Rigidity percolation on Bethe lattices and RBNs

This review section is based on Ref. [12].

We consider trees built in the following way (Figure 2.2). Start with a site (the *root* of the tree) and connect it to z other sites (where integer z is the coordination of the tree). Connect each of these z sites (first generation) to z - 1 other sites (but not the root), so there are z(z - 1) sites in the second generation. Connect each of the sites in the second generation to z - 1 sites not in the previous or current generations, and so on. Suppose we build a finite tree containing n generations of

sites. We will call sites of the last *n*th generation *leaves* of the tree. All sites of the tree are z-fold coordinated, except for the leaves. The "thermodynamic limit" can be considered by taking *n* to infinity, then we end up with a Bethe lattice. Note that the thermodynamic limit is rather pathological, as a finite fraction of sites are leaves, which are undercoordinated surface sites. Trees built as described can then undergo random bond dilution by removing each bond with probability 1 - p and retaining it with probability p.



Figure 2.2: A 3-fold coordinated tree. The root is red, the leaves are green.

Thinking about rigidity of trees, we soon notice that we seem to be in trouble, as stand-alone trees can never be rigid. Imagine, however, that we attach all leaves of the tree to a rigid *busbar* (Figure 2.3), so that all leaves are rigid with respect to the busbar and thus with respect to each other. Then we can ask if, at a given p, rigidity due to the busbar can propagate infinitely far from the busbar towards the root. This is how the rigidity percolation problem on Bethe lattices can be posed. RBNs, on the other hand, can be rigid by themselves due to the presence of rigidifying infinite loops. This is one reason to prefer RBNs to Bethe lattices when studying the rigidity transition; another (and a more serious one) is the pathologically big surfaces of Bethe lattices (as mentioned above) that are absent in RBNs. On the other hand, it is somewhat easier to think in terms of Bethe lattices, rather than RBNs, when constructing equations of the theory. For that reason, we will start with the theory of Bethe lattices and then point out where differences between them and RBNs matter.



Figure 2.3: A tree connected to a busbar. Present bonds are solid lines, missing bonds are dotted lines. Sites in level 0 (green) are rigidly attached to the busbar.

The method used to solve the rigidity problem on Bethe lattices is a variation of the *transfer matrix technique*. To illustrate the method, consider a z-coordinated Bethe lattice with a busbar in 2D. Fix the busbar so it cannot move. Now consider just the first n levels of the lattice (counting from the busbar) cutting them off from the rest (i.e., removing all bonds connecting the nth level to the (n + 1)th level). Consider a site in the nth level. The site can be rigid with respect to the busbar, then it has zero degrees of freedom (d.o.f.). Here and elsewhere in this chapter, we count degrees of freedom corresponding to zero-frequency motions only, unless specified otherwise. If the site is not rigid, it has either one or two d.o.f. Denote $T_{i,n}$ the probability that a site in level n has i d.o.f. (i = 0, 1 or 2). Now repeat the same procedure for n + 1 levels of the lattice, cutting them off from the rest. Since sites in the (n + 1)th level are only connected to sites in the nth level, it is possible to express $T_{i,n+1}$ in terms of $T_{i,n}$. A site in level (n + 1) is connected to up to z - 1sites in level n (the probability of each connection is p). If at least two of these sites have 0 d.o.f., the site in level n + 1 also has 0 d.o.f. The probability that a given connection is present is p; the probability that it is present and leads to a site with 0 d.o.f. is $pT_{0,n}$. Then

$$T_{0,n+1} = 1 - (z-1)pT_{0,n}(1-pT_{0,n})^{z-2} - (1-pT_{0,n})^{z-1}.$$
(2.1)

Since we want to find stationary solutions (or values of $T_{0,n}$ far away from the busbar), we equate $T_{0,n+1} = T_{0,n} \equiv T_0$, then

$$T_0 = 1 - (z - 1)pT_0(1 - pT_0)^{z-2} - (1 - pT_0)^{z-1}.$$
 (2.2)

Similarly we can write equations for T_1 and T_2 , when we need these quantities. For $0 \le p \le 1$, equation (2.2) has either 1 or 3 solutions in the interval $0 \le T_0 \le 1$. There is always the trivial solution $T_0 = 0$, but if $z \ge 4$, there exists p_s such that for $p > p_s$ there are also two non-trivial solutions (Figure 2.4). When considered as stationary solutions of Eq. (2.1), the solutions can be stable or unstable. The trivial solution and the bigger of the non-trivial solutions (when it exists) are always stable; the smaller of the non-trivial solutions (when it exists) is always unstable. Then there are 1 or 2 stable solutions. When 2 stable solutions exist, the choice between them is determined by the initial condition. With rigidity induced by the busbar, the initial condition is $T_{0,0} = 1$, as all sites at the busbar (level 0) are rigid with respect to the busbar. Then the non-trivial solution is realized whenever it exists, i.e., for $p > p_s$.



Figure 2.4: Solutions of equation (2.2) for the probability that a site has zero degrees of freedom, T_0 . There is always a trivial solution identically equal to zero (green). There may also be non-trivial solutions (red), stable (solid line) and unstable (dashed line). The black line shows the solution that is actually realized in RBNs.

The probability for a bond far away from the busbar to belong to the infinite rigid cluster starting at the busbar can be shown to equal $T_0^2 + 2T_0T_1$. Then, whenever the trivial solution $T_0 = 0$ is realized, there is no infinite rigid cluster and the network is floppy; whenever the non-trivial solution is realized, the network is rigid. Then in a Bethe lattice with a busbar, the rigidity transition occurs at $p = p_s$.

How do these results apply to the RBM? Since locally an RBN is a tree, we can still write the same equation (2.2). This can be interpreted as follows. Suppose we define some reference frame and for each site define quantities T_i that are the

probabilities for the site to have i d.o.f. with respect to the chosen frame, when one of the z possible bonds stemming out of the site is chosen arbitrarily and removed if present. In an RBN, T_i are the same for all sites. Then one can write a self-consistency equation by writing T_0 for a site in terms of the same T_0 for its neighbors, and this equation will coincide with Eq. (2.2). The meaning of the solutions of this equation is now as follows. If only the trivial solution exists, then there is no reference frame with respect to which a finite fraction of the network sites are rigid. If there is also a non-trivial solution, there is a possibility that there might be a reference frame, with respect to which a finite fraction of the network is rigid. If such a frame exists, we will say that the non-trivial solution is realized; if it does not exist, the trivial solution is realized (even if the non-trivial solution exists). Note that while the equations are the same as for the Bethe lattice, criteria of choosing between the trivial and the non-trivial solutions may now be different. We will not describe the method of choosing the right solution here, given that the consideration for the 3D bond-bending networks later in this chapter is fully analogous (if a bit more complicated technically); see also Ref. [12]. The final result is indeed different from the Bethe lattice case (no surprise, given that the Bethe lattice result depended on the presence of the busbar, and there is no such thing as the busbar in the RBN). It turns out that switching between the trivial and non-trivial solutions occurs not at $p = p_s$, but at a higher $p = p_c$. Thus, there is a spinodal point p_s , at which the non-trivial solution first appears and a critical point p_c , at which the rigidity transition occurs. This is the standard picture of a *first-order phase* transition. Note also that T_0 , which is related to the size of the percolating cluster and which thus can serve as the *order parameter* for the rigidity transition, jumps at p_c , as it should when undergoing a first-order transition. This is in contrast with connectivity percolation, as well as rigidity percolation on usual, regular lattices (having finite loops, unlike RBN), where this transition is second order.

Other Bethe lattice models were also considered [48], such as site-diluted problems and also more general cases of *body-bar* networks, where sites are replaced by *bodies* having an arbitrary number of degrees of freedom and connected by *bars* consisting of several constraints. The 3D bond-bending networks considered in what follows will also be represented as body-bar networks, but the model is somewhat different from those considered before.

2.2 The random bond model of glass networks

Randomly bond-diluted Bethe lattices were considered before, as we have just described. These are characterized by a single parameter p, the fraction of bonds remaining. Considering glass networks, though, we usually have a certain composition, where there is a *prescribed* number of atoms of each coordination. Besides, in glass networks angular forces are also significant. So, when constructing Bethe lattice or random bond models of glass networks, we should add second-neighbor constraints to trees (or, respectively, RBNs), so that bond-bending networks are formed. An example of a tree with second-neighbor constraints is shown in Figure 2.5.

As we have seen in Chapter 1, for a three-dimensional network with angular constraints, the rigidity transition occurs at the average coordination $\langle r \rangle_c = 2.4$ according to Maxwell counting (and the exact threshold is close to this value for randomly diluted networks). This assumes that there are no sites of coordination 0 or 1. Then the simplest possibility for studying the rigidity transition is a network with just two types of sites, 2-coordinated and *i*-coordinated, where i > 2. In what follows, we consider the case i = 3 in detail. Thus we have 2- and 3-coordinated sites (for brevity, 2- and 3-sites) in the network. If the concentration of 3-sites is x, then that of 2-sites is 1 - x and the average coordination is $\langle r \rangle = 2 + x$. We also



Figure 2.5: A bond-bending tree. Present bonds are solid red lines, missing bonds are dotted red lines. There are angular constraints (blue dashed lines) between every pair of present bonds stemming from the same site.

give the final results for i = 4 (networks with 2- and 4-coordinated sites).

2.2.1 Chemical order

In addition to changing the average coordination by changing the concentration of 3-sites x, we can also change *chemical order* in the network. For example, a site of a given type may tend to be preferentially connected to a site of the *same* or *different* type. Actual situations may range from complete phase separation to perfect chemical order where there is always an atom of the opposite type between two atoms of the same type.

To describe these possible situations, we consider the numbers of bonds, N_{ij} , connecting atoms of types *i* and *j* (*i*-*j* bonds for brevity). We have 3 quantities:

 N_{22} , N_{33} and N_{23} . Twice the number of 2-2 bonds plus the number of 2-3 bonds gives the total number of bonds stemming from 2-sites (which is twice the number of 2-sites N_2); if a bond is a 2-2 bond, it is counted twice. The similar sum rule is for bonds going out of 3-sites (the number of such bonds is $3N_3$, where N_3 is the number of 3-sites). Then we have

$$2N_{22} + N_{23} = 2N_2,$$

$$2N_{33} + N_{23} = 3N_3.$$
 (2.3)

Since N_2 and N_3 can be expressed in terms of the concentration x and the total number of sites N $[N_3 = xN, N_2 = (1 - x)N]$, only one of the three quantities N_{ij} is independent.

The total number of bonds is $N_B = N_{22} + N_{33} + N_{23}$. It is convenient to use *probabilities* for bonds of different types y_{22} , y_{33} and y_{23} defined as

$$y_{ij} = \frac{N_{ij}}{N_B} \tag{2.4}$$

instead of N_{ij} . By definition, $y_{ij} = y_{ji}$. Dividing (2.3) by N_B and using

$$N_B = \frac{\langle r \rangle N}{2} = \frac{2+x}{2}N, \qquad (2.5)$$

we obtain relations for y_{ij} :

$$2y_{22} + y_{23} = \frac{4(1-x)}{2+x},$$

$$2y_{33} + y_{23} = \frac{6x}{2+x},$$
(2.6)

so that again we have two equations for 3 quantities, and these quantities can be expressed in terms of a single parameter. It is convenient to use the correlation parameter θ such that

$$y_{23} = 2\theta, \tag{2.7}$$

then

$$y_{22} = \frac{2(1-x)}{2+x} - \theta,$$

$$y_{33} = \frac{3x}{2+x} - \theta.$$
(2.8)

The obvious sum rule on the probabilities

$$y_{22} + y_{23} + y_{33} = 1 \tag{2.9}$$

is satisfied.

The parameter θ characterizes the chemical bonding properties of the network, or, in other words, correlations between types of first-neighbor sites. We will assume that there are no longer-range correlations, i.e., no correlations between types of second, third, etc., neighbors. Then statistically, the networks are completely characterized by two parameters, x and θ .

It is also convenient to introduce another group of quantities. We define $n_j^{(i)}$ as the probability for a bond known to start at a site of coordination *i* to end at a site of coordination *j*. The obvious sum rules are

$$\sum_{j} n_{j}^{(i)} = 1 \text{ for all } i.$$
 (2.10)

The total number of bonds stemming out of *i*-sites is $(2y_{ii} + \sum_{j \neq i} y_{ij})N_B$, of them

 $2y_{ii}N_B$ go to an *i*-site, $y_{ij}N_B$ go to a *j*-site $(j \neq i)$, then

$$n_{i}^{(i)} = \frac{2y_{ii}}{2y_{ii} + \sum_{k \neq i} y_{ik}},$$

$$n_{j}^{(i)} = \frac{y_{ij}}{2y_{ii} + \sum_{k \neq i} y_{ik}}, \quad j \neq i.$$
(2.11)

Equations (2.10) and (2.11) apply to networks with an arbitrary number of atom types. In our case of 2-3 networks, we have

$$n_{3}^{(2)} = \frac{y_{23}}{2y_{22} + y_{23}} = \frac{\theta(2+x)}{2(1-x)},$$

$$n_{2}^{(2)} = 1 - n_{3}^{(2)},$$

$$n_{2}^{(3)} = \frac{y_{23}}{2y_{33} + y_{23}} = \frac{\theta(2+x)}{3x},$$

$$n_{3}^{(3)} = 1 - n_{2}^{(3)}.$$
(2.12)

Obviously, $0 \le n_j^{(i)} \le 1$ for all *i* and *j*. This determines the range of allowed values of parameters x and θ . Namely, from $n_3^{(3)} \ge 0$,

$$\theta \le \frac{3x}{2+x};\tag{2.13}$$

from $n_2^{(2)} \ge 0$,

$$\theta \le \frac{2(1-x)}{2+x}.\tag{2.14}$$

Thus, in the x- θ plane, the set of allowed pairs of parameters is bounded by the $\theta = 0$ line and two hyperbolas [left boundary (2.13) and right boundary (2.14)], as shown in Figure 2.6 (where instead of θ , we use $y_{23} = 2\theta$).

At $\theta = 0$ ($y_{23} = 0$) there are no 2-3 bonds, thus there is complete phase separation (two separate networks, one 2-coordinated and another 3-coordinated). The maximum value of $\theta = 0.5$ ($y_{23} = 1$) is possible only at x = 0.4; this



Figure 2.6: The allowed region (white) in the plane of two parameters, the mean coordination $\langle r \rangle$ and the fraction of 2-3 bonds among all bonds y_{23} characterizing chemical order. Line 1 is the left boundary [no 3-3 bonds; Eq. (2.13)] and line 2 is the right boundary [no 2-2 bonds; Eq. (2.14) of the allowed region. Line 3 corresponds to networks obtained by random dilution [Eq. (2.20)]. Line 4 corresponds to randomly bonded networks [Eq. (2.15)]. The upper corner of the allowed region represents perfectly chemically ordered networks; the bottom represents completely phase separated networks.

corresponds to having only 2-3 bonds in the network, thus perfect chemical order, where every 2-site has only 3-sites as neighbors and vice versa.

Between these two extremes lies random bonding, where sites are linked up in random fashion, without regard for their coordinations. Then probabilities for a bond to go to a site of a certain type do not depend on the type of the site where the bond started; formally, this means that quantities $n_j^{(i)}$ do not depend on *i*, so $n_2^{(2)} = n_2^{(3)}, n_3^{(2)} = n_3^{(3)}$. From Eqs. (2.12) we then get

$$\theta = \frac{6x(1-x)}{(2+x)^2}.$$
(2.15)

Another case of interest is *random dilution*. When studying rigidity of glass networks with the pebble game in Chapter 1, we constructed sequences of networks by starting with a full lattice (with all sites equally coordinated) and then removing bonds at random with the restriction that no sites of coordination lower than 2 appear. The advantage of this procedure is that the whole sequence can be analyzed in a single pebble game run. We can implement the same procedure for RBNs by starting from a fully 3-coordinated RBN and then removing bonds at random not allowing 1-coordinated sites, as described above. Such sequences are convenient for comparison with the pebble game.

In our case, the random dilution procedure implies that we start from a fully 3-coordinated network (thus x = 1 at the beginning) and then remove 3-3 bonds only, one by one. Suppose we remove ΔN_B 3-3 bonds (ΔN_B is presumed to be small compared to the total number of bonds N_B). From (2.5),

$$\Delta N_B = \frac{N}{2} \Delta x. \tag{2.16}$$

When we remove a 3-3 bond, the sites that were connected by this bond are converted to 2-coordinated from 3-coordinated. So if these sites were connected with other 3-sites, these connections will convert from 3-3 type to 2-3 type and the number of 2-3 bonds will increase. On the other hand, if these sites were connected to 2-sites, the bonds will be converted to 2-2 from 2-3 and the number of 2-3 bonds will decrease. Then the change in the number of 2-3 bonds is

$$\Delta N_{23} = 4(n_3^{(3)} - n_2^{(3)})\Delta N_B = 4\left(1 - \frac{2\theta(2+x)}{3x}\right)\Delta N_B,$$
(2.17)

The factor of 4 is because there are 4 untouched bonds going out of the two 3-sites between which the (fifth) bond is removed. On the other hand, using (2.4), (2.5) and (2.7), $N_{23} = 2\theta N_B = \theta(2+x)N$, and since N = const,

$$\Delta N_{23} = (2+x)N\Delta\theta + N\theta\Delta x. \tag{2.18}$$

Using (2.16), (2.17) and (2.18), we obtain a differential equation

$$(2+x)\frac{d\theta}{dx} = -2 + \frac{8+x}{3x}\theta.$$
(2.19)

The starting network is fully 3-coordinated, thus, obviously, it has no 2-3 bonds; so, when x = 1, $\theta = 0$. This is the initial condition for (2.19) and the solution is

$$\theta = \frac{6x(1-x^{1/3})}{2+x}.$$
(2.20)

Note that the random dilution result (2.20) is different from the random bonding result (2.15), so that randomly diluted networks are not perfectly randomly bonded. This is because of the restriction that does not let 1-coordinated sites appear. Because of this restriction, dilution is not *truly* random; if there was no such restriction, a randomly diluted network would be randomly bonded (but 0- and 1-coordinated sites would appear, of course). It turns out that higher-order correlations also appear in randomly diluted networks. This is too cumbersome to show analytically, but can be checked numerically. Since our theory assumes that such correlations are absent, it is only approximate when applied to randomly diluted networks. However, it works remarkably well, as we will see.

2.2.2 Transfer matrix equations

Now we want to apply the variation of the transfer matrix method described in section 2.1 to our problem. Note, however, that since we have bond-bending networks, second neighbors in the tree are also connected. Then level n + 1 in the tree is connected not only to level n, but also to level n - 1, as well as to sites in its own level. This would make equations for quantities T_i significantly more complicated than in the 2D central-force case. Fortunately, we recall that 3D bond-bending networks can be represented as *body-bar networks* with sites replaced by *bodies* with 6 degrees of freedom, with first-neighbor constraints replaced by *bars* consisting of 5 constraints each and *no* second-neighbor constraints. Then if the tree is cut above level n + 1, this level is connected to level n only, and writing down the equations is easier.

We need to build a tree that locally would be the same as an RBN we want to consider. Our tree will look as shown in Figure 2.7. There are some 2-sites and some 3-sites, and the probability that a bond starting at a site of coordination igoes to a site of coordination j should be given by $n_j^{(i)}$. Note that in the representation of the Bethe lattice that we use here (as shown in Figure 2.7), one of the bonds going out of every site always points up (away from the busbar); this is different from the consideration of 2D Bethe lattices in section 2.1, where a bond pointing up was only present with probability p. When we write equations for T_i , we cut bonds connecting level n + 1 with the next level n + 2; this means that in the
consideration in section 2.1, one bond stemming from a site was removed with probability p, whereas now, one bond will be removed with probability 1. Clearly then, the definitions of quantities T_i are now different; however, this is just a matter of definition and the final results for the number of floppy modes, the location of the transition, etc. will not depend on these details.



Figure 2.7: A tree of bodies and bars. Each bond consists of 5 bars; each site is a body having 6 degrees of freedom (when isolated). There are two types of bodies, 2-and 3-coordinated. In the representation we use there is always one bond pointing upwards from every site, regardless of its coordination.

One apparent paradox worth mentioning is that, as it turns out, if we follow the rule that the probabilities of bonds are equal to $n_j^{(i)}$, as described above, the tree built in such a way will have a "wrong" concentration of sites of different types, i.e., the concentration of 3-sites will not be x. But this is fine, given that the tree should only be identical to the RBN locally, and there is a pathologically big undercoordinated surface in any tree anyway. This paradox implies that interestingly, in any RBN, the concentration of sites of a particular type in a finite region around any specific site is always different than in the whole infinite network.

Now we are almost ready to start writing down the actual equations. First of all, since we have sites of two types, we define quantities T_i for each of these types separately. That is, we introduce quantities $T_{i(j)}$, which are probabilities for a *j*-coordinated site to have *i* d.o.f. with respect to the busbar, when the site is cut from the level above. Obviously, *i* can range from 0 to 6 (since our sites are now bodies with 6 d.o.f.); *j* can be 2 or 3.

