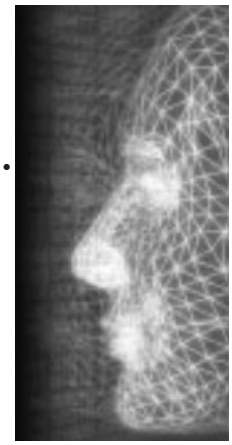
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2004; **15:** 211–218 (DOI: 10.1002/cav.23)

Mesh-free deformations



By Jian Chang and Jian J. Zhang*

Existing physically based deformation techniques, such as the finite element method (FEM) and the mass-spring systems (MSS), require the deformed object to be properly meshed. This is arguably the most expensive manual intervention process. In this paper, we propose a mesh-free deformation technique where only unconnected points are involved. The idea is to develop an approximate analytical solution using the Kelvin solution. Due to the fact that no mesh is involved, deforming a complex shape is as straightforward as deforming a simple one. Furthermore, the trade-off between efficiency and accuracy is easy to achieve by redistributing the points concerned. Experiments show that this method is fast and offers similar accuracy to the FEM. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: physically based modelling; deformation; mesh-free; point-based; computer animation

Introduction

The primary objective of physically based modelling in computer graphics is to help liberate the user/animator from onerous manual operations in generating plausible behaviors. Deformable object modelling has attracted enormous amount of research effort in traditional disciplines, such as mechanical engineering, and in computer animation. Ideally we would like to develop deformation techniques which can meet the following three objectives: good accuracy; high computational efficiency; and minimal manual preparation.

Great progress has been made on the first two objectives, owing to both clever simulation algorithms and improved hardware performance. Manual preparation arguably often contributes to the biggest portion of the project cycle and budget. But little has been done to reduce this burden. Although good progress has been made in computation speedup on various deformation techniques, most techniques, including FEM and MSS, still rely heavily on manual settings.

Mesh generation is usually one of the most time-consuming steps with existing deformation techniques. Although simple models do not suffer considerably, it is difficult for objects with a complex shape and structure. To alleviate this problem, in this paper we propose a point-based (mesh-free) technique for the computation of deformations. With its origin in continuum mechanics, our method is based on true physics of solid materials and is able to provide high physical accuracy efficiently comparing to the FEM.

*Correspondence to: Jian Chang, National Centre for Computer Animation, Media School, Bournemouth University, UK.
E-mail: jchang@bournemouth.ac.uk

Related Work

Physically based deformable object modelling has become increasingly popular in computer graphics and animation. Many researchers have made contributions to the development of more accurate, robust and efficient techniques.[1–6]

The MSS is an easy-to-understand technique in modelling deformable objects and there have been a large number of applications reported in the literature. It is intuitive and able to generate plausible visual results. For example, Miller[2] created convincing results for worms and snakes; Tu and Terzopoulos[3] used the spring to drive the movement of the artificial fish. With the MSS, an object is replaced (meshed) by a collection of mass points connected with springs. However, meshing and specifying the spring constants are understandably a laborious and time-consuming process.

In recent years, there has been a trend in computer graphics to adopt mature techniques from engineering, such as FEM, due to improved hardware performance. Using this technique, Chen and Zeltzer[4] developed a simple biomechanical muscle model. The effect of muscle contraction and expansion was simulated. Koch *et al*.[5] described an application on the deformation of human faces subject to craniofacial and maxillofacial operations.

Hirota[6] gave another example of how a human leg was modelled with the FEM taking anatomical structures into account.

The FEM, however, also requires the object to be properly meshed before any computation can be performed. This is a skilful job and great care must be taken to avoid distorted elements being produced. Distorted elements do not only result in inaccuracy, but sometimes instability in the resolution process. Automated mesh generation, despite numerous research efforts, still proves extremely tricky for complex 3D objects. Intervention by the operator is inevitable. Even for an experienced FEM operator, making a proper mesh for a complex model is not an easy job.

To remove the burden of the meshing task, a number of researchers have attempted various mesh-free methods for engineering applications. Most methods are formulated from the variation principle and require numerical integration. The efficiency of most such methods is, unfortunately, often not as good as the FEM. A recent review of mesh-free methods can be found in [7].

