Part 2

How to Use dForce: Creating a Blanket, Draping Clothes on Furniture and Much More

Copyright 2017-18 by Rich Schafermeyer
(RGcincy on Daz Forums)

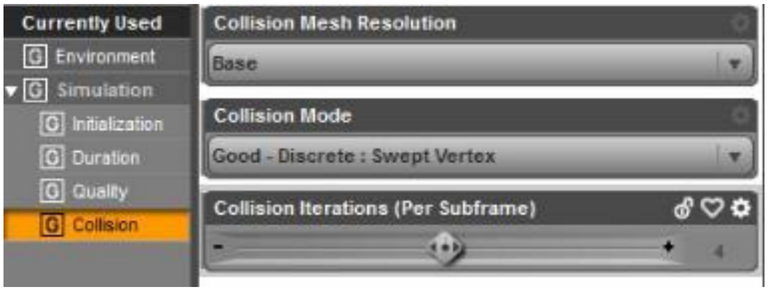
This is a summation of learnings made while trying out Daz Studio's new dForce. These do not involve draping clothes on figures but rather on other ways you can use dForce (such as draping blankets and towels or draping clothing on furniture and other clothing). Because of their simplicity, some of the elements made from primitives are helpful in illustrating what the various dForce global and surface parameters will do.

Part 2 Index

26. **Dropping an Object onto a Surface.** The question was asked on the Daz forums “If you want something to fall until it connects, but not to deform at all, what would you adjust?” I didn’t have an answer so I did some experimentation and got it to (mostly) work.

    a. Create two primitive cubes of 10 divisions each. Raise one so it starts in the air above the first and add a dForce dynamic modifier to it.

    b. Go to the Simulation Settings pane and set Collision mode to Good. This preserves the cube’s shape; otherwise, it begins to collapse in on itself.

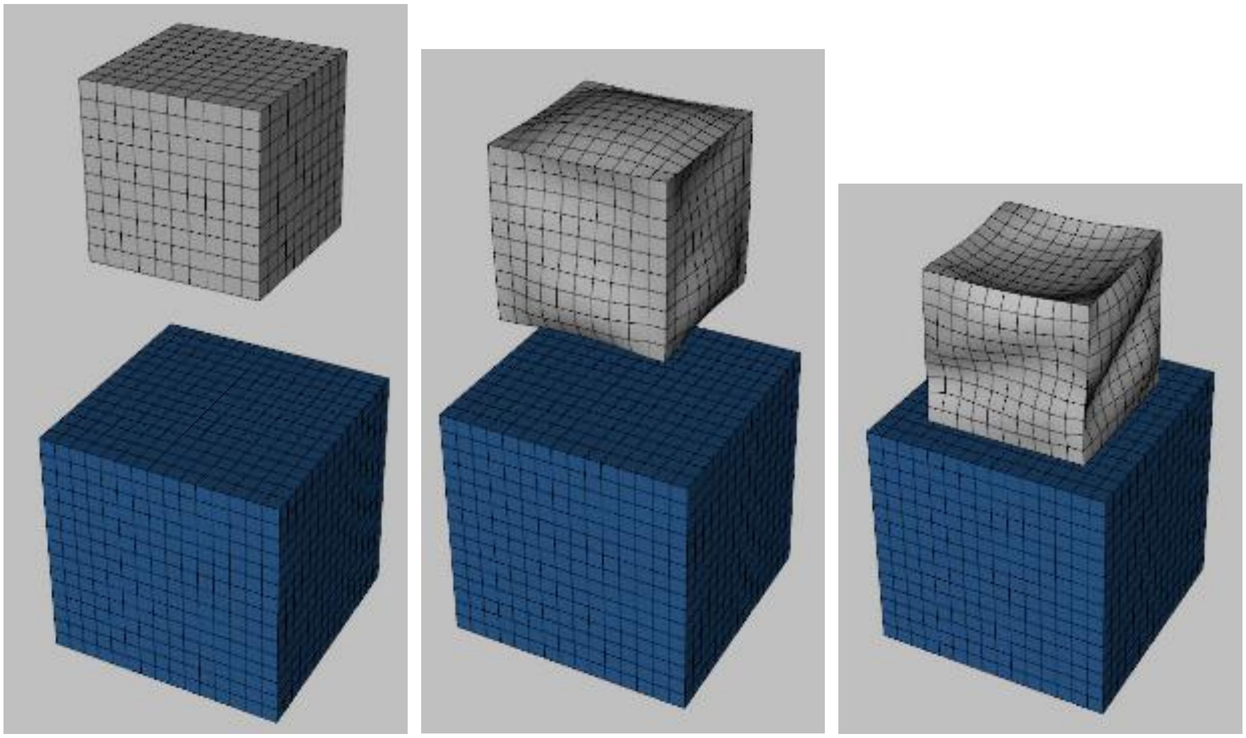


Collision Mesh Resolution
Base

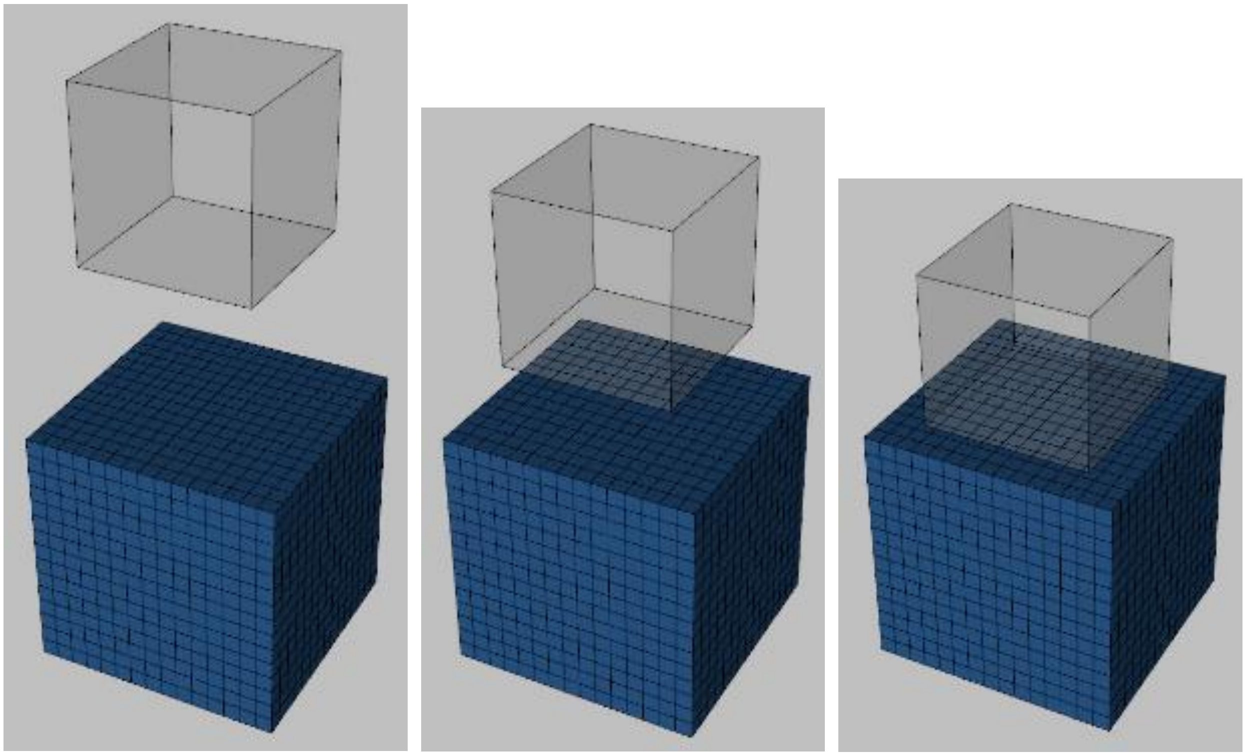
Collision Mode
Good - Discrete : Swept Vertex

Collision Iterations (Per Subframe)
4

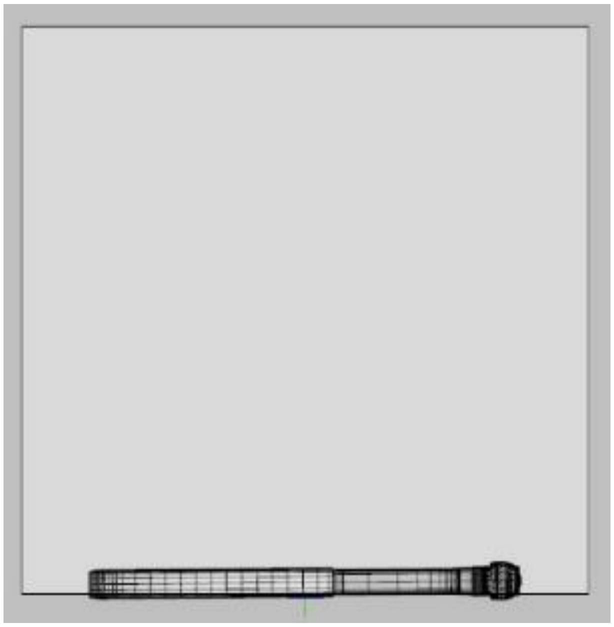
    c. Run an animated simulation. You’ll find that as soon as the simulation starts, the top cube not only falls but also begins to distort. It will land on the lower cube and stop dropping but will continue to distort, although it mostly retains the shape of a cube.



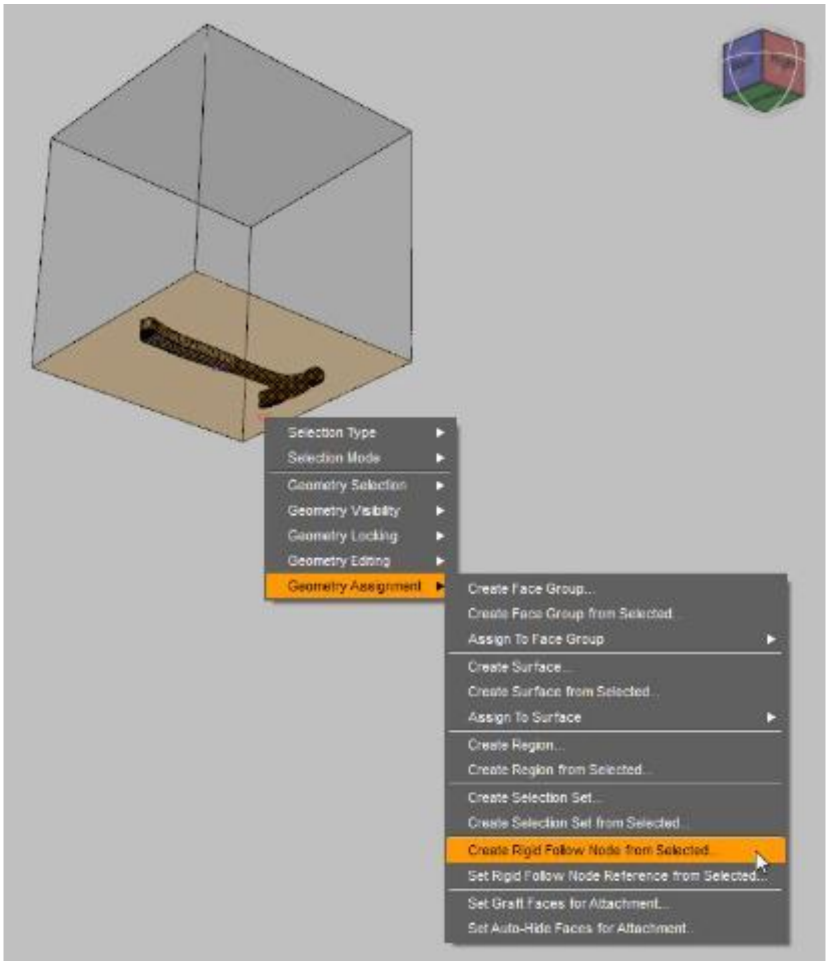
d. Repeat the above steps but make the upper cube only 1 division. Now as it drops it will not distort; however, as it lands on the lower cube it will shrink slightly (hard to see in these images but easy to see in an animation). I could not find a surface property combination that eliminated the slight shrinkage but it doesn't matter for our purpose.



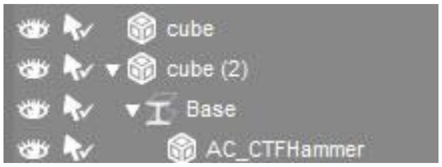
e. Add an object to the scene and position it at the bottom of the upper cube. I found it works best if it's mostly inside the box with a little extending below.



f. Next add a rigid follow node to the box. Go to the Tool Settings pane and from the drop down box choose Geometry Editor. In the viewport left-click on the bottom face of the upper cube. Then right-click in the viewport and select Geometry Assignment/Create Rigid Follow Node from Selected… and give it a name when the dialog box opens.


Selection Type
Selection Mode
Geometry Selection
Geometry Visibility
Geometry Locking
Geometry Editing
Geometry Assignment
Create Face Group...
Create Face Group from Selected...
Assign To Face Group
Create Surface...
Create Surface from Selected...
Assign To Surface
Create Region...
Create Region from Selected...
Create Selection Set...
Create Selection Set from Selected...
Create Rigid Follow Node from Selected...
Set Rigid Follow Node Reference from Selected...
Set Graft Faces for Attachment...
Set Auto-Hide Faces for Attachment...

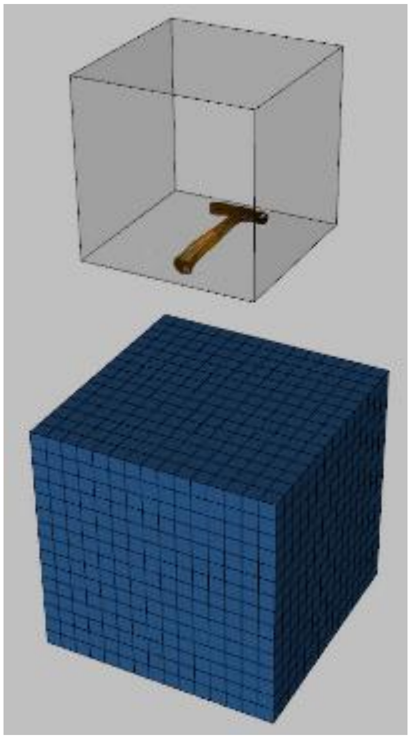
g. Now when you go to the Scene pane you'll see that the upper cube has a child with an icon that looks like an I-beam. Drag the hammer on top of that node to make it a child of the rigid follow node:



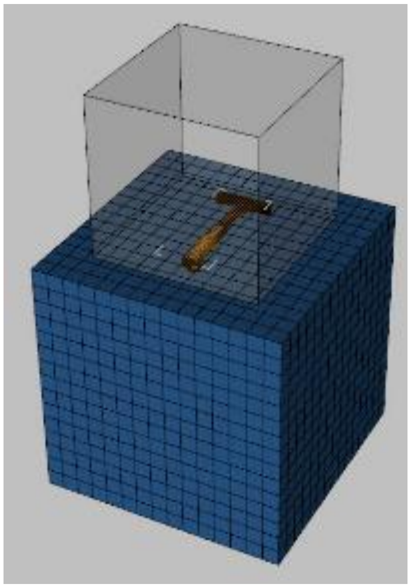
(You don't have to add the rigid follow node before the simulation. You can add it anytime, even after an object has undergone a simulation.)

h. Rerun the dForce simulation. As the box drops, the hammer drops with it. When it hits the lower box, it stops falling but does rotate (most likely because the polygon face is undergoing some twisting transformation that isn't visible on the screen).

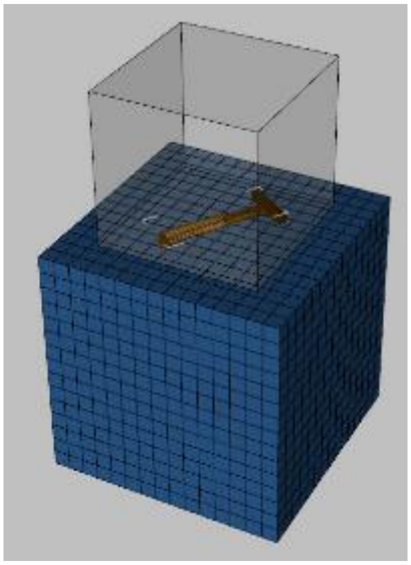
Frame 0:



Frame 11:

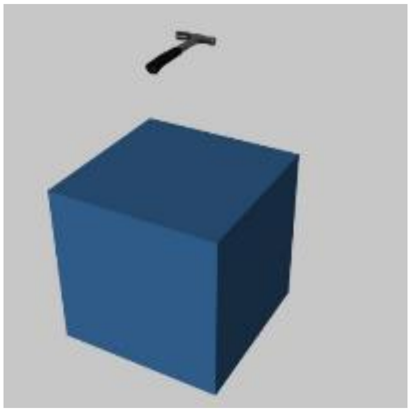


Frame 29:

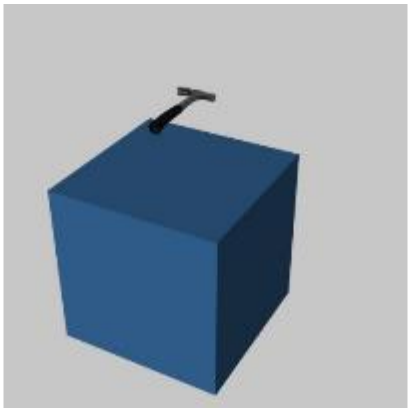


If you hide the cube, you just see the hammer dropping:

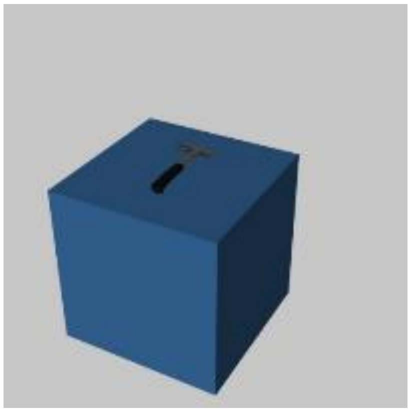
Frame 0:



Frame 7:

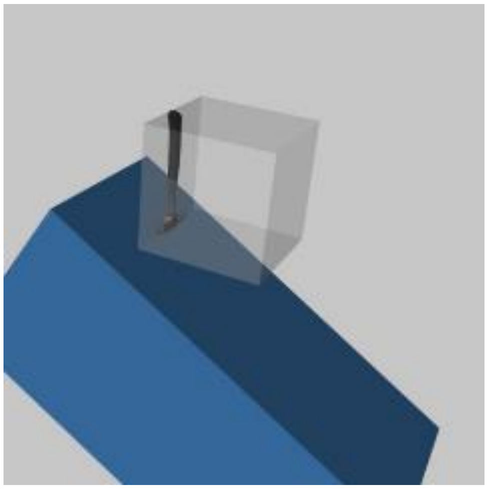


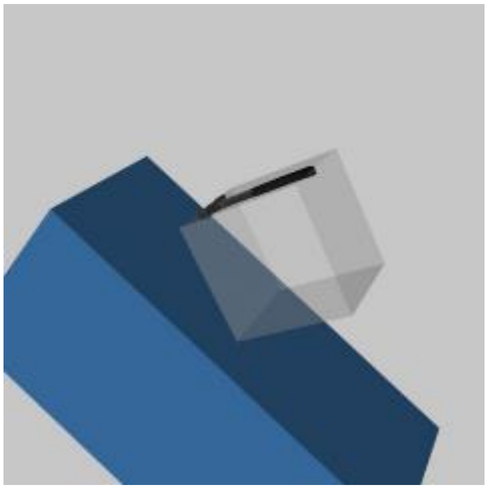
Frame 11:



i. Although it works, I'm not sure what value this flat drop has. If making static images, you could just put the hammer on the cube. If making an animation, put the hammer in the air at frame 0, and at frame 11 put it on the cube. There is no physics other than gravity so you don't get a bounce or other reaction. However, it may be useful with some forms of motion.

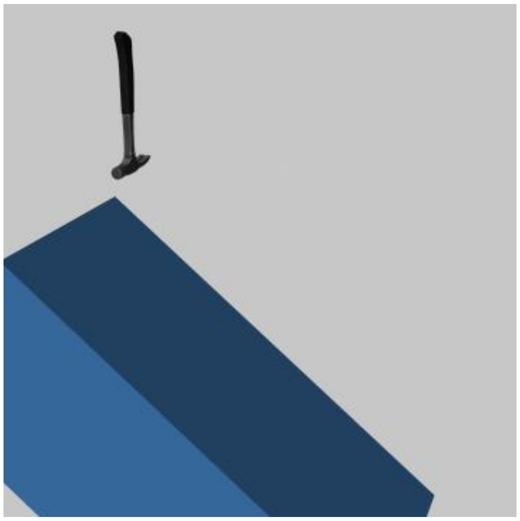
j. I made the lower cube into a ramp and sized the hammer to be the same size as the side of the box. When it drops, it hits the ramp, rotates, and continues to do so down the slope. The hammer follows along, and with the cube hidden, it look like its bouncing.



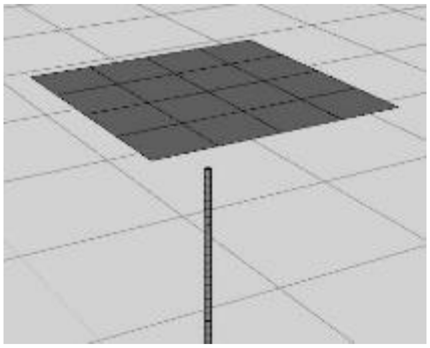


This gif show an animation of the drop and bounce from frame 0 to 42. (There was a glitch with the animation from frame 44-49 where the hammer flipped upside down, then flipped back at frame 50 and stayed that way until the end frame 60. I have no idea what in the simulation caused that short-time reversal.)

(Click here or on image to right to see gif)



27. **Creating a Kite from Primitives**. I used primitive planes and cylinders to assemble a kite and dForce to animate it.

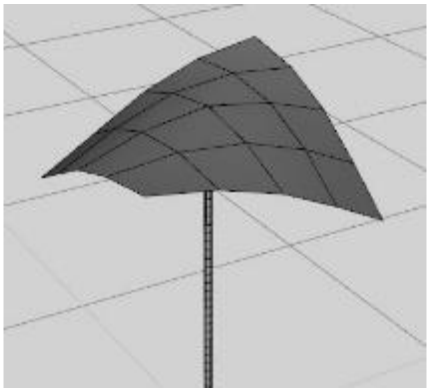


a. Create a 3 foot primitive plane with 4 divisions and primary axis Y positive (from main menu Create/New Primitive). Name it “kite”.

b. Create a 4 foot long 1 inch diameter Y positive primitive cylinder. Be sure to position it so it is at the center point of the plane.

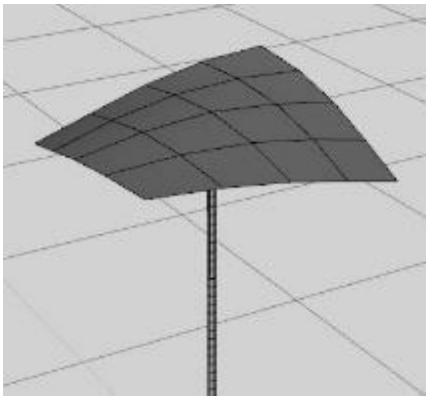
c. Y translate the plane so it is slightly above the cylinder (see image).

d. Add a dForce dynamic surface modifier to the plane.



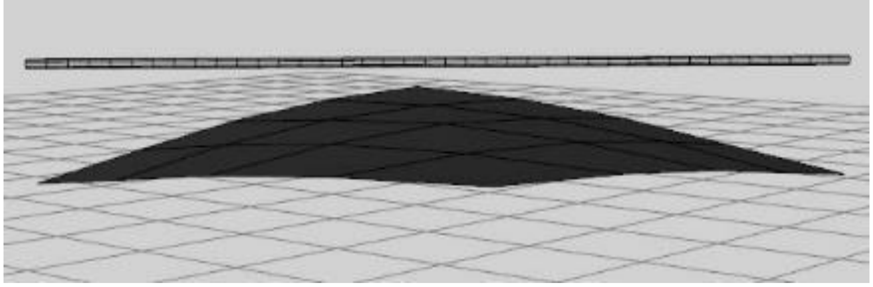
e. Run the dForce simulation. You should get a curved plane similar to what is shown to the right. Two factors affect the amount of curvature: how far the plane is from the cylinder and what Stabilization Time you set in the Simulation Settings pane (I think I used 0.5).

f. I found the curvature more than I wanted so I set Y-scale to 50%:

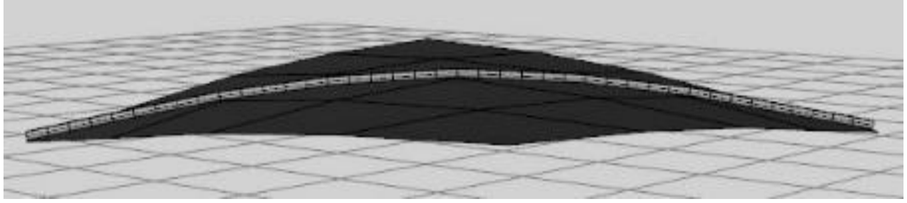


g. Now that you’ve curved the kite, set Freeze Simulation to On in the Parameters pane. Also, you can hide or delete the support stick as its job is done. **IMPORTANT**: although frozen, if you hit the Clear button in the Simulation Settings pane you will restore the plane to flat. If you need to reset another object in the scene after running a simulation , select it and use Ctrl+Shift+D to clear it.

h. Create another cylinder of 4.2 feet and 0.5 inch diameter. Name it “cross spar”. Parent it to the kite. Add a dForce dynamic modifier to it. Position it so it is above the curved kite as shown below. Run a dForce simulation.



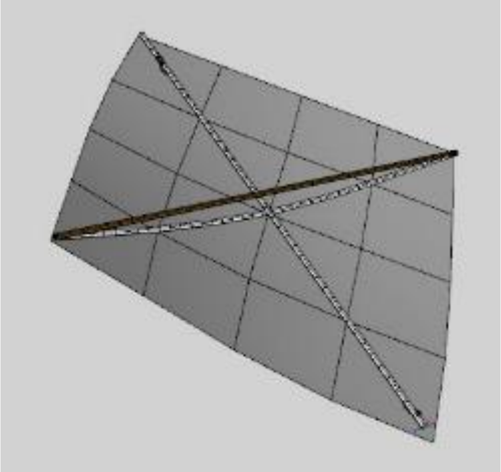
Here’s the result:



Y-translate the spar down so it’s on the underside of the kite. Adjust the scale as needed. Set Freeze Simulation to On in the Parameters pane.

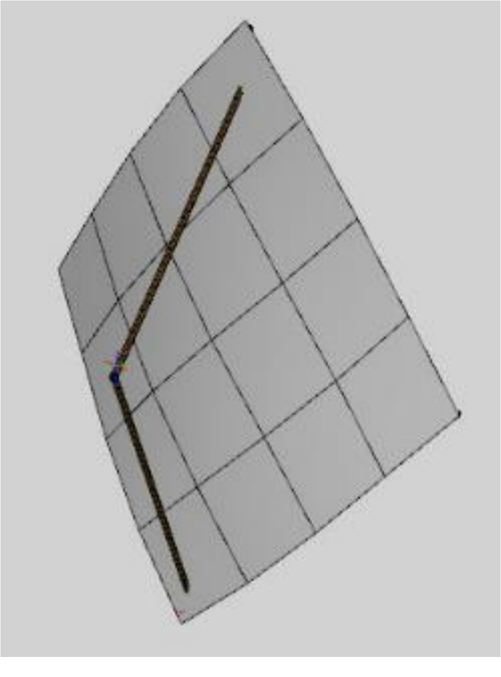
i. Create a second cylinder of 4 feet but only 0.25 inch diameter. Name it “cross tie”. Parent it to the kite. Do not add a dForce modifier to it. Position it so it runs from one tip of the curved spar to the other. Adjust the scale as needed so it fits. This serves as the rope that keeps the cross spar curved.

j. Create a third cylinder of 4.2 feet and 0.5 inch diameter. Name it “strut”. Parent it to the kite. Do not add a dForce modifier to it. Position it so it’s perpendicular to the spar and also on the underside of the kite. The back side of the kite should now look like this:



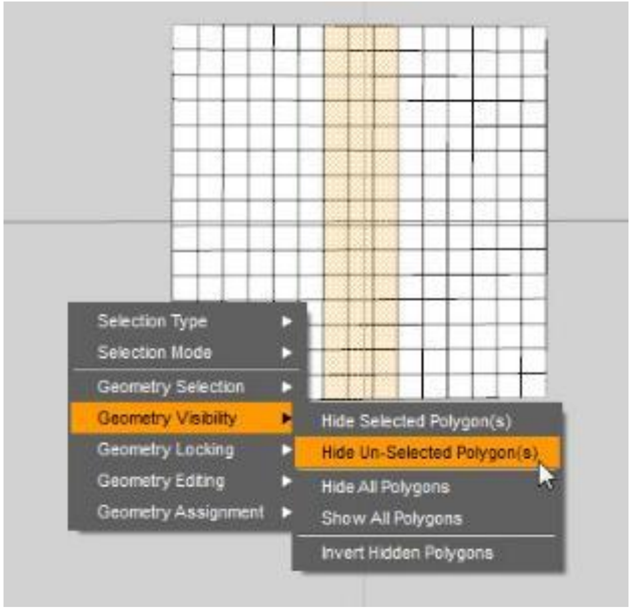
k. Create a 4th cylinder of 2 feet and 0.25 inch diameter. Name it “bridle”. Parent it to the kite. Do not add a dForce modifier to it. Position it so it is on the front side of the kite at an angle with one end near the bottom tip of the kite.

l. Repeat step k but attach it at the upper tip of the kite. The front side of the kite should look like this:

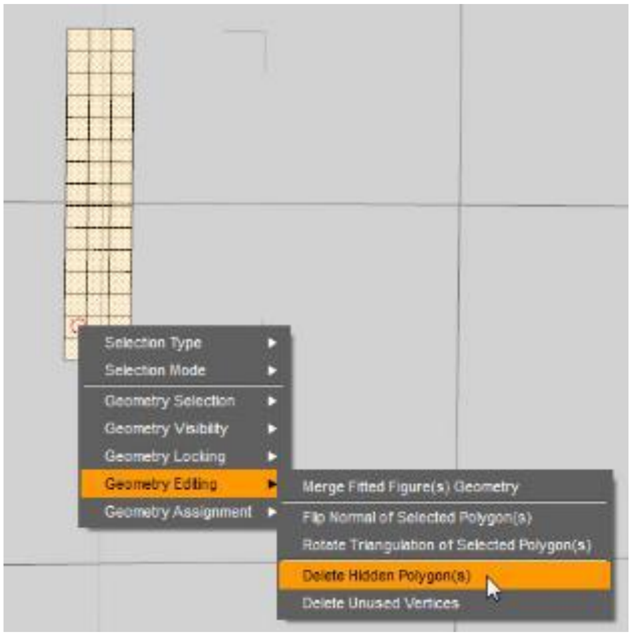


m. Create a plane of 4 feet with 15 divisions. Give it the name “tail”. Add a dForce dynamic modifier to it.

n. Go to the Tool Settings pane and select the Geometry Editor. Select the middle 3 columns of polygons. Then right click in the viewport and select Geometry Visibility/Hide Un-Selected Polygon(s).

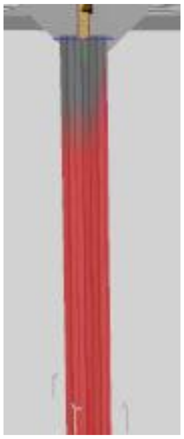


o. Right click in the viewport again and select Geometry Editing/Delete Hidden Polygon(s).

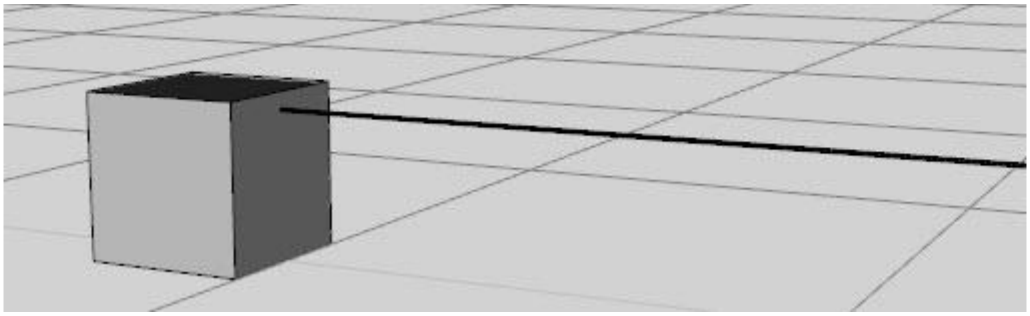


p. Parent tail to bottom tip of the kite. Add a dForce dynamic modifier to it. Set the Dynamic Strength in the Surfaces pane to 0.90 so the tail doesn’t get all wrapped up.

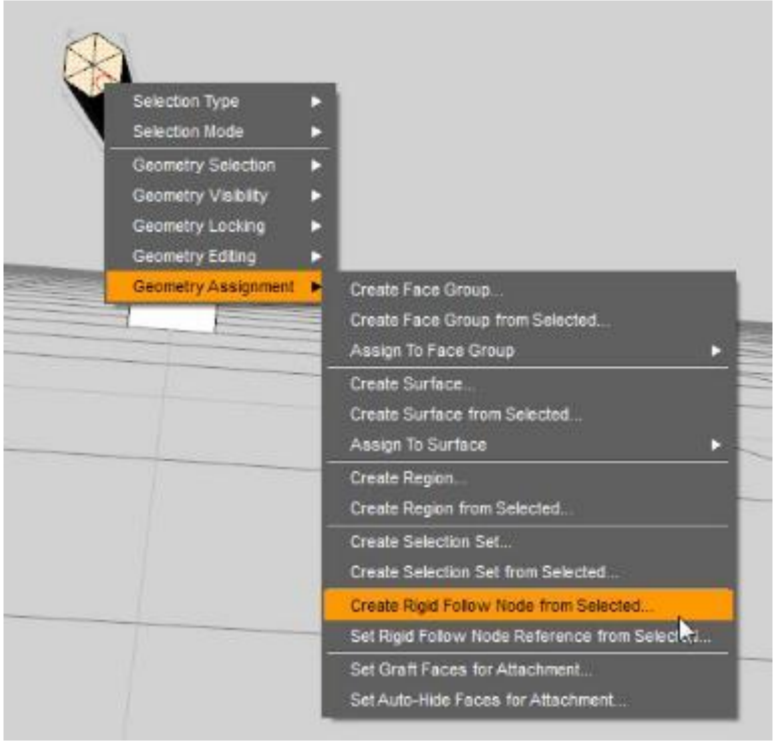
q. Add a dForce weight map to the tail and remove the influence on the polygons next to the kite (see gray area in image to right). This will keep the tail attached to the kite.



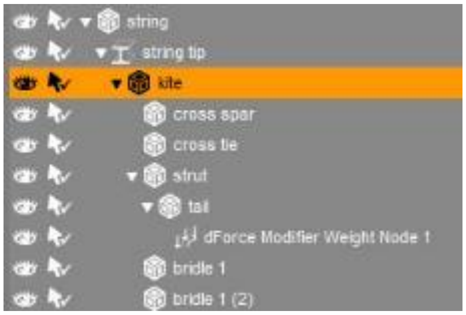
r. Create a small cube and position it away from the kite. Create a 20 foot long 0.25 inch cylinder. Name it "string". Embed one end inside the side of the cube. This will serve to anchor the kite string.



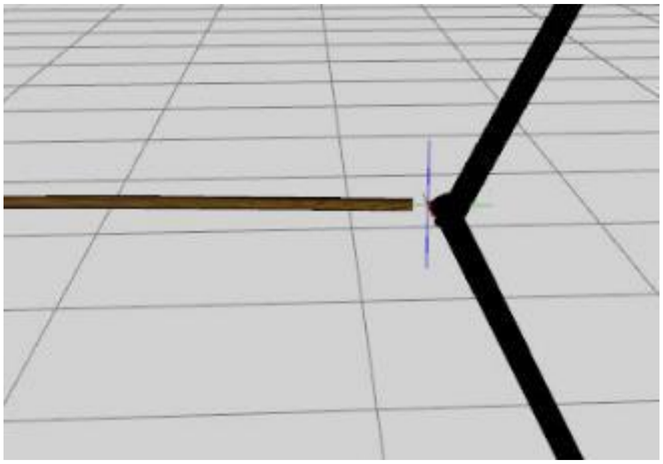
s. At the other end of the string, move the camera and adjust it so you can see the polygons on the end of the string. Go to the Geometry Editor and select one polygon on the end. Right click in the viewport and choose Geometry Assignment/Create Rigid Follow Node from Selected… Give it the name "string tip".


Selection Type
Selection Mode
Geometry Selection
Geometry Visibility
Geometry Locking
Geometry Editing
Geometry Assignment
Create Face Group...
Create Face Group from Selected...
Assign To Face Group
Create Surface...
Create Surface from Selected...
Assign To Surface
Create Region...
Create Region from Selected...
Create Selection Set...
Create Selection Set from Selected...
Create Rigid Follow Node from Selected...
Set Rigid Follow Node Reference from Selected...
Set Graft Faces for Attachment...
Set Auto-Hide Faces for Attachment...

