

# Using Simulated Annealing to Obtain Good Nodal Approximations of Deformable Bodies

Oliver Deussen and Leif Kobbelt and Peter Tücke

Institute for Operating and Dialog Systems, University of Karlsruhe  
76128 Karlsruhe, Germany

Net: [oliver | kobbelt | tuecke]@informatik.uni-karlsruhe.de

WWW: <http://i31www.informatik.uni-karlsruhe.de/~oliver|~kobbelt|~tuecke>

**Abstract.** In this paper we present a method to obtain good approximations of deformable bodies with spring/mass systems. An iterative algorithm based on voronoi diagrams is used to get a good mass distribution. The elastic properties of the system are optimized by simulated annealing. Results are shown, and some applications are discussed.

**Keywords:**

Simulation, Spring/mass lattice, Modeling, Deformable Bodies, Computer Graphics

Real time simulation and interactive control of deformable bodies are of great interest to computer graphics and other fields like pattern matching or medical visualization.

Spring/mass (or nodal) systems are often used to obtain efficient simulations of deformable objects if lower requirements for the simulation precision are given.

Terzopoulos and Waters [15, 13] used them for modeling and simulating facial tissue. Simulation of woven clothes was done by Breen [4]. In an early work, Miller [11] used nodal models to simulate utricular objects.

In medical research similar systems are used to simulate human organs for training tasks in endoscopic surgery [10, 6], and for general surgery simulation and planning [5].

To reproduce a specific mechanic behaviour of a spring/mass system, parameters like spring constants can either be preset, or optimized iteratively. We show that within some general limits, such optimizations yield systems which approximate objects with given elastic properties quite well.

## 1 Nodal Approximations

In general, a spring/mass system (SMS) is a collection of points (or particles) with a specific mass, that are connected by springs, dampers or plastic elements. The discrete Lagrangian equation of motion for a point  $p_i$  at position  $\mathbf{u}_i$  and mass  $m_i$  has the simple

form (cf. [15]):

$$m_i \ddot{\mathbf{u}}_i + \gamma \dot{\mathbf{u}}_i = \mathbf{F}^e - \sum_{j \in N_i} \mathbf{F}_{ij}, \quad (1)$$

where  $\gamma$  is a general damping coefficient, and  $\mathbf{F}^e$  denotes an external force acting on the point. The forces  $\mathbf{F}_{ij}$  are exerted by elastic, viscotic and plastic connectors between  $p_i$  and neighbour points  $p_j$  ( $N_i$ : set of neighbours).

### 1.1 Finite Element Formulation of Nodal Systems

The idea of approximating a deformable body with a system of springs is surprisingly old. In 1868, Kirsch [9] derived the fundamental equations of elasticity from a system of springs approximating a small cube. His work was one of the bases of the finite element method (FEM).

Today pin jointed trusses (assemblages of axially loaded elastic bar elements) are used in structural mechanics as idealizations for many problems. The FEM approach is widely used for such systems.

Those trusses can be seen as spring/mass systems. The only difference is that within trusses bars have a mass and joints are massless whereby in a SMS massless springs are connected by mass points that behave like spherical joints. In both cases, the mass distribution can be expressed by a mass matrix  $\mathbf{M}$ .

If only elastic and viscotic connectors are used, equation (1) can be written in matrix calculus (cf. [2]) by:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} = \mathbf{F}^e - \mathbf{K}\mathbf{u}. \quad (2)$$

In this equation,  $\mathbf{u} = (u_{1x}, u_{1y}, u_{1z}, \dots, u_{nx}, u_{ny}, u_{nz})^T$  is the vector of all point positions<sup>1</sup>,  $\mathbf{D}$  is the damping matrix and  $\mathbf{F}^e$  denotes the external forces acting on the nodes.  $\mathbf{K}$  is the systems stiffness matrix. This matrix describes the forces on nodes as a result of deformations and can be seen as a generalized spring constant.

In comparison to other systems of finite elements, nodal systems do have the advantage of  $\mathbf{M}$  being diagonal, which is simple and convenient for mathematical handling (cf. [8]). In addition, equation (2) can be solved by local application of equation (1) to each node. This leads to more efficient algorithms (cf. [15]), especially if parallel computers are used.

Table 1 shows some computation rates (calculations per second). The local simulation method based on equation (1) is independent of the bandwidth of  $\mathbf{K}$ . FEM methods based on equation (2) do have low computation rates if  $\mathbf{K}$  is recalculated in every step. This has to be done to make the FEM method comparable with the local algorithm. If  $\mathbf{K}$  is updated every eighth step, the computation rates for systems with a small bandwidth are close to that of the local method.

For small deformations (the geometric linear case)  $\mathbf{K}$  can be assumed constant. In this case the computation rates are high, but simulation precision decreases.

The calculation of equilibrium states is computational expensive because a linear equation system based on  $\mathbf{K}$  has to be solved.

