MATHEMATICAL MODELING: THE FOUR SEASONS

Paul Bouthellier
Department of Mathematics and Computer Science
University of Pittsburgh-Titusville
Titusville, PA 16354
pbouthe@pitt.edu

Computer graphics and programming can be used to illustrate mathematical concepts from junior high school through graduate school. In this paper we shall look at modeling and animating 3D objects such as trees, leafs, fireworks, lightning, pumpkins, and snow.

Winter will be illustrated by a 3-dimensional snowfall. Spring shall be represented by a tree and its leaves. Summer will be represented by a lightning storm and Fourth of July fireworks. Fall is illustrated by: falling leafs, Halloween pumpkins, and shooting a pumpkin out of an air cannon.

Some of the mathematical concepts required in our results are: translations (2D and 3D), projections, and rotations in 3D via rotation matrices and quaternions, Boolean adds, rotation of cameras (in space and about their axes), Bezier curves and surfaces, scaling, Euler angles, quaternions, and Gimbal lock, extrusions into 3D, interpolation, coordinate systems (local, global, camera, clipping, object-and mapping from one to another), spherical and cylindrical coordinate systems, normal and dot-products, and projecting images onto surfaces

The packages used in this project were: Studio 3D Max, Maya, Cinema 4D, Poser, Swift3D, Carrara, Paint Shop Pro, Photoshop, Fireworks, Swish Max, and Flash. Programming was done in Python, MaxScript, and ActionScript.

One thing I would like to note: Even though it takes years to become proficient (or even half-way decent) at creating 3D graphics and the required programming, it only takes a few days (or even hours) to learn enough basics to illustrate the mathematical concepts we may need in class.

Figure 1-Thoughts of Spring

Winter

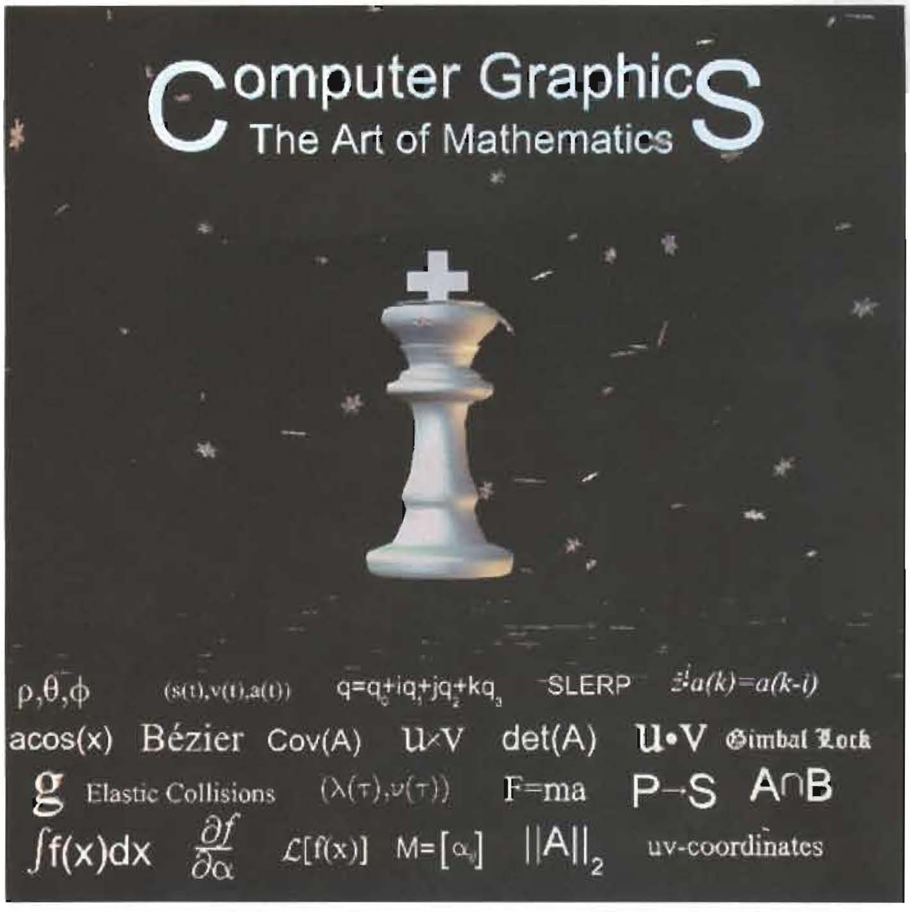

Figure 2 (Poser/Swift3/Python)

In our mathematics classes we encourage our students to break hard problems into simpler pieces; this can be illustrated by trying to create any 3D object. Almost any manufactured object consists of many small pieces which are then put together by some process.