Consider some 2-coordinated site (say site 1 in Figure 2.8); when it is cut off from the upper level, it remains connected to just one site below (site 2 in Figure 2.8). Site 1 has 6 d.o.f., if site 2 has 5 or 6 d.o.f.; site 1 has *i* d.o.f. (for i < 6), if site 2 has i - 1 d.o.f. Note that a 2-site can never have 0 d.o.f. Then the corresponding equations are

$$\begin{split} T_{6(2)} &= n_2^{(2)} X_{(2)} + n_3^{(2)} X_{(3)}, \\ T_{5(2)} &= n_2^{(2)} T_{4(2)} + n_3^{(2)} T_{4(3)}, \\ T_{4(2)} &= n_2^{(2)} T_{3(2)} + n_3^{(2)} T_{3(3)}, \\ T_{3(2)} &= n_2^{(2)} T_{2(2)} + n_3^{(2)} T_{2(3)}, \\ T_{2(2)} &= n_2^{(2)} T_{1(2)} + n_3^{(2)} T_{1(3)}, \\ T_{1(2)} &= n_2^{(2)} T_{0(2)} + n_3^{(2)} T_{0(3)}, \\ T_{0(2)} &= 0, \end{split}$$

$$(2.21)$$

where $X_{(i)} = T_{5(i)} + T_{6(i)}$. We have equated quantities $T_{i(j)}$ for different levels in order to obtain equations for stationary values in the bulk, just as in section 2.1.

Now consider a 3-coordinated site (site 3 in Figure 2.8); when it is cut off from the upper level, it remains connected to two sites (4 and 5 in Figure 2.8). Site 3 has 6 d.o.f., if both sites 4 and 5 have 5 or 6 d.o.f. Site 3 has 5 d.o.f., if one of sites 4 and 5 has 4 d.o.f. and the other has 5 or 6 d.o.f. Site 3 has 4 d.o.f., if one of sites 4



Figure 2.8: A drawing illustrating how the equations for quantities $T_j^{(i)}$, (2.21) and (2.22), are constructed. Sites 1 and 3 are 2- and 3-coordinated, respectively; sites 2, 4 and 5 can have coordinations 2 or 3 and averaging over their possible coordinations should be done.

and 5 has 3 d.o.f. and the other has 5 or 6 d.o.f., or if both of these sites have 4 d.o.f. This can be continued by analogy for site 3 having 3, 2 and 1 d.o.f. For 0 d.o.f., we just use $\sum_{i=0}^{6} T_{i(j)} = 1$. Finally, the equations are

$$\begin{split} T_{6(3)} &= \left(n_2^{(3)}X_{(2)} + n_3^{(3)}X_{(3)}\right)^2, \\ T_{5(3)} &= 2\left(n_2^{(3)}T_{4(2)} + n_3^{(3)}T_{4(3)}\right)\left(n_2^{(3)}X_{(2)} + n_3^{(3)}X_{(3)}\right), \\ T_{4(3)} &= 2\left(n_2^{(3)}T_{3(2)} + n_3^{(3)}T_{3(3)}\right)\left(n_2^{(3)}X_{(2)} + n_3^{(3)}X_{(3)}\right) \\ &+ \left(n_2^{(3)}T_{4(2)} + n_3^{(3)}T_{4(3)}\right)^2, \\ T_{3(3)} &= 2\left(n_2^{(3)}T_{2(2)} + n_3^{(3)}T_{2(3)}\right)\left(n_2^{(3)}X_{(2)} + n_3^{(3)}X_{(3)}\right) \\ &+ 2\left(n_2^{(3)}T_{3(2)} + n_3^{(3)}T_{3(3)}\right)\left(n_2^{(3)}T_{4(2)} + n_3^{(3)}T_{4(3)}\right), \\ T_{2(3)} &= 2\left(n_2^{(3)}T_{1(2)} + n_3^{(3)}T_{1(3)}\right)\left(n_2^{(3)}X_{(2)} + n_3^{(3)}X_{(3)}\right) \\ &+ 2\left(n_2^{(3)}T_{2(2)} + n_3^{(3)}T_{2(3)}\right)\left(n_2^{(3)}T_{4(2)} + n_3^{(3)}T_{4(3)}\right) \\ &+ \left(n_2^{(3)}T_{3(2)} + n_3^{(3)}T_{3(3)}\right)^2, \end{split}$$

$$\begin{split} T_{1(3)} &= 2\left(n_2^{(3)}T_{0(2)} + n_3^{(3)}T_{0(3)}\right)\left(n_2^{(3)}X_{(2)} + n_3^{(3)}X_{(3)}\right) \\ &\quad + 2\left(n_2^{(3)}T_{1(2)} + n_3^{(3)}T_{1(3)}\right)\left(n_2^{(3)}T_{4(2)} + n_3^{(3)}T_{4(3)}\right) \\ &\quad + 2\left(n_2^{(3)}T_{2(2)} + n_3^{(3)}T_{2(3)}\right)\left(n_2^{(3)}T_{3(2)} + n_3^{(3)}T_{3(3)}\right), \\ T_{0(3)} &= 1 - T_{1(3)} - T_{2(3)} - T_{3(3)} - T_{4(3)} - X_{(3)}. \end{split}$$

Note that when writing these equations, we have assumed that the probability for site 4 in Figure 2.8 to be *i*-coordinated and simultaneously for site 5 to be *j*-coordinated is the product $n_i^{(3)}n_j^{(3)}$, i.e., we have used the absence of correlations between the types of second neighbors.

It is convenient to introduce quantities

$$T_{i}^{(j)} = n_{2}^{(j)}T_{i(2)} + n_{3}^{(j)}T_{i(3)},$$

$$X^{(j)} = n_{2}^{(j)}X_{(2)} + n_{3}^{(j)}X_{(3)},$$
(2.23)

j = 2, 3. That is, the new quantities $T_i^{(j)}$ and $X^{(j)}$ are averages of the old quantities $T_{i(k)}$ and $X_{(k)}$ over the possible values of j with the weights equal to the probabilities that a j-site is connected to a site known to be of type k. The equations for the new quantities are considerably simpler:

$$\begin{split} T_{6}^{(j)} &= n_{2}^{(j)} X^{(2)} + n_{3}^{(j)} X^{(3)2}, \\ T_{5}^{(j)} &= n_{2}^{(j)} T_{4}^{(2)} + 2n_{3}^{(j)} T_{4}^{(3)} X^{(3)}, \\ T_{4}^{(j)} &= n_{2}^{(j)} T_{3}^{(2)} + n_{3}^{(j)} \left(2T_{3}^{(3)} X^{(3)} + \left(T_{4}^{(3)}\right)^{2} \right), \\ T_{3}^{(j)} &= n_{2}^{(j)} T_{2}^{(2)} + 2n_{3}^{(j)} \left(T_{2}^{(3)} X^{(3)} + T_{3}^{(3)} T_{4}^{(3)} \right), \\ T_{2}^{(j)} &= n_{2}^{(j)} T_{1}^{(2)} + n_{3}^{(j)} \left(2 \left(T_{1}^{(3)} X^{(3)} + T_{2}^{(3)} T_{4}^{(3)} \right) + \left(T_{3}^{(3)}\right)^{2} \right), \\ T_{1}^{(j)} &= n_{2}^{(j)} T_{0}^{(2)} + 2n_{3}^{(j)} \left(T_{0}^{(3)} X^{(3)} + T_{1}^{(3)} T_{4}^{(3)} + T_{2}^{(3)} T_{3}^{(3)} \right), \\ T_{0}^{(j)} &= 1 - T_{1}^{(j)} - T_{2}^{(j)} - T_{3}^{(j)} - T_{4}^{(j)} - X^{(j)}. \end{split}$$

Just as Eq. (2.2), of which this system is an analog, it always has a trivial solution corresponding to the floppy phase (in this case, this trivial solution is $T_i^{(j)} = 0$ for $i = 0, \ldots, 5, T_6^{(j)} = 1$). Besides, there can be a non-trivial solution that should be found numerically.

2.2.3 Floppy modes and locating the transition

We can now find the solution(s) of Eq. (2.24). However, this by itself is essentially useless, as we do not even know where the trivial solution should be chosen and where the non-trivial solution should be used instead (in other words, we cannot as yet locate the rigidity transition). Besides, we would like to find some quantities that we can compare with the pebble game directly, in particular, the number of floppy modes F.

For 2D RBM, the number of floppy modes was found in Ref. [12] by starting with the fully coordinated network (having zero floppy modes) and then keeping track of changes in F when bonds are removed one by one (these changes can be related to quantities T_i). We use a similar method here. It is, however, more convenient to do *bond decoration*, rather than bond dilution, as we describe below.

We start with a network that is obtained by decorating (i.e., inserting 2-coordinated sites into) some randomly chosen bonds of a fully 3-coordinated network. Such a network has no 2-2 bonds, but is random otherwise (in particular, it has no higher-order correlations). This corresponds to being on the right boundary of the allowed region of the x- θ plane (Figure 2.6). We now start inserting additional 2-sites, but always only next to an existing 2-site. Then, if a neighbor of a 3-site was a 2-site, the corresponding neighbor will always remain a 2-site; if it was a 3-site, it will always remain a 3-site. Then $n_i^{(3)}$ do not change in the process and

from Eq. (2.12), the trajectories in the x- θ plane are given by

$$\theta = \frac{Cx}{2+x},\tag{2.25}$$

where C is a constant determined by the initial concentration of 3-sites (i.e., by exactly where on the right boundary we started). Note that C = 0 corresponds to moving along the phase separation line $\theta = 0$, and C = 3 corresponds to moving along the left boundary of the allowed region, thus varying C from 0 to 3, we cover the whole allowed region. Higher-order correlations are not introduced by this decoration procedure, since all bonds stemming from a 2-site can be decorated completely independently of each other.

Now let us see how the number of floppy modes is changed when one site is inserted as shown in Figure 2.9. Site 1 is inserted next to a 2-coordinated site 2, that prior to insertion, had two neighbors, sites 3 and 4. If we remove bonds connecting sites 3 and 4 to site 2, the probability, say, for site 3 to have *i* d.o.f. is given by $T_i^{(2)}$ (same for site 4). This is because site 3 is a neighbor of a 2-coordinated site 2, so $T_i^{(2)}$ are indeed the appropriate averages over possible types of site 3 [see Eq. (2.23)]. When site 1 is inserted, the number of floppy modes does not change, if, with the bonds removed as described above, one of the sites 3 and 4 had 0 d.o.f. and the other had 3 or less; or if one of the sites 3 and 4 had 1 d.o.f. and the other 1 or 2. Otherwise, the number of floppy modes *F* is increased by 1. Then in a single insertion act the change in *F* is on average

$$\Delta F = 1 - T_0^{(2)} \left(T_0^{(2)} + 2 \left(T_1^{(2)} + T_2^{(2)} + T_3^{(2)} \right) \right) - T_1^{(2)} \left(T_1^{(2)} + 2T_2^{(2)} \right).$$
(2.26)

The number of sites changes by 1 with each insertion, so

$$\frac{dF}{dN} = \Delta F. \tag{2.27}$$

We should work with a size-independent quantity, so we use f = F/3N (3N is the combined number of all degrees of freedom (not just zero-frequency ones) of the original 3D network, before they are replaced by bodies). Given that the number of 3-sites Nx = const (as only 2-sites are added), Ndx + xdN = 0 and

$$\frac{df}{dx} = -\frac{\Delta F - 3f}{3x}.$$
(2.28)

The fraction of floppy modes f can then be determined by numerical integration of (2.28). The initial condition for this integration is known: we start at the right boundary of the allowed region, where there are no 2-2 bonds, and such networks have no floppy modes, as we prove in Chapter 3, where we show that this is always the case, even for regular networks (with finite loops).



Figure 2.9: A drawing illustrating the site insertion procedure used to obtain the number of floppy modes and locate the transition. Site 1 is inserted next to a 2-coordinated site 2. Sites 3 and 4 can have coordination 2 or 3, and averaging over their possible coordinations should be done.

Moving along various trajectories (2.25) for different C, we could in principle find the number of floppy modes everywhere in the allowed region of the x- θ plane, *if* we knew the quantities $T_j^{(i)}$ everywhere. While we can find them solving Eqs. (2.24), there is an obvious caveat: in some parts of the region, there are two possible solutions, the trivial one (corresponding to the floppy phase) and the non-trivial one (corresponding to the rigid phase), and we still do not know where the transition between them (the rigidity transition) is located. However, what we know is

• at the right boundary of the allowed region, we are in the rigid phase, so we

should use the non-trivial solution, and we can check using (2.24) that this solution is $T_0^{(2)} = 1$, $T_0^{(3)} = n_3^{(3)}$, $T_1^{(3)} = n_2^{(3)}$, and the rest of $T_j^{(i)}$ are zero;

• since finite rigid clusters cannot exist in an RBN and thus there are no rigid clusters and hence no redundant constraints in the floppy phase, the number of floppy modes in the floppy phase is given *exactly* by Maxwell counting:

$$f = 2 - \frac{5}{6} \langle r \rangle; \tag{2.29}$$

as the number of floppy modes F can change by at most 1, when a single constraint is removed and thus by at most 5, when a single bond (consisting of 5 constraints) is removed, so in any case, the change is finite, the fraction of floppy modes f is always continuous when moving along any continuous trajectory in the x-θ plane.

This is enough to locate the transition. Namely, we start at the right boundary and move along a chosen trajectory tracking f by integrating Eq. (2.28) substituting the non-trivial solution for $T_j^{(i)}$ (since we start in the rigid phase) into Eq. (2.26). While moving along the trajectory, we compare thus obtained f with the f in the floppy phase given by Eq. (2.29), and the point where they are equal is the rigidity transition point for the given trajectory. In practice, it is convenient to differentiate Eqs. (2.24) along the trajectory and obtain $T_j^{(i)}$ by integration using the initial conditions at the right boundary given above and moving along the trajectory in small increments solving a linear system at each step. The procedure of finding the transition is illustrated in Figure 2.10.

By repeating the above procedure for many different trajectories (2.25) given by different C, we can locate the *phase boundary* in the x- θ plane and thus obtain the *phase diagram*. We show this phase diagram in Figure 2.11, where we also show the *spinodal line* — the line where the non-trivial solution ceases to exist. The



Figure 2.10: The number of floppy modes per degree of freedom f (thick red line) as a function of mean coordination $\langle r \rangle$ along the site insertion line 5 in the phase diagram (Figure 2.11). This plot also illustrates how the rigidity transition can be located. Line 1 is the Maxwell counting line coinciding with the true number of floppy modes in the floppy region (below the transition). Line 2 is obtained by counting the floppy mode changes in the site insertion process (as described in the text). The transition is located at the point where these two lines intersect.

phase boundary starts at the point corresponding to x = 0 ($\langle r \rangle = 2$). This is because if, for example, there is complete phase separation, there are two completely separate subnetworks, a subnetwork of 2-sites and a subnetwork of 3-sites, and the subnetwork of 3-sites is rigid and can percolate even when it is very small compared to the subnetwork of 2-sites. The phase boundary ends at the upper corner of the allowed region corresponding to the perfect chemically alternating network (...-2-3-2-3-...). At this point $\langle r \rangle = 2.4$, so the Maxwell counting prediction for the transition is exact. In fact, in Chapter 3 we prove that this is the case for any perfectly chemically alternating networks consisting of 2- and 3-coordinated sites (even regular ones with loops). Note also that the transition is close to the Maxwell prediction $\langle r \rangle = 2.4$ for a wide range of the chemical order parameter θ : for example, for perfectly random bonding, the transition is at 2.3855...



Figure 2.11: The phase diagram for 2-3 random bond networks. The thick red line is the phase boundary separating the floppy and the rigid phases. The thin purple line is the spinodal line, where the non-trivial solution of equations (2.24) ceases to exist. Lines 1 to 4 are as in Figure 2.6. Line 5 is one of bond insertion lines given by Eq. (2.25).

We can compare our results for the number of floppy modes with numerical results obtained by the pebble game for randomly diluted networks, by using Eq. (2.20) for the trajectory in the x- θ plane and neglecting small higher-order correlations in randomly diluted networks. In fact, one can obtain the analogs of Eqs. (2.26) and (2.28) for the trajectory (2.20) instead of trajectory (2.25) and then obtain f along the whole trajectory (2.20) in a single integration procedure, instead of multiple integrations along many trajectories (2.25). We do not give the details of this calculation here, as it is not essential and just simplifies things; these details can be found in Ref. [6]. In Figure 2.12, we compare the derivative of the fraction of floppy modes f with respect to $\langle r \rangle$ obtained using the theory outlined here and using the pebble game (averaged over several realizations). The agreement is excellent.



Figure 2.12: Comparison between the theory and the pebble game simulation results for the first derivative of the number of floppy modes per degree of freedom with respect to the mean coordination for 2-3 networks obtained by random dilution. The solid lines represent the theory described in this chapter. Brown open circles are simulation results for a sample with 8000 sites, green solid triangles are for a sample with 32000 sites. The inset shows the blowup of the region near the rigidity transition, with the open black diamonds for a sample with 103823 sites (averaged over 30 realizations) and the filled green circles for a sample with 262144 sites (averaged over 20 realizations). Simulation and plot by A.J. Rader.

We can also find the size of the percolating rigid cluster. A site belongs to the percolating rigid cluster, if it has zero degrees of freedom, but when none of its bonds are cut. That is, the probability that a site is in the infinite cluster is not equal to T_0 (which is the probability of having 0 d.o.f., when one bond is cut) and

has to be calculated separately.

If a site has coordination 2, it has *more than* zero d.o.f., when at least one of its neighbors has 5 d.o.f. with the bond to the site under consideration cut; or when one neighbor has 4 d.o.f. with one bond cut and the other has more than 0 d.o.f., again with one bond cut; etc. Thus the probability for a 2-site to have 0 degrees of freedom is

$$P_{\infty(2)} = \left(1 - X^{(2)}\right)^2 - T_4^{(2)} \left(2T_1^{(2)} + 2T_2^{(2)} + 2T_3^{(2)} + T_4^{(2)}\right) - T_3^{(2)} \left(2T_2^{(2)} + T_3^{(2)}\right).$$
(2.30)

Similarly for a 3-coordinated site:

$$P_{\infty(3)} = 1 - X^{(3)3} - 3X^{(3)2} \left(1 - X^{(3)}\right) - 3X^{(3)} \left(\left(T_3^{(3)} + T_4^{(3)}\right)^2 + 2T_4^{(3)} \left(T_1^{(3)} + T_2^{(3)}\right) + 2T_2^{(3)}T_3^{(3)}\right) - T_4^{(3)} \left(T_4^{(3)} \left(3T_2^{(3)} + 3T_3^{(3)} + T_4^{(3)}\right) + 3T_3^{(3)2}\right).$$

$$(2.31)$$

The calculations show that for randomly bonded and randomly diluted (Figure 2.13) networks, the size of the percolating cluster is 80-85% of the network just above the transition. It decreases and tends to zero, as phase separation is

approached along the phase boundary, so the transition is more weakly first order close to phase separation.



Figure 2.13: The fraction of sites in the infinite rigid cluster as a function of the mean coordination for randomly diluted 2-3 random bond networks. The solid lines represent the theory described here: the red line is the solution that is actually realized and the black line between the spinodal point $\langle r \rangle_s$ and the transition point $\langle r \rangle_c$ is the "metastable" non-trivial solution where the trivial solution corresponding to the floppy phase is actually realized. The green dots are the pebble game simulation results averaged over 10 networks of 100000 sites.

2.2.4 Networks with 2- and 4-coordinated sites

The theory can be generalized straightforwardly to networks consisting of 2and 4-coordinated sites. The analog of the system (2.24) is

$$\begin{split} T_{6}^{(j)} &= n_{2}^{(j)} X^{(2)} + n_{4}^{(j)} X^{(4)3}, \\ T_{5}^{(j)} &= n_{2}^{(j)} T_{4}^{(2)} + 3n_{4}^{(j)} T_{4}^{(4)} X^{(4)2}, \\ T_{4}^{(j)} &= n_{2}^{(j)} T_{3}^{(2)} + 3n_{4}^{(j)} \left(T_{3}^{(4)} X^{(4)2} + T_{4}^{(4)2} X^{(4)} \right), \\ T_{3}^{(j)} &= n_{2}^{(j)} T_{2}^{(2)} + n_{4}^{(j)} \left(3T_{2}^{(4)} X^{(4)2} + 6T_{3}^{(4)} T_{4}^{(4)} X^{(4)} + T_{4}^{(4)3} \right), \\ T_{2}^{(j)} &= n_{2}^{(j)} T_{1}^{(2)} + 3n_{4}^{(j)} \left(T_{1}^{(4)} X^{(4)2} + \left(2T_{2}^{(4)} T_{4}^{(4)} + T_{3}^{(4)2} \right) X^{(4)} \right. \\ &\qquad + T_{3}^{(4)} T_{4}^{(4)2} \right), \\ T_{1}^{(j)} &= n_{2}^{(j)} T_{0}^{(2)} + 3n_{4}^{(j)} \left(T_{0}^{(4)} X^{(4)2} + 2 \left(T_{1}^{(4)} T_{4}^{(4)} + T_{2}^{(4)} T_{3}^{(4)} \right) X^{(4)} \right. \\ &\qquad + T_{2}^{(4)} T_{4}^{(4)2} + T_{3}^{(4)2} T_{4}^{(4)} \right), \\ T_{0}^{(j)} &= 1 - T_{1}^{(j)} - T_{2}^{(j)} - T_{3}^{(j)} - T_{4}^{(j)} - X^{(j)}. \end{split}$$

The phase diagram is shown in Figure 2.14. The upper corner of the allowed region corresponding to the fully chemically ordered networks now lies at $\langle r \rangle = 8/3$, which is far above the rigidity threshold, so the phase boundary no longer goes there, but instead ends at the left boundary of the allowed region.