---

<sup>1</sup>More precisely  $\mathbf{u}$  describes the displacements of points to an initial position with zero inner energy of the system (indicated by  $|\mathbf{K}\mathbf{u}| = 0$ ).

Table 1: Computation rates (calculations per second, SUN Sparc 20)

Bandwidth of $\mathbf{K}$	49	37	25	21
1) Local integration	37	37	37	37
2) FEM ( $\mathbf{K}$ updated every step)	8	9	10	10
3) FEM ( $\mathbf{K}$ updated every eighth step)	19	23	31	33
4) FEM ( $\mathbf{K}$ constant)	25	32	43	48
5) Equilibrium calculation	2.5	4	7	9

Investigated was a system with 84 nodes, 231 edges and different bandwidth of  $\mathbf{K}$ . 1) computation with local method; 2) FEM solution with updating of matrix  $\mathbf{K}$  in every step; 3) FEM solution,  $\mathbf{K}$  is updated every eighth step; 4) FEM solution without updating; 5) computation of equilibrium.

As a consequence, we use a finite element method based on band matrices to obtain equilibrium states within the optimization process (in section 5.3) and local methods for the efficient solution of equation (2) during simulation.

## 2 Optimal Elastic Systems

Given a deformable body  $B$  with mass density  $\rho(\mathbf{u})$ , surface  $A$  and elastic properties represented by a function  $\mathbf{G}$  (cf. [7]). This function determines the stress-strain relation

$$\sigma = \mathbf{G}(\epsilon), \tag{3}$$

and thus the forces needed to deform a body in a certain way (see section 5.1).

In the most general case, we have  $\mathbf{G} \in \mathbb{R}^9 \times \mathbb{R}^6$ . For linear problems and elastic homogeneous and isotropic materials,  $\mathbf{G}$  is a symmetric  $6 \times 6$  matrix. In this form the relationship is known as *Hooke's law*. For many natural materials in the linear elastic case,  $\mathbf{G}$  can be expressed by two independent constants, usually  $E$  (*Young's modulus*) and  $\nu$  (*Poisson's ratio*).

The optimization process will be demonstrated in two-dimensional space with linear elastic loads of homogeneous materials. Extensions to 3-D and to anisotropic and inhomogeneous materials are discussed below.

To get an optimal spring/mass system, we have to approximate the mass distribution by discrete mass points, and elasticity by the topology of the connections together with their spring constants. This is done in two steps. First, we find positions and masses of the points, then we insert and optimize the connections.

## 3 First Step: Positioning Points

Given an arbitrary body  $B$  with surface  $A$ , two ways of point positioning were tested. Regular grids (see below) with predefined connection structures were used as well as systems based on randomly given points.

### 3.1 The Iterated Voronoi Approach

Our observation was, that systems with arbitrary given points do not lead to good results in simulating elastic behaviour of homogeneous bodies. To solve this problem we developed a method of moving the points slightly in order to obtain a more homogeneous point positioning.

For each given point  $p_i \in P$ , the corresponding Voronoi region  $V(p_i, P)$  is defined by

$$V(p_i, P) = \{ |q - p_i| < |q - p_j|, q \in \mathbb{R}^3, p_j \in P - \{p_i\} \}.$$

The set of resulting regions for all  $p_i$  is a tessellation of the plane, it can be computed in  $O(n \log n)$  time for  $n$  given points in two-dimensional space (cf. [12]). Since the boundary of each  $V$  is polygonal, we can easily intersect it with  $A$ .

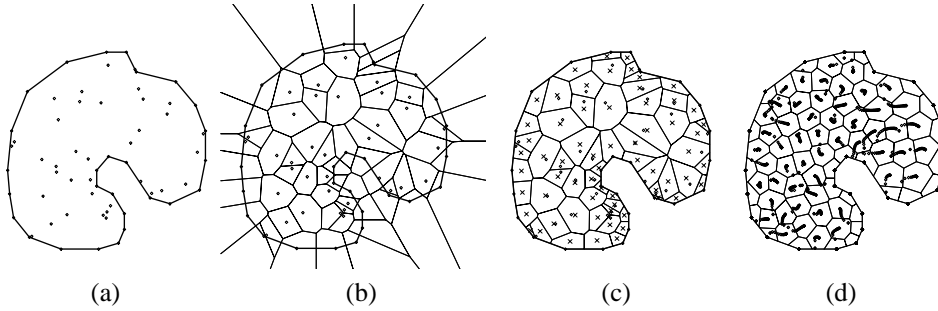


Figure 1: Point positioning - a) random points; b) Voronoi regions; c) cutted regions with moved points; d) regions after 15 iterations, together with point motion.

Next, the  $p_i$  are moved to the center of gravity of the corresponding Voronoi region (Figure 1). This process can be iterated. During iteration, the standard deviation of the point to point distances reduces as well as the standard deviation of the Voronoi-region sizes (Figure 2).

