Chapter 1

Introduction

1.1 Training surgeons

Performing surgical operations is traditionally taught in an apprentice/master setting. The surgeon in training watches accomplished surgeons perform an operation, and after sufficient experience, he may perform operations under expert guidance. After much practice, the trainee then becomes an expert in operations.

Although this system of educating surgeons is effective, it has serious drawbacks. Due to lack of experience, trainees take longer to perform procedures, which increases costs. They are also less skilled, which subjects patients to extra risks. This is not a desirable situation, but the alternatives to training within the operating room also have disadvantages. Books and videos describing operations are not interactive. Test animals do not always reflect human anatomy, and their use is expensive. Synthetic phantoms do not reflect mechanic properties of living tissue, and have to be discarded after dissecting them.

These issues are aggravated by the introduction of *laparoscopic* or *endoscopic* surgery. In this technique, the abdomen can be operated on without major trauma. Instruments, such as graspers, scissors, and staplers, are introduced in the body through small holes in the abdomen. These instruments are mounted on long rods, and surgeons can operate them from the outside. The operating site is viewed through a laparoscope, a tube-like device which contains a camera and a lamp. It also inserted through a small incision. Compared to open surgery, the trauma caused by the openings is small. This speeds up patient recovery, and reduces pain and scarring.

For patients, the benefits of laparoscopic surgery are clear. For surgeons however, it opens up a range of new problems. During an intervention the surgeon cannot directly see the operating site, but must rely on a coarse 2D video image of the site. Since the instruments and the camera are introduced through small holes, their movements are restricted. This makes manipulating them awkward. Elastic response of the tissue is relayed through wires to the surgeon, thus reducing kinesthetic feedback. Hence, surgeons have reduced visual and haptic feedback during laparoscopic surgery, and cannot rely on traditional hand-eye coordination. Training a new laparoscopic procedure takes more practice than learning new traditional surgical procedures.

It has been pointed out [8, 9] that computer-assisted training might offer a solution to these problems. If apprentices are initially trained on “virtual” patients, simulated by computers, no costly operating rooms have to be used, and during practice there is space for making errors without consequences. If the simulation is sufficiently advanced, it may be more realistic than animals and phantoms. Students can make errors and learn from them, and they can experiment with different surgical techniques. Moreover, a virtual environment may be used to recreate unusual complications, and train students for situations that occur only sporadically in practice.

A virtual environment is also conducive to experimentation. Virtual environments can be used in experiments to determine skills and techniques that are effective for surgical procedures. This is useful because little is known of the exact nature of surgical skills [94].

1.2 Interactive surgery simulation

In summary, interactive surgery simulations could be a highly useful tool for training surgical procedures. Unfortunately, constructing such simulations is a technically challenging task. Consider such a hypothetical system, consisting of a computer connected with a fancy display and specialized “joystick” that also renders reaction forces (a *force feedback* or *haptic* device) . Such a system lets the student manipulate virtual organs, and feel their reactions. When the student uses the joystick to push, cut or otherwise manipulate simulated organs, the system should respond to these actions with a credible reaction. The time available for computing these reactions is severely limited. Typically, the display must be updated within 40 milliseconds, and reaction forces should be relayed to the haptic device within 2 milliseconds. Producing a realistic deformation of a virtual organs in so little time is a hard task, and it is solely this topic that the rest of the thesis is focused on.

Full-fledged systems for the problem at hand, which include realistic visualization and simulated surgical instruments, already exist. The most popular technique for simulating the soft tissue in these systems are mass-spring-damper systems [12, 13, 20, 28, 46, 67, 70, 71, 76, 83, 95]: these consist of mass points connected by a network of damped springs. Other techniques have also been presented, for example space-filling spheres [91] and ChainMail [83]. The role of such models is to provide deformations that look qualitatively convincing: the result should “look good.” The heuristic nature of these models makes it impossible to go beyond looking good. For example, mass-spring-damper networks do not account for the volume in between the mass points, and hence they cannot simulate truly volumetric properties of material, such as volume preservation.

The lack of quantifiable realism in heuristic deformation models has motivated the move towards methods that are based on the physics of deformation. Deformations of an elastic object are described by laws of physics. These laws can be translated into precise mathematical descriptions in the form of partial differential equations, which can be discretized and solved. The solutions thus found approximate reality in a quantifiable manner. Such discretization methods include Finite Element [14, 103], Finite

Difference [64], and Boundary Element Methods [58].

