T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Mass Spring Models for the Simulation of Deformable Objects
著者(和文)	MACIEJ KOT
Author(English)	Maciej Kot
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9660号, 授与年月日:2014年9月25日, 学位の種別:課程博士, 審査員:長橋　宏,熊澤　逸夫,小池　康晴,中村　健太郎,山口　雅浩
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第9660号, Conferred date:2014/9/25, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Powered by T2R2 (Science Tokyo Research Repository)

Mass Spring Models for the Simulation of Deformable Objects

東京工業大学
Tokyo Institute of Technology

Maciej Kot

Information processing department. Nagahashi lab.

This dissertation is submitted for the degree of

Doctor of Engineering

August 2014

We hereby recommended that the dissertation by Mr. Maciej Kot entitled "*Mass spring models for the simulation of deformable objects*" be accepted in the fulfillment of the requirement for the degree of Doctor of Engineering.

Commitee fo Final Examination

1. Professor Nagahashi Hiroshi

2. Professor Koike Yasuharu

3. Professor Kumazawa Itsuo

4. Professor Nakamura Kentaro

5. Professor Yamaguchi Masahiro

Date of the defense: 20^{th} August 2014

Abstract

Mass Spring Models (MSM) are a popular choice for representation of soft bodies in computer graphics and virtual reality applications. Here we investigate their physical properties aiming to provide a solid theoretical basis for the realistic simulation with MSMs. We take under consideration both regular networks as well as disordered ones and provide models with which any homogeneous isotropic material can be successfully described.

MSMs are believed to be fast, but not as accurate as the alternatives. The main problem is to establish the relationship between spring properties and real physical properties of modeled materials. Obtaining a realistic and fast MSM behavior based on physical laws is especially useful for VR applications, where interaction with the environment is a common problem. Most of computer graphics applications ignore physical properties of materials, and try to "guess" appropriate MSM parameters to achieve realistic behavior of their models. Many applications are however more demanding, especially if they often involve both interaction and feedback problems. In such cases "fake realism" is not sufficient.

To address this problem, we formulate a methodology for constructing the mass spring models with well-defined physical properties. In particular, we emphasize the importance of isotropy of the models and the limitations it places on the geometry of the network. First, we discuss the classical result of the theory of elasticity that while any value of the Young's modulus can be obtained by tuning the spring coefficients in the network, the value of the Poisson's ratio is constrained to

be 1/4 (in 3D). Next, we derive a simple formula, linking the Young's modulus with the mean value of kL^2 throughout the network, where k and L are the spring stiffness coefficient and its length, respectively.

Next, we extend our models to represent materials with other values of Poisson's ratio as well. By introducing an extended force interaction mechanism we can achieve a plausible behavior for a wide range of materials. We measure the elastic properties of both cubic and random lattices in a series of test simulations and show that they agree satisfactory with theoretical predictions and thus are well suited for the modeling of elastic materials.

Acknowledgements

I would like to extend my sincere and profound gratitude to Prof. Nagahashi Hiroshi, my supervisor for his kindness and understanding, his enormous moral and academic support, constructive comments and guidance throughout the entire period of my doctoral study.

I wish to express my deepest gratitude to dr. Piotr Szymczak for his valuable comments, help with improving my academic writing and effort of proof reading my publications.

I also wish to extend my great appreciation and sincere thanks to the doctoral Advisory Committee, Prof. Koike Yasuharu, Prof. Kumazawa Itsuo, Prof. Nakamura Kentaro and Prof. Yamaguchi Masahiro for their valuable contribution toward the completion of my PhD study program.

My PhD study would not have been possible without the scholarship offered by Japanese Ministry of Education and support of JSPS KAKENHI (Grant Number 24300035).

Contents

1 Introduction **1**

2 Mass spring models **5**

2.1 Linear elasticity theory 5

2.2 Introduction to mass spring models 9

2.3 Simple MSMs 10

2.3.1 Triangular lattice MSMs (2D) 10

2.3.2 Cubic lattice MSMs 12

2.3.3 Disordered MSMs 14

2.3.4 Elastic constants of simple mass spring models 17

2.4 Conclusions 22

3 Model verification **23**

3.1 Strain and stress measurement 23

3.1.1 Strain 23

3.1.2 Stress 24

3.2 Tests 25

3.2.1 Uniaxial compression 25

3.2.2 Cubic MSM – Resolution scaling 26

3.2.3 Cubic MSM – Isotropy 27

3.2.4 Random MSM 27

3.2.5 Point force in an infinite solid 29

3.2.6 Deflection test 34

3.2.7 Complex shapes 38

3.3 Conclusions . . . 38

4 Extended mass spring models **41**
4.1 Other works . . . 42
4.2 Particle model of matter . . . 42
4.2.1 Overview . . . 43
4.2.2 Particle interaction model . . . 45
4.3 Non-linear springs . . . 45
4.4 The second elastic parameter . . . 47
4.5 Cubic lattice eMSM . . . 51
4.6 Disordered eMSM . . . 53
4.7 Conclusions . . . 54

5 Anti-aliasing **57**
5.1 Anti-aliasing . . . 57
5.2 Tests . . . 59
5.2.1 Elastic parameters . . . 59
5.2.2 Internal force distribution . . . 61
5.2.3 Complex shapes . . . 63
5.3 Conclusions . . . 65

6 Aspects of dynamics of mass spring models **67**
6.1 Introduction . . . 67
6.2 Collisions . . . 68
6.2.1 Verlet integration scheme . . . 68
6.2.2 Elastic collisions . . . 69
6.2.3 Elastic collisions with overlap prevention . . . 72
6.3 Contact friction . . . 73
6.4 Tests . . . 76
6.4.1 Scenario 1 – energy dissipation . . . 78
6.4.2 Scenario 2 – complex movement . . . 80
6.5 Conclusions . . . 80

7 Conclusions and final remarks **83**

Bibliography **87**

Chapter 1

Introduction

Mass spring models (MSM, or lattice spring models, LSM) provide a simple and easy to implement method for the simulation of the behavior of deformable objects. Using MSM is a popular approach for modeling soft bodies in virtual reality (VR) applications as well as computer graphics (CG) [11, 31] and it is often the method of choice where a fast response of a non-rigid body is required [29].

MSMs are believed to be fast, but not as accurate as the alternatives such as FEM (*finite element method*). Physical correctness greatly helps to improve plausibility of any CG application, and for some of them is absolutely essential (i.e. VR environment involving both interaction and feedback). For the MSM to be physically correct, a relationship needs to be established between spring constants and the real physical properties of modeled materials.

As in every expanding field, the literature on the topic covers by now a large number of publications. It would be impossible to list all the relevant articles, we therefore direct the reader to the review article by Ostoja-Starzewski [32], which presents physically-oriented approach to the different types of MSMs as well as approaches of estimating their physical properties. Another publication discussing MSM approaches, particularly in the context of CG, is the paper by San-Vicente et al. [37]. An extensive survey of deformable modeling in computer graphics can be found e.g. in Nealen and Müller et al. [31], whereas Meier et

al. [29] summarize approaches focused on surgery simulation applications, which offer a more detailed analysis of MSMs.

Although MSMs have been carefully investigated by many CG researchers, some of the results and ideas from the classical theory of elasticity did not seem to percolate into CG community. A good example is provided by the problem of calculating the Poisson's ratio of the mass spring model. A well-established result of the continuum mechanics is that the Poisson's ratio of a homogeneous and isotropic material is identically 1/4 (or 1/3 for a 2D system), if the constituents of this material interact with the central forces dependent upon distance alone [28, 22]. Unfortunately, many of the studies of mass spring models do not seem to recognize this fact: either trying to estimate the Poisson's ratio of their models (and finding invariably that it is close to 1/4 or 1/3) [7, 26, 1] or even trying to impose a different value of the Poisson's ratio, a task doomed to failure [42, 37]. This has lead to the conviction in CG community that MSMs can not represent elastic materials accurately and that they do not converge to the exact solution of elasticity equations when their resolution is increased [31].

In chapter 2 of this work we show what can and what cannot be achieved with simple mass spring models, when trying to represent a homogeneous isotropic elastic material characterized by specific elastic moduli using models of arbitrary resolutions. We will analyze two approaches of creating such MSM. First one assumes mass point distribution on a cubic lattice – such models will be called *cubic lattice MSM*. Second approach is based on random node placement (*random MSM*; described in detail in the next section).

We present both theoretical predictions of elastic parameters as well as their verifications in several test cases.

There were several attempts to identify MSM parameters of particular materials ([43, 20, 1, 26]). Models presented by San Vicente et al. in [43] are not isotropic, and Baudet et al. [1] introduce additional modifications for MSM model that can not be represented by linear springs, therefore they describe more specific models. Lloyd et al [26] present the expression for spring parameters in

simple MSM. They approximate the equations for the material behavior using FEM, and then try to find an equivalent mass-spring model for each element. Their solution is based on tetrahedron meshes, and the explicit equation is given for regular tetrahedra. Ladd and Kinney [20] predict the MSM parameters from linear elasticity equations for *cubic lattice*. We will verify those predictions and use *cubic MSM* as a reference model to compare it to *random MSM* and propose a method of estimating parameters of *random MSM*.

In chapter 4 we will show how to modify *simple MSM*, to overcome some of its limitations, such as a fixed value of Poisson's ratio. The chapter contains very interesting extension to standard MSMs, which is attractive both computationally and as a concept which allows to unify wide range of the behaviors of continuous matter into a set of few simple rules.

In chapter 5, we analyze an approach of constructing *cubic lattice MSMs* that takes into account "aliasing" effect. Similarly as it is done in classical antialiasing techniques, we weaken cubical elements used for representation of the surface regions, adjusting them accordingly to represent the properties of the portion of the border they overlap with.

Chapter 6 contains an analysis of some problems related to dynamics of elastic models. Defining a deformable object is only half of the solution needed for performing realistic animation. Once we have such representation of an object, we have to simulate its evolution in time, which includes both evolution of its shape (internal relaxation) as well as interaction with the environment, that is – collisions with other objects. In particular to achieve a consistent behavior of different resolutions of MSM a valid model of friction has to be implemented, that would allow for similar energy dissipation rates between different resolutions of MSMs. We analyze dynamic properties of our models keeping special attention to the problem of how resolution changes affect them.

I

Chapter 2

Mass spring models

In this chapter we discuss the static properties of mass spring models. Network parameters which influence these properties are spring coefficients k, and connections rules (network topology). Mass and damping coefficients do not. This means, that models described in this chapter are intended to give a physically plausible behavior in static equilibrium setups (i.e. no net motion present). An example of such setup is a bent beam, in which for a certain deformation, a certain tension in the material is expected (and vice versa). In experiments such as those presented in this chapter, the displacement of each point of a body can be found in a number of ways, including explicit simulation of the movement, or simply by energy minimization [34].

2.1 Linear elasticity theory

In this section we introduce some basic linear elasticity results that can be found e.g. in [28] and [23].

The position $\bar{r}$ of each point in a deformed body can be characterized by its displacement $\bar{u}$ from the original position $\bar{r}_0$ (Fig. 2.1). Relative displacements

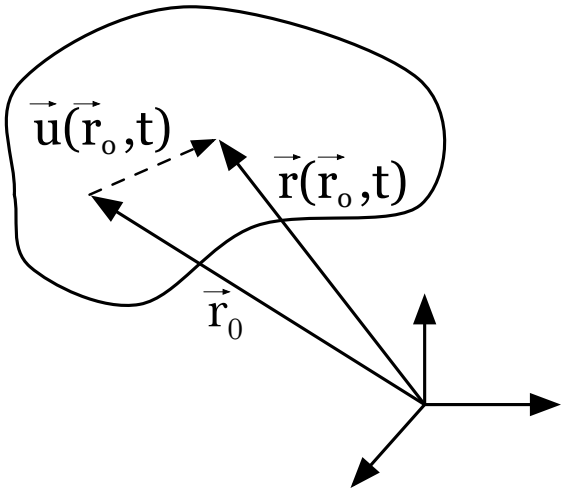

Figure 2.1: Displacement of a material point in the continuous body.

can be locally expressed by a 2-rank tensor

$$\tau_{ik} = \frac{\partial u_i}{\partial x_i}, \tag{2.1}$$

so that $\bar{u}(\bar{r} + d\bar{r}) = \bar{u}(\bar{r}) + \hat{\tau}(\bar{r}) \cdot d\bar{r}$.

It is convenient to decompose it into symmetric and anti-symetric parts:

$$\hat{\tau} = \hat{\alpha} + \hat{\epsilon}, \tag{2.2}$$

$$\alpha_{ik} = \frac{1}{2}(\frac{\partial u_i}{\partial x_k} - \frac{\partial u_k}{\partial x_i}), \tag{2.3}$$

$$\epsilon_{ik} = \frac{1}{2}(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}). \tag{2.4}$$

The anti-symmetric part holds the information about local rotation of a body. The actual deformation is contained in the symmetric part $\hat{\epsilon}$, known as the *strain* tensor.

The momentum flow inside a deformed body is expressed by the *stress* tensor $\hat{\sigma}$, and a basic assumption of linear elasticity theory is that stress in an elastic

body is a linear function of the strain:

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl}, \tag{2.5}$$

where $\hat{\sigma}$ denotes stress tensor, $\hat{\epsilon}$ - deformation (strain) tensor and $\hat{C}$ the elastic moduli tensor. Both $\hat{\sigma}$ and $\hat{\epsilon}$ are symmetrical from which it follows that in three dimensions $\hat{C}$ has 21 independent coefficients. This number is reduced to 2 if we assume that the material is isotropic. In such case $\hat{C}$ can be expressed as a linear combination of 4-rank numerical tensors:

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \tag{2.6}$$

where δ_{ij} is the Kronecker delta. The relation (2.5) now becomes

$$\sigma_{ij} = \lambda\delta_{ij}\delta_{kl}\epsilon_{kl} + 2\mu\epsilon_{ij}, \tag{2.7}$$

or equivalently:

$$\sigma_{ij} = K\delta_{ij}\epsilon_{kk} + 2\mu(\epsilon_{ij} - Tr(I)^{-1}\delta_{ij}\epsilon_{kk}), \tag{2.8}$$

where λ and μ are Lamé parameters, I is an identity matrix and K is a bulk modulus obeying $K = \lambda + \frac{2}{3}\mu$ in 3D or $K_{2D} = \lambda + \mu$ in 2D. Einstein summation notation is used here (i.e., the appearance of a repeated index in a term implies summation over it, e.g. $\epsilon_{kk} = \sum_{k=1}^{d} \epsilon_{kk}$, where d is dimensionality of the space).

Elastic energy F of a deformed body is given by

$$F = \frac{1}{2}\sigma_{ij}\epsilon_{ij} = \frac{1}{2}\lambda(\epsilon_{ii})^2 + \mu\epsilon_{jk}^2, \tag{2.9}$$

whereas the equation of motion reads

$$\rho\frac{\partial^2\bar{u}}{\partial t^2} = \lambda\nabla(\nabla\cdot\bar{u}) + \mu(\nabla^2\bar{u} + \nabla(\nabla\cdot\bar{u})), \tag{2.10}$$

where $\bar{u}$ is a displacement and ρ mass density.

The relative change of the volume of a body can be expressed by strain as:

$$\frac{V+\Delta V}{V} = (1+\epsilon_{11})(1+\epsilon_{22})(1+\epsilon_{33}) \approx 1+\epsilon_{11}+\epsilon_{22}+\epsilon_{33}, \tag{2.11}$$

therefore with accordance to Eq. (2.8), K gives the resistance to compression. It can also be defined as:

$$K = -V\frac{dp}{dV}, \tag{2.12}$$

where V is the volume of a solid body, and p pressure acting on its surface.

Beside λ, μ and K, two other elastic parameters are often defined – the Young's modulus E and the Poisson's ratio ν. The Young's modulus is the ratio of stress to strain measured along the same axis under an uniaxial stress condition, whereas the Poisson's ratio is the ratio of transverse to axial strain i.e. $E = \sigma_{xx}/\epsilon_{xx}$ and $\nu = -\epsilon_{yy}/\epsilon_{xx}$, in an experiment where all components of the stress tensor are zero except σ_{xx}. The above parameters can be expressed in terms of λ and μ:

$$E = \frac{\mu(3\lambda+2\mu)}{\lambda+\mu} \qquad \nu = \frac{\lambda}{2(\lambda+\mu)}, \tag{2.13}$$

or

$$E_{2D} = \frac{4\mu(\lambda+\mu)}{\lambda+2\mu} \qquad \nu_{2D} = \frac{\lambda}{\lambda+2\mu}. \tag{2.14}$$

Any two of the above discussed parameters are sufficient to describe elastic properties of an isotropic material, thus given any two, any other of the elastic moduli can be calculated.

The assumption that all the internal forces in the material are the central forces (i.e. that they can be represented by the potential $V(r)$, with r – the

distance between the particles) leads to 6 additional relations between the components of C_{ijkl} (note B in [28]). Thus in a non-isotropic case $\hat{C}$ will have 15 independent coefficients instead of 21, and for an isotropic material 1 instead of 2. In the latter case $\lambda = \mu$, from which it follows, that

$$\nu = 1/4 \qquad \nu_{2D} = 1/3. \tag{2.15}$$

2.2 Introduction to mass spring models

Mass spring models represent an elastic material by means of discretization. It is important to note that in such representations there are two things that are discretized. The first is a mass distribution which in real materials is usually treated as continuous, the second, represented by springs, characterizes the interactions between the material points and thus defines elastic properties of a physical body. Both of these aspects are important and will affect the realistic appearance of simulated objects. If we increase the number of mass points representing an object, the resolution of the model becomes higher resulting in more detailed deformations. On the other hand, increasing the number of springs attached to each mass point allows one to represent the local elastic properties of the material in a more accurate way.

We emphasize the distinction between these two aspects because in most popular variants of the mass spring models, the springs connect the neighbouring nodes only. In principle, however, one can introduce also the connections between the further nodes. This means that such networks cannot be decomposed into a sum of non overlapping volumetric primitives, as it is the case in other discretization approaches (e.g. FEM). Most of the systems analyzed in the present study are of this kind.

As already mentioned, independently of the discretization technique, if only radial springs are used, then, in accordance with (2.15), any isotropic homogeneous structure will have a fixed value of Poisson's ratio. Other values of the Pois-

son's ratio can be obtained by incorporating non-central forces into the model, e.g. the angular terms, or forces that do not depend on distance alone or by introducing anisotropy [22]. Such extensions are often present in more advanced models [1, 26].

2.3 Simple MSMs

First we will analyze the basic variation of MSM, that is, networks composed of mass points and linear springs. We will refer to them as *simple MSMs*, to make the distinction between such models and more advanced MSMs.

2.3.1 Triangular lattice MSMs (2D)

While this work is focused on volumetric objects and their MSM representations, for completeness we include a brief discussion on two-dimensional networks. Much attention in computer graphics has been given to two dimensional MSMs (mostly for the purpose of the cloth representation) and many advanced models have been created. The body of work on the subject is quite extensive, and it is not our intention to review this area. Instead, we focus on the simplest triangular meshes capable of representing isotropic materials.

In one of the pioneering CG works on this topic Van Gelder [42] analyzes 2D spring meshes and tries to find the spring stiffness coefficients that would allow mass spring model to behave like a reference FEM model.

The results seem to be discouraging, with the author stating that *"assigning the same stiffness to all springs badly fails to simulate a uniform elastic membrane"*. It seems, however, that part of the problem here is the attempt to represent materials with arbitrary values of ν, whereas as it was already noted only $\nu = 1/3$ materials can be modeled with radial 2D spring networks. For such materials the 2D MSM models based on hexagonal lattice and uniform stiffness coefficients were shown to perform reliably [32, 20, 26]. The use of hexagonal lattice (composed of equilateral triangles) is necessary to retain the isotropy of the

system [32]. Another potential problem in [42] is that the calculation of elastic properties is performed for a single triangular element of the network, whereas the elastic response of the system will also depend on the relative placement of such triangles in the periodic cell.

The relation between the spring coefficient k and the Lamé constants for a hexagonal network are given by

$$\lambda = \mu = \frac{3}{4\sqrt{3}}k, \tag{2.16}$$

from which all the other elastic moduli can be derived (note that there is a typo in this particular expression in [32]).

This model can be easily extended to achieve other values of ν by incorporating angular springs between triangle edges which is known as a Kirkwood model of an isotropic material [32]. Elastic moduli are then given by

$$K_{2D} = \frac{3}{2\sqrt{3}}k \qquad \nu_{2D} = \left(1 - \frac{3\beta}{2ka^2}\right) \Big/ \left(3 + \frac{3\beta}{2ka^2}\right), \tag{2.17}$$

where a is a length of an edge of a triangle, k is a spring stiffness coefficient of regular springs and β of angular ones. As observed, $\nu_{2D} \to 1/3$ when $\beta \to 0$.

As mentioned above, the fact that $\nu_{2D} = 1/3$, is a general property of central force models and it usually resurfaces in one form or another in any work that investigates such models, even if its analytical justification is unknown to the authors. For example Delingette [7], while analyzing similar triangular meshes notices that *"without the addition of angular springs, MSMs with the right stiffness parameters can at best approximate the behavior of a membrane with $\nu_{2D} = 1/3$."*. Lloyd et al. [26] arrive at equation (2.16) and also notice that membranes can be better approximated when $\nu = 1/3$, than for other values of ν, not recognizing, however, that it is not an approximation but an exact result for the model chosen.