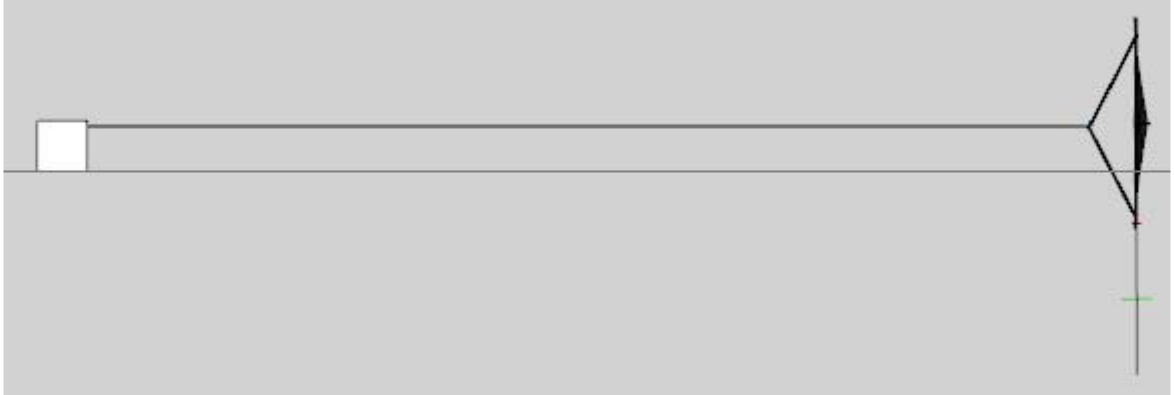
In the Scene Pane, you will see the rigid node, showing an I-beam icon, parented to the string.



t. Move the kite so the tip of the bridle is near the rigid node on the end of the string. I found better results when I left a small gap between the two (so small you will usually not see it – if you can, just close the gap after the dForce simulation and before rendering.

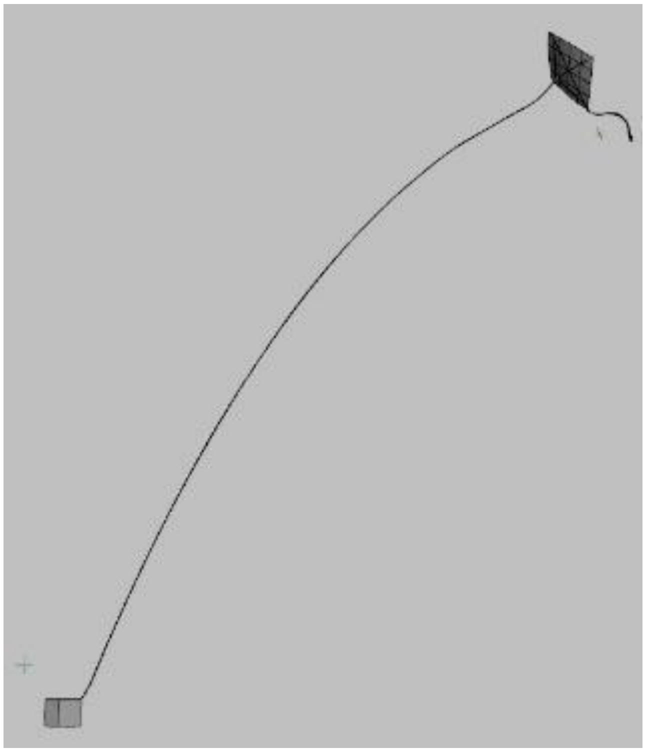


u. This completes the setup of the kite which should look something like this:



v. To summarize, dForce was used to curve a plane to make the kite itself. dForce was also used to curve a cylinder to match the shape of the kite. Both these simulations were then frozen to keep them from changing shape. dForce was used at a reduced dynamic strength to make a plane into a tail. dForce was also used on a long, very narrow cylinder to make it into the kite string. The tip has a rigid follow node so the kite will follow the movement of the string.

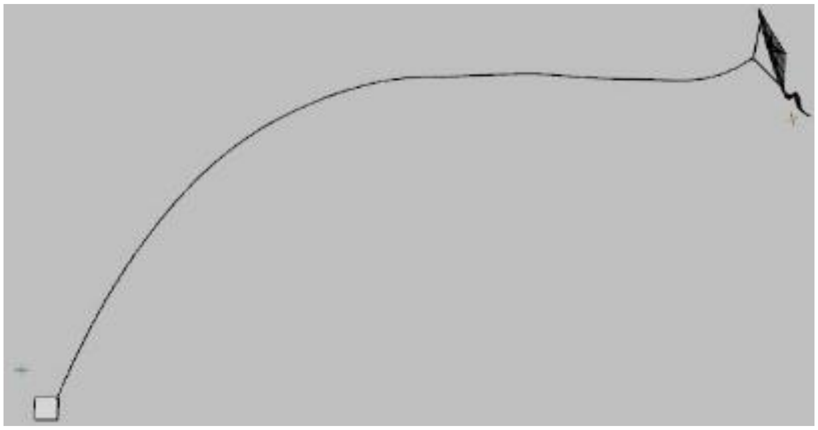
w. I did a Current Frame simulation with Stabilization Time set to 1.5 and Gravity to -1.5 (note that's negative gravity, which is needed for the kite to rise instead of fall). The result is shown at the right:



I also tried an animated simulation, but I found wild behavior of the kite (unusual twisting, momentary flipping, and a slack wavy string near the kite). Not sure why they simulate differently.

I also got somewhat better results if I hid the bridles during the simulation and turned them back on after the fact.

Before I ran the simulation above, I also tilted the kite string towards the sky at a 30 degree angle. If I started with the string horizontal, I got a shape that flattened in the middle as shown to the right.



Here are two finished renders:

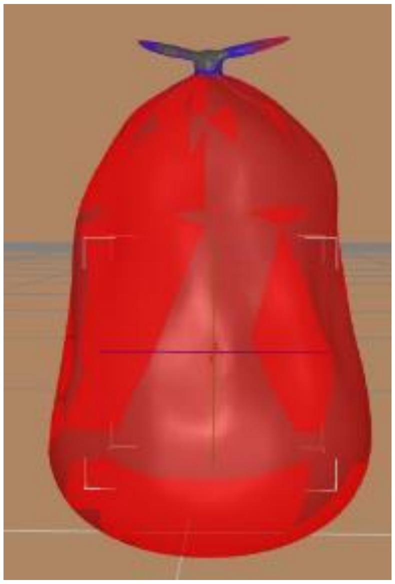
I tried using a wind node but it didn't really add anything to the simulation, negative gravity is enough.

28. **Creating a Bag of Fruit.** There's many occasions for putting items into a soft-sided bag. As the items settle to the bottom, the bag distorts to wrap around them. This tutorial uses fruit as an example but you can use many other objects.

a. Start with some type of bag. I have several in my runtime from various sets. For this topic, I used a trash bag from Trash Stash.

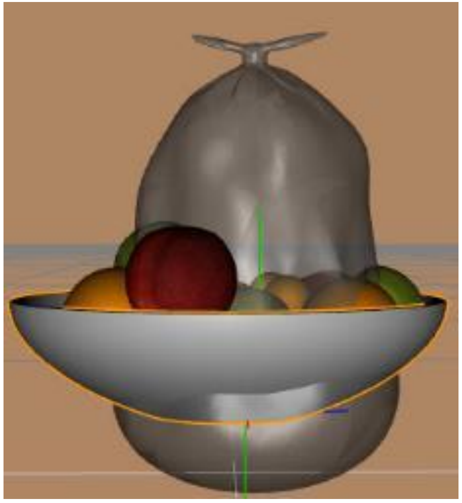
b. To make it easier to see what's going on, I changed the color of the bag to a medium gray and set the opacity to 50%.

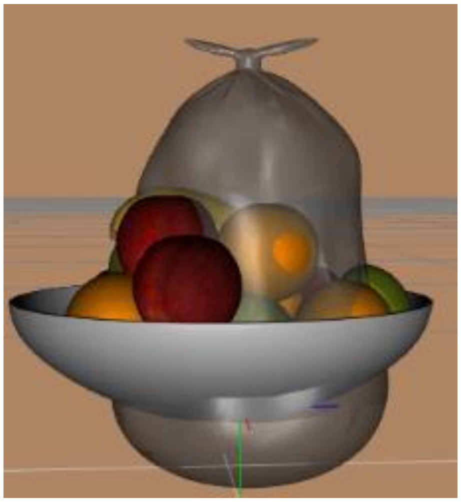
c. Add a dForce dynamic modifier to the bag.



d. Add a dForce Modifier Weight Node to the bag. Use the Node Weight Map brush to remove the color from around the bag's knot (hold the alt key while painting). This will lock the knot in place so the bag doesn't fall to the ground during simulation.

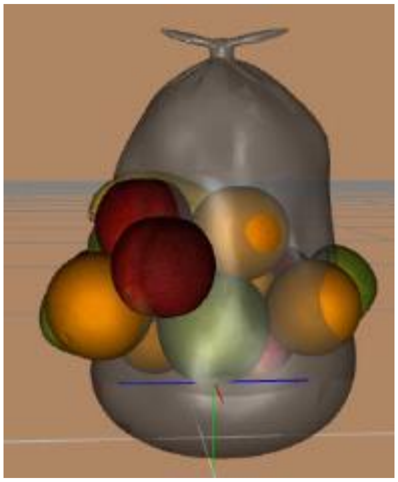
e. Add the fruit. I used MHE:Fruit. I started with the bowl of fruit and added an extra apple, banana, peach, and orange on top.





f. We don't want the bowl, so go to the Tool Settings pane and pick the Geometry Editor. Right click on the bag in the viewport and choose Geometry Selection by surface, pick the bowl surface, then hide the selected polygons using Geometry Visibility, then delete the hidden polygons with Geometry Editing.

g. Create a group with all the fruit in it. This will make it easier to scale the fruit during the dForce simulation.



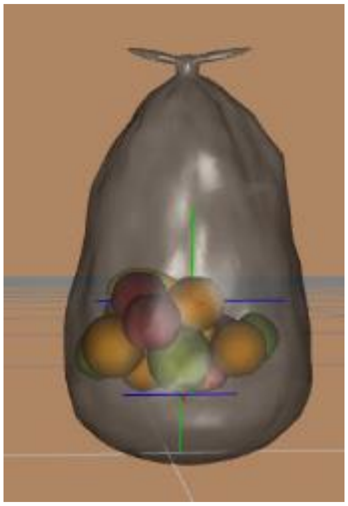
h. You'll see that some of the fruit pokes through the bag. We want the fruit to be this size at the end of the simulation, but in order to stretch the bag during simulation, we need to shrink the fruit so it all fits inside the bag at the start. Set the scale of the group to 60% at Frame 0. Also, move the fruit up to Y-translate 6.5 so it can drop down as the bag simulates.

i. The bag needs some time to drape before the fruit expands too much, so add a keyframe at frame 15 and keep the group scale at 60% but drop the fruit to Y-translate 1.

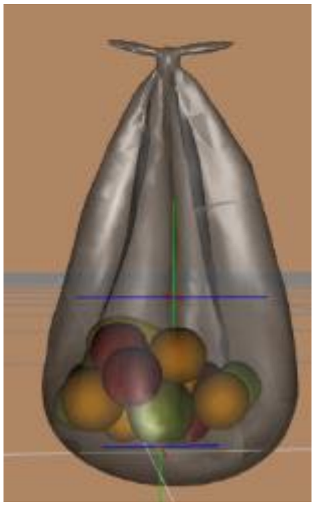
j. At keyframe 30, restore group scale to 100%. Let the simulation run to frame 60.

k. Here are the results. As you can see, the bag now conforms to the fruit.

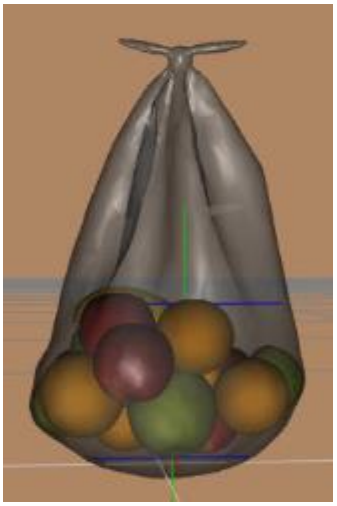
Frame 0:



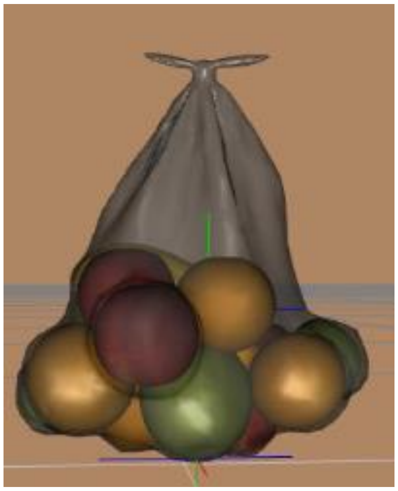
Frame 15:



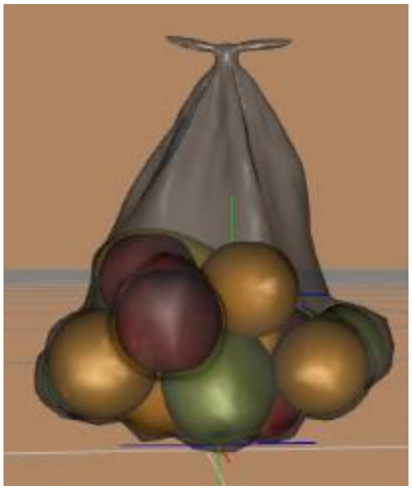
Frame 22:



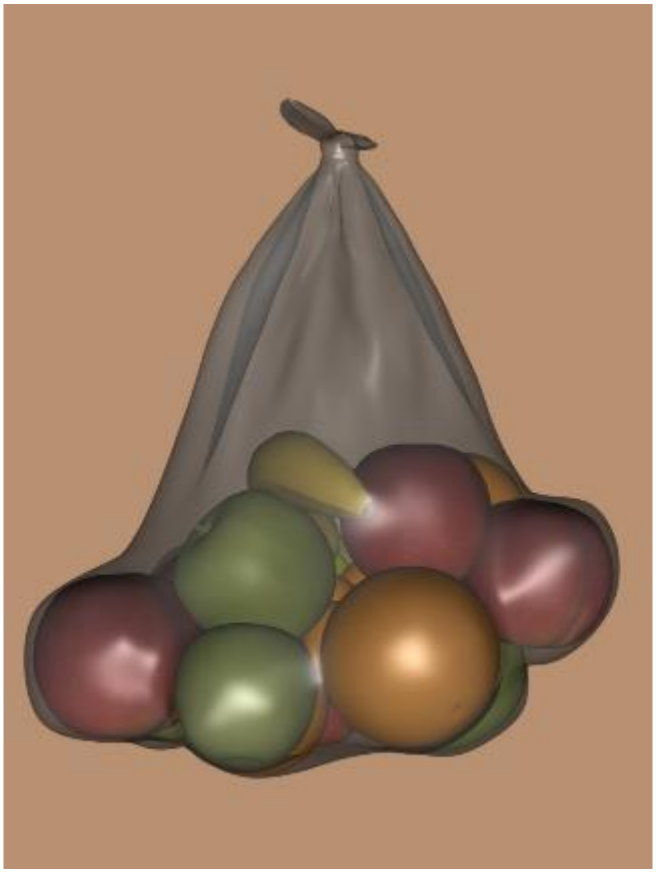
Frame 30:



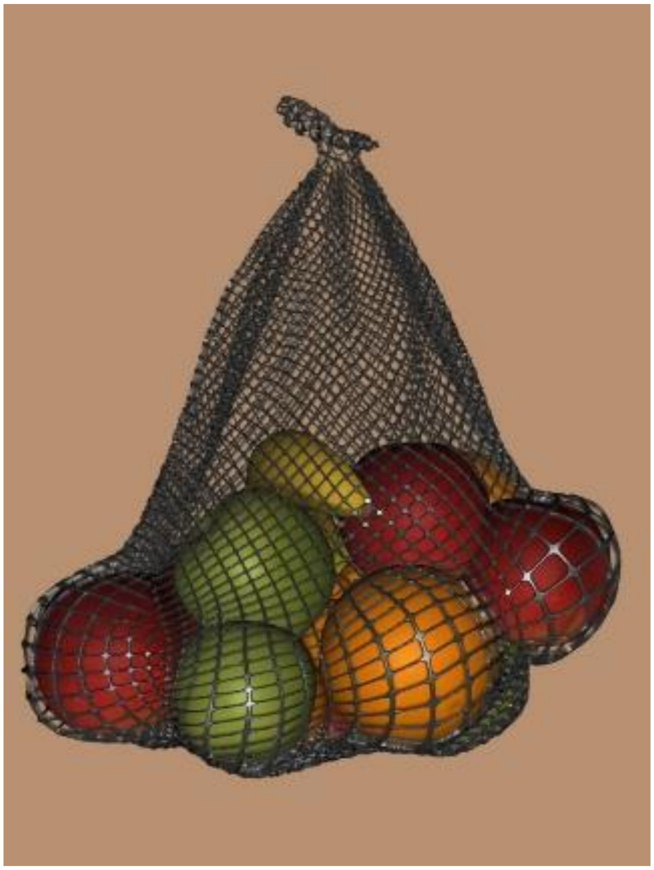
Frame 60:



A 3DL render of the bag of fruit:



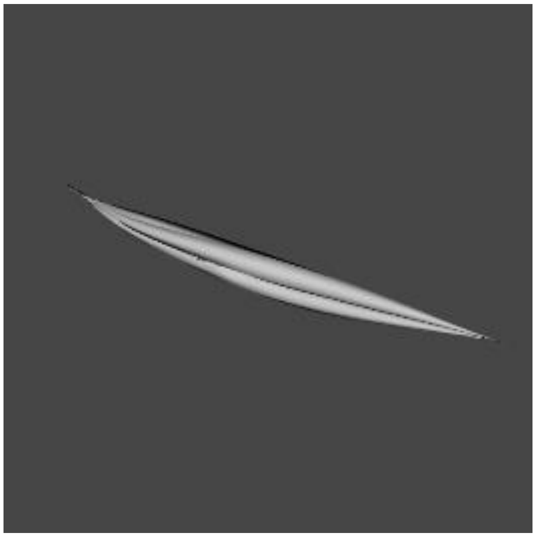
l. You can use any shader on the bag. Some fruit and vegetables come in mesh bags, so I used a shader from Super Shaders – Mesh Magic to create that effect. Here's the render:



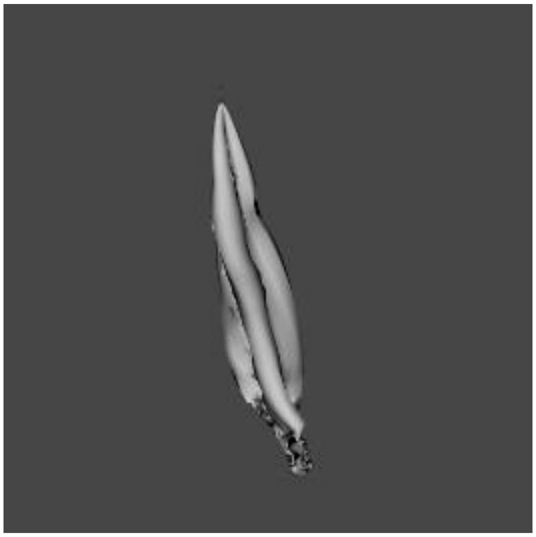
29. **Deflated Soccer Ball.** Applying dForce to a primitive plane is pretty straightforward but if you apply it to a primitive sphere you get wild results. I was finally able to find some surface settings that help and also made a big learning about scale in the process.

    a. Create a primitive sphere with 10 foot diameter, 64 divisions, and 64 sides (I'll come back to why 10 feet later).

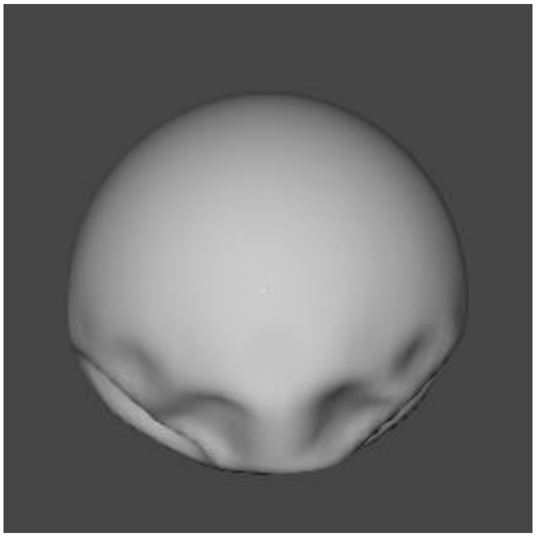
    b. If you use the default surface parameters, you'll get a result like this: a fully deflated, stretched out mesh.



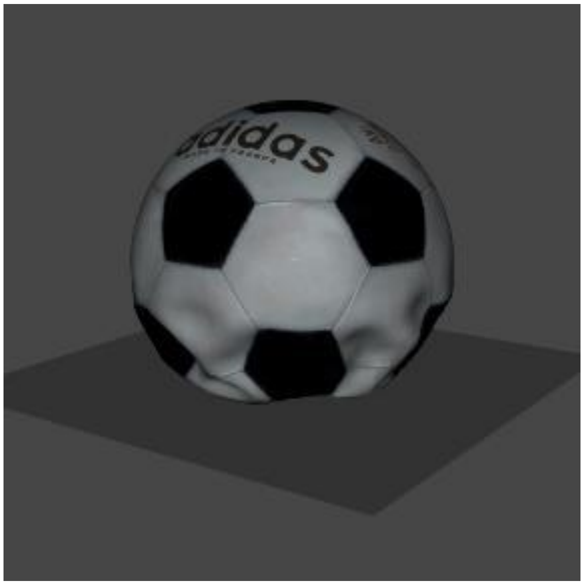
    c. Reduce Bend Stiffness from the default of 0.5 down to 0.1. It's still deflated but not stretched out as much.



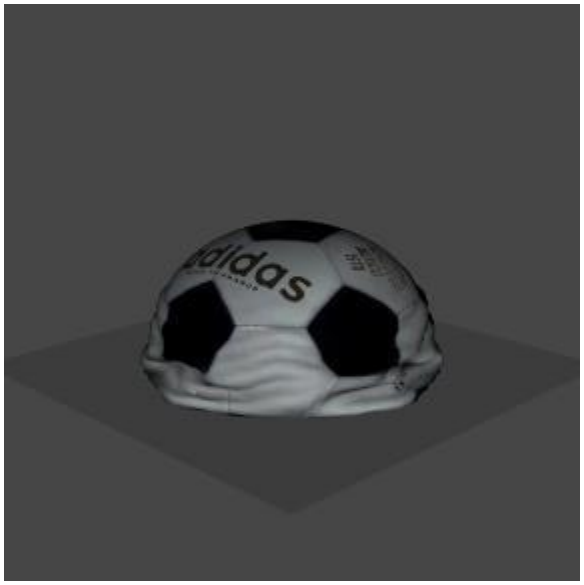
    d. Next reduce Stretch Stiffness from the default of 0.8 down to 0.1. The upper surface of the sphere retains its shape while the lower part starts to pull in. Now you have a ball that looks deflated.



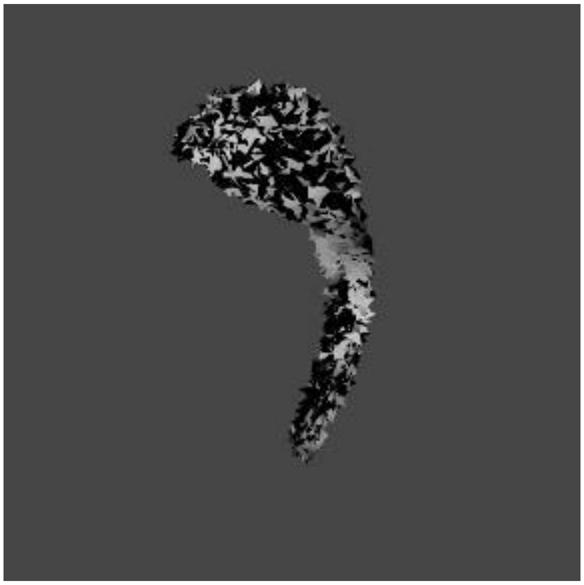
e. Add a soccer ball texture and it looks like someone forgot to pump up the ball.



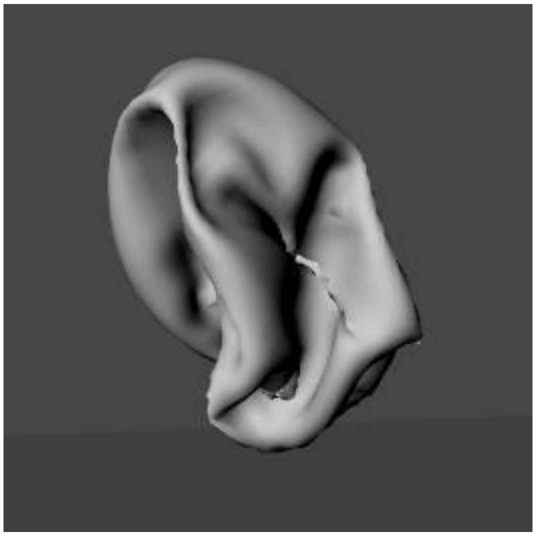
f. You'll notice in the previous image the ball is sitting on a plane but I did not have it collide with the plane during the simulation. It was lowered afterwards. In this image the ball was allowed to collide with the plane and you get a different look. Either works but the one above appears to me more like the ones I've seen in real life.



g. This is where it gets interesting. No one plays with a 10-foot ball. So here's a 1-foot diameter ball with the same 64 divisions and 64 sides after simulation using default surface values. It explodes instantly when you start the simulation.

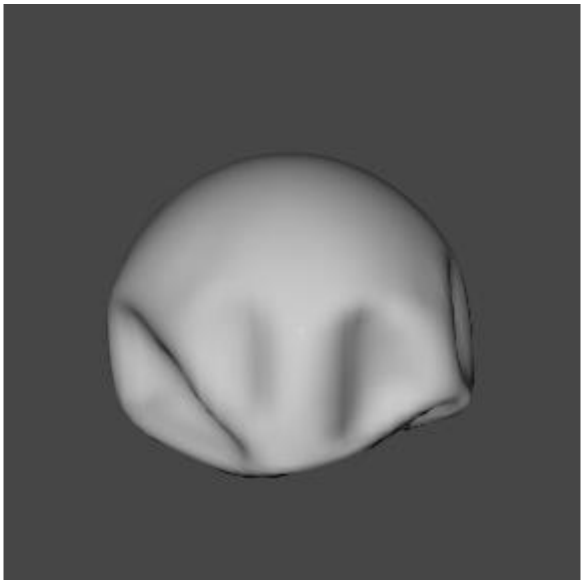


h. Set the Stretch and Bend Stiffness to 0.1 and it holds together, but this smaller ball collapses in on itself much more than the 10-foot diameter ball. I could not find surface settings that gave the results I got with the 10-foot ball.

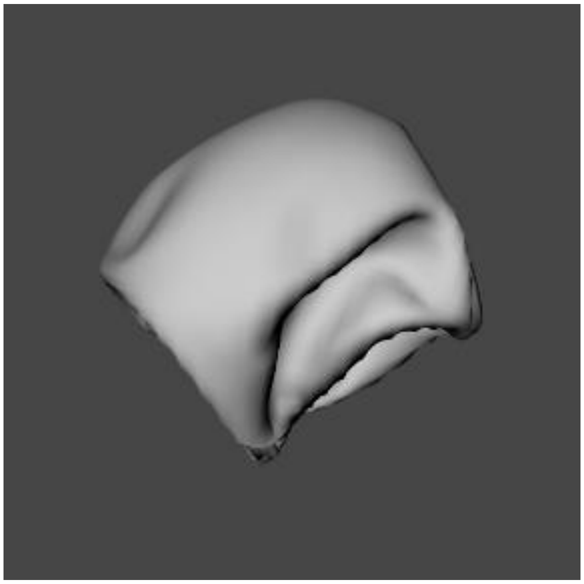


i. Another way to get to a smaller ball is to scale down the 10-foot ball. This series does it BEFORE simulation.

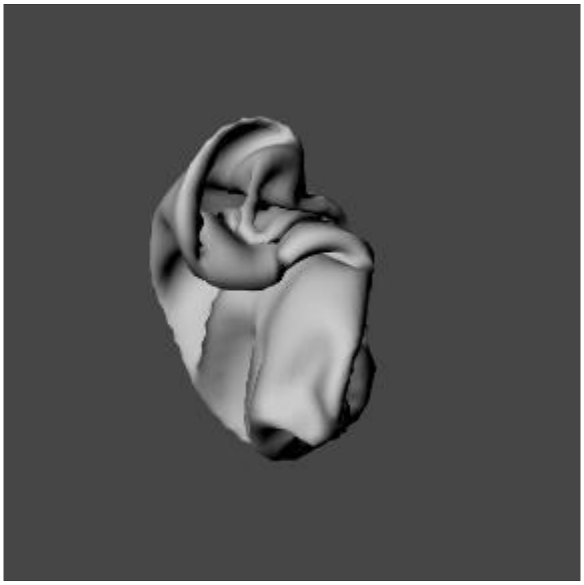
50% scale (5-foot):



25% scale (2.5 foot):



10% scale (1 foot):



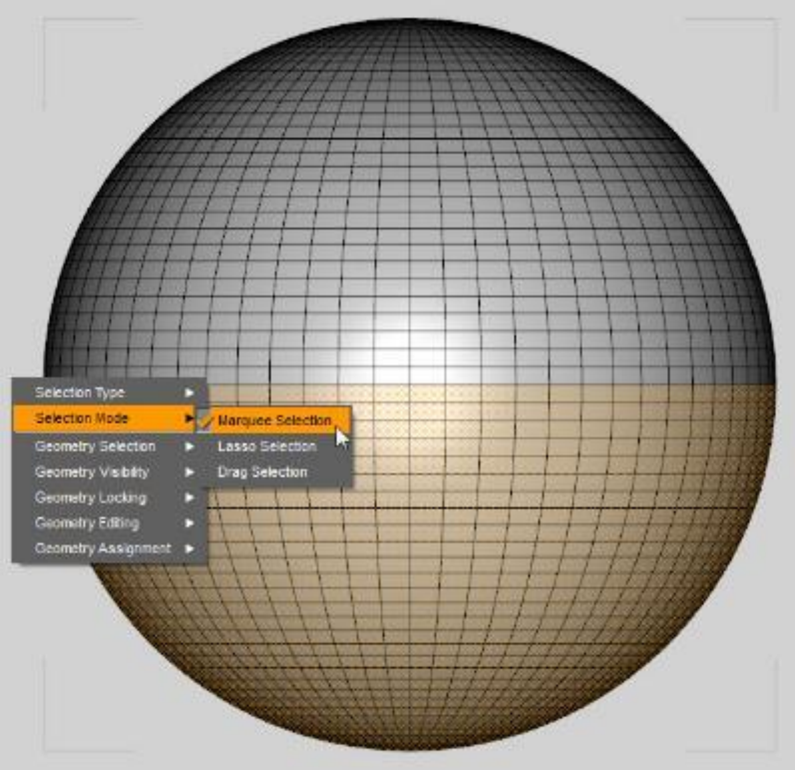
The 10-foot ball with 10% scale makes it equal in size to a 1-foot ball and it has a somewhat similar simulation result as the 1-foot ball without scaling.

Clearly dForce is affected by scale!

So the best way to get a smaller ball with the look of the deflated ball in step e is to start with a 10-foot ball, run the simulation, then scale it down.

Since dForce is affected by scale, what works on one size object will not necessarily work on a larger size object. That makes it more difficult to predict your result when scale also plays a role.

30. **Explorations of the Half Sphere.** For this section, nothing in particular was planned except to see how the surface parameters from the deflated soccer ball impact a half sphere.

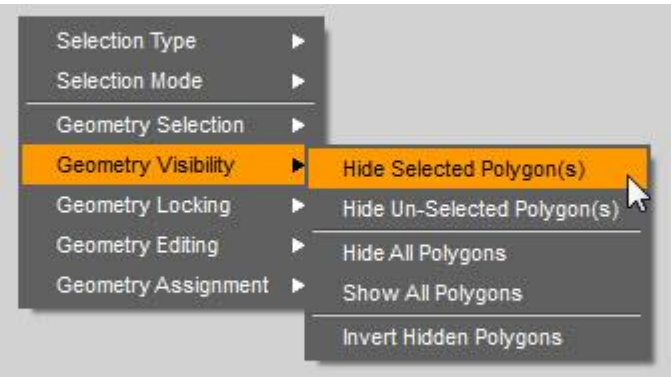


    a. First, create a primitive sphere with 10 foot diameter, 64 divisions, and 64 sides. Add a dForce dynamic modifier.

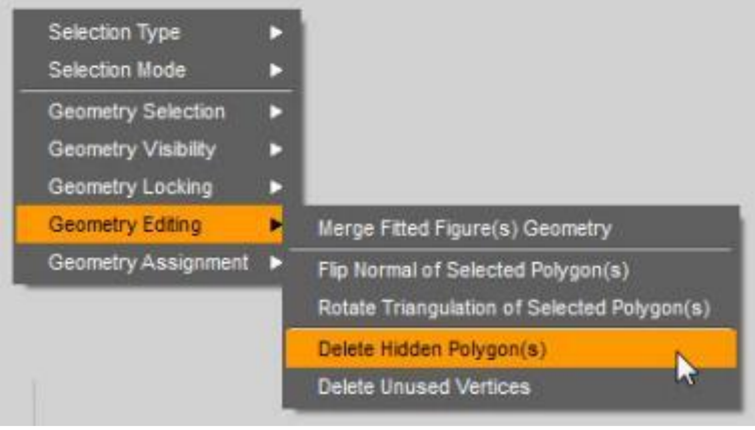
    b. Go to the front view for ease of selection. Turn on wire shaded mode.

    c. Go to the Tool Settings pane and select the Geometry Editor. Right click in the viewport and choose Selection Mode/Marquee Selection. Hold down the left mouse button and select the bottom half of the sphere.

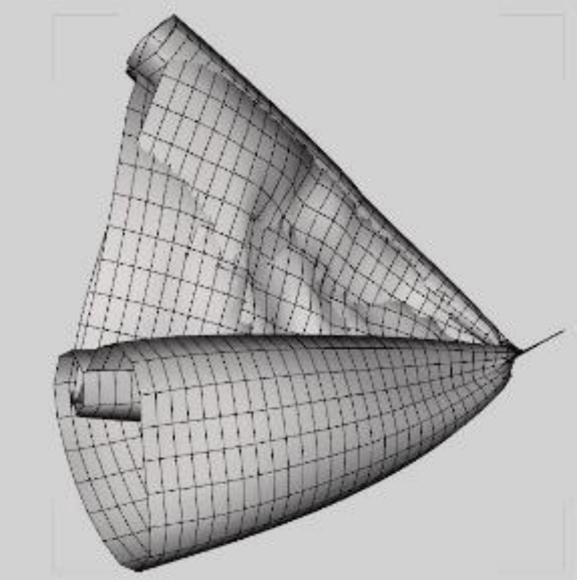
    d. Right click in the viewport and choose Geometry Visibility/Hide Selected Polygon(s).



e. Right click in the viewport and choose Geometry Editing/Delete Hidden Polygon(s). Now you'll have a half sphere.

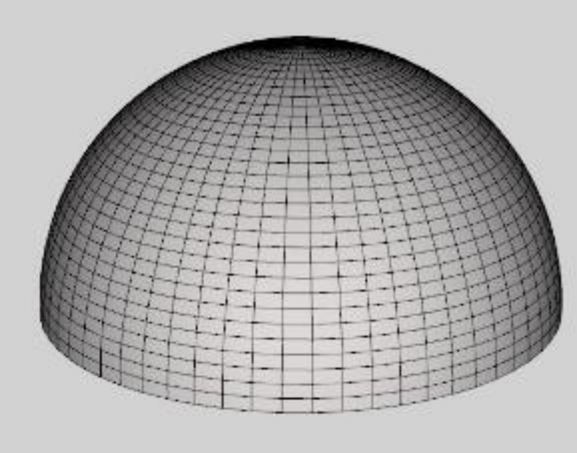


f. Run the dForce simulation. You'll see the sphere twist into a weird shape.

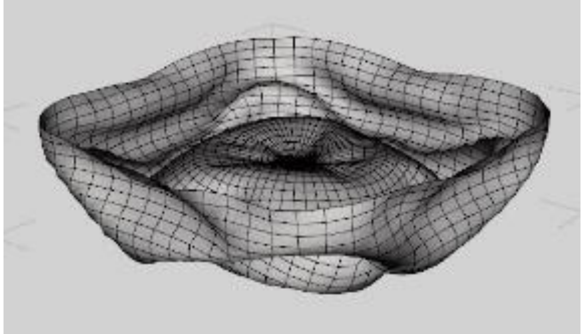


g. Select the surface and on the Surfaces pane, change Stretch Stiffness to 0.1 and Bend Stiffness to 0.1. Rerun the simulation. Now you'll see that the sphere does not transform at all but only drops a short distance.

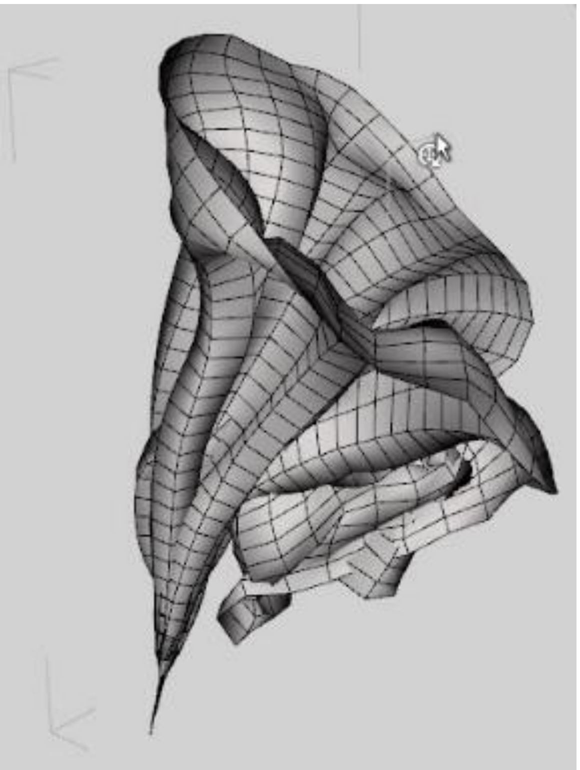
If Stretch and Bend Stiffness are set to 0.2, the sphere will wrap up like in step f. At 0.15 it will look like this. There's a sharp transition from remaining stable to wrapping up but I don't know why.