This thesis only considers the Finite Element Method (FEM). The FEM is a well-studied and highly popular solution method to compute solutions to partial differential equations defined on irregularly shaped objects. Among others, it can be applied to problems of heat conduction, elasticity and electromagnetism. The FEM solves such problems by subdividing the object into a collection of geometric primitives, called *elements*. This process is called meshing. These elements, e.g., triangles, tetrahedra or bricks, can only deform in elementary ways. Physical laws of the material specify how neighboring elements of a mesh interact. The relation between neighboring elements can thus be captured in an equation. The aggregate of all elements together leads to an enormous system of equations, which can be solved with numerical techniques. The solution to this discretized system is an approximation to the solution of the continuous system. A schematic example is shown in Figure 1.1.

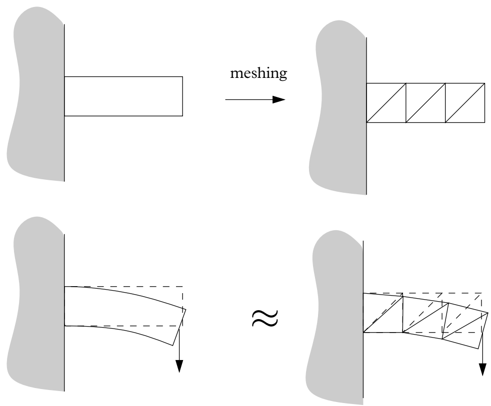

Figure 1.1: In the Finite Element Method, the object simulated (top left) is meshed (top right). The exact solution (bottom left) is then approximated by the deformation of the meshed object (bottom right).

The FEM has been applied before to interactive deformation and surgery simulation. We can distinguish two techniques. Linear elasticity simplifies the equations of elasticity into a linear system. The inverse of this system can be computed beforehand, so that elastic response to external forces can be computed within a guaranteed amount of time. This is a desirable property in interactive simulations, and such precomputation techniques have been used widely in both prototype surgery simulations [18, 37], and generic deformable object simulations [53]. Unfortunately, the linear approximation is only valid when deformations are small, an assumption that is questionable for soft material. In particular, linear elasticity cannot realistically model deformations that involve rotations; an example of linear and nonlinear elasticity is shown in Figure 1.2. Accurately describing these deformations requires nonlinear elasticity, which cannot be used in conjunction with precomputation techniques. Such problems must be solved iteratively: the configuration is moved from some starting configuration to the solution in small steps. We also refer to such a method as a *relaxation method*. The most popular iterative method is explicit dynamic time-stepping, or *dynamic relaxation*. The object is considered to have mass and damping, and the evolution of its movements is described by Newton's laws of motion, which can be numerically computed. Dynamic relaxation has been used for computing solutions to nonlinear elasticity problems in prototype simulation systems [32, 77, 92, 100, 101]. Hybrid systems of static linear precomputed and time-stepping techniques have also been presented [27, 49]. Time-stepping is sometimes also used for linear elasticity, since velocity-dependent friction forces can be integrated in a dynamic model naturally [2].

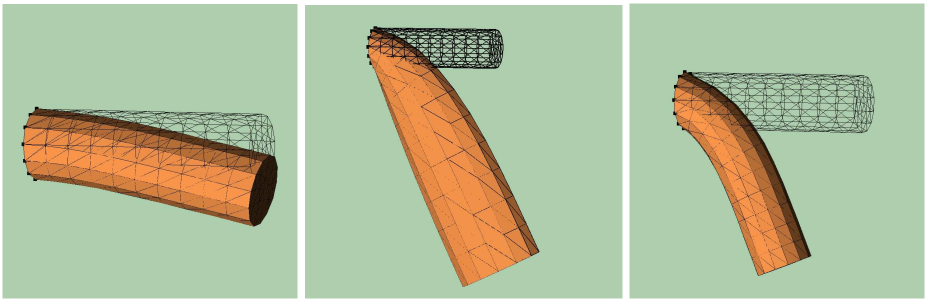

Figure 1.2: Linear elasticity is accurate for small deformations: on the left, an object fixed on the left under gravity load. The original is shown in wireframe. These deformations do not scale linearly: in the center, the same object with force scaled by 10 with linear elasticity. On the right, the same object with a nonlinear elasticity model.