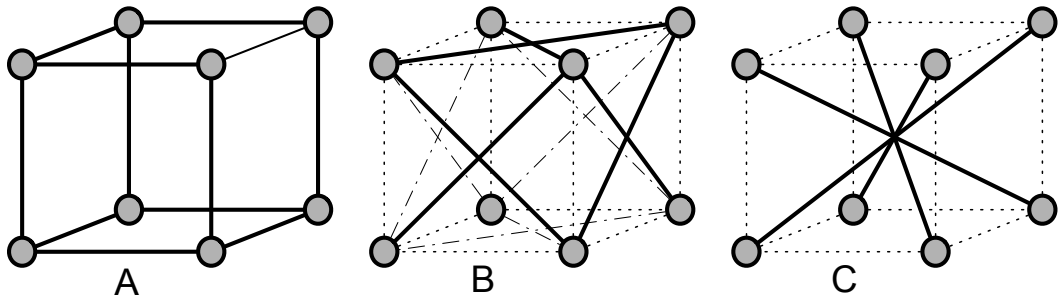

Figure 2.2: Elementary cell of a cubic lattice; Three types of spring connections. A – springs between nearest neighbors, B – springs along face diagonals and C – along cube diagonals

2.3.2 Cubic lattice MSMs

Just as hexagonal lattice MSM in two dimensions, a cubic lattice MSM can be used to represent homogeneous isotropic materials in 3D, if the connections between the nodes and their stifnesses are chosen in an appriopriate way [20].

Unfortunately, there is no work in CG that satisfactory explores the properties of such networks. The most recent publication on this topic is by San Vicente et al. [37]. It runs, however, into similar problems to van Gelder's work discussed above – the authors attempt to impose the values of the Poisson's ratio different from 1/4 (which is an exact result for any 3D radial spring network). The performance of this model is rather questionable, as even in the test of simple bending, the deformation (and thus also the elastic moduli) seem to depend strongly on the resolution of the MSM, in contrast to the results shown in the present work (*cf.* Sec. 3.2). Comparison of Fig 10. in [37] with Fig. 3.14 of the present work shows that the cubic MSMs considered here show almost no dependence on the resolution and are thus better suited for the practical applications.

Contrary to what is suggested in [37] an isotropic system can be reliably modeled with 3d cubic lattices, provided the connectivities and spring stiffnesses are chosen in an appropriate way. The models of that kind were constructed in [20]. The two simplest variants have 18 and 14 connections per node, respectively.

In a cubic lattice, neighbours of each node can be classified in three groups forming connections of different lengths as shown in Fig. 2.2:

- Group A: Springs between the nearest neighbors. There are 6 of them, two along each axis (along the cube edges).
- Group B: Springs between nodes lying on the other side of face diagonal (12 such springs).
- Group C: Springs between nodes lying on the other side of internal cube diagonal (8 such springs).

The first model allows nearest-neighbor and next-nearest-neighbor connections (group A and B in Fig. 2.2), all with the same spring coefficient k. The second model uses A-springs with $k = \kappa$, and C-springs with $k = \frac{3}{8}\kappa$, however it is reported to be less stable.

In both cases the macroscopic displacement field of the spring network is shown to obey:

$$\rho\partial_t^2\bar{u} = \frac{k}{a}[2\nabla\nabla\cdot\bar{u} + \nabla^2\bar{u}] + O(\nabla^4 u), \tag{2.18}$$

where a is the length of an edge of an elemental cube, and $\bar{u}$ the displacement vector.

From eq. (2.10), we can see that both Lamé constants have the same value, equal to k/a. Using equation (2.13) we can write E and v as a function of k and a:

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} = 2.5\frac{k}{a} \qquad \nu = \frac{\lambda}{2(\lambda + \mu)} = 1/4 \tag{2.19}$$

Note that this MSM is constructed as a *superposition* of cubical elements. For each "element", the corresponding mass is distributed equally into its corners. When two elements are placed beside each other, their corners (masses) coincide,

and the effective mass at these points increases. Analogously spring coefficients increase as well. Therefore in order to get a correct representation of boundaries, masses placed on a border of an object have to be smaller and springs weaker with comparison to those inside the object. A single cubical element with mass m has $\frac{1}{8}m$ at each corner, its A-springs have $k = 0.25\kappa$ and B-springs $k = 0.5\kappa$. Such construction allows to represent boundaries in a convenient way, however it has some drawbacks, which we will discuss in detail in Sec. 3.2.6.

Another work that tackled the problem of cubic MSMs for CG applications is Baudet et al. [1]. They present an extended MSM capable of representing materials with ν other than 1/4. However, in their numerical analysis they treat Young's modulus, Poisson's ratio, shear modulus and bulk modulus as four independent quantities, not recognizing that only two of them are independent, and in case of radial springs – only one. Hence, by imposing four arbitrary values for these quantities, they run into a risk of constructing a self-contradictory problem, which cannot be resolved, no matter what values of spring coefficients are used. Despite that, they arrive at $\nu = 1/4$ solution, but discard it as not versatile and unsatisfying. Instead they introduce additional "orthogonal correction force" to the model, which effectively allows to achieve other values of ν. Such *extended MSM* performs entirely satisfactory in compression and shearing tests for $0.1 < \nu < 0.5$, with errors being minimal for $\nu = 0.25$, however the results of the bending test are again strongly dependent on the resolution, similarly as in [37].

2.3.3 Disordered MSMs

Cubic MSMs are simple and their properties are well defined as elucidated in the previous section, however in some situations it is beneficial to relax the constraint of regular placement of the nodes. In order to give a general description of MSMs we consider representations with nodes placed randomly in a given volume.

Models closest to this has been analyzed by Lloyd et al. [26] as they too consider disordered node distribution. They construct their MSMs as tetrahedral

meshes and by means of minimization try to match the behavior of a FEM model. We will compare our results with theirs after presenting our method first.

The core of the discretization procedure used among others by Lloyd (et al.) [26], is to decompose the object into a set of primitives (i.e. tetrahedra) which do not overlap except on the faces, and each of them represents a portion of the volume of the whole object. Such representations are convenient, because for each primitive the elastic properties of its interior can be defined completely and expressed in a relatively simple manner (cubic lattice MSM belongs to this group as well). The important difference of our models is, that this time we not only consider a network with disordered node placement, but additionally abandon the FEM-like approach of constructing objects out of volumetric primitives. We propose a node-centered approach in which we treat a mass point as a basic building block, to which any number of springs can be attached representing channels for the flow of momentum throughout the system. Elastic properties are now expressed statistically by spring density and the description is approximate, however we gain the flexibility of adjusting per node connections, which is a powerful tool for improving local behavior of the network.

In order to stay as general as possible, we will perform numerical experiments on MSMs constructed with a very few assumptions about the node placement as well as simple connection rules. First, we generate the positions of the nodes using *random sequential addition algorithm* [40], which simply means, that we iteratively pick random positions within the sample and check whether a newly created point is sufficiently separated from the neighboring nodes. If not, the point is discarded and a new position is sampled. The minimal allowed distance between the nodes is given by $minL$ parameter (Fig. 2.3). The procedure runs until it fills the volume with a desired number N of mass points (which may not be possible, if N is too large for a given volume). Next, all nodes which are less than $maxL$ apart get connected by springs.

In practice it is more efficient to generate the random nodes in several stages, in smaller parts of the system. In this work, we have used a spherical 'sampling

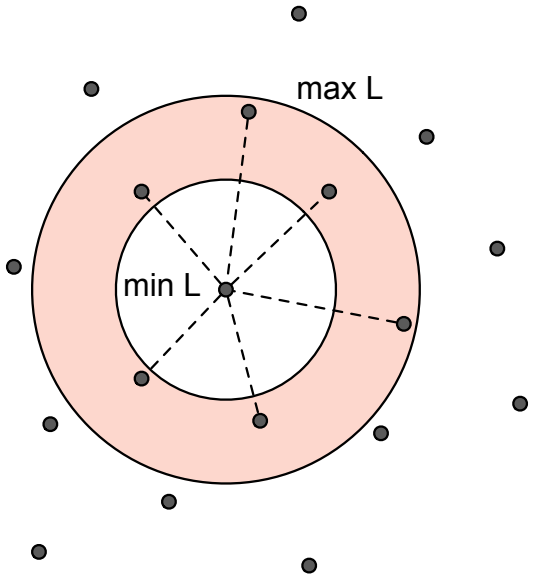

Figure 2.3: Random point distribution. Example of spring connections for a chosen node (in 2D for better readability; real models are three dimensional). Minimal allowed distance between two points equals *minL*. Maximal range of spring connection – *maxL*.

window' with radius $R_s = 3minL$, which advanced through the volume with $\sqrt{2}R_s$ steps. A sphere was chosen to ensure that the shape of boundaries of the system would not influence the isotropy of the node distribution. The initial sampling volume was slightly larger than the size of the system and the nodes positioned outside of the modeled block were discarded in the final step of the MSM generation process.

Random MSMs tested in this work differ by *minL*, *maxL* and the total number of nodes, N (which sets the resolution for a given volume). However it will prove more convenient to express our results in terms of another set of parameters: mean spring length $< L >$, standard deviation of the spring lengths d_L and average number of springs attached to a node $< S >= 2S_{tot}/N$, where S_{tot} is the total number of springs in the system. We have analyzed models with $< L >$ varying from $0.97l$ to $1.52l$, $< S >$ from 8.2 to 29 and standard deviation between $0.1l - 0.33l$, where l is a characteristic distance between the mass points, $l = (V/N)^{1/3}$. These values correspond to distributions generated with *minL* between $0.81l$ and $0.94l$ and *maxL* between $1.17l$ and $1.99l$.

Treating cubic lattice model as a "random" model with $l = a$ (where a is a lattice constant), $minL = a$ and $maxL$ in the range between $\sqrt{2}a$ and $\sqrt{3}a$ we can introduce a loose correspondence between cubic lattice MSM and random MSM; we achieve the same elastic properties with roughly the same number of nodes and springs for both models.

2.3.4 Elastic constants of simple mass spring models

We have already established that the Poisson's ratio of all the models based on central forces only, is equal to 1/4, which applies to disordered networks as well. Since for a homogeneous and isotropic medium, there are two independent elastic constants, we need one more to close the description. The simplest quantity to calculate is the bulk modulus K (see eq. (2.8) and (2.12)).

To estimate it, let us imagine uniform compression of our system as a result of which all the lengths of the springs are compressed by the same degree

$$L_i' = L_i(1 + \epsilon), \tag{2.20}$$

where L_i' is the new, compressed length of the i-th spring and $\epsilon < 0$ in case of compression. The change in the energy density associated with this deformation is then

$$\Delta e = \frac{\sum_i \frac{1}{2} k_i \epsilon^2 L_i^2}{V}, \tag{2.21}$$

where V denotes the volume of the object. However, from the basic elasticity theory [23] Δe in the uniform compression is given by

$$\Delta e = \frac{9}{2} K \epsilon^2. \tag{2.22}$$

Thus

$$K = \frac{1}{9V}\sum_i k_i L_i^2 = \frac{1}{9}\frac{n < S >}{2} < kL^2 >, \tag{2.23}$$

where k_i and L_i denote spring coefficient and natural length of i-th spring, $n = N/V$ is the concentration of the nodes, the average $< kL^2 >$ is taken over all the springs in the system and $< S >$ over all the nodes and denotes the average number of springs connected to a node (one spring connects two nodes, hence division by 2). Once the bulk modulus (and Poisson ratio) is known, all the other elastic constants can be estimated, e.g. Young's modulus E:

$$E = 3K(1 - 2\nu) = \frac{1}{6V}\sum_i k_i L_i^2, \tag{2.24}$$

where $\nu = 1/4$ was assumed in the last equality.

As an example, let us calculate the Young's modulus of the first model of the cubic lattice from Section 2.3.2. The spring constant is then uniform and the mean square length of the spring reads

$$< L^2 >= \frac{6 \times 1 + 12 \times 2}{18} a^2 = \frac{5}{3} a^2 \tag{2.25}$$

whereas $< S >= 18$. This leads to the bulk modulus:

$$K_c = \frac{5}{3}\frac{k}{a}, \tag{2.26}$$

which yields $E = 2.5\frac{k}{a}$ if we take $\nu = 0.25$, giving the same result as the one obtained previously. It is easy to show that the same value is obtained in the case of the second model of Sec. 2.3.2 as well.

We can also try to compare the estimations of our formula with the ones given by Lloyd et al. [26] for tetrahedral meshes. In order to get a Young's modulus of a regular tetrahedron we take its volume $V_r = \frac{\sqrt{2}}{12} l^3$, where l is the length of its

edge, and after assuming the same k for all 6 springs, and $\nu = 0.25$, the equation (2.24) yields:

$$\begin{aligned} E &= 6\sqrt{2}\frac{k}{l}, \\ k &= \frac{\sqrt{2}}{12}lE. \end{aligned} \tag{2.27}$$

The analogous formula computed by means of minimization by Lloyd et al. with use of the Matlab Symbolic Math Toolbox, as the one giving best approximate to the reference FEM model reads:

$$k_{ij} = \sum_e \frac{2\sqrt{2}}{25}lE, \tag{2.28}$$

where we have a summation over all tetrahedral elements e adjacent to the spring between nodes i and j, and the expression within the sum gives the influence of each of those elements to the spring's stiffness coefficient. It is very similar to our result, but they do not match perfectly. It is mentioned however that several similarity measures have been tried (for the comparison with FEM), each giving similar, but slightly different result. It is possible that the choice of another similarity measure would lead to the equation identical to ours.

The similarity of the expression for regular tetrahedron obtained by us and Lloyd et al. is encouraging, however there is a number of problems when comparing such estimations that should be mentioned here. Firstly, regular tetrahedra do not fill the 3D space and in order to make these equations applicable to a real tetrahedral mesh, in general case, k has to be scaled by V_i/V_r for a tetrahedron with volume V_i, as suggested by Lloyd (et al.). This leads to the question of how exactly should the sum in Lloyd's formula (2.28) be interpreted, as there are about $5\frac{1}{10}$ regular tetrahedra needed to wrap the whole 360° around a chosen edge. Secondly our equation (2.23) was derived with the assumption of the isotropy of the medium – in particular (2.22) holds for an isotropic material – and a single tetrahedron can hardly be considered an isotropic structure. Therefore

it is illegal for us to say, that an isolated tetrahedron would have such elastic properties. Nevertheless if we imagine for example a 3d space filled with a soup of regular tetrahedra (overlapping and rotated in all directions and somehow connected with each other), then eq. (2.27) would give the influence of a single one.

The equations (2.23) and (2.24) allow one to calculate the elastic constants of a material modeled with any MSM, not limited to a particular node distribution or spring connection rules. However, the more homogeneous and isotropic the model is, the more faithfully it can be described by the two elastic moduli K (or E) and ν. As we recall, such a two-parameter description is possible for isotropic solids only. Moreover simple MSMs are based on linear elasticity theory, and are expected to give a plausible elastic response only within its limits, that is for small relative displacements.

Before we proceed to the section with tests, let as have a closer look at the local properties of our random MSM. The formula (2.23) as presented in this section, requires counting all the springs and nodes in the system, thus giving the "average" K for the whole network. By applying it to a smaller region of the material, as we did above for tetrahedral mesh, it is possible to obtain an estimation of the elastic properties of an object locally (with discretization errors increased greatly if not done carefully). The procedure in which we derived equation (2.23) assumes that the volume of an object is filled with a soup of springs, however the spring distribution in the random MSM as we create it, is not uniform on a local scale. Springs are not placed arbitrarily and everywhere, but are radially coming out of nodes. This allows us to treat the nodes as building blocks of the network, and ask a question, what is the contribution of a single mass point (with its springs) to the bulk modulus, and in turn what is the contribution of a single spring into that. We may treat a mass point with its springs as a sphere with radius $R = 0.5 < L^* >$, where $< L^* >$ is the average length of the springs connected to this mass point (fig. 2.4a). Moreover we can treat each spring, as giving an influence to some spherical region of the material (fig. 2.4b).

The bulk modulus of a spherical object can be calculated with the eq. (2.12):

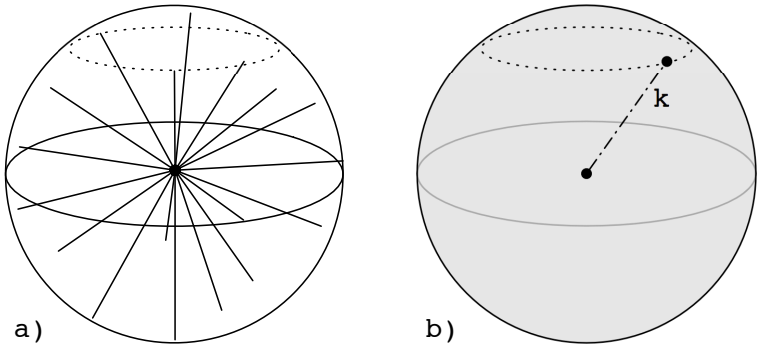

Figure 2.4: a) A mass point with its springs as a building block of the MSM. b) A spring giving an influence to the spherical region of the material.

$$K = -V\frac{dp}{dV}.$$

Decreasing the radius of the sphere by ΔR we induce the tension in the springs equal to $2k\Delta R$, hence the total pressure acting on the surface of the sphere becomes $p = S^* < k > \Delta R/2\pi R^2$ whereas the relative change in the volume is $\Delta V/V = -3\Delta R/R$. This gives

$$K = \frac{S^* < k >}{6\pi R} = \frac{2}{9}s^* < k > R^2 \tag{2.29}$$

where S^* is the number of half-springs in the sphere, and $s^* = S^*/V$ their concentration; $V = \frac{4}{3}\pi R^3$. If we assume the whole network to be a superposition of such spheres, we can rewrite the spring concentration as $s^* = n < S >$ (note that we don't need to divide by 2 here, because we were considering half-springs), and arrive at:

$$K = \frac{1}{9}n < S > \frac{< k >< L >^2}{2},$$

which essentially is the eq. (2.23), with averages taken in a different way. This shows, that formula (2.29), can be used to estimate the contribution to the bulk modulus of a single node. Moreover, if we consider that each half-spring in the system belongs to some sphere, we can write the contribution of a single (full)

spring as:

$$\chi = 2\frac{2}{9}kR^2 = \frac{1}{9}kL^2, \tag{2.30}$$

which brings us to the building block of eq. (2.23) independently of its original derivation procedure.

If we take $k' = k/L^2$ for each spring, the contribution to the bulk modulus K becomes the same for each spring and it is independent on the spring's length. This allows to freely scale lengths of the springs and create springs between arbitrary pairs of nodes without affecting their influence on the bulk modulus (we do however have to keep the spring distribution isotropic on per node basis and spring density approximately constant).

2.4 Conclusions

In this chapter we have formulated a methodology for constructing mass spring models with well-defined physical properties for the use in Computer Graphics. In particular, we emphasized the importance of isotropy of the models and the limitations it places on the geometry of the network. Next, we have discussed the classical result of the theory of elasticity that while any value of the Young's modulus can be obtained by tuning the spring coefficients in the network, the value of the Poisson's ratio is constrained to be 1/4 (in 3D). Finally, we have derived a simple formula, Eq. (2.24), linking the Young's modulus with the mean value of kL^2 throughout the network, where k and L are the spring stiffness coefficient and its length, respectively. This is a relatively simple and yet general formula which can be applied to virtually any MSM and does not require a complicated analysis of the model.

Chapter 3

Model verification

3.1 Strain and stress measurement

In this section we will briefly discuss the numerical techniques allowing to estimate strain and stress in MSMs. They will be used to verify if the elastic response of our models agrees with theoretical predictions [18].

3.1.1 Strain

Displacement of each node can be simply computed by subtracting its actual and starting positions (Fig. 2.1). Resulting vector will contain information about overall movement of the point, including global translation and rotation of the object. In linear elasticity theory, the stress tensor is given by:

$$\epsilon_{ki} = \frac{1}{2}(\frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_k}) \tag{3.1}$$

Unfortunately in order to estimate $\frac{\partial \bar{u}}{\partial \bar{r}}$, the knowledge about the displacement in the neighborhood of our point is needed. The only knowledge we have about our discretized system is contained in the nodes, and displacement at an arbitrary point of the volume has to be computed by means of interpolation. In order to do that in a way that does not depend on node distribution, we tesselate the

undeformed version of the object, identify tetrahedron containing the point in question, and do the linear interpolation using barycentric coordinates.

This way, displacement at any point inside the volume of an object can be calculated based on the displacement of nodes. This allows to compute strain tensor in a straightforward way using the equation (3.1). Value of partial derivatives is estimated as:

$$\frac{\partial \bar{u}}{\partial \bar{n}}(\bar{r}) = \frac{\bar{u}(\bar{r} + \bar{n} \cdot d) - \bar{u}(\bar{r} - \bar{n} \cdot d)}{2d} \tag{3.2}$$

Parameter d is set to $0.5\tilde{a}$, where $\tilde{a}$ is the characteristic length defined as the edge of the cube in case of cubic lattice, or is estimated as the distance between nearest neighboring nodes in case of random node distribution.

3.1.2 Stress

Measurement of stress constitutes a bigger challenge. Our application uses the method proposed by Hardy [12, 45], which we observed to give more accurate results than any other method tested. Instead of calculating the tensor from forces that act on nodes, the algorithm estimates its value by analyzing the tension inside the probe sphere placed in the point of interest. Tension in each spring is weighted by the bond function. If only a fraction of the spring is contained within the probe sphere, the corresponding fraction of the tension is taken into account:

$$\hat{\sigma} = \frac{1}{2} \sum_{\alpha=1}^{N} \sum_{\beta \neq \alpha}^{N} \bar{x}_{\alpha\beta} \otimes \bar{F}_{\alpha\beta} B_{\alpha\beta}(\bar{x}), \tag{3.3}$$

where:
$x_{\alpha\beta}$ – distance between nodes α and β
$\bar{F}_{ab}$ – force acting on the spring ab
$B_{\alpha\beta}$ – *bond value*, weighted length of the fraction of the bond between nodes, that lies within the characteristic volume.