The weighted residual method[8,9] is based on the design of a trial displacement function. It is efficient, but finding a proper general trial function has proven extremely difficult.

Overview

The mesh-free method in this paper is oriented for computer graphics applications. It bears the merits of high efficiency, easy implementation, good accuracy and flexible control. To animate a deformable object, we can place displacements (displacement boundary conditions) or tractions (force boundary conditions) on the surface/boundary of our model and the object deforms accordingly.

Instead of meshing the object, we only distribute a necessary number of points in the region of interest. A set of collocation points (CPs) are created on the surface of the object to capture the geometric shape and the boundary conditions. Another set of points, known as the virtual source points (VSPs), are distributed around the object, where virtual forces are exerted. The deformation is approximated in such a manner that both the equilibrium state and the boundary conditions are satisfied.

The remainder of the paper is structured as follows. The second section presents the idea and basic formulation of our method in detail, and the third section describes the numerical implementation of the method. The fourth section discusses the accuracy and performance of the method with a number of simulation examples. The fifth section concludes the paper.

Mesh-Free Deformations

From the principle of superposition in elastic mechanics, the deformation under a sophisticated load case can be equivalently produced with a combination of simple load cases on the same object[10]. For instance, if the deformations of an elastic beam under pull, twist and bend are known, separately, the deformation of the beam under a combined load of these three actions is their linear combination. Thus using the known solutions under specific boundary conditions, we can construct the solutions for more general cases.

However, it is often difficult to obtain solutions for a given model, even under the simplest load case, without the help of numerical methods. The Kelvin problem,† where an infinite body is subject to an individual concentrated force, is one of the small number of problems where the analytical solution exists. The core of our technique is to develop an approximate analytical solution using the Kelvin solution.

Let us assume that an object is embedded into an infinite body. When a concentrated force acts on the infinite body, this embedded object will deform together with it due to the constraint imposed from the infinite body. The location of the concentrated force corresponds to some pattern of boundary constraints (forces and displacements) over the surface of the object, which can be formulated analytically with the solution to a Kelvin problem. The Kelvin solution defines also the physical state (stress, strain, etc.) of the object.

The concentrated force of the infinite body is called a virtual force that is acting on a VSP. With the virtual forces acting on different VSPs, various patterns of boundary conditions are constructed. Once the boundary conditions that the object is subject to are known, they can be approximated by a combination of these boundary condition patterns generated by the virtual forces. As each pattern of boundary conditions generates one deformation, which is given out as the Kelvin solution, the deformation of the object can be approximated by their combination.

One thing has to be kept in mind is that the Kelvin solution is singular on the VSPs. The VSPs, therefore, should not be located inside the model in order to avoid singularity appearing in our solution.

†The detail of the solution of a Kelvin problem can be found in Appendix.

To meet a set of boundary conditions exactly, we generally need an infinite number of virtual forces. In practice, however, a satisfactory approximation can be achieved with only a relatively small number of virtual forces. To represent the geometric shape and the boundary conditions, a set of CPs are distributed on the surface of the object. Each CP is assigned a value of boundary constraint, which is either a displacement or force. Where no constraint applies, the CPs are given a zero force value.

Let Q be a CP, P_i be a VSP. Let us also denote $U_{kl}(P_i, Q)$ be the x_l component of the displacement (solution) at point Q caused by an unit virtual force exerting on point P_i in direction x_k, and denote $T_{kl}(P_i, Q, \boldsymbol{n})$ the x_l component of the surface force at point Q, with surface normal $\boldsymbol{n}$, caused by the same unit virtual force. The expressions of both $U_{kl}(P_i, Q)$ and $T_{kl}(P_i, Q, \boldsymbol{n})$ are given in the Appendix. Both the displacements and surface forces at any collocation point Q within the domain subject to given constraints are

$$u_l(Q) = \sum_{i=1}^{n}\sum_{k=1}^{3} \alpha_{ik} U_{kl}(P_i, Q), \quad l = 1,2,3, \quad Q \in \Omega, \quad P_i \notin \Omega \tag{1.1}$$

$$p_l(Q, \boldsymbol{n}) = \sum_{i=1}^{n}\sum_{k=1}^{3} \alpha_{ik} T_{kl}(P_i, Q, \boldsymbol{n}), \quad l = 1,2,3, \quad Q \in \Omega, \quad P_i \notin \Omega \tag{1.2}$$

where $u_l(Q)$ represents the x_l component of the displacement at point Q and $p_l(Q, \boldsymbol{n})$ the x_l component of the surface force at point Q, where the surface normal is $\boldsymbol{n}$. α_{ik} are the weight coefficients to be determined, which stand for the strength of the corresponding virtual forces.