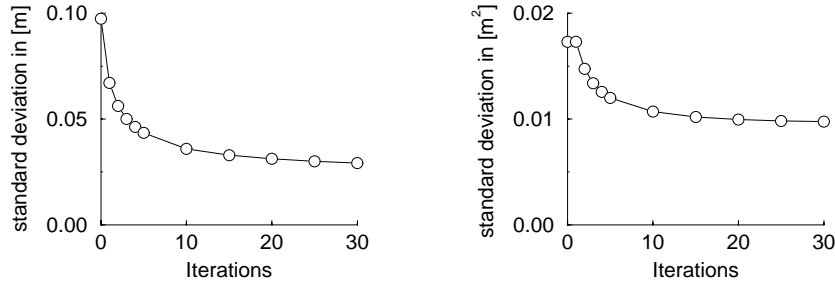


Figure 2: Standard deviation during iteration - a) point to point distances; b) sizes of Voronoi-regions.

Both results indicate a more homogeneous point positioning after some iterations. After the last iteration the weights are computed according to section 4 (see below).

### 3.2 Regular Grids

Besides random positioning of points, we used regular grids together with specific connection topologies. These configurations lead to good approximation results with the investigated loads (see below), but may have an anisotropic behaviour because of their regularity (like crystals do).

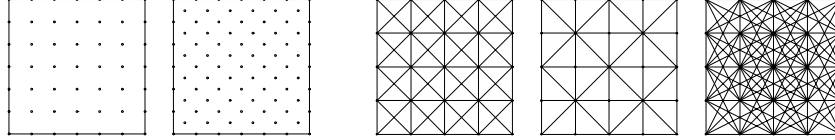


Figure 3: a) regular grids used; b) Suitable topologies to connect the grids.

To combine regular grids with arbitrary surfaces, tessellation is necessary. In the context of our quantitative study in the following paragraphs, we omit such tessellations because their results are hard to interpret.

## 4 Second Step: Calculation of the Mass Distribution

If the point positions  $\mathbf{u}_i$  are known, we have to approximate the mass distribution of  $B$  by assigning masses  $m(\mathbf{u}_i) := m_i$  to the points. The approximation of  $B$  by a set of tuples  $(\mathbf{u}_i, m_i)$  is considered to be good, if the mass moments  $p_{jkl}$  up to order two coincide. These moments determine the linear and angular accelerations to external forces and torques.

The relevant mass moments  $p_{jkl}$  of  $B$  are computed by

$$p_{jkl} = \int_B u_x^j u_y^k u_z^l \rho(\mathbf{u}) d\mathbf{u} \quad j + k + l \leq 2. \quad (4)$$

Substituting  $B$  by a finite set of points  $\mathbf{u}_i$  corresponds to approximating this integral by the sum

$$p_{jkl} = \sum_{i=1}^n u_{x_i}^j u_{y_i}^k u_{z_i}^l m_i \quad j + k + l \leq 2. \quad (5)$$

Equation (5) can be considered as the application of a cubature formula for the integrand function  $f(\mathbf{u}) = u_x^j u_y^k u_z^l \rho(\mathbf{u})$  over the domain  $B$ , where the points  $\mathbf{u}_i$  are the sample points and  $m_i$  are the weight coefficients for the formula. Hence, the approximation  $(\mathbf{u}_i, m_i)$  of  $B$  meets all relevant mass moments if and only if this cubature formula has at least *quadratic polynomial precision*, i.e., if and only if

$$\mathbf{A} \mathbf{m} = \mathbf{p}, \quad (6)$$

where  $\mathbf{m} = (m_1, \dots, m_{n-1})^T$  is the vector of mass coefficients,  $\mathbf{p} = (p_{000}, p_{100}, p_{010}, p_{001}, p_{110}, p_{101}, p_{011}, p_{200}, p_{020}, p_{002})^T$  is the 10-dimensional vector of exact mo-

ments of  $B$ , and  $\mathbf{A}$  is the  $(10 \times n)$ -matrix with

$$\mathbf{A}^T = \begin{pmatrix} 1 & u_{x1} & u_{y1} & \dots & u_{y1}u_{z1} & u_{x1}^2 & u_{y1}^2 & u_{z1}^2 \\ 1 & u_{x2} & u_{y2} & \dots & u_{y2}u_{z2} & u_{x2}^2 & u_{y2}^2 & u_{z2}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 1 & u_{xn} & u_{yn} & \dots & u_{yn}u_{zn} & u_{xn}^2 & u_{yn}^2 & u_{zn}^2 \end{pmatrix}.$$

The solution of the under determined system (6) with least Euclidian norm  $\|\mathbf{m}\|_2$  is of the form  $\mathbf{A}^T \mathbf{x}$  with  $\mathbf{x} \in \mathbb{R}^{10}$  [3]. Hence, the mass coefficients  $\mathbf{m} = \mathbf{A}^T \mathbf{x}$  can be found by solving

$$\mathbf{A} \mathbf{A}^T \mathbf{x} = \mathbf{p}. \quad (7)$$

Since  $\mathbf{A}$  is a Vandermonde matrix,  $\mathbf{A} \mathbf{A}^T$  is regular if there is a collection of 10 points  $\mathbf{u}_i$  which does not lie on a polynomial surface  $p(x, y, z) = 0$  of degree less than or equal to two.