In Hardy's technique with spherical step probe stress error vanishes faster than 1/R. When we increase the radius, spring density on the surface should remain constant. Only then, together with linear bond we get constant stress independent on the radius. However due to discretization, both of these factors hold much better for big surfaces (large R) than small ones.

Radius of the probe sphere that we use in our measurements, has been set to $1.3\tilde{a}$; parameter $\tilde{a}$ is defined in the same way as above. We have found that such radius gives well localized readings at the same time keeping errors at small levels.

Some solution to improve readings for smaller radiuses is to measure the bond not as a linear overlap of the spring and a probing sphere, but instead treat the spring as a sphere (with a center in the middle of the spring and a radius $R = 0.5L$) and measure overlap between two spheres. This allows for much more localized readings with the radius of the probe sphere $0.6\tilde{a}$ still giving good results.

3.2 Tests

In this section, we will compare the theoretical predictions of Eqs. (2.23) and (2.24) with the corresponding values measured as a result of a numerical experiment (which we will refer to as *measurements*). The numerical experiments mimic the real experimental setup used to obtain a given elastic modulus, e.g. to obtain the Young's modulus and Poisson's ratio we measure the deformation of the material during uniaxial compression, as reported in detail below.

3.2.1 Uniaxial compression

In order to estimate numerically the values of E and ν the following numerical experiment has been performed: a block cut out of an elastic material was compressed by applying static displacement to its opposite ends along x direction, and the resulting deformation was measured both for random and cubic MSMs.

The Young's modulus and Poisson's ratio are related to the elastic response of such a system by

$$E = \frac{F/A}{\Delta x/L_x} \qquad \nu = \frac{\Delta y}{\Delta x}$$

where F is the reaction force, A the cross-sectional area of the block (in yz plane), and Δx and Δy are the deformations of the block along x and y directions respectively. The initial block dimensions were $70a_0 \times 15a_0 \times 15a_0$ (where a_0 is an arbitrary unit of length) and the base spring constant has been set to k_0. The static displacement in x direction was imposed on all the nodes within $1.5maxL$ from the boundary of the block.

The value of Δy could be measured directly by comparing displacements of extreme points, however we have not taken any steps to ensure that random MSMs have sufficient connectivity at the boundaries. It happens occasionally that a point with low connectivity behaves like a "floppy tail" introducing additional uncertainty to such measurements. To avoid this problem, we have measured an average expansion Δy of a specific region in the center of the block (by comparing positions of points in deformed and undeformed blocks). The width of this region was set to 80% of the width of the block. The reaction force was estimated by measuring the momentum flow through the YZ plane in the middle of the system. (By momentum flow across a given plane we understand the sum of the forces in all the springs intersecting the plane projected on the vector normal to that plane).

3.2.2 Cubic MSM – Resolution scaling

The purpose of the following test is to confirm that we are capable of changing the resolution of the system while keeping its physical characteristics (E and ν) constant. We will consider a *cubic lattice MSM* characterized $E = 1[k_0/a_0]$, and $\nu = 0.25$. The numerical results are presented in table 3.1. As we can see, both E and ν match theoretical values very well, with relative errors below 2%. Notably, the errors for high resolution MSMs are slightly larger than for low resolution models. This is most likely caused by the minimizer, which works more

efficiently for low resolution models, and may stop short before finding a true minimum in case of high resolution ones. For high resolution models a single spring contributes much lower value to the total energy and the precision errors are greatly increased.

Table 3.1: *Cube lattice* MSM with various resolutions. Numerical results for *Young's modulus* and *Poisson's ratio* measurement. Theoretical values: $E = 1[k_0/a_0]$, $\nu = 0.25$. Measurement uncertainty for E is estimated as 4% of measured value, for ν as 3%.

elemental cube edge length $[a_0]$	1.0	0.5	0.25	0.125
E $[k_0/a_0]$	1.012	1.013	1.013	1.013
ν	0.2506	0.2507	0.2509	0.2509

3.2.3 Cubic MSM – Isotropy

In this test, a block of material as described in Sec. 3.2.1, has been modeled with *cubic lattice MSM* rotated around arbitrarily chosen axis, by various angles (the same axis for every test). Results are presented in Figures 3.1 and 3.2, and they confirm isotropic properties of the material.

3.2.4 Random MSM

In this test a block of material as described in Sec. 3.2.1 is modeled with various random MSM models. The results are shown in the Fig. 3.3 and Fig. 3.4. Two sets of test cases are presented there, with slightly different resolutions ($n = 1.29$ and $n = 0.85$ nodes per unit volume). The values of E and ν are plotted against the average number of springs attached to a node, $< S >$. In most of the tests, the value of Poisson's ratio was equal to 1/4 with a good accuracy (within 2%). Only models with very low $< S >$ diverged from this value by more than 5%. As we can see in Figs. 3.3 and 3.5, values of E estimated with the use of equation (2.24) are in a very good agreement with the measured ones. The relative error is smaller than 5% for almost all the models with $< S >$ between 15 and 25. However, for smaller $< S >$, discretization errors increase significantly. There

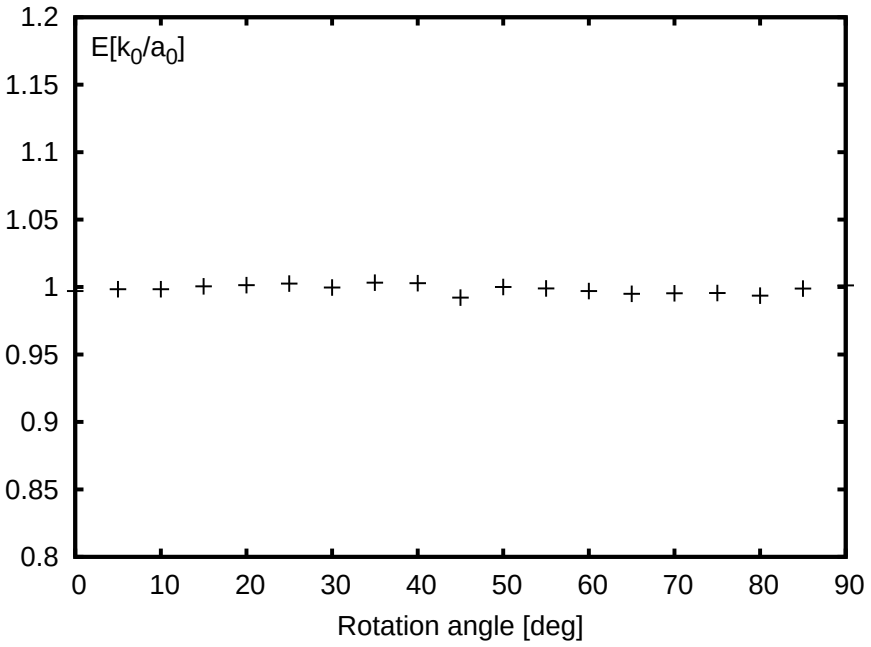

Figure 3.1: Cubic lattice MSM isotropy test. Material represented with lattice rotated along (0.4, -0.3, 0.7) axis. Young's Modulus $E = 1[k_0/a_0]$.

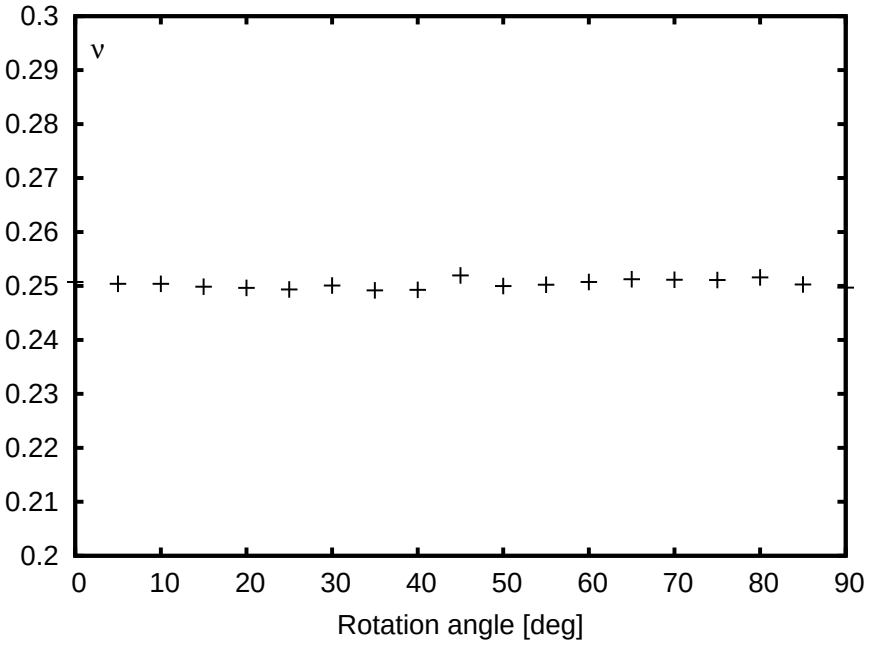

Figure 3.2: Cubic lattice MSM isotropy test. Material represented with lattice rotated along (0.4, -0.3, 0.7) axis. Poisson's ratio $\nu = 0.25$.

is thus a trade-off between the accuracy and computational costs: larger values of $< S >$ offer a more accurate representation of the material, which comes, however, at a price of an increased computation time.

As we can see in the experiments, for low connectivity networks, increasing $< S >$ increases the accuracy of the estimate in Eq. 2.23. However, after $< S > \approx 15$ is reached, further increase of the number of springs does not improve accuracy. This also means, that for a given number of springs S (spring budget), networks with $< S >= 15$ will offer the highest resolution (highest N; most detailed models), while keeping the accuracy at the highest level (provided that the total number of nodes N, is sufficent to form a stable network). One needs to be aware, though, that these estimates are based on the numerical experiments with the random networks constructed according to the procedure presented in Sec. 2.3.3, and it is conceivable that another way of constructing MSMs might lead to a smaller value of optimal $< S >$.

3.2.5 Point force in an infinite solid

To further validate the model, we have considered a classical problem of linear elasticity, in which a point force is applied to an elastic solid of an infinite extent [23, 3]. The strain and stress tensors are then given by:

$$\begin{aligned}\epsilon_{ij} =& \frac{-(1+\nu)}{8\pi E(1-\nu)R^2}\Big(\frac{3F_k x_k x_i x_j}{R^3} - \frac{F_k x_k \delta_{ij}}{R} + \\ &+ (1-2\nu)\frac{F_i x_j + F_j x_i}{R}\Big),\end{aligned} \tag{3.4}$$

and

$$\begin{aligned}\sigma_{ij} =& \frac{1}{8\pi(1-\nu)R^2}\Big(\frac{3F_k x_k x_i x_j}{R^3} + \\ &+ (1-2\nu)\frac{F_i x_j + F_j x_i - \delta_{ij}F_k x_k}{R}\Big),\end{aligned} \tag{3.5}$$

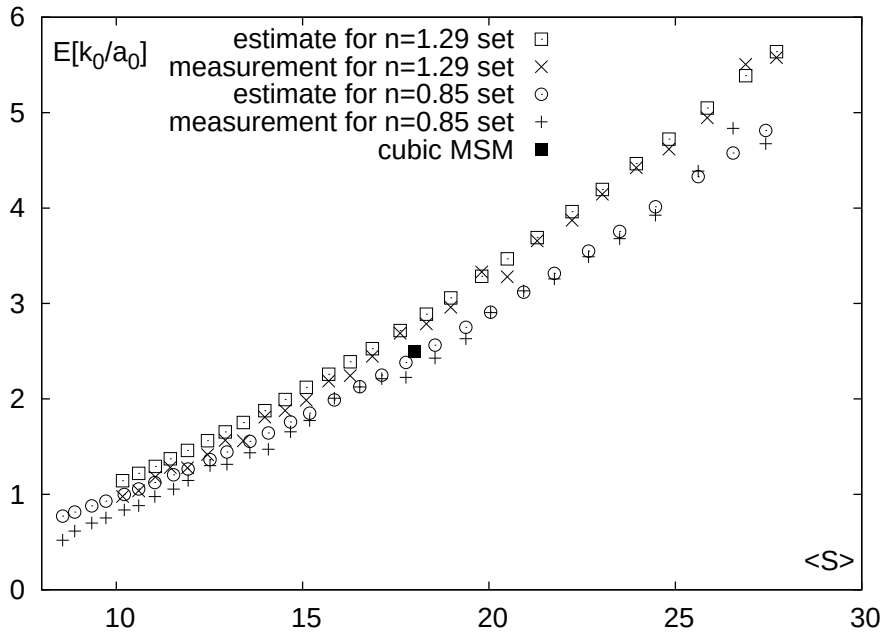

Figure 3.3: Young's Modulus for random MSMs with two different $minL$ parameters (varying $maxL$). Set A (open squares) with $minL = 0.8a_0$ and $n = 1.29$ nodes per unit volume; set B (circles): $minL = 0.9a_0$, $n = 0.85$. The estimates based on Eq. (2.24) as well as the values measured in the compression test are plotted as a function of the average number of springs per node $< \mathrm{S} >$. Measurement uncertainty 4%.

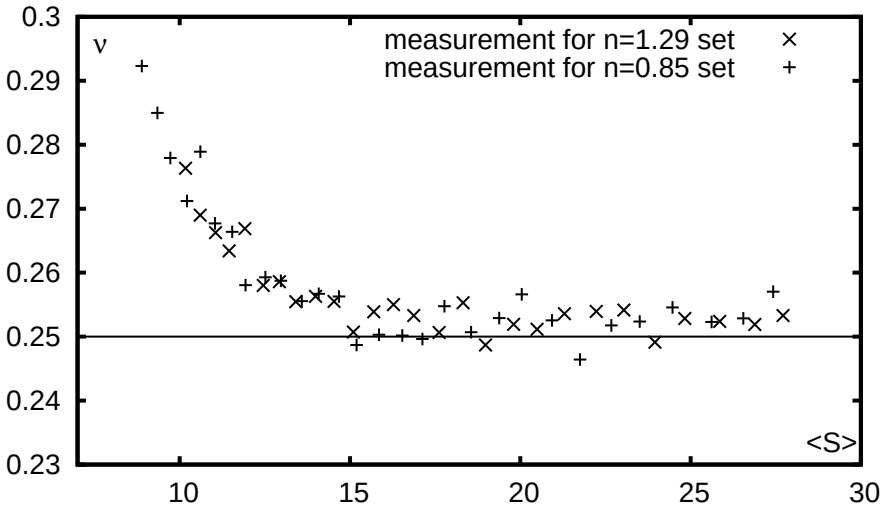

Figure 3.4: Poisson's ratio for random MSMs from Fig.3.3. Theoretical value $\nu = 0.25$. Measurement uncertainty 4%.

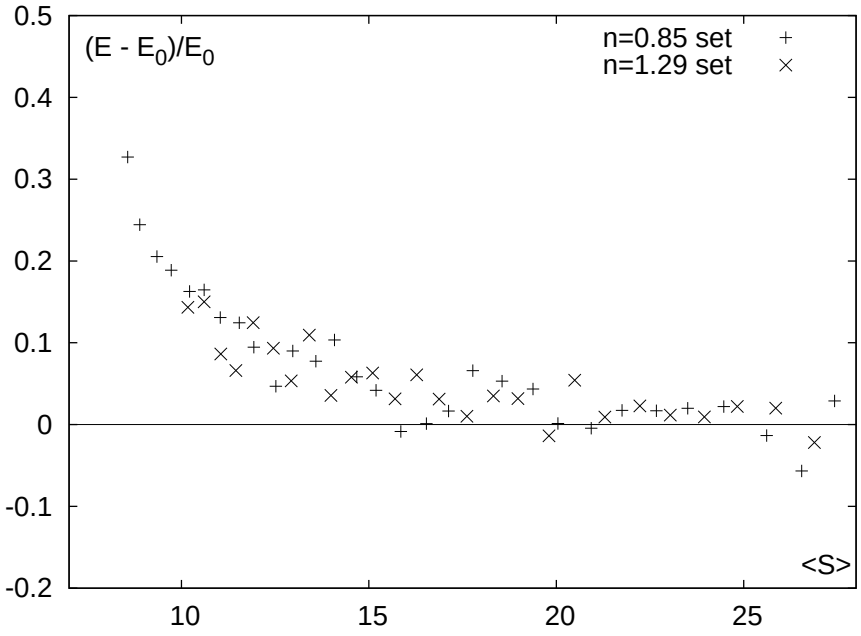

Figure 3.5: The relative difference between the values of E measured in the compression test and the estimates based on Eq. (2.24) for random MSM.

where $R = \sqrt{x_i x_i}$ and $\boldsymbol{F}$ is the force (applied at the origin).

In our tests, the solid characterized by $E = 100[\frac{k_0}{a_0}]$ and $\nu = 1/4$ has been modeled as a sphere with radius $5a_0$ represented by a cubic MSM model with $a = 0.25a_0$ and $< S >= 18$ as well as three random MSMs with $l \approx 0.25a_0$ and $< S >= \{11, 15, 18\}$, which we will refer to as S_{11}, S_{15} and S_{18} models.

A static force $F = 10[k_0 a_0]$ has been applied along y direction to the mass-point in the center of the sphere. Stress and strain tensor values were measured at 100 points uniformly distributed along the $[1, 1, 1]$ line from the center of the sphere to its border. Example profiles of strain ϵ_{12} and stress σ_{12} for random MSMs S_{18} and S_{11}, are shown in Figs. 3.6 - 3.9. Horizontal axes are scaled in such a way that 0 corresponds to the center of the sphere, and 100 – to the border. Smooth red line represents theoretical values. Stress measurements were carried out using Hardy's method [12, 45].

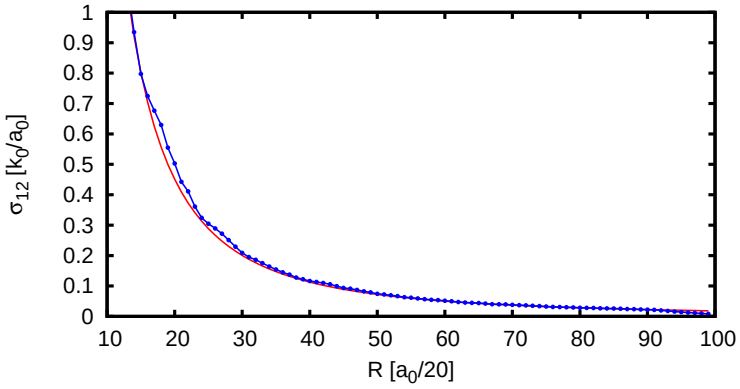

Figure 3.6: The stress component σ_{12} for S_{18} random MSM.

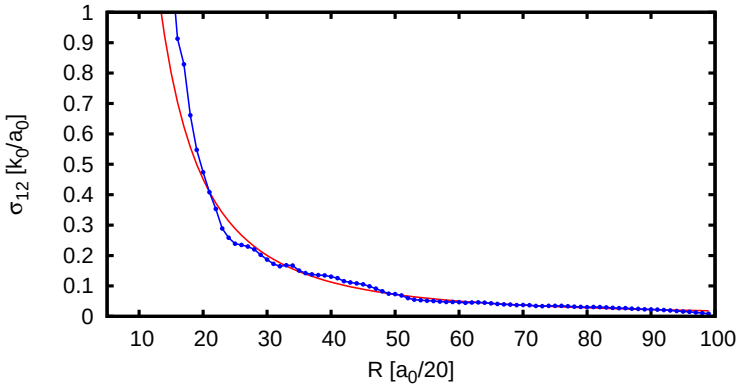

Figure 3.7: The stress component σ_{12} for S_{11} random MSM.

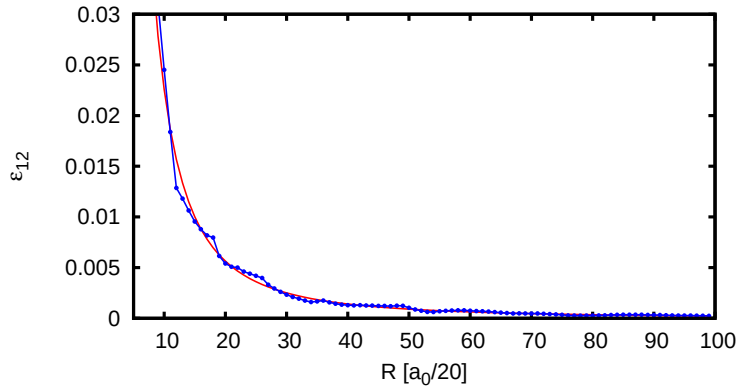

Figure 3.8: The strain component ϵ_{12} for S_{18} random MSM.

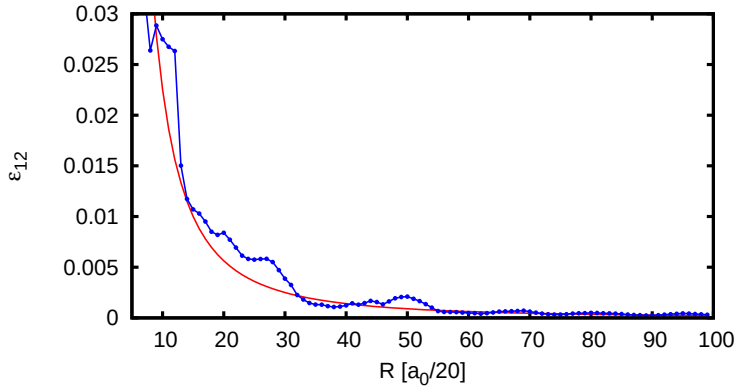

Figure 3.9: The strain component ϵ_{12} for S_{11} random MSM.