Suppose we have n VSPs around an object and m CPs on the boundary of the object to be deformed. We define the residual value R as:

$$R = \sum_{l=1}^{3}\sum_{j=1}^{mu} [u_l(Q_j) - u_l^0(Q_j)]^2 + \beta \sum_{l=1}^{3}\sum_{j=1}^{mf} [p_l(Q_j, \boldsymbol{n}) - p_l^0(Q_j)]^2 \tag{2}$$

where u_l and p_l are the approximations given by (1.1) and (1.2), u_l^0 and p_l^0 are the known boundary conditions defined at the CPs.

There are combined mu displacement CPs and mf force CPs, leading to a total of $mu + mf = m$ CPs. $\beta = \xi/E$ normalises the contribution of forces and displacements in (2), where E is Young's modulus, and ξ is a positive constant which is set to one in our paper. Minimising the residual value R we can determine the strength α_{ik} of virtual forces. The deformation is then computed by (1.1).

Implementation

Locating CPs and VSPs

The CPs are used to catch the geometric shape and allow boundary conditions to be represented. The density of CP distribution depends on the level of detail to be represented. From our experience, a sparse and regular distribution is sufficient for an object of a regular shape (Figure 1); a random spread of CPs works well for an object of free form nature (Figure 5). Many complex models are range-scanned where the object is represented by a large number of points. Without establishing a mesh, these points (or a selection of them) can be directly used as CPs.

The VSPs are located outside the model for virtual forces to be added. The greater the variation of the boundary conditions, the more VSPs should be used. It is easy to locate VSPs for a regular geometric model (Figure 1). For a complex model, one can randomly sample the surface points of the model and displace these points some distance away from the model (Figure 5). It is also acceptable to distribute the VSPs on a bounding box, such as a cube, sphere or cylinder.

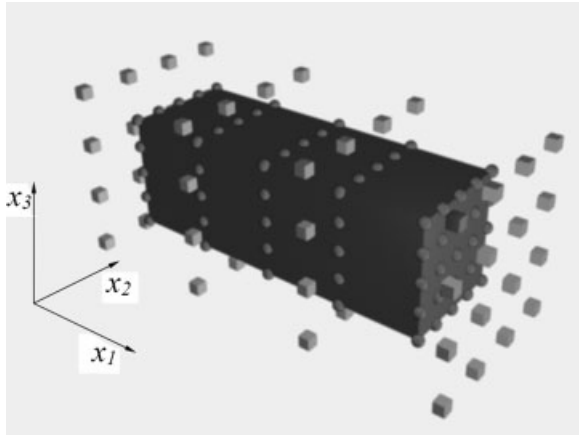


Figure 1. Deformable domain. The red spheres represent the CPs, and the green cubes are the VSPs where the virtual forces are exerted.

Deformation Computation

(1.1) and (1.2) can be rewritten in a matrix form as follows:

$$\begin{Bmatrix} u \\ p \end{Bmatrix} = \begin{bmatrix} U \\ T \end{bmatrix} \{\alpha\} \tag{3}$$

where $\{u\}$ stands for the displacements and $\{p\}$ for the surface forces on the CPs, the elements of matrices U and T are the values of corresponding Kelvin solutions at the CPs, and $\{\alpha\}$ are assembled by the weight coefficients α_{ik}.

Minimising the residual value R of (2) with respect to the weight coefficients α_{ik} results in the following:

$$\frac{\partial R}{\partial \alpha_{ik}} = 0, \quad i = 1, 2, \ldots, n; \quad k = 1, 2, 3 \tag{4}$$

which gives the linear equations:

$$\begin{bmatrix} U \\ \beta \cdot T \end{bmatrix}^T \begin{bmatrix} U \\ \beta \cdot T \end{bmatrix} \{\alpha\} = \begin{bmatrix} U \\ \beta \cdot T \end{bmatrix}^T \begin{Bmatrix} u^0 \\ \beta \cdot p^0 \end{Bmatrix} \tag{5}$$

Here, β is the same as that of (2). u^0 and p^0 are the known boundary conditions on the CPs.