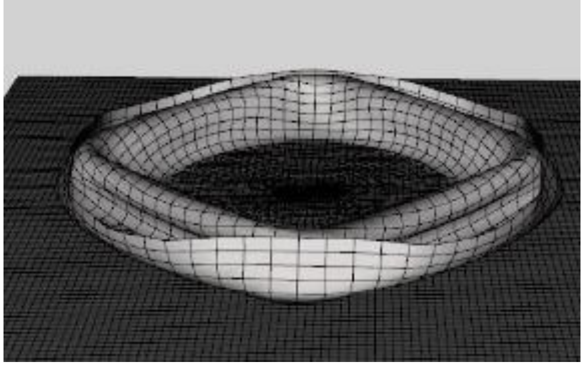
h. Flip the sphere by setting x-rotation to -180. Keep Stretch and Bend Stiffness at 0.1. Run the simulation. In this case, you will see deformation that looks like this:



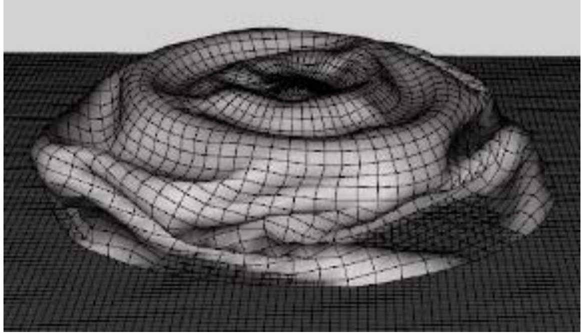
Compare that to the default setting of Stretch Stiffness 0.8 and Bend Stiffness of 0.5 (image at right).



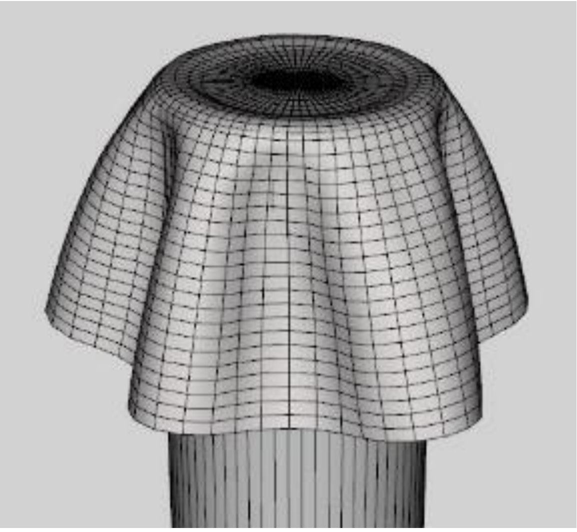
i. None of the above simulations had any collisions with other objects. They are just the natural deformation from settling under the default gravity and air resistance settings. If you collide the upside down half sphere with a plane you get this:



Here’s the result of colliding the upside right half sphere with a plane. Whereas the half sphere without collision showed no transformations, this one clearly does.

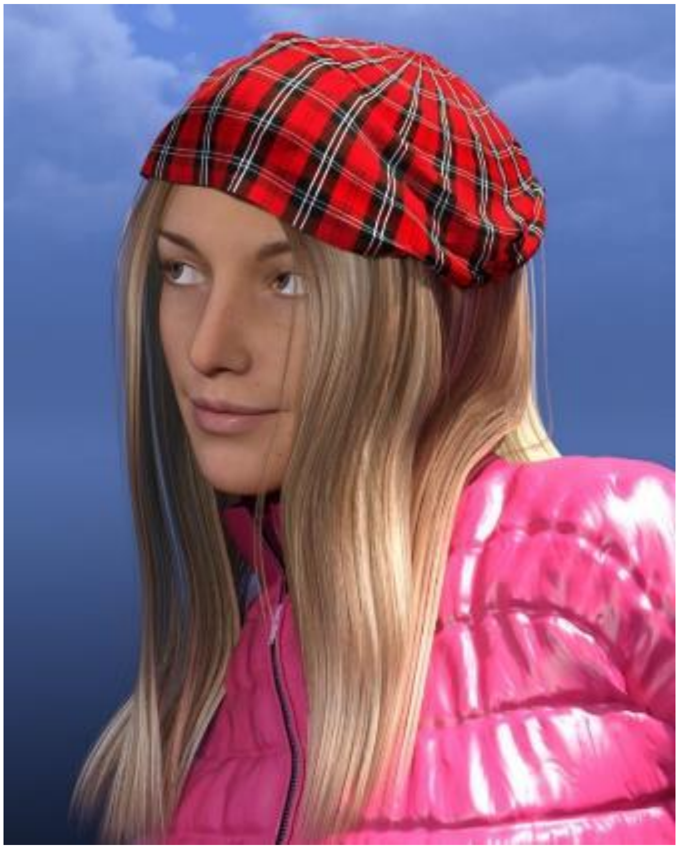


j. You can also collide the upside right half sphere with a cylinder. In this case it looks a bit like a tablecloth. It required a longer duration (3 instead of 1) to get this degree of draping.



k. I decided to test the half sphere dropping against a figure's head to see what it would look like. I scaled it down to head size and let it drape onto a G3F.

Here's an iRay render of the "beanie".



Since the sphere is only a single layer of polygons, the edge has no dimension to it. Using a full sphere and stopping the simulation short can give you a fuller looking beanie.

31. **Creating a Round Floor Cushion with Negative Gravity.** Previously I showed how to make a foot stool out of a primitive cube. At the time, I could not get a sphere to deform without it going crazy. With the settings I used in the last two sections, I can now work with primitive spheres. In this section, I'll discuss two ways to make a floor cushion look like a figure is sitting down into it. Both rely on a weight maps but they are prepared using two different techniques.

    a. Create a primitive sphere with 2.5 foot diameter, 64 divisions, and 64 sides. Set the Y-scale to 30%. Add a dForce dynamic modifier and then add a dForce Modifier Weight Node (how to do this is covered in previous sections).

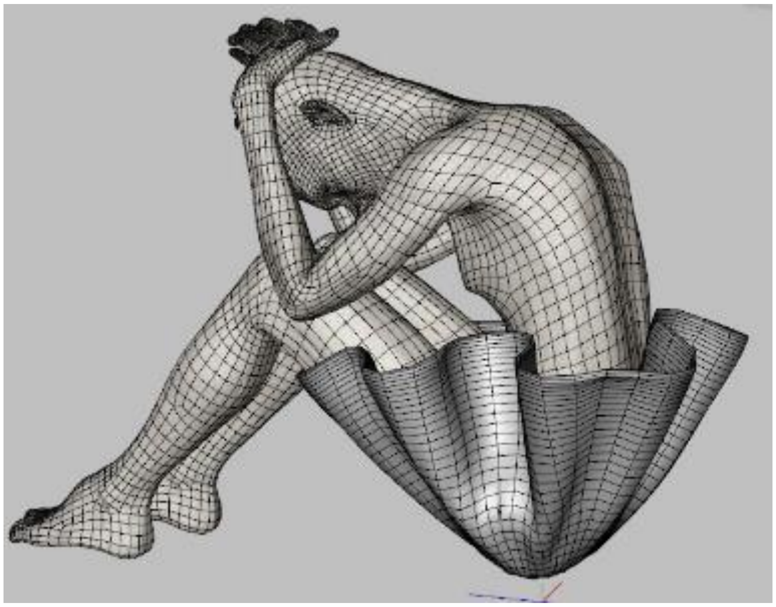
    b. Add a figure in a sitting pose. I used Woody for Genesis 3 Males and a Desolation Road sitting pose. Drop the figure to the floor. With the pose I use, the feet are on the ground but the hips leave some space beneath them.

    c. Set up an animated timeline. Have the figure above the sphere at frame 0, then drop it down at frame 15. End the simulation at frame 30.

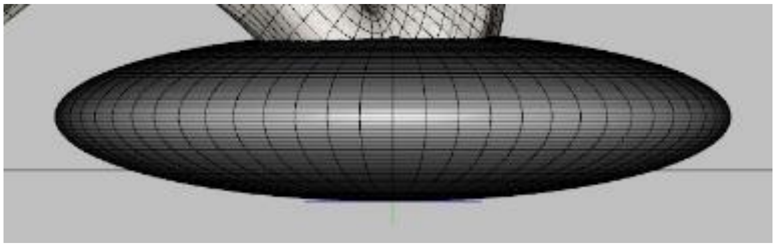
    d. Select the sphere's surface. In the Surfaces pane, set the Simulation Stretch Stiffness and Bend Stiffness to 0.1.

    e. On the Simulations Settings pane, set Gravity to -0.5.

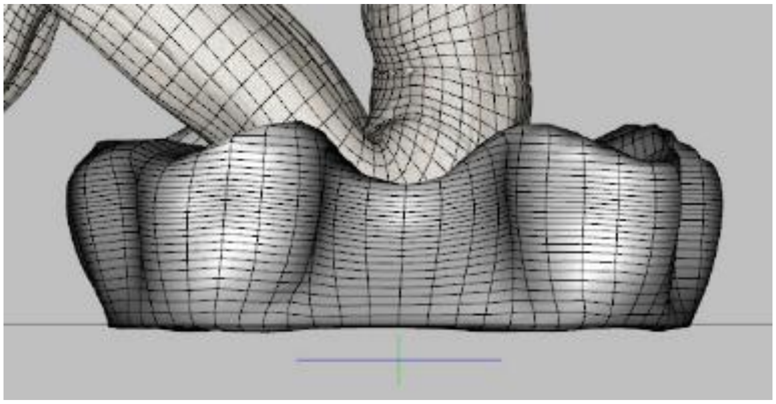
f. If you run the simulation as it is now set up, the negative gravity will cause the sphere to go upward and wrap around the figure. It will not look like the figure is sitting on the cushion.



g. One way to keep an object secured from wrapping up and around a figure under negative gravity is to embed the bottom into another object. In this case I used a plane as a floor with the bottom of the sphere below the floor.

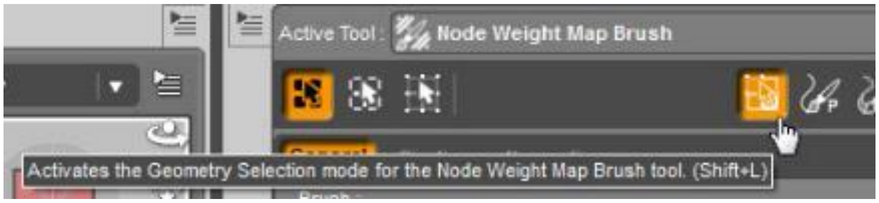


It works but the results are not realistic.



h. Select the weight node in the Scene pane and go to the Tool Settings pane and select the Node Weight Map brush. Add the Influence Weight Map. You will see the sphere turn red, meaning 100% influence on all vertices.

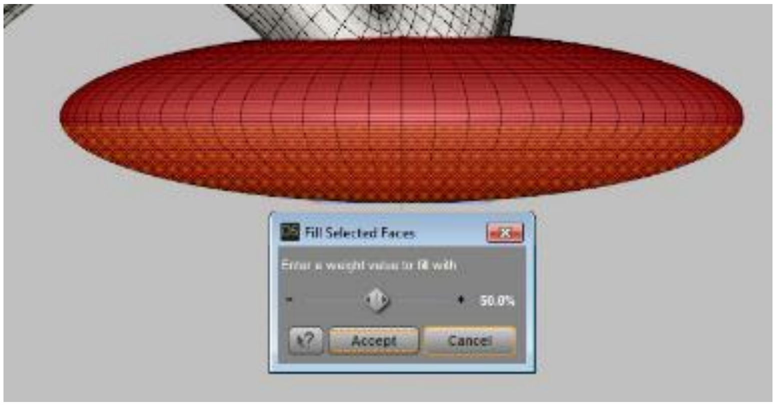
i. Use the Side or Front view. Click on the Geometry Selection Mode (the first of the far right icons at the top of the pane).



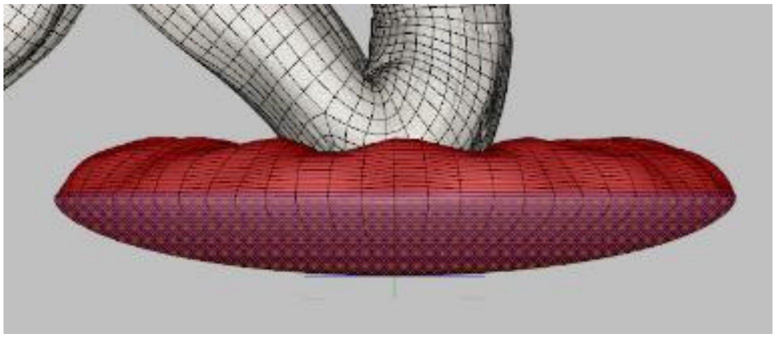
Active Tool : Node Weight Map Brush

Activates the Geometry Selection mode for the Node Weight Map Brush tool. (Shift+L)

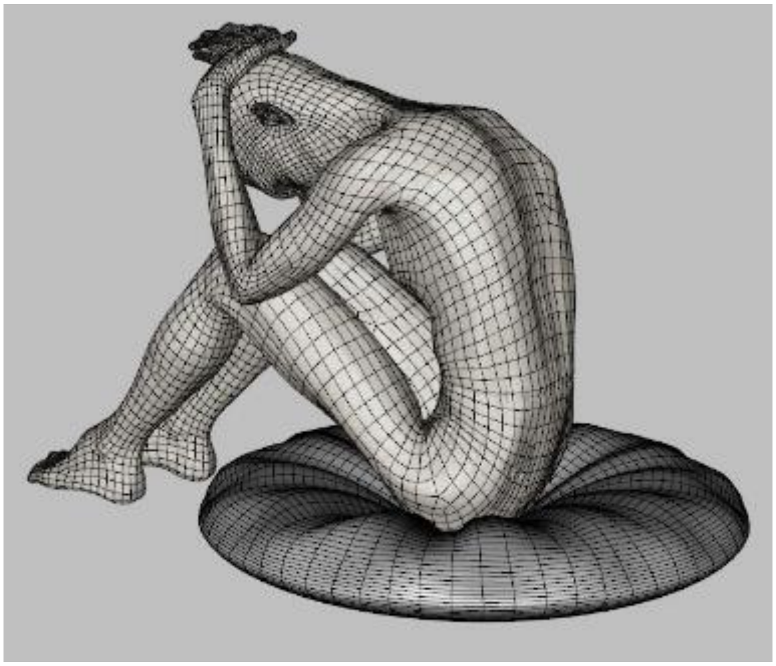
j. Right click in the viewport and choose Selection Mode/Marquee Selection. Hold down the left mouse button and select the bottom half of the sphere (in the image, it’s the orange area that starts slightly below the midline of the squashed sphere). Right click in the viewport and choose Weight Editing/Fill Selected… and enter a value of 50%.



k. Run the simulation. Notice that we suppressed transformation on the bottom of the cushion, resulting in an abrupt transition from a smooth bottom to a tucked in top.

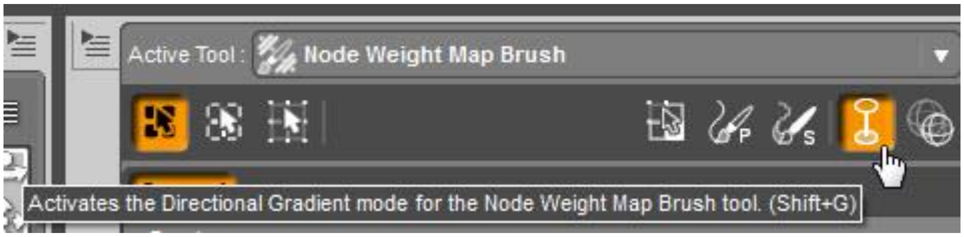


This looks OK from a higher viewpoint but we can do better.



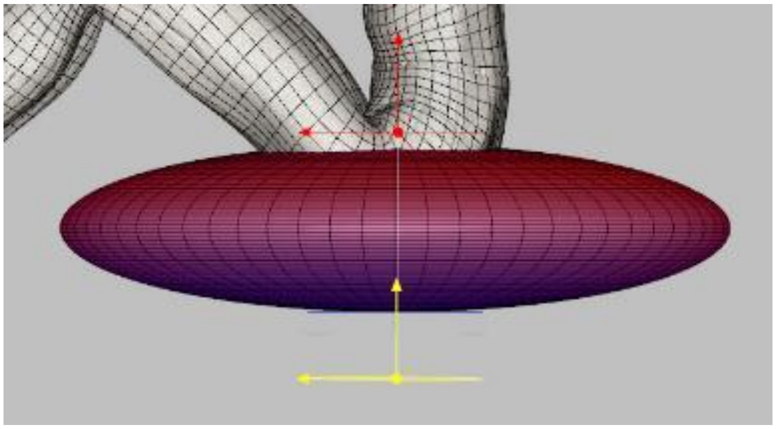
l. Delete the weight map by right-clicking on the words Influence Weights in the Tool Settings pane and choosing Delete Selected. Click on the Add Map button to re-add the weight map (this is just a quick way to clear out any old settings and start fresh).

m. Use the Side or Front view. Click on the Directional Gradient Mode (the fourth of the far right icons at the top of the pane).



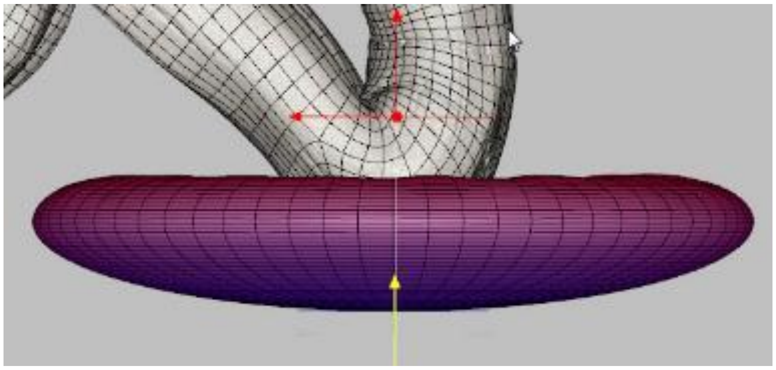
Activates the Directional Gradient mode for the Node Weight Map Brush tool. (Shift+G)

n. This tool will show red and yellow positioning icons in the viewport. Move the red dot (drag on the arrowheads) so it is slightly above the top of the sphere. Move the yellow dot so it is somewhere below. Click on the button Apply Gradient. This will give you a color gradient of red at the top (influence weight 1) to a purple-blue color at the bottom (influence weight less than 50%).

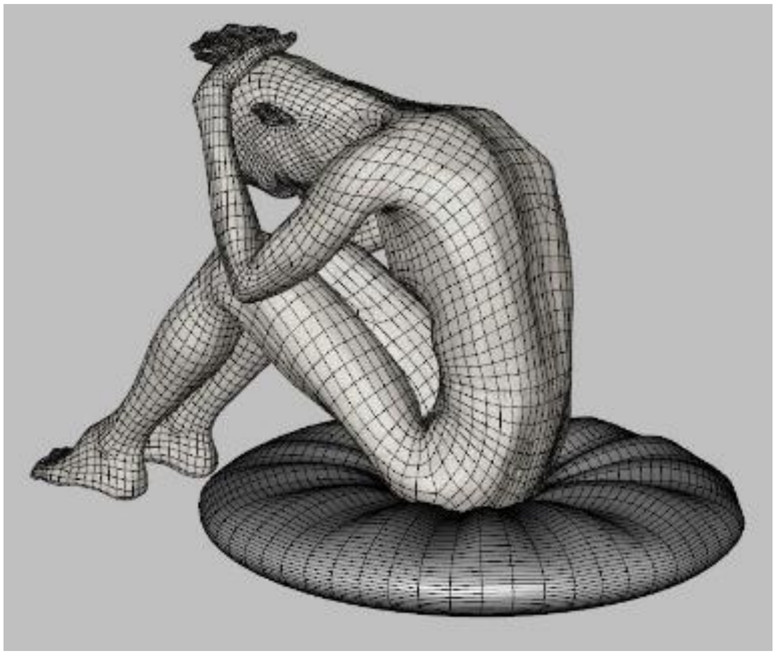


Be sure to click on Apply Gradient any time you change the position of the red and yellow icons.

o. Run the simulation. Now we have a smooth transition without the abrupt change at the midline.



Here’s how it looks from a higher viewpoint.

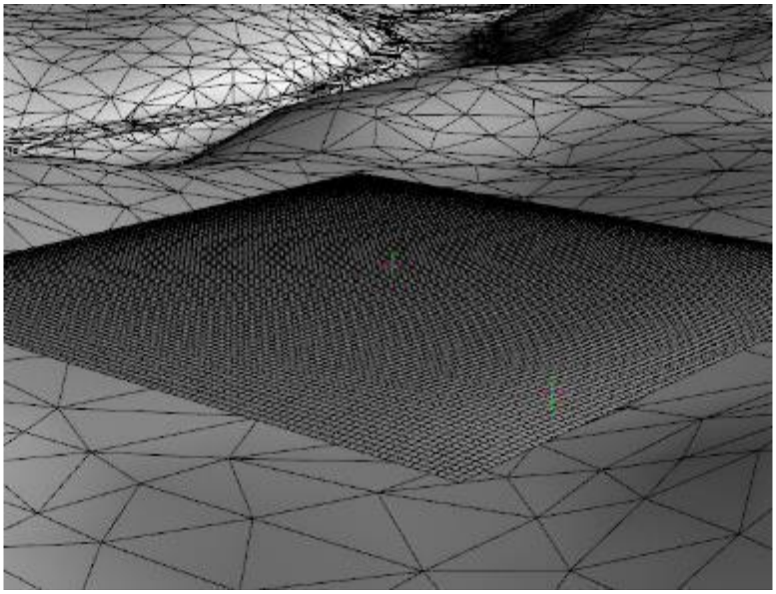


32. **Footsteps in the Snow, Sand or Mud.** When a person walks along a beach or through the snow or mud, they leave footprints. Most 3D scenes do not show them as they are hard to build into a mesh, especially considering they could go just about anywhere. Here I show how to use dForce to create them in place. This approach uses a primitive plane, the animated timeline, a large setting on the Collision Offset surface property, and a push modifier.

    a. Create a primitive plane of size 20 feet and 100 divisions. The more divisions the better to capture the shape of a footprint, but too many slows simulation time.

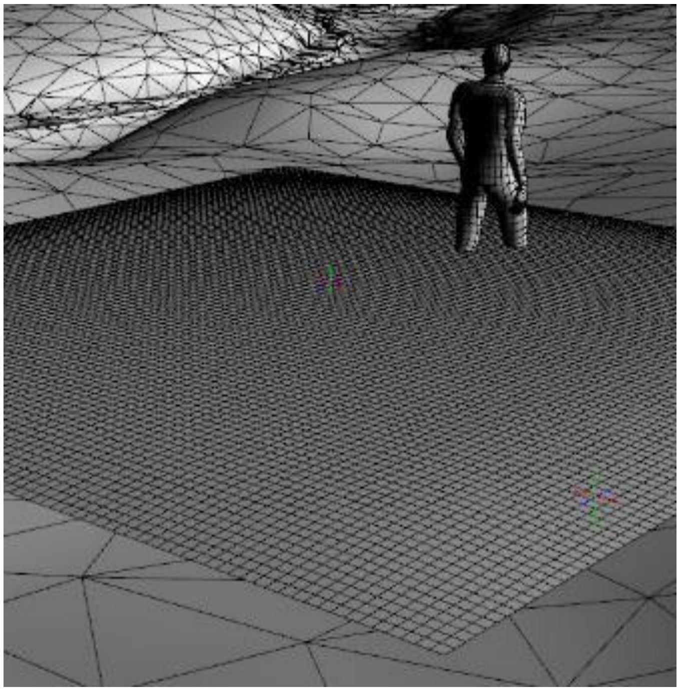
    b. Add a dForce dynamic modifier to the plane. Select the plane's surface, and in the Surfaces pane, set Simulation/Collision Offset to 6 (default is 0.2). This will keep the plane hovering over the ground when the dForce simulation is completed.

    c. Position the plane above a suitable landscape. I used the ground from Forest Winter which has a snowy texture and slightly rolling surface. Be sure the entire plane is above the surface of the ground.

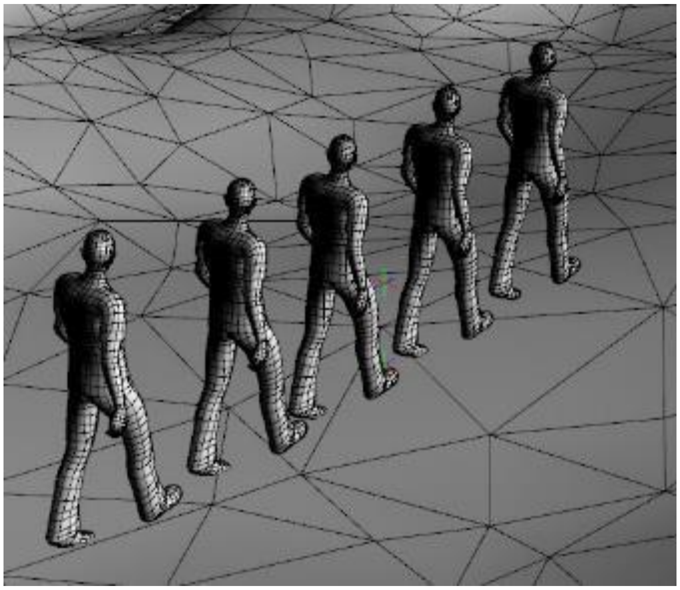


    d. Add in a figure with a walking pose. You can use any figure but I used Lorenzo LoRez as it is low-poly and you need multiple copies, then replaced him at the end with a G3M figure.

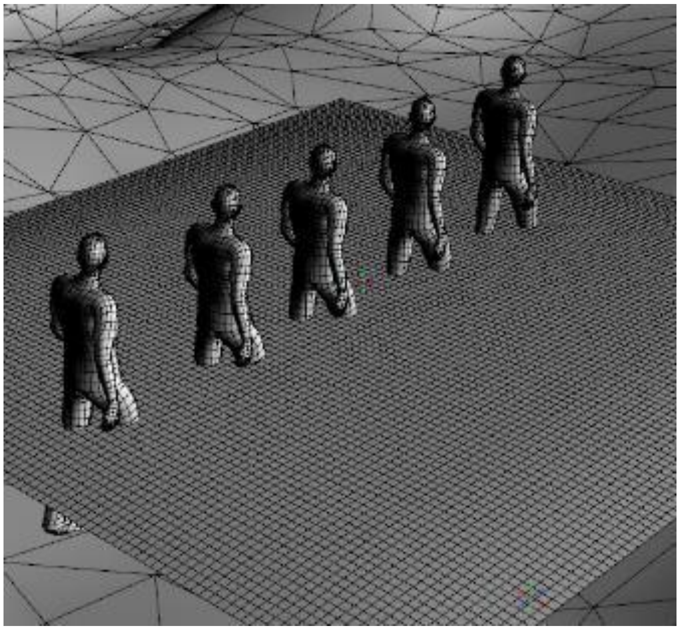
    e. Set the first figure at the end of the path you want to create. Go to the main menu and choose Edit/Duplicate/Duplicate Node Hierarchies. Do this 4 times so you end up with a total of 5 figures. DO NOT CREATE INSTANCES – dForce ignores them during simulation.



f. Position the 5 figures in a line at an appropriate spacing for the pace. Turn off the plane and set each figure at ground level or slightly above. Make a group of the figures for easier manipulation later.

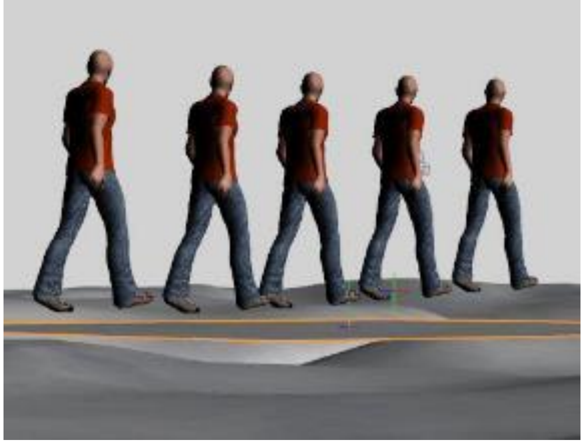


With the plane turned back on.

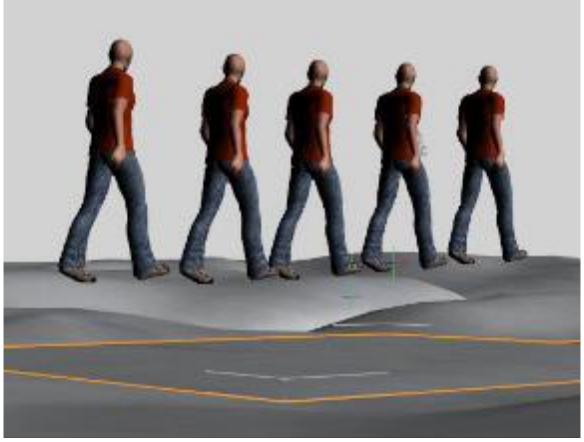


g. Go to your Timeline pane. Set the counter at Frame 30. Select the group of figures and click on the Create Keyframe icon. Go to Frame 15. Raise the group so all figures are above the plane you positioned in step c. This will also set the group's Frame 0 height. Be sure there is no timeline movements for the plane – dForce will bring it to the ground. As the simulation runs, the plane will fall first, then at frame 15 the figures will start to fall. This is important as you want the plane to conform to the shape of the ground before the figures push down to create the footprints.

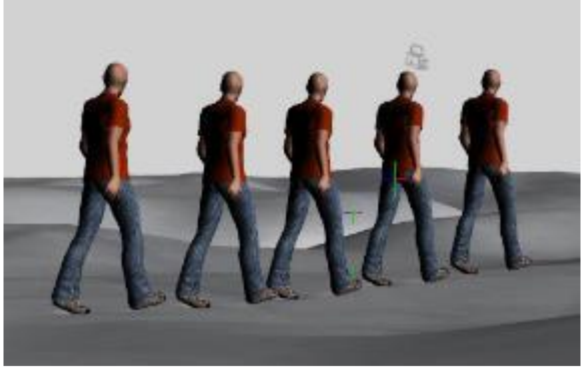
Frame 0: figures and plane above the ground



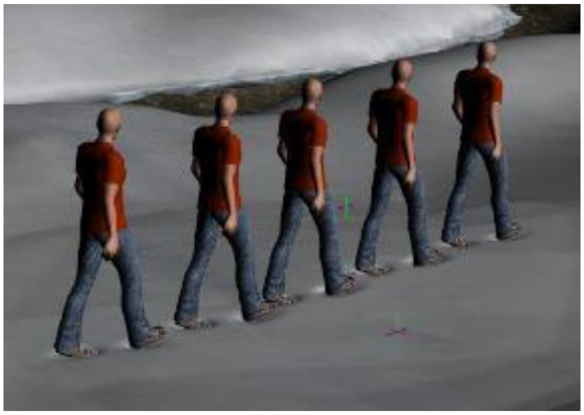
Frame 15: plane draped on ground, figures still above



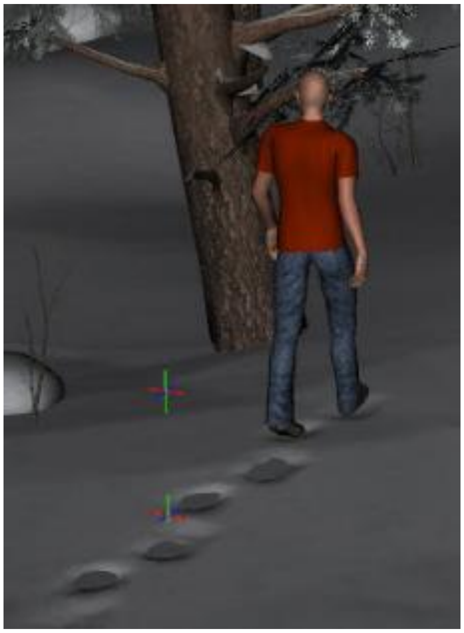
Frame 28: figures just making impressions

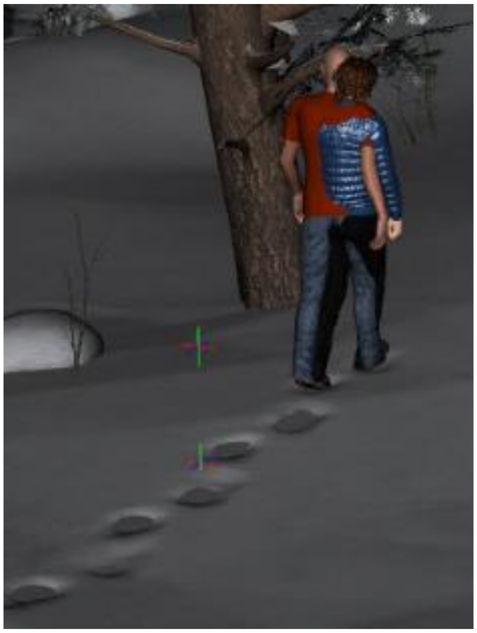


Frame 30: figures on ground forming footsteps



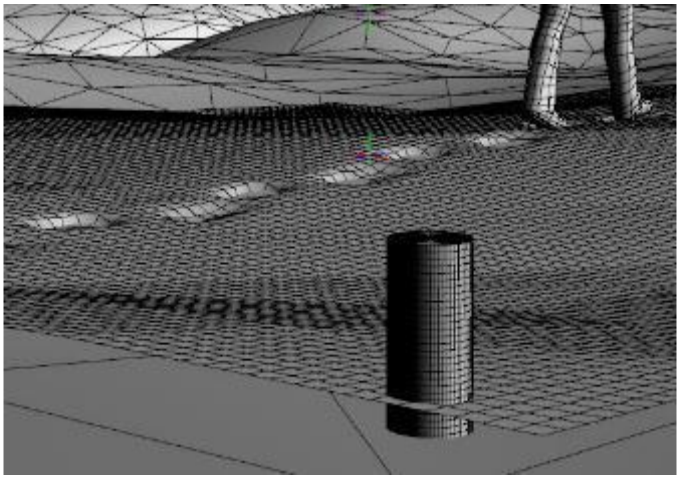
h. After the simulation, hide the 2nd through 5th figures, leaving only the first one in line. If you used a low resolution figure for the simulation, you can now load a G3M with a similar walking pose and position it in place of the low resolution figure. Be sure to adjust the legs so the feet match the position of the original figure.





i. For some camera angles, this is sufficient. But if you can see the edge of the plane, it will be hovering above the ground and can cast a shadow or show items poking through (see image).

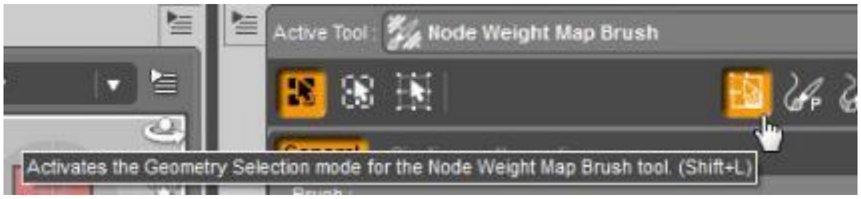
To correct this, we will add a push modifier and use a weight map to blend the edges of the plane into the ground.



j. Select the plane and from the main menu choose Create/New Push Modifier Weight Node…

k. Select the push modifier in the Scene pane, then go to the Tool Settings pane and select the Node Weight Map Brush. Halfway down the tool pane click on Add Map button next to the box that will show <New Push Modifier>. Set the displacement value of the new push modifier to -6 (to compensate for the +6 Collision Offset we used earlier).

l. Go to the top view. Click on the Geometry Selection Mode (the first of the far right icons at the top of the pane). Right click in the viewport and choose Geometry Selection/All. Next choose Weight Editing/Fill Selected… and enter a value of 0%.



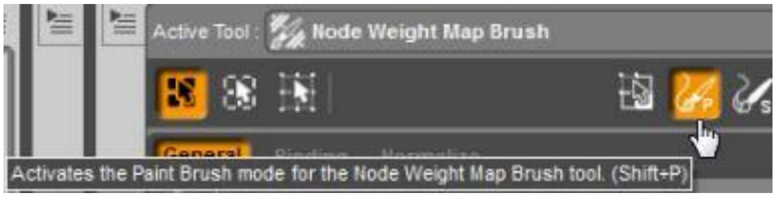
Active Tool : Node Weight Map Brush

Activates the Geometry Selection mode for the Node Weight Map Brush tool. (Shift+L)

m. For the following steps, access the pop up menu by right clicking in the viewport and then selecting the appropriate command.