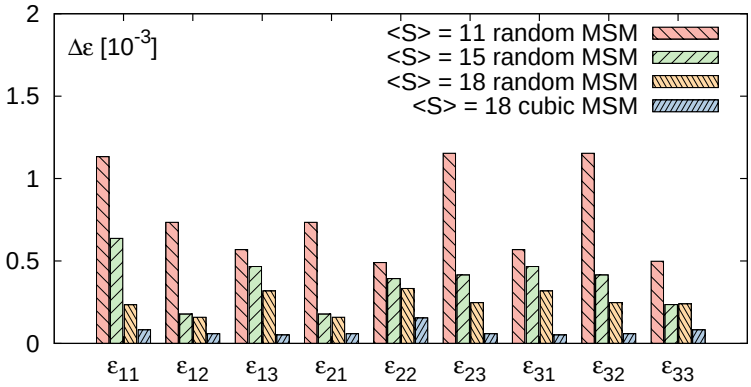

Figure 3.10: Mean deviation of the strain components measured in the MSM models from those calculated based on Eq. (3.4).

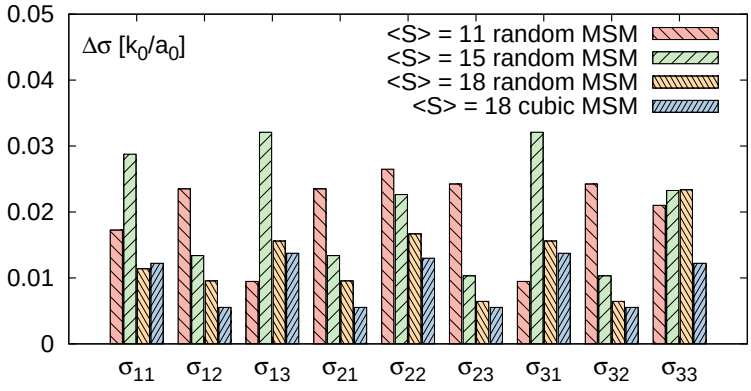

Figure 3.11: Same as in Fig. 3.10 but for the stress components.

Figures 3.10 and 3.11 present the mean deviation from theoretical values for the measurements of strain and stress tensors respectively. We concentrate on the region in the center of the sampling line (points 15-80), in order to minimize the effect of boundaries as well as the singularity at $\boldsymbol{r} = 0$.

For S_{18} model all the components of strain and stress tensors are in a good agreement with the theoretical predictions. As could be expected, random MSMs exhibit a somewhat more erratic behavior than cubic lattice MSM; oscillations around theoretical curves are present for all the test cases. For the strain tensor the errors systematically increase as $< S >$ is decreased. Stress components, however, do not seem to be affected by the change of $< S >$ to the same extent. In the random networks with low $< S >$ values, statistical noise plays a significant role, which results in the locally anisotropic and inhomogeneous structures, and corresponding inaccuracies in local node displacements. This, however, does not affect the momentum flow to the same degree, and the network still gives a good elastic response.

3.2.6 Deflection test

In order to verify how our models behave in a typical situation we have performed a deflection test, where a bar of elastic material with dimensions $4a_0 \times 1a_0 \times 1a_0$ bends upon a constant force applied perpendicularly to one of its ends. Besides comparing the numerically measured deflection with the theoretical predictions, one can also directly assess the visual quality of the model.

A deformation of such beam is given by [23]:

$$\Delta y = \frac{F}{E} 2x^2(3L - x), \tag{3.6}$$

where F is the applied force, E Young's modulus, L length of the beam and x denotes the distance from the fixed end of the beam. The equation holds for deflections which are small (the radius of the bend) when compared with the thickness of the beam.

As we have mentioned in section 2.3.3, many popular discretization approaches treat a continuous model as composed of non-overlapping volumetric primitives. This includes cubic lattice described in section 2.3.2. This experiment exposes a weakness of such approaches. To illustrate the problem let us consider a small segment of a bent beam as shown in the Fig. 3.12. The upper half of the material is stretched, the lower is compressed and in the middle there is a neutral surface, which does not experience any tension. The total momentum flow through both lower and upper parts is proportional to the force applied at the far end. In continuous model the tension increases linearly with the distance from neutral surface, and the total momentum flow can be calculated as an integral from zero to σ_{max} present near the surface (shaded triangle in Fig. 3.12). On the other hand, in discretized system the flow does not change continuously, but is localized on the springs. In case of a beam, which thickness is represented by a single cubical element, this means that the whole momentum flow of the upper and lower parts will be localized on one spring respectively (shaded square on the Fig. 3.12). For these two flows to match (continuous and discretized), the strain on the spring has to match the strain in the middle between the neutral surface and the border of a continuous beam. This means that the force applied to low resolution cubic MSM has to be two times larger than corresponding force applied to continuous model, to achieve the same deflection. If the resolution of the discretized model is increased, the stress gradient can be approximated more accurately and the response to a bending force will be closer to the response of continuous model, with the bending force obeying $F_a = F/(1 - 0.5a)$, where a is the length of elementary cube, the thickness of the beam is assumed to be 1 and F is the force applied in the continuous model.

The node-oriented discretization approaches, in which mass of a region of the material is localized in the center of this region, and not distributed into its corners, do not suffer from this artefact. This includes random MSM described in Sec. 2.3.3. Cubic lattice MSM from Section 2.3.2 can be adapted easily as well, by following Eq. 2.23. The nodes would no longer be placed on the surface of an object, but in the distance $0.5a$ within it (see the leftmost model on the Fig.3.14). For the example from Fig. 3.12 it means that springs will be already

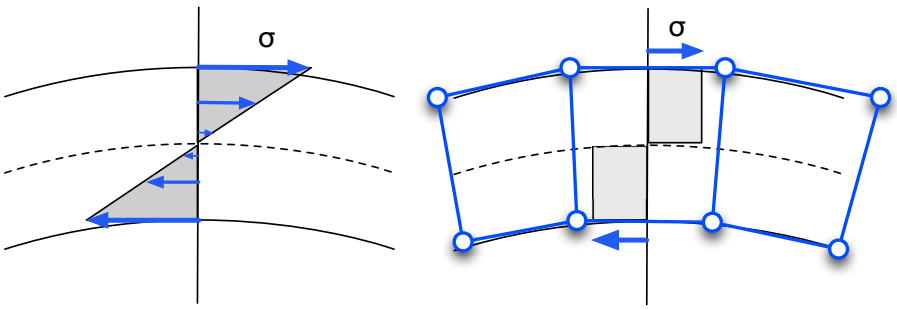

Figure 3.12: A segment of a bent beam. Left - continuous model, right - lowest possible resolution of cubic MSM.

placed in the middle between neutral surface and the border of the beam and will correctly approximate the momentum flow present in continuous system.

Numerical experiments confirm that the deformation of such models depends only weakly on the resolution of the network. We have tested various resolutions for both cubic and random MSMs. Example profiles of small deflections of four chosen models are presented in Fig. 3.13, together with theoretical curve. Small differences between low and high resolution cubic MSMs are most likely caused by the change of effective length of the beam caused by discretization (the force is no longer applied precisely at the end of the beam, but at the point $0.5a$ inside of the beam). We set the origin of x-axis to be in the middle of the frozen region; for random MSM this region extends to $1.1minD$ from the end of the bar, for cubic MSM it is simply the leftmost layer of nodes. Medium resolution random MSM curve should be treated as having 5% uncertainty, and low resolution one as having $10-15\%$ (based on the changes observed between different runs of the test).

Figure 3.14 shows results for a bigger force with large deflection. Even though Eq. 3.6 no longer holds, the deformation of all MSMs remains consistent among different resolutions and MSM types. In all the conducted tests, various resolutions of cubic MSMs behave almost identically. Both medium and low resolution random MSMs behave surprisingly well, exhibiting relatively small deviations from lattice models, which shows that even such unsophisticated method of generating MSM allows one to obtain models with a decent quality.

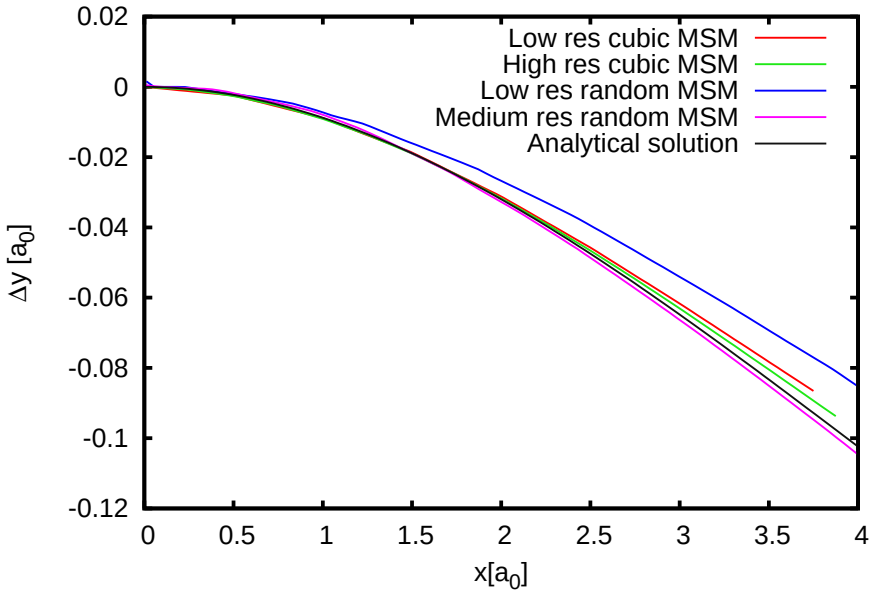

Figure 3.13: Deflection test for four different MSMs. Low resolution cubic MSM with the edge of an elemental cube $a = 0.5a_0$, number of mass points $N = 32$ and $S = 132$ springs. High resolution cubic MSM, $a = 0.25a_0$, $N = 256$, $S = 1632$. Low resolution random MSM, $N = 57$, $S = 258$. Medium resolution random MSM, $N = 161$, $S = 934$. Applied force $F = 0.1[k_0a_0]$, Young's modulus $E = 250k_0/a_0$. The origin of the x-axis is placed in the middle of the frozen region.

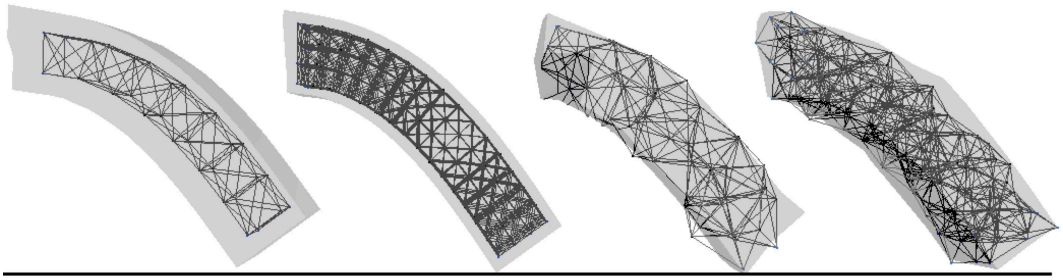

Figure 3.14: Deflection test for four different MSMs. In each panel, a wire frame of a MSM model of an elastic bar of dimensions $4a_0 \times 1a_0 \times 1a_0$ is shown together with visual representation of the bar (a mesh within which the MSM is embedded). From the left: **1)** Low resolution cubic MSM, **2)** High resolution cubic MSM, **3)** Low resolution random MSM, **4)** Medium resolution random MSM. Numbers of nodes and springs same as in Fig. 3.13. Applied force $F = 4.1[k_0a_0]$, Young's modulus $E = 250k_0/a_0$.

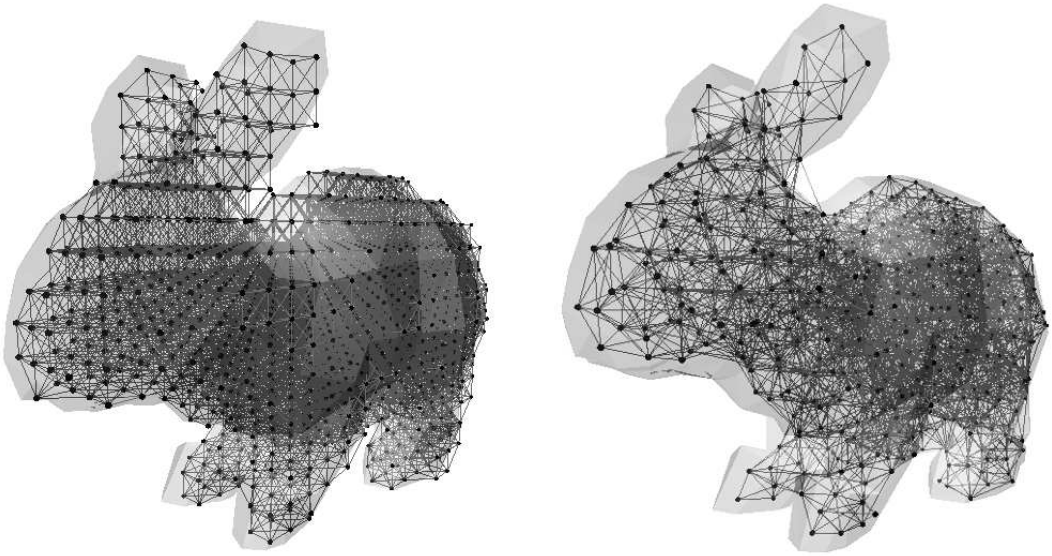

Figure 3.15: Two representations of a rabbit. Left: cubic lattice MSM, N = 2055 nodes. Right: disordered MSM N = 888.

3.2.7 Complex shapes

In this test we verify our ability to construct complex objects with non-trivial shapes using MSMs of various resolutions and types. We embed a MSM into a triangular mesh of a rabbit (Fig. 3.15).

Next we apply a force to its head and observe the resulting deformation for three levels of detail. Results can be seen in the fig. 3.16 and are very satisfactory. Even though we don't have an analytic solution for such deformation to compare to, the fact that resulting deformation changes very little with the resolution of the MSM, shows that our models give consistent response and converge very fast to the high-resolution solutions.

3.3 Conclusions

In this chapter, we have measured the elastic properties of both cubic and random lattices in a series of simple test simulations and shown that they agree satisfactory with theoretical predictions and thus are well suited for the modeling of elastic materials. The random MSMs seem to perform almost as good as the

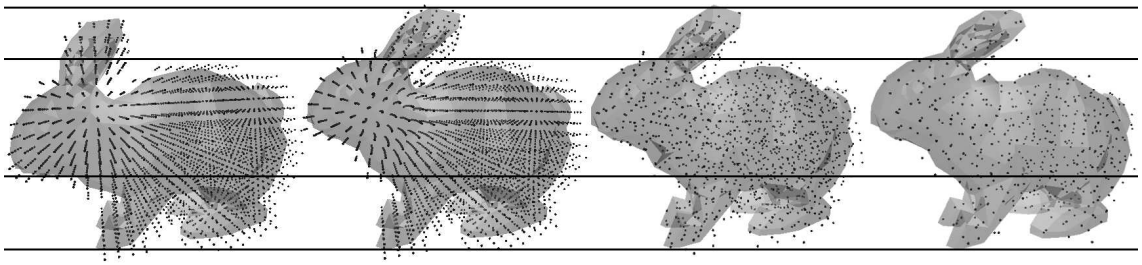

Figure 3.16: A force is applied to rabbit's head pulling it backwards. From the left: initial undeformed hi-res model (2479 nodes), next, resulting deformation of the hi-res model, mid-res (1085 nodes) and low-res (482 nodes) models.

cubic ones when the global deformations of the system are concerned, but slightly underperform at the smaller scales (of the order of few internode distances). Even though the difference is not large, it does show that MSMs, particularly low resolution ones, require attention during generation process in order to minimize discretization errors. Overall, however, the performance of random models is good enough for them to be an attractive alternative to the cubic models, as in many situations a slight decrease of accuracy may be an affordable price to pay for the flexibility of choosing node positions.

Chapter 4

Extended mass spring models

Mass spring models presented in chapter 2 have quite limited applicability. The first limitation comes from the linear elasticity itself – if we assume that the reaction force is linearly proportional to the relative displacement, deformations with large strain will be non physical. If a strong force is applied to simple MSM, it may easily displace mass points to such degree, that their arrangement within the network will reverse, causing deformations that are plainly incorrect even by the standards of computer graphics. Ironically, the higher resolution of the network is, the more susceptible MSM will be to such behavioral artifacts. High frequency momentum waves, which should be accompanied by very localized displacements, will simply average themselves over low resolution network causing it to respond with different frequency, which will most likely be barely noticeable in an animation. On high resolution networks on the other hand, they will cause node reversal and the MSM behavior will be obviously wrong.

The second limitation of simple MSMs comes from the central force interactions between network nodes. As we have shown in chapter 2 such forces result in the fixed value of Poisson's ratio for isotropic homogeneous materials.

In this chapter we will show how to modify *simple MSM*, to overcome these two problems. Our focus stays on homogeneous isotropic materials. Networks presented in this chapter potentially could be used to represent anisotropic materials as well, however this aspect will not be analyzed here.

4.1 Other works

The problem of constructing MSM-like models capable of representing isotropic materials with arbitrary values of Poisson's ratio have been explored before by a number of researchers. As elucidated in Chapter 2, such approaches require incorporating additional elements into the model, besides ordinary springs [1, 7, 22, 44]. An overview of well established techniques can be found in [36]. Typically additional degrees of freedom are added to the model by incorporating beams, angular springs and other custom three or four node connections which react not only to stretching, but shearing or applying toque as well. Introducing such elements allows to obtain wide range of materials, but it increases the complexity of the model as well as computational costs.

In contrast in this work we will explore an approach of constructing MSM which allows to achieve arbitrary values of ν, without introducing additional structures into the model. Our approach will in fact make use only of nodes and relative pairwise distances.

4.2 Particle model of matter

The development of the elasticity theory was accompanied by speculations concerning the nature of matter and the nature of forces (interactions). It was unknown to scientists at that time whether the matter was continuous or composed of particles, and how gases, liquids and solids related to each other. Many models were studied; the ones involving particles usually assumed that they would interact with each other by means of central forces. Cauchy showed in the middle of 19th century, that solids represented in this way would have Poisson's ratio equal to 1/4. This caused a debate, lasting many years, on the problem of how many parameters are necessary to characterize elastic materials. The multi-constant theory assumed that 21 parameters are needed to characterize aeolotropic elastic material, and 2 for the isotropic one. The rari-constant theory assumed 15 and 1 constant respectively.

4.2.1 Overview

Let us recap a historical overview as given by Love [28] in the very beginning of 20th century:

Navier was the first to investigate the general equations of equilibrium and vibration of elastic solids. He assumed the Newtonian conception of the constitution of bodies (interacting particles) and that the elastic reactions arise from variations in the intermolecular forces upon changes in the molecular configuration. The force was given by $F = f(\bar{r_0}) \cdot \Delta x$.

(...)

Objection has been raised to the procedure of replacing summations by integrations.

(...)

By the Autumn of 1822 Cauchy has discovered most of the elements of the pure theory of elasticity. He introduced the notion of stress at a point.

(...)

The methods of Navier, of Poisson, and of Cauchy's later memoirs lead to equations of motion containing fewer constants than occur in the equations obtained by the methods of Green, of Stones, and of Cauchy's first memoir. The questions in dispute are these – Is elastic aeolotropy to be characterized by 21 constants or by 15, and is elastic isotropy to be characterized by two constants or one? (The multi-constant theory and the rari-constant theory.)

(...)

Poisson's theory leads to the conclusions that the resistance of a body to compression by pressure uniform all round it is 2/3 of the Young's modulus [1]*. Stokes observed that resistance to compression and resistance to shearing are the two fun-*

[1] From which follows that $\nu = 1/4$.

damental kinds of elastic resistance in isotropic bodies. He introduced definitely the two principal moduli of elasticity by which these resistances are expressed – the modulus of compression *and the* rigidity *as they are now called.*

(...)

According to Green there exist, for a material of the most generally aeolotropic character, 21 independent elastic constants. The molecular hypothesis, as worked out by Cauchy leads to 15 constants. In such case there must be 6 independent relations among Green's 21 coefficients (Cauchy's relations).

The direct consequence of the rari-constant theory, is that all isototropic elastic solids would have a Poisson's ratio $\nu = 1/4$. For many years physicists were however unable to determine with certainty if this is the case, due to technical limitations of experimental techniques. One problem was to measure the value of ν, the other to determine whether or not the material was actually isotropic.

The evidence was pointing towards multi-constant theory (Love 1906 [28]):

The best modern experiments support the conclusion that Poisson's ratio can differ sensibly from the value $1/4$ *in materials which may without cavil be treated as isotropic and homogeneous.*

This raises however a question of how the constituents of a material interact with each other, if not by means of central forces. Classical mechanics seemed to be unable to provide a good answer to such question, and the laws of quantum mechanics had to be adopted (Rowlinson 2005 [35]):

The result of a hundred years of debate is that it is now established that the Cauchy relations hold for a system of particles to which classical (not quantum) mechanics apply, which owes its cohesion to pairwise additive central forces, which adopts a stable structure in which each particle is at a centre of symmetry of the whole lattice, and which initially is in a state free from strain.