Capturing third or higher order moments of  $B$  yields no significant improvement of the approximation und thus the degrees of freedom for the choice of the  $m_i$  are used in a way that  $\|\mathbf{m}\|_2$  is minimal. In this case the masses are distributed as uniform as possible.

## 5 Third Step: Optimizing Elasticity

Given  $n$  points with positions and masses. The connections can be found as follows: If the points are distributed randomly, we use a Delaunay triangulation to obtain pairs of points to be connected by springs. Two points are joined if they have a common edge in the Voronoi diagram. This triangulation method is widely used in finite element mechanics.

For regular grids, we tested several topologies in order to find suitable elastic approximations. Good results are given by structures as those shown in Figure 3(b).

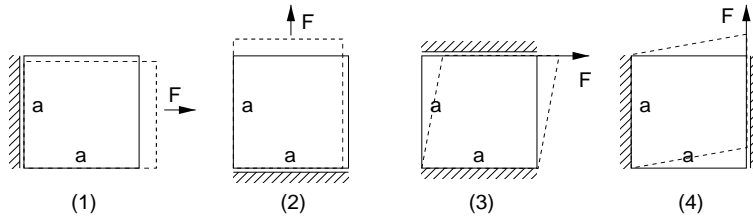


Figure 4: Test configurations with resulting deformations of a surface.

A good approximation of the elastic properties is found if the system's displacement due to some basic loads is similar to that of the given body  $B$ . Figure 4 shows the configurations used,  $B$  is a square plate. In situations (1) and (2), stretching loads are applied, shearing is done in situations (3) and (4).

## 5.1 Getting Reference Values

For some simple bodies and special configurations it is possible to get an analytic solution of the resulting displacements due to external forces (cf. [2]).<sup>2</sup>

As a reference, we calculate the deformation of the square plate shown in Figure 4(1). In this situation a uniform distributed force is applied to one side, the opposite one is constrained in  $x$ -direction, but not in  $y$ -direction.

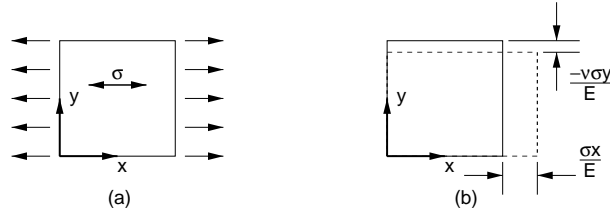


Figure 5: a) Plate with uniform distributed load to two opposite sides; b) resulting deformation.

This situation is equivalent to the application of a uniform distributed load  $\sigma$  to two opposite sides, whereby the body is constrained such that rigid body motions are permitted (Figure 5).

We assume a linear elastic deformation and a material that can be expressed by Poisson's ratio  $\nu$  and Young's modulus  $E$ . In section 2 we gave the general formulation of the strain-stress relation (see equation (3)). This relation, now written in matrix calculus, has to be satisfied:

$$\begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{pmatrix} = \frac{1}{E} \begin{pmatrix} 1 - \nu & \nu & 0 \\ \nu & 1 - \nu & 0 \\ 0 & 0 & 2(1 + \nu) \end{pmatrix} \begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix}, \quad (8)$$

as well as the equilibrium condition

$$\begin{pmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix} \begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \begin{pmatrix} F_x \\ F_y \end{pmatrix} \quad (9)$$

and the strain-displacement relation

$$\begin{pmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{pmatrix}. \quad (10)$$

Since rigid body motions are permitted, we have  $F_x = 0$ ,  $F_y = 0$  and assume

$$\sigma_x = \sigma, \quad \sigma_y = 0, \quad \tau_{xy} = 0$$

<sup>2</sup>For more complicated bodies and situations more precise finite element methods are used to get reference values.

which satisfies equation (9). The stresses can be computed by application of Hooke's law (equation (8)):

$$\epsilon_x = \frac{\sigma}{E}, \quad \epsilon_y = -\nu \frac{\sigma}{E}, \quad \gamma_{xy} = 0.$$

Integration according to equation (10) leads to the desired displacements

$$u_x = \frac{\sigma x}{E} + f(y), \quad u_y = -\nu \frac{\sigma y}{E} + g(x).$$

We set  $f(y)$  and  $g(x)$  to zero, the proof of correctness for doing this is omitted here.