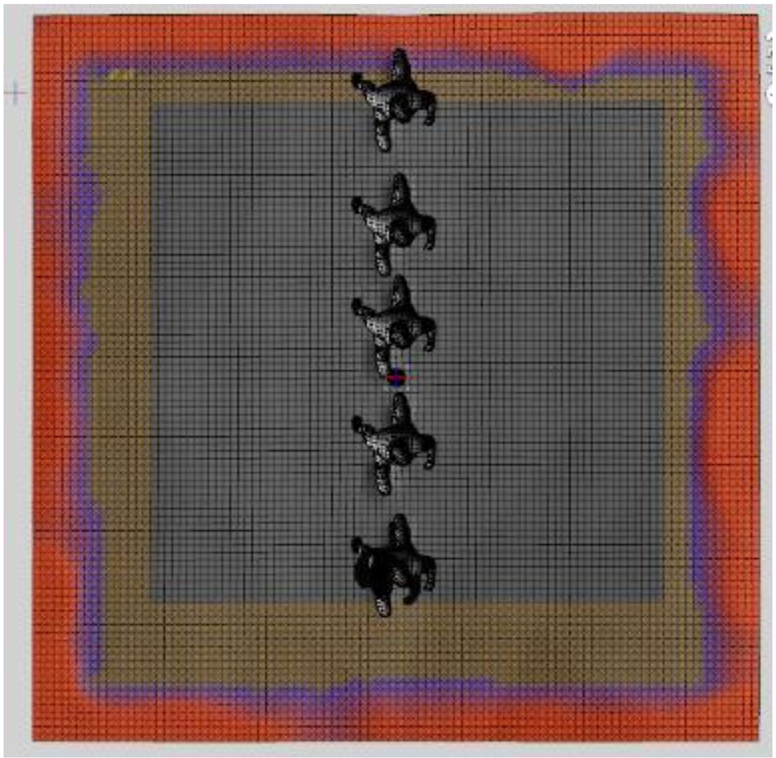
n. Set Selection Mode to Marquee Selection. Hold down the left mouse button and select the central 75% of the plane. Choose Geometry Selection/Invert Selection so now the outside perimeter is selected.

o. Click on the Paint Brush mode (the second of the far right icons at the top of the pane). Set the sensitivity to 0.6 and paint a red edge along the outside of the plane.

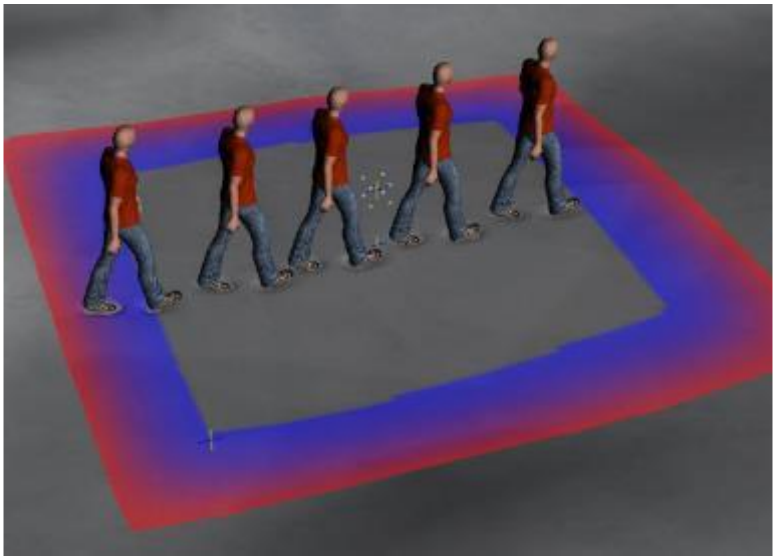


Active Tool : Node Weight Map Brush

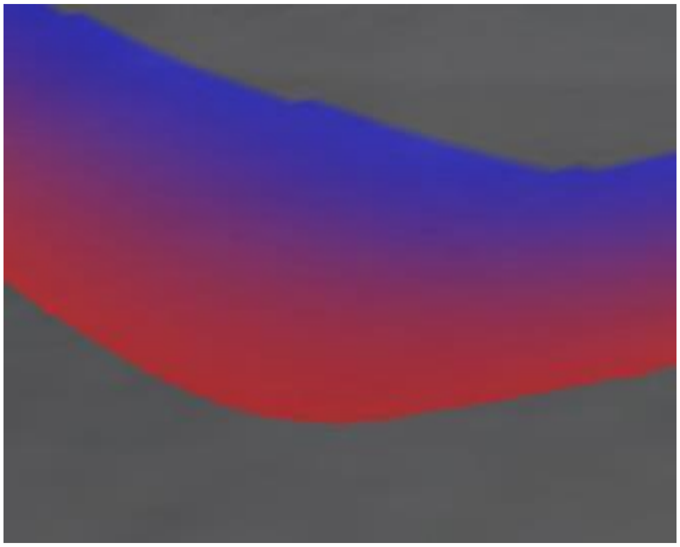
Activates the Paint Brush mode for the Node Weight Map Brush tool. (Shift+P)



p. Choose Weight Editing/Smooth Selected… and enter a value of 50%. This will smooth your painting from the edge into the middle of the selection.

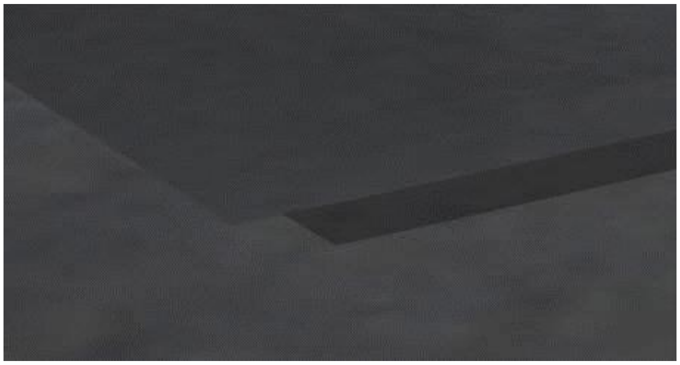


q. Go to Perspective view and look at the edges of the plane. Use the weight brush to touch up edges so they disappear beneath the surface. You may have to adjust the weight map value from -6 to -6.2 or more.

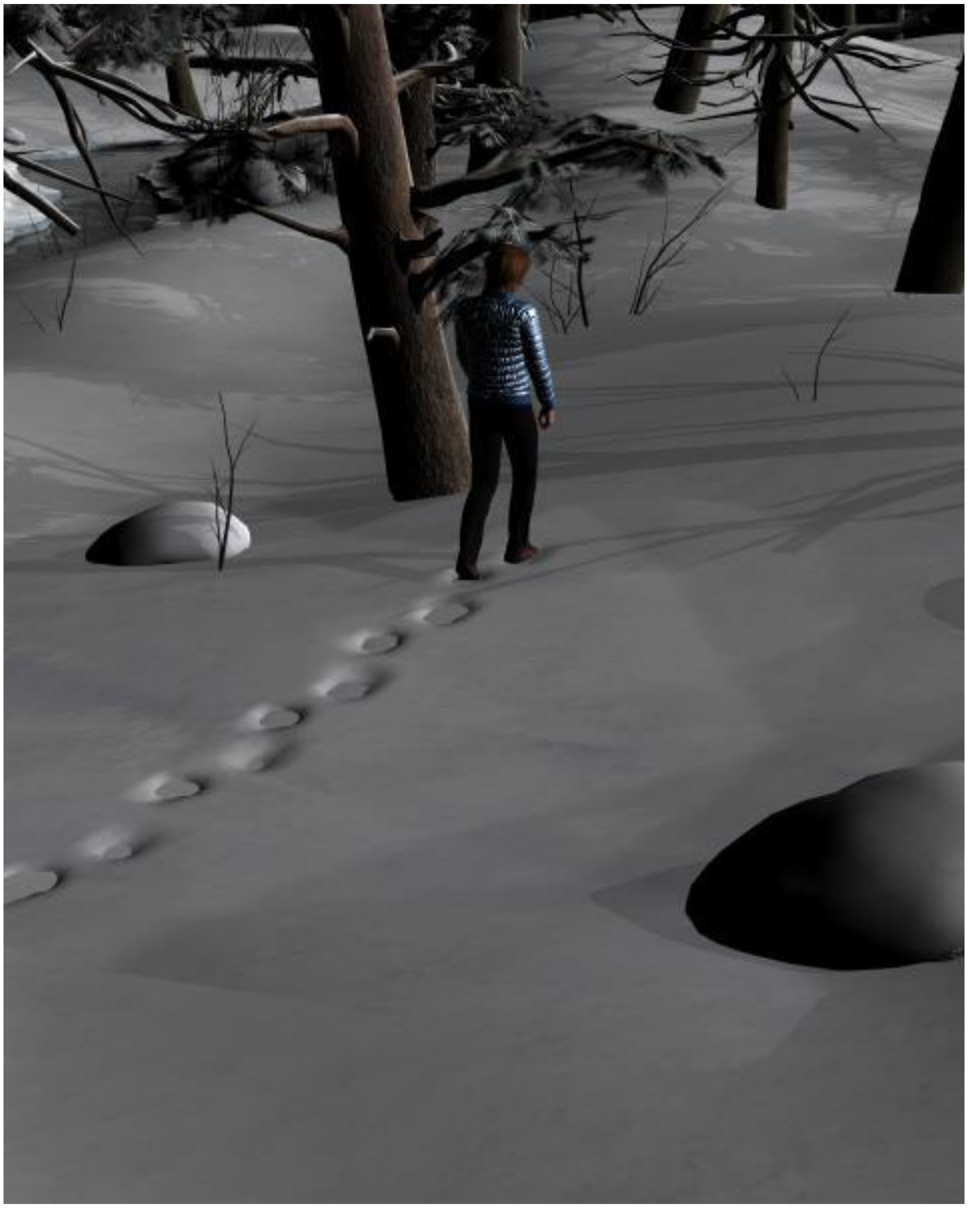


A spot render showing the plane blended into the ground.

A spot render showing what it would be like without the push modifier.



r. Copy the ground texture to your plane. You may have to adjust tiling and other surface parameters to get it to match. If all is done right, you'll have a final result that looks something like this.



Areas for improvement:

a. Some of the steps are smaller than the others. I would rotate the figure or change the right leg position on those.

b. The footsteps don't have enough of a foot shape (more important if bare feet in sand). Start with a higher resolution plane.

Since weight maps can be a bit tricky to paint, I've attached to the forum post a .duf file with a 100 division plane with the weight map I used to make this scene.

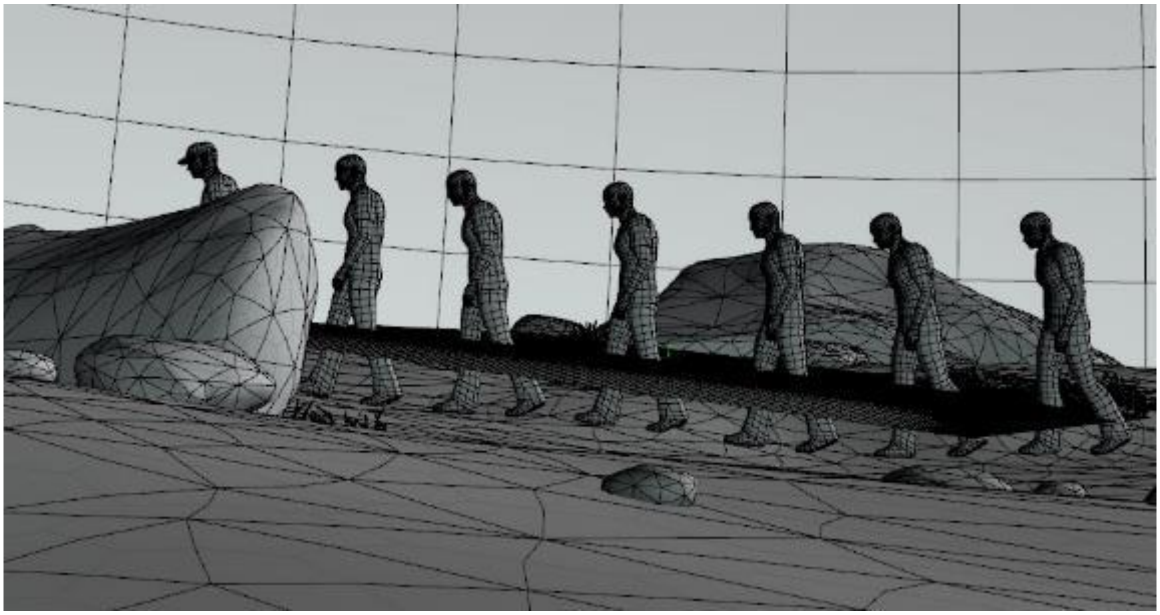
33. **More Footprints**. Here's another example of footprints using Nature - Canyon as the setting. This has a ground mesh with even more height variation than the one used in the previous section. The hardest part of creating footsteps using dForce is merging the plane's edges into the ground mesh and making the surface texture match up with the ground.

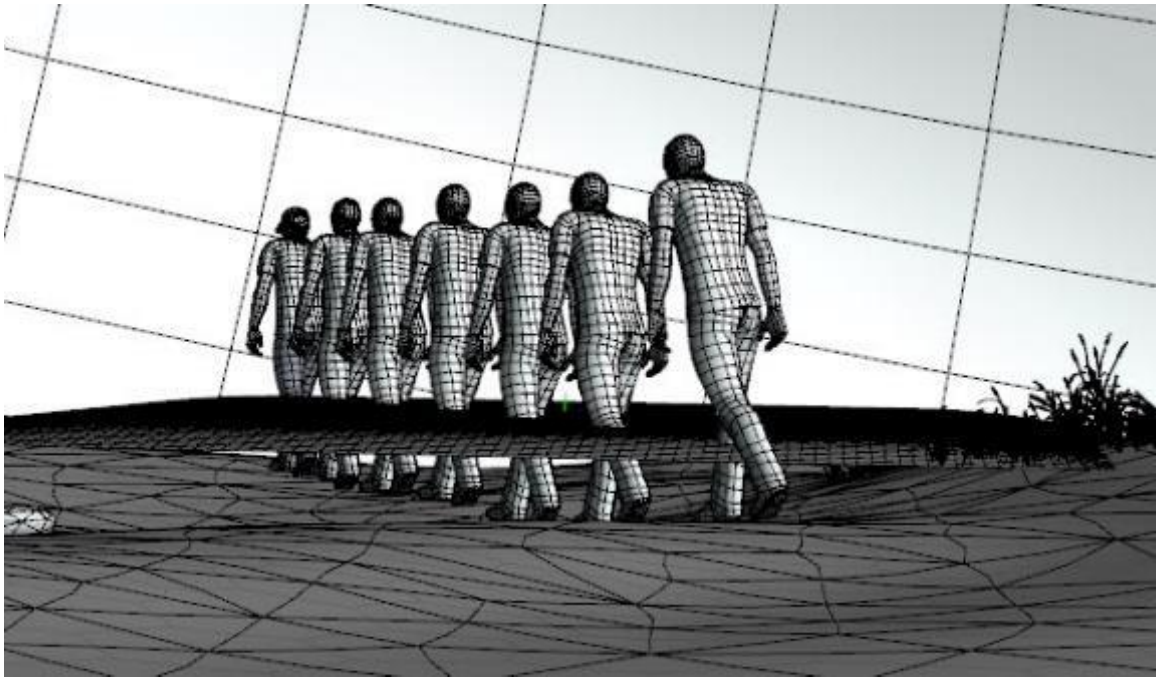
   a. Repeat the steps from 32 a-g. If you have a copy of the .duf file I posted to the Daz forum, you can skip steps a-b.

   b. I created a total of 7 figures in step 32 d-e, as I wanted a longer walk.

   c. Because of how the ground bends, I slightly rotated the plane on the X, Y, and Z axis to somewhat follow the tilt of the ground.

d. This is what the figures and plane look like at Frame 0 of the simulation.



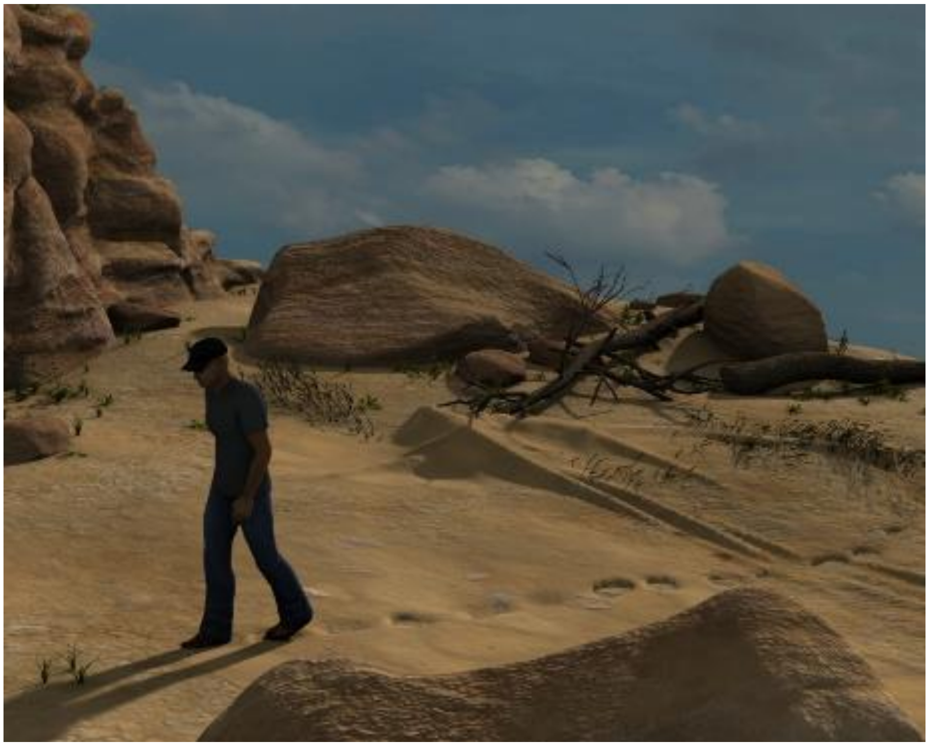


e. Run the simulation.

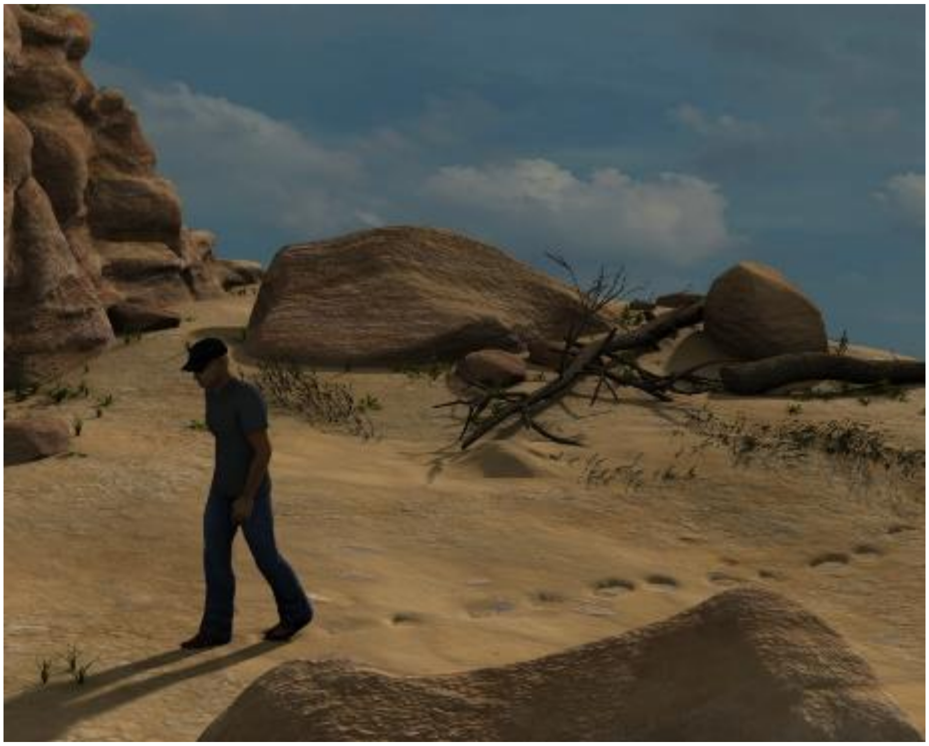
f. If you use the plain downloaded from my post on the Daz forum, you already have a weight map in place. If not, follow steps 32 j-q. I found that to hide the edges, I had to set the value for the push modifier displacement value to -10 instead of the -6 I used in step 32 k.

g. Copy the surface texture from the ground to the plain. For the Nature – Canyon set, the plane is about 1/5 the size of the ground, so I set the Horizontal and Vertical tiles to 0.2. I then changed Horizontal and Vertical Offset to match the edges as much as I could. It helps to use the Aux Viewport to see the changes to the texture as you change the offset. Also for this set, the bump is quite high and can hide the footsteps, so I reduced it by half.

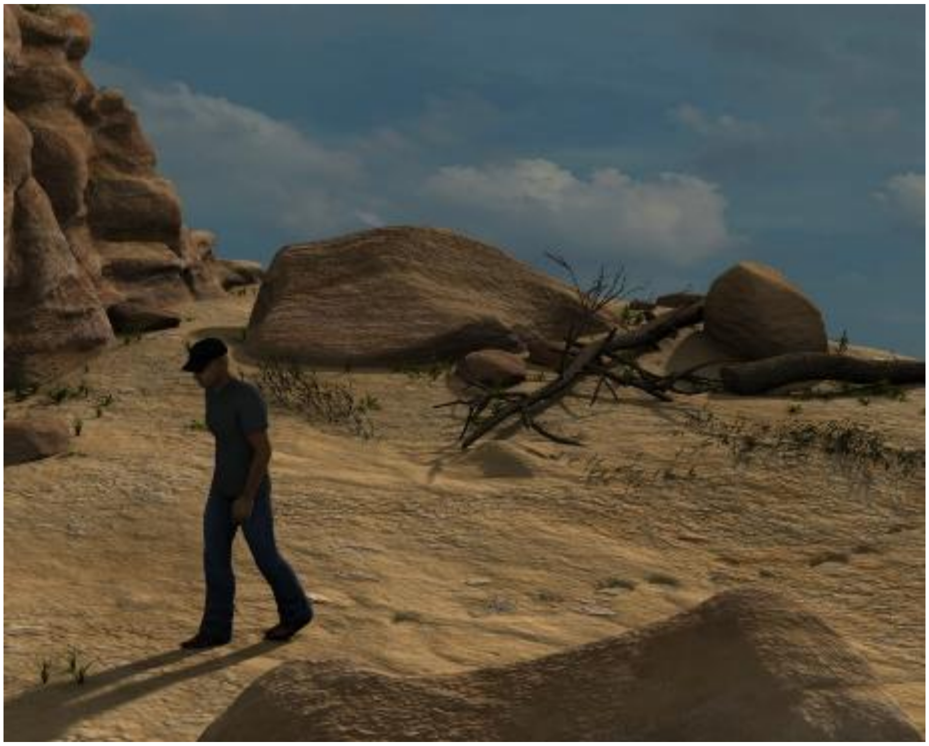
h. Here’s the result of the simulation. You’ll notice something odd: several long ridges starting in the center of the image and extending to the lower right. This is where the plane has bunched up during draping. Many of the rocks you see are part of the ground mesh and there’s a rock hidden underneath the plane in the center and it’s holding the plane up in that location. This bunching make sense when you remember that dForce treats the plane like fabric.



i. You can correct this bunching by changing four of the Simulation surface parameters. Set Stretch Stiffness and Bend Stiffness to 0.01. Set Bucking Stiffness to 0.1% and set Contraction-Expansion Ratio to 101%. This make the plane form fitting and you get the result you see below.



j. This image shows the default bump of the ground in Nature – Canyon. As you can see, the bump competes with the footsteps; they are still visible but not to the extent as seen above. Which you use depends upon your artist eye.

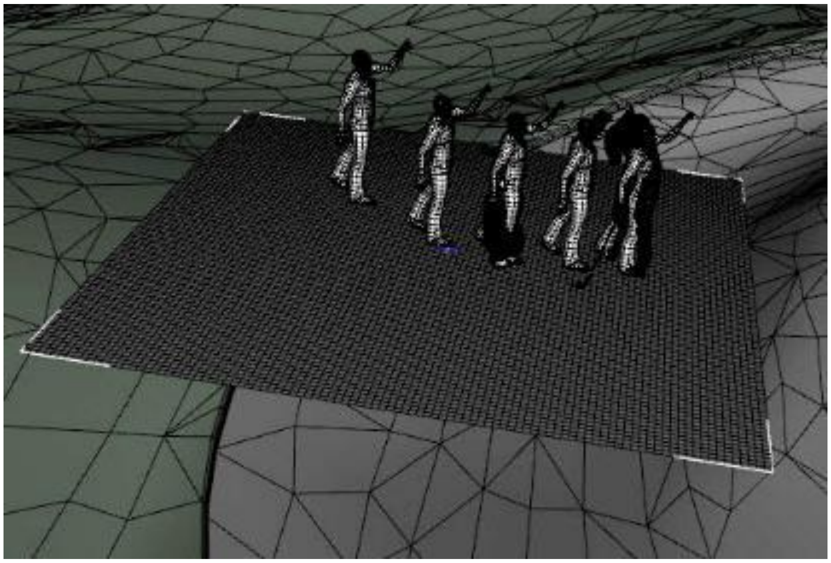


34. **Sand Trap.** The golfer's nightmare, landing in the sand trap. The sand is soft and the ball and the golfer sink in – except in Daz Studio. There the sand is stiff and hard, even if the shader gives the look of soft sand. This section uses the Golf Course Bundle: the course itself, the props (club, ball and bag), and the poses for G3M. It also uses the plane with weight map I used in sections 32 and 33.

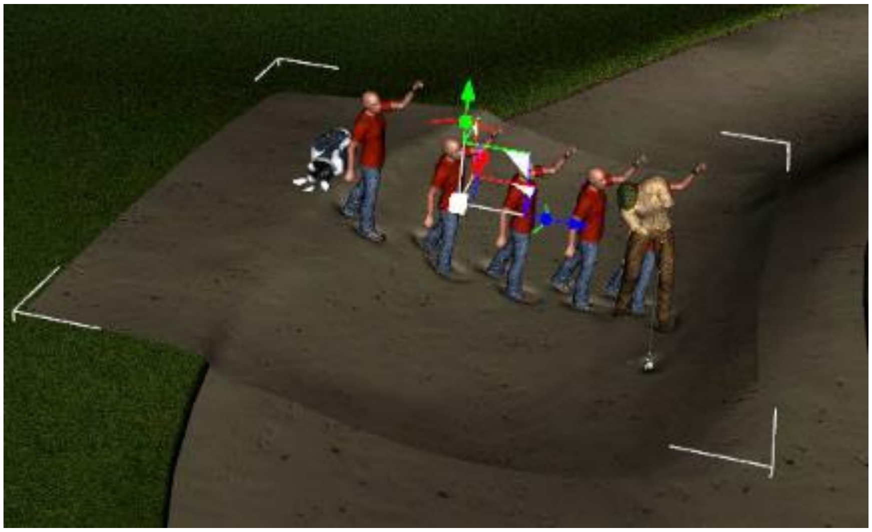
    a. This terrain is a little trickier to add footsteps to, as there's a sharp division between the fairway's grass and the sand sloping down into a pit. I also found if I embedded the plane into one edge of the sand trap, it shriveled up and moved away. Not sure why that is, and there may be some surface settings to prevent that, but in the meantime I used an alternate approach.

    b. Follow the steps from section 32. For the most part, you do the same setup and simulation.

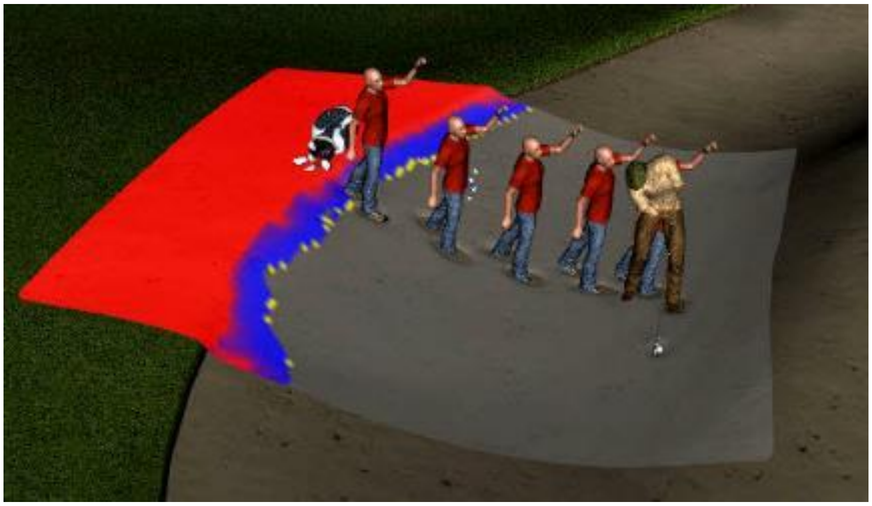
    c. I used the Lorenzo Lorez figures, positioning them from the edge of the sand to the lowest point. I then added in a G3M figure overlapping the last figure (as if you walked down, then turned in place, disturbing the sand in a larger area). The image to the right shows the Frame 0 setup. Note the plane is above the grass and sand.



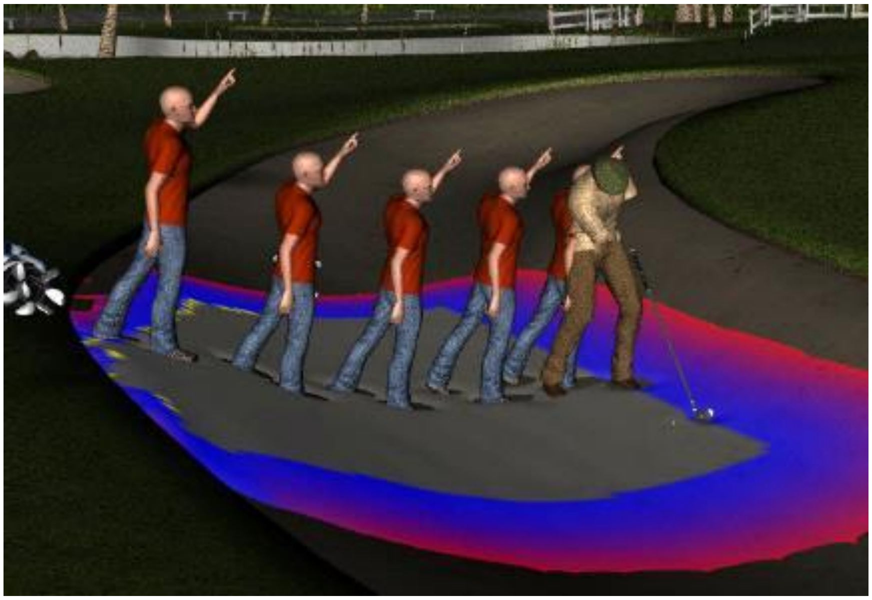
d. Run the animated simulation. You'll end up with something like this. There is no weight map activated yet, so the plane hovers slightly above the grass and sand.



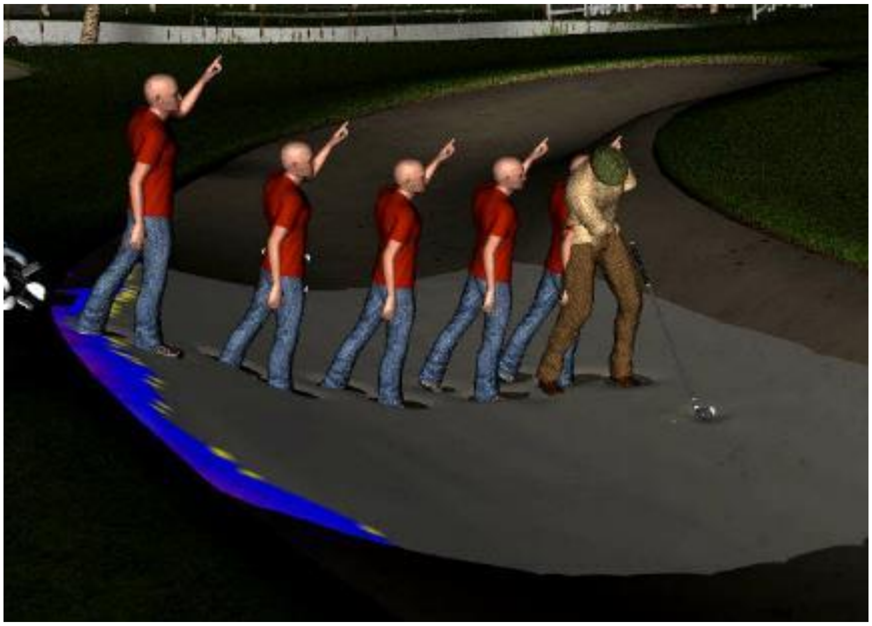
e. The part of the plane above the grass can be hidden by using the Geometry Editor. Use the Lasso selection method to choose polygons and then Geometry Visibility to hide the selection. You can do this in pieces, which is a good idea when you are working near the junction of grass and sand. In the image, the part I hid is shown in red.



f. I started with the same weight map as I did for section 32 with a push modifier displacement value of -10 to merge the sides of the map into the sand. But since I hid the area over the grass, the adjacent part of the plane did not have any weights to smooth it into the sand, so I used the Node Weight Map Brush to add some weight along that edge.

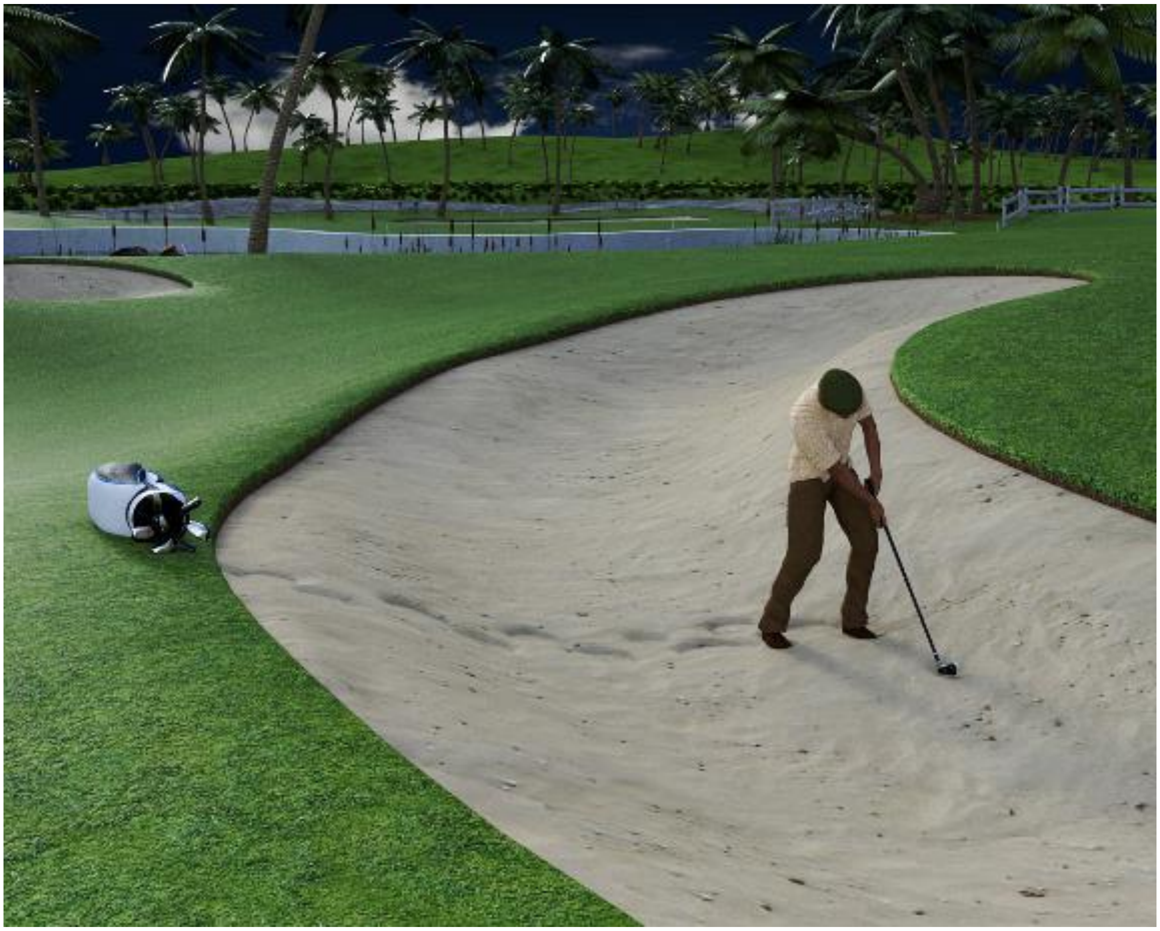


g. I found that I could not fully depress the edge with the brush, and if I set the push modifier displacement value lower than -10, the footprints started to disappear. So I added a SECOND push modifier to the plane, this time with a displacement value of -15. I only had to paint a small amount of the edge with the paint and smoothing brushes. This allowed me to depress the plane enough so a jagged saw tooth edge caused by the hidden square polygons was gone.



h. Hide all the extra figures and render.

With footprints:



With no footprints: He must have flown or jumped to avoid disturbing the sand!

35. **Creating Icicles.** Although the concept of draping cloth doesn't appear to apply to an icicle, you can use dForce to make one.

    a. Create a primitive cone of 10 inch length, 2 inch diameter, 64 segments, and 32 sides. Rotate it on the X-axis 180 degrees so it points down.

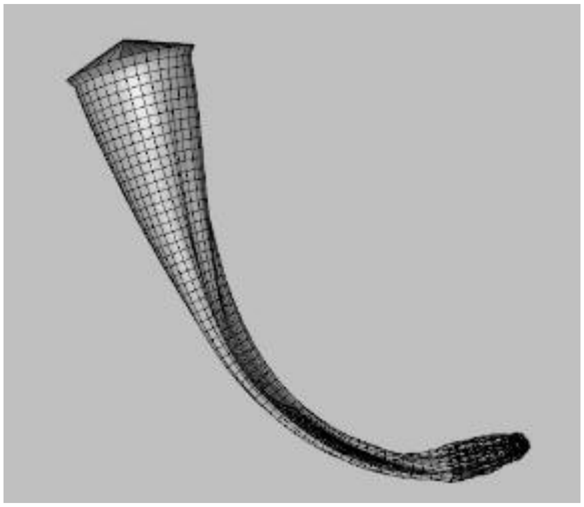
    b. We need to create two new surfaces on the cone. They will get different dForce surface properties so the mesh doesn't blow up during simulation. Using the Geometry Editor tool, select all the polygons that make up the top circular surface of the cone. Right click in the viewport and choose Geometry Assignment/ Create New Surface from Selected and give it the name Top.

    c. Next, select the bottom couple of rows of polygons at the pointed tip and create a surface named Tip.