2.3 Networks with dangling ends

The case of networks having sites of coordination 1 and thus dangling ends (Figure 2.15) deserves special consideration. As we show in Chapter 1, in this case Maxwell counting predicts that the transition location deviates from the "standard"



Figure 2.14: The phase diagram for 2-4 networks. The thick red line is the phase boundary between the rigid (blue) and the floppy (yellow) phases. The thin purple line is the spinodal line. The black dashed line is the random bonding line.

value $\langle r \rangle_c^M = 2.4$ and is given by

$$\langle r \rangle_c^M \simeq 2.4 - 0.4 n_1,$$
 (2.33)

where n_1 is the concentration of 1-coordinated sites. However, this equation may not be entirely adequate. In particular, if there are 1-sites that are bonded to 2-sites, these 1-sites can terminate chains of 2-sites. Such chains are completely irrelevant to rigidity and can be removed before rigidity is analyzed. Yet this process of removal can raise the mean coordination of the network significantly. We can consider this process of removal analytically in the framework of the random bond model.

In this section we first consider the case when 1-sites are only bonded to sites of coordination 3 and higher and show that in this case Eq. (2.33) is indeed as accurate as original Maxwell counting for networks without dangling ends, but only for low n_1 . Then we consider the case of networks with 1-, 2- and 3-sites (1-2-3



Figure 2.15: A random bond network with 1-fold coordinated sites (gray) and thus dangling ends (pale blue lines and sites with paler colors). Roots of dangling end trees are blue; other sites are green (2-coordinated) and red (3-coordinated).

networks) with completely random bonding. We show that in this case the calculation can be carried out analytically to the very end and the final result is remarkably simple: for *arbitrary* n_1 , the dependence of the threshold on n_1 is linear, just as in Eq. (2.33), *but* the slope is twice as big.

2.3.1 Networks without 1-2 bonds

In this case, if the concentration of 1-sites n_1 is low, we can neglect sites of coordination 3 or higher with more than one bond to 1-sites. Then removing bonds going to 1-sites never creates new 1-sites. If the number of sites is N and the number of bonds is $N_B = \frac{\langle r \rangle N}{2}$, then after removal of "dangling bonds" going to 1-sites, the number of remaining bonds is $N'_B = N_B - n_1 N$ and the number of remaining sites is

 $N' = N - n_1 N$, so the mean coordination of the residual network is

$$\langle r \rangle' = \frac{2N'_B}{N} = \frac{2(N_B - n_1N)}{N - n_1N} = \frac{\langle r \rangle - 2n_1}{1 - n_1}.$$
 (2.34)

If one assumes that Maxwell counting is correct, then, since the final network has no 1-coordinated sites, $\langle r \rangle'_c = 2.4$ and we obtain Eq. (2.33). Note, however, that we have made an assumption that n_1 is small. For higher n_1 , there is a non-negligible fraction of dangling ends created when original dangling ends are removed, and this would lower the transition compared to Eq. (2.33). So there will be terms of order higher than linear in n_1 .

Note that the above consideration is valid for both RBNs and regular networks.

2.3.2 Randomly bonded 1-2-3 networks

Consider a Bethe lattice with 1-, 2- and 3-sites and perfectly random bonding. In this case, quantities $n_j^{(i)}$ defined as in the previous section do not depend on the upper index. We denote

$$n_{1}^{(i)} = \frac{n_{1}}{\langle r \rangle} \equiv m_{1},$$

$$n_{2}^{(i)} = \frac{n_{2}}{\langle r \rangle} \equiv m_{2},$$

$$n_{3}^{(i)} = \frac{n_{3}}{\langle r \rangle} \equiv m_{3},$$
(2.35)

where n_i are concentrations of *i*-sites. Consider an arbitrary 2- or 3-coordinated site. It can be considered as a root of respectively 2 or 3 *subtrees*. Each of the subtrees can be finite or infinite. If a subtree is finite, it is a dangling end and should be eliminated. If all subtrees starting at the site are finite, the site is obviously eliminated when the subtrees are cut. But it is also eliminated, if just one of the subtrees is infinite and the rest one or two are finite, since after eliminating the finite tree(s), the site becomes 1-coordinated and thus a dangling end.

Let the probability that a subtree is finite be P (for perfectly random bonding, this probability does not depend on whether the root is a 2- or a 3-site, as we will see). We can write a self-consistent equation for P reasoning as follows. In order for a subtree to be infinite, first, starting at the subtree root, we should encounter a branching point (a 3-site) before encountering a dead end (a 1-site). The probability of this is $m_3(1 + m_2 + m_2^2 + m_2^3 + ...) = m_3/(1 - m_2)$. But in addition to this, at least one of the two branches starting out at the branching point should also be an infinite subtree (the probability of which is $1 - P^2$). Then the equation is

$$1 - P = \frac{m_3}{1 - m_2} (1 - P^2), \qquad (2.36)$$

or

$$P = \frac{1 - m_2 - m_3}{m_3} = \frac{m_1}{m_3} = \frac{n_1}{3n_3}.$$
 (2.37)

A 2-site remains in the network, if both of the subtrees stemming out of it are infinite, the probability of which is $(1 - P)^2$. Then it obviously remains a 2-site. A 3-site remains in the network, if two of the three subtrees stemming out of it are infinite and the third is finite (probability $3P(1 - P)^2$), in which case it becomes a 2-site, or if all three subtrees are infinite (probability $(1 - P)^3$), then it remains a 3-site. Then the mean coordination after the removal is

$$\langle r \rangle' = \frac{2n_2(1-P)^2 + 2n_3 \times 3P(1-P)^2 + 3n_3(1-P)^3}{n_2(1-P)^2 + 3n_3P(1-P)^2 + n_3(1-P)^3} = \frac{n_1 + 2n_2 + 3n_3}{1 - n_1/3} = \frac{\langle r \rangle}{1 - n_1/3}.$$
 (2.38)

Then

$$\langle r \rangle_c = \left(1 - \frac{n_1}{3}\right) \langle r \rangle_c',$$
 (2.39)

Within Maxwell counting, $\langle r \rangle'_c = 2.4$, since the final network has no 1-coordinated sites, so that finally

$$\langle r \rangle_c \simeq 2.4 - 0.8 n_1.$$
 (2.40)

Note that this equation is similar to (2.33), but with the slope twice as big. Also, unlike in the previous subsection, the relation holds for *arbitrary* (not necessarily small) n_1 .

This result is obtained for Bethe lattices or RBNs, but it hopefully holds reasonably well for regular lattices. For RBNs themselves, we can do even better. The final network is randomly bonded, which can be shown by explicitly calculating $n_j^{(i)}$ for it, but is obvious anyway, since the original network was randomly bonded, so dangling ends that were removed were located in random places in the network. Then from our theory in the previous section we know the exact answer $\langle r \rangle_c = 2.3855...$ and the *exact* result is

$$\langle r \rangle_c = 2.3855...(1 - n_1/3).$$
 (2.41)

Of course, we can consider correlated (not perfectly random) networks in a similar fashion, although the derivations are more cumbersome, in particular, the probability of a finite subtree now depends on whether the root is a 2-site or a 3-site. The analog of self-consistent equation (2.36) should now be written for a 3-site, and the probability for a 2-site should be obtained afterwards. Otherwise, the derivation is similar, but there is no simple result like (2.40) or (2.41).

2.4 Conclusion

We have considered the theory of rigidity percolation on 3D bond-bending glass networks within the random bond model. We have obtained a set of equations that have to be solved numerically, but have the advantage of being written for an infinite network, so, unlike any pebble game simulation, there are no finite-size effects. We have allowed for the possibility of correlations in bonding. The Maxwell counting result for the rigidity threshold turned out to be very good in a wide range of the bonding correlation parameter, but deviations are strong close to phase separation. The biggest difference from regular networks is that the rigidity transition is first, not second order.

We have also done some studies (not described here) of how the rigidity threshold and the number of floppy modes can be obtained approximately in a more physically intuitive manner, rather than as a result of a rather involved procedure, by considering explicitly the regions of the network giving the biggest contributions to the number of floppy modes and adding up these contributions.

As for the possibilities for future research inspired by the results obtained here, it would certainly be interesting to see if the influence of chemical order is the same in regular networks as in RBNs. We have seen that chemical ordering has only a very weak effect on the rigidity threshold, except close to the phase separation and we expect the same to hold true for regular networks as well, but this is certainly worth checking. This can be done by constructing regular networks and studying them with the pebble game, although analysis should be done for each network at each concentration and chemical order, unlike the special case of random dilution, where the whole sequence of networks can be studied in one pebble game run. It would also be interesting to see if Eq. (2.40) for networks with 1-coordinated sites obtained for RBNs is good enough for regular networks as well.

Chapter 3: Chemically ordered networks

In Chapter 2, we have studied, in particular, the effect that chemical ordering in networks has on location of the rigidity transition. In this short chapter, our goal is more narrow and more broad at the same time. On the one hand, instead of the whole range of possible chemical correlations, we now confine ourselves to studying just those networks that are as close to perfect chemical order as possible for given concentrations of sites of different types. But on the other hand, results that we obtain in this chapter are very general and apply to a very broad class of disordered networks, including regular ones, rather than just to pathological random bond networks. We show that in all networks in this broad class, there is an extremely sharp first-order transition, such that the network goes from completely rigid and stressed to completely floppy with addition of just two sites, no matter how big the network is. The results presented in this chapter were published in Ref. [49].

Consider a disordered bond-bending network, all sites of which are 3-fold coordinated. This can be, for example an amorphous arsenic (As) network, as As atoms are trivalent. An example of such a network is in Figure 3.1 (only central-force bonds are shown). The network in the figure is planar, but it is assumed that atoms can move in 3 dimensions and all the results of this chapter apply to truly 3-dimensional networks as well. The particular example in Figure 3.1 was obtained in [50] by starting from a honeycomb lattice and amorphizing it by a procedure similar to the WWW method (see section 1.2.5). The network is constructed with *periodic boundary conditions*, this results in connections between sites at opposite boundaries and ensures that there are no free boundaries and all atoms are 3-coordinated even in a finite sample.



Figure 3.1: The fully 3-coordinated network of 1800 sites obtained in Ref. [50], as described in the text, and used in the simulations in this chapter.

Now start decorating bonds with atoms [Figure 3.2,(b)], so that 2-coordinated sites appear (such as selenium [Se] atoms having valence 2). At this stage, we do not allow more than one 2-coordinated atom to be placed between 3-coordinated atoms. This ensures that there are no 2-2 bonds and thus maximum chemical order is obtained. Doing this, we can reach $\langle r \rangle = 2.4$, which corresponds to the As₂Se₃ composition [Figure 3.2,(c)]. At this point all bonds of the original 3-coordinated network are decorated with exactly one atom and the network is fully chemically ordered: there are only 2-coordinated atoms next to 3-coordinated atoms and vice versa. In the phase diagram in Chapter 2 (Figure 2.11) this corresponded to the upper corner of the allowed region, and the decoration process up to this point corresponds to moving along the right boundary of the allowed region. We continue the process of atom insertion further, by decorating bonds of the original 3-coordinated network one by one with a second atom [Figure 3.2,(d)], never allowing the third atom to be inserted in the same bond of the original 3-coordinated network; this way we can go below 2.4 and reach $\langle r \rangle = 2.25$. In the diagram in Figure 2.11 this corresponds to going along the left boundary of the allowed region; note, however that the fact that only two sites at most can decorate any original bond implies higher-order correlations in bonding assumed absent throughout Chapter 2.

We can study, e.g., the specific network in Figure 3.1 using the pebble game algorithm, decorating the network as described above and studying its rigidity properties as a function of $\langle r \rangle$. We find that in this case, there are no redundant bonds in the network below the rigidity transition, just like in the random bond model case. In addition to that, the transition is always exactly at $\langle r \rangle = 2.4$ and above this point the whole network becomes rigid and stressed (Figure 3.3) and there are no floppy modes at all (Figure 3.4). Note that since there is a jump in the sizes of the percolating rigid cluster and the percolating stressed region, this is a



Figure 3.2: An illustration of the bond decoration process. The starting point is a fully 3-coordinated network (a). Bonds are chosen in random order and decorated with one site each (b), until all bonds are decorated (c). Then bonds of the original network start to be decorated with a second site (d) — bonds decorated with two sites are circled.

first-order transition, just like in the RBM case (but note that this is a regular network, not a pathological one like the random bond network). Not only there is a jump, it is also the biggest possible, as *the whole network* goes immediately from being floppy to completely rigid and from completely unstressed to completely stressed.



Figure 3.3: The fractions of sites in both the percolating rigid cluster and the percolating stressed region (going from 0 to 1 at the transition) for the chemically ordered networks obtained by the bond decoration procedure described in the text (red line), compared to the fractions of sites in the percolating rigid cluster (green open circles) and in the percolating stressed region (green solid circles) for a network obtained by random bond dilution of the diamond lattice (as in Figure 1.10).

Let us now look more carefully how this rigidity transition occurs and at finite-size effects in particular. Since the whole network is rigid above the transition, the number of floppy modes F = 6 there. On the other hand, below the transition Maxwell counting is exact and F = 6 is reached for a network with all original bonds decorated with one atom, except for six bonds that are decorated with two



Figure 3.4: The number of floppy modes per degree of freedom for the chemically ordered networks obtained by decorating a network of 3×3 supercells shown in Figure 3.1 (red circles) and for the randomly bond diluted diamond lattice, as in Figure 1.9 (green diamonds). The black dashed line is the Maxwell counting result.

atoms. It turns out that such a network is completely rigid, yet stressless (isostatic). In the thermodynamic limit these six extra atoms are, of course, negligible, and the transition is at $\langle r \rangle = 2.4$ exactly. When just one more atom is added, so that seven bonds are decorated with two atoms, Maxwell counting (which is exact) gives F = 7, so there is just one internal floppy mode. But at the same time, the only rigid clusters are single sites with their associated central-force and angular constraints and there are no bigger rigid clusters, so in this respect the network is completely floppy and the only internal floppy mode is spread over the whole network. On the other hand, if only five bonds are decorated with two atoms, the network is not only completely rigid, but also all constraints are stressed.

Thus at $\langle r \rangle = 2.4$ plus five extra atoms, the network is completely rigid and stressed; when one more atom is added, it goes from completely stressed to



Figure 3.5: A set of panels illustrating finite size effects for chemically ordered networks. Moving along the horizontal axis corresponds to changing the number of decorating sites. The right border of the plot corresponds to a network with one atom decorating each bond, which is exactly $\langle r \rangle = 2.4$. Moving to the left corresponds to adding second sites to bonds one by one; the number of added sites is specified. Panel (a) shows the number of floppy modes F (red circles). The solid green line is Maxwell counting. Panels (b) and (c) show the fractions of sites in the percolating rigid cluster and in the percolating stressed region, respectively (red).

completely unstressed; and with just one more atom added, it becomes completely floppy. So the transition is extremely sharp. This is illustrated in Figure 3.5.

We now prove that the above observations are true for a very broad class of networks. Our proof even applies to finite networks, as long as all atoms of the starting network are still 3-coordinated. In the case in Figure 3.1 that we treated with numerical simulations, this was ensured by using periodic boundary conditions; but even with free boundaries, this can be achieved; this is a minor issue, of course, as we are mostly interested in the thermodynamic limit. The only other assumption we make is that every subnetwork of the original network is connected to the rest of the network with at least four bonds (except, of course, those subnetworks consisting of just a single site, or vice versa, lacking just a single site - a site is connected with just three bonds, of course). This is a rather weak assumption that the network in Figure 3.1 certainly satisfies. Note, however, that the initial network should contain no triangles, as a triangle is always connected with exactly three bonds to the rest of the network. In Figure 3.1, by the way, the smallest ring is a pentagon.

The idea of the proof is to do constraint counting for the whole network, as well as for all subnetworks. If, for example, we see that none of the subnetworks not coinciding with the whole network have enough constraints to be rigid, then there can be no rigid clusters smaller than the whole network. But if at the same time the network as a whole has enough constraints to be rigid, the natural conclusion is that the whole network is rigid, i.e., there is just one single rigid cluster coinciding with the whole network. Similarly we can make conclusions about the stressed regions. The underlying assumption is that all rigid clusters and all stressed regions in bond-bending networks are *contiguous* and *rigid by themselves*, which is implied by the *molecular framework conjecture*. We defer the detailed discussion of these assumptions to Chapter 4.

We start with a fully 3-coordinated network, finite or infinite, that satisfies the

above connectivity assumption (no piece connected with three bonds or less), but is arbitrary otherwise, and first decorate every bond with exactly one atom. Recall that such a network has the mean coordination of exactly 2.4. Consider now networks obtained from it by adding an extra atom to (a) 5 bonds; (b) 6 bonds; (c) 7 bonds. That is, in such a network all of the bonds of the original 3-coordinated network are decorated with one atom, except (a) 5; (b) 6; and (c) 7 bonds are decorated with two atoms. Note that in the thermodynamic limit these few extra atoms are negligible and all these networks still have $\langle r \rangle = 2.4$.

We first apply constraint counting to whole networks, i.e., do the usual Maxwell counting. If the number of atoms in the original 3-coordinated network was M, the number of 2-coordinated atoms is (a) (3/2)M + 5; (b) (3/2)M + 6; (c) (3/2)M + 7 in cases (a), (b) and (c), respectively. The number of 3-coordinated atoms is still M, the total number of sites is thus (a) N = (5/2)M + 5; (b) N = (5/2)M + 6 and (c) N = (5/2)M + 7. The total number of bonds is (a) B = 3M + 5; (b) B = 3M + 6; (c) B = 3M + 7. For Maxwell counting, we can use the body-bar representation (see Chapter 1), where a bond-bending network is replaced by a central-force one, every site is represented as a body with 6 degrees of freedom and each bond as a bunch of 5 bars. Then Maxwell counting gives F = 6N - 5B and (a) F = 5; (b) F = 6; (c) F = 7.

We now do constraint counting for subnetworks smaller than the full network. We are going to prove that in all three cases, (a), (b), and (c), constraint counting always gives more than 6 floppy modes for such networks (excluding, of course, trivial ones, consisting of just a single six-dimensional body). In view of this, it is clear that we need not consider subnetworks having dangling pieces (Figure 3.6), since these dangling pieces are always floppy and can only increase the floppy mode count. We now consider three kinds of subnetworks separately.

1. Subnetworks obtained by removing just a single chain going between two

3-coordinated atoms from the full network. Such a chain will contain one or (very rarely) two 2-coordinated sites and 2 or 3 bonds, respectively. Then the number of atoms in the subgraph is N - 1 or N - 2, respectively, and the number of bonds is B - 2 or B - 3, respectively, and by checking explicitly each of the cases (a), (b), and (c), we can see that the number of floppy modes is always bigger than 6 (the least favorable case is (a) with the number of atoms N - 2 and the number of bonds B - 3, which gives 8 floppy modes).

- 2. Subnetworks obtained by removing a single 3-coordinated site with three adjacent chains. Here the lowest number of floppy modes is in the case (a) and when all the three chains happen to have two sites. In this case the number of floppy modes given by constraint counting is 8 > 6.
- 3. All other subnetworks not having any dangling parts. By our assumption, such subnetworks are separated from the rest of the network by cutting 4 or more bonds. Cutting each of these bonds converts a 3-coordinated site into a 2-coordinated site. Each of the remaining 3-coordinated sites has three adjacent chains, each of them having at least one 2-coordinated site. Each of the chains is shared between two 3-coordinated sites, so if the number of 3-coordinated sites in the subnetwork is M₁, the number of chains is (3/2)M₁. Moreover, since at least four 3-coordinated sites became 2-coordinated, then at least either 4 of all chains consist of 3 sites each, or 2 of all chains consist of 3 sites each, or 1 chain consists of 7 sites and 1 chain consists of 3 sites, or 1 chain consists of 7 sites and 1 chain consists of 3 sites is at least (3/2)M₁ + 8 and the number of floppy modes is at least 8 more than 6.

Thus in case (a), for the whole network F = 5, so there must be redundant bonds and stress, but any subnetwork has no redundancy and no stress when

isolated from the rest of the network. The conclusion is that the whole network is stressed; if only a part was stressed, we could have considered this part alone as a subnetwork, and it would have been stressed alone. Of course, if there are even fewer decorating sites, all the more the whole network will be stressed. In case (b), for the whole network F = 6, for all subnetworks the number of floppy modes is bigger than 6. Then there are no redundant bonds, so no stress, but the whole network is rigid - a smaller subnetwork cannot be rigid by itself. Finally, in case (c), there are no redundant bonds still, and also, since now the number of floppy modes is more than six for all subnetworks and for the full network, there is no percolating rigid cluster and in fact, no rigid clusters at all, except trivial ones (a single site with all its neighbors that is represented as a single isolated body in the body-bar representation). When the number of decorated sites is higher, this conclusion is still going to hold, of course, and since there are no redundant bonds, Maxwell counting is exact. Thus the proof is complete.

