Dynamics and Particle Effects, Part 1

By Audri Phillips

From their very inception, 3D programs have been used to imitate natural phenomena, creating realistic, stylized, or artistic effects. A greater range of effects is now possible than in the past, before the advent of computer technology. The dynamics/particle effects component of 3D software combined with compositing programs—many of which also have particle capabilities—have primarily been used to create these effects, which have become a mainstay in film, commercial, and game production.

Artists are also using them in the creation of personal artwork. I use them in my own “postexperimental animations.” A lot of the dynamics code has been written using real-life physics or “Newtonian mechanics” as a model. Artists have had to learn to use this more science-related terminology in the software interfaces. In this article, I touch upon this trend and provide hands-on demonstrations of the technical and artistic methods used to achieve some of these effects.

I teach a class in animation at a small university in California, and the first effect I usually teach my students how to create is smoke. I ask my students to look at smoke in real life, photos, and animations. Visual observation can be a powerful aid. Even an image search on google.com can be helpful. There are many stylized versions of smoke as well as naturalistic images. Stylized effects are used in all types of media. I ask my students to study how smoke moves, the color of it, the opacity of it, how the look of smoke changes from where it was first emitted to where it goes.

Basics of Particles in Autodesk* Maya*

For the purposes of this article, consider a particle to be an object of negligible size. Particles are essentially points in space: They can each have their own sets of properties that influence their behavior as well as look.

There is a selection of particle types you can create in Autodesk* Maya*. Many of the particle types, such as clouds, that you will be using to create your smoke effect are, when first created, assigned to a particle cloud material by default. Different types of particles also require different renderers in Maya*. Although particle types such as points and streaks need to be hardware rendered when using the Maya* renderer, when using the mental ray* plug-in that comes with Maya*, they can be rendered in a software render.

Creating Smoke

For this article, you are going to be working in Maya*. To start, go into the dynamics module and create an emitter. You can create a variety of emitters, including omni, directional, and volume emitters, as well as emit particles from an object. In this case, you’re just selecting the emitter type, omni. Set the rate of the particles to 100 per second and the emission speed to 0, as Figure 1 shows.
Figure 1. Once the emitter is created, the initial values you set can always be changed in the attribute editor of the emitter. You can set key frames for values like rate and speed.

Next, pick the particles you just created and, in the smoke attribute editor (particleShape1), choose cloud as the particle type. From the Lifespan Mode drop-down list, choose lifespanPP only (see Figure 2). If, for example, you chose constant instead of the lifespan mode, and then put a value in the lifespan box underneath, all your particles would live the same amount of seconds. If you entered 1 into the lifespan box, each particle would live for 24 frames after it was born.

Figure 2. I have selected the particleShape1 tab at the top of the attribute editor. The particle shape is where most of the attributes you want to set are found. Per particle (PP) means that each particle in a particle object can have its own value for any attribute.
Per-object attributes give one attribute value to the whole particle object. Here, you choose lifespanPP (lifespan per particle) because you want to vary how long the individual smoke particles live. This setting provides a more natural, randomized look to your smoke.

Now, look at the Per Particle(Array) Attributes section of the particle attribute editor. Figure 3 shows this section of the editor.

**Figure 3.** The attributes with the red dots are those you’ll be using to create smoke. There are three other PP attributes not on this list that you will be adding: Color (RGB PP), Opacity PP, and Radius Size (Radius PP).
Creating the Missing Radius PP, Opacity PP, and RGB PP Attributes

To create a radius size per particle (Radius PP), in the Add Dynamic Attributes window, click the General tab, as Figure 4 shows.

![Figure 4](image)

You can add a number of interesting attributes. Try out the Rotation PP, with which individual particles can be made to spin differently.

When the Add Attribute window pops up, click the Particle tab. There, you see a list of attributes you can add. Select radiusPP; a Radius PP tab will appear in your particle attributes editor.