For the above snow fall (Figure 2):

- Create a snowflake using Bezier curves then extrude it into R^3 (shown in Figure 3)
- Use trigonometry and parametric curves to move the flakes in R^3
- Rotate each snowflake using Euler angles or quaternions
- Randomly scale each snow flake
- Create the illusion of the snow hitting the ground by using projections of the flakes onto the ground
- The chess piece was created as follows:
    - Create the outline of the body using a Bezier curve and rotating the curve about the vertical axis
    - Create the "crown" using Bezier curves and extruding them
    - Using a Boolean add to join the geometries

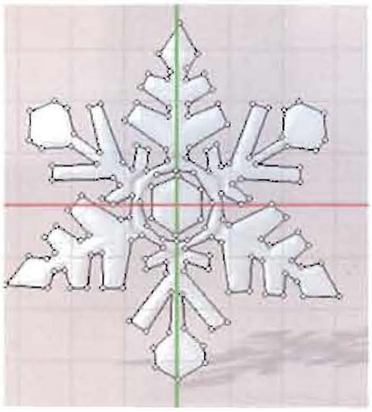

Figure 3 (Swift 3D)

Spring

Figure 4 (Carrara)

Figure 5 (Carrara/3DMax/Poser)

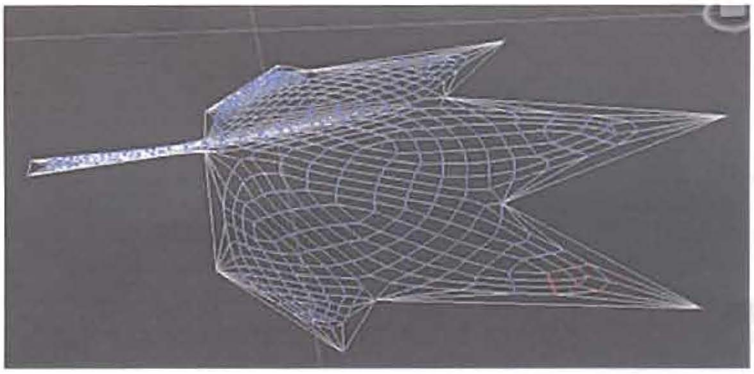

Figure 6 (Studio 3D Max)

Spring begins with bare trees. Leafs are created and then grown on the trees. These are created as follows:

- The tree in Figure 4 was created in Carrara. By placing each branch at the Fibonacci angle of 137.5° from the branch above it, we get a very realistic tree
- Splines were used to create the 3D leafs illustrated in Figure 6
- The leaves were placed on the tree by modeling the tree as a cylinder and placing the leaves randomly using cylindrical coordinates (Figure 5)
- The leaves were randomly oriented using Euler angles and scaled from 0 to full size to create an animation of the leaves growing on the trees.

Summer


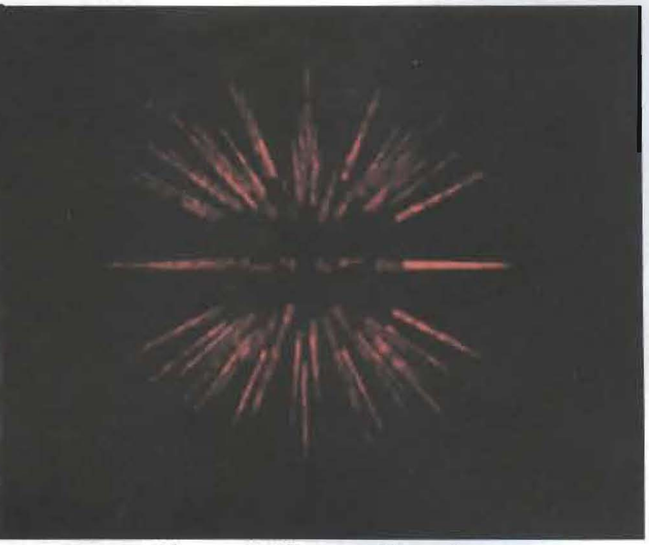

Figure 7 (Paint Shop Pro-KPT Filters) Figure 8 (Carrara/Poser)

The lightning storm is nothing more than a branching algorithm. While most algorithms create 2D bolts, it is easy to create them in R^3 as well. Once created they can be viewed from any point of view in a 3D package.