These considerations are also applicable to a similar model, in which in the starting lattice all sites are 4-coordinated. Similarly, bonds are first decorated one by one with one atom, then, after all bonds are decorated, by the second atom, then by the third atom, etc. At $\langle r \rangle = 2.4$, there is again perfect chemical order: all original bonds are decorated with exactly two atoms. All considerations can be repeated and the same conclusions are reached: there is a sharp rigidity and stress transition at $\langle r \rangle = 2.4$ and Maxwell counting is exact below the transition point. The connectivity condition is even more relaxed: now every piece has to be connected to the rest of the network with 3 bonds or more.

The same result holds if bonds in the 4-coordinated network are decorated by *pairs* of sites. Then $\langle r \rangle = 2.4$ still corresponds to the point where every bond is decorated with one pair of sites. This is useful when considering networks of the *Si-O type*, in which angular constraints associated with the oxygen atoms are weak

and generally assumed to be broken. There is an assumption that in terms of rigidity, this is equivalent to replacing each oxygen with two 2-coordinated atoms and restoring all angular constraints. This assumption was checked using, in particular, the relaxation algorithm for studying rigidity of general (non-bond-bending) networks described in the next chapter, and it seems to be true indeed. Then, a chemically ordered SiO₂ network, with an O atom between every two Si atoms, corresponds to an initially 4-coordinated network with each bond decorated with two 2-coordinated atoms with angular constraints present everywhere. If we consider starting from a Si network and decorating bonds with O atoms, this will correspond to decorating with pairs of 2-coordinated atoms with angular constraints, and the rigidity transition will correspond exactly to the SiO₂ composition, with $\langle r \rangle = 8/3 = 2.667$.



Figure 3.6: Illustration of a subnetwork with a dangling end. Adding a dangling end (green) to a subnetwork (red) increases the floppy mode count, so subnetworks with dangling ends need not be considered in the proof in this chapter.

We note that in all cases considered here, the networks were fully chemically ordered at the threshold. Indeed, it is this absence of fluctuations in the chemical order that gives rise to the described behavior. If, for example, we start with a network that has both 3- and 4-coordinated sites and decorate it, then at $\langle r \rangle = 2.4$ there will be no perfect chemical order, as the number of 2-coordinated atoms per original bond will be between 1 and 2 on average, so this number will vary from bond to bond, and because of these variations, Maxwell counting will no longer be exact.

Chapter 4: Rigidity of general 3D networks

As we know, in two dimensions the Laman's theorem allows studying rigidity of networks through a fast algorithm, the pebble game. However, the theorem cannot be generalized to three-dimensional networks, except for a special class of bond-bending (BB) networks. While this special class is very important due to the fact that many covalent glass networks are BB networks, it is still just a narrow class, after all, and there are many applications, in which BB constraints should rather be considered absent, such as broken angular constraints at the oxygen atoms in the Si – O networks mentioned in Chapter 3 and other situations. Besides, there is purely theoretical interest, since the properties of the rigidity transition may turn out to be different for purely central-force (CF) networks (indeed they are, as we will see). Therefore studying non-BB 3D networks is of interest. This chapter describes our work on this topic.

It is, of course, possible to study rigidity with simple-minded direct algorithms, such as diagonalization of the dynamical matrix or relaxation of the corresponding network of springs. After a short discussion of basic notions about rigidity of general 3D networks in section 4.1, we describe such a method of studying rigidity based on relaxation in section 4.2. While the relaxation part is rather straightforward indeed, we do use some knowledge of rigidity theory, which allows us to include in a simple way most elements of rigidity analysis usually done with the pebble game and also identify potential trouble spots where the straightforward generalization of Laman's theorem and thus the pebble game would fail. The disadvantage is that the procedure is rather slow and essentially not applicable to networks bigger than a few thousand atoms because of round-off errors. As an

example of application of the algorithm, we describe rigidity analysis of networks obtained in colloidal glass experiments by D.A. Weitz's group.

While we know that the Laman's theorem is wrong in general in 3D, from the practical point of view it is certainly reasonable to ask a question just how wrong it is [51]. That is, if we apply the pebble game straightforwardly, pretending that the Laman's theorem works, how big are the errors? We do some tests in section 4.3, after outlining the details of the pebble game procedure; it appears that the results, though not exact, are quite good. In section 4.4 we test the applicability of the pebble game to purely CF diluted face-centered cubic (fcc) and body-centered cubic (bcc) networks, for which we conclude that the agreement is perfect and thus we can use the pebble game to study much bigger networks than possible with the relaxation procedure. We conclude that the rigidity transition is *first order*, unlike in 2D and BB 3D networks, where it is second order, but similar to random-bond networks (see Chapter 2). However, for networks that are two-dimensional, but with sites still able to move in 3D, the transition is apparently second order.

Our ultimate goal should be developing an integer algorithm, ideally as fast as the pebble game, that would be exact for general 3D networks. While we have not achieved this goal yet, we discuss in section 4.5 some approaches that may hopefully succeed. We end with conclusions in section 4.6.

Much of the research described in this section should be considered work in progress. We, however, decided that it was reasonable to include it here, as some very interesting results have already been obtained.

4.1 Basic notions of 3D rigidity

As we said, the Laman's theorem is not correct in spaces of dimension higher than two. As we know from Chapter 1, the straightforward generalization of the

Laman's theorem in 3D that is a part of the molecular framework conjecture for BB networks is that there are no redundant constraints in the network, if and only if in any subnetwork larger than 2 sites, the number of constraints is fewer than the triple number of sites by at least six. The simplest example of violation of this statement is the *double-banana graph* (Figure 4.1). In this case, the above condition is satisfied for every subgraph, therefore the conclusion would be that there are no redundant constraints, so that Maxwell counting should be exact and it gives the number of floppy modes equal to 6, so there are no internal floppy modes. In fact, there is one redundant constraint and consequently, one internal floppy mode.



Figure 4.1: The double-banana graph. Two bananas (green and yellow) can rotate around the implied hinge (red dashed line). The 3D generalization of the Laman's theorem is wrong for this network.

One important fact about the network in Figure 4.1 is that it contains an implied hinge. Hinges are axes around which two or more rigid parts of the network
can rotate. These axes have to go through two of the network's sites. These two sites can be directly connected by a constraint, in which case the hinge is *explicit*, otherwise the hinge is implicit, or *implied*. In Figure 4.1, the implied hinge is shown with the dashed line; two rigid "bananas" rotate around it. Such implied hinges are *never* present in BB networks [14]. Note that if the implied hinge is placed explicitly as a constraint, this does not change the rigidity properties, but the condition of the 3D generalization of the Laman's theorem will no longer be satisfied, so presence of a redundant constraint will be correctly detected.

In fact, in three dimensions the *only* reason for violation of the Laman's theorem is presence of implied hinges. This statement is only a *conjecture*, known as the *Dress conjecture* [51]. It was originally proposed to be true in all dimensions, but was subsequently disproved in 4- and higher-dimensional spaces. In three dimensions, there are still no known counterexamples, and, just as with the molecular framework conjecture for BB networks, we feel safe to rely on it. The Dress conjecture does not solve the rigidity problem in the way that the Laman's theorem does in 2D. But it essentially reduces the problem to identifying implied hinges. This will help us build the relaxation algorithm in the next section; this also gives us better hopes that eventually an integer algorithm for analyzing rigidity will be constructed.

Not every implied hinge causes errors in floppy mode counting. Imagine taking the double-banana graph and removing one constraint in one of the bananas, as shown in Figure 4.2. Again, the condition of the theorem is satisfied and Maxwell counting now gives one internal floppy mode, which is indeed the case. The essential difference between situations where the floppy mode count is wrong (as in Figure 4.1) and those where it is correct (as in Figure 4.2) will be explained below. Note, however, another potential problem: the right banana in Figure 4.2, from which a constraint was removed, is still rigid and forms a *rigid cluster*, since all other

sites in the graph are not rigid with respect to it. However, if one considers just this banana alone (i.e., just the red sites and constraints in the figure separately from the rest of the network), it is not rigid anymore. This cluster is rigidified by the rest of the network (or, one might say, by the "missing" implied hinge). Such a situation is never possible in 2D or in BB 3D networks: there, a rigid cluster is always rigid by itself, in isolation from the rest of the network. This is not a requirement of the Laman's theorem *per se*, but is implied by it and used when rigid cluster decomposition is done. This means that in situations like that in Figure 4.2, rigid cluster decomposition may become problematic. Again, the implied hinge is the reason for the situation: once it is placed in the network explicitly as a constraint, it should be considered part of the cluster, and the cluster becomes rigid by itself.



Figure 4.2: The double-banana graph from Figure 4.1 with one constraint in one of the bananas removed. The red subnetwork is a rigid cluster, but is only rigid due to the left banana and not by itself. However, inserting the implied hinge (green dashed line) explicitly makes the subnetwork rigid by itself.

Another problem due to implied hinges one might encounter is that some rigid clusters may become *non-contiguous*, in the following sense. Consider a *three-banana graph* shown in Figure 4.3. Each banana is rigid by itself, but different bananas are not mutually rigid. Yet, sites 1, 2 and 3, where the bananas "touch" each other, are mutually rigid and form a rigid cluster, with no other sites in the network being rigid with respect to all three sites simultaneously. Since these sites are not connected to each other directly, this rigid cluster is non-contiguous. Note that there are implied hinges 1–2, 2–3 and 1–3 around which bananas rotate, and placing these implied hinges makes the cluster contiguous.



Figure 4.3: The three-banana graph. Sites 1, 2 and 3 form a non-contiguous rigid cluster.

Applying the Laman's theorem (when it works) to the network itself only merely indicates if redundant constraints are present, but does not tell exactly how many redundant constraints there are, so the number of floppy modes cannot be found. This is one reason why, as we know, the pebble game actually starts from scratch and inserts one constraint at a time, checking each of them for redundancy, so that at the end the exact number of redundant constraints is known. What this means is that even if the final network has no implied hinges, but they appear at some intermediate stage in the process of constraint insertion, the counting can be wrong. This certainly makes sense, given that if an implied hinge is placed explicitly, it is always a redundant constraint (since it connects a mutually rigid pair of sites), and redundant constraints are not covered by a pebble and thus are not "seen" by the pebble game, so it is as if the hinge still remains implied. But if rotation around a hinge is locked at some point in the constraint insertion process (see Figure 4.4), so at the end the hinge (either implied or explicit) does not exist anymore, the error in the floppy mode count gets corrected, since the locking constraint is deemed redundant, while in fact it locks a rotation and thus is independent. Thus only those implied hinges that still remain hinges (either implied or explicit) at the end of the insertion process can spoil the floppy mode count. Likewise, if a hinge is locked, there is no danger to rigid cluster decomposition anymore.

One important thing to notice is that the "only if" part of the statement of the Laman's theorem remains valid in 3D. That is, if the number of constraints in any subnetwork does exceed $3N_s - 6$ (N_s is the number of sites in the subnetwork), then there certainly are redundant constraints. Because of this, the pebble game always gives a *lower bound* of the true number of floppy modes. But there is no similar statement for rigid cluster decomposition. For example, in Figure 4.1, there are two rigid clusters rotating around a hinge, whereas the pebble game would determine that there are no internal floppy motions and thus just one single rigid cluster; on the other hand, in Figure 4.2, since the red rigid cluster is not rigid by itself, it may be wrongly found to consist of several clusters.

A few words about rigid clusters in 3D are in order. In connectivity



Figure 4.4: The double-banana graph with hinge rotation locked by a constraint (red).

percolation, a site always belongs to just one cluster. In rigidity percolation, this is no longer the case even in 2D. In Figure 4.5 there are two examples of configurations with sites shared between two rigid clusters and serving as *pivots* around which several rigid clusters rotate with respect to each other. These examples are valid in both 2D and 3D. On the other hand, a constraint in 2D always belongs to just one rigid cluster. In 3D, this is no more the case, and a constraint can be a *hinge* shared between several clusters (Figure 4.6). Note that in 3D *bond-bending* networks, a constraint can be shared between two clusters at most and pivots do not exist at all; moreover, only CF constraints can be shared between clusters (be hinges) and there are no implied hinges. Since pivots do not exist, sites can be uniquely assigned to rigid clusters. But in *general* 3D networks all these restrictions are lifted: there can be pivots, hinges can be shared between more than two clusters and there can be implied hinges; separation into CF and angular constraints does not make sense, of course.



Figure 4.5: Two examples of networks with pivots. Constraints and sites belonging to the same cluster have the same color; pivots shared between several clusters are black. Both examples are valid in 2D and in 3D.



Figure 4.6: A network in 3D with a hinge (black).

At the same time, an *angle* between two constraints stemming from the same site always belongs to just one cluster in 3D (assuming that the angle is rigid, i.e., all three sites forming the angle are mutually rigid). Indeed, consider an angle formed by sites A, B and C. If sites D and E are both rigid with respect to all three sites A, B and C, then D and E are mutually rigid and thus belong to the same cluster. Thus each rigid angle can be assigned to just one rigid cluster, and this assignment, together with the list of constraints that are not part of any rigid angle, as well as isolated sites (if any), fully specifies rigid cluster decomposition of the network. Of course, one should remember that rigid clusters can be non-contiguous, if there are implied hinges, but become contiguous again, once implied hinges are inserted as regular constraints; thus, implied hinges should be considered as regular constraints for rigid cluster decomposition purposes and, in particular, angles formed by implied hinges with regular constraints and with other implied hinges should also be added to the list to obtain full rigid cluster decomposition. Note that insertion of implied hinges as explicit constraints does not change the rigid clusters or the number of floppy modes (but it can change the distribution of stresses).

If somehow the rigid cluster decomposition of the network is known, as well as all implied hinges, then assuming that indeed placing the implied hinges eliminates the problems with the Laman's theorem and non-contiguity of rigid clusters, we can find the number of floppy modes in the following way. First, include all implied hinges in the network explicitly, so all rigid clusters are contiguous. Then, for each rigid cluster, do constraint counting and determine the number of redundant constraints (knowing that since the cluster is rigid, it has exactly 6 floppy modes corresponding to its rigid body motions). The numbers of redundant constraints for all clusters are added up to get the total number of redundant constraints in the network. Then Maxwell counting is done for the whole network, and adding the total number of redundant constraints, we get the exact number of floppy modes. This procedure is illustrated in Figure 4.7 for the two-banana example, for which the pebble game counting fails.

The above procedure also helps to explain the difference between those networks with implied hinges, for which the pebble game floppy mode counting fails (as in Figure 4.1), and those for which it does not fail (as in Figure 4.2). In Figure 4.1, insertion of an implied hinge changes the number of redundant constraints by two (one in each cluster); in Figure 4.2, the change is by one (since only one of the clusters becomes stressed when the hinge is inserted). The change in



Figure 4.7: Illustration of floppy mode counting for the double-banana graph. First, the implied hinge (red) is inserted explicitly. The first cluster (green plus the hinge) has 5 sites and 10 constraints, so Maxwell counting for it gives $5 \times 3 - 10 = 5$ floppy modes, and there must be 1 redundant constraint. Same for the second cluster (blue plus the hinge). Thus there are a total of two redundant constraints, and by adding them to the global Maxwell count for the whole network, the correct number of floppy modes is obtained.

the total number of constraints is by one in each case. Then, according to the recipe for the floppy mode counting outlined above, the net number of floppy modes changes by one in the first case, when the implied hinge is added, and does not change in the second case. This is why in the first case ignoring the hinge influences the floppy mode counting, while in the second case it does not. Generally, one should consider each of the rigid clusters sharing an implied hinge separately, together with the hinge. If the hinge is stressed just once (taken with just one of the clusters), this is the situation analogous to that in Figure 4.2 (the number of floppy modes is not affected, but the rigid cluster decomposition might be). We will call such a hinge *beniqn*. If, on the other hand, the hinge is stressed more than once, this is analogous to Figure 4.1 and the number of floppy modes is affected (as well as rigid cluster decomposition); we will call such a hinge *malignant*. This should be extended to *explicit* hinges as well, since they can be implicit at some point in the constraint insertion process. It is possible that an explicit hinge is not stressed at all (something not possible with an implied hinge, since even without its being inserted explicitly, there is at least one cluster rigid by itself). Such an explicit hinge can never be implied even at an intermediate stage, so it can cause no problems whatsoever. The rest of the explicit hinges are *dangerous* and further classified as benign and malignant as above.

4.2 Rigidity analysis through network relaxation

In this section we propose an algorithm for analyzing rigidity of general 3D networks that is exact in principle. This algorithm identifies implied hinges and thus is able to fix the problems with the pebble game (assuming that indeed, all these problems are due to implied hinges only). Its disadvantage is that it is not an integer algorithm, and thus round-off errors are possible. This leads to bad convergence for bigger networks, especially close to the rigidity transition. The algorithm is also slower than the pebble game. On the other hand, even though the round-off errors can in principle lead to mistakes in rigidity analysis, there are several self-consistency checks that should be able to make the algorithm work perfectly or at least detect the mistakes when they occur. Networks of a few hundred sites can be analyzed *exactly* with ease, even close to the rigidity transition, and bigger networks are tractable away from the transition.

4.2.1 The algorithm

We place atoms in completely random positions within some volume and connect them with springs according to the desired topology. The lengths of the springs are chosen equal to the initial distances between atoms they connect, so initially, the network is unstressed [Figure 4.8,(a)]. We then displace all atoms randomly [Figure 4.8,(b)] and relax the system by the conjugate gradient method [52] [Figure 4.8,(c)]. Assume for now that the displacements before relaxation are *infinitesimal*. After relaxation is complete, the final 3N-dimensional displacement vector (where N is the number of sites) represents one of possible zero-frequency motions. As the initial displacement was chosen at random, it is extremely unlikely that the distance between a pair of sites that are not mutually rigid would be the same at the end as at the beginning (in an unstressed state). On the other hand, for all mutually rigid pairs of sites the distance after relaxation should be the same as before displacements were introduced. Thus mutual rigidity of all pairs of sites can be determined in principle.

In any practical realization, displacements are never infinitesimal. Finite displacements are dangerous, as there is a possibility of "multistability" (existence of several distinct minima of the potential energy to which the network can relax). Of course, we can always choose very small displacements, but it is not known a



Figure 4.8: Schematic illustration of the relaxation procedure. In the network of interest, the spring equilibrium lengths are chosen equal to the initial distances between sites they connect, so initially the network is unstrained (a). All sites are displaced at random (b) and then relaxed (c). The distances between sites 3 and 6 and between sites 4 and 5 in (c) differ from those in (a), so these pairs of sites are not mutually rigid. All other distances are preserved, so all other pairs of sites are mutually rigid. The actual procedure differs from this in that the problem is linearized, so all displacements are always effectively infinitesimal.

priori exactly how small they should be. A better alternative is *linearizing* the problem, so the potential energy is exactly quadratic in displacements from the initial equilibrium. This also helps the conjugate gradient relaxation procedure to converge faster. Then the potential energy is

$$V = \frac{1}{2} \sum_{\langle ij \rangle} [(\mathbf{r}_j - \mathbf{r}_i) \cdot (\mathbf{u}_j - \mathbf{u}_i)]^2, \qquad (4.1)$$

where the sum is over all constraints in the network, \mathbf{r}_i is the radius vector of site *i* in equilibrium, and \mathbf{u}_i is its displacement from equilibrium. As a consequence, the

condition that sites 1 and 2 are mutually rigid becomes

$$(\mathbf{r}_2 - \mathbf{r}_1) \cdot (\mathbf{u}_2 - \mathbf{u}_1) = 0, \tag{4.2}$$

instead of the previous condition that the distance is unchanged. This can be thought of as making all displacements infinitesimal in effect.

In theory, the conjugate gradient procedure is exact for a linear system with n degrees of freedom after n^2 iterations [52]. In practice, because of round-off errors, sometimes it takes more iterations, especially close to the rigidity transition. Preconditioning [53] may improve convergence, but this has not been done so far. Inevitably, the calculations of the potential and the scalar product in Eq. (4.2) are themselves subject to round-off errors, and they are especially severe, because, as the form of these expressions suggests, these calculations often involve subtraction of two big numbers that is supposed to give a small number. Say, if a rigid cluster is being translated and/or rotated, the displacements \mathbf{u}_i are big, but the scalar products should be zero. Because of this, the approach eventually fails for bigger system sizes, especially close to the rigidity threshold, where there are big regions that are close to being isostatic (either on the floppy or on the rigid side), with probably the only possible solution being increasing the numerical precision (i.e., going to quadruple precision numbers, with significant costs in computation time).

To analyze a particular graph, we run at least two realizations, with different equilibrium positions of sites and initial random displacements (but the same connectivity as needed, of course). This is done to avoid situations, in which the change in distance is occasionally close to zero numerically even for pairs of sites that are not mutually rigid, because of special initial conditions (or because the initial configuration was occasionally chosen close to a non-generic one). For each pair of sites, the sum over realizations of absolute values of the scalar product of



Figure 4.9: The logarithm of the sum over two realizations of the absolute value of the scalar product, Eq. (4.2), for all pairs of sites of a network with 216 sites. The gap between "zero" and "non-zero" values is clearly seen. Pairs with the values below the gap are mutually rigid, those with the values above the gap are not.