## 5.2 Initial SMS Configuration

The initial spring constants of the SMS are set according to Young's modulus ( $E$ ). For a spring of resting length  $s_0$ , the spring constant is set heuristically to  $K = E * s_0/4$ . The upper part of Figure 8(a) shows the calculated displacements of the initial SMS, approximating the deformations of the given surface. They are drawn using dotted lines.

## 5.3 Fast Calculation of Displacements

During the optimization process, the displacements  $\mathbf{u}$  of the SMS which are due to the applied forces  $\mathbf{F}^e$  have to be calculated very often. Within the equilibrium state of the system they are obtained by solving

$$\mathbf{F}^e - \mathbf{K}\mathbf{u} = -\mathbf{F}^b, \quad (11)$$

which is the solution of equation (2) with vanishing  $\ddot{\mathbf{u}}$  and  $\dot{\mathbf{u}}$ . The additional vector  $\mathbf{F}^b$  describes the bearing forces in fixed points. The equation system is solved on the base of band matrices, the necessary node enumeration is done with an optimization method in a preprocessing step. Because  $\mathbf{K}$  is changed in every optimization step, it can not be inverted in advance.

## 5.4 Quality Criteria

For each point position of the SMS and each of the test configurations, the reference displacement  $\mathbf{u}^{ref}$  is calculated according to paragraph 5.1. The aforementioned FEM approach is used to compute the actual displacements  $\mathbf{u}$  during optimization.

One possibility to describe the quality of an SMS approximation is the standard deviation between actual and reference displacements of all its points:

$$G_s := \sqrt{\sum_{i=1}^4 (\mathbf{u}^{i,ref} - \mathbf{u}^i)^2}. \quad (12)$$

Here,  $\mathbf{u}^i$  denotes displacements according to the test configuration  $i$  of Figure 4. Other criteria are weighted standard deviations where border and corner points have a stronger influence. We denote the weight factors as  $G_{a:b:c}$ , where  $a$  is the weight of inner points,  $b$  that of border points and  $c$  of corner points.



## 5.5 Optimization by Simulated Annealing

Simulated annealing (cf. [1, 14]) is used as optimization method. This algorithm was originally designed for discrete problems but was later adapted to continuum problems. The method seems to be adequate because a pure gradient-descent delivered bad results, and analytic methods were very inefficient because of the high degree of freedom within the system. Another advantage is the simple algorithm (Figure 6).

```

temperature := initial_temperature;
G_min := 1034; { or some other good initial value :-) }
for i := 0 to no_of_temperatures-1 do begin
  temperature := temperature * ri; { over-exponential }
  good_steps := 0;
  k := 0;
  repeat
    Choose randomly an edge e;
    Change randomly spring constant of e;
    Calculate G according to equation (12);
    if (G < G_min) then begin { the system is now a better approximation }
      G_min := G;
      good_steps := good_steps + 1;
    end else begin
      Choose random number p ∈ [0, 1];
      if (p < e $\frac{G_{min}-G}{temperature}$ ) { take worse approximation }
        then G_min := G;
        else put back old spring constant of e;
      end
    end
  until (k>nlimit) or (good_steps>glimit)
end

```

Figure 6: Simulated annealing algorithm.

A drawback of the method is the high sensitivity to the number of temperatures, the initial temperature, the reduction function and the number of good steps (glimit) allowed in each temperature. Although, we found generally applicable values for those parameters (e = number of edges):

Parameter	initial_temperature	r	nlimit	glimit
Value	[0.0001..0.001]	[0.97..0.99]	e * 2	e

The simulated annealing process converges if nearly no bad steps are done any more (due to temperature) and the improvement at each temperature is below some threshold. We used an over-exponential temperature reduction to get convergence. Another possibility is to stop the algorithm after a specific number of temperatures (e.g. no\_of\_temperatures = 2\*e) as done in Figure 6. In this case, the time complexity of the algorithm is  $O(e^2)$ .

## 6 Extension to 3-D

Because equations (2) and (11) are formulated independently to the dimension, the method can be extended to 3-D without general changes. Instead of optimizing the four situations of Figure 4, now we have to consider nine basic loads (Figure 7).