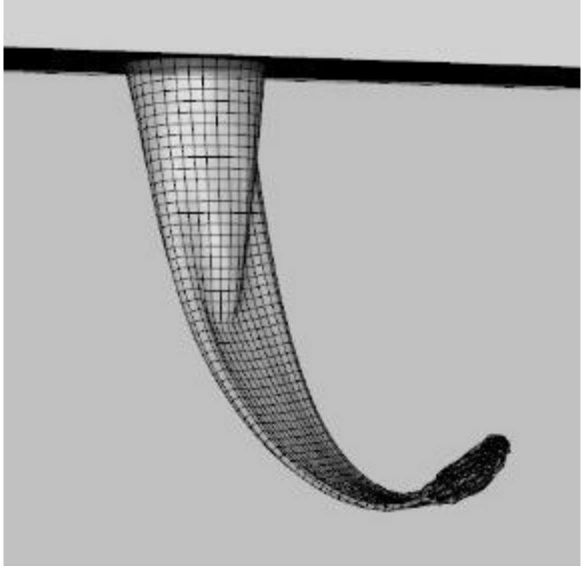
    d. Add a dForce dynamic modifier.

    e. Select the Top surface and go to the Surfaces pane and set Dynamics Strength to 0, as we don't want it to participate in the simulation. Select the Tip surface and set Stretch and Bend Stiffness to 0.1 and set Contraction-Expansion Ratio to 0.9. Select the default surface of the cone and set Contraction-Expansion Ratio to 0.9. We do this so the mesh can pull in and form around the string of spheres we are going to create shortly.

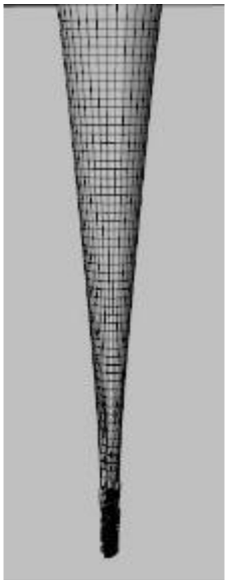
f. If you run the simulation now, you'll find the mesh wants to tilt and curve.



g. Add a cube at the top and move the cone so it touches. This will keep the top from tilting.



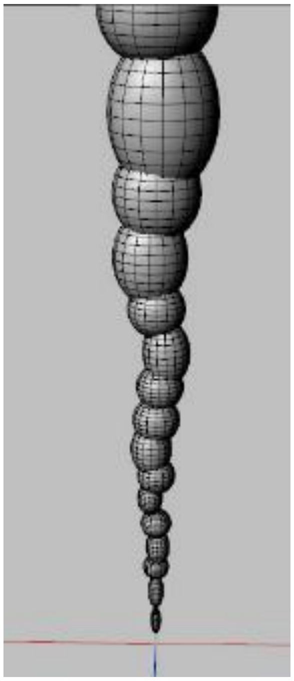
h. Set the Dynamic Strength of the Tip surface to 0.25. This will hold the tip in place and keep the cone vertical.



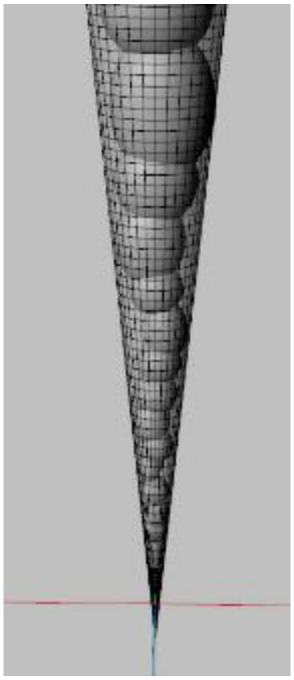
i. What we have right now is not very icicle-like. Icicles are not a smooth cone but have bumps and indentations created as water flows down the surface and freezes.

j. Create a sphere and position it at the top of the cone. Size it so it so it fits inside the cone with a little space left over. Select the sphere, go to Edit/Duplicate/Duplicate Node(s) and create a copy. Scale this a little smaller than the first sphere and move it down so the two spheres have some overlap. Repeat this down the length of the cone (I ended up with 16 spheres). Go back up the chain of spheres and elongate some. Move others slightly off-center. Then run the simulation. Add an ice shader and render.

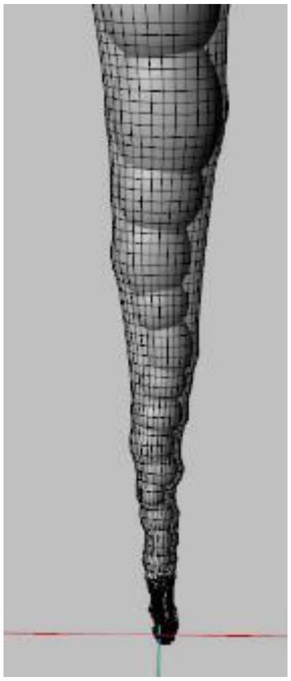
Chain:



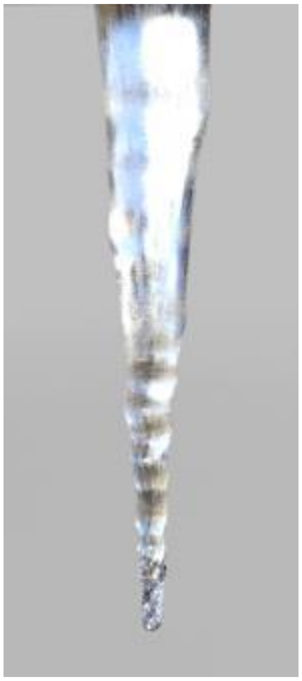
Inside Cone:



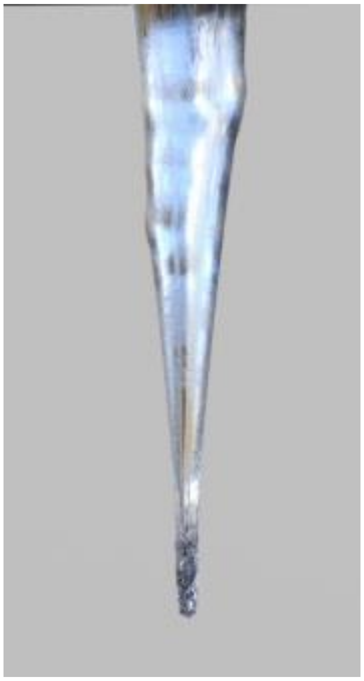
After Simulation:



Render:



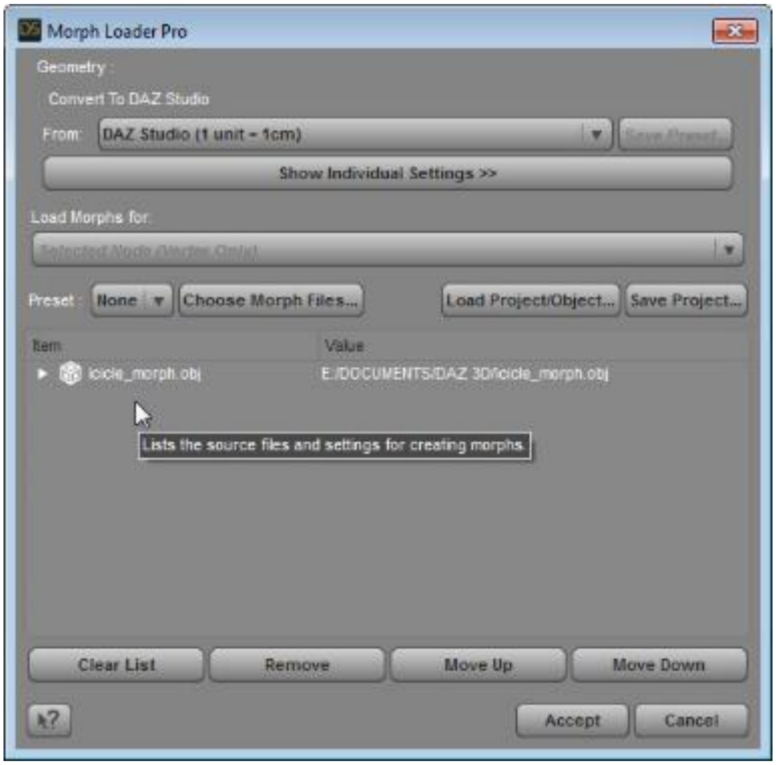
k. For a variation, hide the lower half of the spheres and run the simulation.



l. If you want multiple icicles, and you duplicate the first one, you’ll find the copy is just a straight-sided cone. You will need to duplicate your chain of spheres and simulate each new icicle in turn. Alternatively, you can change your first icicle into a morph that you can apply to subsequent cones.

m. Hide everything in the scene except the cone. From the main menu, choose File/Export and save an .obj file named icicle_morph.

n. Duplicate the first cone. With it selected, from the main menu choose Edit/ Object/Morph Loader Pro. Click on the Choose Morph Files button and pick the icicle_morph.obj file you just saved. Then click Apply.



Morph Loader Pro

Geometry :

Convert To DAZ Studio

From: DAZ Studio (1 unit = 1cm)

Load Morphs for:

Selected Node (Vertex Only)

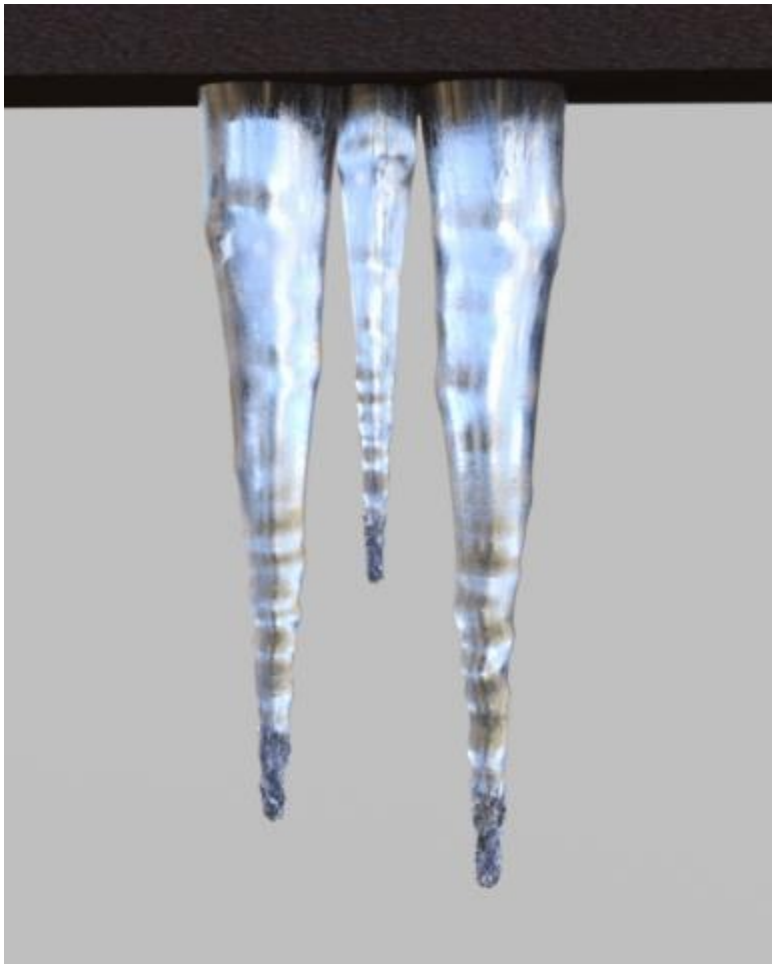
Preset : None

Item	Value
icicle_morph.obj	E:/DOCUMENTS/DAZ 3D/icicle_morph.obj

Lists the source files and settings for creating morphs.

o. In the Parameters Pane, there will now be a new section labeled Morphs. Set the icicle morph to 1.0. Now you can duplicate this icicle as many times as you want. Change the scale and rotation of the copies to give some variation.

(If the copy flips over, that’s because X or Z rotate is also being applied, just turn it back around).

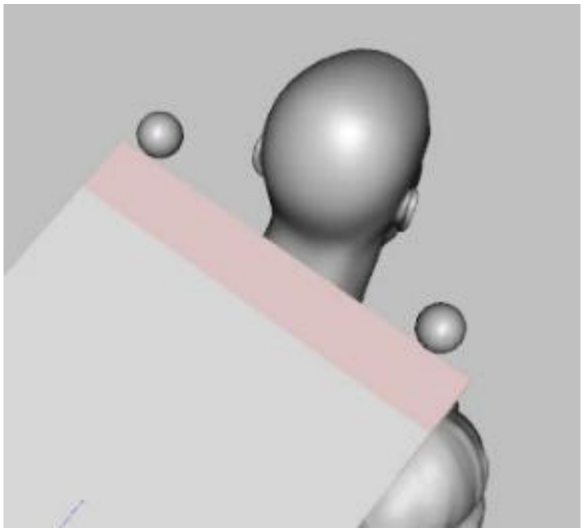


36. **Creating a Cape.** Someone asked how to make a cape. I don't know how to make clothes so there's most likely a better approach than what I show here, but this will do the basic job.

   a. Create a primitive plane of size 2 feet and 100 divisions. This will be the cape. Stretch it in the Z-direction and narrow it along the X. Add a dForce dynamic modifier to it.

   b. Create two primitive spheres of size 2 inches and 16 segments and sizes. These do not need a modifier as they will only be used to lock the cape to the shoulders. During renders, they will be hidden.

   c. We want to add a new surface to the cape so it can have a lower dForce Dynamic Strength to slow its movement. Select the plane, go to the Geometry Editor, and using the Marquee Selection Mode, select a small strip at the front edge of the cape. Give it the name Shoulders. See image below.

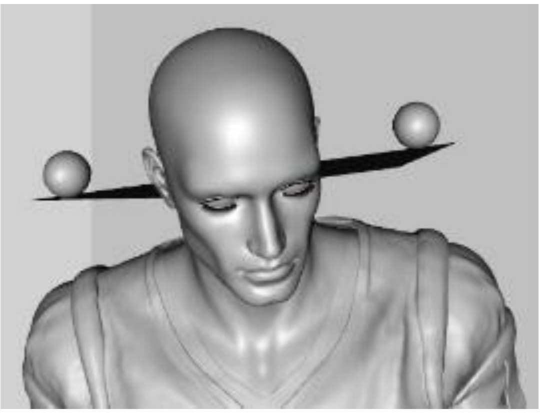


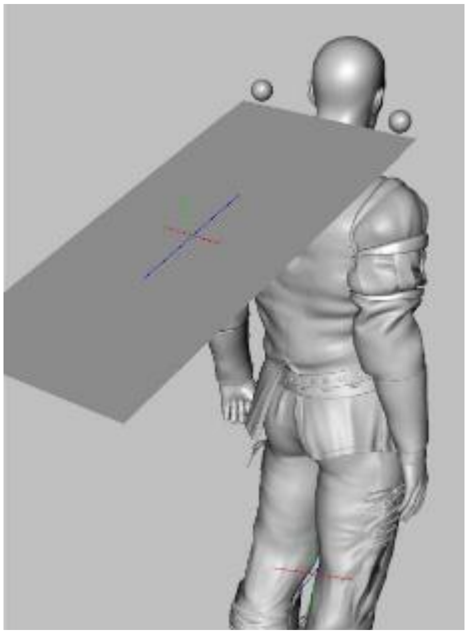
   Because we created the surface after adding the dForce modifier to the plane, you won't see the Simulation settings in the Surface Pane hierarchy for the new surface. Just add the modifier again and it will appear.

   For the Shoulder surface, set Simulation Dynamics Strength to 0.9. For the Default surface, set Contraction-Expansion Ratio to 110%.

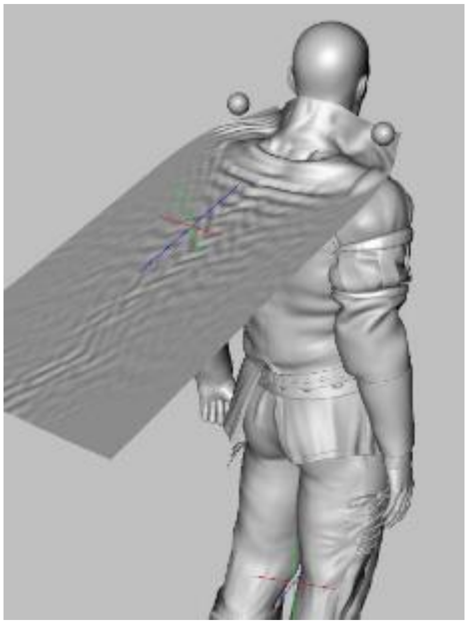
   d. Load a figure and wardrobe. I used Dmitri G3M and the Bard Outfit for G3M. Add a pose.

   e. Position the plane so it is square to the shoulders and above them a few inches, almost touching the neck. Adjust the rotations so it is parallel to the slope of the shoulders. Set the two spheres near the outer edges of the plane, above the plane, and above the midline of the shoulders. This is frame 0.

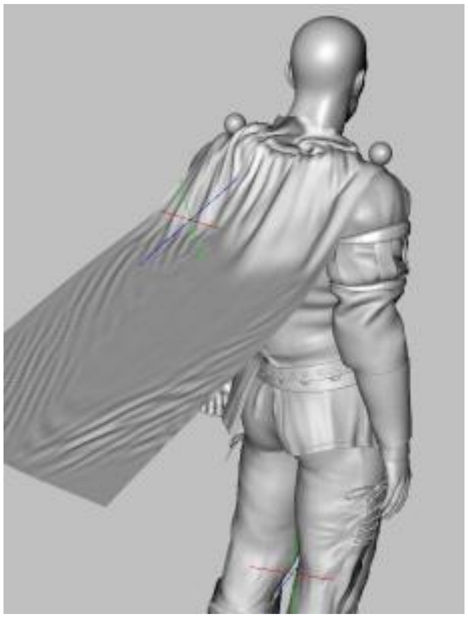




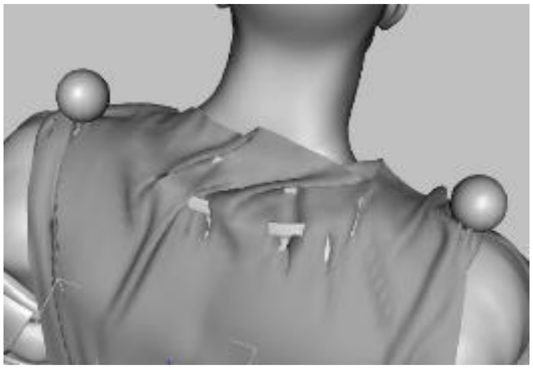
f. On the animation timeline, at frame 3, move the figure back into the plane’s mesh. Before simulation, it will look like it penetrates the neck, but during the simulation, it will actually wrap around it.

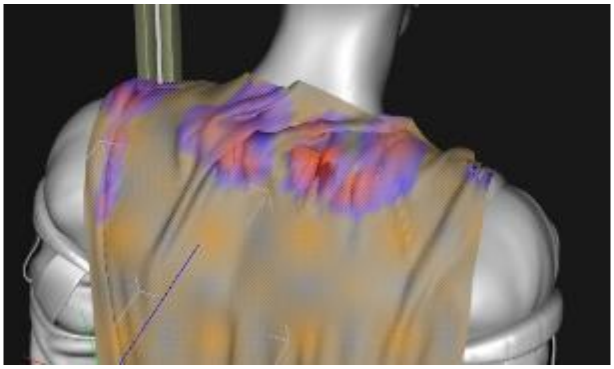


g. At frame 6, move the sphere’s down so they just touch the shoulders. By this time, during the simulation the plane will have settled down along the neck and the front edge will have slid back to the midpoint of the shoulders where they will be caught by the descending spheres and held in place. You may have to adjust which frame this occurs in as if the movement is too late, the cape will slide partially off the shoulders and down the back.

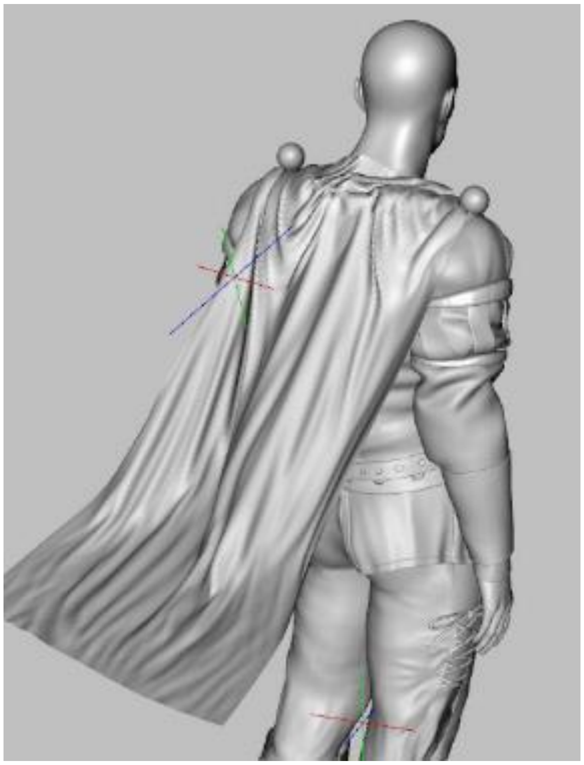


h. Run the simulation. Afterwards, hide the spheres. With this wardrobe, there was some poke-through on the back of the plane. Add a Smoothing modifier and fill it with a weight of 0. Then with the brush, paint some weight in over the spots.

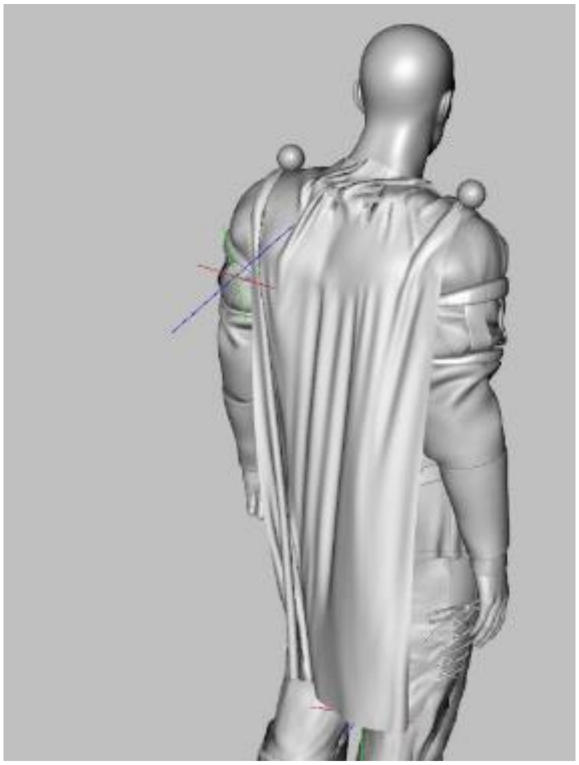




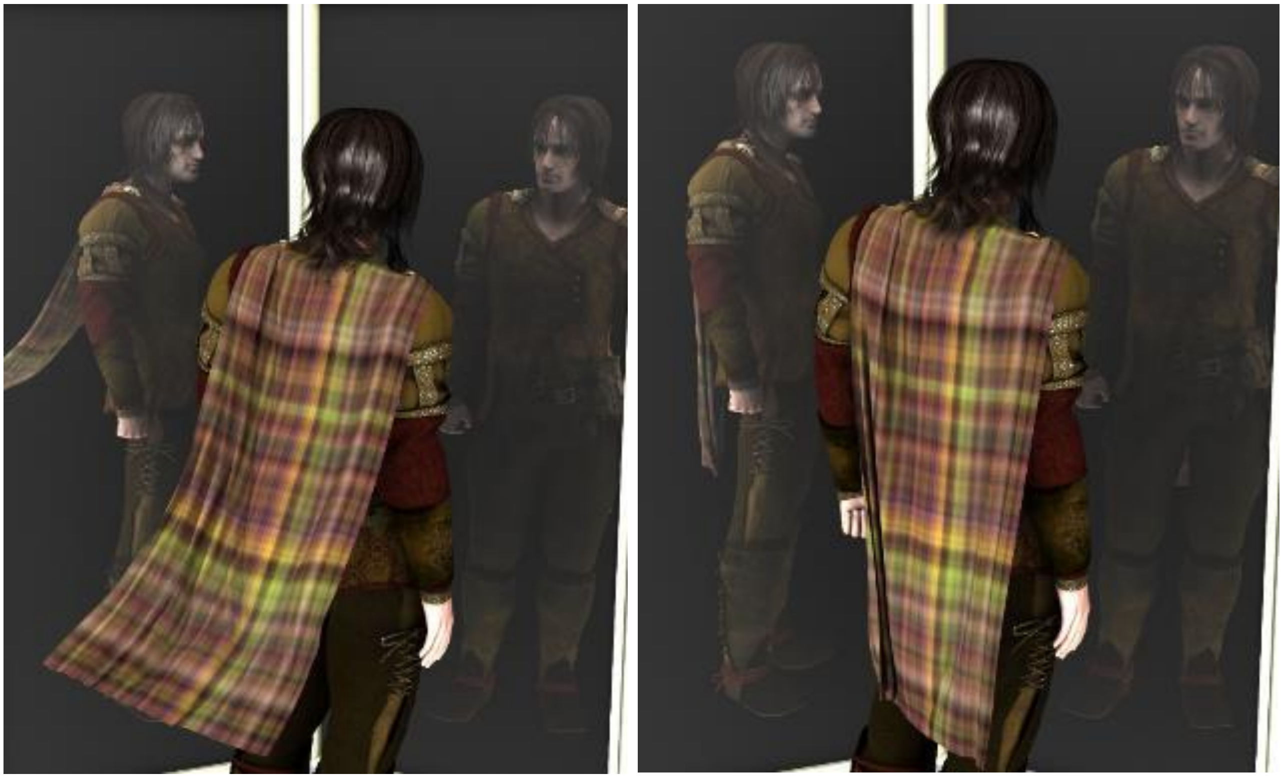
Frame 12:



Frame 30:



Renders:



i. I have the plane and spheres parented to the figure's neck, but if you change poses, you will have to reset the position and rotations of the plane and spheres. A little clumsy but doable. Here's a jumping pose render:

37. **Mummy Wrappings!** I've posted a number of items made or modified using dForce. In addition to those, I've run a lot of experiments that have not worked out. This is one that mostly works, although it could be improved upon. It took dozens and dozens of simulations before I got what I thought was an acceptable outcome.

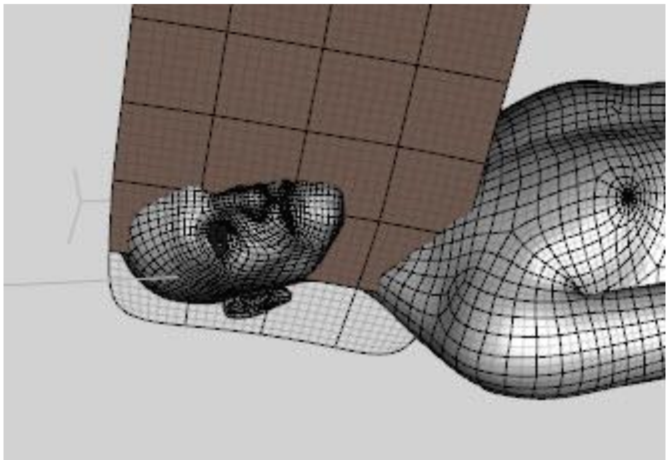
    a. Load a figure. I used G3M Abel. Add a laying pose and adjust the limbs as needed.

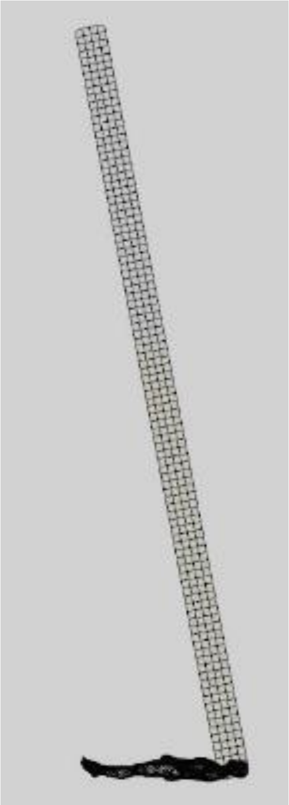
    b. Create a primitive plane of 10 yards, 100 divisions, with a Z-positive axis.

    c. Use the geometry editor to select all the polygons except for a vertical strip of 4 polygons. Hide the selected polygons then delete them.

    d. Select the four bottom polygons and assign them to a new surface called Tip.

    e. Move the plane so its Tip is embedded in the figure's head. Tilt the plane so it angles towards the feet. By varying this angle, you will change how the plane wraps around the figure and how much overlap there is between layers (see images next page).





f. Add a dForce dynamic modifier to the plane.

g. For the default surface, set Bend and Stretch Stiffness to 0.01. Set Buckling Ratio to 0.1%, Contraction-Expansion Ratio to 99%, and Dynamic Strength to 0.92. I found that a 0.95 and a 1.0 Dynamic Strength allowed the wrap to partially unwrap as the figure rotates; you may have to adjust this depending upon your setup.

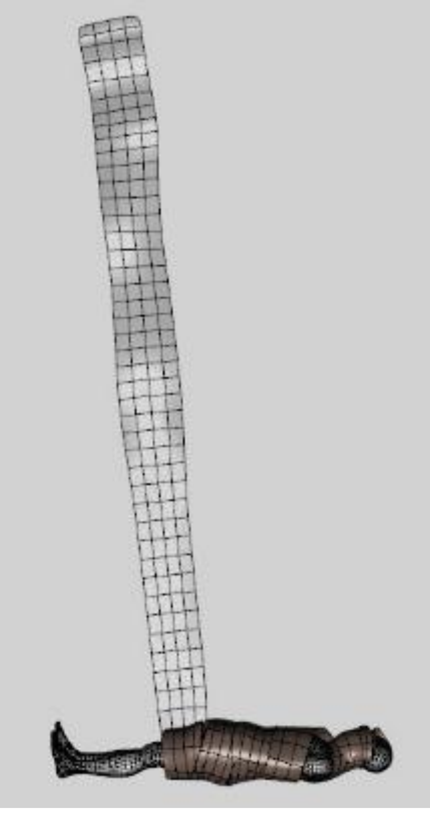
h. For the Tip surface, set Dynamic Strength to 0.9 and Contraction-Expansion Ratio to 87%.

i. Set the Simulation Settings Gravity to 0.5 and Air Resistance to 0.1. Turn Start Pose from Memorized Pose to Off, use Animated Timeline for Duration, and increase Frames Per Second Multiplier to 3.

j. I used a 30 second timeline, which ended up using about half the plane and wrapped half the figure. Double this to 60 seconds if you want to wrap the entire figure. I added a keyframe every 2 seconds, with the figure rotating 180 degrees in the same direction at each keyframe. That resulted in 7 full rotations.

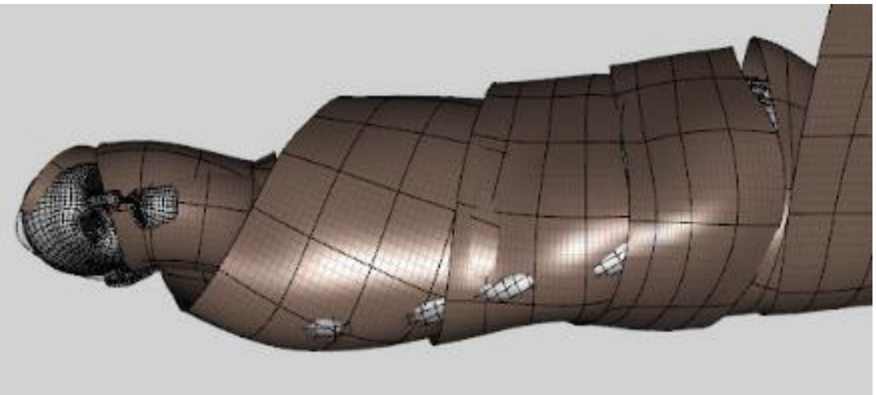
k. Run the simulation.

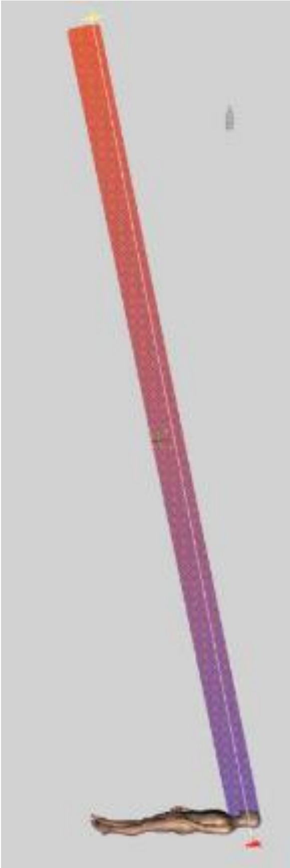
l. You can see there's a long unused strip of the plane (see image to right). I used the geometry editor to select the unused polygons and assigned them to a new surface and set opacity to 0.



m. To smooth out the wrapped mesh, I added sub-D and set it to 3.

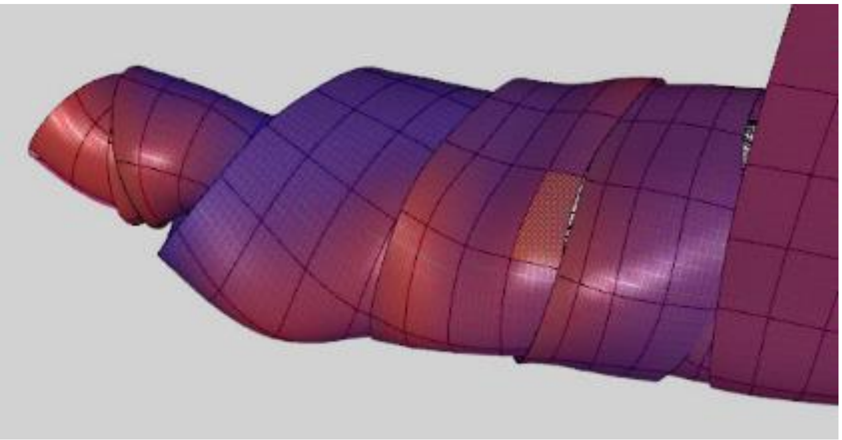
n. I found the wrap did not fully respect the figure's surface nor the underlying wrapped plane and so there was poke-through both of the figure and underlying fabric (see image below).



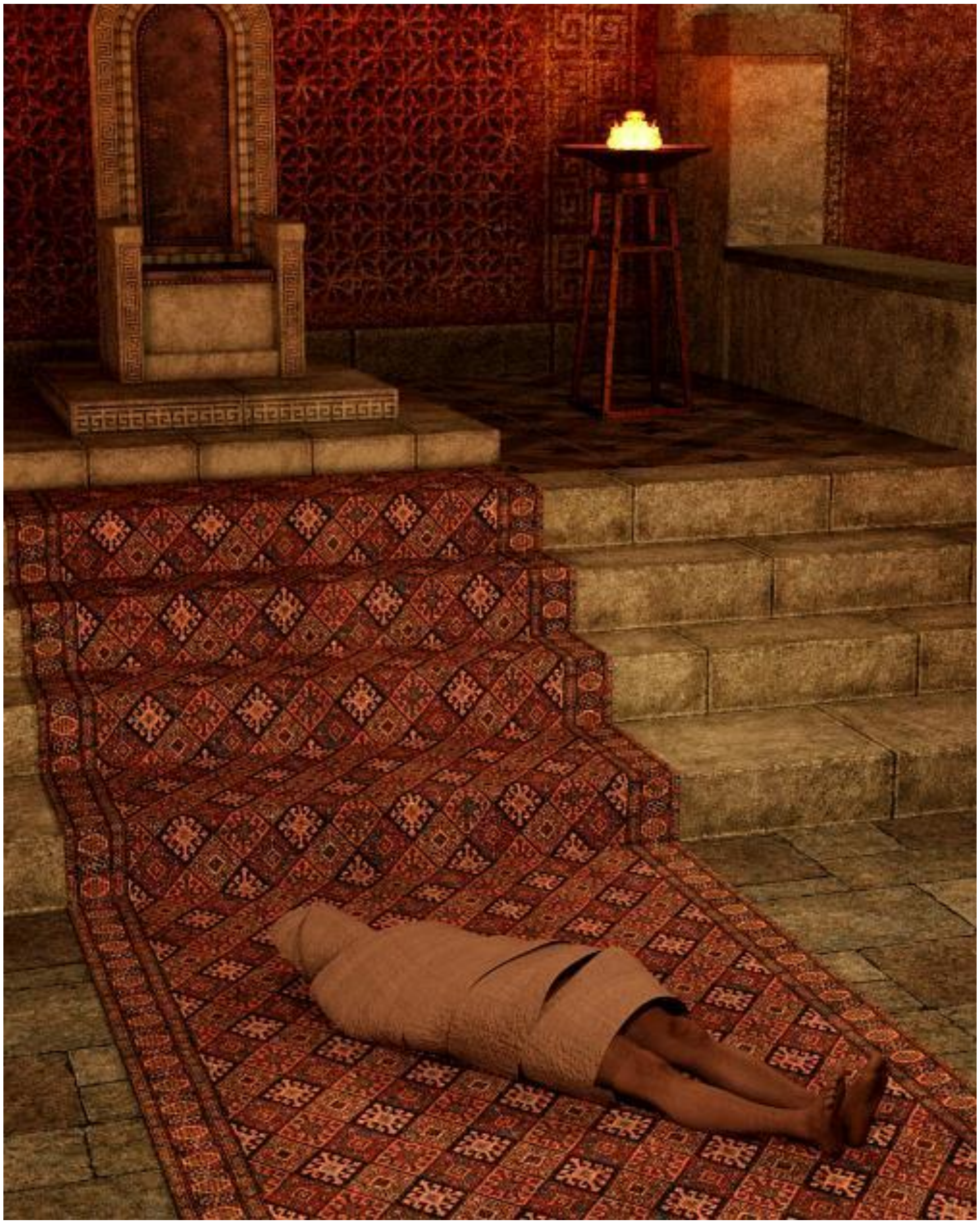


o. To correct this, I added a Push Modifier and use the spherical gradient tool to add an inverse gradient to the plane - more blue at the lowest part of the plane and red at the top. This means later wrappings have more push and that will help them sit higher than lower wrappings (see image at right). I found I had to set the displacement level to 8.5 to hide the tip of the nose.

p. I used the geometry tool on the Node Weight Map Brush tool to choose polygons on the wrapping that were out of place - either above or below the previous wrappings. I then used the painting brush on the tool to move those out or in to hide poke through. Be sure to check the box "Respect Selection" so you paint only the polygons you choose.