To add an RGB PP attribute, under Add Dynamic Attributes, click the Color tab. In the window that pops up, select the Add Per Particle Attribute option, and then click Add Attribute. Similarly, to add an Opacity PP attribute, under Add Dynamic Attributes, click the Opacity tab. In the window that pops up, select the Add Per Particle Attribute option, and then click Add Attribute, as shown in Figure 5.

![Figure 5](image)

If you selected the Add Per Object Attribute option, the whole particle object would have the same opacity.
Adding Expressions

Next, you will add expressions that change the size of the particles and how long they live. So, the numbers you choose to put in the expressions can greatly vary the way your smoke animation looks.

First, add a creation expression to the Lifespan PP. To do this, click Lifespan PP in the attribute editor. From the Lifespan Mode drop-down list, choose Create New Expression (see Figure 6). In the expression editor, type:

```
particleShape1.lifespanPP= rand(4,6);
```

This expression controls how long the individual particles live. The particles are randomly assigned a lifespan somewhere between 4 and 6 seconds when they are born.

![Figure 6](image)

**Figure 6.** I have added an expression for lifetimePP. If I had wanted the particles all to live the same amount of time, I could have written an expression with only one value for the life of the particle, such as `particleShape1.lifespanPP =4;` Then, all of my particles would have lived for 4 seconds after they were born. Also, notice that Creation option is selected in the expression editor.

**Expressions for RadiusPP**

Here are a few expressions you can write to control the radius size of the particles. First, you can write a simple creation expression for the radius PP of the individual particles:

```
particleShape1.radiusPP= rand(.5,1);
```
This expression randomly assigns a radius size between 0.5 and 1 for every particle born.

Another option for assigning a radius is to add two expressions—one a creation expression, and the other a runtime before dynamics expression. First, add the creation expression for the radius PP of the individual particles:

```plaintext
particleShape1.radiusPP=.5;
```

The particles will be created at 0.5 (of a unit) size. Then, add a runtime before dynamics expression:

```plaintext
particleShape1.radiusPP = .5 + (2*linstep(0, particleShape1.lifespanPP, particleShape1.age));
```

The above expression gives the effect of the radius of the particle starting at 0.5, increasing in size as the age of the particle increases. Generally, referencing smoke in real life, the smoke increases in size as it goes farther from the source. In this case, I have made sure that the creation expression also starts out with the particles at a 0.5 radius size.

**Note:** `linstep` and `smoothstep` are built-in functions. Maya® enters a value between 0 and 1 over a specified range. `Smoothstep` gives you an ease in and out: `Linestep(start, end, unit parameter)`. 

### The Simpler Solution: Adding a Ramp Instead of an Expression

Ramps are simple and easy to use, and although they may not give you as many options as using an expression, they often serve the purpose well. So, here is the ramp way to assign a radius size to particles. Instead of picking an expression in the drop-down list, choose **Create Ramp**, as Figure 7 shows.

![Image of Create Ramp](image)

**Figure 7.** Many times, ramps are a simple and intuitive way to change an attribute’s value over a particle’s lifetime. The particle starts its life at the bottom of the ramp and dies at the top. Of course, if you choose for a particle to live forever, it will keep the value at the bottom of the ramp.

If you want to edit the ramp, click inside the Radius PP box (which now has the words **arrayMapper** inside it), and choose **Edit Ramp** (see Figure 8 and Figure 9). Anytime you want to get rid of a ramp or expression from an attribute, you can always click **Break Connection**.
Figure 8. When you click Edit Ramp, the ramp comes up and you will be able to edit the values of the colors on the ramp. The color values give you the numbers used, in this case, to dictate the radius size (Radius PP) of the particles.

Figure 9. Because the last value on this ramp—2—is about 3/4 of the way up, the particles get a radius of 2 at that point in their lives and hold that value until they die.

Note: Remember that in Maya®, the value of white (or any color) can be increased beyond 1. This is particularly useful when you want the final particle radius size to be 2.