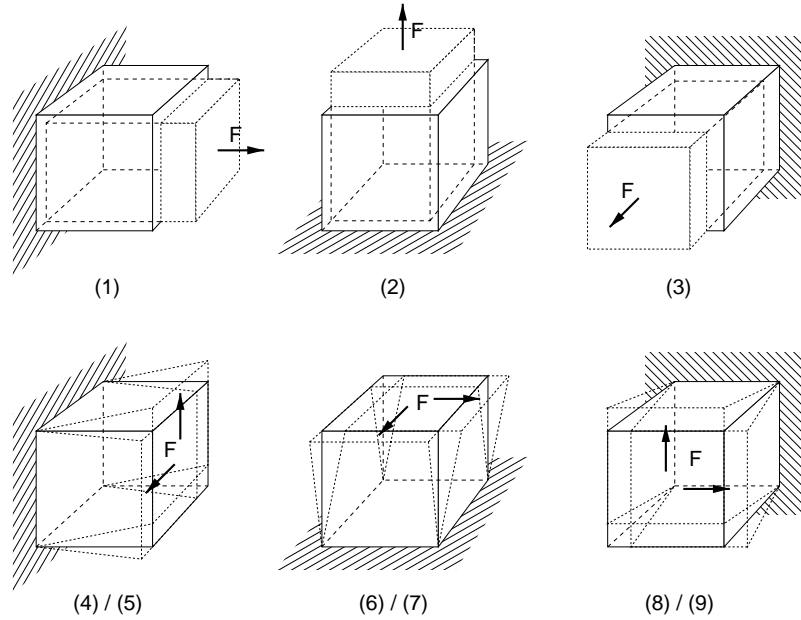


Figure 7: Basic loads in three dimensional space.

Reference values can be obtained similar to the 2-D method demonstrated in section 5.1. The annealing process works the same way as shown in Figure 6.

Some more effort is needed to calculate the Voronoi regions (now polyhedrons) within the mass point iteration. The number of edges obtained by the Delaunay triangulation is  $O(n)$  if working in 2-D, but  $O(n^{\frac{3}{2}})$  in three dimensional space (cf. [12]). This is of interest because the optimization has quadratic time complexity.

## 7 Results

In the upper part of Figure 8, initial displacements of a system with 25 nodes and connection structure 1 (see Figure 3(b)) are shown, in the lower part the system is optimized. The constants for horizontal, vertical and diagonal springs resp. are set to the same value, which is convenient for homogeneous bodies and reduces optimization effort. The same system with all constants optimized separately is shown in Figure 9.

Rectangular cells are useful to simulate anisotropic behaviour. This can be seen in Figure 10, where different elasticities in  $x$  and  $y$  direction are modeled.

An inhomogeneous material is simulated in Figure 11, the results are yet in need of improvement, but demonstrate the capability of the optimization.

Figure 12 shows the influence of Voronoi iterations on homogeneity and optimization quality of a nodal approximation.

## 8 Conclusion and Future Work

Spring mass systems are often used for real time simulation of complex deformable objects. Our method allows the generation of systems with definite mechanic behaviour. We found a way to obtain an adequate mass distribution and simulated homogeneous materials as well as inhomogeneities and anisotropies. Systems up to some hundred points were optimized successfully. For large systems the optimization is computational expensive but this has to be done only once.

Besides the full implementation of the 3-D case, we are working on frequency analysis. This is a promising approach because natural frequencies give some more information about the dynamic behaviour of deformable systems. Natural frequencies  $\omega_i$  can be obtained by solving the eigenvalue problem

$$(\mathbf{K} - \lambda\mathbf{M})\mathbf{u} = 0. \quad (13)$$

The frequencies are derived by  $\lambda_i = \omega_i^2$ . Once again we have the advantage that  $\mathbf{M}$  is diagonal, what makes efficient calculation methods applicable.

Frequency analysis is to be inserted into our optimization process in order to obtain nodal systems with good equilibrium and natural frequency approximations.

In addition, natural frequencies can be used to find very efficient integration methods for specific systems and situations. No stepsize control is needed if the highest natural frequencies are integrated with respect to the sampling theorem. This can be used in virtual reality applications like surgery simulation or virtual sculpting, for which our optimization process as well is of high interest.

## References

1. E.H.L. Aarts and P.J.M. van Laarhoven. *Statistical cooling: A general approach to combinatorial optimization problems*. Philips Journal of Research, (40):193–226, 1985.
2. K.-J. Bathe and E.L. Wilson. *Numerical methods in finite element analysis*. Prentice-Hall, 1976.
3. W. Boehm and H. Prautzsch. *Numerical methods*. AK Peters Wellesley, 1993.
4. D.E. Breen, D.H. House, and P.H. Getto. *A physically-based particle model of woven cloth*. The Visual Computer, (8):264–277, 1992.
5. H. Delingette, G. Subsol, S. Cotin, and J. Pignon. *A craniofacial surgery simulation testbed*. Research report 2199, Institute National de Recherche en Informatique et Automatique, 1994.
6. O. Deussen and Chr. Kuhn. *Echtzeitsimulation deformierbarer Objekte über nodale Modelle*. in: Th. Strothotte and P. Lorenz, Hrsg., Proc. Integration von Bild, Modell und Text. ASIM Mitteilungen No. 46, University of Magdeburg, 1995.

