CHAPTER - 1

MATHEMATICAL, GEOMETRIC AND SIMULATION MODELING

1. Mathematical, Geometric and Simulation Modeling

The chapter 1 deals with the need and applications of simulation, visualization and interactions along with the concepts of different types of modeling such as mathematical, geometric and simulation.

More and more disciplines are contributing to the modeling of complex phenomena in computer graphics, and the richness of the visual effects in animation now being popular is due to the variety of techniques that are being used together to produce these phenomena. To distinguish animation and scientific visualization, first one is based on making something that looks good (ad-hoc method), whereas scientific visualization is based on replicating the underlying physics or mathematics behind the objects, being modeled. Modeling fluid and its dynamics are particularly significant challenges. Even within the realms discussed in the literature, there is significant scope, for further work.

Many real world objects like fluid smoke clouds steam and fog are inherently smooth and much of computer graphics involves modeling the real world. The need to represent free surface arises in two cases: First in modeling existing objects (such as surface of a car, a face, or a mountain) and second in modeling “from scratch “where no pre-existing physical object is being represented. In the first case, a mathematical description of the object may be unavailable. Of course one can use the coordinates of the infinitely many points of the objects to model. But this is not feasible for a computer with finite storage. More often, we merely approximate the object with pieces of planes spheres or other shapes that are easy to describe mathematically and require that the points on our model be close to corresponding points on the object. In the second case, when there is no pre-existing object to model, the user creates the object in the modeling process; hence the object matches its representation exactly,
because its only embodiment is the representation. To create the object, the user may sculpt the object interactively, describe it mathematically or give an approximation description to be 'filled in' by some program.

1.1 Mathematical Modeling

A **mathematical model** uses mathematical language to explain a system. Mathematical models are used in the natural sciences and engineering disciplines such as physics, biology, earth science, meteorology, and electrical engineering as well as in the economics, sociology and political science. Physicists, engineers, computer scientists, and economists use mathematical models most comprehensively.

Eykhoff (1974) defined a **mathematical model** as a representation of the essential aspects of an existing system or a system to be constructed, which presents knowledge of that system in usable form.

Mathematical models can take many forms, including but not limited to dynamical systems, statistical models, differential equations, or game theoretic models. These and other types of models can overlap, with a given model involving a variety of abstract structures.

Normally when engineers analyze a system to be controlled or optimized, they use a mathematical model. In analysis, engineers can build a graphic model of the system as a hypothesis of how the system could work, or try to estimate how an unforeseeable event could affect the system. Similarly, in control of a system, engineers can try out different control approaches in simulations.

A mathematical model usually describes a system by a set of variables and a set of equations that establish associations between the variables. The values of the variables can be basically anything; real or integer numbers, Boolean values or strings, for example. The variables symbolize some properties of the system, for
example, measured system outputs often in the form of signals, timing data, counters, and event occurrence. The actual model is the set of functions that describe the relations between the different variables.

The question of whether the model describes well the properties of the system between data points is called interpolation, and the same question for events or data points outside the observed data is called extrapolation.

As an example of the typical limitations of the scope of a model, in evaluating Newtonian classical mechanics, we can note that Newton made his measurements without advanced equipment, so he could not measure properties of particles traveling at speeds close to the speed of light. Likewise, he did not measure the movements of molecules and other small particles, but macro particles only. It is then not surprising that his model does not extrapolate well into these domains, even though his model is quite sufficient for ordinary life physics.

1.2 Geometric Models

A geometric model describes the shape of a physical or mathematical object by means of geometric concepts. Geometric modeling is the construction or use of geometric models. Geometric models are used in computer graphics, computer-aided design and manufacturing, and many applied fields such as medical image processing.

Geometric models can be built for objects of any dimension in any geometric space. Both 2D and 3D geometric models are extensively used in computer graphics. 2D models are important in computer typography and technical drawing. 3D models are central to computer-aided design and manufacturing, and many applied technical fields such as geology and medical image processing.

Geometric models are usually distinguished from procedural and object-oriented models, which define the shape implicitly by an algorithm. They are also
contrasted with digital images and volumetric models; and with implicit mathematical models such as the zero set of an arbitrary polynomial. However, the distinction is often blurred: for instance, geometric shapes can be represented by objects; a digital image can be interpreted as a collection of colored squares; and geometric shapes such as circles are defined by implicit mathematical equations. Also, the modeling of fractal objects often requires a combination of geometric and procedural techniques.