The solution of (5) determines the unknown weight coefficients α_{ik}. Thus, the deformation of any point within the deformed object or on the boundary can be computed by (1.1) and the surface forces on the boundary can be computed by (1.2). The conjugate gradient procedure[11] is adopted to solve the linear equations. The complexity is $O(m \times n)$ for each iteration. It converges with a small number of iterations.

Results and Discussion

In this section, we compare our method with both the analytical results and the FEM in terms of accuracy and performance, and demonstrate its application with a few examples. All the computations were undertaken on a Dell Pentium III 500 MHz PC. The material parameters for all examples are given as follows: shear modulus, 120 MPa and Poisson's ratio, 0.45.

Computational Accuracy

To investigate the computational accuracy, we compared the results from our methods with the analytic results in Figure 2 when a beam is under pull, bend or twist. Both CPs and VSPs are distributed similarly to those in Figure 1. We used 450 CPs for these examples.

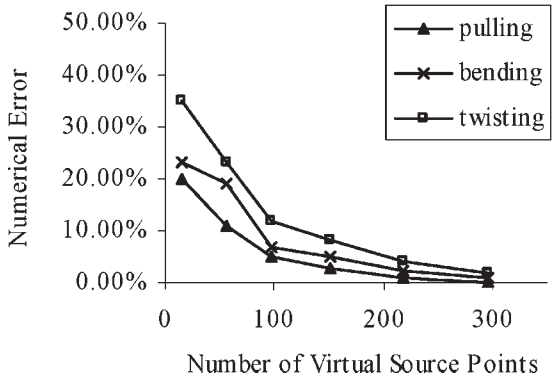


Figure 2. Computational accuracy affected by the number of VSPs.

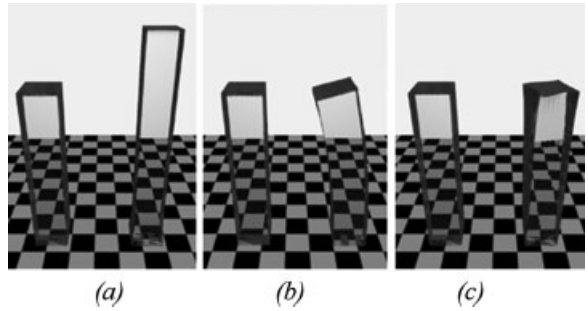


Figure 3. Beam under pull, bend and twist (218 VSPs; 450 CPs).

From the diagram it is seen that the accuracy is directly related to the number of VSPs. The greater the number of VSPs, the better accuracy it achieves. Nevertheless, even with a sparse distribution, such as 100 VSPs, a reasonable approximation can be achieved. Increasing the number of VSPs to 200, the error becomes negligible. It is worth pointing out that the number of CPs has a similar effect to resolution, i.e. the higher the number of CPs, the more detail can be represented. There is no need to distribute more CPs than the required modelling resolution.

Comparison with FEM

In Figure 4, a heavy rigid ball is placed on top of a soft block that has a cylindrical hole in the middle. Figures 4(a) and 4(b) were produced with our method (both represent the same configuration). Figures 4(c) and 4(d)

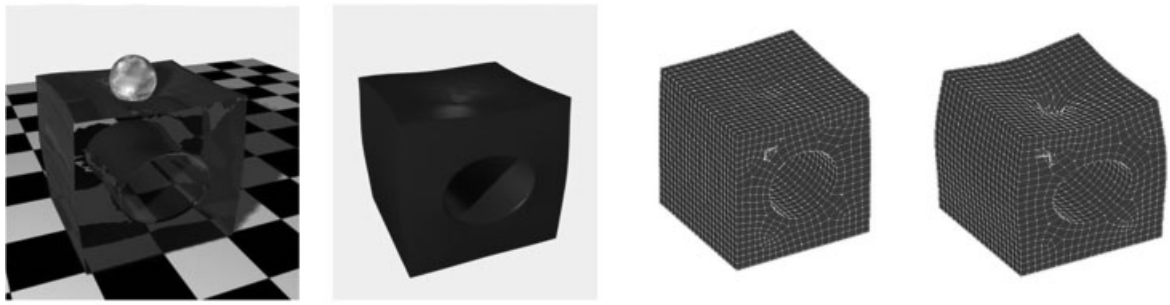


Figure 4. A soft block with a heavy rigid ball resting on it.