Eq. (4.2) is calculated. There should be a well-defined gap between "zero" and "non-zero" values. An example of the result is shown in Figure 4.9, where the gap is clearly seen. If the gap has not formed, more realizations are run, until the gap reaches at least one order of magnitude.

Sometimes, especially close to a very sharp rigidity transition, where the network goes from being almost completely floppy (with only small rigid clusters) to being almost completely rigid, there is a possibility of having two gaps in the distribution, so if the cutoff is chosen inside one gap, the network is identified as floppy and if the cutoff is inside the other gap, it is rigid. This can often be misleading, but the self-consistency checks described below, as well as running the pebble game once "dangerous" hinges are identified (also described below) can help identify the correct gap and check the validity of the results.

Once all mutually rigid pairs of sites are found, we can find all implied hinges. This is done as follows. First, all sites whose neighbors are not all mutually rigid are found — only such sites can be hinge endpoints. This is not an essential step, but doing this speeds things up. Next, among such sites, we find pairs of sites that are mutually rigid, but not directly connected (as we are looking for *implied* hinges). For each such pair of sites, A and B (Figure 4.10), we pick a site C rigid with respect to both A and B. Then we go through the list of all sites rigid with respect to both A and B, and if at least one of them, site D, is not rigid with respect to C, then A-B is an implied hinge, otherwise it is not an implied hinge.

We now place all implied hinges explicitly in the network as regular constraints — we recall that this does not change the number of floppy modes or the configuration of rigid clusters. We assume that now that all implied hinges are placed explicitly, the two problems with rigid clusters mentioned above are eliminated: all rigid clusters are contiguous and rigid by themselves. The contiguity can be checked explicitly (since we know all pairs of mutually rigid sites), and this serves as one self-consistency check for the algorithm. Once rigid clusters are contiguous, the full rigidity information contained in the list of all mutually rigid pairs of sites can be replaced by labeling all rigid angles in the network, so that angles belonging to the same rigid cluster have the same label. This is a more conventional way of representing rigid clusters. We recall that this is possible, because a rigid angle can never be shared between several rigid clusters.

Now that all rigid clusters are rigid by themselves and contiguous, the number



Figure 4.10: Finding an implied hinge. Candidate endpoints of an implied hinge (A and B) have to be rigid and not directly connected. An arbitrary site C rigid with respect to both A and B is chosen. If there exists a site D also rigid with respect to both A and B, but not rigid with respect to C, then A–B is a hinge; otherwise it is not a hinge.

of floppy modes can be found by counting redundant constraints in each cluster, as described in the previous section. Note that the number of redundant constraints in each rigid cluster should be non-negative, and this serves as yet another self-consistency test of the algorithm.

Also, we can find what hinges (both implied and explicit) can cause problems with the pebble game, according to the analysis in the previous section, by using the counts of redundant constraints in each rigid cluster. If none of the clusters sharing an explicit hinge has redundant constraints, the hinge is definitely unstressed and is no danger whatsoever (this cannot happen to an implied hinge). Note that this is a sufficient, but not a necessary condition for the hinge to be unstressed, since it is possible that the hinge is unstressed and yet there is stress (and thus redundant constraints) elsewhere in one of the clusters sharing the hinge. All other hinges (remaining explicit and all implied hinges) are potentially (but, in case of explicit hinges, not necessarily) dangerous. They can be either benign or malignant. If just one of the clusters sharing a hinge has redundant constraints, the hinge is either benign or not dangerous at all. If more than one of such clusters have redundant constraints, then the hinge *may* be malignant and spoil the floppy mode count by the pebble game. But again, there is a possibility that stress associated with redundant constraints is located elsewhere and the hinge is actually benign or not dangerous at all. The good news is that all malignant and all dangerous hinges are certainly detected — but there might be a few "false alarms", and of course, even a potentially malignant explicit hinge may not influence the floppy mode count if inserted early enough, so it is never an implied hinge even at an intermediate stage in the constraint insertion process.

The algorithm we have described is capable of doing two of the analyses done by the pebble game: the floppy mode counting and the rigid cluster decomposition. To have the full functionality of the pebble game (at least for those network sizes that can be studied), we also need to be able to do stressed region decomposition.

It is possible at least in theory to find stressed constraints in the network by doing the same network relaxation procedure, but lifting the requirement that the lengths of constraints are chosen so that they are unstressed in the initial configuration. Then, after relaxation, constraints that remain strained are stressed constraints. In practice, however, we have encountered problems doing this, because of rampant round-off errors. Besides, this only allows finding stressed constraints, but not grouping them into *stressed regions*. We recall from Chapter 1 that stressed regions are defined as sets of stressed constraints such that deformations of all constraints within the set are interdependent, whereas deformations of constraints outside the set do not influence the constraints in the set. In the pebble game procedure, stressed regions are determined naturally by merging failed pebble search regions. It seems like the best choice then to find stressed regions by running the

pebble game correcting it using the information about dangerous hinges provided by the relaxation procedure. The idea is to insert all malignant implied and explicit hinges explicitly at the beginning of the constraint insertion process, so as to ensure that they are covered by pebbles and thus are "seen" by the pebble game. This ensures that, say, all constraints in Figure 4.1 are found stressed, which they are, whereas, as we know, the pebble game finds this graph to be isostatic, if the implied hinge is not inserted explicitly. Note that it is very important *not* to insert implied hinges that are not malignant. Indeed, if the benign hinge in Figure 4.2 is inserted explicitly, the left banana becomes stressed, but should be identified as unstressed. This presents us with a problem, since the relaxation procedure only gives a list of potentially malignant hinges, but does not specify exactly which of them are actually malignant. Therefore additional testing has to be done for every potentially malignant hinge. This can be done by seeing for every potentially malignant hinge if placing it at the beginning and at the end of the constraint insertion process makes a difference in the floppy mode count. A faster process can probably be developed as well. The details of the procedure still have to be worked out. Another issue is that the stress determination procedure (and the whole pebble game procedure) has to be modified somewhat compared to the case of BB networks, where some shortcuts based on distinction between CF and BB constraints were possible. This will be described in detail in section 4.3.

There is a possibility that the above procedure for finding stressed regions can fail, because there are so many malignant hinges that even by themselves, without any other constraints, some of them would be redundant and not covered by a pebble. However, given that, as we will see in the next sections, the malignant hinges are quite rare, this is unlikely to happen and will be detected anyway.

Another possible universal approach to rigidity is based on numerical diagonalization of the dynamical matrix for the spring network. The number of zero

eigenvalues is the number of floppy modes and the floppy modes themselves can be found, from where it is possible in principle to obtain rigid cluster decomposition. The computational time should be similar to that for relaxation. We have not done any comparisons between relaxation and diagonalization yet. We believe, however, that relaxation is more reliable, since the crux of both algorithms is distinguishing "zero" from "non-zero" values (of the eigenvalues for diagonalization and of the scalar product in Eq. (4.2) for relaxation); but there are $O(n^2)$ scalar products and only O(n) eigenvalues, so there is better statistics in the case of relaxation to determine the proper value of the cutoff. Yet, diagonalization has the advantage that the number of floppy modes can in principle be determined directly, without any assumptions about the role of implied hinges; so studies of the diagonalization procedure and its comparisons with relaxation are certainly desirable.

4.2.2 An example: colloidal glass networks

As an example of application of the relaxation algorithm, we describe rigidity analysis of colloidal glass networks obtained in experiments of D.A. Weitz *et al.* [54, 55, 56]. They study colloidal solutions of PMMA spheres ~ 1 μ m in diameter. Confocal microscopy is used [57] to image the system (Figure 4.11) and pinpoint the location of each particle as a function of time, so that the structure and dynamics can be studied in very great detail. By adding a polymer to the solution, they can induce depletion attraction [58] between the spheres, whose strength and range can be varied. Depletion attraction is due to the fact that the excluded volume for the polymer molecules is smaller and thus the entropy is bigger, when spheres are closer to each other than the size of the polymer molecules. Thus this attraction is entropic in origin. When attraction between spheres is weak, the interaction is essentially hard-core repulsion, and we have a system of jammed particles forming a colloidal glass when the volume fraction of the spheres is ~ 50% [55]. When the

attraction is stronger, a gel can be formed even when the volume fraction of the particles is just ~ 5% [54]. So far we have dealt with the results for weakly attracting particles (provided by J.C. Conrad [59]). In this case, it is a nice model system for ordinary glasses, where dynamics on the "molecular" level can be studied in detail. As is usual in liquids and glasses, a particle usually remains jammed in a particular location for a relatively long time, after which a local rearrangement occurs. One can find sets of "slow" particles that do not change their neighbors during a specified interval of time and study properties of networks formed by such particles assuming that all first neighbors are connected by a constraint. The word "neighbors" is understood here in a purely geometrical sense, i.e., neighbors are determined by building Voronoi polyhedra around the particles (two particles then are neighbors, if their Voronoi polyhedra share a face). Weitz and his group are interested, in particular, in studying rigidity of such networks. It is not clear if rigidity, as we understand it, has any physical meaning here, since geometrical neighbors do not necessarily interact. So far, we simply consider these networks as good tests for our rigidity algorithms. Perhaps rigidity analysis is more meaningful for colloids with stronger attraction forming gels. The attraction is short-range and there is a definite cutoff distance for it (basically determined by the size of the polymer molecules), therefore it is meaningful to say what particles interact and what particles do not, so the definition of neighbors and constraints is more physically motivated.

In any case, here, as an example, we present the results of rigidity analysis using the relaxation procedure for two of the networks, for different time intervals. As these networks are relatively small (up to 3000 atoms) and have no big regions close to being isostatic, applying the relaxation method was no problem whatsoever. As we know, in 3D it is angles that are uniquely assigned to rigid clusters. However, if, in order to show rigid cluster decomposition, we attempt to color angles, the figure



Figure 4.11: A confocal microscope image of a colloidal glass [60].

will be a real mess. Therefore we color constraints instead: constraints that belong to just one rigid cluster are painted some color, with adjacent constraints belonging to the same cluster painted the same color; constraints belonging to more than one cluster (i.e., hinges) are black. Non-adjacent clusters are allowed to have the same color, as otherwise there are not enough visually distinct colors. We also show implied hinges with dashed lines. Sites are colored if all constraints stemming out of it belong to the same cluster; otherwise they are left black. This coloring scheme can be somewhat misleading. Figure 4.12 gives an example of a rigid cluster (a tetrahedron) that has just one constraint colored, while the rest of its constraints are hinges shared with other clusters, so the true size of the rigid cluster is not revealed. This situation is somewhat extreme, though. In our case, the coloring described above is adequate, as bigger clusters that are of interest have relatively few hinges.



Figure 4.12: A rigid tetrahedron with many hinges (black) and one non-hinge constraint (red). Such rigid clusters do not show up well when the visualization scheme described here is used.

In Figures 4.13 and 4.14, we show two examples of rigid cluster decomposition with coloring as above. When the chosen time interval is short, the network is bigger and more rigid (Figure 4.13). There are big rigid clusters. When the time interval is longer, it is smaller and floppier, with fewer big rigid clusters (Figure 4.14).

As these networks are relatively small (up to 3000 atoms) and have no big regions close to being isostatic, applying the relaxation method was no problem whatsoever. It is interesting to note that out of 16 networks that were analyzed, just two had one potentially malignant hinge each; others had no malignant hinges. So at least the floppy mode count by the pebble game would be perfect or nearly perfect. We find a similar result for a different class of networks in the next section.



Figure 4.13: Rigid cluster decomposition for a colloidal glass network obtained as described in the text for the time delay 252 s. The network appears almost flat, as it is rather thin, but is treated as three-dimensional. Bonds and sites belonging to just one cluster are colored, otherwise they remain black. Adjacent bonds/sites belonging to different rigid clusters have different colors. Implied hinges are shown with dashed black lines.



Figure 4.14: Same as in Figure 4.13, but for the time delay 3240 s.

4.3 Pebble game for general 3D networks

Given that there is no exact integer algorithm analogous to the pebble game for general networks so far, and use of the relaxation algorithm is only limited to relatively small networks, the question of how good the straightforward pebble game is typically is of practical interest. In this section we first describe the specifics of application of the pebble game to general 3D networks. Then to test the pebble game, we study BB networks, by inserting constraints in an arbitrary order, instead of the proper order of insertion giving the correct result. More tests of the pebble game will be done in the next section.

4.3.1 The algorithm

The basis for the pebble game algorithm for general (non-BB) networks is still the straightforward generalization of Laman's theorem (or, one might say, of the molecular framework conjecture) to 3D non-BB networks as if it were perfectly valid. So the algorithm essentially remains the same, with a few complications due to the fact that some shortcuts used for BB networks are no longer possible.

As in the BB algorithm, constraints are inserted one by one. However, since there is no distinction between CF and BB constraints, there are obviously no special requirements of the order of insertion (recall that in the BB algorithm, all induced angular constraints were inserted immediately after every CF constraint). Still, if the network is of partially BB character, say, some, but not all BB constraints are present, preserving partial order may help reduce errors. Just as in the BB case, six pebbles are collected at the just inserted constraint's ends (three at each end) — this should always be possible — and after that, freeing one pebble is attempted at each neighbor of one end of the constraint in turns. The difference is that since there is no distinction between CF and angular constraints, all constraints are treated on equal footing, so all neighbors of the end of the constraint are checked (in the BB case, it was enough to check just those neighbors connected with a CF constraint). If all attempts of freeing a pebble are successful, the new constraint is independent and is covered by one of the six free pebbles at its ends; if at least some attempts failed, the constraint is redundant.

The stressed region determination is slightly more complicated than for BB networks. In the BB case, once pebble search at one of the neighbors failed, the region of the failed search was merged with the previously found stressed regions to produce a stressed region, and further pebble searches (at other neighbors) were not needed, since they would all produce the same region. Now all pebble searches at all neighbors have to be done, and the *intersection* of all regions of failed searches is found before merging with the previously found stressed regions. The justification for this procedure is as follows. First of all, as it was mentioned, the pebble game does not "see" redundant constraints, since they are not covered by a pebble. Thus if a test constraint that is being inserted turns out to be stressed, the region of the network, in which it is inserted, is isostatic for the pebble game purposes. The stressed region created when a single redundant constraint is inserted in an isostatic network is the set of constraints such that removal of any of them makes the newly inserted constraint independent. In other words, when any constraint belonging to the stressed region is removed and its associated pebble is freed, this freed pebble should be able to move to every neighbor of the ends of the inserted constraint where the pebble search failed, to become available at each neighbor, so the test constraint is no more redundant. This is the case, if and only if the site to which the freed pebble belongs is accessible to pebble search from every such neighbor and thus lies in every failed pebble search region.

The merger process for stressed regions is actually similar to the 2D case: for two regions to be merged, there should be at least a two-site overlap. In 3D BB

networks, all neighboring regions have at least two site overlap (just as all rigid clusters have at least two-site overlap, because there are no pivots).

The rigid cluster decomposition procedure is more subtle. In the BB case, an angle between CF constraints is taken, six pebbles are collected at its vertices and then the region where the seventh pebble cannot be found is the rigid cluster; then the next angle not in any rigid cluster found so far is taken and so on. This was based on the fact that in a BB network, any angle between two CF bonds is rigid. In a non-BB network, the essence of the procedure is the same; however, the above should be done for all angles (obviously, there is no distinction between CF and BB constraints) and not all angles are rigid. So before the above procedure, the angle should be tested for rigidity. There are several ways of doing so, and none are entirely reliable, when implied hinges are present. One possible procedure is as follows. Suppose the angle is ABC and its vertex is B. Then to determine if the angle is rigid, we imagine inserting a constraint AC and testing it for redundancy. If it is redundant, the angle is rigid, otherwise it is not rigid. This procedure has an advantage that it gives correct results for some benign dangerous hinges, such as in the case in Figure 4.15, when the angle ABC is not rigid by itself, but only rigid due to the shaded rigid cluster, and the above procedure still correctly identifies it as rigid. We call such benign hinges *trivial*. It turns out that often a vast majority of dangerous hinges are trivial, so the above procedure solves a lot of problems due to implied hinges. But other variants are possible and may be better. This issue deserves further study.

4.3.2 Test for bond-bending networks

One estimate of how big the problems with the pebble game could be is provided by studying ordinary BB networks. Of course, the pebble game is *exact* for such networks, *if* the proper ordering of constraints is obeyed, i.e., after each CF



Figure 4.15: A schematic drawing of a trivial benign implied hinge. A shaded body is a rigid cluster. Three blue sites and two red bonds form a rigid cluster that, although not rigid by itself, is simple enough to be taken care of in the rigid cluster decomposition procedure. The implied hinge is the green dashed line.

constraint, all induced BB constraints follow immediately, then the next CF constraint is placed, etc. However, if constraints are inserted in random order, implied hinges may appear, so there will be errors, which can be found by comparing with the exact result. This can hopefully give an idea of how big the errors can be in other, non-bond-bending networks.

The simplest way of creating an implied hinge is shown in Figure 4.16. In the figure, there is a 4-coordinated site, and one of the CF constraints stemming from it is a hinge. If this constraint is inserted last, after all other constraints, it is going to be an implied hinge just before being inserted. In this case, it is a benign hinge (moreover, a trivial one). A configuration built in the same spirit, but with a malignant hinge, is shown in Figure 4.17 and consists of two 4-coordinated sites with a hinge between them. Note that without the hinge, this is nothing but the two-banana configuration of Figure 4.1. In any case, it is clear that, while there are

many hinges in any BB network below and just above the rigidity threshold, only a relatively small fraction of them (namely, those that are stressed) are dangerous and can be treated as implied by the pebble game if inserted late; but even fewer are malignant. It is also clear that the problem is going to be most significant near the rigidity threshold, as far below the threshold, most hinges are unstressed, and far above, there are very few hinges.



Figure 4.16: A configuration with a trivial benign hinge shared by two clusters (blue and green) that can arise in bond-bending networks. Central-force constraints are solid lines, angular constraints are dashed lines.



Figure 4.17: A configuration with a malignant hinge shared by two clusters (blue and green) that can arise in bond-bending networks. Central-force constraints are solid lines, angular constraints are dashed lines. Without the hinge, this is topologically equivalent to the double-banana graph in Figure 4.1.

To get specific estimates, we have considered a usual model for studying 3D rigidity in glass networks — a BB diamond lattice diluted in such a way that no sites of coordination lower than 2 ever appear. The algorithm used is described in the previous subsection.

The results are shown in Figure 4.18 and indicate that the error in the number of floppy modes occurs mostly in the region just below the transition, is only about 1% at its maximum, and seems fairly independent of the network size. In the region where the error is biggest, it is still only about 20% of the deviation of the exact result from Maxwell counting, so the pebble game procedure, albeit approximate, offers a significant improvement over Maxwell counting.



Figure 4.18: The relative deviation of the floppy mode count obtained with constraints inserted at random in the pebble game procedure from the correct result obtained with the proper order of insertion. Filled green circles are for networks of 1000 sites averaged over 100 realizations. Open red circles are for networks of 8000 sites averaged over 10 realizations.

As for rigid cluster decomposition, it is not as successful as the floppy mode

counting — not surprising, given that there are many more benign hinges than malignant ones (even though the majority of the benign hinges are trivial ones and thus should not matter). Still, the results are decent, especially if one is interested just in the size of the percolating cluster (moreover, we have some indications that the results for the size of the percolating cluster may be exact with some variants of the decomposition procedure).

4.4 Central-force networks — first-order transition

We now apply the methods developed so far to studying rigidity of CF networks obtained by bond dilution of regular lattices.

First of all, the natural first step is applying Maxwell counting. For a network of N sites there are a total of 3N degrees of freedom and the number of constraints is $\langle r \rangle N/2$. Then the number of floppy modes is $F = 3N - \langle r \rangle N/2$ and the rigidity transition is at the point where this becomes zero, i.e., at $\langle r \rangle = 6$.

We now consider two cases: truly three-dimensional networks and networks that are two-dimensional, but with sites able to move in three dimensions. The latter is obviously a problem in 3D rigidity as well.

4.4.1 Topologically 3-dimensional networks

The above result for the location of the rigidity transition means that we need lattices with coordination more than 6 to study the transition. The natural candidates are the bcc lattice with coordination 8 and the fcc lattice with coordination 12.

Studying rigidity of bcc and fcc lattices with the relaxation method for moderate lattice sizes (500-1000 atoms) accessible to the method, we find a very

curious result: below the rigidity transition, only very small rigid clusters are present, namely, isolated sites, single bonds and also triangles in the fcc lattice (bcc has no triangles). All these clusters are isostatic, so below the transition there are no redundant constraints and thus Maxwell counting is exact. At the rigidity transition, the percolating rigid cluster emerges and all other rigid clusters are still the same small ones (except there are now triangles formed by two bonds and a trivial benign implied hinge in the bcc as well, as depicted schematically in Figure 4.15). This strongly resembles the situation with the random bond model (see Chapter 2) and indeed, the transition seems to be *first order*. Besides, we see no malignant hinges and all benign ones are trivial. This is quite natural. Indeed, there is only one cluster in the network that can be stressed (the percolating one), and a hinge must be shared between more than one stressed cluster to be malignant; since all non-percolating clusters are triangles at most, the benign hinges can only be trivial. If we assume that these statements also apply to bigger networks, we can use the pebble game to study those bigger networks that cannot be studied by relaxation. The results for the number of floppy modes will be exact, since there are no malignant hinges; the rigid cluster decomposition should also be exact, since all benign hinges are trivial.