In this chapter, we show that quantum mechanics is not necessary to explain the multi-constant theory of elastic matter. A consistent model can be constructed with classical mechanics only, however slight modifications of the assumptions made by Cauchy are necessary.

4.2.2 Particle interaction model

In contrast to 18th century, today we know that on the scale relevant to classical elasticity, the matter can be treated as composed of particles. From particle physics we know, that interactions can also be treated as being carried out by means of traveling particles (e.g. photons, bosons).

The underlying model of matter that we will use for constructing our discretized models and explaining their properties is following. We assume the material to be composed of particles which interact with each other by means of exchanging interaction carriers.

We will use the terminology of "particles" and "carriers" in the following section, having in mind that "particles" and "carriers" are just convenient terms for expressing certain interaction mechanisms; they may not correspond to actual physical, existing entities. "Carriers" serve as mediums of transferring momentum/energy and may be considered to be a different type of particles, larger in number and smaller in size then the "particles". On the other hand in some situations "particles" may be considered to be clusters of "carriers" (and even clustered only conceptually, that is not really forming clusters in the sense of density).

This basic concept is expected to approximate wide range of materials and interaction mechanisms throughout a range of scales. On one extreme we can imagine grains of sand in a rock, and on the other atoms in a metallic crystal.

4.3 Non-linear springs

Having in mind our particle-carrier approximation, let us analyze a flow of momentum between two particles with masses M, caused by the interaction with

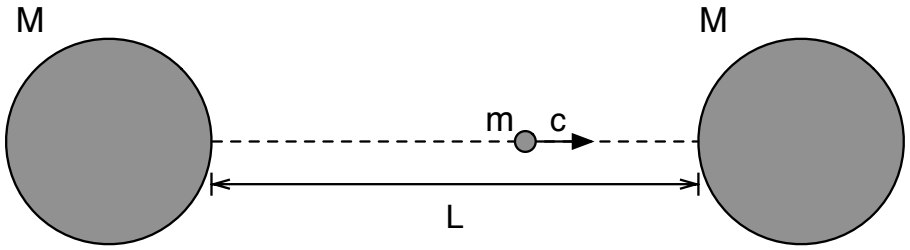

Figure 4.1: Momentum transfer between two large particles.

carriers with mass m. To avoid too many parameters, lets assume that the speed c of a carrier is constant and the momentum is transferred by perfectly elastic collisions, so that each time a carrier collides with a particle, it gains a quant of momentum $\Delta p = \pm 2mc$ (Fig. 4.1).

Let us assume, that external force is applied to both particles M, such that it equalizes the momentum inflow from carriers, and the system is in an equilibrium state. The flow of momentum between the particles can be expressed as

$$j_p = N_c \frac{c}{L} mc, \tag{4.1}$$

where N_c denotes the number of carriers and L the distance between particles (as in Fig. 4.1). The force which accompanies a displacement from the equilibrium distance L_0 can be expressed as

$$F = N_c mc^2 \big(\frac{1}{L} - \frac{1}{L_0}\big) = N_c mc^2 \frac{-\Delta L}{(\Delta L + L_0) L_0}, \tag{4.2}$$

which for small displacements has a behavior of regular spring $F = -\kappa \Delta L$. As we can see $F \to \infty$ when $\Delta L \to L_0$ and the network will no longer collapse under heavy loads.

In the above model, classical spring coefficient is expressed as:

$$\kappa = \frac{N_c mc^2}{L_0^2}. \tag{4.3}$$

Moreover

$$\frac{N_c m}{L_0} = \rho, \tag{4.4}$$

where ρ is the 1-d mass density of carriers, and if we assume $M = 0$, of the material. (In such case big particles serve only as reference points, and are not treated as existing matter.)

The carrier speed c

$$c^2 = \frac{\kappa L_0}{\rho} = \frac{E}{\rho} \tag{4.5}$$

can be recognized as a speed of a wave propagating in a continuous material characterized by Young's modulus E and mass density ρ.

The energy of the deformation is given by

$$W = \int F d(\Delta L) = N_c m c^2 (\frac{\Delta L}{L_0} - \ln(1 + \frac{\Delta L}{L_0})) \approx \frac{k \Delta L^2}{2} \tag{4.6}$$

$$log(1 + x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n} = x - \frac{x^2}{2} + \frac{x^3}{3} + ... \tag{4.7}$$

4.4 The second elastic parameter

Simple MSM described in chapter 2 can be viewed as a network in the Fig. 4.2. Forces in such networks act along straight lines between particles, and as we know, this leads to $\nu = 1/4$, which may be considered a geometric property of the space (the way distance to neighboring nodes changes, when we change the position of a node).

We generalize this model by introducing additional phenomenon. When two nodes approach, they start repelling each other as usual, and additionally each of them starts radiating momentum in "random" directions.

To visualize how we can justify such behavior, let us recall the particle interaction model from section 4.2.2. Carriers interact with particles (and with each other) by means of central forces (elastic collisions), therefore this model still uses

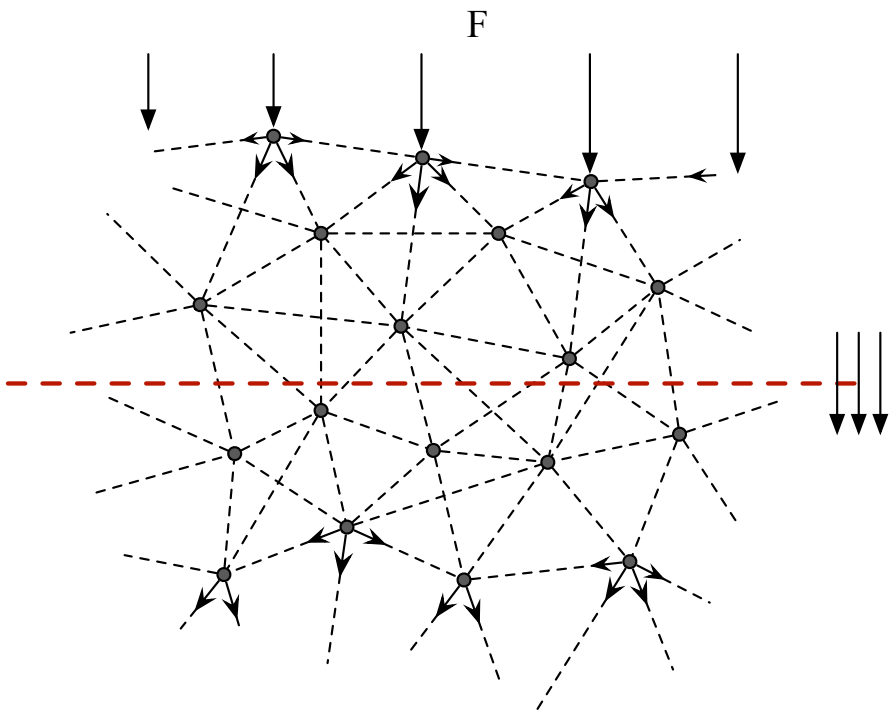

Figure 4.2: Momentum flow through a simple MSM.

central forces as most basic means of interaction. Carriers however have a finite velocity so the transfer of momentum does not happen instantaneously; the big particles move as well. Having this in mind, we may assume that a carrier shot form one particle, may not hit the other particle with a sniper's precision. In such case it will disperse in a "random" direction and will not come back to the first particle (Fig. 4.3). If a particle is bombarded from all directions uniformly, the dispersed carriers will also appear to be radiating from the particle uniformly (even if the actual distribution of dispersion angles is not trivial).

Figure 4.4 illustrates the dissipative part of the interaction, where each node takes the incoming carriers and radiates them uniformly in all directions. Such network will behave like a gas or fluid with no viscosity.

We will model this phenomenon using MSMs in the following way. When two nodes approach each other, causing increase of momentum flow between them,

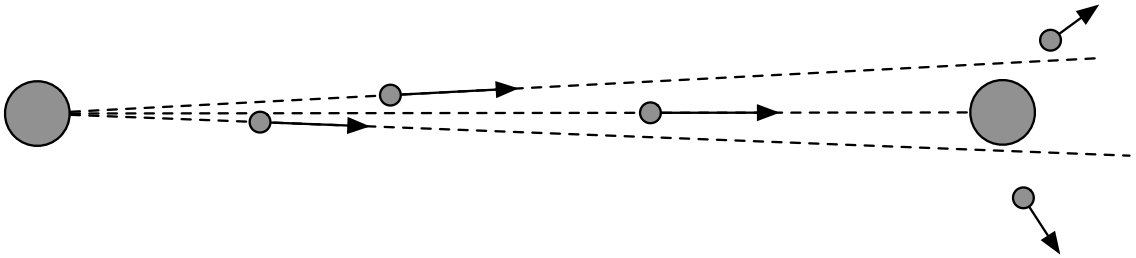

Figure 4.3: Momentum dispersion

only a fraction of the flow will participate in the direct exchange of momentum $F^{\mu} = -\kappa^{\mu}\Delta L$ between these two nodes, and appear as a regular force on the spring. The rest of the flow will be distributed uniformly to all springs connected to a node (scaled to the length of a spring), causing dissipation of momentum $F^{*} = -q\kappa^{\mu}\Delta L$. Parameter q denotes the ratio between dissipative momentum flow, and the direct one.

Material modeled in such way is now characterized by two parameters – μ gives the strength of direct interactions, and $q[\mu]$ of the dispersive ones.

The standard expression for a stress in an elastic body

$$\sigma_{ij} = K\delta_{ij}\epsilon_{kk} + 2\mu(\epsilon_{ij} - \frac{1}{3}\delta_{ij}\epsilon_{kk}), \tag{4.8}$$

can be rewritten, by treating our material as a superposition of Cauchy's isotropic structure, in which $\lambda = \mu$ with a fluid in which $\sigma_{ij} = B\epsilon_{kk}\delta_{ij}$, where $B\epsilon_{kk}$ corresponds to pressure.

$$\sigma_{ij} = \mu\delta_{ij}\epsilon_{kk} + 2\mu\epsilon_{ij} + B\delta_{ij}\epsilon_{kk}, \qquad B = \lambda - \mu \tag{4.9}$$

$$\sigma_{ij} = \frac{5}{3}\mu\delta_{ij}\epsilon_{kk} + 2\mu(\epsilon_{ij} - \frac{1}{3}\delta_{ij}\epsilon_{kk}) + B\delta_{ij}\epsilon_{kk} \tag{4.10}$$

The direct interactions will correspond to μ parameter, dispersive ones to B. Both can be controlled, which allows to achieve other values of Poisson's ratio:

$$\nu = \frac{B + \mu}{2(B + 2\mu)}, \tag{4.11}$$

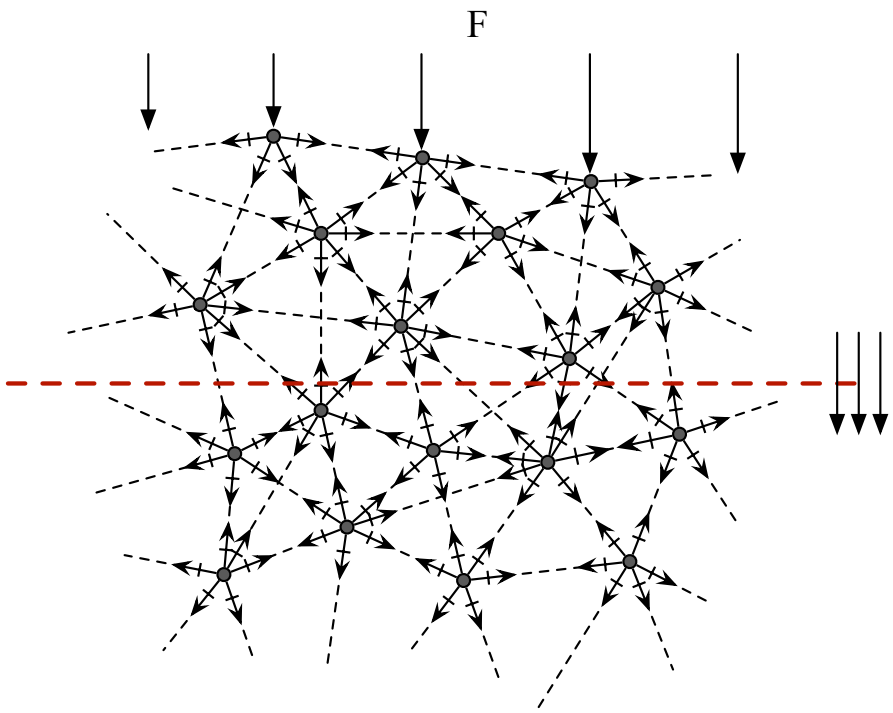

Figure 4.4: Dispersive momentum flow

$$\nu = \frac{1 + \frac{5}{3}Q}{2(2 + \frac{5}{3}Q)}, \qquad Q = \frac{B}{\frac{5}{3}\mu}, \tag{4.12}$$

where in a perfect structure $Q = 2q$, and it denotes the ratio between dispersed and not dispersed carriers. The multiplication by the factor of 2 is introduced for convenience (one spring connects two nodes).

Here we have assumed that the dispersion happens uniformly in all directions. In MSMs, it will translate into equal redistribution of the incoming force, to all the springs connected with a node (scaled by the length). Because redistribution is isotropic, so will be the elastic properties of the material. However by introducing non uniform dispersion mechanisms we can easily achieve non isotropic behaviors, without really extending the current model. Such networks however are not analyzed in this work.

4.5 Cubic lattice eMSM

As mentioned in the previous section, we incorporate the dispersion mechanism into MSMs, by redistributing fraction q of the incoming force for each node, to all the springs connected with it (it will act on both of the nodes connected by the spring).

For a cubic lattice, we can explicitly calculate the equations of motion by taking Taylor series in the displacement of neighboring nodes n and neighbors i of those as well.

$$u_{n\alpha} = u_\alpha + x_D \nabla_D u_\alpha + \frac{x_D x_E}{2} \nabla_D \nabla_E u_\alpha + ... \tag{4.13}$$

$$u_{ni\alpha} = u_{n\alpha} + x_{iB} \nabla_{iB} u_{n\alpha} + \frac{x_{iB} x_{iC}}{2} \nabla_{iB} \nabla_{iC} u_{n\alpha} + ... \tag{4.14}$$

The displacement of node i relative to node n is denoted as u_{ni}. The displacement of node n relative to the central node is denoted as u_n. The displacement of the central node is denoted as u. The total force exerted on a central node can be decomposed to a direct interaction force F^μ and dispersive one F^*:

$$F = F^\mu + F^* \tag{4.15}$$

The direct component is given by

$$F^\mu = -\kappa \sum_{n=1}^{b} g_n [(\bar{u} - \bar{u}_n) \cdot \hat{x}_n] \hat{x}_n, \tag{4.16}$$

which can be approximated as

$$F^\mu_\alpha = -\kappa \sum_{n=1}^{b} \frac{g_n}{2x_n^2} x_{n\alpha} x_{n\beta} x_{n\gamma} x_{n\delta} \nabla_\beta \nabla_\gamma u_\delta. \tag{4.17}$$

The dispersive component is given by

$$F^* = -\sum_{n=1}^{b} q_n z_n \hat{x}_n, \tag{4.18}$$

where z_n is the momentum inflow from direct connections and is in turn given by

$$z_n = \kappa \sum_{i=1}^{b} G_i(\bar{u}_n - \bar{u}_{ni}) \cdot \hat{x}_{ni}. \tag{4.19}$$

This gives the expression for the dispersive force

$$F^*_\alpha = \kappa \sum_{n=1}^{b} q_n \frac{x_{n\alpha}}{|x_n|} \sum_{i=1}^{b} G_i \frac{x_{niA}}{|x_{ni}|} (x_{iB} x_C \nabla_{iB} \nabla_C + \frac{x_{iB} x_{iC}}{2} \nabla_{iB} \nabla_{iC}) u_A. \tag{4.20}$$

For a cubic lattice with lattice spacing a, if we take $g_i = 1$, $G_i = q/L_i$ and $q_n = L_i/b$, where L_i is the length of i-th spring and b is the number of springs connected to a node, we get:

$$F^*_\alpha = \frac{10}{3} q\kappa a^2 \delta_{\alpha A} \delta_{BC} \nabla_B \nabla_C u_A \tag{4.21}$$

and $Q = 2q$ (note that $G_i = qL_i/b$ and $q_n = 1/L_i$ gives the same result).

This gives the desired total force acting on a node

$$F_\alpha = [\mu(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta}) + (\mu + B)\delta_{\alpha\delta}\delta_{\beta\gamma}] \nabla_\beta \nabla_\gamma u_\delta. \tag{4.22}$$

The results of experimental verification of this equation are presented in the Fig. 4.5. We have performed similar test to the one described in Sec. 3.2.1. It confirms that extended cubic lattice MSMs can achieve any value of Poisson's ratio (within classical limits of its value, that is form -1 to 0.5).

Additionally, figure 4.6 presents cuboid blocks made from four different materials with Poisson's ratio equal to -1, 0, 0.25 and 0.47. Their response to compression is visibly different, showing how materials with various ν deform under uniaxial compression.

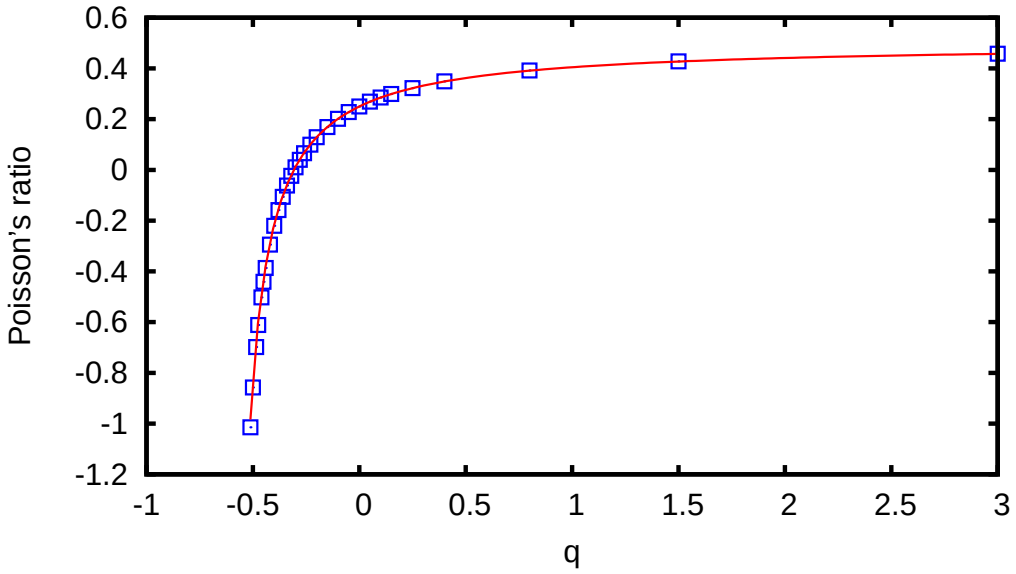

Figure 4.5: Poisson's ratio in Cubic lattice MSM. Red curve represents a theoretical prediction. Blue points are the results of measurements.

In some implementations taking $G_i = q$ and $q_n = 1/18$ (there are 18 connections per node), may reduce computational costs as it does not require evaluating lengths of the springs. The equation (4.20) then gives

$$F^*_\alpha = \kappa a^2 (2 + \frac{16}{18}\sqrt{2})\delta_{\alpha A}\delta_{BC}\nabla_B\nabla_C u_A \tag{4.23}$$

that is $B \approx 3.25\mu$, and $Q \approx 0.977 \cdot 2q$. Therefore q (the fraction of momentum redistributed to other nodes), does not exactly match with Q, the fraction of momentum dispersed in continuous isotropic structure, therefore values of q need to be slightly bigger than for the model with $2q = Q$. Materials with any ν can still be modeled reliably.

4.6 Disordered eMSM

In case of disordered networks relation $q(Q)$ cannot be determined so easily, as it will depend on the internal structure of the network (parameters such as deviation of the spring length from the average will play an important role). We can expect

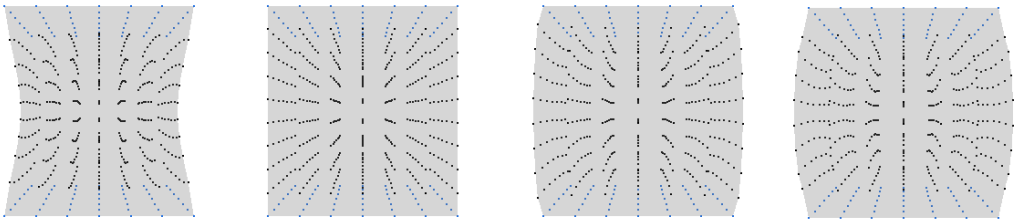

Figure 4.6: A block of material with dimensions $2\times3\times2$ compressed in y direction to 75% of its natural length. Different values of Poisson's ratio, from the left: -1, 0, 0.25, 0.47.

however, that $Q \approx 2q$, just like in case of cubic lattice models. Our tests show, that for networks with $G_i = q/L_i$ and $q_n = L_i/b$ we get $Q = 2q$ with high accuracy. Experimental values of ν for such networks are shown on the Fig. 4.8 (together with the theoretical curve from eq. (4.12)). The errors are minimal and theoretical ν is achieved with good accuracy.

A test similar to the one from Sec. 3.2.7 demonstrates the behavior of a rabbit with a force applied to its head. Two random MSMs with $E = 250[k_0/a_0]$, $\nu = 0$ and different resolutions perform reliably giving consistent results.