Mesh-free method (Figures 4a & 4b)	FEM linear small strain (Figure 4c)	FEM non-linear large deformation (Figure 4d)
2ls	63s	629s

Table I. Computation time for the example in Figure 4

were produced with the commercial FEM software, Ansys, where Figure 4(c) is computed with the linear small strain analysis module and Figure 4(d) with the nonlinear large deformation analysis module. 1200 CPs were distributed on the surface of the object, and 432 VSPs were used in our model. For the FEM computations, 6000 elements were used. The computation time is given in Table 1.

It is interesting to see that not only is our method fast, but the result of our method matches well with the nonlinear analysis of the FEM. The reason is that our method rephrases the displacement as a combination of solutions that have global influence, whereas the FEM decomposes the displacement into solutions of the small elements, which turns to localize the influence.

The Bunny Model

In this section, we deform the Stanford bunny model[12] to demonstrate how our method can be applied to the deformation of complex objects.

940 CPs are distributed on the surface of the bunny (Figure 5(a)). Two settings of VSPs are used. The coarser setting consists of 94 VSPs (Figure (5b)) and the denser setting of 432 VSPs (Figure (5c)).

Since the method computes not only deformations (1.1), but also surface forces (1.2), it is useful also for force feedback applications. In Figure 6, the bunny model (432 VSPs) is fixed at the bottom and is pushed from the behind by a glass brick. This figure shows the colour-coded force distribution both on the back surface (left) and the bottom surface (right).

Another deformation case is given in Figure 7 where a ball rests on the bunny's back. Figures 7(b) and 7(c) take the same boundary conditions and collocation points, but with a different number of VSPs. In Figure 7(b), 94 VSPs are chosen, and the model looks a little stiffer than that in Figure 7(c) where 432 VSPs are used. A bump is noticeable on the bunny near the glass ball in Figure 7(c), whereas Figure 7(b) is smoother, revealing less subtle

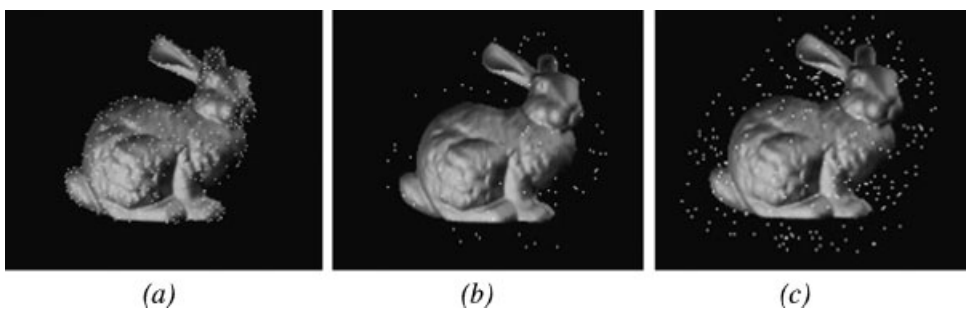


Figure 5. Distribution of CPs and VSPs on the bunny model.

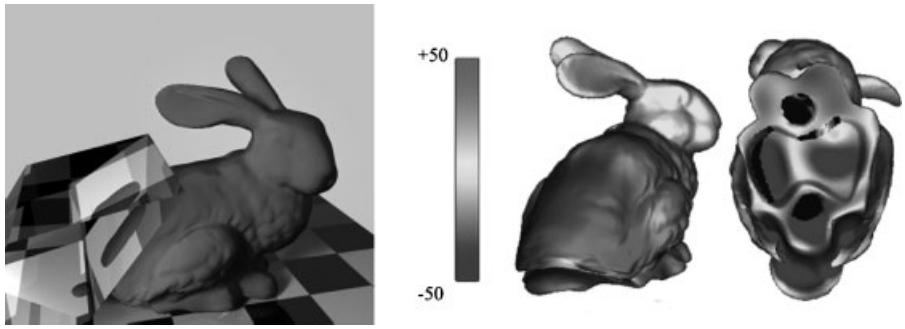


Figure 6. Force feedback (one unit of surface force stands for 1Mpa).