Of course, in reality the probability of having a bigger finite rigid or even stressed cluster is non-zero. For example, for the fcc lattice, if we take an atom and its first coordination shell with all bonds between them, there are 13 atoms and 36 constraints, so Maxwell counting gives $13 \times 3 - 36 = 3$ and there are thus 6 - 3 = 3redundant constraints. However, the probability of having all 36 constraints (or at least 33 of them that would make the cluster isostatic) present is, of course, very low close to the transition, which, according to Maxwell counting, should be located close to the bond probability p = 1/2. Indeed, the probability of having 33 to 36 constraints present is ~ 10^{-7} . For the bcc lattice, a rigid cluster has to involve more

coordination shells, so the probability is apparently even lower. Plus, the probability of a non-trivial dangerous hinge is even lower than that of a bigger rigid cluster, so we can safely neglect this possibility and use the pebble game for our studies.

The results of the pebble game runs for bigger networks (up to 62500 sites) are in Figure 4.19 for the bcc lattice and in Figure 4.20 for the fcc lattice. Indeed, Maxwell counting is exact below the transition, as expected, there are still no non-percolating rigid clusters bigger than triangles, and the transition clearly looks first order and overall the picture is qualitatively very similar to that for the random bond model (Chapter 2). The jumps in the size of the percolating rigid cluster and the percolating stressed regions are very big: their sizes are all about 90% of the network or more.

Given that the rigidity transition is apparently first order for purely CF networks in 3D, one could expect that the elastic constants of these networks exhibit a jump at the transition. The old results by Feng, Thorpe and Garboczi [61] (Figure 4.21) indicate that this is not the case, and, in fact, the elastic constants go continuously to zero at the transition, just as in bond-bending networks, where the transition is second order. Thus the stressed backbone, although big, is very fragile close to the transition.

We have also done some preliminary studies of partially bond-bending networks obtained by adding some of the angular constraints between bonds, so that each angular constraint has a probability s of being present. Probability s = 0corresponds to just described purely CF networks; s = 1 means purely BB networks. We know that the transition is first order for s = 0 and second order for s = 1, and it is interesting to see how the crossover from first to second order happens. Knowing from the previous section that for BB networks the pebble game algorithm even with constraints placed at random is very accurate and for CF networks there are hardly any errors at all, we expect the algorithm to be adequate for all



Figure 4.19: The number of floppy modes per degree of freedom in the top panel and the sizes of the percolating rigid cluster (green) and stressed region (red) in the bottom panel for the bond diluted central-force bcc networks. Averages over 10 realizations on networks of 54000 sites. The dashed line in the top panel is Maxwell counting.


Figure 4.20: Same as in Figure 4.19, for the fcc networks. Averages over 10 realizations on networks of 62500 sites.



Figure 4.21: The elastic moduli for the FCC network (figure adapted from [61]). Quantity p in the graph is the fraction of present bonds; thus $\langle r \rangle = 12p$.

situations in between. Moreover, we expect to even improve it by trying to be close to the proper order of placing constraints for BB networks, i.e., placing angular constraints whenever present immediately after CF constraints that induce them. This makes the algorithm exact in the limit s = 1 and virtually exact in the limit s = 0. Our preliminary results indicate that apparently, the transition is second order for all $s \neq 0$ but gets sharper as s approaches zero. However, this is far from certain and more studies are needed. In fact, it should first be established that even for s = 0 we really have a discontinuous transition and not just an extremely sharp continuous one. This can probably be done by Monte Carlo sampling of (extremely rare) big but finite rigid clusters close to the transition.

4.4.2 Topologically 2-dimensional networks

We now look at topologically two-dimensional lattices embedded in 3D so the sites can move in three dimensions. None of the regular 2D lattices have coordination higher than 6, so we use a square lattice with both main bonds and diagonals of squares as possible constraints (Figure 4.22). Such a network has the full coordination of 8. We assume that the pebble game is adequate for these lattices as well, even though the probability of finite rigid clusters bigger than triangles is definitely higher, as we will see.



Figure 4.22: A square lattice with diagonals used here in the study of a topologically 2D network in 3D.

The preliminary results (Figure 4.23) are very different from those for topologically three-dimensional CF networks and look similar to those for bond-bending networks. Maxwell counting is no longer exact and the transition is apparently second order.



Figure 4.23: The number of floppy modes per degree of freedom for the diluted square network with diagonals (as in Figure 4.22). Average over 10 networks with 40000 sites.

4.5 **Prospects for an exact algorithm**

As we have seen, the pebble game algorithm is usually quite successful even for 3D CF networks, for which it is not supposed to work in principle. Still, special situations when it is not as good are certainly possible, so one would like to have an analogous *exact* algorithm. So far this goal is not achieved, unfortunately. A less ambitious goal (not reached as well so far) would be to be able to put exact bounds on, say, the number of floppy modes (the pebble game provides a lower bound, and we would like to have an upper bound as well).

A promising approach to the problem was proposed recently by D.J. Jacobs. I outline this proposal here even though I can take no credit for it, to make the picture complete and since further development of these ideas is my goal for the near future. So far the proposal only deals with finding the number of floppy modes and not with rigid cluster decomposition.

The idea is to start with a *bond-bending* network consisting of the same sites as the original network of interest and constructed so that all extra constraints (those not in the original network) are angular constraints. The simplest way is, of course, to start with the original network as a CF network and add all induced angular constraints (but this may not be the most economical way, since, as we will see, the fewer extra constraints the faster it is to get the result). The pebble game is then run on the BB network, and since the network is bond-bending, we know that the procedure is *exact* for that network. The idea now is to try to remove all those extra induced constraints whose removal does not change the number of floppy modes. Since the possible floppy motions do not change in the process, no implied hinges will ever be created. Once the process is complete, all remaining induced extra constraints are independent and removal of any of them adds one floppy mode. Since an independent constraint cannot become redundant upon removal of other constraints, so that removal of all of them adds as many floppy modes as constraints that were removed, the number of such extra constraints should be added to the number of floppy modes to get the *exact* number of floppy modes in the original network.

First of all, we can easily remove all those induced extra constraints that are not covered by a pebble. These are definitely redundant and can be removed without any reservations. On the other hand, those induced constraints that are not stressed definitely cannot be removed — their removal will introduce an extra floppy mode. Remaining induced constraints (i.e., those that are stressed yet covered by a pebble) require more careful handling. The idea is to try to move the pebble from the induced constraint to one of the uncovered constraints belonging to the original network. Then the induced constraint gets uncovered and can be removed. This is a

legitimate move, but only provided that an implied hinge is not created in the process, otherwise the pebble game does not work anymore. That is, there has to be a test of whether the induced constraint locks a rotation around some implied hinge, in which case the above procedure is not allowed and the induced constraint definitely cannot be removed, as it locks a rotation that would appear once the constraint is removed. So far all proposed tests have failed to account for all possible situations, so this remains the major stumbling block.

Note that if we settle for a less ambitious goal of creating an algorithm that gives the upper bound of the number of floppy modes, it is enough to have a test that would identify all those situations where a constraint removal is impossible, but may miss some of situations where it is possible. Of course, such a test needs to correctly identify a vast majority of these situations, otherwise the upper bound will be not very good and thus essentially useless. In the extreme case, we can deem all covered induced constraints unfit for removal, but this would usually give a ridiculously high upper bound. A better test has not been devised as well so far.

4.6 Conclusion

In this chapter we have described our studies of rigidity of general (non-bond-bending) 3D networks. There were essentially two broad themes. The first theme is algorithm development. We have described an exact algorithm based on network relaxation, as well as the ongoing work on a faster and more reliable algorithm based on the pebble game. The second theme is studying specific systems. We have first shown that in many cases, the original pebble game algorithm is very accurate and in some cases virtually exact and then used the pebble game to study the rigidity transition in 3D networks, showing numerically that the transition is first order for topologically three-dimensional networks and strongly resembles the transition in the random bond model; the transition is apparently second order in topologically 2D networks.

In the future, we hope to make progress on both themes. On the first theme, we hope to develop a fast integer algorithm for general 3D networks that is exact or at least gives a useful upper bound for the number of floppy modes. We also hope it will be possible to extend it to the rigid cluster decomposition. We will also continue trying to improve the relaxation algorithm, as well as do some comparisons with diagonalization. On the second theme, we would like to have a better proof that the rigidity transition is indeed first order in 3D CF networks and also study the crossover from the first to the second order transition as the network changes its character from central-force to bond-bending.

Chapter 5: Self-organization

We have seen that we can analyze rigidity of a given network quite straightforwardly using the algorithms described in Chapters 1 and 4. On the other hand, when trying to study realistic systems, such as covalent glasses, the natural question is how the networks themselves are modeled and how their peculiarities influence their rigidity properties. As we have seen, sequences of glass networks of varying mean coordination are commonly obtained by starting with a fully coordinated network (say, a fully 4-coordinated Si network, either crystalline or amorphized by the WWW algorithm [see subsection 1.2.5] or its variants [62]) and then diluting this network at random. This may not be entirely adequate, of course, as often energy considerations that favor certain structural arrangements are important and this leads to the network being non-random in various ways. One might think about building the networks from first principles. Unfortunately, various ab initio approaches (such as Car-Parrinello [63]) produce models that are too small (only ~ 100 atoms) for most purposes. The small size, coupled with the periodic boundary conditions, leads to spurious internal strain. Modifications based on linear-scaling electronic structure calculations [64] or molecular dynamics with empirical potentials [65] can alleviate the size problem, but are still too slow to achieve full relaxation of the network at appropriate temperatures in reasonable time. Some hopes are being placed on new methods of accelerated dynamics, such as the activation-relaxation technique by Barkema and Mousseau [66] and other methods [67, 68]. At present, however, our goals are more modest: we can introduce various aspects of non-randomness "by hand" to see what effects they can have on the properties of networks. In Chapters 2 and 3 we have already studied how one such aspect, chemical ordering, can influence the properties and location of the rigidity transition. In this chapter we consider another model that tries to take into

account more subtle effects of non-randomness that we describe as *self-organization*, as the glasses organize themselves in such a way as to minimize their free energy at the temperature of glass formation.

We start with the description of our model of self-organization and study this model first for 2D central-force networks and then for 3D bond-bending networks (obviously more relevant to glasses) in section 5.1. In both cases we find that there are two phase transitions instead of one and an *intermediate phase* between them that is rigid but unstressed. We then consider an analogous model for *connectivity* percolation in section 5.2. This model, as it turns out, was in part introduced and considered before (and, in fact, rediscovered several times), but we add new aspects to it and in particular, in section 5.3 we study how the conductivity of the corresponding random resistor network depends on the bond concentration above the upper boundary of the intermediate phase. We find numerically that this dependence is surprisingly close to linear and might well be *exactly* linear, just like in the mean-field theory of connectivity percolation [69]. We were not able to prove or disprove this linearity for this particular model, but we have indeed proved this linearity for a closely related model. In the same section we also look at elastic properties of self-organized networks in the rigidity case finding that the network is marginally rigid in the intermediate phase having zero elastic constants in the thermodynamic limit. Finally, in section 5.4 we review some possible experimental evidence of two transitions and the intermediate phase.

Many of the results presented in this chapter were published in Refs. [70, 71, 72].

5.1 Self-organization in rigidity percolation

5.1.1 Description of the model

We have seen in Chapter 1 that starting from an "empty" lattice (without bonds) and adding one constraint at a time, we can use the pebble game to analyze whether the constraint we are adding is independent of those already in the network or redundant. We also know that redundant constraints cannot be accommodated without changing the natural bond lengths and angles of the network and so stressed (overconstrained) regions would be created. Thus within the present approach we have a rather unique opportunity to construct stress-free networks without a huge computational overhead.

The idea is to start, as before, from an "empty" lattice and add one bond at a time to it (meaning the corresponding central-force constraint, as well as all associated angular constraints in the bond-bending case), applying the pebble game at each stage. If adding a trial bond would result in that bond being redundant and hence create a stressed region, then that move is abandoned. Thus the network self-organizes in such a way that there is no stress in it at all. Note that the pebble game now serves not only as a tool to analyze the network, as before, but also as a decision-making mechanism when building the network.

It is certainly not possible to keep adding bonds up until all bonds are inserted without introducing stress. How should we proceed once stress becomes inevitable? While going on with some sort of self-organization would be reasonable (as some bonds would create less stress than others), it is impossible to analyze this within our approach, so we start inserting bonds completely at random, once avoiding stress becomes impossible.

5.1.2 General properties

First of all, how long is it possible to keep adding bonds to a network without introducing stress? It is certainly impossible to have more independent constraints then there are degrees of freedom in the network. Now recall that in the Maxwell counting approximation, the rigidity transition occurs when the numbers of constraints and of degrees of freedom balance. Thus it is certainly not possible to have an unstressed network with the mean coordination above where Maxwell counting predicts the transition (that is, above the mean coordination $\langle r \rangle_c^M = 4$ for central-force (CF) networks in 2D and $\langle r \rangle_c^M = 2.4$ for bond-bending (BB) glass networks in 3D). This provides an upper limit (still not always reachable, as we will see) for the unstressed networks. Note, though, that since the Maxwell counting percolation limit is not exact, this does not mean that rigid networks are necessarily stressed! The actual rigidity transition may occur below the point where Maxwell counting puts it. This is a very important point that leads to possibility of an *intermediate phase*, as described below.

Secondly, we know that the Maxwell counting result for the number of floppy modes would be exact if all constraints in the network were independent. But this is exactly what we have in our case! Thus the number of floppy modes in Maxwell counting is exact for as long as we are able to keep the network unstressed. Hence we follow the Maxwell result for the number of floppy modes before reaching the point where stress becomes unavoidable.

We now analyze some specific cases in more detail.

5.1.3 Intermediate phase in 2D central-force networks

Let us first prove that it is actually possible to reach the Maxwell counting limit without any stress in this case (and for any CF networks), provided that the fully coordinated (undiluted) network has no floppy modes (which is certainly the case, say, for triangular networks). In the case of CF networks each bond has only one associated constraint, so "bonds" and "constraints" are identical. Recall once again that every single constraint can be either independent (in which case it reduces the number of floppy modes of the network by 1), or redundant (so it does not change the number of floppy modes). Now, assume the opposite of our statement. This means that at some $\langle r \rangle_0 < \langle r \rangle_c^M$ we have an unstressed network, but any trial bond would cause stress (be redundant). So any bond would not change the number of floppy modes, as the only constraint associated with the bond is redundant. We know that since the network is unstressed, the number of floppy modes given by Maxwell counting is exact, thus the number of floppy modes per site f > 0 for $\langle r \rangle_0 < \langle r \rangle_c^M$. If we try a constraint at some point and it turns out to be redundant, it will certainly remain redundant upon trials at any later point (i.e., after some other bonds are inserted). Therefore even inserting all of the remaining bonds would not change the number of floppy modes compared to their number at $\langle r \rangle_0$, so it will remain greater than zero even for a fully coordinated network, which is not true, so we come to a contradiction. Thus the exact limit for stressless networks $\langle r \rangle_0 = \langle r \rangle_c^M$ (= 4 in 2D) is established. We would like to emphasize that the fact that each bond has just one associated constraint is essential for this proof. See the next subsection for comparison.

Secondly, it is possible to establish a relation between the self-organized networks and those obtained by usual completely random insertion (to which we for simplicity refer as "random" in contrast to "self-organized" in what follows). Indeed, assume we are using the same random list of M bonds to build a random network and a self-organized one, trying to insert bonds as they are listed. For the random network, all the M bonds will get in; for the self-organized network, some of them will be, generally speaking, rejected, so that $M_0 \leq M$ will be inserted. The bonds rejected in the self-organized network will be redundant in the random one;

they do not influence the number of floppy modes, the configuration of rigid clusters (and thus whether or not rigidity percolation occurs) and the redundancy or independence of all the subsequently inserted bonds. Thus all these characteristics will be identical for the two networks. The consequence is that there is a correspondence between self-organized and random networks having the same number of floppy modes; in particular, rigidity percolation occurs at the same number of floppy modes.

This analysis allows us to make a very important conclusion. Since in random networks rigidity percolates at a non-zero f and the same has to be true for self-organized networks (because of the just mentioned correspondence), yet stress appears exactly at f = 0, we conclude that there exists an *intermediate phase*, which is rigid (i.e., the infinite rigid cluster exists), but unstressed (so, evidently, there is no stress percolation). This is different from the situation with random insertion, where the *rigidity* and *stress* percolation thresholds always coincide (see Figures 5.1 and 5.2).

It could be possible that stress does not percolate immediately after it is introduced; we will see from simulation results that this is not the case, so the upper boundary of the intermediate phase (the stress transition) may be defined as either the point where stress first appears, or equivalently, the point where it percolates. As is seen from our consideration, it lies at $\langle r \rangle_0 = 4$.

As we have mentioned in Chapter 1, the fractions of bonds in the percolating rigid cluster P_{∞}^{r} and the percolating stressed region P_{∞}^{s} can serve as order parameters. Now, since there is an intermediate phase where rigidity percolates, while stress does not, these two parameters turn zero at different points, between which the intermediate phase lies. Besides, since the number of floppy modes is zero above the stress transition, the whole network is rigid, and thus P_{∞}^{r} is identically 1. These facts are illustrated in Figure 5.2.



Figure 5.1: The number of floppy modes per degree of freedom for random and selforganized diluted triangular networks in 2D. The thresholds are shown with various symbols, as specified in the legend. The intermediate phase in the self-organized case is shaded. Note that the rigidity transition occurs at the same f in the random and self-organized cases. The number of floppy modes in the self-organized case is strictly linear and coincides with the Maxwell counting result all the way up to the stress transition at $\langle r \rangle = 2.4$.

Given the discussion of the correspondence between random and self-organized networks, it is tempting to suggest that the same relation holds for the rigidity order parameter P_{∞}^{r} . The subtlety is that the relation is defined in terms of sites (i.e., same sites are in the percolating cluster in corresponding random and self-organized networks and same sites are pivot joints on its border), while in 2D we commonly define the rigidity order parameter in terms of bonds, since bonds always belong to just one cluster, whereas sites can be shared between the percolating cluster and other clusters. This is just a matter of definitions, though,



Figure 5.2: The fractions of bonds in the percolating rigid cluster and percolating stressed region for random and self-organized diluted triangular networks in 2D. Averages over two realizations on 400 \times 400 networks. The intermediate phase in the self-organized case is shaded.

and in any case, it might be safely assumed that the rigid cluster size critical exponents are the same for rigidity percolation in random and self-organized networks. Other critical exponents may be different, though.

It is interesting to note that since f given by Maxwell counting is exact in the whole unstressed region, then across both the floppy and the intermediate phases fis a perfect *straight line* and the rigidity transition *does not show up* in f.

Results of our simulations of this model are shown in Figures 5.1 and 5.2. The simulations were done for networks with periodic boundary conditions in both directions. There are several facts to be inferred (besides confirming all the results we have obtained so far). We see that stress percolates immediately after it appears at $\langle r \rangle = 4$ (this fact was mentioned above). Second, the critical exponent for the size of the stressed region is quite small (smaller than the one for the rigid cluster). In random networks, the stressed region exponent is larger than the rigid cluster exponent, which is obvious, as the stressed percolating region is smaller than the rigid cluster (the former being a subset of the latter) and the two thresholds coincide in the random case.

5.1.4 Intermediate phase in 3D bond-bending networks

In the case of 3D bond-bending glass networks there is a slight problem with implementing our general algorithm of self-organization. In the central-force case we were starting from an empty lattice to ensure that it had no stress initially. In the present case, as we avoid sites of coordination lower than 2, the initial dilution can only go as far as to the point where any further dilution would create a 1-coordinated site. At this limit there are no bonds with both ends being sites of coordination 3 and higher, so that further dilution is impossible. It is generally not true that this final network is unstressed. For smaller networks ($\sim 10^4$ sites and less), it is possible to pick those that are unstressed; for larger ones such cases are rare, and it is reasonable to assume that the fraction of constraints that are redundant is a constant in the thermodynamic limit. This constant seems to be very low, though (in our simulations, typically about 0.05% of constraints were redundant). Besides, the number of redundant constraints does not grow when new bonds are inserted according to our algorithm (up to the stress transition), so this problem is largely irrelevant.

Unlike the case of CF networks, BB networks have more than one constraint associated with each bond. When a new bond is added, not only the distance between the sites it connects is fixed, but the angles between the new bond and those stemming out of the two sites at either end of that bond are fixed as well.

Any bond that has at least one redundant constraint associated with it would cause stress. Some of the stress-causing bonds have only part of the associated constraints redundant and the rest independent, and such a bond will change the number of floppy modes. This makes some of our conclusions made for CF networks invalid in this case.

First, this invalidates the proof of the reachability of the Maxwell counting limit ($\langle r \rangle = 2.4$ in this case) without stress. This is because even when at the upper reachable limit all the as yet uninserted bonds would cause stress, some of these bonds may further decrease the number of floppy modes and thus this number is not necessarily zero at this point.