4.7 Conclusions

In this chapter we showed an approach of constructing MSM which allows to achieve arbitrary values of ν, without introducing additional structures into the MSM. We demonstrated that incorporating the concept of momentum dispersion additional degree of freedom can be added to the classical MSM, allowing it freely represent homogeneous isotropic materials characterized by two different constants (in contrast to one constant description given by classical MSM). Because our method does not introduce additional elements to the MSM itself, the memory costs remain unchanged and the computational costs rise by much lower degree when compared to methods which introduce additional elements to the MSM.

The simplest implementation of our model, when used with explicit time integrator, requires additional iteration through all the springs in the system,

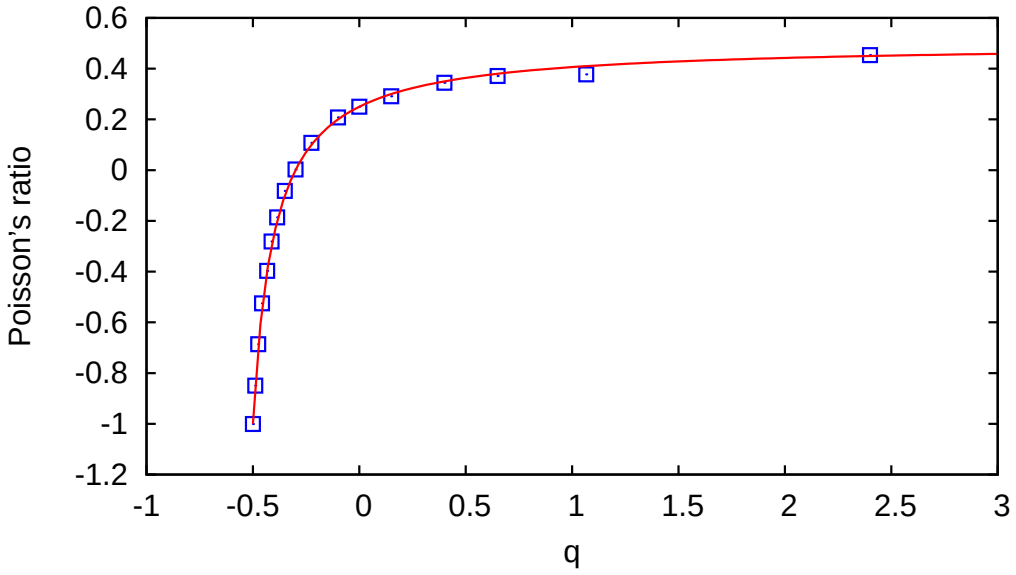

Figure 4.7: Poisson's ratio in random MSM. Red curve represents a theoretical prediction. Blue points are the results of measurements.

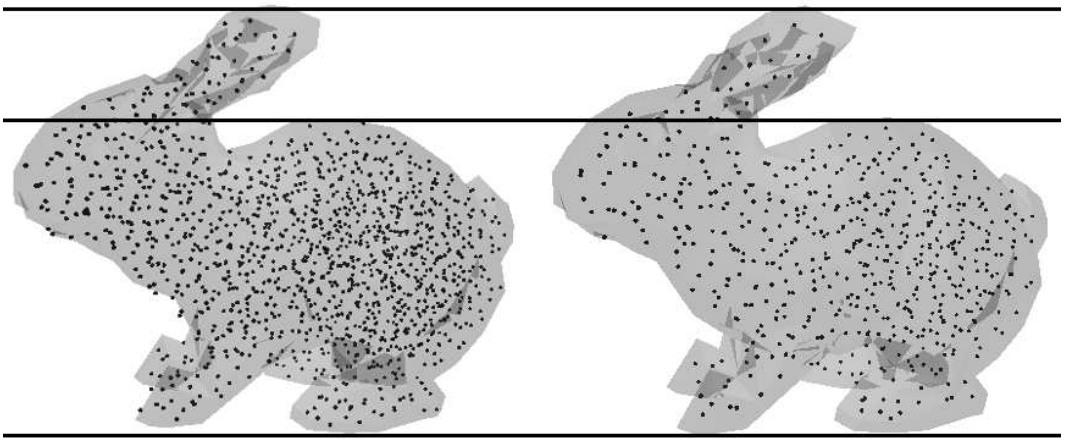

Figure 4.8: A force is applied to rabbit's head pulling it backwards. From the left: Mid-res random MSM 1321 nodes 24170 springs, Low-res random MSM 581 nodes, 9806 springs

effectively doubling the computational time when compared to the simple MSM. Some improvements are however possible; for instance the value of momentum dispersed on the nodes in the 'previous' frame could be used to approximate the value in the current frame, which eliminates the need of additional iteration through the springs, but may influence the stability of the simulation. However, in case of quasi-static simulation (such as point based integration techniques [2]), it should not affect the stability.

Chapter 5

Anti-aliasing

In this chapter we analyze an approach of constructing *cubic lattice MSMs* which takes into account "aliasing" effect [16]. Similarly as it is done in classical antialiasing techniques [6, 21], we weaken cubical elements used for representation of the surface regions and adjust them accordingly to represent the properties of the portion of the border they overlap with.

5.1 Anti-aliasing

Cublic lattice MSMs are simple and fast, but they suffer from the same problems as all discretization techniques. Usually node positions cannot be adjusted to fit border perfectly and in such cases, if the surface of the object does not coincide with faces of elemental cubes it will not be represented correctly (Fig. 5.1). If the resolution of the lattice is higher than a characteristic length of the surface features we want to capture, this will not be a problem. However for low resolution lattices, it is difficult to correctly model even simplest shapes even when they are perfectly aligned with the lattice axes (for example if we want to model a cuboid of dimensions 10 x 3.14 x 2.3 with a cubical MSM with lattice spacing equal to 1).

To overcome these limitations we propose the following procedure. For each cubical element we calculate the portion of its volume that overlaps with the

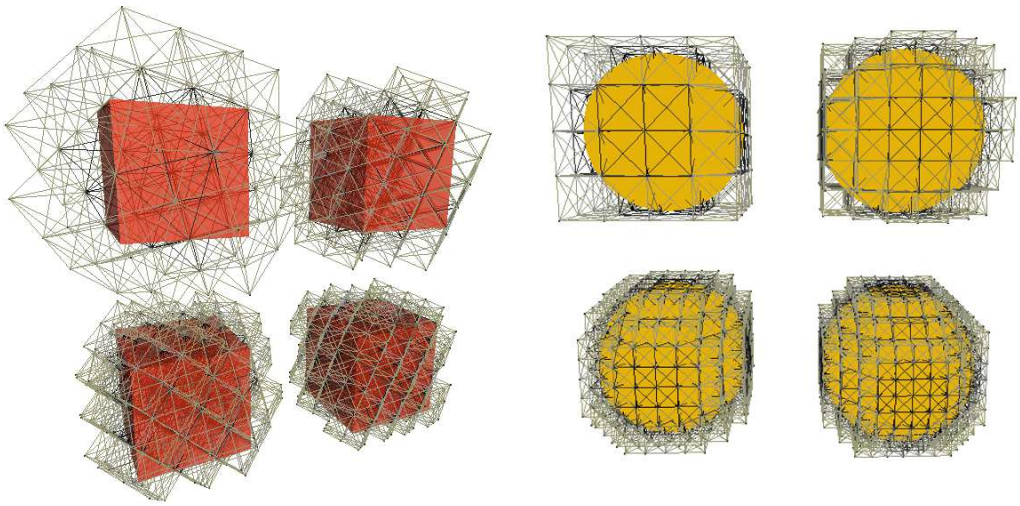

Figure 5.1: Examples of cubic lattice MSM with antialiasing. On the left a cube 1x1x1 represented by rotated cubic lattice MSM with four different resolutions, that correspond to node spacing $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$ and $\frac{1}{5}$. On the right, a ball with R=1 modeled with the same set of resolutions.

object, χ_i. Next, we renormalize both node masses and spring constants, $m'_i = f(m_i, \chi_i)$ and $k'_{ij} = g(k_{ij}, \chi_i)$ where f and g are certain functions of the masses, spring constants and overlap volumes.

This technique may be viewed as a volumetric anti-aliasing applied to border regions of the system. We have tested two variants of this procedure. In the first one, the mass corresponding to an elemental cube $m'_i = m_i\chi_i$ is distributed equally to the corners of the cube and all springs' coefficients are scaled by the same factor $(m_{overlap}/m_0)$.

In the second approach, we use $m'_i = m_i\chi_i$, but the MSM node is created in the center of the cube. Then we adjust springs' coefficients with $k'_{ij} = 0.5k_{ij}(\chi_i + \chi_j)$. Results produced by both of the approaches are similar, however the second one gives better response for test cases which invole bending, such as the one presented in Section 3.2.6 (drawbacks of corner representations of elemental cubes are discussed in that secion as well).

Figure 5.1 shows two simple shapes of a cube and a ball that are modeled with various resolutions of *MSM*. Springs that are weakened have a brighter color. The cube is modeled with mass network that is not aligned with its axes, which is a general case. As we can see, low resolution networks do not resemble the shape of modeled objects themselves, however as appriopriate masses and springs are weakened, the overall behavior is close to the one of the intended shape.

As will be shown in the next section, incorporating antialiasing technique to the algorithm of constructing MSMs improves physical properties of modeled objects considerably. There is however one problem with such representations. Some of the border elements may overlap with only very small percentage of modeled body, and will play a small role as an adjustment for the properties of border region (which is a main idea behind this technique). In fact they can be arbitrarily weak. Unfortunately presence of very light elements as well as very weak springs together with regular masses and strong springs in the same model is undesirable, as it presents great challenges for the motion integration algorithm. Moreover interactions with such objects can no longer be carried out on per node basis (for example picking), but would require to involve groups of nodes instead.

To avoid some of the problems concerning dynamics simulation of such models, a certain threshold should be imposed on the border elements, so that they are prevented from being arbitrarily weak. We have chosen in our tests 10% for overlapping volume to be this threshold. If any elemental cube is weaker than that it is simply discarded.

5.2 Tests

5.2.1 Elastic parameters

In our test we focus on simple *Cubic lattice MSMs*, which can model homogenous isotropic materials with Poisson's ratio $\nu = 0.25$ and arbitrarily chosen Young's

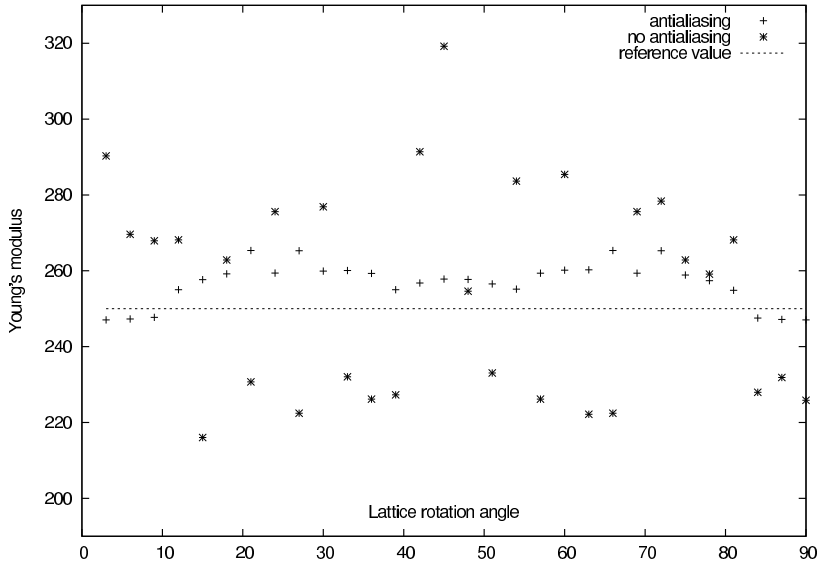

Figure 5.2: Reaction to an uniform compression of an elastic box for two models – with and without antialiasing. Measurement of Young's modulus.

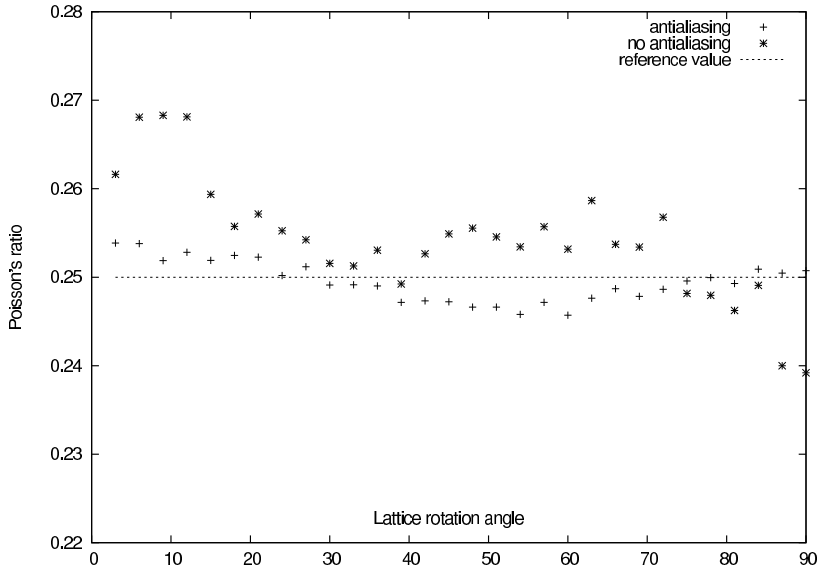

Figure 5.3: Reaction to an uniform compression of an elastic box for two models – with and without antialiasing. Measurement of Poisson's ratio.

modulus E. By compressing the object in one direction and measuring its response force as well as relative expansion in other directions we can verify values of E and ν and confirm whether modeled material is really the one which was intended. Big blocks with high lattice resolution give very good results even if their faces are not aligned with natural axes of the lattice (however there still is a problem of defining border and therefore volume of such models). However as the sample gets smaller (or the resolution lower), it becomes increasingly difficult to achieve intended behavior. Test case presented below shows, that using antialiased models allows to preserve global elastic properties of modeled objects with much lower resolutions.

In this test, we investigate how an elastic box of size 5x5x5 reacts to compression. The box is modeled with *cubic lattice MSM*, that is not aligned to its axes. The box is being compressed in Y direction to 95% of its length and the reaction force as well as X and Z expansion are measured. The test is carried out for 30 representations of the box, each of which is based on lattice rotated by a different angle. Intended value of Young's modulus of the material is 250, Poisson's ratio 0.25. We measure how the response of the box changes when it is modeled with a lattice rotated by different angle.

The results are shown on figures 5.2 and 5.3 for both not antialiased and antialiased models. We can see, that models with antialiasing respond to compression in more consistent way and they give values of E and ν closer to theoretical ones.

5.2.2 Internal force distribution

The second test we performed considers more subtle effects that may be caused by fine details of the model. In order to assess the quality of our MSM representation we have compared the internal stress distribution in our model with an analytical solution for a given solid mechanical problem. The analyzed test case is a problem of a spherical cavity in an elastic solid taken from A. F. Bower, in [3].

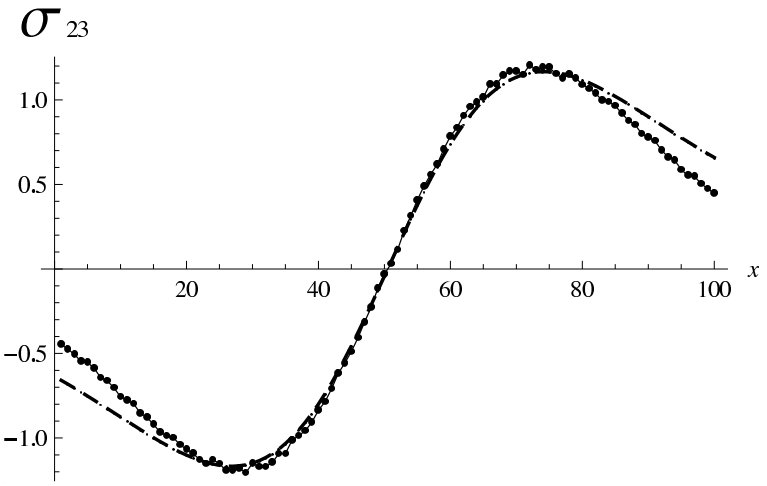

Figure 5.4: Spherical cavity in solid material. Stress tensor component σ_{23}, antialiasing enabled. Dotted line corresponds to theoretical values. Horizontal axis corresponds to the position within an object.

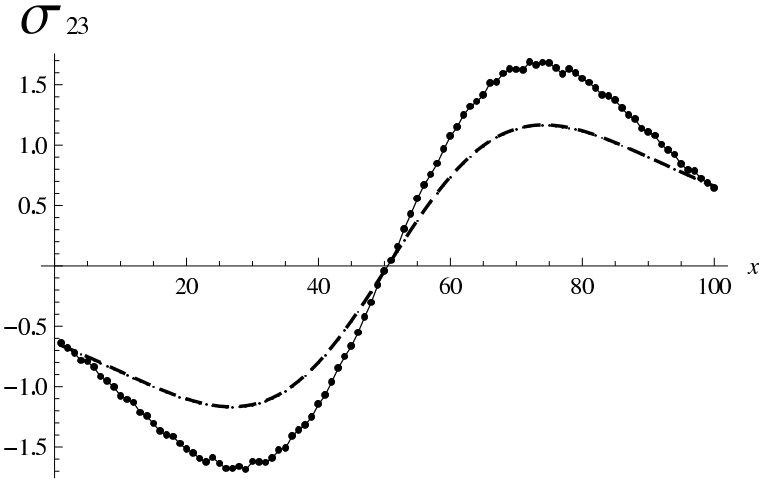

Figure 5.5: Spherical cavity in solid material. Stress tensor component σ_{23}, antialiasing disabled. Dotted line corresponds to theoretical values. Horizontal axis corresponds to the position within an object.

A large (infinite) elastic object has a spherical cavity in its center. Far from the cavity, the object is subjected to a tensile stress σ_{33} with all other stress components zero. The presence of the cavity will result in non trivial stress distribution inside the object. To verify whether our model properly responds to such situation, we will measure resulting stress tensor component σ_{23} near the cavity. The σ_{23} tensor component is given by:

$$\sigma_{23} = \frac{15\sigma_{33}a^3x_2x_3}{2(7-5\nu)R^5}(\frac{6}{5} + 5\nu - 7\frac{a^2}{R^2} + 2\frac{x_3^2}{R^2}), \tag{5.1}$$

where a is a radius of the cavity and $R = (x^2 + y^2 + z^2)^{\frac{1}{2}}$.

In this experiment we have modeled a block of material with $E = 250$ and $\nu = 0.25$ of dimensions 40x40x40 and a cavity with radius $a = 5$. We have tested both antialiased and not antialiased MSM for representing the cavity borders. In our setup, we imposed a stress σ_{33} high enough to cause the block to compress by about 10% of its length. The resulting distribution of stress tensor component, σ_{23} was measured along the line perpendicular to compression direction not far from the cavity.

Results are shown in Figures 5.4 and 5.5. We can see, that if antialiasing is not applied, measured stress differs from theoretical values considerably. This experiment shows, that it is possible to create models of custom objects by antialiasing that will give a good physical response to the deformation not only globally, but also on a local scale. We can expect such models to give a good feedback when deformed by a picking tool in virtual reality applications or in similar setup.

5.2.3 Complex shapes

As the final test case we investigate a deformation of a complex object, where the quality of an MSM can be judged with a human eye. We model the Stanford Bunny with MSM and apply a force to its left ear. As a result ear bends to a

certain degree (Fig. 5.6).

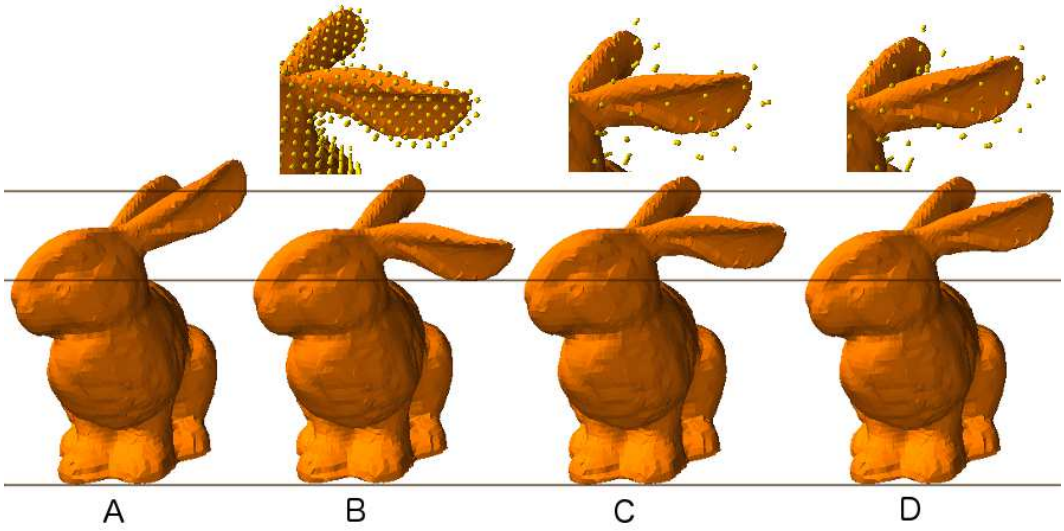

Figure 5.6: A force applied to rabbit's ear. Resolution $res = \frac{1}{a}$, where a is a spacing between lattice nodes. A – undeformed model, B – high resolution MSM $res = 5$, C – low resolution MSM with anti-aliasing $res = 2$, D – low resolution MSM without anti-aliasing $res = 2$.