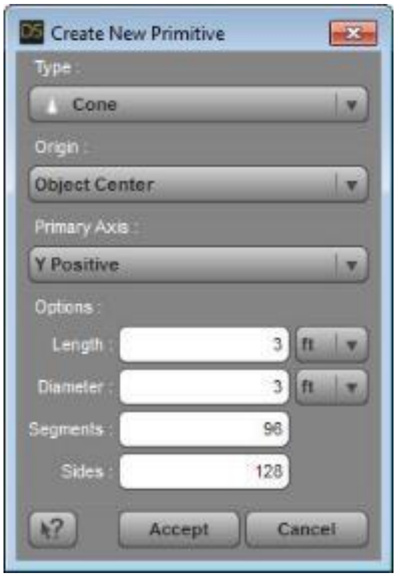


q. Here’s a finished render:



38. **Create a Skirt from a Cone**. One of the simplest clothing items you can make from a primitive object.

    a. Create a primitive plane using the following parameters. Adjust X-scale to 132% and Z-scale to 105%.



Type : Cone

Origin : Object Center

Primary Axis : Y Positive

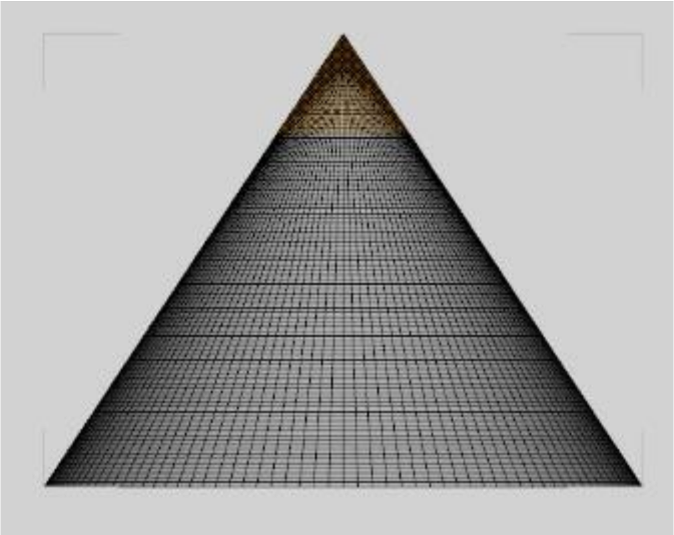
Options :

Length : 3 ft

Diameter : 3 ft

Segments : 96

Sides : 128

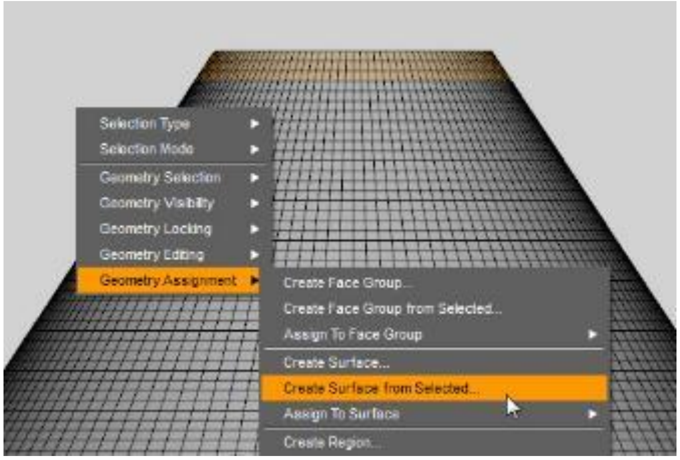


    b. Using the Geometry Editor tool and the Marquee selection method, select the top 20 rows or so of the cone. Right click in the viewport and choose Geometry Visibility/Hide Selected Polygon(s).

    c. Using the Geometry Editor tool and the Marquee selection method, select the bottom edge of the cone. Right click in the viewport and choose Geometry Visibility/Hide Selected Polygon(s). This hides the bottom surface.

    d. Right click in the viewport and choose Geometry Editing/Delete Hidden Polygon(s).

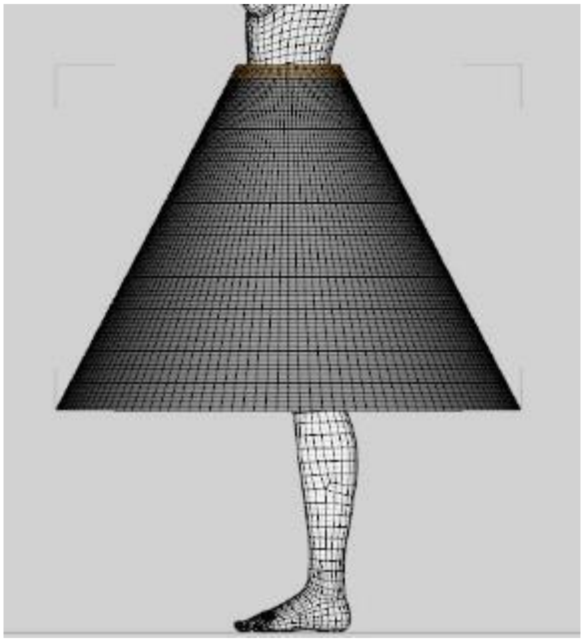
    e. Using the Geometry Editor tool and the Marquee selection method, select the top 3 rows of the truncated cone. Right click in the viewport and choose Geometry Assignment/Create Surface from Selected and label the surface Belt.



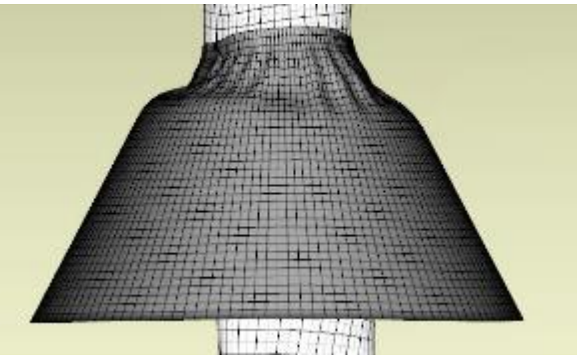
    f. Apply a dForce dynamic modifier to the cone.

    g. Select the Belt surface and set Contraction-Expansion Ratio to 62%

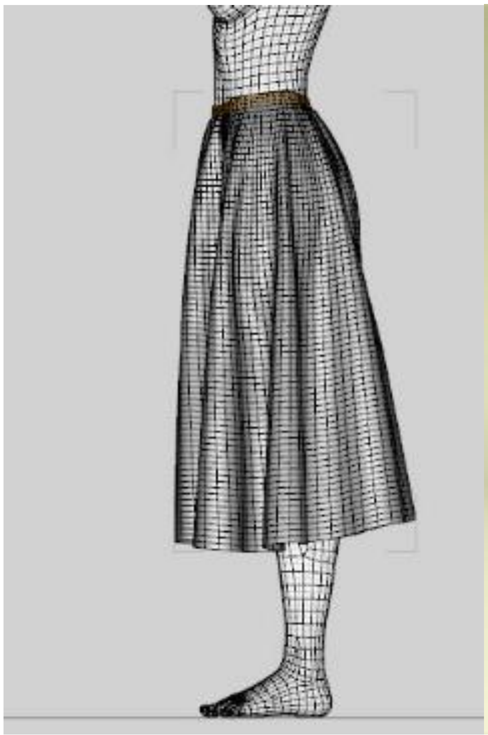
h. Load a figure (I used Genesis 3 female). Position the top opening of the cone around the figure's waist. Adjust X- and Z-translate parameters so the cone is clear of all skin.

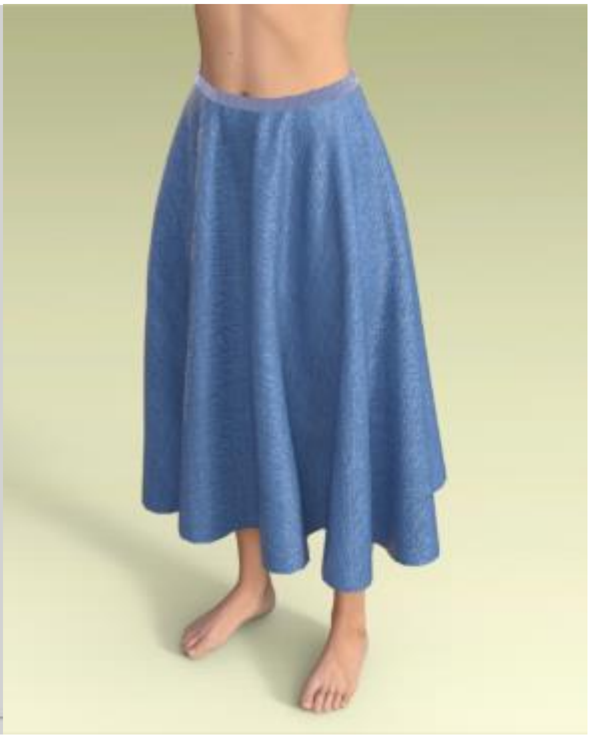


i. Run the simulation and you get this result. The belt cinches in first.



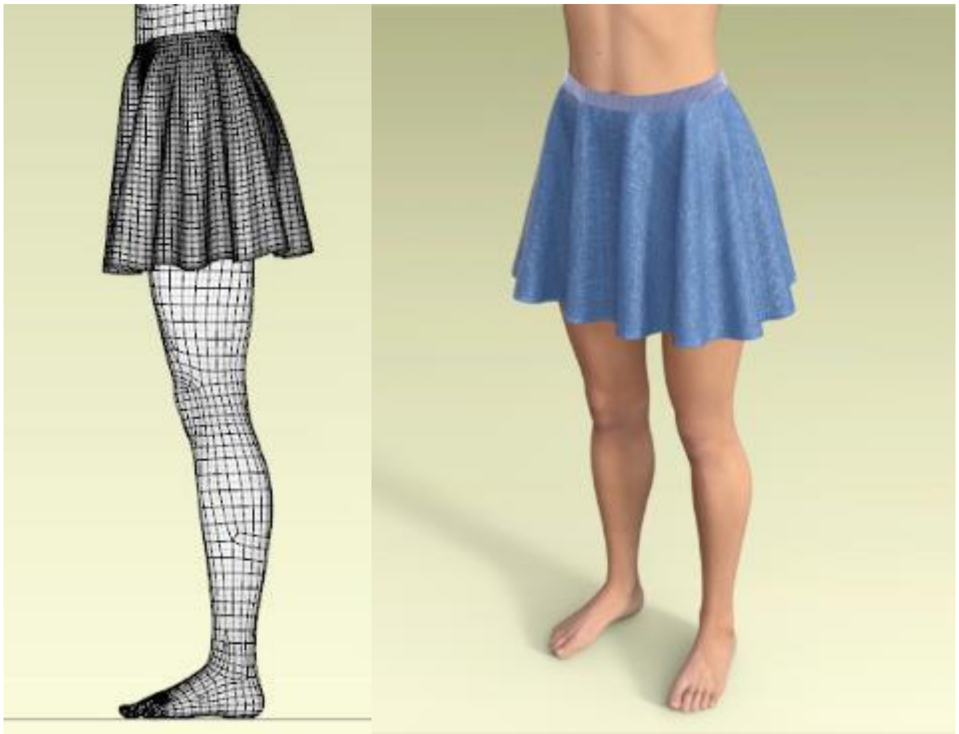
Then the simulation completes.





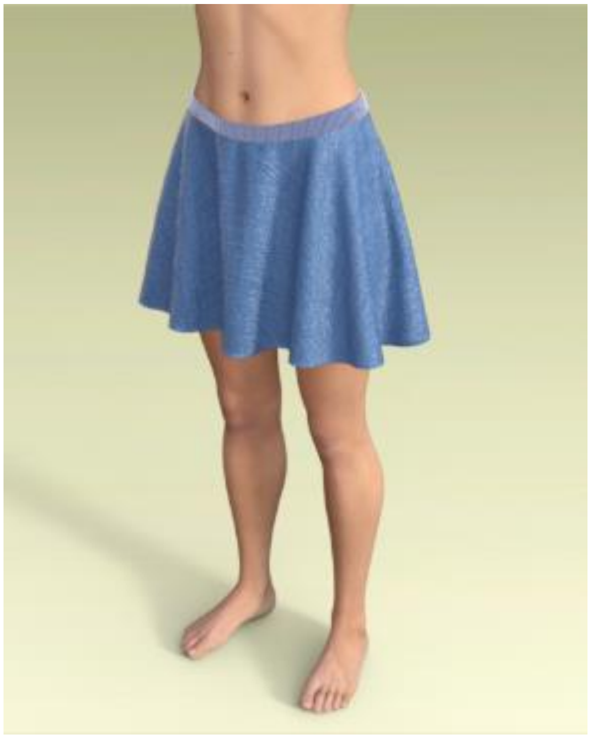
j. If you want the skirt to be more hip hugging (down lower) set Contraction-Expansion Ratio to 75%.

k. If you want a shorter skirt you might think all you have to do is adjust the Y-scale, but that does not change the slant length of the cone’s outside edge so you will get the same skirt length after simulation. Instead, use the geometry editor to choose some of the lower polygons and hide them (same process as step b).

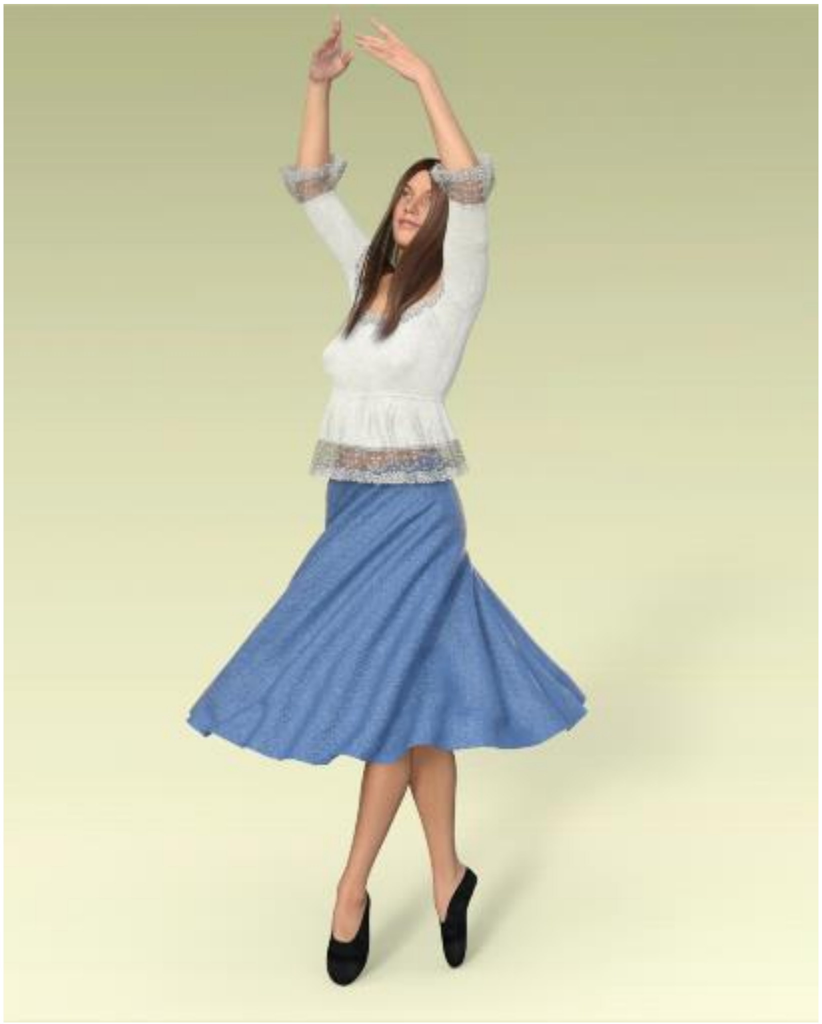


With the shorter skirt, it does not pull down as much on the belt. To get a hip hugging effect, you need to increase Contraction-Expansion Ratio to 85%.

This is also another example that shows it can be hard to predict dForce behavior, as length of skirt (or perhaps it's the number of polygons) influences what the surface property settings need to be.



l. I added a series of dance moves so the figure rotated along the Y axis. I had to reduce the density of the skirt surface to 70 to keep it from riding up too high. The result is a nice swirl and uplifting of the skirt. Gravity and air resistance are at default, so the movement is all from rotation. This render is frame 10 in the middle of the animation. If you let the rotation stop and have any relaxation time, the dress will settle back into a typical drape.



39. **Create a Bandeau from a Cylinder.** Another simple clothing item that can be made from a primitive cylinder

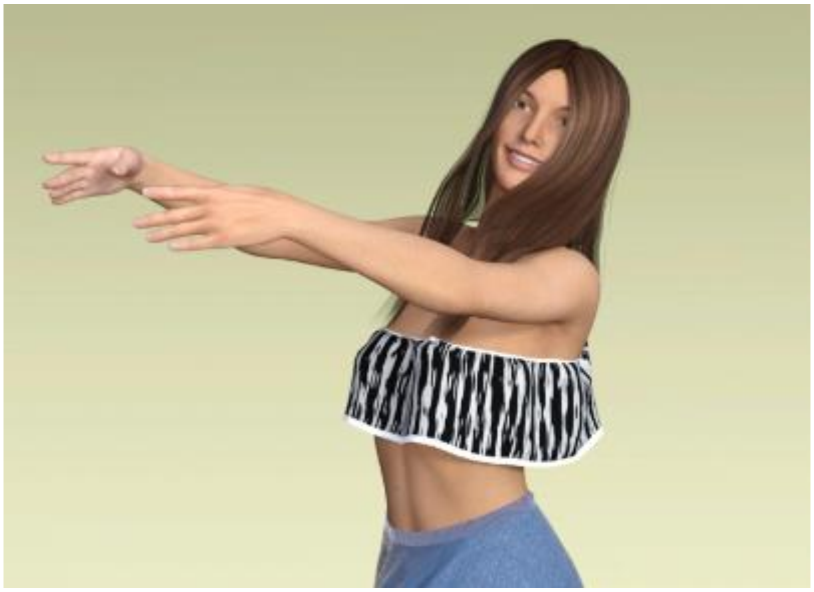
   a. Create a primitive cylinder of 3 feet diameter, 1 foot height with 50 segments and 64 sides. Position it around a figure and adjust the scale as needed.

   b. Similar to step 38 c, select the top and bottom surfaces of the cylinder, hide them, then delete them.

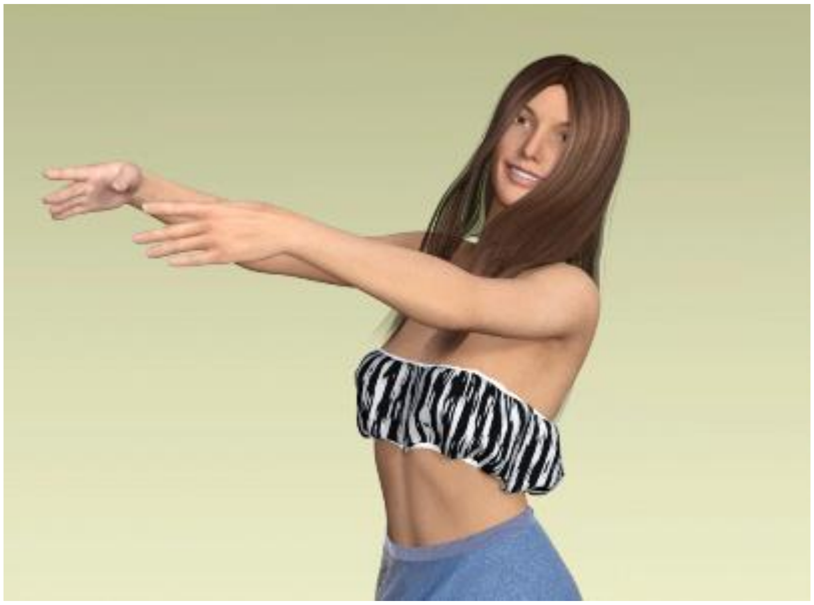
   c. Similar to step 38 e, create a new surface from the top few rows and call it Top Band. Do the same for the bottom few rows and call it Bottom Band.

   d. Add a dForce dynamic modifier to the cylinder.

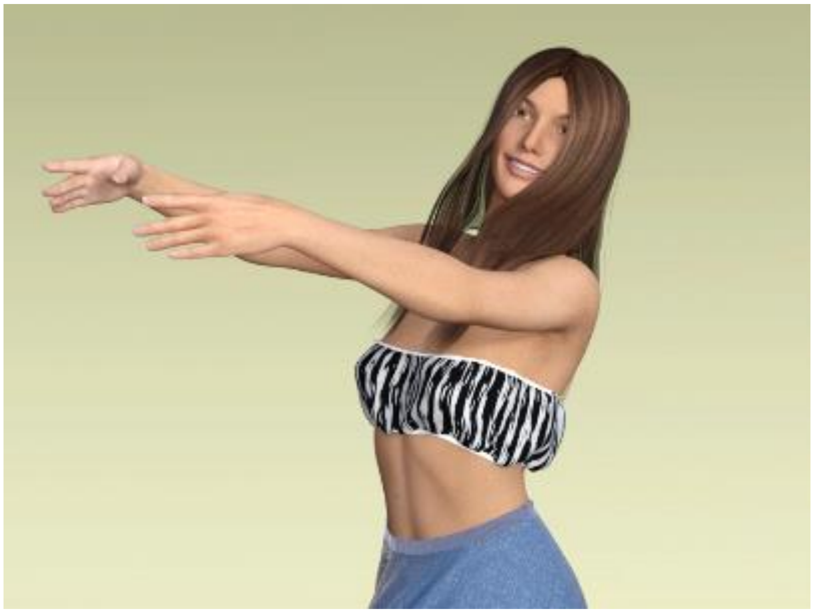
   e. Change the Contraction-Expansion Ratio of the top band to 65%. Run the simulation. You will get a tight fit of the top elastic band and a very loose fit of the cloth and the lower band.



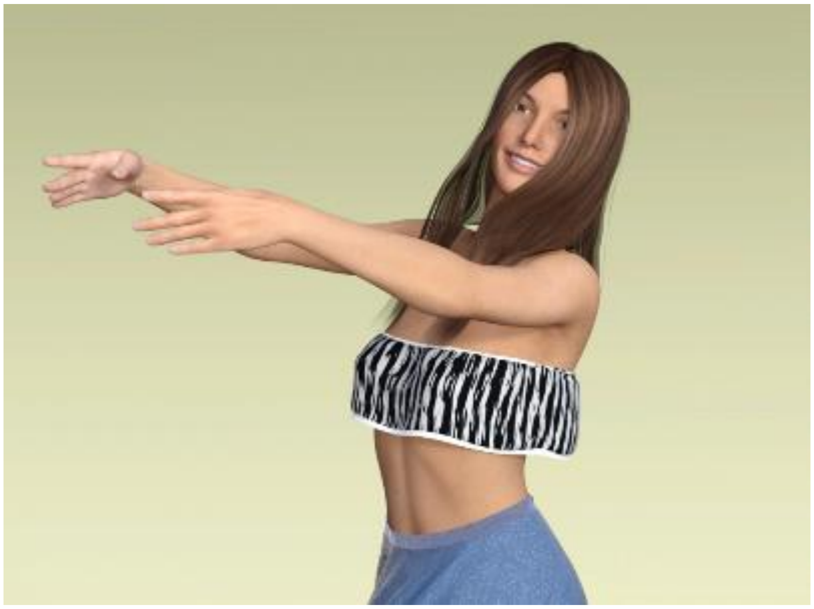
   f. Keeping the previous change, now change the Contraction-Expansion Ratio of the bottom band to 65%. Run the simulation. You will get a tight fit of the top and bottom elastic bands and a loose fit of the cloth. Note the droop in the back.



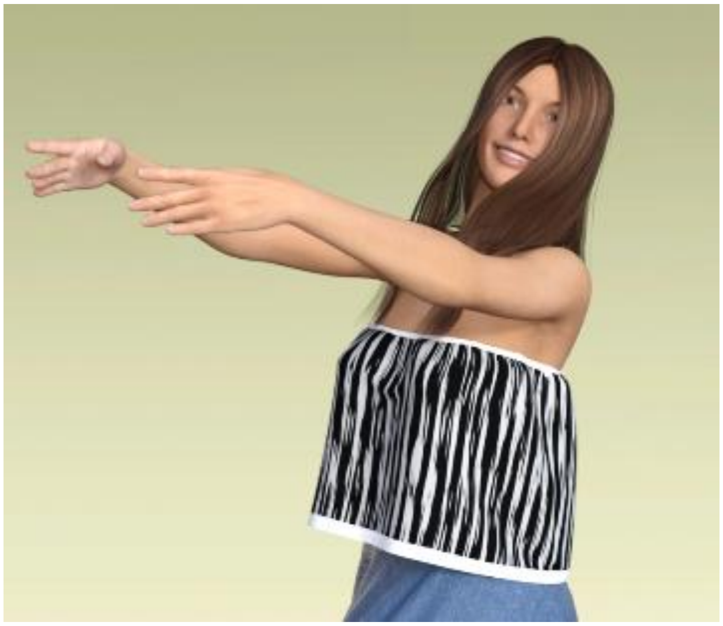
g. Keeping the previous changes, now change the Contraction-Expansion Ratio of the default surface band to 90%. Run the simulation. You will get a tighter fit of the cloth. If you go much lower on the C-E Ratio, the cloth will start to distort.



h. Set the Contraction-Expansion Ratio of the top band to 65%, the cloth to 90%, and the bottom band to 85%. Run the simulation. This will give you a tighter fit than "e" and a looser fit than "g".

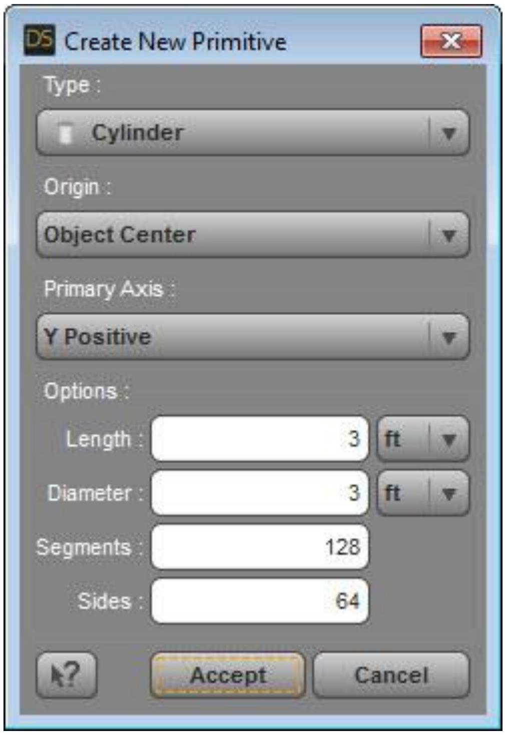


i. You can also make it into a longer top by scaling on the Y axis.



40. **Create a Dress with Gathered Waist.** This is a bit more complex than the bandeau. It has a band at the top to hug the chest above the breasts and a band around the waist too. I used a cylinder but you could use a cone the same way. It took some time to find settings that worked. With default surface properties, the top would slide down and the skirt would rise up and gather around the waist. Feel free to experiment with different property settings.

    a. Create a primitive cylinder with the parameters shown in the image. Position the cylinder around a figure with the top just under the armpits (start with a T-pose or with the arms even higher). I also had a slight (-2 degrees) X-rotation to better fit the figure.



Create New Primitive

Type : Cylinder

Origin : Object Center

Primary Axis : Y Positive

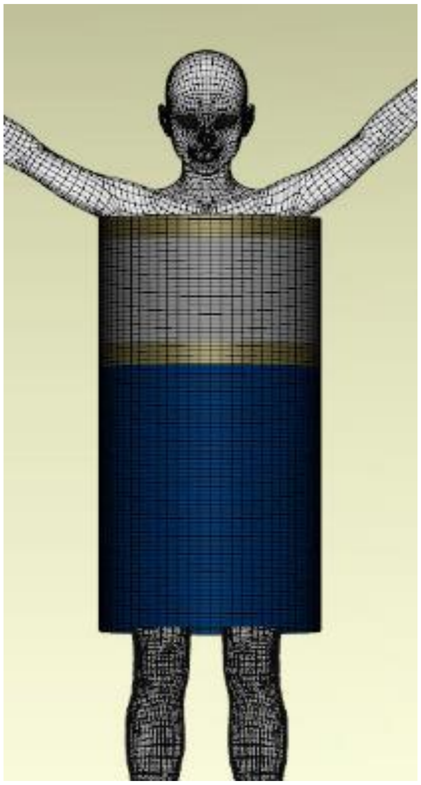
Options :

Length : 3 ft

Diameter : 3 ft

Segments : 128

Sides : 64



    b. Follow step 38 c to hide and delete the upper and lower cylinder faces.

    c. Follow step 38 e to create 3 additional surfaces: a Top Band using the upper 3-5 rows and a Waist Band of 3-5 rows about where the figure's waist is located. Select the polygons between the two bands and assign it to a new surface called Top. The skirt portion will retain the name Default. See image above.

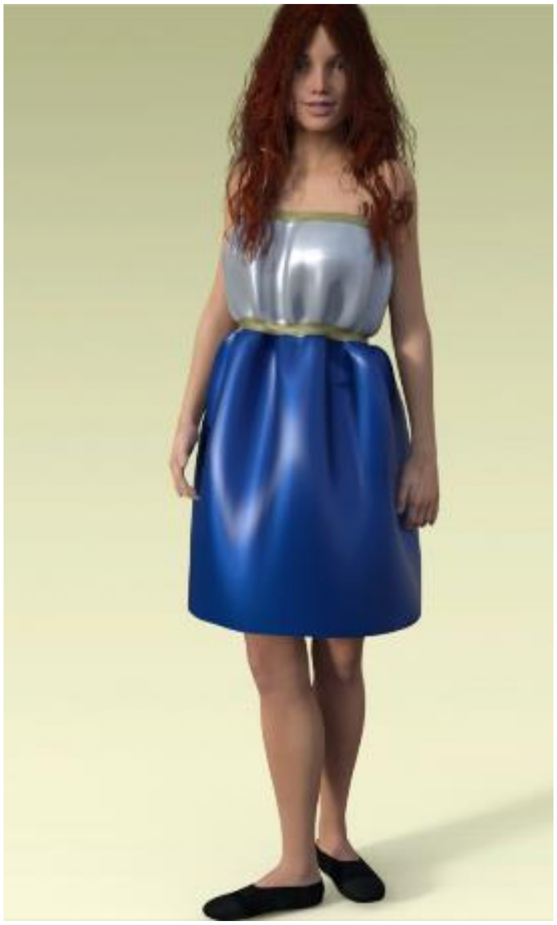
    d. Add a dForce dynamic modifier to the cylinder.

    e. We need the Top Band and Waist Band surfaces to cinch quickly to keep the cylinder from sliding down too far. For both of those, I set Contraction-Expansion Ratio to 50%. The Top will also be a bit baggy compared to the skirt, so I set its Contraction-Expansion Ratio to 90% while the Default surface remained at the default 100%.

    f. At default density, the bands tended to bunch up, so I set both the Top Band and Waist Band to 500 density. I set the Default (skirt) to 70 while the top remained at the default of 180. Density values below 70 and above 500 caused the mesh to blow up during simulation.

    g. Since they need to react quickly, I kept Dynamic Strength for the bands at the default of 1. The Top was set to 0.92 and the Default (skirt) was set to 0.75.

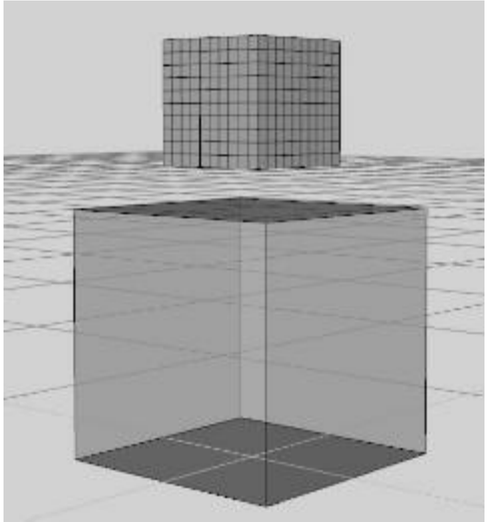
h. Run the simulation using an animated timeline. The image below shows what you get. I could not get the skirt portion to drape down as much as I would like. The tucked waist causes it to go out further from the body and hang as if very stiff fabric. Using a cone instead of a cylinder might help.



i. If you want a shorter dress, select some of the lower polygon rows and hide/delete them using the Geometry Editor. If you simply hide them with the Surfaces Opacity slider, they will still influence the behavior of the lower edge of the cylinder.

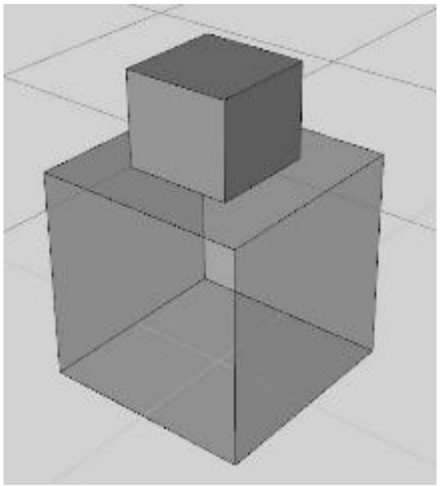
41. **How Resolution Affects Simulation Results**. If you plan to design your own props or clothes to use with dForce, it's important to understand how resolution can influence simulation results. In this section, I use primitive cubes of varying resolution to show what happens.

    a. Create a 2-foot cube with 1 division. Create a series of 1-foot cubes with 1, 2, 5, 10, 25, and 50 divisions. Center the smaller cubes slightly above the larger cube. Add a dForce dynamic modifier to all the 1-foot cubes.

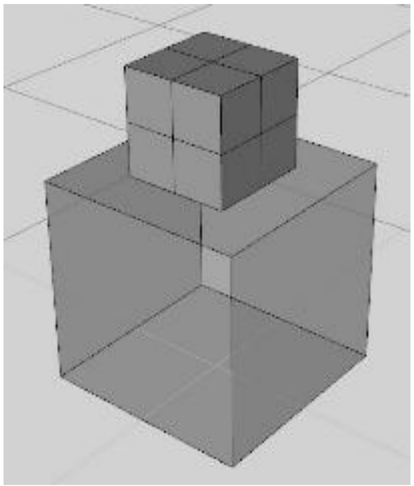


    b. Hide all but one of the 1-foot cubes and run the simulation. Repeat for all the rest. As you can see in the images below, the more divisions in the cubes the more distortion that occurs under dForce. With 1 division, there is only very slight compression. It takes 5 divisions before you see the sides warping. With 25, the cube is mostly collapsed and with 50 it's fully collapsed.