Second, the nice relation between random and self-organized networks no longer holds, because out of the redundant bonds by which the two differ, some (namely, the partially redundant ones) change f, rigidifying the network and changing the configuration of rigid clusters. We would assume that the equality of critical exponents β for rigid cluster sizes in random and self-organized cases still probably holds, but we have not checked this explicitly carefully enough yet.

At the same time, some facts are unchanged. In particular, f given by Maxwell counting is still exact in the unstressed region. Most importantly, the intermediate phase still exists.

The results of simulations done for the diluted diamond lattice are given in Figures 5.3 and 5.4. As in the previous subsection, we use periodic boundary conditions in all directions. We note in addition to the graphs that, as in the CF case, stress percolates immediately after it appears. The intermediate phase extends from $\langle r \rangle = 2.376$ to 2.392 (not reaching 2.4 indeed). Again, the stress transition is sharper than the rigidity transition. Our results are consistent with the second order transition with the very small critical exponent $\beta_{\rm str} \approx 0.1$, but a first order transition cannot be ruled out.



Figure 5.3: The number of floppy modes per degree of freedom for random and selforganized diluted diamond networks in 3D. The thresholds are shown with various symbols, as specified in the legend. The intermediate phase in the self-organized case is shaded. The rigidity thresholds are close in both cases, which is probably coincidental. The rigidity transitions definitely occur at different f, unlike the 2D CF case in Figure 5.1. The stress transition in the self-organized case is below the Maxwell counting threshold $\langle r \rangle = 2.4$.

Another feature of the plot in Figure 5.4 is that the rigidity order parameter is not exactly unity in the stressed phase (which is expected, as some floppy modes remain in the stressed phase) and the second transition shows up as a kink in the rigidity order parameter.

In conclusion to this section, we would like to mention that it is possible within our approach to establish a hierarchy of stress-causing bonds (by the number of associated redundant constraints) and when stress becomes inevitable, first put those having one redundant constraint, then those having two, and so on. Exactly



Figure 5.4: The fractions of sites in the percolating rigid cluster and in the percolating stressed region for self-organized diluted diamond networks in 3D. The intermediate phase is shaded. Circles are averages over 4 networks with 64000 sites, triangles are averages over 5 networks with 125000 sites. The dashed lines are the power law fit below the stress transition and for the guidance of the eye above. Note the break in the slope at the stress transition.

at $\langle r \rangle = 2.4$ only those bonds having no associated *independent* constraints will remain uninserted. It is unlikely, though, that there is good correlation between the number of redundant constraints and the actual increase in stress energy, as the distribution of stresses caused by different bonds is quite wide, so this complication seems unreasonable.

5.2 Self-organization in connectivity percolation

5.2.1 The model

It is interesting and useful to see if similar phenomena are possible in the more familiar case of connectivity percolation, especially as connectivity percolation is easier to study and understand.

The essence of our algorithm of building self-organized networks in the rigidity case is rejecting stress-causing bonds (or those having redundant constraints). As we have seen, in the CF case, when each bond has just one associated constraint, we may equivalently formulate this as rejecting redundant or irrelevant bonds. In bond connectivity percolation we also can build the networks by inserting bonds one by one; most importantly, there is a clear analog to redundant bonds. The relevant property now is connectivity, by which we mean the presence or absence of paths connecting any two sites of the network. Redundant bonds are those which connect sites already connected, that is would close a loop in the network. Thus the analog of self-organization is building *loopless networks*.

There are other equivalent ways to draw this parallel. The first is based on the fact that connectivity percolation can be considered as rigidity percolation with the sites having one degree of freedom regardless of the lattice dimensionality. Each site thus has one coordinate and each bond is a constraint (i.e., a relation between the coordinates of the sites it connects). Then the concepts of rigid clusters and clusters in the usual connectivity sense coincide. The number of floppy modes F is now the number of clusters. A redundant bond in the rigidity sense is the one that does not change F, it is also stress-causing, as it would introduce a relation between coordinates that cannot generally be satisfied. On the other hand, viewed from the connectivity perspective, such a bond connects the sites belonging to the same cluster and closing a loop, and our model is again recovered. Yet another way is to

recall that rigidity percolation with angular constraints in 2D (or with angular and dihedral constraints in 3D) is equivalent to connectivity percolation. Then stresslessness is equivalent to looplessness.

Connectivity percolation and related phenomena were studied so extensively in all imaginable flavors that it would be strange if this and similar models were not studied before. Indeed exactly this model was proposed as far back as 1979 [73] and rediscovered in 1996 [74]. Besides, there were extensive studies of loopless graphs (trees) in relation to various phenomena ranging from resistance of a network between two point contacts (considered by Kirchhoff in mid nineteenth century [75]) to river networks [76] to certain optimization problems [77, 78]. In many of these and other works the algorithm for building trees was equivalent to ours. Still, we consider some new aspects of this model and extend it somewhat, as we will see.

Given that connectivity percolation can be considered as rigidity percolation with one degree of freedom per site, we can apply the usual two-dimensional pebble game with the following modifications:

- 1) there is one pebble per site instead of two;
- for a trial bond to be independent, it must be possible to free two pebbles at its ends (one pebble can always be freed).

As before, the number of free pebbles equals the number of floppy modes (which for connectivity is the number of clusters). We emphasize that this algorithm is absolutely independent of the actual dimensionality of the network.

Of course, the essence of our self-organization algorithm is still the rejection of bonds that are not independent. Just as in the rigidity case, once avoidance of "stress" (loops) becomes impossible, we start inserting bonds at random. Thus we can go beyond the limit where the networks can no longer be loopless. This is where our model extends the previous considerations.

5.2.2 The intermediate phase

In this section we carry out the same kind of analysis as was done for rigidity percolation.

First of all we describe Maxwell counting, as this, although simple, is rarely discussed in relation to connectivity percolation. For a network with N sites the number of degrees of freedom is now simply N, the number of constraints is, as before, $N\langle r \rangle/2$, so the number of floppy modes per degree of freedom (or per site) is $f = 1 - \langle r \rangle/2$ and this becomes zero at $\langle r \rangle_c = 2$ — the same limit as given by the mean-field theory of percolation.

Since, as we have seen, connectivity percolation is nothing but a kind of rigidity percolation on a CF network with 1 degree of freedom per site, all of the general analysis for CF networks in the previous section is valid. Specifically, Maxwell counting is exact in the "unstressed" (this now means loopless) phases; the limit $\langle r \rangle_c = 2$ is reachable without creating loops; the relation between random and self-organized networks also holds. We note that the network in the upper limit reachable without stress is a *spanning tree*, i.e., it is a single tree that spans all sites in the network.

The order parameters are defined analogously to the rigidity case. The first parameter is (by analogy) the size of the percolating (connectivity) cluster. The second order parameter is, logically, the fraction of bonds in the percolating region consisting of loops (the analog of the stressed region in rigidity percolation), which essentially coincides with the percolation *backbone*.

The results of simulations for the square lattice are shown in Figure 5.5. Existence of the intermediate phase is confirmed in the range from $\langle r \rangle_c = 1.805$ to 2 for the square net. The lower transition coincides with the result for the percolation transition obtained in Ref. [74].



Figure 5.5: The fractions of bonds in the percolating rigid cluster (blue dots) and in the percolating "stressed" region (green dots) in the connectivity self-organization model for the diluted square lattice. Ten realizations on the 400×400 lattice without averaging. The intermediate phase is shaded.

5.3 Conductivity and elasticity

In this section we will look at physical manifestations of self-organization by studying the conducting properties of self-organized networks in the connectivity case and the elastic properties in the rigidity case.

5.3.1 Conductivity

A diluted network can be made into a random resistor network, if present bonds are replaced with resistors all having the same resistance. Then one can apply a potential difference across the network and find the current thus measuring the electrical conductance of the network. One can also introduce the effective conductivity as conductivity of a uniform material of the same size and shape as the network having the same conductance.

The conductivity properties of randomly diluted resistor networks have been studied extensively. Below the percolation threshold, the conductivity is zero, as the opposite sides are not connected; as the fraction of present bonds p grows and the threshold is crossed, the conductivity starts growing from zero. A mean-field (or effective medium) theory exists for conductivity [69], which predicts that this growth occurs linearly as a function of p. In reality, the dependence is nonlinear, with the critical exponent depending on the dimensionality (e.g., $\simeq 1.30$ in 2D [4]), but independent of the lattice type (e.g., square, triangular, etc. in 2D).

We now consider the conductivity in our model of self-organized networks.

First of all, we can implement different types of boundary conditions. One possibility is *periodic boundary conditions*, when we build a network on a torus according to our self-organization algorithm and then require that the potential changes by a constant when going a full circle around the circumference of the torus in one direction and does not change in the other direction(s). Since there are no loops in the network in the intermediate phase, the conductance of such a network (a finite or an infinite one) is *exactly zero*, despite the presence of an infinite cluster. That is, there is an infinite cluster, but no *conducting backbone*, which is a set of bonds that carry current. Other possible variants of boundary conditions and there are two busbars at either end with a voltage applied to them (Figure 5.6). Yet another possibility is the *source-sink geometry*, when we add two sites, a *source* next to one open boundary and a *sink* next to the other open boundary and connect these two sites to all sites at the respective boundary (Figure 5.7). In the latter case, the

conducting backbone in the (loopless) intermediate phase is a single filament; in the former case, it branches near the sides; in either case, the conductivity is not exactly zero for finite samples, but still goes to zero in the thermodynamic limit. We find the backbones using the pebble game analog of the Moukarzel's algorithm [79].



Figure 5.6: The conducting backbone (black) in the busbar geometry (open boundaries with busbars) in the intermediate phase. Bonds in the percolating cluster, but not in the backbone are red. Bonds not in the percolating cluster are green.

In the "stressed" phase (i.e., above the spanning tree limit), the conductivity is no more zero. Of course, it cannot be found with the pebble game and has to be studied numerically. We do this by solving the system of Kirchhoff equations directly using the conjugate gradient method. The surprising result (Figure 5.8) is that the conductivity dependence seems to be *exactly linear*, just as in the mean-field theory. This linearity also holds in 3D (Figure 5.9). Note that the spanning tree threshold corresponds to $\langle r \rangle = 2$, which is exactly the mean-field percolation threshold.



Figure 5.7: The conducting backbone in the source-sink geometry in the intermediate phase. Bonds between the source (sink) and the respective edge of the network are treated on equal footing with regular lattice bonds and can be present or absent. The backbone (black) is a single path between the source and the sink. Bonds in the percolating cluster, but not in the backbone, are red. Bonds not in the percolating cluster are green.



Figure 5.8: The conductivity for the random (green circles) and self-organized (red circles) diluted square networks. Averages over 25 realizations on the 200×200 lattice. The mean field result (the black straight line reaching 1 at $\langle r \rangle = 4$ is also shown. The threshold for the self-organized model is always at $\langle r \rangle = 2$. The threshold for the random model is also at $\langle r \rangle = 2$ in this case, which is a peculiarity of the square lattice.



Figure 5.9: The conductivity for the random (green circles) and self-organized (red circles) diluted cubic networks. Averages over 25 realizations on the $30 \times 30 \times 30$ lattice. The mean field result (the black straight line reaching 1 at $\langle r \rangle = 6$ is also shown. The thresholds now do not coincide (unlike in Figure 5.8 for the square lattice).

5.3.2 Mean-field behavior of conductivity

We can prove the above-mentioned linearity of conductivity, not for our self-organization model, but for a closely related one. In our self-organized model, we add bonds at random to a spanning tree. This spanning tree is constructed by using a random list of bonds and rejecting those that close loops. This is nothing but the *Kruskal algorithm* for building *minimal spanning trees* (MST) [80], if the list of bonds is ordered according to their cost. Thus our spanning tree belongs to the ensemble of MST on the network. This ensemble differs from the ensemble of uniform spanning trees (UST) [80], which are trees chosen at random among all possible spanning trees. In other words, our construction of trees is biased, and indeed, MST even have the fractal dimensionality of their branches different from that of UST.

Suppose now that we start from a UST instead of a MST and start adding bonds choosing their places at random among those where they were missing, just like in our self-organization model. We are now going to prove the linearity of conductivity for this model, which only differs from the original self-organization model in that the starting spanning tree is chosen without a bias.

We will use a result due to Kirchhoff [75, 81] that expresses the resistance between two points of a resistor network in terms of sums over trees built on the network. Namely, for the resistance between the source s and the sink s', we consider two sets of trees: T is the set of all possible spanning trees on the original lattice and T' is the set of all spanning trees on the same lattice, to which a bond between s and s' is added (we assume that there was no bond directly connecting sand s'), but only those of such trees that include the added bond. Then the sums Dand D' are introduced:

$$D(\{x_{ij}\}) = \sum_{t \in T} \prod_{E(t)} x_{ij}, \qquad (5.1)$$

$$D'(\{x_{ij}\}) = \sum_{t \in T'} \prod_{E(t)} x_{ij}, \qquad (5.2)$$

where x_{ij} is the conductance of the bond between *i* and *j*, the sums are over trees belonging to sets *T* and *T'*, respectively, and the products are over bonds belonging to each tree. Then the resistance is

$$R_{ss'} = \left(\frac{\partial}{\partial x_{ss'}} D'(\{x_{ij}\})\right) / D(\{x_{ij}\}).$$
(5.3)

In our particular case, when all resistances are equal to unity, this reduces to a tree-counting procedure. First, find the number of spanning trees in set T (we denote this number N_T). Second, find the number of spanning trees in set T' (denote $N_{T'}$). Note that finding $N_{T'}$ is equivalent to counting on the original network all graphs consisting of two trees, such that the source is in one tree and the sink in the other. Following Bollobás [81], we will refer to such graphs as *thickets*. Then the conductance S is the ratio of these two numbers:

$$S = \frac{N_T}{N_{T'}}.$$
(5.4)

For our proof that the conductivity dependence is linear, it is convenient to visualize the following diagram (Figure 5.10). First, imagine we have a set of all spanning trees that can be built on the full lattice. This set is denoted schematically as the left column of dots in Figure 5.10. Now consider a set of all networks that can be obtained from these trees by adding exactly B bonds (the middle column of dots in Figure 5.10). From each tree we can obtain

$$N_n = \begin{pmatrix} B_0 \\ B \end{pmatrix} \tag{5.5}$$

different networks, where B_0 is the total number of bonds missing in the tree compared to the full lattice (note that B_0 is the same for all trees, as the number of bonds in any spanning tree on a network of N sites is N - 1). This establishes connections between the set of trees and the set of networks, so that every tree is connected to N_n networks that can be obtained from it. These connections are shown schematically in Fig. 3 as the links between the dots in the left column and the dots in the middle column. The total number of connections is

$$C = N_n T_f, (5.6)$$

where T_f is the total number of possible spanning trees on the full lattice.

Conversely, these same connections specify which trees can be built on each network (i.e., are subgraphs of the network).



Figure 5.10: A schematic diagram showing relations between trees and networks and between thickets and networks. The left column of dots denotes the set of all possible trees on the full lattice; the right column is the set of all thickets; the middle column is the set of all spanning networks with a certain number of bonds. The connections in the left part (between trees and networks) show what networks can be built by adding bonds to a tree, or, conversely, what trees are subgraphs of a given network. The connections in the right part show similar relations between thickets and networks. The ratio of the numbers of connections in the left part and in the right part is proportional to the conductivity of a network with a certain number of bonds in the thermodynamic limit, as discussed in the text.

Similarly, we can consider the set of all possible thickets on the full lattice (denoted by the right column of dots). Every thicket has N - 2 bonds, so there are $B_0 + 1$ empty bonds. If we add a set of B + 1 bonds to a thicket, we will almost certainly obtain one of the networks under consideration (there is a possibility that none of these bonds connect the two trees of the thicket together, but this is negligibly rare in the thermodynamic limit). Then the number of networks that can be obtained from each thicket by bond insertion is

$$N_n' = \binom{B_0 + 1}{B + 1}.\tag{5.7}$$

and the total number of connections is

$$C' = N'_n T'_f, (5.8)$$

where T'_f is the total number of possible thickets on the full lattice. These connections are also shown in the diagram. Again, they also specify which thickets can be built on each network.

A plausible assumption that we make is that conductance is self-averaging, i.e. in the thermodynamic limit the conductance is the same for all but a negligible fraction of realizations. Then for almost every network the ratio of the number of trees connected to it to the number of thickets connected to it is the same (as this ratio equals the conductivity, according to Eq. (5.4)). The anomalous networks, for which this is not the case, have a negligible probability of occuring. The probability to obtain a particular network is proportional to the number of connections between this network and various trees; thus anomalous networks have a negligible amount of connections with the trees; this is also true for their connections with the thickets, as the ratio of the number of connections with the the thickets for every network is its conductance, and the conductivity is expected to be O(1) for all networks, including anomalous ones. Then all connections and the ratio of the total number of connections between the left and the middle columns to that between the right and the middle columns is again the conductance. On the other hand, this ratio is

$$S = C/C' = \frac{N_n T_f}{N'_n T'_f} = \frac{T_f(B+1)}{T'_f(B_0+1)} \propto B \text{ for } B \gg 1,$$
(5.9)

as B_0 , T_f and T'_f do not depend on B, and the proof is complete.

Some comments are in order. First, we have not made any assumptions about the underlying lattice, so that the result is independent of the lattice type and dimensionality, although there may be problems with the assumptions that we made in pathological cases, such as RBNs (Chapter 2), when connections between sites infinitely far apart are possible. Second, we had in mind a situation with the source and the sink at the opposite sides of the lattice, but have not used this fact anywhere. We did make an assumption that conductance is realization-independent and this is only true when the source and the sink are infinitely far apart.

While the above proof is interesting, it gives no significant insight as to why the conductivity dependence is still linear for our self-organization model, even though MST are different from UST. On the other hand, the analogous model for shortest path trees [80] seems to show tiny deviations from linearity that probably are not entirely due to finite-size effects. Clearly, deeper insights into the problem are needed.

We should mention that it is easy to produce spanning networks, for which their conductivity dependence is definitely nonlinear. Trivial examples can be obtained by adding bonds to anisotropic trees, in which most branches are directed, say, perpendicular to the applied potential difference. More interesting is the following *isotropic* case. Start from the full lattice and start picking bonds at random, but removing them only if this removal does not separate a piece of the network from the rest. In this way, we obtain spanning networks, with no finite clusters, just as in our original model. Obviously, this dilution procedure can be

continued until a spanning tree is obtained. Despite certain similarity with our previous models, the conductivity dependence now is not mean-field-like. Indeed, it can be shown that the conductivity becomes zero *before* reaching the spanning tree limit, i.e., at a higher bond concentration (Figure 5.11). Thus, another interesting example of an *intermediate phase* is formed: in a certain range of bond concentrations, the infinite cluster exists, but its structure is such that the conductance is very low and vanishes in the thermodynamic limit.



Figure 5.11: Conductivity for the restricted dilution model described in the text. Average over 25 realizations on the 100×100 square lattice. There is an "intermediate phase" of a different type, with the spanning cluster, but zero conductivity. The straight line is the mean-field result, from which there is obvious deviation.

To see this, suppose we create a random list of bonds intended for removal. In the case of random bond dilution, all of these bonds are removed in the order given by the list. In the restricted dilution case, some of these bonds will be rejected and not removed. Note that rejected bonds do not belong to loops and so their removal would not change the configuration of loops and also would not change conductivity (at least not in the case of PBC). Then, there is a correspondence between sequences of networks obtained by random dilution and those obtained by restricted dilution using the same list of bonds whose removal is attempted. If random dilution procedure proceeds below the percolation threshold, the conductivity of obtained networks becomes zero, so the conductivity of the corresponding networks obtained by restricted dilution is also zero, even though by construction these networks still span all sites.

An analogous model can be considered for rigidity: bonds are removed so that the percolating cluster does not break up. Similarly, there is an intermediate phase, which, unlike the intermediate phase in our self-organization model, has stress, but stress still does not percolate while rigidity does.

5.3.3 Superconducting networks

We have seen that in the thermodynamic limit the conductivity σ is zero in both the disconnected and intermediate phases. These results make us wonder if the lower transition shows up in any physical quantities for infinite networks. One possibility is to consider *superconductor networks* instead of resistor networks. In this model all the existing bonds are replaced with conductors of zero resistance ("superconductors"), while all the absent bonds are resistors with equal finite resistance.

It turns out that the same kind of correspondence between random and self-organized networks with the same f we had for clusters is valid for the conductance in this case. Indeed, these networks differ by redundant bonds that connect sites already connected. All the connected sites have zero potential difference (as they are connected with superconductors), so putting redundant bonds does not change the distribution of the potential and thus does not influence
the conductance.

It is known [4] that in the random case the resistivity is zero above the threshold and non-zero below it, with the critical exponent the same as for the conductivity of resistor networks (≈ 1.30). Thus in the self-organized case the resistivity will turn zero in the point related to the percolation threshold of random networks by the above relation, i.e., at the lower transition (Figure 5.12). The critical exponent will be the same as in the random case (≈ 1.30), but this is now different from the value for σ of resistor networks (≈ 1 or perhaps exactly 1).