7. J. Easley. *Mechanics of elastic structures*. Prentice Hall, 1989.
8. Xiaochun Gao, Zhiying King, and Qixian Zhang. *A hybrid beam element for mathematical modelling of high-speed flexible linkages*. *Mech. Mach. Theory*, 24(1):29–36, 1989.
9. G.E. Kirsch. *Die Fundamentalgleichungen der Theorie der Elastizität fester Körper, hergeleitet aus der Betrachtung eines Systems von Punkten, welche durch elastische Streben verbunden sind*. *VDI-Zeitschrift*, 1868.
10. U.G. Kühnapfel, B. Neisius, H.G. Krumm, and M. Hübner. *CAD-Based simulation and modelling for endoscopic surgery*. *Endoscopic Surgery*, 1993.
11. G.S.P. Miller. *The motion dynamics of snakes and worms*. *Computer Graphics*, 22(4):169–173, 1988.
12. F.P. Preparata and M.I. Shamos. *Computational geometry*. Springer-Verlag New York, 175 Fifth Avenue, New York, New York 10010, USA, 3 Auflage, 1990.
13. D. Terzopoulos and K. Waters. *Analysis and synthesis of facial image sequences using physical and anatomical models*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993.
14. P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated annealing, theory and applications*. Reidel Publishing, Dordrecht, 1987.
15. K. Waters. *A physical model of facial tissue and muscle articulation derived from computer tomography data*. *Visualization in Biomedical Computing*, pp. 574–583. SPIE, 1992.

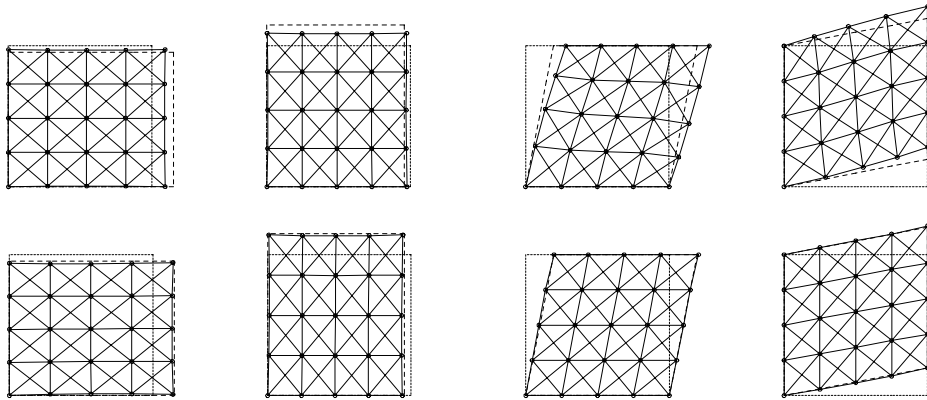


Figure 8: Initial (first line) and optimized displacements of a system with 25 nodes and connection structure 1. Stiffnesses for horizontal, vertical and diagonal springs are set to the same value ( $G_{1:6:18} : 2.8 \rightarrow 0.55$ )

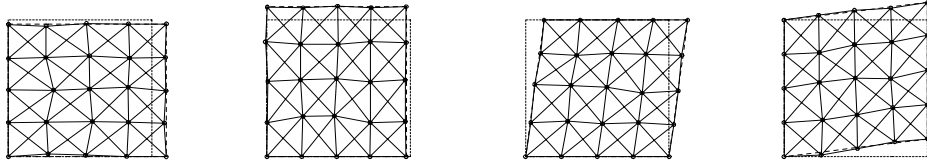


Figure 9: If all spring constants are optimized separate (which is to be done for inhomogeneous bodies) the system approximates the given deformations ( $G_{1:3:15} : 2.3 \rightarrow 0.6$ ) but has some irregularities within the grid.

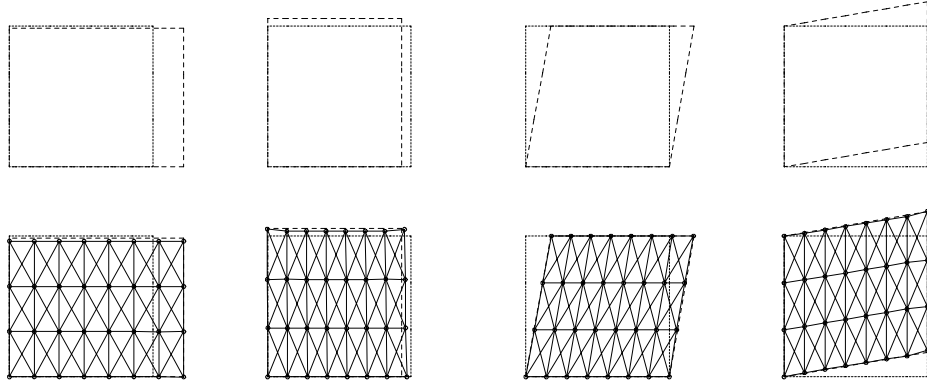


Figure 10: Modeling anisotropic behaviour with rectangular cells ( $G_{1:3:15} : 10.5 \rightarrow 0.48$ ). The system has different elasticities in  $x$  and  $y$ -direction, thus the deformations due to the test loads are different.