In 3D computer graphics, **polygonal modeling** is an approach for modeling objects by representing or approximating their surfaces using polygons. Polygonal modeling is well suited to scanline rendering and is therefore the method of choice for **real-time computer graphics**. Alternate methods of representing 3D objects include NURBS surfaces, subdivision surfaces, and equation-based representations used in ray tracers. See polygon mesh for a description of how polygonal models are represented and stored.

![Elements of polygonal mesh modeling](image)

**Figure 1.1: Elements of polygonal mesh modeling.**

**Geometric theory and polygons**

The basic object used in mesh modeling is a vertex, a point in three-dimensional space. Two vertices connected by a straight line become an edge. Three vertices, connected to the each other by three edges, define a triangle, which is the
simplest polygon in Euclidean space. More complex polygons can be created out of multiple triangles, or as a single object with more than 3 vertices. Four sided polygons and triangles are the most common shapes used in polygonal modeling. A group of polygons, connected to each other by shared vertices, is generally referred to as an element. Each of the polygons making up an element is called a face.

In Euclidean geometry, any three non-collinear points determine a plane. For this reason, triangles always occupy a single plane. This is not necessarily true of more complex polygons, however. The flat nature of triangles makes it simple to determine their surface normal, a three-dimensional vector perpendicular to the triangle's edges. Some rendering systems use vertex normals instead of surface normals to create a better-looking lighting system at the cost of more processing. Note that every triangle has two surface normals, which face away from each other. In many systems only one of these normals is considered valid – the other side of the polygon is referred to as a backface, and can be made visible or invisible depending on the programmer's desires.

Many modeling programs do not strictly implement geometric theory; for example, it is possible for two vertices to have two distinct edges connecting them, occupying the exact same spatial location. It is also possible for two vertices to exist at the same spatial coordinates, or two faces to exist at the same location. Situations such as these are usually not desired and many packages support an auto-cleanup function. If auto-cleanup is not present, however, they must be deleted manually.

A group of polygons, which are connected together by shared vertices, is referred to as a mesh. In order for a mesh to appear attractive when rendered, it is desirable that it be non-self-intersecting, meaning that no edge passes through a polygon. Another way of looking at this is that the mesh cannot pierce itself. It is also
desirable that the mesh does not contain any errors such as doubled vertices, edges, or faces. For some purposes it is important that the mesh be a manifold — that is, that it does not contain holes or singularities (locations where two distinct sections of the mesh are connected by a single vertex).

There are many disadvantages representing an object using polygons. Polygons are incapable of accurately representing curved surfaces, so a large number of them must be used to approximate curves in a visually appealing manner. The use of complex models has a cost in lowered speed. In scanline conversion, each polygon must be converted and displayed, regardless of size, and there are frequently a large number of models on the screen at any given time. Often, programmers must use multiple models at varying levels of detail to represent the same object in order to cut down on the number of polygons being rendered.

The main advantage of polygons is that they are faster than other representations. While a modern graphics card can show a highly detailed scene at a frame rate of 60 frames per second or higher, ray tracers, the main way of displaying non-polygonal models, are incapable of achieving an interactive frame rate (10 frame/s or higher) with a similar amount of detail.

The area of geometric modeling is quite broad, some of the common representation of geometric models are, polygon mesh surfaces, parametric surfaces, quadric surfaces and particle-based methods. We choose the polygon model for the following need and the idea to use of parametric surfaces is discussed in the future scope. We have also discussed brief description of particle systems, which is used in section 4.2.
1.2.1 Need for polygon Model:

With reference to the above discussion a polygon mesh is a set of connected polygonal bounded planar surfaces. Open box cabinets and building exteriors can be easily and naturally represented by polygon meshes, as can volumes bounded by planar surfaces. Polygon meshes can be used, although less easily, to represent objects with curved surfaces; however the representation is only approximate. The obvious errors in the representation can be made arbitrarily small by using more and more polygons (increasing the resolution of meshes) to create a better piecewise linear approximation, but this increases space requirements and the execution time of algorithms processing the representation. Furthermore, if the image is scaled up (enlarged), the straight edges again become obvious, similar to the aliasing problem.