Figure 7. (a) Original bunny model. (b) Deformed with 94 VSPs. (c) Deformed with 432 VSPs.

detail. This is because more VSPs produce more accurate approximation to the real deformation.

The Hippopotamus Model

In Figure 8, a hippopotamus model[13] is being squeezed from both sides. It is noted that the hippopotamus deforms naturally and its volume remains nearly unchanged during the process. As the model is assumed to be elastic and to behave like a rubber toy, the shape of the head also deforms due to the squeezing force on the body.

Conclusion and Future Work

In this paper, we have presented a deformation method whose distinctive feature is the elimination of the meshing process. Both the deformations and surface forces can be computed accurately and efficiently.

In addition to eliminating the laborious meshing process, being mesh-free, this method has many advantages. First, deforming a geometrically complex object is just as easy as for simple objects, since the CPs and VSPs are easily distributed. Second, it can even be used for point clouds where the internal geometrical relations are

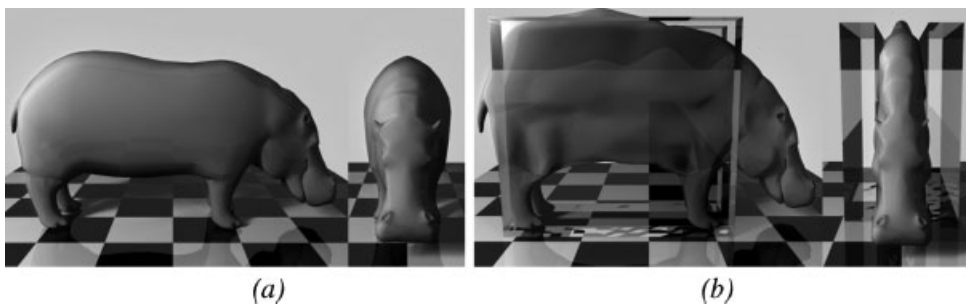


Figure 8. Squeezed hippopotamus model.

unknown. This property is especially valuable in situations where topological and geometrical information is missing, such as for point based graphics applications. Third, the resolution and accuracy can be controlled by having a different number of CPs and VSPs. Thus the trade-off between speed and accuracy is controllable. Fourth, our method is fast, because the discretization is done only on the surface of the object and all the coefficients are determined directly without requiring integration. Finally, we have observed that the result of our method is more visually plausible than that of the FEM linear module, as it does not suffer from the drawback of deformation localisation caused by the meshing process. Figure 4 shows that our method is able to generate similar effects to those which are only achievable with the FEM large deformation module, but with a fraction of the computing cost.

Although the examples we have given concern only surface deformation, this method is equally applicable to volumetric models. The collocation points can be distributed both on the surface and inside of an object, where deformation can be picked up. This property may not seem crucial to animation. It will be significant in other graphical simulation applications, such as medical visualisation.

The computational efficiency of the presented technique is comparable with other physically based deformation methods. However, more and more applications require real-time performance. In the future, we intend to plug this gap and develop a real-time mesh-free deformation technique based on this work.

ACKNOWLEDGEMENTS

We would like to thank Joyce Power and Mark Haines for proofreading the manuscript. This project has been funded in part by the Arts and Humanities Research Board (grant no. B/RG/AN5263/ABPN12727). The authors would also like to thank the anonymous reviewers for their valuable suggestions.