Iterative methods have another advantage over linear elasticity with precomputation. In a surgery simulation, destructive operations such as cuts and cauterizations, can change the mesh. Mesh changes invalidate the precomputed structures that are used in the linear model. By contrast, mesh changes are handled naturally in an iterative method.

Mesh changes can include both simulated surgical procedures and mesh refinements to increase the precision of the FEM solution. In this thesis we will consider both cuts and refinements. Cuts have previously been simulated using subdivision methods: a virtual scalpel slices through an object represented by a mesh, and all elements in contact with the virtual scalpel are subdivided. An example of a subdivision cut in 2D is given in Figure 1.3. In 3D, subdivision techniques for cutting were pioneered by Bielser et al. [12]. Their work has been followed by many other researchers [22, 45, 46, 66]. Subdivision methods can represent cuts accurately. Unfortunately, they increase mesh size substantially.

1.3 Overview of thesis

In this thesis, we will consider nonlinearly deformable objects where the mesh representing the object can be changed by cuts; hence the title "Cutting in deformable objects." More precisely, we will use the FEM for modeling interactive deformation, and solve

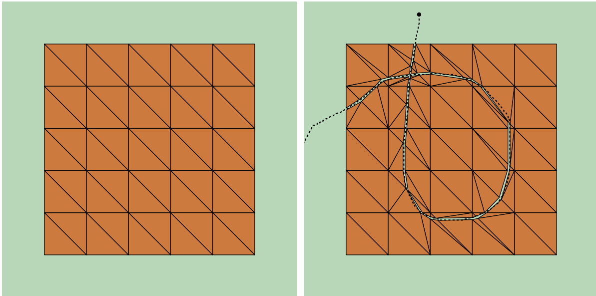

Figure 1.3: A triangle mesh (left), and cut in that mesh produced by a subdivision method (right). The mesh mirrors the scalpel path (dotted) accurately, but uses many small and skinny triangles to do so.

the resulting equations with an iterative method, allowing both the use of nonlinear elasticity models and run-time mesh-changes.

The FEM is a technique traditionally used in engineering: it is a tool to compute solutions to engineering problems with high accuracy. In these cases the analysis proceeds in three separate steps. First, the object is meshed. Then, for each element the local equations are generated and assembled in a large system of equations. Finally, this system of equations is solved. Typically, these three steps are performed by routines that communicate via files on disk or matrices stored in memory.

In an interactive simulation, low response times are very important, while accuracy is not. For quick responses, meshing, equation assembly and system solution should be tightly integrated; structures such as matrices in memory or files on disk cause undesirable overhead. Hence, at the surface, building a deformation simulation seems like a problem in program design: the mesh and the deformations should be stored such that communication between different modules can be done efficiently, while a certain flexibility in the complete system is maintained.

It is true that complex integrated systems can only be implemented when they are designed in a modular manner. However, in the case of FEM computations, the mesh and the solution process must be integrated on a deeper level than program design. Both the accuracy and speed of solution processes strongly depend on the granularity and the quality of the mesh: larger meshes slow down computations, as do meshes with skinny and flat elements. Moreover, flat elements introduce errors in the approximation. Figure 1.4 illustrates this phenomenon. If the mesh is changed on-line, for example due to cuts, care must be taken that the changes do not adversely affect the performance of the total system. So, meshing and deformation are closely coupled, and it is not realistic to treat both problems separately.

Our first venture into the problem of cuts in deformable objects is described[1] in Chapter 3. Here, both problems were treated separately. This has led to the implementation of a system that separates deformation and meshing with the highest degree of

[1] Results from Chapter 3 were presented at the Medical Image Computing and Computer Assisted Intervention (MICCAI) conference 2001 [73] and EuroGraphics 2000 [72]

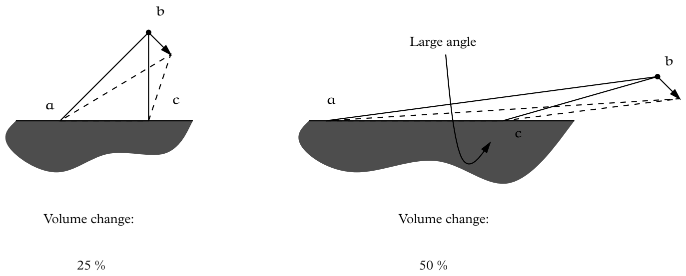

Figure 1.4: Moving a single point of a triangular element. The area of the original element (solid lines) is equal in all cases. The same movement leads to volume reduction of 25% on the left, and 50 % on the right. In the case on the right, an innocuous movement leads to a large volume change, and therefore, large internal forces. The element is stiffer than the real material, and using such flat triangles in the Finite Element discretization yields an inaccurate approximation.