However, in our work similar discretization of the model in time and space arises in the function approximation part of the simulation part (FEM). To achieve a real time simulation and visualization it is required to integrate both the geometrical model as well as simulation model. In this context we found finite element mesh in computational domain which is almost similar to the polygon mesh in computer graphics domain (wire frame model) and hence it is easy to plug-in or merge both of them together for data transformation as well as mapping. So we choose polygon mesh a suitable for our framework to start with.

1.2.2 Particle Systems:

Particle systems have been normally applied for modeling fuzzy objects that do not show deterministic shapes. Fluids with Breakable surfaces are distinctive examples of such objects. Particle systems are helpful because particles are simpler primitives to implement than polygons, they can be generated and controlled by procedural methods, and particle systems permit easy modeling of dynamic behavior.
Simple particles: It is usual practice to refer to representation of a particle by its initial location, velocity and other individual properties as simple particles, as against, grouping of particles and giving them collective descriptions. Simple motion blurred particles have been used to model explosions [1.2], fireworks [1.1], waterfalls [1.3], and fire. Particle systems have the advantage that can be rendered by both polygonal techniques and volume rendering approaches.

Surface particles: An example of particles being associated with information, which allows for photo realistic shading, is fluid flow visualization using surface particles [1.4]. A surface particle is a small facet. The shape of the facet is irrelevant as it is minute. The particle is represented as a point with a normal.

1.3 Simulation Modeling:

Simulation is the imitation of some real thing, state of affairs, or process. The act of simulating something generally entails representing certain key characteristics or behaviours of a selected physical or abstract system.

Simulation is used in many contexts, including the modeling of natural systems or human systems in order to gain insight into their functioning. Other contexts include simulation of technology for performance optimization, safety engineering, testing, training and education. Simulation can be used to show the eventual real effects of alternative conditions and courses of action.

A computer simulation is an attempt to model a real-life or theoretical situation on a computer so that it can be studied to see how the system works. By changing variables, predictions may be made about the behaviour of the system.

Computer simulation has become a useful part of modeling many natural systems in physics, chemistry and biology, and human systems in economics and social science as well as in engineering to gain insight into the operation of those
systems. A good example of the usefulness of using computers to simulate can be found in the field of network traffic simulation. In such simulations, the model behaviour will change each simulation according to the set of initial parameters assumed for the environment.

Traditionally, the formal modeling of systems has been via a mathematical model, which attempts to find analytical solutions enabling the prediction of the behaviour of the system from a set of parameters and initial conditions. Computer simulation is often used as an adjunct to, or substitution for, modeling systems for which simple closed form analytic solutions are not possible. There are many different types of computer simulation; the common feature they all share is the attempt to generate a sample of representative scenarios for a model in which a complete enumeration of all possible states would be prohibitive or impossible.

Several software packages exist for running computer-based simulation modeling (e.g. Monte Carlo simulation and stochastic modeling) that makes the modeling almost effortless. But the limitation of software is a constraint for research scientists. In this context we have designed and developed even system (chapter 2) as well as our application (chapter 4) to overcome such constraints of the commercial packages.

On basis of these models discussed above, many mathematicians like C.Truesdell, R.P.Kanwal, Indrasena, G.Purushotham, C.S.Bagewadi, K.N.Prasannakumar, Siddabasappa, A.N.Shantharajappa and B.J.Gireesha have obtained the both numerical and analytical solutions for flow problems through mathematical modeling. Such as C.S. Bagewadi and Girisha B.J have extended the study to two-dimensional steady dusty gas flow [1.8-1.13] in Frenet Frame Field System and devoted to study the geometry of compressible/ incompressible flow. Mathematical models are constructed and the study is analysed for various parameters
like steady/unsteady, dusty gas/dusty fluid flow under varying pressure gradient [1.9-1.10], Temperature [1.11-1.12] and Transverse magnetic field [1.13].

By keeping the above mathematical models as basis, we started our work to extend it into the area of computer graphics. Finally we have focused our work to visualize the fluid flow by using geometrical models such as polygon mesh model and particle system to extend the work for the simulation of flow, like, flow through pipes along with fluid interaction, simulation of free surface flow for virtual reality applications based on scientific visualization and computer Haptics. The requirements to implement the application are geometric models (from the context of computer graphics) and simulation model that are suitable to visualize the same (with time dimension) are focused in real time are introduced in this section.