The highest resolution representation is composed of 12205 MSM nodes and serves as a reference for the comparison between lower resolution models (Fig. 5.6 B). Medium and low resolution representations consist of 3024 and 1042 nodes respectively. Antialiased versions (AA) outperform their not antialiased (NOAA) counterparts decisively. Figures 5.6 C and D show the low resolution models. Models without antialiasing are more stiff and deform to a lower degree. It is worth to note, that even medium resolution NOAA performs worse than low resolution AA in this test (not shown on the picture).

Anti-aliased models pass simple compression tests such as the one presented in section 5.2.1 without any problems even for very low resolution representations (for example a cube 1.3 x 1.3 x 1.3 modeled by 8 mass points with spacing 1 gives

very precise force response). However as we can see in this test they still suffer some accuracy losses when bending. Overall quality of MSM strongly depends on the shape of the object it represents and it is difficult tell which resolutions can be sufficient in general case. Lowering the resolution allows for much faster computation, which is especially true for volumetric objects, however the deformations become less detailed until in the extreme case the whole object behaves like a homogenous elastic block.

In our current implementation one step of physical simulation of the bunny model takes 0.9ms for low, 2.5ms for medium and 10ms for high resolution MSM, however the high resolution model requires two times shorter time step for stable computation than models medium and low which additionally decreases speed of the animation. The timings were taken on a system with 4 core 2.4 GHz processor.

5.3 Conclusions

In this chapter we have analyzed a method of refining cubic lattice mass spring model representations of elastic objects. Masses and stiffness in an MSM are adjusted according to the fraction of the volume overlapping with reference shape. Step-like features that normally appear in cubic lattice MSMs can be greatly reduced by this technique, allowing to model non trivial shapes even with relatively low resolution latices, while preserving their physical properties. Improvement of the behavior of antialiased models is significant compared with the standard MSMs.

We have verified two types of tests. Both global response to large deformations and local force distribution inside the object significantly improved, and showed that discretization errors of the representation are reduced. With presented antialiasing technique it is possible to represent custom shapes in much more detail without the need of increasing the resolution.

Antialiased models are suitable both for the use in animations and usually more demanding virtual reality applications, because they allow to achieve proper local deformations as well as force feedback response for the deformation.

Chapter 6

Aspects of dynamics of mass spring models

In this chapter we will analyze dynamic properties of our models keeping special attention how resolution changes affect them. For all the MSMs analyzed in this chapter, the damping coefficients of the springs are scaled proportionally to the spring coefficients κ [27].

6.1 Introduction

As mentioned before, mass spring models are a popular choice for the representation of soft bodies in computer graphics and virtual reality applications. For a long time they were preferred over Finite element method (FEM) in real time applications, as they tend to offer better computational efficiency (although recent FEM implementations are quite fast as well [38]), but they are believed to be less accurate in terms of physical plausibility [31].

The accuracy of the description of an elastic object is not a problem in MSM representations, as we have seen in the previous chapters. The standard lattice based models used in physics, mechanical engineering and other related fields offer a description of elastic solids, which is as accurate as the limitations of linear elasticity theory allow it to be [32] [20] [18].

The bigger problem, when creating an animation, is to carry out the time evolution of such models (which can be achieved in a number of ways, both quasi-static or dynamic [10] [39] [25] [34] [24] [2]). The number of techniques used for this purpose in computer graphics is very large, but it should be noted, that most of them are not physically based or the physical correctness does not hold for certain aspects of the simulation.

In this work we use explicit Verlet integration scheme, in order to achieve a physically-based simulation and we show how to couple it with collision responses [17]. One of our goals is to design a collision handling technique which preserves the energy of the system (affects stability of the simulation), and if the friction is present, obeys the Newton's law of dry friction.

The disadvantage of explicit integrators is the necessity of using variable time step to avoid instabilities when large accelerations are present. They are however very useful for the purpose of studying certain aspects of the simulation and dynamic evolution of MSMs. For practical purposes, techniques which add some mechanism of controlling energy of the system are much more popular. The examples include position based or shape matching algorithms [2] [30].

6.2 Collisions

Defining a deformable object is only half of the solution needed for performing a realistic animation. Once we have the representation of an object, we have to simulate its evolution in time, which includes both evolution of its shape as well as interaction with the environment, that is – collisions with other objects.

6.2.1 Verlet integration scheme

The time integration algorithm which we use is constructed as a Verlet scheme following the Trotter expansion[19, 41, 10]:

$$e^{A+B} = \lim_{P\to\infty} (e^{A/2P} e^{B/P} e^{A/2P})^P, \tag{6.1}$$

where e^{A+B} is understood to be an operator which is advancing the state of the system by making P steps of "size" $1/P$. If P is finite, the equation is approximate. In an MSM simulation, the two qualities that characterize our system are positions q and velocities v (defined for each mass point), and the system advances in time (i.e. from 0 to t) with steps dt.

The Eq. (6.1) can be translated into the following algorithm:

1. advance positions by $0.5dt$; (q += $v \cdot 0.5dt$)

2. compute forces F

3. advance velocities by dt; (v += $F/m \cdot dt$)

4. advance positions by $0.5dt$,

which is known as a *position Verlet* integration scheme. It can be used to simulate the time evolution of a mass spring system. It does not account for collisions, as all the forces in the system come from springs or gravity. Consequently collision handling requires the algorithm to be extended.

6.2.2 Elastic collisions

In our approach the collision of an object represented by an MSM is carried out by colliding mass points of the MSM, that is, through microcollisions. Detection of collisions between MSM and meshes or objects with well defined surfaces is straightforward. Collisions between different MSMs as well as self collisions require defining the surface of an MSM in some way. In this work, we associate a collision sphere with each mass point lying on the border of an object; the radius of the sphere is equal to the average of the half-length of springs connected to the node in question. Additionally we assume that if two spheres are connected by a spring they never produce a collision.

Each microcollision should preserve energy and momentum of the system, that is for any two colliding mass points (spheres) m_1 and m_2 moving with velocities v_1 and v_2 we have:

$$\begin{aligned} m_1\bar{v}_1 + m_2\bar{v}_2 &= m_1\bar{v}_{1'} + m_2\bar{v}_{2'} \\ m_1\bar{v}_1^2 + m_2\bar{v}_2^2 &= m_1\bar{v}_{1'}^2 + m_2\bar{v}_{2'}^2, \end{aligned} \tag{6.2}$$

which gives new velocities as:

$$\begin{aligned} \bar{v}_{1'} &= \frac{2m_2\bar{v}_2 + (m_1 - m_2)\bar{v}_1}{m_1 + m_2} \\ \bar{v}_{2'} &= \frac{2m_1\bar{v}_1 + (m_2 - m_1)\bar{v}_2}{m_1 + m_2}. \end{aligned} \tag{6.3}$$

Non movable objects (such as a ground) are treated as having infinite mass.

A computationally efficient way of handling collisions is to process them in a serial way, as if none two happened at the exact same moment. However, the time resolution of our simulation is defined by the time step, and it is not desirable for it to be too small. This leads to multi-collisions, that is, situations in which an object is colliding with multiple other objects during one time frame. If we detect collisions once per frame, serial collision handling may lead to objects penetrating each other (i.e. imagine a sphere in between two planes with $\pi/6$ angle between them). Increasing the number of detection/handling evens per time frame is not a solution, as it may lead to significantly increased computational costs per frame in some situations. The solution may be to abandon serial collision handling, and instead solve all the collisions simultaneously (e.g. as a constraint reaction problem). The disadvantage of such technique is again the increased computational cost.

Below we will show how to prevent object penetration while processing collisions in a serial way.

First of all, before we proceed to multi collision problem, let us show how to achieve a stable simulation for a simple setup of a ball bouncing on the ground in gravitational field. Our time integration algorithm consists of two conceptual parts – position update and velocity update. Collision reaction can be added as a third part, and by applying eq. (6.1) recursively, we arrive at one of possible algorithms:

1. advance positions by $0.5dt$
2. compute forces
3. advance velocities by $0.5dt$
4. handle collisions and friction by updating velocities (not forces)
5. advance velocities by $0.5dt$
6. advance positions by $0.5dt$

This scheme will produce energy preserving simulations which are stable over long time scales, and will work if the bouncing mass is receiving forces from springs as well. First we advance positions (half step), then velocities, which by itself consists of two steps – applying force influence and collisions. Note, that collision handling procedure uses the values of velocities in the middle of the frame (at the half step). It is a popular approach of incorporating collision response into the simulation used e.g. in [4].

This algorithm, if collision handling is processed in a serial way, may produce object overlaps if a single mass point of the MSM is colliding with multiple independent objects. It should be noted, that for volumetric MSMs this is unlikely to happen unless the object's curvature is high. Consequently this algorithm could be used for volumetric MSMs in low curvature environments. Unfortunately the behavior of clothes would be affected considerably, as 2d objects stacked together produce exactly this kind of multi-direction multi-collisions, which cause overlaps.

6.2.3 Elastic collisions with overlap prevention

For the purpose of preventing object overlaps, we modify the algorithm in the following way:

1. advance positions by $0.5dt$
2. apply half of the collision/friction response
3. compute forces
4. advance velocities by dt
5. compute collision/friction response
6. apply half of the collision/friction response
7. advance positions by $0.5dt$

The collision response was switched with velocity update when compared to the previous algorithm. The second step uses collision/friction response information from the "previous" frame (although we may very well say that each frame starts at the step 4, and the whole procedure will look more natural). Steps 2 and 6 (apply c/f response) modify velocities.

In a perfectly elastic collision of two hard objects (eq. (6.3)), the state in which half of the response is applied results with a situation where normal component of the relative velocity between these two colliding objects equals zero. Therefore steps 7 and 1 are guaranteed not to produce any additional overlap between them. This means, that if an MSM node collides with multiple elements of static environment (i.e. triangles of an unmovable mesh), the node will remain at rest state relative to these static elements (in terms of normal velocities).

If there are additional MSM nodes colliding with each other they too, can be brought together to the state of the same relative velocity, however it may require applying collision response to them, with different proportions than those present in steps 2 and 6 (i.e. instead half-half, we may need to do e.g. 0.3 and 0.7 response). In such situations energy preservation is no longer guaranteed (with

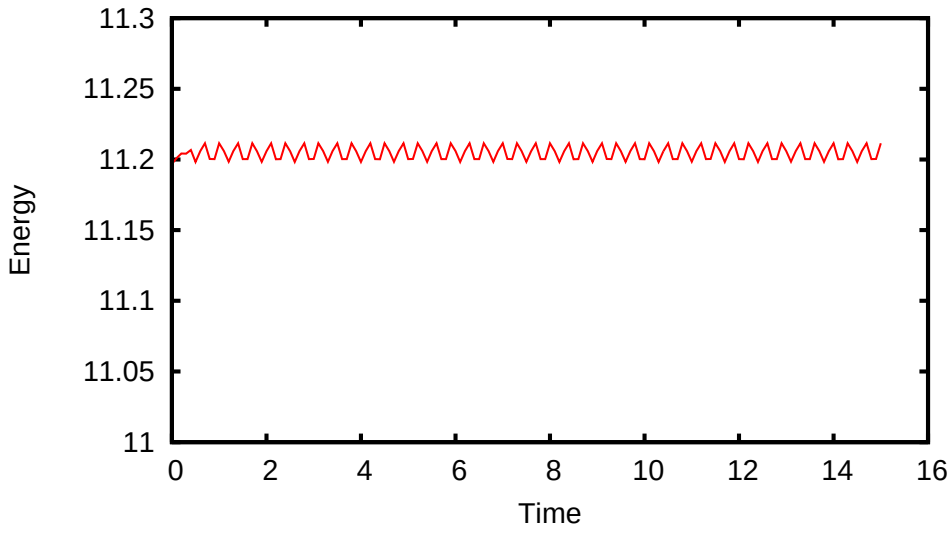

Figure 6.1: Energy of the system in a typical multi collision situation.

energy differences of the same magnitude as those present e.g. in Verlet with adaptive time-step).

The effects of this can be seen in Figs. 6.1 and 6.2. The Figure 6.1 shows an energy of a system, where 3 MSM nodes lie on top of each other (and on the ground). The Figure 6.2 shows an energy of the same system, but with the top two nodes are connected by a spring. In both cases the bottom node undergoes a constant collision with a ground and the node above. There is a noticeable energy drift in the system if spring forces are present. Its effects can be eliminated by incorporating friction and damping effects.

6.3 Contact friction

The problem of collisions with frictional contact has been studied previously for of various model representations and time integration schemes [9] [13] [5] [15] [8] [33], and the common approaches include, among others, penalty and constraint based methods. In the context of mass spring models, the focus has been placed on 2d meshes designed for a cloth representation [4] [33]. While such methods have a good potential to be extended to volumetric objects, the existing works lack

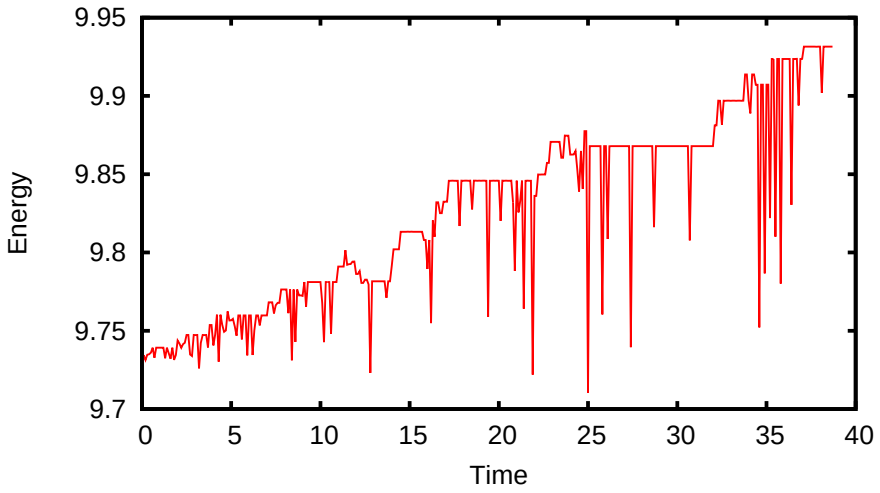

Figure 6.2: Energy drift present in a multi collision setup, when multiple objects are stacked on each other and spring connections are present.

argumentation and convincing testing schemes that would verify their physical correctness.

The Newton dry friction model assumes that:

- Friction force is proportional to the normal pressure between contacting objects and is parallel to the contact surface
- It does not depend on the area of contact between objects
- It does not depend on the velocity of sliding surfaces

Let us consider a volumetric object sliding on a flat surface with friction. The friction force will be proportional to the gravitational force acting on the object:

$$F = \mu F_g = \mu \cdot M \cdot g, \tag{6.4}$$

where μ is the friction coefficient (which in general will depend on both colliding surfaces), M is the mass of the moving object and g denotes gravitational acceleration coefficient.

In rigid body dynamics, the mass of the whole object has to be known for the friction force to be calculated correctly. In our case however all the interactions between the surface and the object happen through a handful of boundary mass points that collide with the surface and the frictional force should be applied only to those points. For the sake of efficiency the friction handling algorithm should not require the knowledge about any other nodes of the colliding MSM or MSM stacked on top of it. Consequently the equation 6.4 is not really usable. Instead its properties should be recreated by simple rules for single mass point collisions.

Fortunately these rules are very easy to find and understand. Let us consider a box sliding on a flat surface. We will represent it by MSMs with three different resolutions (Fig. 6.3). The first box – $3 \times 2 \times 2$ mass points, the second by $5 \times 3 \times 3$ lattice and the third – $9 \times 5 \times 5$. The total mass of each box is the same, therefore elementary masses of the MSM nodes will differ. Also the number of the nodes colliding with the surface will be different: 6 mass points for the low resolution MSM, 15 for medium and 45 for high. This constitutes 50%, 33% and 20% of the total number of nodes for each model respectively. Consequently the number of elementary collisions will differ greatly between these models, and each elementary collision will result in a different elementary momentum flow between object and the surface. Moreover for each point the frequency of collision events will differ. What will be invariant between high, medium and low resolution representations however, is the total flow of momentum that passes from the object through the colliding surface.

If we consider, that each mass point m_i represents some elemental volume and consequently elemental area of contact a_i, bounces of the surface with average frequency f_i, changing is momentum by Δp_i with each bounce (according to eq. (6.3)), giving a flux $j_i = \frac{\Delta p_i f_i}{a_i}$, the total momentum flow must sum up to Mg, because it is the condition of objects not penetrating each other:

$$j \cdot A = \sum_{i}^{C} j_i a_i = \sum_{i}^{C} f_i \Delta p_i = \frac{\Delta p}{\Delta t} = Mg. \tag{6.5}$$

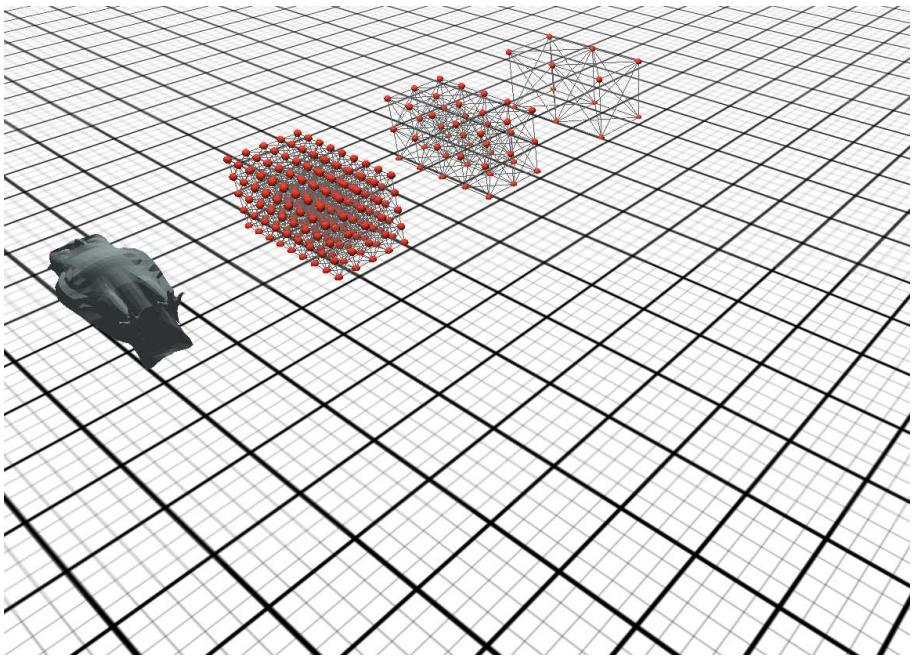

Figure 6.3: Four elastic objects sliding with friction on a flat surface. Initial position. Three resolutions of a box (high, medium, low), and a bird. Boxes have the same mass. Bird has different mass and non flat surface.

The only quantity that we control directly when giving a response to an elementary collision is Δp_i, however we know, that all the other quantities will adjust themselves in such way, that the eq. (6.5) will hold (i.e. if we decrease Δp_i, the frequency f_i will increase so that the momentum flow will remain constant). Therefore if we associate with each collision an elementary friction impulse $F_i = \mu \Delta p_i$, the total frictional force exerted on an object will accumulate to $F = \mu M g$, independently on quantities such as A, f_i and even Δp_i. The latter means that incorporating non elastic collisions will not affect frictional force during continuous contact of surfaces.

6.4 Tests

In order to confirm that such micro collision approach to friction allows to achieve a correct macroscopic response we have performed a series of experiments in which elastic objects slide with friction on a flat surface. According to Newton's dry

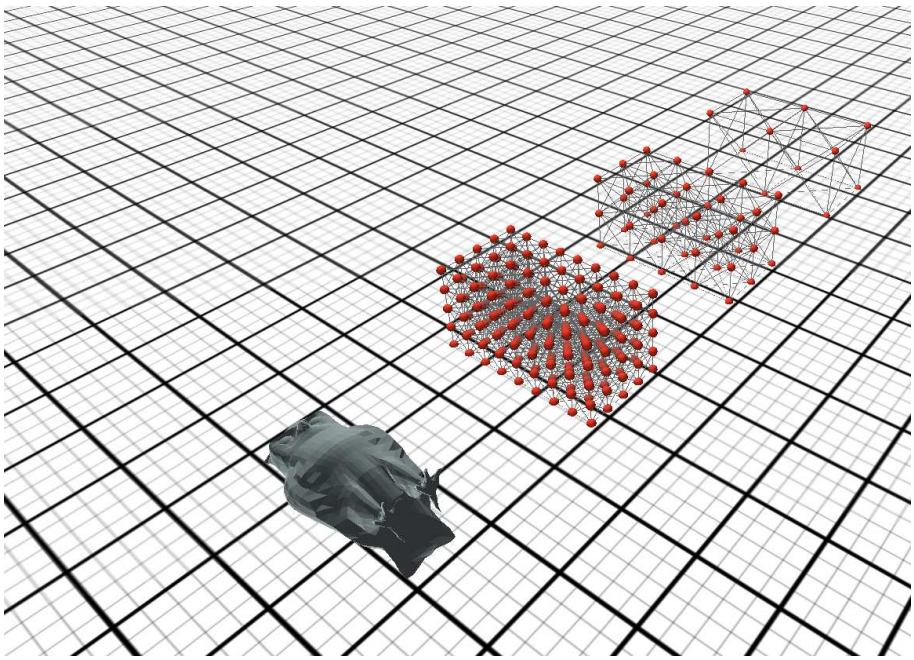

Figure 6.4: Final position of the objects from Fig. 6.3.

friction law the distance which each object travels before completely stopping should be independent on the mass of the object, its shape or apparent area of contact. We have performed the test for various shapes – from simple boxes to complex objects with non trivial shapes. Our tests indicate that travel distance does not depend on the resolution of MSM, mass of the object or even its shape, thus confirming that our method gives physically correct results.