modularity. The system uses linear elasticity using a static iterative relaxation method. A cutting technique was tried that does not increase mesh size like subdivision does. The system succeeds in combining static FEM deformation and interactive cuts on relatively large meshes in a stable manner. However, the cutting method also introduces flat, undesirable elements in the mesh, which must be removed in a separate step.

Chapter 3 uses a *static* approach: an approach where physical time is not simulated. Most other work in deformable object modeling uses dynamic relaxation. In Chapter 4, we take a more in-depth look at static relaxation: this chapter presents a computational study to determine which method performs best on nonlinear problems. Both methods are coded in the same framework. They are compared using a standardized test problem. By timing how long it takes before they reach the final solution, we can determine which one performs best. The conclusion is that the static method is at least as good as the dynamic one: with an optimal choice of parameters, dynamic relaxation is as fast as static relaxation, otherwise it is slower. In this chapter it is also determined how much computational power is needed for running a simulation. The observation that mesh quality influences the performance of an iterative method is confirmed for the nonlinear case as well.

Chapter 3 also proposes a method for cutting in 3D objects represented as tetrahedral meshes. During the development of this method, we have encountered many problems that are not intrinsically three-dimensional, but are partially caused by limitations of both computers and our minds. Computers can only display 2D pictures and we can only see objects from the outside; visualizing what happens inside an object is much harder, both mentally and technically. Therefore, in Chapter 5 we take a step back, and present a method for making cuts in 2D triangle meshes.[2] Existing methods use subdivision, a process that increases the size of the mesh and decreases the quality of the mesh. Both are undesirable, since they make the relaxation more expensive.

[2] Chapter 5 was based on a paper accepted for the Fifth International Workshop on Algorithmic Foundations of Robotics (WAFR 2002) [74].

Our approach uses a recognized technique for making high-quality triangle meshes, the Delaunay triangulation. The result is a technique that produces meshes that are measurably better and smaller than those produced with subdivision techniques. The technique is also generalized to curved 3D surfaces, where one scalpel can cause multiple incisions.

In Chapter 6 we turn to a more specific problem, in which the full generality of arbitrary cuts in arbitrary objects is not necessary. This problem concerns simulating how needles are inserted in deforming objects. It has been solved for 2D models with a linear elastic model [36–38], but the extension of this technique to 3D objects and nonlinear elasticity has not been addressed yet. In Chapter 6, we investigate techniques for 2D needle insertion, that might make 3D needle insertion in nonlinear material more tractable. Unlike the work in the preceding chapters, the focus in this chapter is more on quantified accuracy than on visual realism. In this case mesh changes take the form of refinements close to the location of the needle. The resulting simulation has a performance comparable to the work cited, but can also simulate a number of nonlinear models, and readily generalizes to 3D.

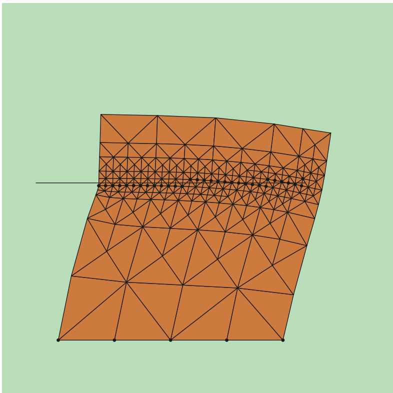

Figure 1.5: Needle insertion in nonlinear material. The mesh is refined close to the needle to increase accuracy.

Finally, Chapter 3 to 6 all manipulate triangle and tetrahedron meshes. For efficiency reasons, the connectivity information of these meshes must be stored explicitly, in the form of pointers that link neighboring triangles and tetrahedra. A side-result of the implementation work of these chapters is a data structure that separates the low-level work of maintaining this connectivity information from the rest of the program, thus making it easy to robustly implement high-level mesh-changes. Chapter 7 describes the data structure, and gives pseudo-code for the low-level operations.