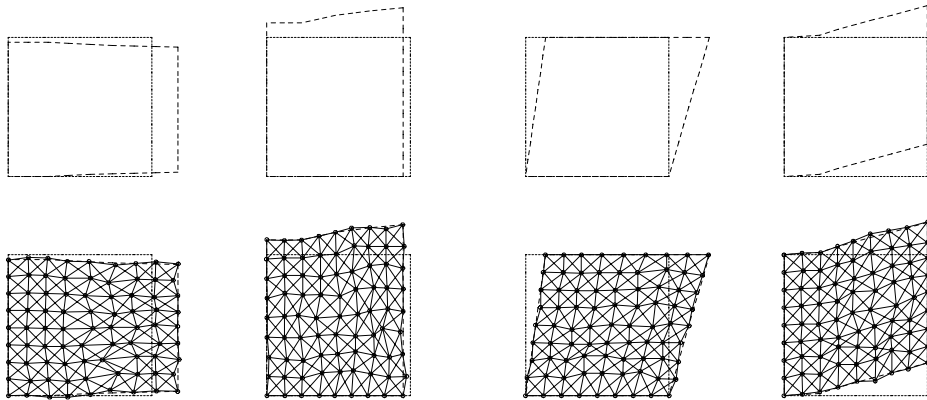
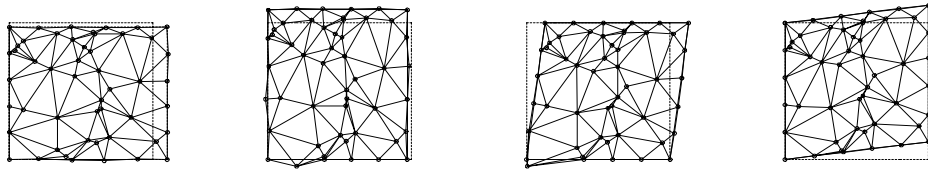
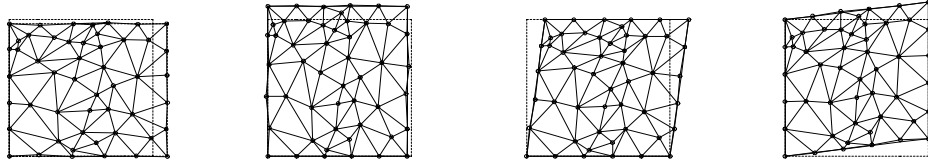


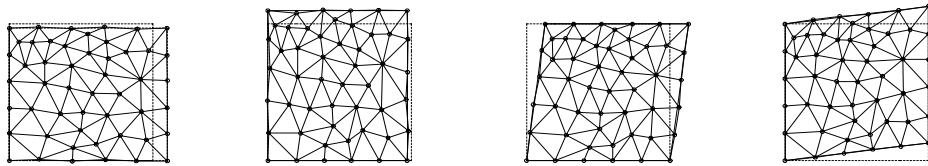
Figure 11: Simulation of an inhomogeneous material. The Young's modulus of the left third of the plate is twice as high as the right part. In the upper part of the figure the initial approximation of the spring/mass system was omitted. The lower part shows the optimized system.



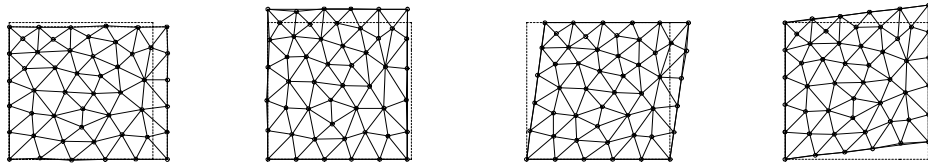
a) Initial given points



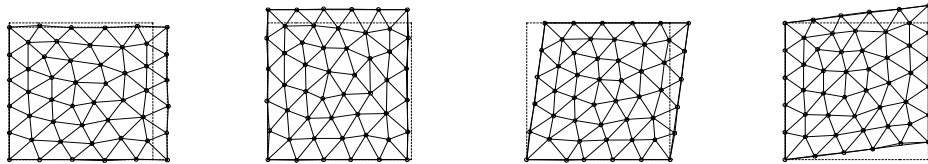
b) System after one iteration



c) System after five iterations



d) System after 15 iterations



e) System after 50 iterations

Voronoi iterations	0	1	5	15	50
$G_{1:3:15}$ before optimization	8.6	7.8	7.6	6.0	6.1
$G_{1:3:15}$ after optimization	0.45	0.47	0.44	0.42	0.37

Figure 12: Optimized displacements with different number of preceding Voronoi-iterations. The starting quality raises more than the quality after the optimization with 100 temperatures.