Figure 5.12: Conductivity for the superconductor (blue dots) and resistor (green dots) networks in the self-organized model on the square lattice. The conductivity of resistor networks is not exactly zero in the intermediate phase, since open boundaries with the busbars were used, unlike in Figure 5.8, where periodic boundary conditions were used and the conductivity was exactly zero in the intermediate phase. This non-zero conductivity in the intermediate phase is a finite-size effect. All results are averages over ten square 100×100 networks.

5.3.4 Elasticity

In rigidity, the quantities analogous to conductivity are the elastic moduli.

Again, just as in the connectivity case, different boundary conditions are possible. In the case of periodic boundary conditions, again the elastic moduli are exactly zero in the intermediate phase, regardless of the size of the supercell and despite the existence of the percolating rigid cluster. Indeed, periodic boundary conditions mean that positions of images of same site in different supercells are fixed with respect to each other. The network is built stressless with these additional constraints taken into account. The exact specification of these constraints beyond stating what sites are involved is determined by the particular size and shape of the supercell, but is never taken into account (just as particular constraint lengths never matter in determination of stressed regions). So straining the network by changing this size and shape leaves it stressless. The important thing here is that straining does not add any new constraints. We confirmed this result numerically. Thus it can be said that in the intermediate phase the network is just *marginally rigid*.

For other boundary conditions, similarly to the connectivity case, the elastic moduli are non-zero for finite samples, but become zero in the thermodynamic limit. We will have non-zero stress in the intermediate phase when an external strain is applied, and some of the bonds will be stressed. These bonds are said to belong to the *applied stress backbone* [82] (which we refer to as simply backbone in what follows). It can be found easily by the pebble game using a method essentially equivalent to that proposed by Moukarzel and Duxbury [82], which consists in putting an additional bond across the network emulating the external strain, and finding those bonds in which stress is induced. A typical result is shown in Figure 5.13, in which it is seen that the backbone has filamentary structure. We note that stress in this backbone was created by putting just one extra bond and thus it is enough to take any one bond out of the backbone for it to be destroyed, so it is

155

extremely fragile.

In the stressed phase, the elastic moduli are non-zero and have to be found numerically. Figure 5.14 shows the dependence of the uniaxial strain modulus c_{11} on the mean coordination. The triangular lattice was first distorted by random displacement of atoms. For displacements along each axis uniform distribution on an interval (-0.1;0.1) in units of the lattice constant was chosen, but the results are only slightly sensitive to the width of the distribution. Equilibrium lengths of springs were chosen equal to the distance between the atoms they connect, so the initial network is unstressed. Thus subtraction of two large energies when finding elastic constants is avoided. The results may differ, of course, compared to the case when the constraint equilibrium lengths are just assigned at random, but both situations can be regarded as legitimate model cases. Pre-determining the applied stress backbone speeds up the relaxation greatly, as was first pointed out in Ref. [82]. We note that the elastic constant dependence is definitely *not* linear in our model, unlike the dependence of the conductivity in the analogous connectivity model.

5.4 Experimental evidence

It is possible that the intermediate phase described in this chapter has been observed experimentally. Boolchand et al. have carefully studied a number of systems by Raman scattering and modulated differential scanning calorimetry (MDSC), monitoring various quantities as the chemical composition was varied. Their experiments are reviewed in detail in [83]. In particular, in Si_xSe_{1-x} glasses, they observed two kinks in the composition dependence of the Raman mode corresponding to the symmetric stretch of Si(Se_{1/2})₄ tetrahedra, at $\langle r \rangle = 2.40$ and $\langle r \rangle = 2.54$ [84] [Figure 5.15,(a)]. Also, in their temperature-modulated differential



Figure 5.13: An elastic self-organized network in the intermediate phase with the applied stress backbone (black) shown. Bonds in the percolating cluster, but not in the backbone, are red. Bonds not in the percolating cluster are green.



Figure 5.14: The elastic modulus C_{11} for the random and self-organized diluted square networks. Averages over 10 realizations on the 100×100 lattice. The intermediate phase is shaded. The inset shows the blowup of the critical region. Deviations from the mean-field line (black) are clearly seen in both the random and the self-organized cases.

scanning calorimetry measurements in the same system, they observed a broad minimum of the non-reversing heat flow measured across the glass transition, and the minimum is bounded by the same values of $\langle r \rangle$ [84] [Figure 5.15,(b), filled circles]. Boolchand et al. interpret this as an indication of the intermediate phase, so that the rigidity transition occurs at 2.40 and the stress transition at 2.54. Note that this range is wider than the theoretical intermediate phase discussed here, but it is probable that the theoretical model is just too simple to get all the details correct. Wide minima in the non-reversing heat flow were also observed in other systems (Ge-As-Se [85], Ge-Se [26, 86], As-Se [87]), but apparently no sharp thresholds. On the other hand, in the Ge_{0.25}S_{0.75-y}I_y system [88], the minimum is very narrow [Figure 5.15,(b), open circles] and it is concluded that the intermediate phase is absent in this system; the threshold is shifted down from 2.4 in exact accordance to Eq. (1.9). Returning to Raman scattering, in some cases the results seem to depend qualitatively on the input laser power: in the Ge-Se system, two transitions are seen at low power [86] and one at high power [26].

We should mention that there is an alternative model for the intermediate phase seen in Boolchand's experiments, due to Micoulaut and Phillips [89]. In their model, in the intermediate phase the network has stress, but all stressed regions are very small and do not percolate. This is not unlike the intermediate phase in our restricted dilution model that we briefly described in section 5.3.2.

Finally, J.C. Phillips [90] advocates the point of view that high-temperature superconductors represent an intermediate phase between two transitions (as a function of dopant concentration) to non-superconducting phases, with filamentary conducting backbones, just like in our model.



Figure 5.15: (a) Frequencies of two Raman modes corresponding to the motion of corner-sharing tetrahedra in $Si_z Se_{1-x}$ shown as a function of mean coordination; (b) Non-reversing heat variation in $Si_z Se_{1-x}$ (red filled circles) and $Ge_{0.25}S_{0.75-x}I_x$ (green open circles) glasses. Both panels are adapted from Ref. [83].

5.5 Conclusion

In this chapter, we have considered a model of self-organization of networks that exhibits two separate phase transitions, a rigidity and a stress transition, and a marginally rigid and stressless intermediate phase between them. We have also introduced an analogous model for connectivity percolation. The behavior of conductivity in the latter is, surprisingly, exactly or nearly exactly mean-field, and even though this behavior was proved for a closely related model, the deep underlying reasons for it still remain a mystery. Certainly, some new insights are desirable.

Other than that, main efforts should be directed at trying to get better understanding of how the theoretical results obtained here are related to experiment.

Chapter 6: Conclusion and outlook

In this dissertation we have studied rigidity of various systems using a variety of methods. The networks ranged from totally random in Chapter 4 to chemically ordered in a simple way in Chapters 2 and 3 to self-organized in a non-trivial fashion in Chapter 5. We have done analytical calculations for the random bond model in Chapter 2 and numerical simulations in Chapters 4 and 5; we have proved some non-trivial results: the first-order transition for chemically ordered networks in Chapter 3 and mean-field behavior of conductivity in Chapter 5. We have seen a variety of different behaviors: first- and second-order transitions, as well as two separate transitions and the intermediate phase between them in self-organized systems.

Usual connectivity percolation was widely studied by many groups. A simple problem has become a paradigm of phase transitions and critical phenomena. Rigidity percolation is in a sense even more interesting and certainly more diverse. As an example, Table 6.1 shows the order of the rigidity transition for different randomly diluted (i.e., uncorrelated) networks. Note that the connectivity percolation transition is second order in all these cases.

d.o.f.	2		3		
topology	2D	RBN	2D	3D	RBN
CF	2nd	1st	2nd	1st	1st
BB	2nd	2nd	2nd	2nd	1st

Table 6.1: The summary of rigidity transitions in networks of various topologies in both two and three dimensions. The top line shows the dimensionality of the space in which the network is embedded. The second line specifies the topology of the network. The table shows the order of each transition to our best knowledge. The entries discussed in this dissertation are in red.

Throughout this work we mentioned several outstanding issues of rigidity theory. Perhaps the most important of those already mentioned is constructing a fast exact algorithm for general 3D networks. Also, we have emphasized qualitative features, rather than specific numbers; as a result, many of the relevant critical exponents are not determined yet, in particular, critical behavior of elastic constants is still not well-understood.

Most importantly, perhaps, rigidity theory in its current form, being a purely mechanical theory from the physical point of view, is only applicable to systems at zero temperature. At $T \neq 0$, entropy also plays a role, not just energy, and as a result, networks that are identified as floppy have non-zero elastic constants (vanishing in the limit $T \rightarrow 0$) due to the effects of entropic elasticity, as shown by Plischke *et al.* [91]. This effect smears the rigidity transition, which, together with the similar smearing effect due to presence of neglected weaker interactions, makes the rigidity transition hard to observe in practice. Certainly, any progress in studying systems at non-zero temperature, as well as taking the weaker forces into account within the framework of rigidity theory, would be very welcome. Work in this direction is currently carried out by Huerta and Naumis [92] and by Jacobs [93].

In short, rigidity is still an exciting field. Much has been done, but much still has to be done.

Bibliography

- [1] P.A.M. Dirac, Proc. Roy. Soc. A **123**, 714 (1929).
- [2] R.B. Laughlin, Talk at the APS Centennial Meeting, audio and transparencies available at http://www.apscenttalks.org/ (1999).
- [3] See, e.g., K. Ohno, K. Esfarjani, and Y. Kawazoe, Computational Materials Science, from ab initio to Monte Carlo methods (Springer-Verlag, Berlin, 1998).
- [4] D. Stauffer and A. Aharony, Introduction to Percolation Theory, 2nd ed. (Taylor & Francis, London, 1992).
- [5] D.J. Jacobs and M.F. Thorpe, Phys. Rev. Lett. 75, 4051 (1995).
- [6] M.F. Thorpe, D.J. Jacobs, N.V. Chubynsky, and A.J. Rader, in *Rigidity Theory and Applications*, edited by M.F. Thorpe and P.M. Duxbury (Kluwer Academic/Plenum Publishers, New York, 1999), p. 239.
- [7] J.C. Maxwell, Philos. Mag. 27, 294 (1864).
- [8] D.J. Jacobs and B. Hendrickson, J. Comp. Phys. 137, 346 (1997).
- [9] G. Laman, J. Engrg. Math. 4, 331 (1970).
- [10] D.J. Jacobs and M.F. Thorpe, Phys. Rev. E 53, 3682 (1996).
- [11] C.M. Fortuin and P.W. Kasteleyn, Physica 57, 536 (1972).
- [12] P.M. Duxbury, D.J. Jacobs, M.F. Thorpe, and C. Moukarzel, Phys. Rev. E 59, 2084 (1999).
- [13] T.-S. Tay and W. Whiteley, Structural Topology 9, 31 (1984); W. Whiteley, in *Rigidity Theory and Applications*, edited by M.F. Thorpe and P.M. Duxbury (Kluwer Academic/Plenum Publishers, New York, 1999), p. 21.
- [14] D.J. Jacobs, J. Phys. A: Math. Gen. **31**, 6653 (1998).
- [15] D.J. Jacobs, L.A. Kuhn, and M.F. Thorpe, in *Rigidity Theory and Applications*, edited by M.F. Thorpe and P.M. Duxbury (Kluwer Academic/Plenum Publishers, New York, 1999), p. 357.
- [16] D.J. Jacobs, private communication.
- [17] W.H. Zachariasen, J. Am. Chem. Soc. 54, 3841 (1932).
- [18] A.C. Wright, in Amorphous Insulators and Semiconductors, edited by M.F. Thorpe and M.I. Mitkova (Kluwer Academic, Dordrecht, 1997), p.83.
- [19] M.F. Thorpe, J. Non-Cryst. Solids 57, 355 (1983).
- [20] P. Boolchand and M.F. Thorpe, Phys. Rev. B 50, 10366 (1994).

- [21] F. Wooten, K. Winer, and D. Weaire, Phys. Rev. Lett. 54, 1392 (1985).
- [22] B.R. Djordjević, M.F. Thorpe, and F. Wooten, Phys. Rev. B 52, 5685 (1995).
- [23] W. Whiteley, Structural Topology 1, 46 (1979).
- [24] C. Moukarzel, J. Phys. A: Math. Gen. 29, 8079 (1996).
- [25] J.C. Phillips, J. Non-Cryst. Solids **34**, 153 (1979).
- [26] X. Feng, W.J. Bresser, and P. Boolchand, Phys. Rev. Lett. 78, 4422 (1997).
- [27] R. Böhmer and C.A. Angell, Phys. Rev. B 45, 10091 (1992).
- [28] U. Senapati and A.K. Varshneya, J. Non-Cryst. Solids 185, 289 (1995).
- [29] P. Boolchand, X. Feng, D. Selvanathan, and W.J. Bresser, in *Rigidity Theory and Applications*, edited by M.F. Thorpe and P.M. Duxbury (Kluwer Academic/Plenum Publishers, New York, 1999), p. 279.
- [30] M. Tatsumisago, B.L. Halfpap, J.L. Green, S.M. Lindsay, and C.A. Angell, Phys. Rev. Lett. 64, 1549 (1990).
- [31] D.J. Jacobs, A.J. Rader, L.A. Kuhn, and M.F. Thorpe, Proteins 44, 150 (2001).
- [32] C.M. Dobson, A. Sali, and M. Karplus, Angew. Chem. Int. Edit. 37, 868 (1998).
- [33] A.J. Rader, B.M. Hespenheide, L.A. Kuhn, and M.F. Thorpe, PNAS 99, 3540 (2002).
- [34] G. Biroli and M. Mézard, Phys. Rev. Lett. 88, 025501 (2002).
- [35] H.A. Bethe, Proc. Roy. Soc. A **150**, 552 (1935).
- [36] R.E. Peierls, Proc. Cambridge Philos. Soc. **32**, 471 (1936).
- [37] M. Kurata, R. Kikuchi, and T. Watari, J. Chem. Phys. 21, 434 (1953).
- [38] M.E. Fisher and J.W. Essam, J. Math. Phys. 2, 609 (1961).
- [39] F. Peruggi, F. Diliberto, G. Monroy, J. Phys. A: Math. Gen. 16, 811 (1983).
- [40] F. Wagner, D. Grensing, and J. Heide, J. Phys. A: Math. Gen. 33, 929 (2000).
- [41] M.F. Thorpe, D. Weaire, and R. Alben, Phys. Rev. B 7, 3777 (1973).
- [42] M. Mezard and G. Parisi, J. Stat. Phys. 111, 1 (2003).
- [43] R. Bruinsma, Phys. Rev. B **30**, 289 (1984).
- [44] D. Dhar, P. Shukla, and J.P. Sethna, J. Phys. A: Math. Gen. **30**, 5259 (1997).

- [45] R. Dobrin, J.H. Meinke, and P.M. Duxbury, J. Phys. A: Math. Gen. 35, L247 (2002).
- [46] M.F. Thorpe, in *Excitations in Disordered Systems*, edited by M.F. Thorpe (Plenum Publishers, New York, 1982), p. 85.
- [47] C. Moukarzel, P.M. Duxbury, and P.L. Leath, Phys. Rev. Lett. 78, 1480 (1997).
- [48] C. Moukarzel, P.M. Duxbury, and P.L. Leath, Phys. Rev. E 55, 5800 (1997).
- [49] M. Chubynsky and M.F. Thorpe, in *Physics and Applications of Disordered Materials*, edited by M. Popescu (INOE, Bucharest, Romania, 2002), p. 229.
- [50] H. He, Ph.D. dissertation, Michigan State University (1985).
- [51] W. Whiteley, private communication.
- [52] W.H. Press, S.A. Teukolsky, V.T. Vetterling, and B.P. Flannery, Numerical Recipes in FORTRAN, 2nd ed. (Cambridge University Press, New York, 1992).
- [53] G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, MD, 1983).
- [54] A.D. Dinsmore and D.A. Weitz, J. Phys.: Cond. Mat. 14, 7581 (2002).
- [55] E.R. Weeks and D.A. Weitz, Chem. Phys. 284, 361 (2002).
- [56] V. Prasad, V. Trappe, A.D. Dinsmore, P.N. Segre, L. Cipelletti, and D.A. Weitz, Faraday Discuss. 123, 1 (2003).
- [57] A.D. Dinsmore, E.R. Weeks, V. Prasad, A.C. Levitt, and D.A. Weitz, Applied Optics 40, 4152 (2001).
- [58] P.N. Pusey, A.D. Pirie, and W.C.K. Poon, Physica A **201**, 322 (1993).
- [59] J.C. Conrad, private communication.
- [60] http://www.deas.harvard.edu/projects/weitzlab/
- [61] S. Feng, M.F. Thorpe, and E. Garboczi, Phys. Rev. B 31, 276 (1985).
- [62] N. Mousseau, G.T. Barkema, and S.M. Nakhmanson, Phil Mag. B 82, 171 (2002); R.L.C. Vink, G.T. Barkema, M.A. Stijnman, and R.H. Bisseling, Phys. Rev. B 64, 245214 (2001).
- [63] R. Car and M. Parrinello, Phys. Rev. Lett. 55, 2471 (1985); L.J. Lewis, A. De Vita, and R. Car, Phys. Rev. B 57, 1594 (1998).
- [64] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).

- [65] P. Vashishta, R.K. Kalia, A. Nakano, and I. Ebbsjö, in *Amorphous Insulators and Semiconductors*, edited by M.F. Thorpe and M.I. Mitkova (Kluwer Academic, Dordrecht, 1997), p. 151.
- [66] N. Mousseau and G.T. Barkema, Phys. Rev. E 57, 2419 (1998).
- [67] A.F. Voter, Phys. Rev. Lett. 78, 3908 (1997); M.R. Sorensen and A.F. Voter, J. Chem. Phys. 112, 9599 (2000).
- [68] G. Henkelman and H. Jonsson, J. Chem. Phys. 115, 9657 (2001).
- [69] S. Kirkpatrick, Rev. Mod. Phys. 45, 574 (1973).
- [70] M.F. Thorpe, D.J. Jacobs, M.V. Chubynsky, and J.C. Phillips, J. Non-Cryst. Solids 266-269, 859 (2000).
- [71] M.F. Thorpe and M.V. Chubynsky, in Properties and Applications of Amorphous Materials, edited by M.F. Thorpe and L. Tichy (Kluwer Academic, Dordrecht, Netherlands, 2001), p. 61.
- [72] M.F. Thorpe and M.V. Chubynsky, in *Phase Transitions and Self-Organization in Electronic and Molecular Networks*, edited by J.C. Phillips and M.F. Thorpe (Kluwer Academic/Plenum Publishers, New York, 2001), p. 43.
- [73] J.P. Straley, Phys. Rev. B **19**, 4845 (1979).
- [74] S.S. Manna and B. Subramanian, Phys. Rev. Lett. 76, 3460 (1996).
- [75] G. Kirchhoff, Ann. Phys. Chem. 72, 497 (1847).
- [76] M. Cieplak, A. Giacometti, A. Maritan, A. Rinaldo, I. Rodriguez-Iturbe, and J.R. Banavar, J. Stat. Phys. 91, 1 (1998).
- [77] R.C. Prim, Bell Syst. Tech. J. **36**, 1389 (1957).
- [78] M. Cieplak, A. Maritan, and J.R. Banavar, Phys. Rev. Lett. 72, 2320 (1994);
 M. Cieplak, A. Maritan, and J.R. Banavar, Phys. Rev. Lett. 76, 3754 (1996).
- [79] C. Moukarzel, Int. J. Mod. Phys. C 9, 887 (1998).
- [80] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, Introduction to Algorithms (MIT Press, Cambridge, MA, 1990).
- [81] B. Bollobás, Modern Graph Theory (Springer Verlag, New York, 1998), Section II.1.
- [82] C. Moukarzel and P.M. Duxbury, Phys. Rev. Lett. 75, 4055 (1995).
- [83] P. Boolchand, D.G. Georgiev, and B. Goodman, J. Optoelectron. Adv. Mater. 3, 703 (2001).

- [84] D. Selvanathan, W.J. Bresser, P. Boolchand, and B. Goodman, Solid State Commun. 111, 619 (1999).
- [85] Y. Wang, P. Boolchand, and M. Micoulaut, Europhys. Lett. 52, 633 (2000).
- [86] P. Boolchand, X. Feng, and W.J. Bresser, J. Non-Cryst. Solids 293, 348 (2001).
- [87] D.J. Georgiev, P. Boolchand, and M. Micoulaut, Phys. Rev. B 62, 9228 (2000).
- [88] Y. Wang, J. Wells, D.J. Georgiev, P. Boolchand, and K. Jackson, Phys. Rev. Lett. 87, 185503 (2001).
- [89] M. Micoulaut and J.C. Phillips, Phys. Rev. B 67, 104204 (2003).
- [90] J.C. Phillips, Phys. Rev. Lett. 88, 216401 (2002).
- [91] M. Plischke, D.C. Vernon, B. Joos, and Z. Zhou, Phys. Rev. E 60, 3129 (1999).
- [92] A. Huerta and G.G. Naumis, Phys. Rev. Lett. 90, 145701 (2003).
- [93] D.J. Jacobs, unpublished.