A 4th of July fireworks can be created as follows:

- Take any 3D object and create their normal vectors (pointing outwards)
- Remove the original geometric shape
- Texture the normal vectors so that they will be visible against a dark background
- Scale the normal vectors over time to create the exploding fireworks.

Fall

Figure 9 (Carrara/3D Max/Poser)

To create the leaves as shown in Figure 9:

- Interpolate the hexadecimal code for the colors of the leaves. Begin with green and end with a variety of yellows, golds, and purples
- Use simple parametric curves to create the falling leafs (Figure 10)

Figure 10 (Swift3D/Flash)

Fall (Halloween)

As part of Fall we celebrate Halloween.

- First we can create a pumpkin using splines, Bezier curves, and Boolean additions (as shown in Figure 11)

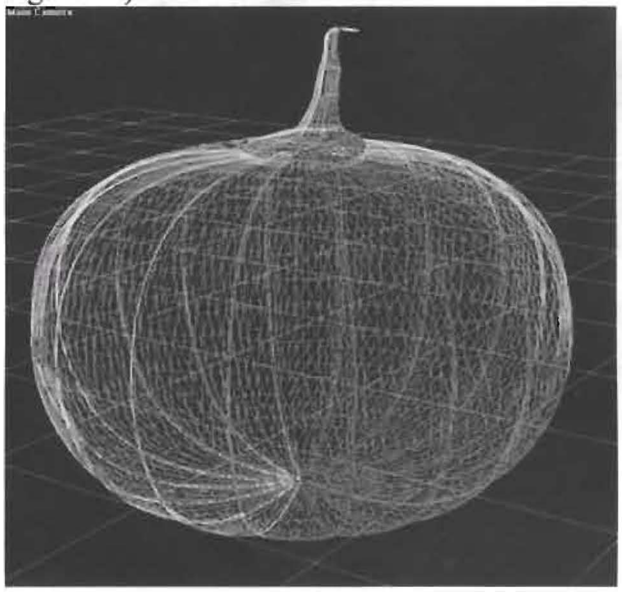

Figure 11 (Carrara/3D Max)

- Using texturing and creating instances, an animated pumpkin screensaver can be created using simple trigonometry and Python.

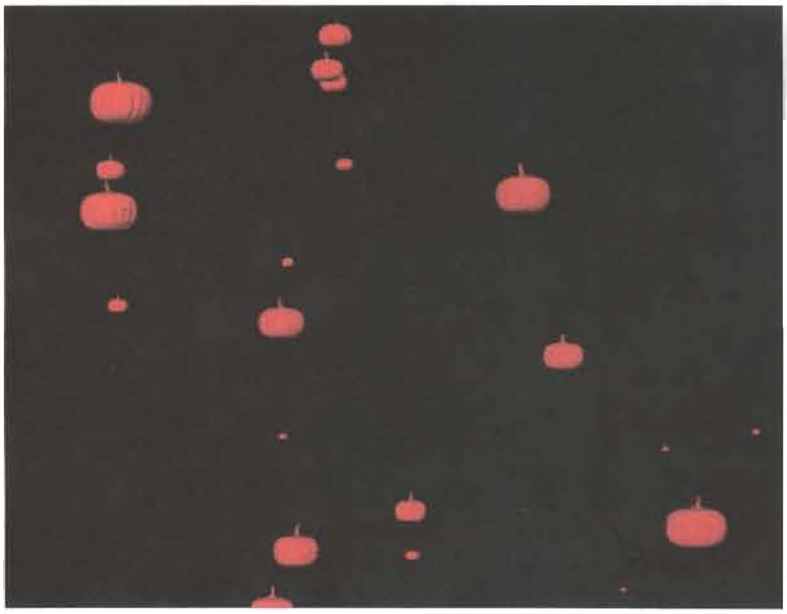

Figure 12 (Carrara/Poser)

Pumpkin chunkin is a contest about throwing pumpkins via mechanical devices (such as air guns) with the goal of maximizing distance. To model such a contest:

- First a pumpkin is created as discussed above
- To get the mass and cross-sectional area of the pumpkin-needed for the flight equations-the vertices of the model can be used to estimate the needed values. (Figure 13 shows using the vertices to estimate the surface area of a sphere in Studio 3D Max).