In our case the target problem is formulated by discretizing the model and further processing it through the finite element method, which results in a large system of simultaneous ordinary differential equations. By solving these equations, we have designed the models, which simulate the dynamics at each step. The details of mathematical models and the computational algorithms/techniques are discussed in chapter 3. Now we give a brief importance about the need for studying Real Time Interactive Simulation and Visualization

1.4 Need for Real time Interactive Simulation & Visualization

Some of the studies [1.5] apply computer graphics to predict situations based on simulation, for example, problems encountered in the fire research, including study of flame structure, interior fires, and visibility in fires, simulation of battlefield plumes, study of visibility in foggy conditions and simulation of explosions. Similarly in the field of medical, virtual surgery simulators are becoming an important tool in the education of surgeons and medicine students. Techniques and procedures can be
practiced in a virtual environment before conducting the corresponding operations on real human beings. One of the most important applications is simulation of blood flow (fluid flow) and its interactions during surgery [1.7]

Thus, the ubiquity of fluids in real world scenes, the continuous quest for greater realism of computer generated images and expanding scope of problems addressed by computer graphics simulations, make investigation of techniques to model and simulate fluid flow important.

Further we briefly describe some cases in computer graphics where fluid modeling simulation and real time interactive visualization has been applied.

1.5 Applications of interactive fluid simulation and visualization

Apart from the application examples discussed in chapter 4 we focus on the application that is discussed in the article [1.7].

The medical domain provides excellent opportunities for the application of computer graphics, visualization and virtual environments, with the potential to help improve healthcare and bring benefits to patients. Survey paper of F.P.Vidal, et.al.[1.6] provides a comprehensive overview of the state-of-the-art in this exciting field. It has been written from the perspective of both computer scientists and practicing clinicians and documents past and current successes together with the challenges that lie ahead. The article begins with a description of the software algorithms and techniques that allow visualization of and interaction with medical data.

Medical education program involves teaching and laboratory courses for the students. The students to be trained in basics of surgery with hands on training in tissue cutting, suturing, etc. usually this is done with mannequins. To do this effectively there should be a device where in the students can be trained with virtual
tools such as haptic devices. This haptic device measures all the physiological functions where the surgery is to be done and arrives at conclusive force required to penetrate particular regions of the tissues, bones etc. This device will be assisted with experts systems, so that, best of the practices can be recorded to the system and the system would learn through such experiences (cognition) and can assist novice surgeons during the course of training (recognition). This device will also be trained with actual images (MRI, FMRI, 2D, 3D, CT) prior to the surgery; the patient safety is taken at most care because the images (3D) give the better control to the surgeons.

In the literature many surgical simulators have been developed and used, but all provide interactions (but not physically realistic response). Here [1.7], with “Real Time Virtual Surgical Planning Using Computational Steering (RTVSPUCS)”, we introduce computational steering in the system, which provides the system to be interactive in real time. The system would be promising since surgeons would have the look and feel of what he would be thinking about. In short, the system would give the complete experience of real surgery with the use of force feedback, Haptics devices, and patient specific 3D model with some physical and tissue properties. In future, the system would be also used to test surgical skills of the surgeon, for conducting the desired surgery.

Since it is interdisciplinary area, lot of work is going on as an independent area. But, when it comes to the integration part, to synchronize all the information, an extensive information fusion work need has to be done. Real-time issues also arise to get physical realistic responses for the interaction in virtual environment. These researches are not found in the literature as yet.

Hence, we are proposing to build a generic system for interactive scientific visualization based on computational steering and it is discussed in the Chapter 2
(section 2.3, of Dynamic System Modeling). As an application part to this system model fluid flow models have been used as a test case in simulation part, interrelated to the blood flow in virtual surgery.

References


1.9 C.S Bagewadi and Girisha. BJ “A study of Two Dimensional Steady Dusty Fluid Flow under Varying Pressure Gradient” Communicated to Mathematical Proceedings of the Royal Academy, IRELAND.


1.11 C.S Bagewadi and Girisha. BJ “A Study of Two dimensional Steady Dusty Fluid Flow under Varying Temperature” Communicated to Tensor N.S. Japan.