In the ramp shown in Figure 10, the particles are born with a radius of size 0.125 and die with a radius of 1.5.
Figure 10. The birth and death radius for the example ramp. As you can see from the red numbers, each particle starts with a radius of 0.125 and ends life with a radius of 1.5.

Figure 11 provides comparisons of values created for the Radius PP attribute by changing ramp values.

Figure 11. You can see how the values you put into the ramp can radically change the look of your smoke animation.

Finally, create another ramp for the **Ramp Acceleration** attribute. This attribute controls the acceleration rate of the particles. Start with black at the bottom of your ramp (0 acceleration), and end with a value around 0.3, as shown in Figure 12. Acceleration requires three values (vectors). These values are taken from the RGB values on the ramp. The values on this ramp are gray, so the particles are accelerating equally in all directions.
Adding Fields to Particles

You can also influence the movement of the particles by adding fields to them. Particles can be keyframed or controlled dynamically. Dynamics in Maya* are created using the rules of physics, and you use these rules to create the different fields. An obvious example is the field called gravity. Connect two different fields to your particles: a vortex field and a turbulence field. To do that, select the particles, and then select the field you want from the drop-down list under the Fields tab on the top bar of the main Maya* window (see Figure 13).

Figure 12. A ramp for the Ramp Acceleration attribute.

Figure 13. I change the magnitude of the vortex field to 7, and the vortex rotates around the Y-axis. I also create a turbulence field and give it a magnitude of 0.1.
Figure 14 shows the image with both fields added.

![Image of vortex and turbulence field]

**Figure 14.** A vortex and turbulence field have been added.

*Note:* You can change the settings of the fields to provide the type of smoke movement you would like. If it is a windy day, you might want to use a field such as uniform, which would pull the particles along the axis of your choice.

**Particle Looks**

By default, when the cloud particles are created, they are assigned a Maya* particle cloud shader. I recommend selecting the particles, creating a new particle cloud shader, and assigning that shader to the particles.

The next thing you want to change is the opacity of the particles over time. You do this the same way you did the radius, adding a ramp to the Opacity PP (see Figure 15). In this case, you want your particles to be more opaque when they are born and totally transparent when they die—the exact inverse of how you set your values for the radius. This is how smoke generally acts in real life. In addition, because your particles are becoming more transparent as they age, you won’t have to worry about your particles suddenly popping off the frame.
Figure 15. The ramp for Radius PP

Because earlier you assigned your particles a bit of randomness as to how long they live (see the expression put into the Lifespan PP), they will have variation as to their transparency and radius. For example, a particle with a lifespan of 1 will turn transparent earlier than a particle with a lifespan of 2. So, if the two particles were born at the same time but have different life spans, they will increasingly look different as they age. To put it another way, the particle with a lifespan of 1 will reach the top of the ramp twice as fast as the particle with a lifespan of 2. Remember: When a particle is born, it has the values at the bottom of the ramp. As it ages, it travels up the ramp, taking on the new values. At the end of its life, it will reach the top of the ramp.

Next, pick the colors of your particles: the rgbPP. You create a ramp for that, as well (see Figure 16).
Dynamics and Particle Effects

Figure 16. Different types of smoke will obviously have different colors. In this ramp, my colors start a bit warm, and then graduate to black as the smoke dies.

Test Render

To let the per-particle attributes of color and opacity control the opacity and color of the particles, you must disconnect the transparency and the color of the particle cloud material from the particles. Then, the look you assigned the particles in the per-particle attributes show up. Until then, the particles will have the color and opacity of the particle cloud material (see Figure 17).

Figure 17. Here is a first test software render. Notice that it does not look the same as it looks in the hardware window.

It doesn't look very impressive. And why is it blue? What happened to the color and opacity settings?

The color and opacity settings for the smoke are still being retrieved from the particle cloud shader, as shown in Figure 18.
**Figure 18.** This is the particle cloud material that, by default, your cloud particles were assigned to. The look of this shader can be completely changed, or you can choose to disconnect certain attributes from this material and give them to the particle attribute editor to control.