Figures 6.3 and 6.4 show the initial and final positions of four elastic objects.

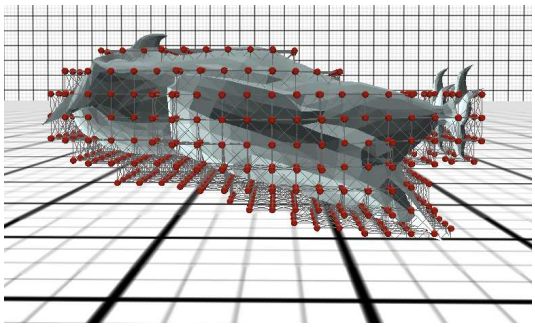

Figure 6.5: Initial position of the model of a bird.

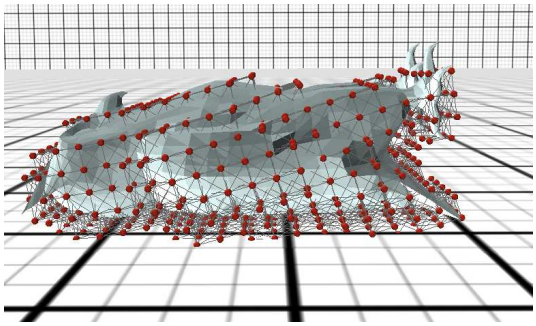

Figure 6.6: Final position of the model of a bird.

The three boxes represented by different resolution MSMs travel exactly the same distance before stopping, which shows that energy dissipation is rate is invariant with the resolution and depends only on the friction coefficient. The fourth object – a bird with non trivial shape and different mass, slides beside the boxes. At the beginning only its tail touches the ground (Fig. 6.5); soon due to gravitational force the rest of its body descends and the area of contact with the ground is increased. Even though the bird's movement is complex and includes shape relaxation with various vibrational modes, the total distance that it travels is the same as in the case of boxes.

Next we further test consistency of the behavior of MSMs with two additional scenarios:

6.4.1 Scenario 1 – energy dissipation

In the first scenario a deformable spherical object with dense core and soft shell is thrown horizontally just above a flat surface. It falls down and after a small bounce starts rolling until it stops due to energy dissipation that occurs mostly in the soft shell (Fig. 6.7 and 6.8). We have measured and compared the traveling distance for three different representations of such object with different resolutions (Level of details, LODs). The distance is very volatile to the properties of the shell, as the overall movement is quite long, and all the differences in the behavior are accumulating with a snowball effect. The results are presented in the Table 6.1. It also contains information about the number of mass points and springs used for the object representation. As we can see the behavior of

low resolution models does not diverge significantly from the high resolution one. Visual quality of the animation is satisfactory for all generated MSMs, and the most simplified one had about 18% of the original hi-res number of nodes and springs.

Figure 6.7: Energy dissipation test. Left: object falls on the ground. Right: object is rolling.

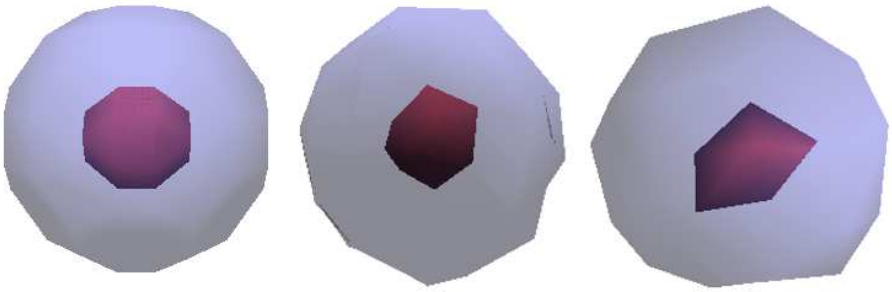

Figure 6.8: Dense core with soft shell. From the left: *cubic lattice MSM* model, $N = 281$, $S = 4094$, disordered MSM LOD-1: $N = 114$, $S = 1596$, LOD-2: $N = 51$, $S = 670$.

model	distance	difference	nodes	springs
Base model	6.1	0	281	4094
LOD-1	5.6	0.5	114	1596
LOD-2	6	0.1	51	670

Table 6.1: The distance a spherical object travels before it stops in the scenario 1.

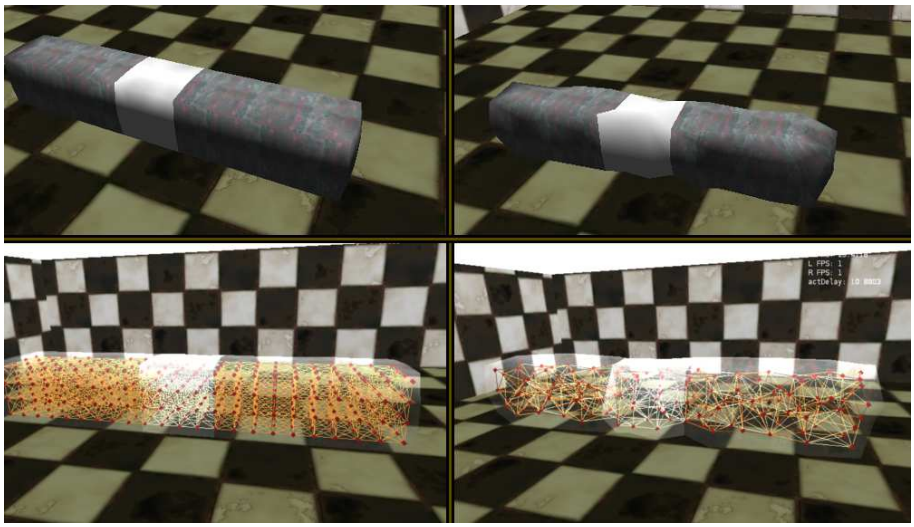

Figure 6.9: Two stiff bars connected by rubber (solid and wireframe). On the left *cubic MSM*, number of points $N = 368$, and springs $S = 4724$. On the right the simplified representation of the same object, disordered MSM $N = 154$, $S = 1746$.

6.4.2 Scenario 2 – complex movement

In the second test case, an object composed of two very stiff rods connected by rubber is being thrown at the wall, while rotating (Fig. 6.9 and Fig. 6.10). It hits the wall, bounces off, falls to the ground possibly still rotating until it finally stops due to friction. The exact movement is quite complex and the trajectory depends heavily on initial velocity of the object.

Despite such complex movement, the behavior of the two different MSM models is very similar.

6.5 Conclusions

In this chapter we have shown tests of general behavior of MSMs that involve not only static deformations, but dynamics as well. We have shown how to incorporate a collision response into Verlet integration scheme. The presented technique is very robust and will work satisfactory for various conditions and situations. Frictional force acting on an object does not change even if the area of contact or the resolution of an MSM differs. Because the response to each micro collision is

Figure 6.10: Animation for the test scenario 2. An object composed of two stiff bars connected by rubber rotates and hits the wall. Comparison between an cubic MSM $N = 368$, $S = 4724$ (left) and disordered model $N = 154$, $S = 1746$ (right).

local, the presented algorithm will also work in case of multiple objects stacked on each other. Lastly, the presented technique has a very high computational efficiency, as it does not require any knowledge about global properties of colliding objects.

Moreover, we show that total frictional force does not depend on the single microcollision response, but all the responses summed together. This means, that careful analysis of single microcollisions is unnecessary and does not affect the final result of the total frictional force. Such analysis is often present in collision responses in CG, and require e.g. measuring penetration depth [14], which greatly increases costs of collision response.

Dividing the collision response into two steps allows to prevent progressing penetration of colliding objects in serial collision handling. This approach however has a limitation which manifests itself when groups of MSM nodes collide with each other. In such cases, the energy of the system may change slightly during the collision response, which may lead to significant energy drifts over long periods of time. It has to be noted however, that for volumetric MSM such situations occur very rarely, and only for the models with high surface curvature. For 2d MSMs (clothes) on the other hand, such multi-collisions occur frequently when a cloth is folded, or when multiple layers of material are stacked together. In this case however, spring connections parallel to the collision direction, which are causing the energy drifts in case of 3d models, are not present. Consequently, energy drifts occur rarely, and in typical situations when friction and damping are present, they do not pose any problem, as the energy dissipation due to these phenomena is of much higher magnitude, than the one caused by numerical inaccuracies.

The proposed method for friction handling, coupled with collision response yields good results. Complex motion of non trivial object can be simulated consistently for various resolutions of MSM. In our test schemes, objects simplified by as much as 50% exhibit very small differences from the original behavior. Further simplification leads to the changes in elastic properties that are noticeable, however the behavior of such objects is still plausible.

Chapter 7

Conclusions and final remarks

This thesis investigates properties of Mass Spring Models for the purpose of the simulation of deformable objects for Computer Graphics. The underlying theory is closely related to that of the theory of elasticity for solids or fluids, however in contrast to physical applications, computer graphics is less restrictive and very often fake realism is sufficient. Nevertheless physically based models are still important for various applications; most notably Virtual Reality simulators with interaction feedback.

We have explored the limitations of simple (ordinary) mass spring models and formulated a methodology for constructing the mass spring models with well-defined physical properties. In particular, we emphasized the importance of isotropy of the models and the limitations it places on the geometry of the network. Next, we have discussed the classical result of the theory of elasticity that while any value of the Young's modulus can be obtained by tuning the spring coefficients in the network, the value of the Poisson's ratio is constrained to be 1/4 (in 3D). Finally, we have derived a simple formula, Eq. (2.24), linking the Young's modulus with the mean value of kL^2 throughout the network, where k and L are the spring stiffness coefficient and its length, respectively. This is a relatively simple and yet general formula which can be applied to virtually any MSM and does not require a complicated analysis of the model.

We have analyzed two approaches of creating such MSM. First one assumes mass point distribution on a cubic lattice. Second approach is based on random node placement.

The tests in which we measure the elastic properties of both cubic and random lattices show that our models agree satisfactory with theoretical predictions and thus are well suited for the modeling of elastic materials. The random MSMs seem to perform almost as good as the cubic ones when the global deformations of the system are concerned, but slightly underperform at the smaller scales (of the order of few internode distances). Even though the difference is not large, it does show that MSMs, particularly low resolution ones, require attention during generation process in order to minimize discretization errors. Overall, however, the performance of random models is good enough for them to be an attractive alternative to the cubic models, as in many situations a slight decrease of accuracy may be an affordable price to pay for the flexibility of choosing node positions.

Simple Mass Spring Models are easy to implement and provide good approximations of linear elasticity theory. In practical applications they suffer however form some problems imposed on them by the limitations of the linear elasticity – they are expected to give a plausible elastic response only within its limits, that is for small relative displacements. Unfortunately in typical CG application displacement may become very large and thus some mechanism is needed to properly handle such situations. We have not explored this area thoroughly, however the extensions we offer in chapter 4, greatly improve the behavior of our models. We also present an extension, that allows to achieve arbitrary values of Poisson's ratio, not known previously in this field.

Additionally the antialiasing method proposed in chapter 5 allow for refining mass spring model representations of elastic objects. In our approach, an elastic body is represented as a mass spring network constructed with cubical elastic elements, whose mass and stiffness are adjusted according to the fraction of the volume overlapping with reference shape. Step-like features that normally appear in cubic lattice MSMs can be greatly reduced by this technique, allowing to model non trivial shapes even with relatively low resolution latices, while preserving their physical properties. Improvement of the behavior of antialiased models

is significant compared with the standard MSMs.

Mass Spring Models presented in this work, can be treated as a means of representing elastic objects. They provide a correct mapping of elastic properties between a physical object and a model which is a prerequisite for any physically correct simulation, be it quasi-static or dynamic. The examples of the simulations with explicit time integration technique shows, that these models behave correctly under various circumstances and that their properties do not change drastically with the change of resolution. We were able to achieve a consistent collision and friction responses for modeled objects. The dynamical aspect of the simulation is however a very broad topic, and many of its aspects have not been discussed. A future work on this topic may include more careful study of the evolution of shape changes of the models during the simulation. Currently available simulation techniques produce results in which the motion of an object in general depends on the resolution of the MSM. This is especially true for vibrating objects, whose frequency differs for different resolution of MSMs.

Future direction of the implementation of our system, may include the adaptation of presented models for the work with parallel computation units such as GPU. Because their properties are very localized, MSMs have a great potential for parallelization.

Bibliography

[1] Vincent Baudet, Michael Beuve, Fabrice Jaillet, Behzad Shariat, and Florence Zara. Integrating Tensile Parameters in 3D Mass-Spring System. Technical Report RR-LIRIS-2007-004, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumiére Lyon 2/École Centrale de Lyon, February 2007. 2, 10, 14, 42

[2] Jan Bender, Matthias Müller, Miguel A. Otaduy, and Matthias Teschner. Position-based methods for the simulation of solid objects in computer graphics. In *EUROGRAPHICS 2013 State of the Art Reports*. Eurographics Association, 2013. 56, 68

[3] A. F. Bower. *Applied Mechanics of Solids*. CRC Press, Taylor & Francis, Boca Raton, 2009. 29, 61

[4] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3):594–603, July 2002. 71, 73

[5] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. Efficient Linear Elastic Models of Soft Tissues for Real-time Surgery Simulation. Technical Report RR-3510, INRIA, Oct 1998. 73

[6] Franklin C. Crow. The aliasing problem in computer-generated shaded images. *Commun. ACM*, 20(11):799–805, November 1977. 57

[7] Herve Delingette. Triangular springs for modeling nonlinear membranes. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):329–341, 2008. 2, 11, 42

[8] Christian Duriez, Student Member, Frdric Dubois, Abderrahmane Kheddar, and Claude Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 12:36–47, 2006. 73

[9] Susan Fisher and Ming C. Lin. Deformed distance fields for simulation of non-penetrating flexible bodies. In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation*, pages 99–111, New York, NY, USA, 2001. Springer-Verlag New York, Inc. 73

[10] D. Frenkel. *Understanding Molecular Simulation - From Algorithms to Applications*. Academic Press, 2002. 68

[11] Sarah F. F. Gibson and Brian Mirtich. A survey of deformable modeling in computer graphics. Technical report, Mitsubishi Electric Research Laboratories, Cambridge, MA, 1997. 1

[12] R. J. Hardy. Formulas for determining local properties in molecular-dynamics simulations - Shock waves. *J. Chem. Phys.*, 76:622–628, January 1982. 24, 31

[13] Shoichi Hasegawa and Makoto Sato. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. *Comput. Graph. Forum*, 23(3):529–538, 2004. 73

[14] Bruno Heidelberger, Matthias Teschner, Richard Keiser, Matthias Müller, and Markus Gross. Consistent penetration depth estimation for deformable collision response. In *In Proceedings of Vision, Modeling, Visualization VMV'04*, pages 339–346, 2004. 82

[15] Doug L. James and Dinesh K. Pai. Real time simulation of multizone elastokinematic models. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 927–932, 2002. 73

[16] Maciej Kot and Hiroshi Nagahashi. Anti-aliasing for cubic lattice mass spring models. In *Proceedings of NICOGRAPH International*, 2013. 57

[17] Maciej Kot and Hiroshi Nagahashi. Collision response in mass spring model simulations. *Technical report of IEICE. Multimedia and virtual environment*, 113(470):287–290, 03 2014. 68

[18] Maciej Kot, Hiroshi Nagahashi, and Piotr Szymczak. Verification of physical properties of materials modeled with mass-spring systems. *Technical report of IEICE. Multimedia and virtual environment*, 110(457):201–206, February 2011. 23, 67

[19] A. J. C. Ladd. *Numerical methods for molecular and continuum dynamics*. 3rd Warsaw School of Statistical Physics, B. Cichocki, M. Napiorkowski, J. Piasecki, eds., Warsaw Univesity Press, Warsaw, 2010. 68

[20] Anthony J. C. Ladd and John H. Kinney. Elastic constants of cellular structures. *Physica A: Statistical and Theoretical Physics*, 240(1-2):349–360, June 1997. 2, 3, 10, 12, 67

[21] Sarang Lakare and Arie Kaufmany. Anti-aliased volume extraction. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pages 113–122, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 57

[22] R. S. Lakes. Deformation mechanisms in negative Poisson's ratio materials: structural aspects. *Journal of Materials Science*, 26:2287–2292, 1991. 2, 10, 42

[23] L. D. Landau and E. M. Lifshitz. *Theory of elasticity*. Pergamon, London, 1959. 5, 17, 29, 34

[24] Joshua A. Levine, Adam W. Bargteil, Christopher Corsi, Jerry Tessendorf, and Robert Geist. A peridynamic perspective on spring-mass fracture. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Jul 2014. 68

[25] Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. Fast simulation of mass-spring systems. *ACM Transactions on Graphics*,

32(6):209:1–7, November 2013. Proceedings of ACM SIGGRAPH Asia 2013, Hong Kong. 68

[26] B. A. Lloyd, G. Szekely, and M. Harders. Identification of spring parameters for deformable object simulation. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1081–1094, September 2007. 2, 10, 11, 14, 15, 18

[27] Bryn A. Lloyd, Sercan Kirac, Gbor Szkely, and Matthias Harders. Identification of dynamic mass spring parameters for deformable body simulation. In Katerina Mania and Erik Reinhard, editors, *Eurographics 2008 - Short Papers*, pages 131–134, 2008. 67

[28] A. E. H. Love. *A treatise on the mathematical theory of elasticity*. Cambridge University Press, 1906. 2, 5, 9, 43, 44

[29] U. Meier, O. López, C. Monserrat, M. C. Juan, and M. Alcañiz. Real-time deformable models for surgery simulation: a survey. *Comput. Methods Prog. Biomed.*, 77(3):183–197, March 2005. 1, 2

[30] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24:471–478, July 2005. 68

[31] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, Mark Carlson, and Novodex Ageia. Physically based deformable models in computer graphics. *Comput. Graph. Forum*, 25(4):809–836, 2006. 1, 2, 67

[32] Martin Ostoja-Starzewski. Lattice models in micromechanics. *Applied Mechanics Reviews*, 55(1):35–60, 2002. 1, 10, 11, 67

[33] Simon Pabst, Bernhard Thomaszewski, and Wolfgang Strasser. Anisotropic friction for deformable surfaces and solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 149–154, New York, NY, USA, 2009. ACM. 73

[34] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. 5, 68

[35] J. S. Rowlinson. *Cohesion: A Scientific History of Intermolecular Forces*. Cambridge University Press, 2005. 44

[36] M. Sahimi. *Rigidity and Elastic Properties: The Discrete Approach. Heterogeneous Materials: Linear Transport and Optical Properties. Interdisciplinary Applied Mathematics v 1, (Chapter 8)*. Springer-Verlag New York, 2003. 42

[37] Gaizka San-Vicente, Iker Aguinaga, and Juan Tomas Celigueta. Cubical mass-spring model design based on a tensile deformation test and nonlinear material model. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):228–241, February 2012. 1, 2, 12, 14

[38] Funshing Sin, D. Schroeder, and Jernej Barbic. Vega: Non-linear fem deformable object simulator. *Comput. Graph. Forum*, 32(1):36–48, 2013. 67

[39] M. O. Steinhauser. *Computational Multiscale Modeling of Fluids and solids*. Springer-Verlag Berlin Heidelberg, 2008. 68

[40] S. Torquato. *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer-Verlag, New York, 2002. 15

[41] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *The Journal of Chemical Physics*, 97(3):1990–2001, 1992. 68

[42] Allen Van Gelder. Approximate simulation of elastic membranes by triangulated spring meshes. *J. Graph. Tools*, 3(2):21–42, February 1998. 2, 10, 11

[43] Gaizka San Vicente, Carlos Buchart, Die1go Borro, and Juan Tomas Celigüeta. Maxillofacial surgery simulation using a mass-spring model derived from continuum and the scaled displacement method. *Int. J. Computer Assisted Radiology and Surgery*, 4(1):89–98, 2009. 2

[44] Gao-Feng Zhao, Jiannong Fang, and Jian Zhao. A 3d distinct lattice spring model for elasticity and dynamic failure. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(8):859–885, 2011. 42

[45] J A Zimmerman, E B WebbIII, J J Hoyt, R E Jones, P A Klein, and D J Bammann. Calculation of stress in atomistic simulation. *Modelling and Simulation in Materials Science and Engineering*, 12(4):S319, 2004. 24, 31

Publications

Journal Articles

1. Maciej Kot, Hiroshi Nagahashi, Krzysztof Gracki: *Resolution scaling for mass spring model simulations*, IEICE Transactions on Information and Systems Vol.E97-D, No.8, pp. 2138-2146

2. Maciej Kot, Hiroshi Nagahashi, Piotr Szymczak: *Elastic moduli of simple mass spring models*, Accepted to The Visual Computer

International Conferences

1. Maciej Kot, Hiroshi Nagahashi, Krzysztof Gracki: *Simplification of Mass Spring Models*, WORLDCOMP CGVR12 International Conference on Computer Graphics and Virtual Reality (July 16-19, 2012, USA).

2. Maciej Kot, Hiroshi Nagahashi *Anti-aliasing for Cubic Lattice Mass Spring Models*, NICOGRAPH International 2013

Domestic Conferences

1. Maciej Kot, Hiroshi Nagahashi, Piotr Szymczak: *Verification of physical properties of materials modeled with Mass-Spring Systems*, IEICE Tech. Rep. Multimedia and virtual environment 110(457), 201-206, 2011.

2. Maciej Kot, Hiroshi Nagahashi: *Collision response in Mass Spring Model simulations*, IEICE Tech. Rep., vol. 113, no. 470, MVE2013-120, pp. 287-290, March 2014.