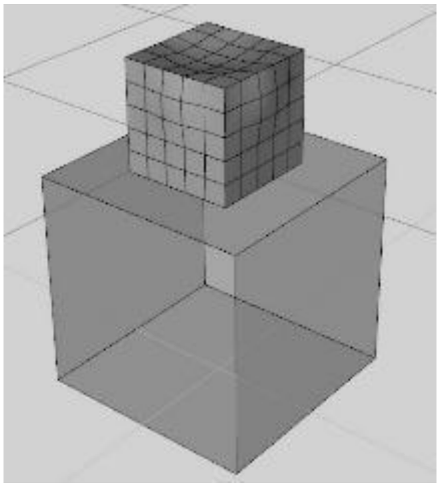
1 Division



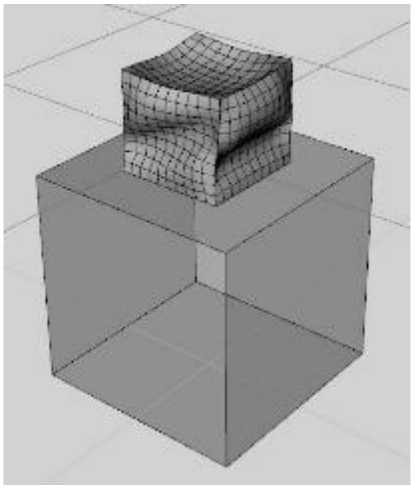
2 Divisions



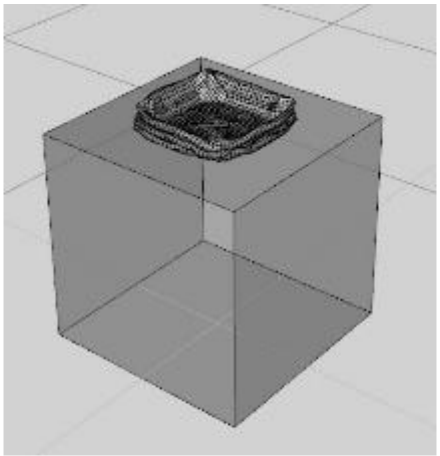
5 Divisions



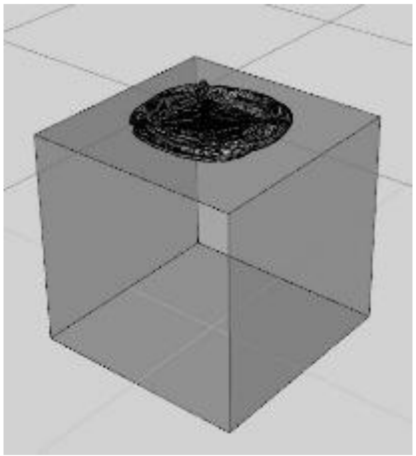
10 Divisions



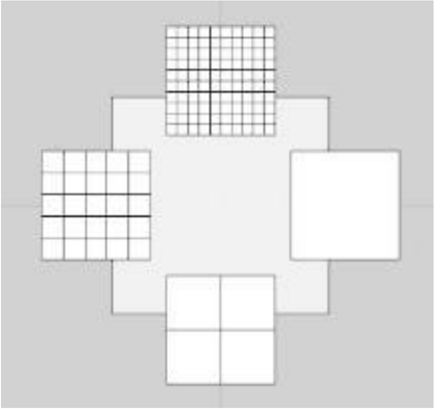
25 Divisions

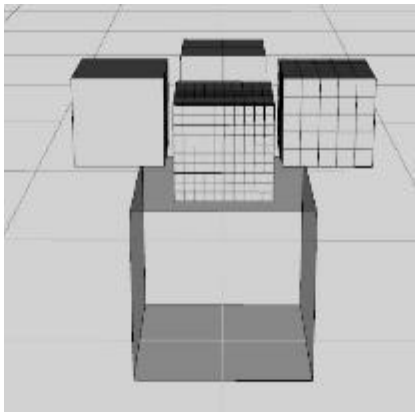


50 Divisions

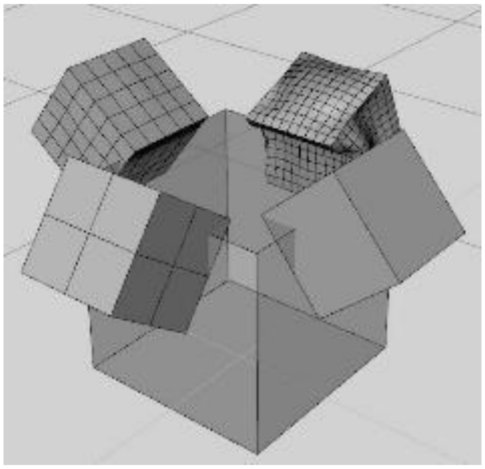


c. Next position a 1-foot cube over each side of the larger base cube. Run the simulation.

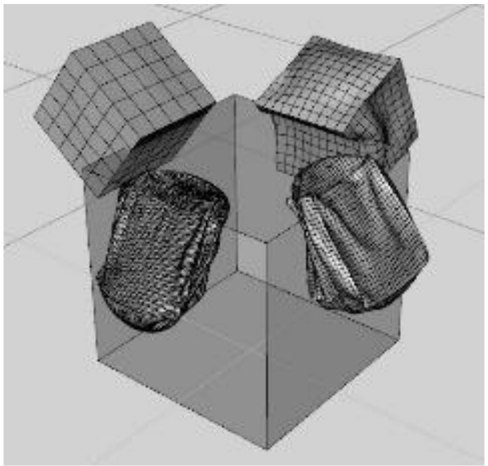


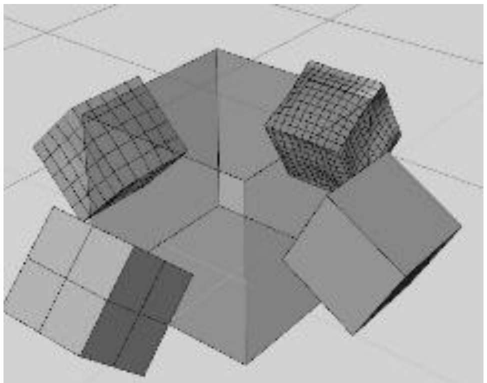


Divisions 1-2-5-10 at Frame 11:



Divisions 5-10-25-50 at Frame 11:

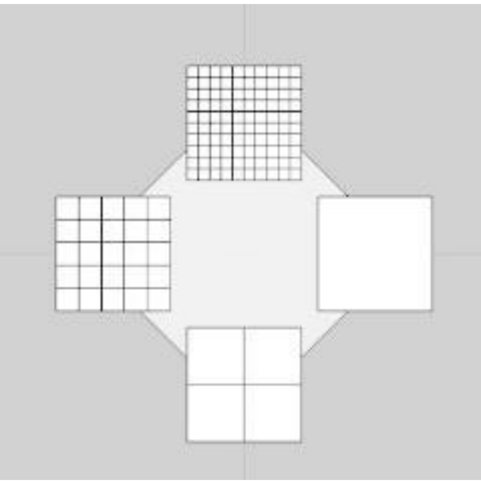




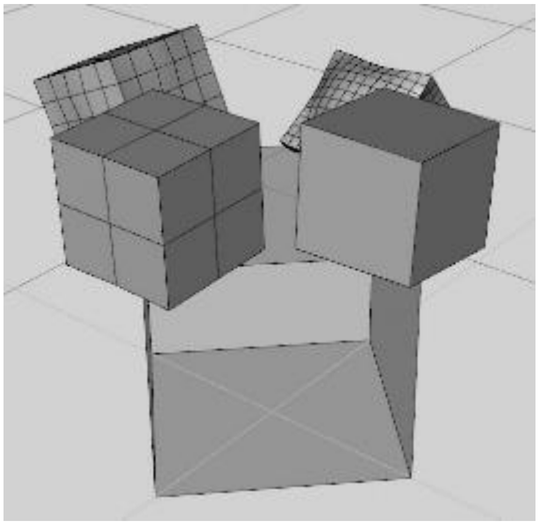
<- Divisions 1-2-5-10 at Frame 18

The lower resolution cubes distort little to none and fall completely off the base cube. Only the 25 and 50 division cubes remain attached as they distort and collapse.

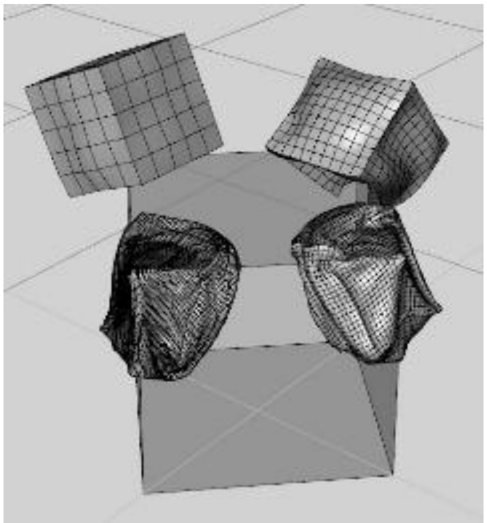
d. Now rotate the base cube so the smaller cubes are over a corner. Run the simulation. You get similar results, with divisions 1-2-5-10 falling off while 25 and 50 stick and collapse.



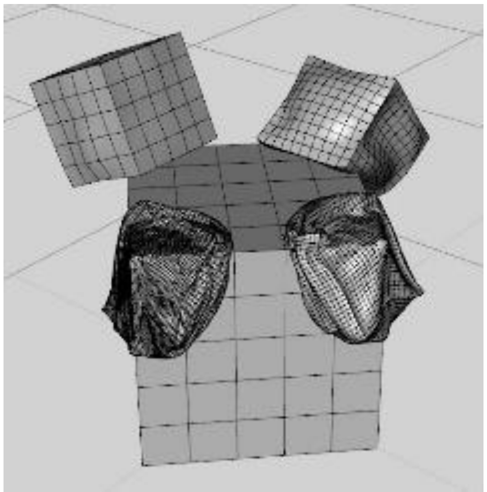
Divisions 1-2-5-10 at Frame 11:



Divisions 5-10-25-50 at Frame 11:

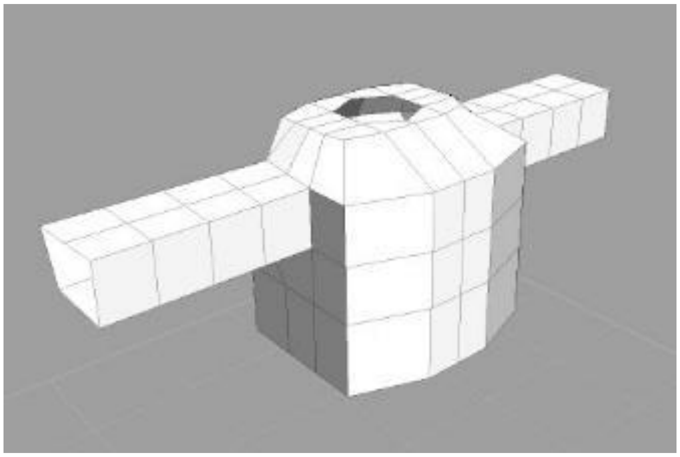


Replacing the 1 division base cube with a 5 division base cube had no apparent effect:

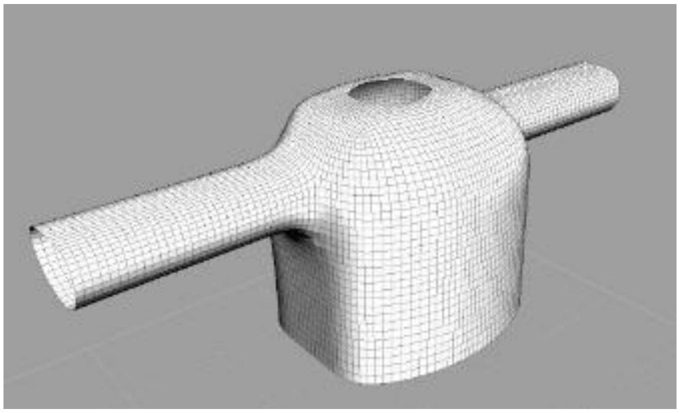


42. **Using Silo 2.5 to Create a Shirt**. With the success I had using primitive cylinders and cubes, I thought I'd give a try to making a shirt. Since the basic primitives in Daz Studio don't have arm extensions, I used Silo 2.5 to create a model. I haven't used Silo much so my attempt is crude (as you will see) but surprisingly after applying dForce it looks pretty good.

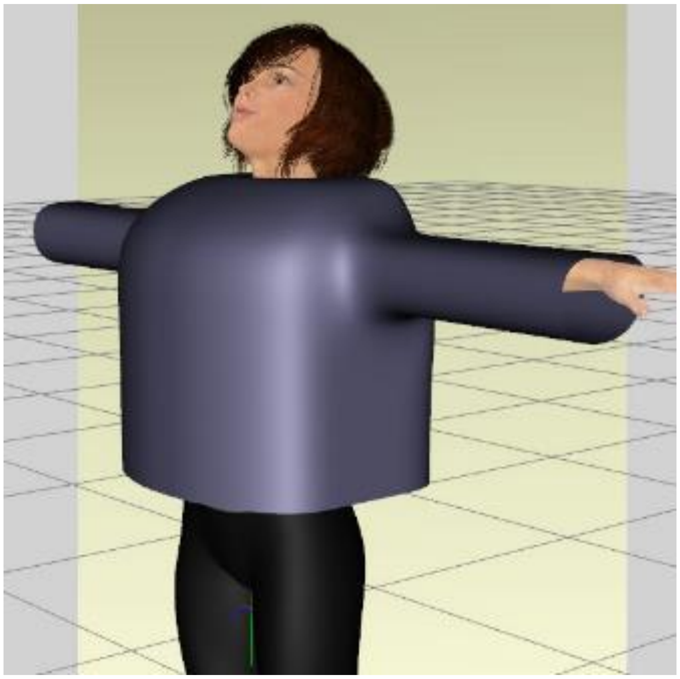
    a. This is the model made in Silo. I started with a 9-division per side cube and did some loop splitting, arm extrusion, vertex moving, and face deletion. Not much to look at.

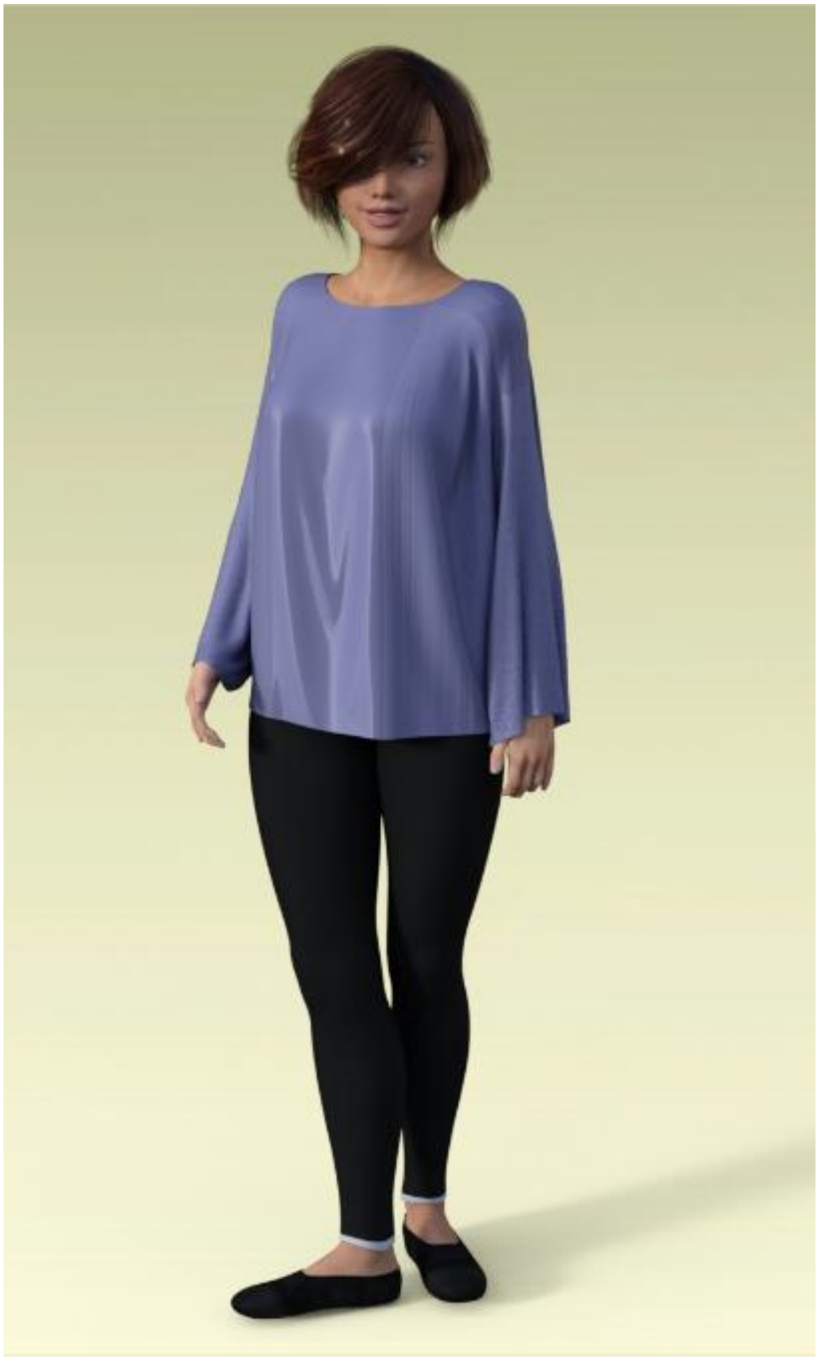
    b. One attractive feature of Silo is its subdivision abilities. This is the same crude model after multiple applications of subD. You can see how the square openings have become rounded without any work on my part. I made this more detailed mesh into the control mesh (so it will be preserved upon export) and created an .obj file.

    c. Here's the model after importing the .obj file into DS and placing it around a G3F figure. As you can see, it has poor fit and I had to tilt the head to avoid polygon clinging during the simulation. That wasn't a problem as I did an animated simulation and moved the head later.

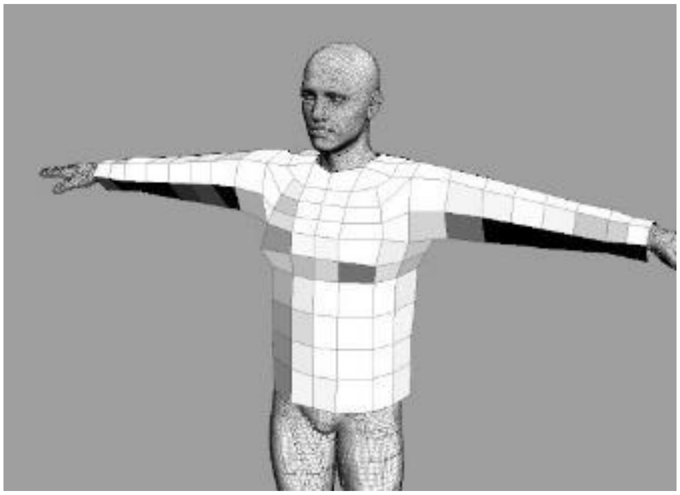
d. Here's after dForce simulation. Looks good I think!



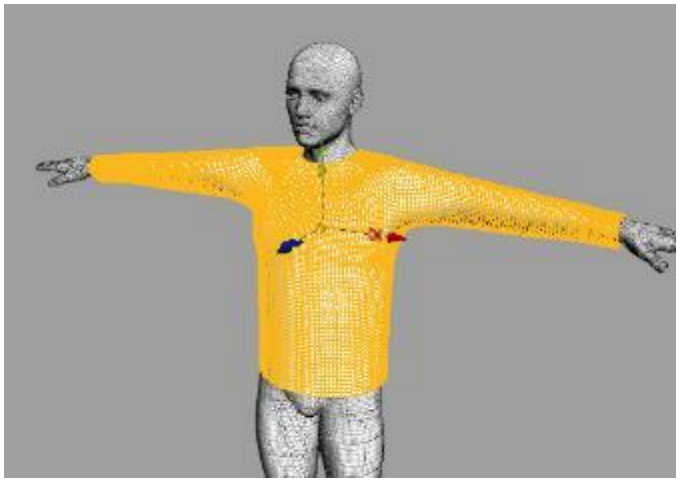
e. For those who want to experiment, the .obj file is available on my thread on Daz Forums. As I learn more about Silo, I'll post an improved version so others can use.

43. **Another Shirt Made with Silo**. Since my first attempt was a reasonable success, I tried a second. Instead of using Silo's built in female figure as a model, I exported a G3M figure as an .obj and imported that into Silo to build the shirt around. I started with the same basic model I did in section 42, but added more polygons and moved a lot of vertices around to better follow the shoulders and arms of the male. I also learned how to rearrange the UV so the texture would follow the shape. Below is the base model, the sub divided model, the shirt before simulation, and the resulting render.

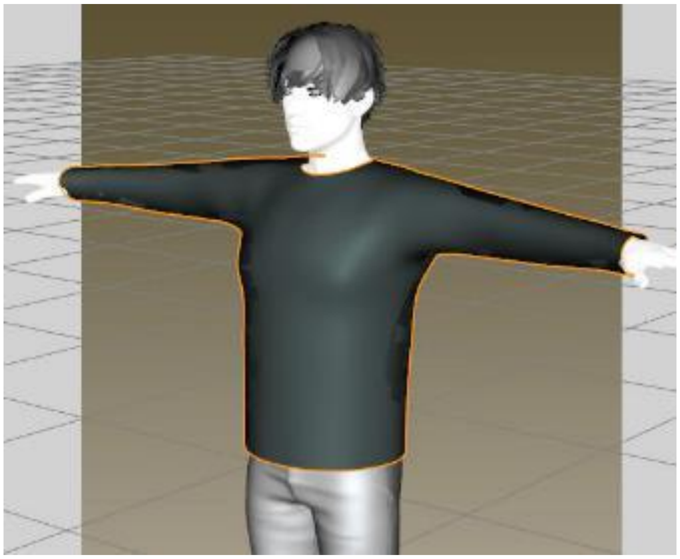
Base Model in Silo:



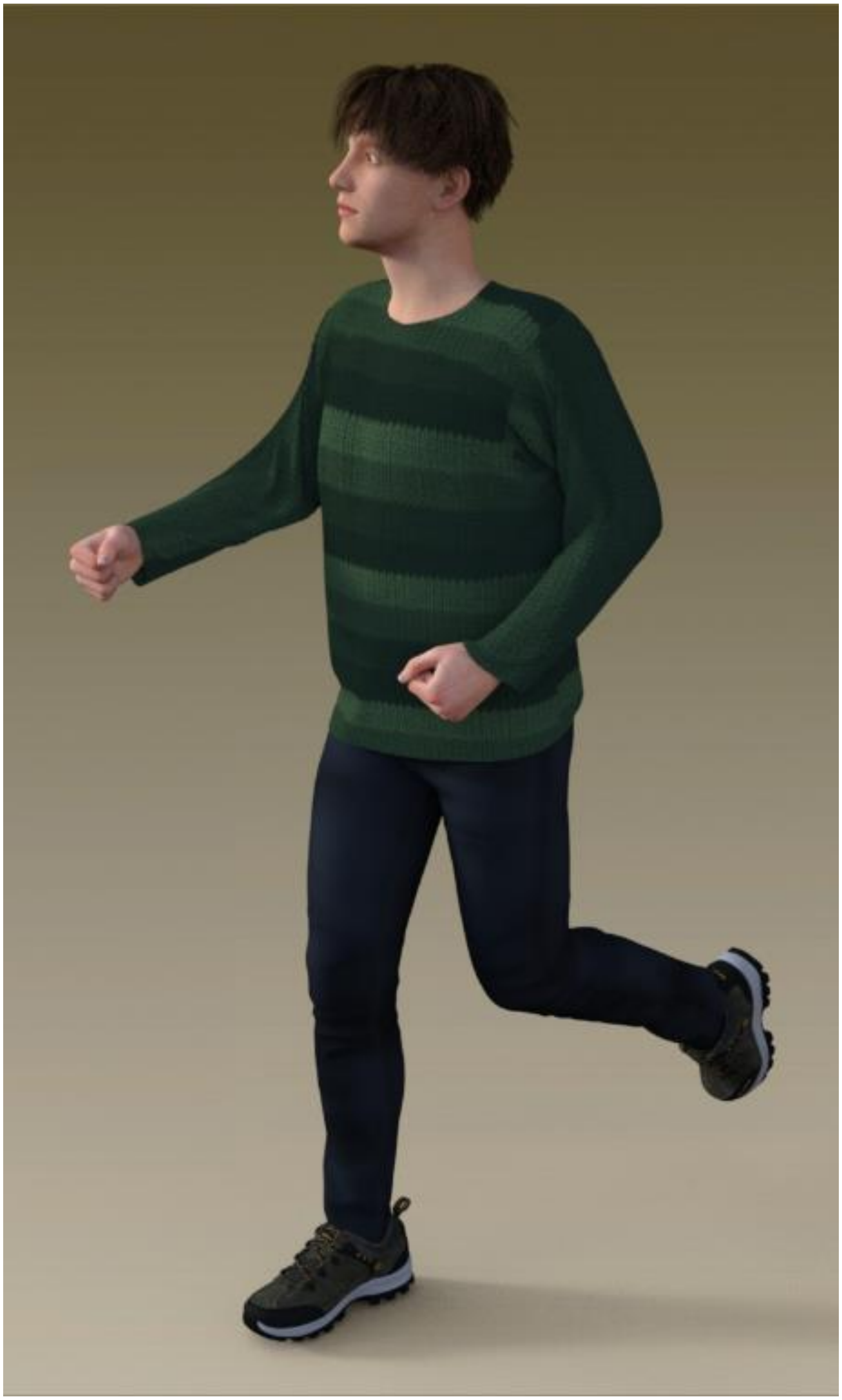
SubD Model in Silo:



Model Imported into Daz Studio:

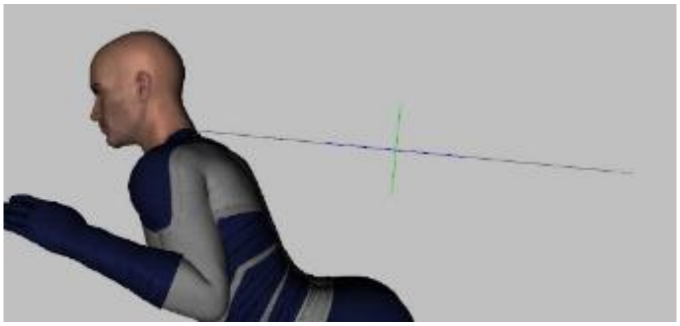


Final render after simulation:



44. **Flying Cape.** A question was asked in the Daz forums about keeping a cape spread out in the wind. For this example, I used a primitive plane stuck into the neck above the shoulders. Since there is no modeling around the neck and shoulders, it would only be useful for a distant pose as the cape has poor fit.

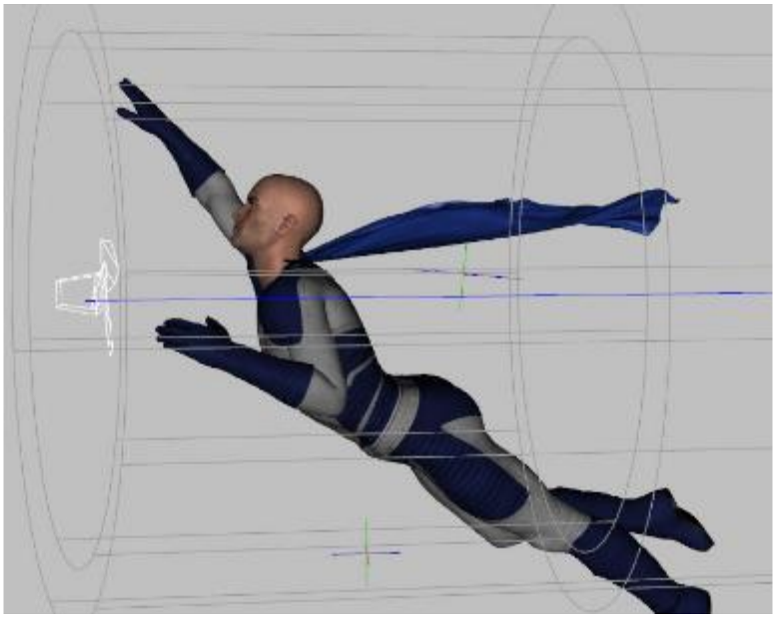
    a. Create a plane with 100 divisions and adjust the x and z scale to fit the figure. Position it just above the shoulders.



    b. Add a dynamic dForce modifier to the plane.

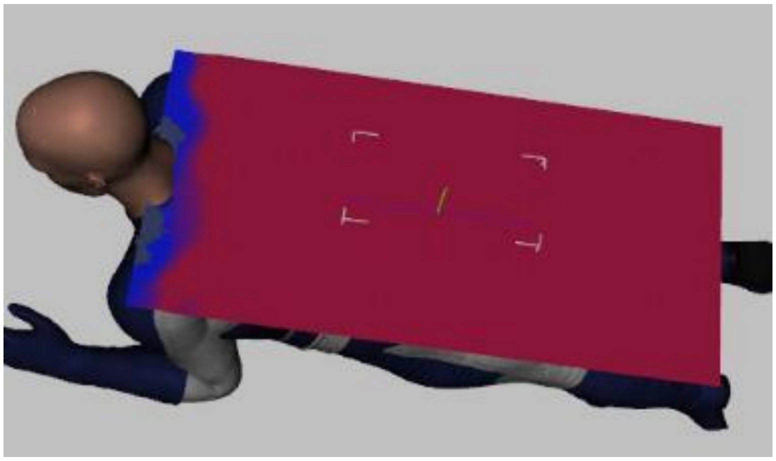
    c. Use default Simulation surface settings except set Stretch Stiffness to 0.1, Buckling Ratio to 5%, and Contraction-Expansion Ratio to 110%.

    d. Add a wind node directed parallel to the cape. Set the speed to 1 mph.



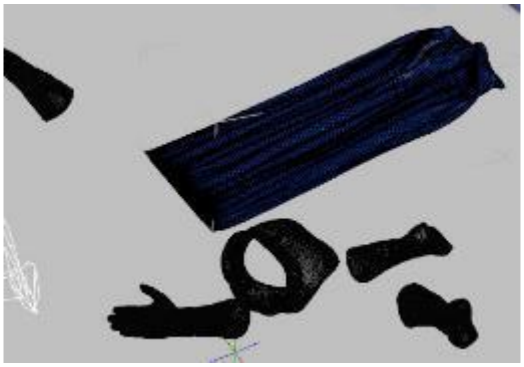
    e. In the Simulation Settings pane, set Gravity to -0.02 to give a slight upward movement. Set Air Resistance to 0. If you want more up/down motion, adjust the gravity setting accordingly.

    f. Add a dForce modifier weight node to the plane. While holding down the Alt key, paint around the neck and leading edge by the shoulders to remove influence.

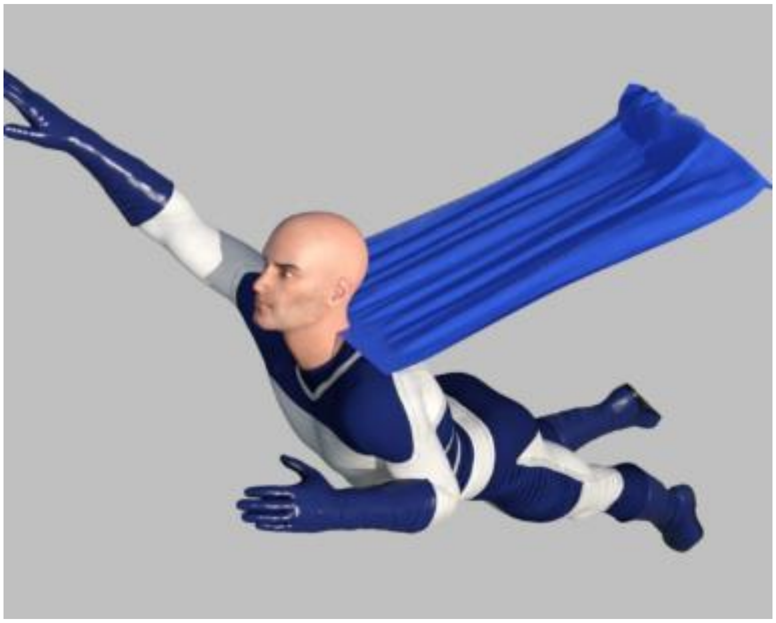


    g. I did an animated simulation with 40 frames. The figure’s arm was raised at frame 10.

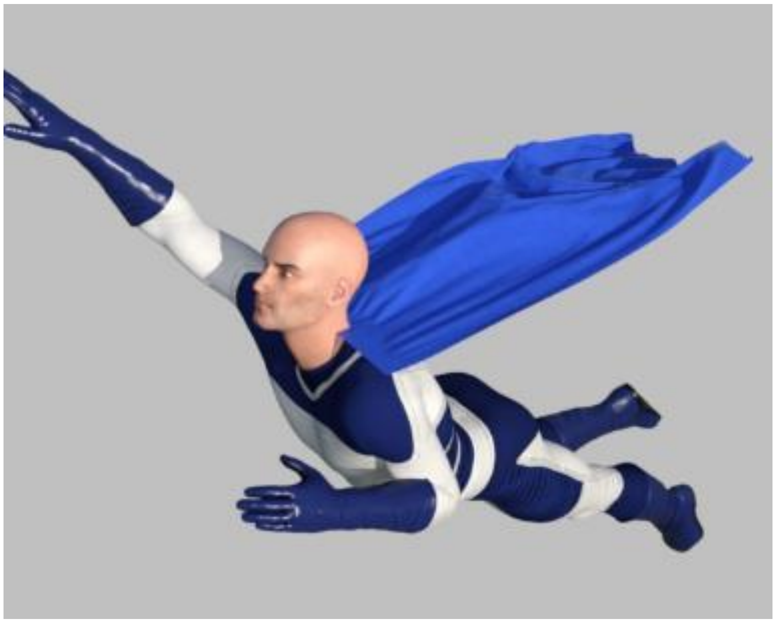
h. The result you get with the wind depends upon whether the figure and clothing are visible or not (image at right shows body hidden). When visible, they block the wind and you get a more turbulent result.



Body Hidden during simulation:



Body showing during simulation:



45. **Cylinder Helpers – Create a Shawl.** You can use primitives like spheres and cylinders to hold things in place or use them to move an object around. In this example, a plane is used to make a shawl and two cylinders are used to help in its placement.

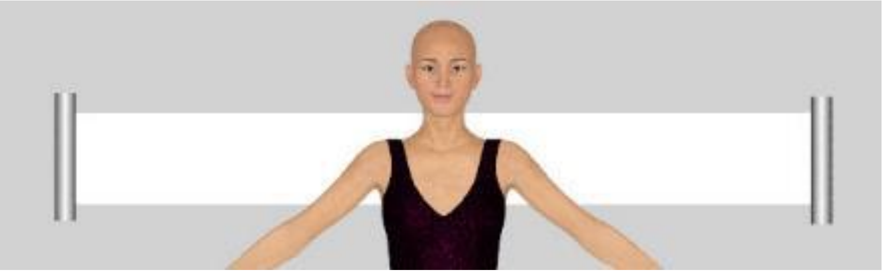
    a. Create a primitive plane with a Z-positive axis, 6 feet in size, with 100 divisions. Set Y-scale to 12%. Position it behind a figure (I used G3F) so the shoulders are in the middle. Add a dForce dynamic modifier.

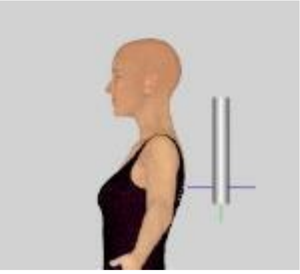
    b. Create a cylinder with a Y-positive axis, 1 foot in size, 2 inch diameter, 10 segments and 12 divisions. Position it at one end of the plane with the end of the plane embedded into the cylinder.

    c. Duplicate the cylinder and position it at the other end of the plane.

    d. Create a new group containing the plane and cylinders for easier animation.

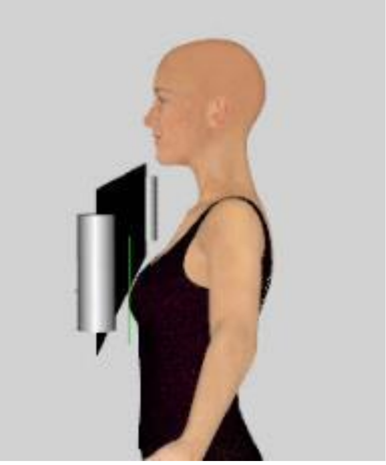
    e. Create an animated timeline. At frame 0, set a key frame by clicking on the key icon. The group should be positioned as shown below.



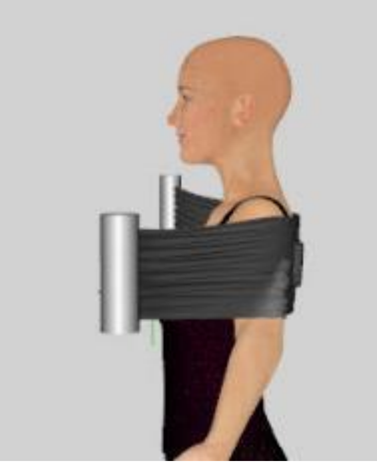


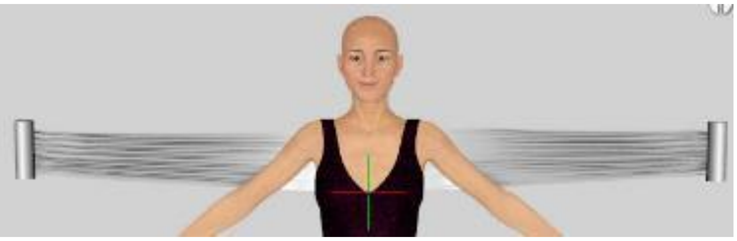
    f. At frame 4, move the group forward so it is now positioned in front of the figure. Also reduce the Y-scale of the cylinders to 50%. This will cause the shawl to wrap smoothly around the shoulders while also creating some folds in the unwrapped ends as they shrink.