**Disconnecting the Color and Transparency from the Particle Cloud Material**

To disconnect either the color or opacity (or both) from the particle cloud so that the settings you put into the per-particle (array) attributes can take over, you must first go into the hypershade and select the particle cloud **Material**. In the Material particle cloud, you must connect (map) both the opacity and the color to the particle sampler found in the utilities, as shown in Figure 19.
Figure 19. *When you map on the particle sampler, note that the particle cloud material changes from blue to black.*

Figure 20 shows the shader network when you have connected the color and transparency of the particle cloud shader to the particle sampler found in the utility section (see the left side).
Figure 20. Notice that the particle cloud shader is black. Now, the color and transparency will be controlled by the settings in the per-particle attributes in the attribute editor of the particle shape. See the right side.

Now, try another render (see Figure 21).
Figure 21. The image on the left is the render of the smoke. The image on the right is with the addition of a solid fractal texture mapped into the blob map attribute of the particle cloud Material.

Using the Shader to Create the Color and Transparency of the Particle Effect

Another option for creating the look of your smoke is, instead of disconnecting the color and transparency (opacity) from the particle cloud material, to use them to create the look of your smoke. Under Transparency in the particle cloud shader (material), there is also a blob map, roundness, and density setting that you can map to help change the round shape of the cloud particles (see Figure 22). I often use a combination of per-particle attributes and settings from the particle cloud shader.
Figure 22. The shader network I set up for the smoke. I have mapped a solid fractal onto both the color and blob map. The transparency has a middle grey value on it. Note that the particle cloud is not connected to the particle sampler. The shader is in its original state, so the shader can control transparency and color, not the per-particle attributes found in the attribute editor of the particles.

Figure 23 shows the smoke render using the transparency and color attributes of the particle cloud material.
Figure 23. *Note that unless you animate the solid fractal texture at about the same rate the particles are moving, it will look like the particles are traveling through the texture.*

In Figure 24, I created a hot lava look using a simple material setup for the color and transparency of my cloud particles.

Figure 24. *I mapped a ramp with multiple colors and a bit of noise onto the color of the particle cloud material and a ramp of different grey values onto the particle cloud transparency. This is a good example of how using some of the particle cloud attributes pays off. I also added a touch of glow to the material.*
Reusing Your Particles by Having Them Come Out of Other Emitters

To emit your particles from another emitter, you create a new emitter, and then—in the Dynamics Relationship window, which can be found under Window > Relationship Editors > Dynamic relationships—you can assign or unassign any particles you want to it (see Figure 25). You can assign and unassign forces in the same way.

Figure 25. Here, I have emitted the smoke particles I created from an emitter parented to the rocket so that it travels with the rocket.

Conclusion

By going through this smoke tutorial and creating smoke, you learned some of the basics for setting up particle effects in Maya* and can go on to trying other designs and effects. And oh yes: There is a Create Smoke effect under Dynamics that you might try, but it depends on sprites—a type of particle we haven’t talked about. They always face toward the camera and require an image or images to map them. You will also be presented with a series of options as to forces to use. But you still do not have the same flexibility as you do when you start creating your own effects from scratch. There are always many ways to create an effect in most good software programs.

About the Author

Audri Phillips is a Los Angeles-based artist currently working and exhibiting in a variety of media that range from computer animation and motion graphics to her more personal work, oil paintings and video art (visual poetry). She has worked for more than 20 years as an artist/digital artist in the entertainment industry in Los Angeles at such large studios as Disney Feature Animation, Sony Imageworks, DreamWorks Feature Animation, Rhythm and Hues, Digital Domain, and Electronic Arts as well as a number of small design boutiques. Her production experience includes art direction, design, visual development, storyboards, color and lighting, efx, layout, modeling, and compositing. Currently, she is an adjunct professor at Woodbury University. Check out her work at http://www.vimeo.com/audri/videos.