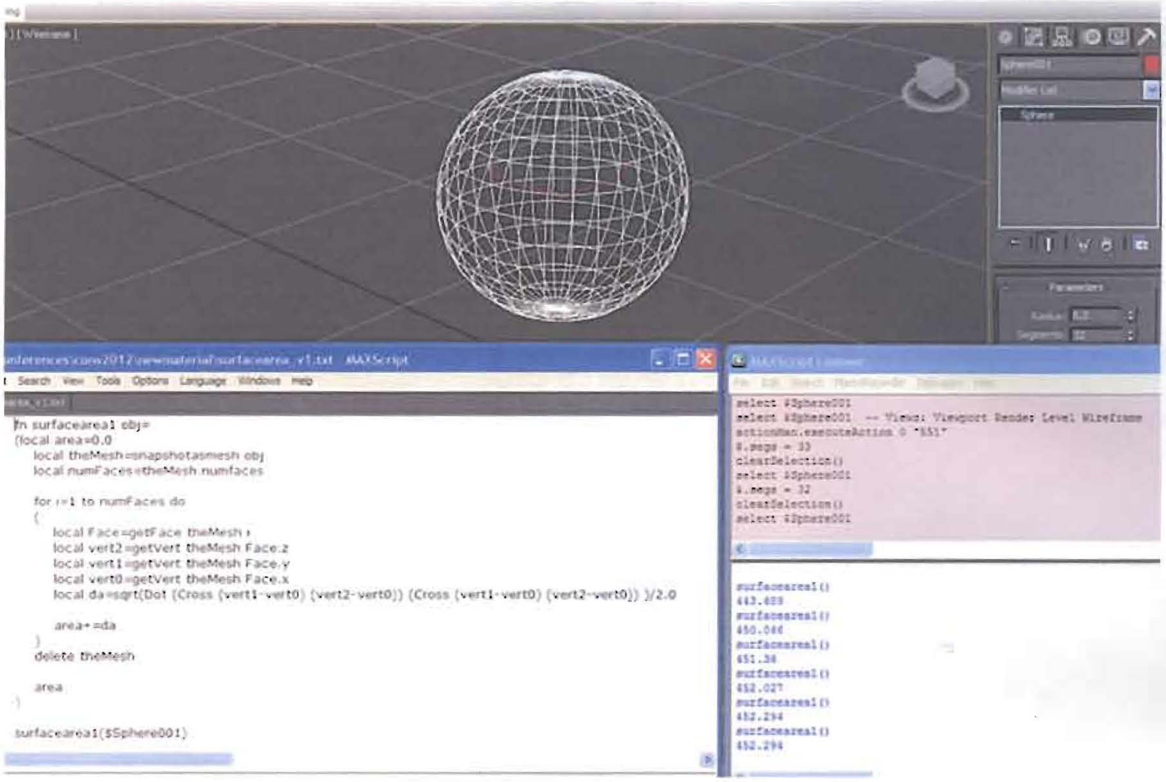

Figure 13 (Studio 3D Max)

- A system of differential equations are derived to take into account:
    - The speed, angle, and vector of the initial launch
    - The mass of the pumpkin
    - The cross-sectional area and the effects of air-resistance
    - The effects of the wind on the flight
    - Approximating the solutions of the equations by either Euler's method or by a Runge-Kutta method. (Figure 14 shows a still from such an animated flight.)

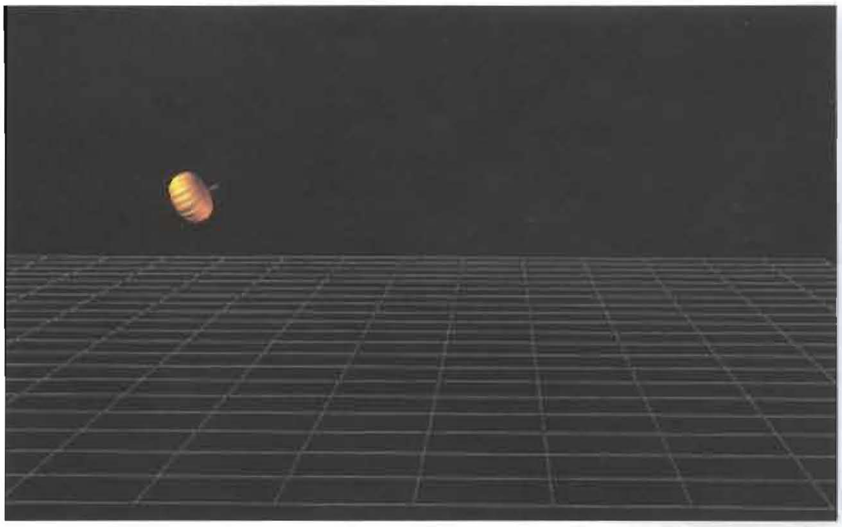

Figure 14 (Carrara/Poser)

Such an animation can be viewed from any position and angle in 3-dimensional space.