References

1. Terzopoulos D, Platt J, Barr A, Fleischer K. Elastically deformable models. In *Proceedings of SIGGRAPH*, 1987; 205–214.
2. Miller GSP. The motion dynamics of snakes and worms. In *Proceedings of SIGGRAPH*, 1988; 169–178.
3. Tu X, Terzopoulos D. Artificial fishes: physics, locomotion, perception, behavior. In *Proceedings of SIGGRAPH*, 1994; 43–50.
4. Chen DT, Zeltzer D. Pump it up: computer animation of a biomechanically based model of muscle using the finite element method. In *Proceedings of SIGGRAPH*, 1992; 89–98.
5. Koch RM, Gross MH, Carls FL, Buren DF, Fankhauser G, Parish YIH. Simulating facial surgery using finite element methods. In *Proceedings of SIGGRAPH*, 1996; 421-428.
6. Hirota G. An improved finite element contact model for anatomical simulations. PhD thesis, University of North Carolina, 2002.
7. Liu GL. *Mesh Free Methods, Moving Beyond the Finite Element Method*. CRC Press LLC; 2003.
8. Baraff D, Witkin A. Dynamic simulation of non-penetrating flexible bodies. In *Proceedings of SIGGRAPH*, 1992; 303–308.
9. You LH, Zhang JJ, Comninos P. A solid volumetric deformable muscle model for computer animation using weighted residual method. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**: 853–863.
10. Saada AS. *Elasticity Theory and Applications* (2nd edn). Krieger Publishing Company, 1993.
11. Golub GH, Charles F, Loan V. *Matrix Computations*. The Johns Hopkins University Press: Baltimore, MD, USA; 1983.
12. Stanford computer graphics laboratory. http://www-graphics.stanford.edu
13. Wiedermann J. 500 3D-objects. Taschen, 2002, http://www.taschen.com

Appendix. Solution of Kelvin Problem

A Kelvin problem defines the mechanical behavior when a single concentrated force acts in the interior of an infinite body, as illustrated in Figure A.1 where concentrated force R acts on the origin point P along x_3.

$U_{ij}(P, Q)$ is the x_j component of the displacement at any point Q caused by a unit concentrated force exerting

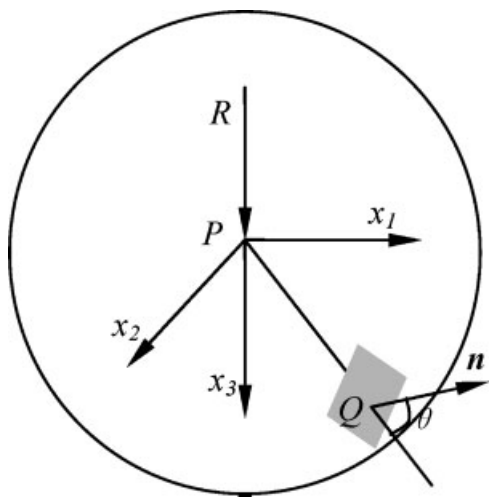


Figure A.1. A single concentrated force exerts on an infinite elastic body.

on point P in direction x_i, and $T_{ij}(P, Q, \boldsymbol{n})$ is the x_j component of the surface force at point Q, with surface normal $\boldsymbol{n}$, caused by the same unit concentrated force.

For $i=1, 2, 3$ and $j=1, 2, 3$, we have:

$$U_{ij}(P,Q) = \frac{1}{16\pi G(1-\nu)r_{PQ}}\left[(3-4\nu)\delta_{ij} + \frac{x_i^{PQ}x_j^{PQ}}{r_{PQ}^2}\right]$$

$$T_{ij}(P,Q,\boldsymbol{n}) = \frac{-1}{8\pi(1-\nu)r_{PQ}^2}\left\{\left[(1-2\nu)\delta_{ij} + 3\frac{x_i^{PQ}x_j^{PQ}}{r_{PQ}^2}\right]\right.$$
$$\left.\cos\theta - (1-2\nu)\left[\frac{x_i^{PQ}n_j}{r_{PQ}} - \frac{x_j^{PQ}n_i}{r_{PQ}}\right]\right\} \qquad \text{(A.1)}$$

where G is the shear modulus, ν Poisson's ratio, and

$$\delta_{i,j} = \begin{cases} 0 & i \neq j; \\ 1 & i = j; \end{cases}$$
$$x_i^{PQ} = x_i(Q) - x_i(P)$$
$$r_{PQ} = \sqrt{(x_1^{PQ})^2 + (x_2^{PQ})^2 + (x_3^{PQ})^2}$$
$$\cos\theta = \frac{x_1^{PQ}}{r_{PQ}}n_1 + \frac{x_2^{PQ}}{r_{PQ}}n_2 + \frac{x_3^{PQ}}{r_{PQ}}n_3$$

In[10], more detailed description of Kelvin problem can be found.