Animation Setup:



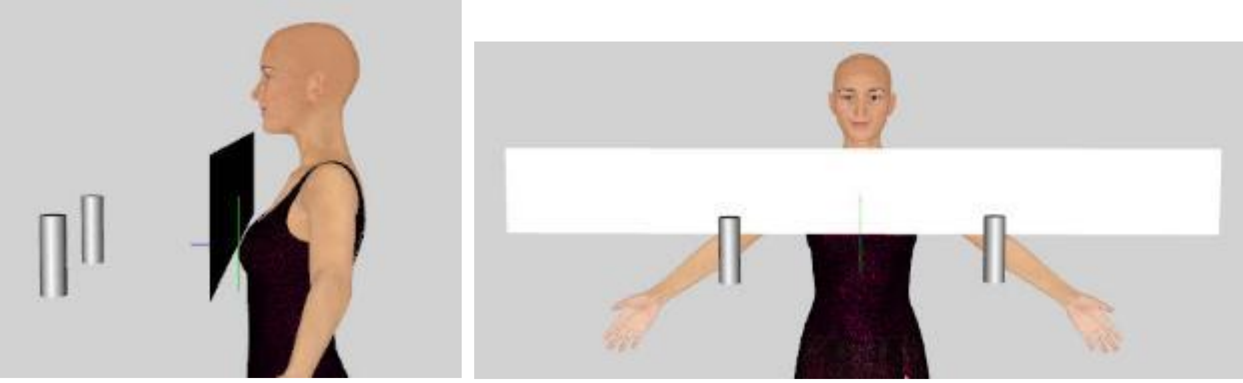
After dForce Simulation:



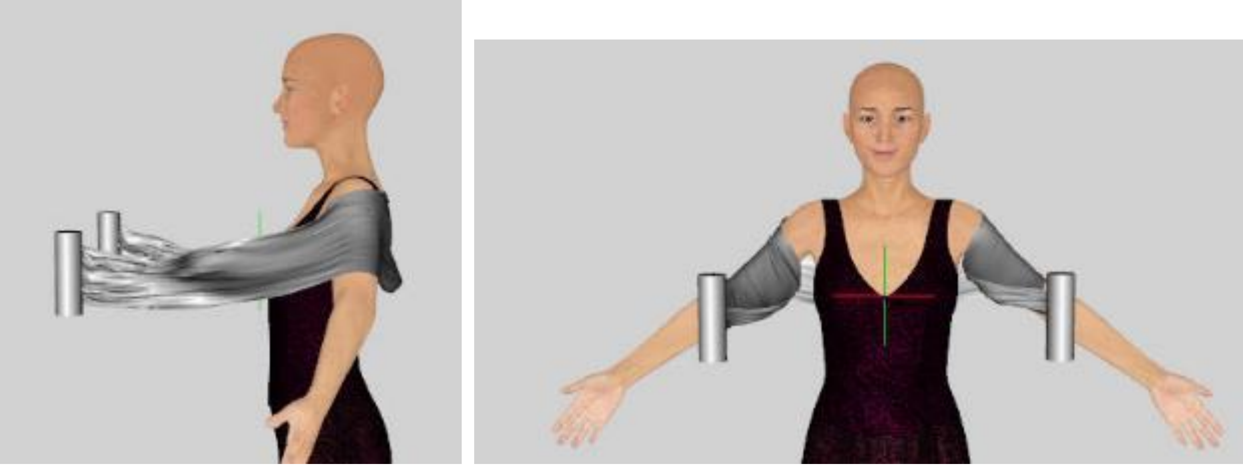


g. At frame 8, move the cylinders only (not the group) forward and slightly down (see side view). Also bring them in from the sides so they are under the arms where you want the shawl to wrap (see front view). Although the plane is not being animated, its ends are stuck inside the cylinder and will follow along.

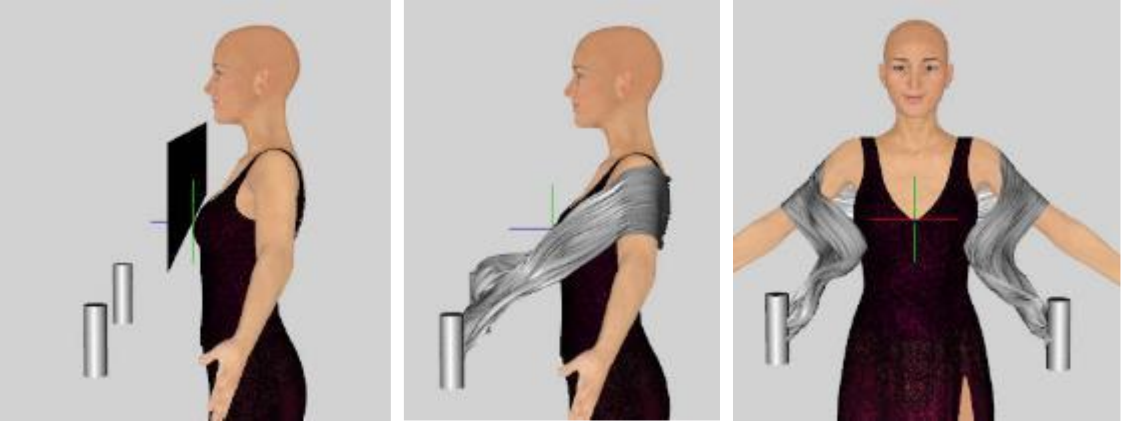
Animation setup:



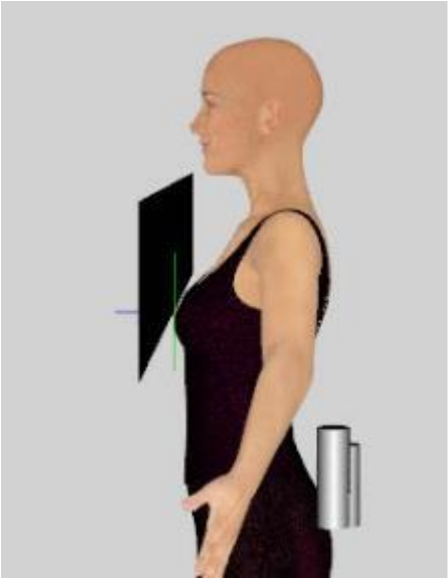
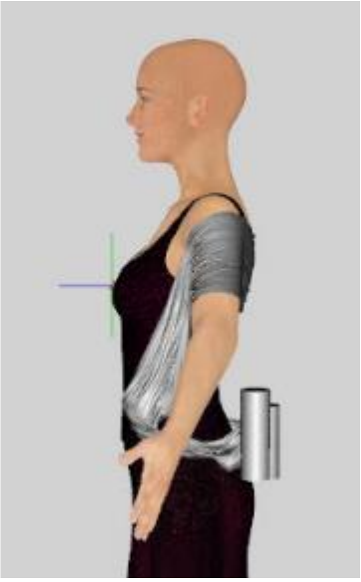
After simulation:



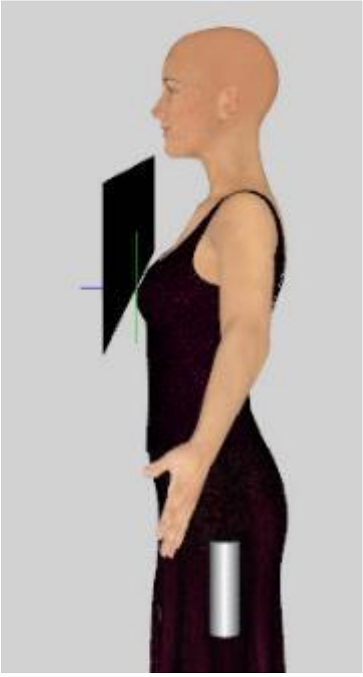
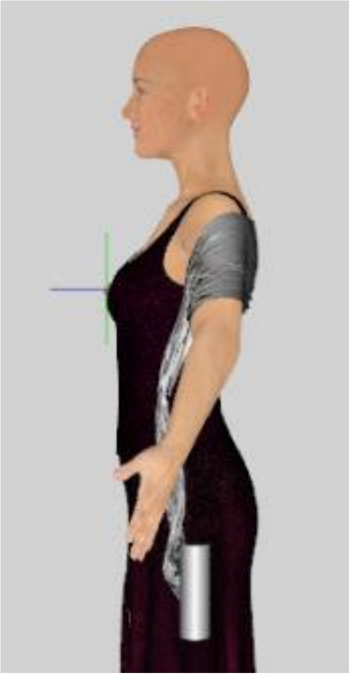
h. At frame 12, move the cylinders down and back towards the shoulders.



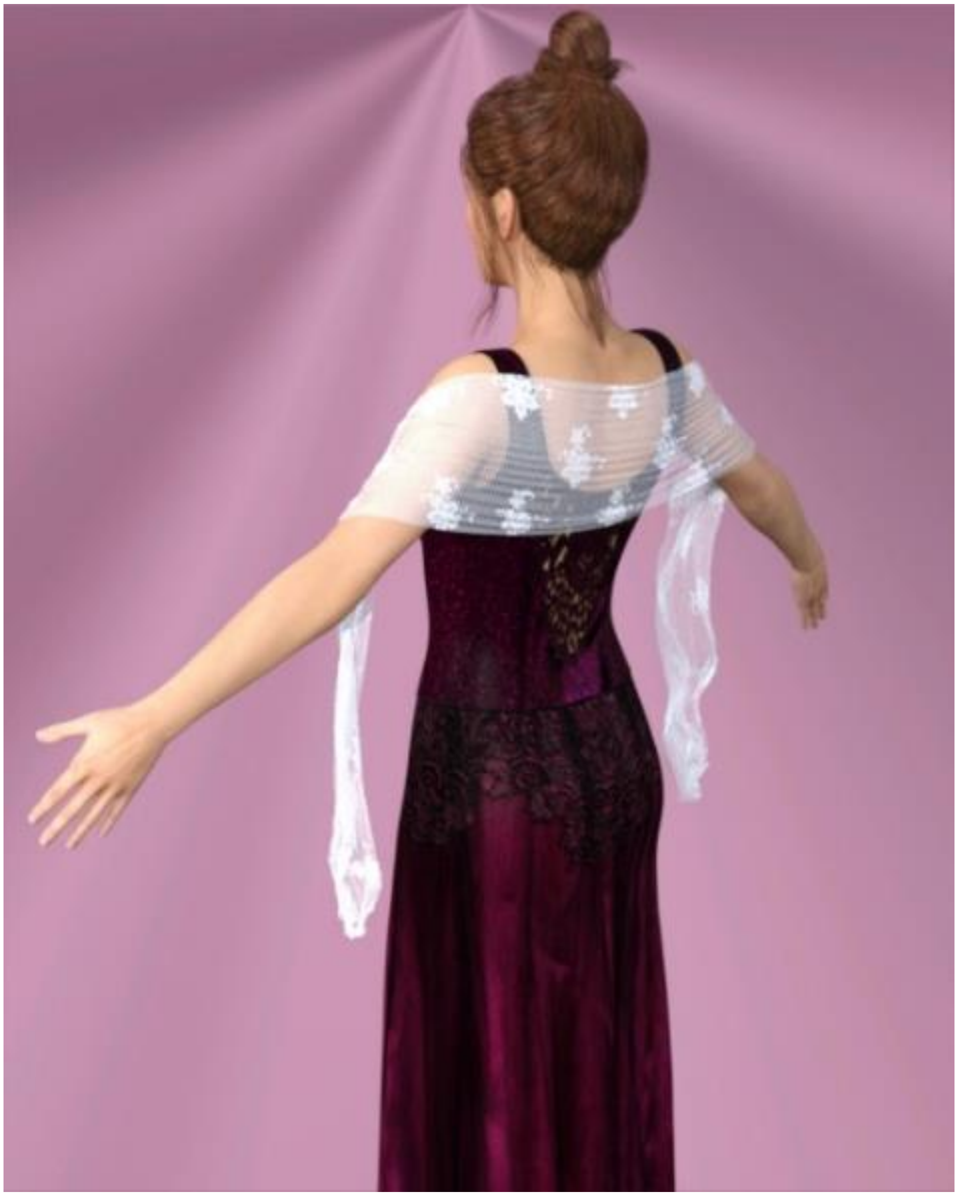
i. At frame 16, move the cylinders back to wrap the plane around the arms

j. At frame 20, move the cylinders forward so they are under the armpits

k. Hide the cylinders, add hair, and render.

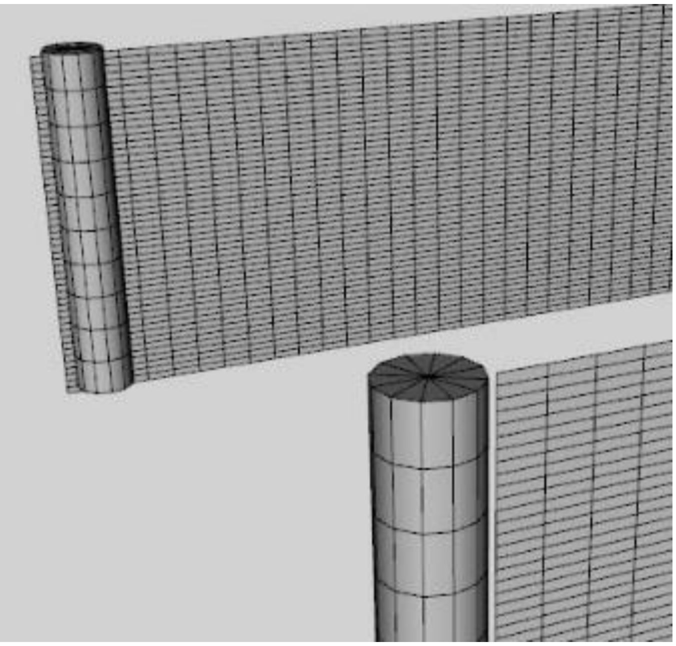


46. **Cylinder Helpers - Part 2.** This illustrates how embedding the end of a plane into a cylinder results in the plane following the animated primitive.

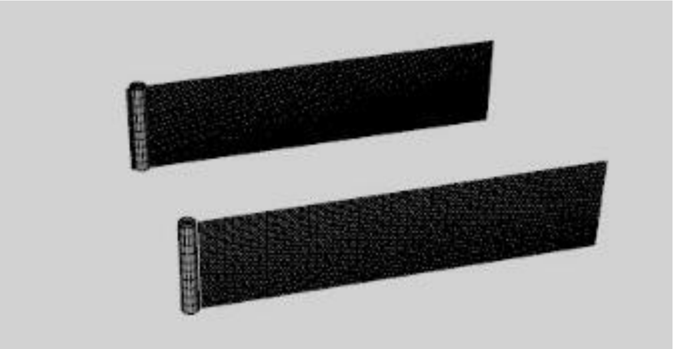
    a. Create two primitive planes with a Z-positive axis, 2 feet in size, with 100 divisions. Set X-scale to 200% and Y-scale to 50%. Add a dForce dynamic modifier to both.

    b. Create two cylinders with a Y-positive axis, 1 foot in size, 2 inch diameter, 10 segments and 12 divisions.

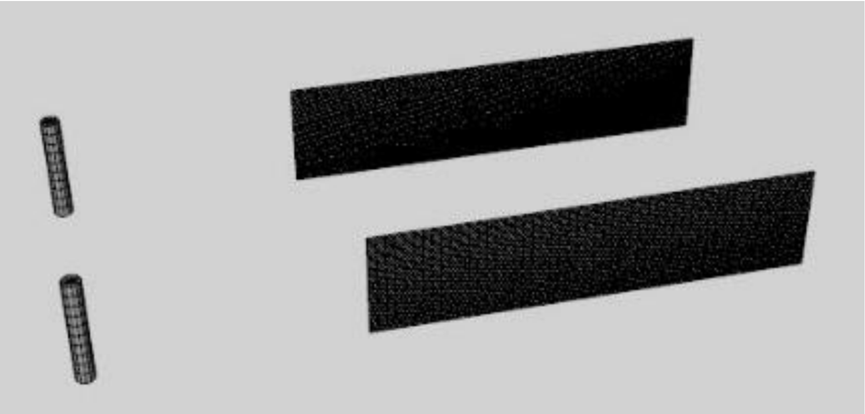
    c. Position one cylinder at the end of each plane. For one cylinder, be sure the plane extends out both sides of the cylinder (back plane in image). For the other cylinder, have the plane just be short of touching the cylinder (front plane in image). Do not parent the cylinders and planes.



    d. Create an animated timeline. At frame 0, set a key frame by clicking on the key icon. The objects should be positioned as shown to the right.

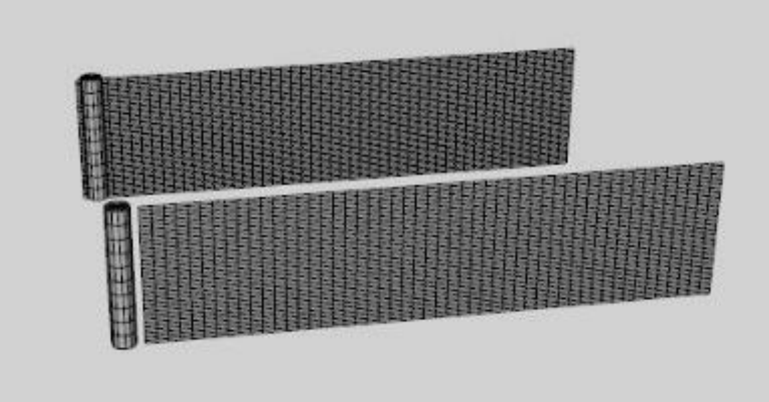


    e. At frame 20, move the cylinders to the right. Do not move the planes.

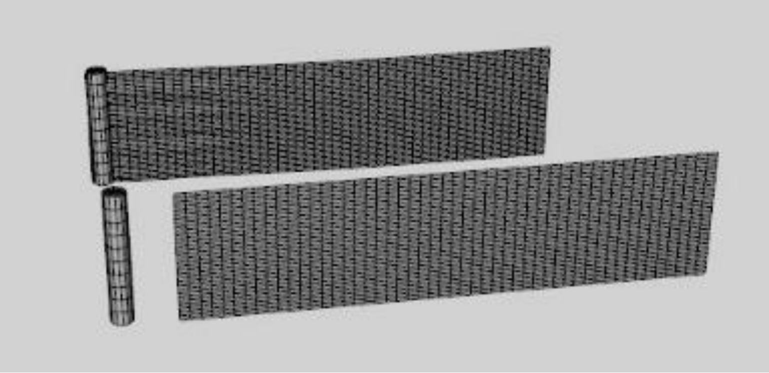


f. Go to the Simulation Settings pane and select the Environment group. Change gravity to 0.1 (you can leave it at 1.0 but the unattached plane will drop rapidly. By setting it to 0.1, it will stay within view as the simulation proceeds). As you can see below, the non-embedded plane is left behind while the embedded plane moves and experiences distortion from the movement.

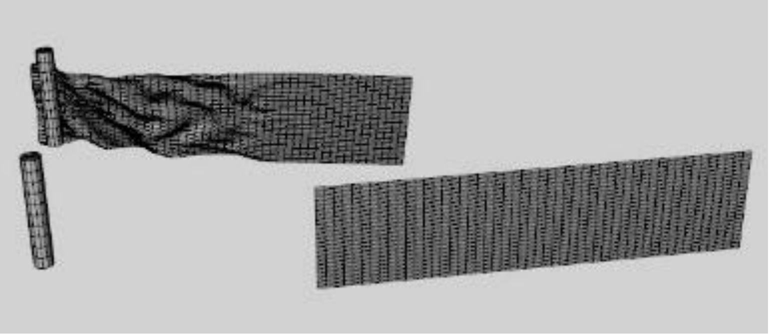
Simulation Frame 0:



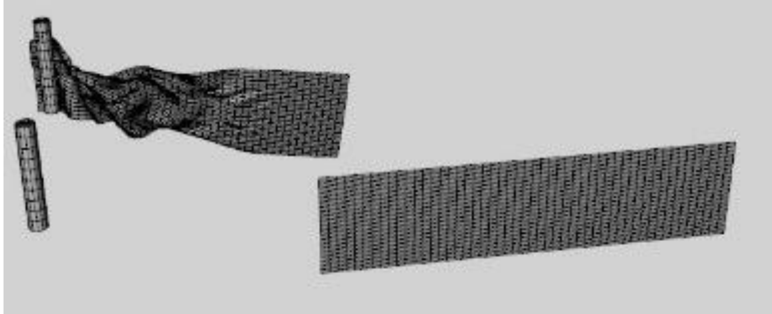
Simulation Frame 4:



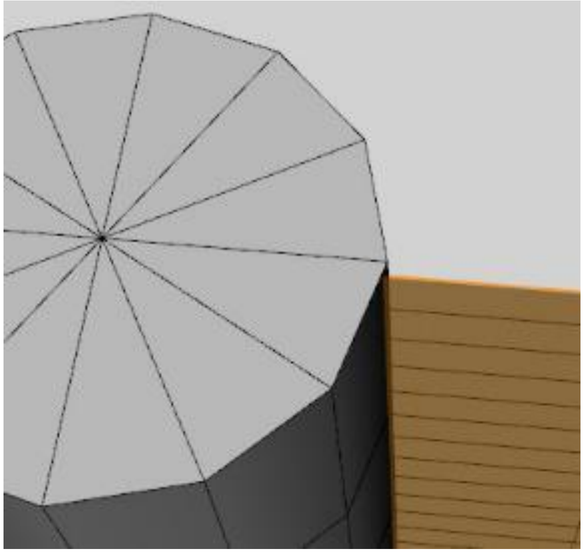
Simulation Frame 16:

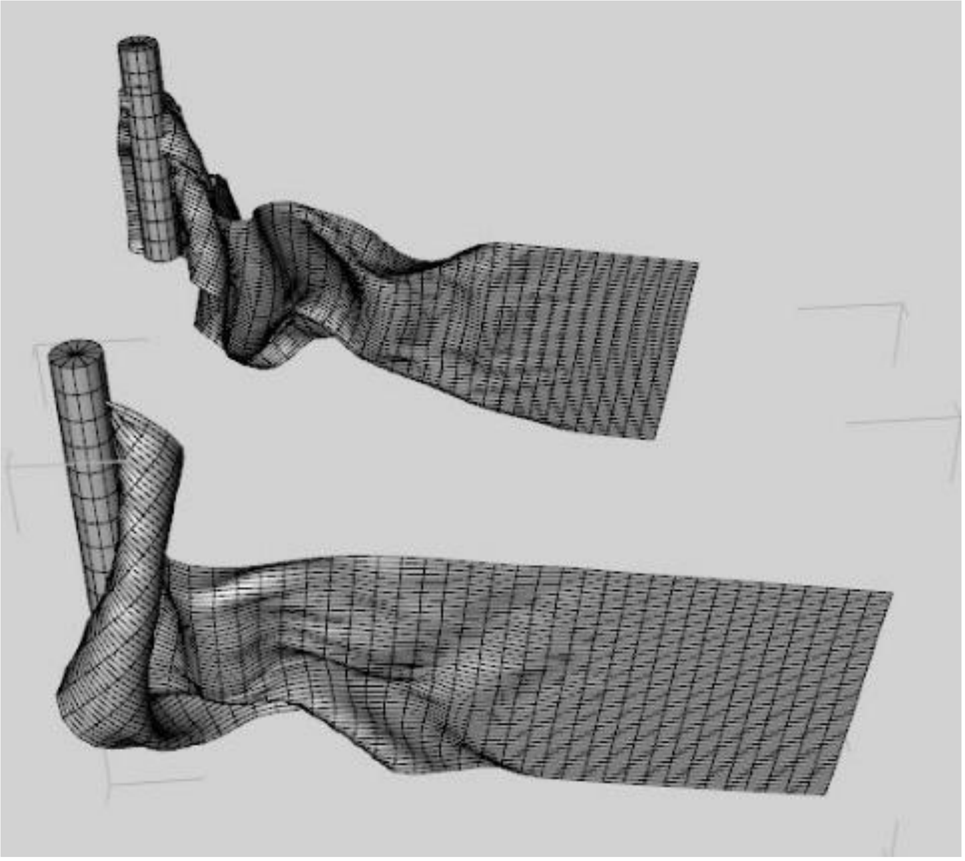


Simulation Frame 20:



g. The plane does not have to go all the way through the cylinder. As long as a bit of it is inside (see orange area in image at right), it will follow along.

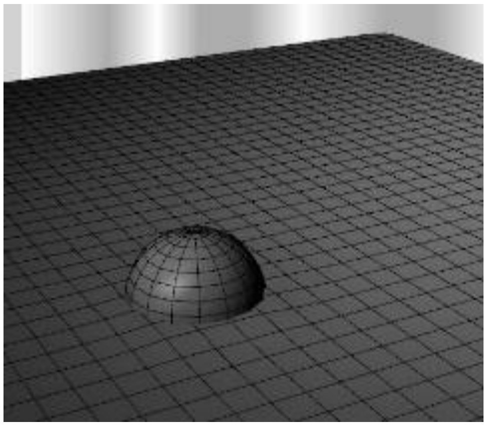




47. **Sphere Helper – Pull Cloth through Ring.** In addition to cylinders, you can use a sphere to move an object around. In this case, the sphere pulls a cloth through a ring and leaves it hanging, like a washcloth in a round towel holder.

    a. Create a 12 inch plane with 50 divisions. Add a dynamic dForce modifier. Add a smoothing modifier too if you'd like.

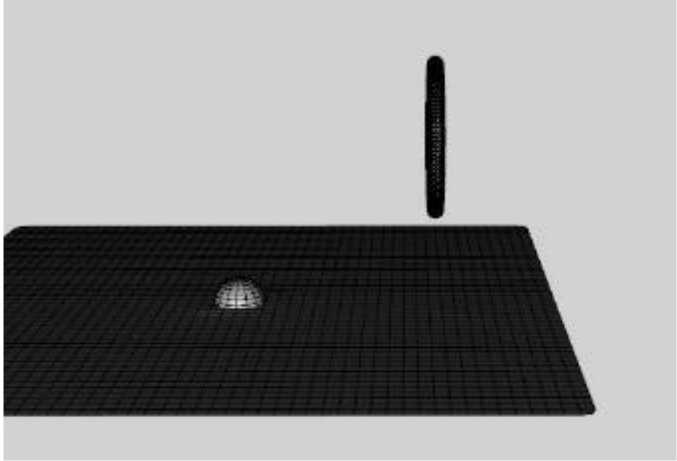
    b. Create a 1 inch sphere with 16 sides and 16 divisions and embed it in the top of the plane.



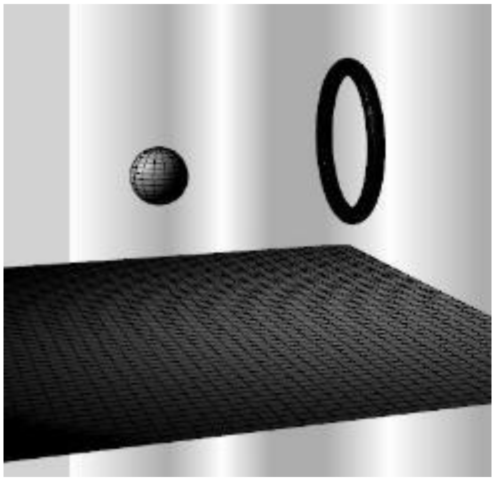
    c. Create a torus with X-positive axis, a 4 inch major diameter, a 0.25 inch minor diameter, with 250 segments and 32 sides. I used such a high number of segments as it helps keep the cloth from sliding down into the torus.

    d. Angle the torus 90 degrees. Add a static dForce modifier and set Friction to 0.001. This will help the cloth slide through the torus all though it will still catch the cloth and hold on at some point during the simulation.

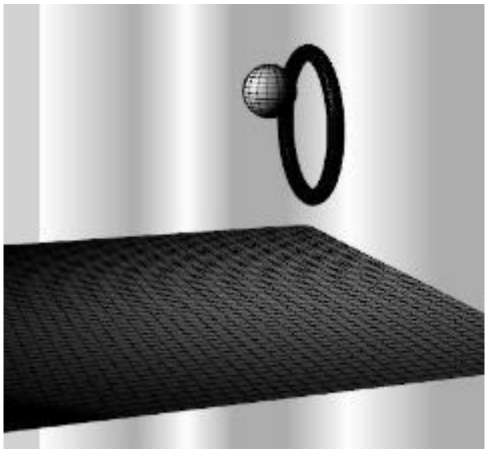
    e. Create an animated timeline. At frame 0, set a key frame by clicking on the key icon. The objects should be positioned as shown below.



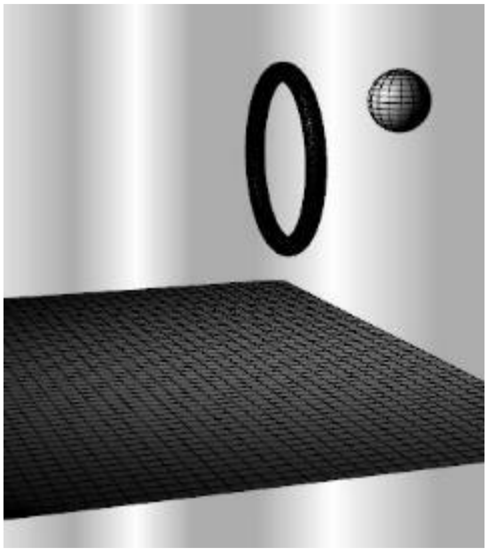
    f. At frame 4, raise the sphere to the same height as the bottom of the torus.



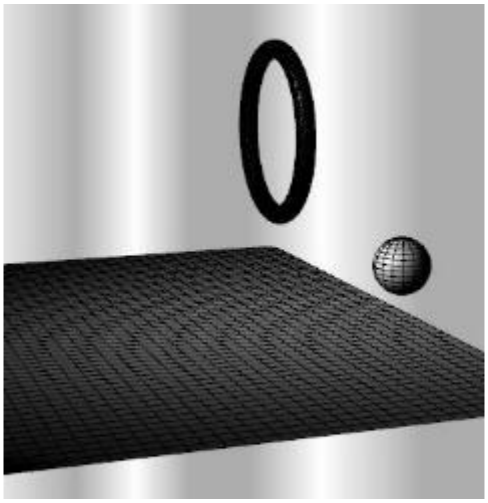
g. At frame 10, move the sphere close to the torus and raise it to the same height as the top of the torus.



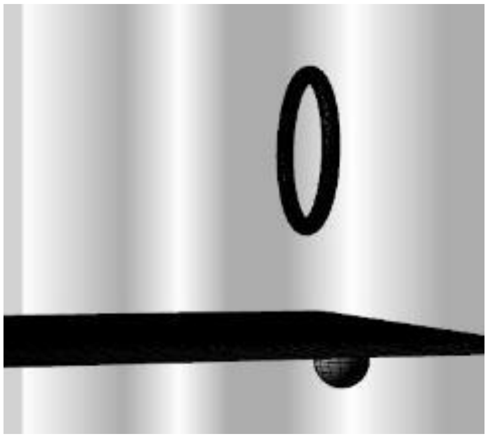
h. At fame 16, move the sphere to the other side of the torus.



i. At frame 20, swing the sphere down.

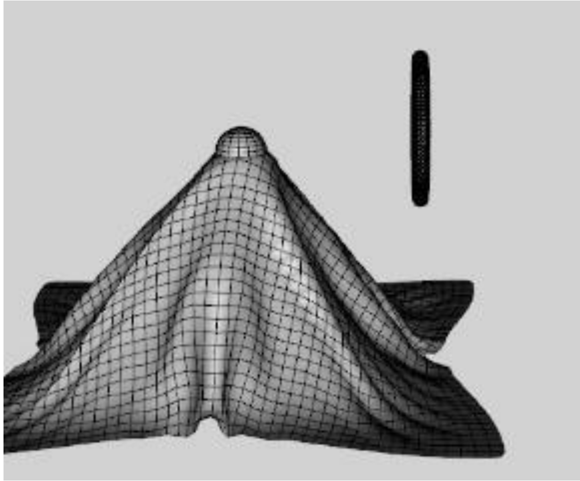


j. At frame 24, swing the frame down so it is under the torus and the plane.

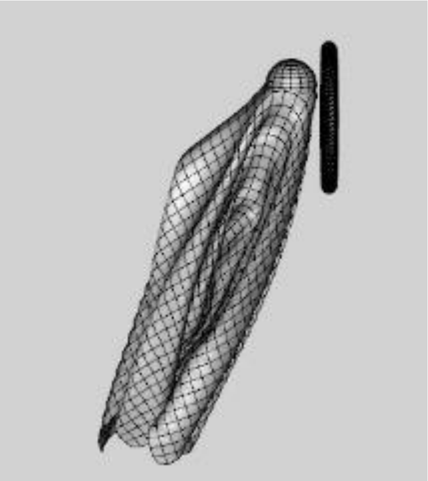


k. Run the dForce simulation. You should see the plane move along with the sphere. If not, check to be sure the sphere is embedded into the center of the plane and that the Simulation Settings for Collision is set to Better or Best (if set to Good, the plane will not follow).

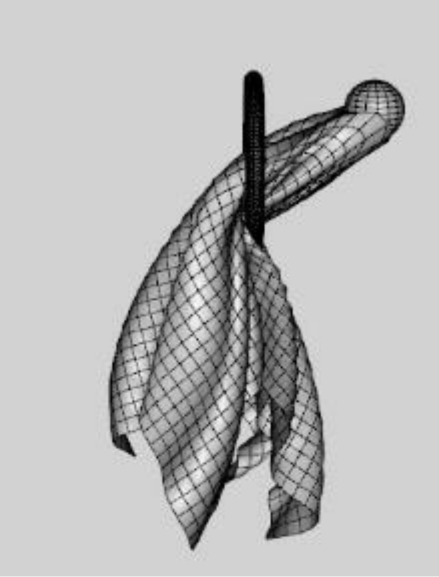
Frame 4:



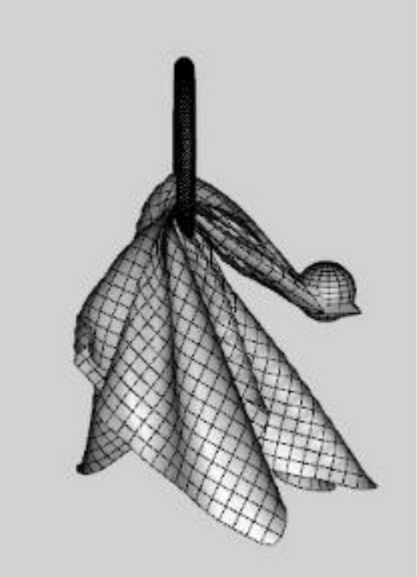
Frame 10:



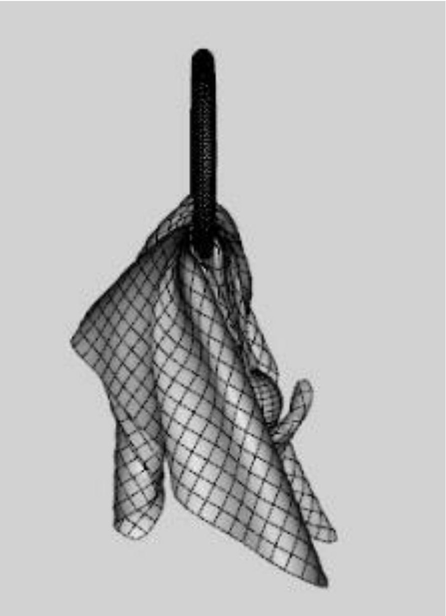
Frame 16:



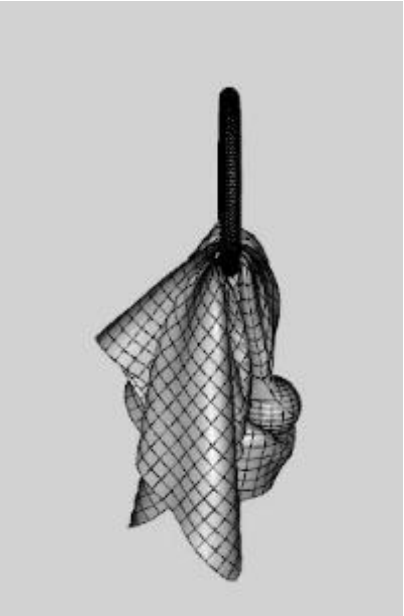
Frame 20:



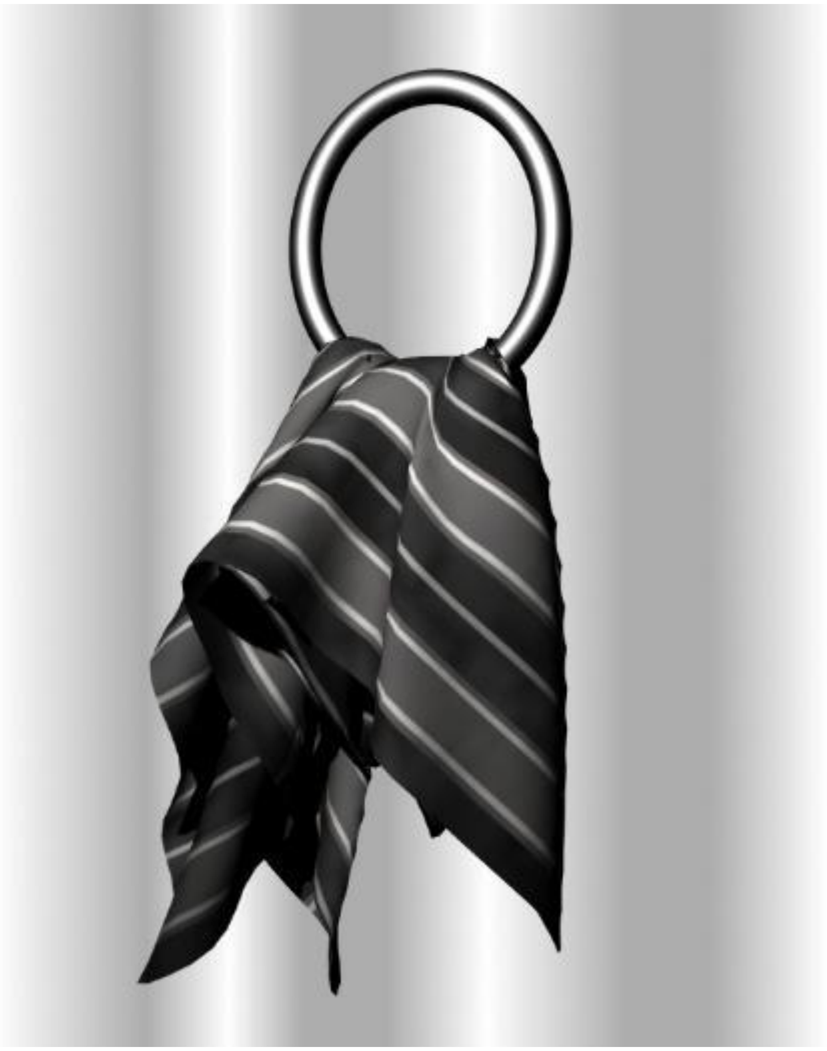
Frame 24:



Frame 30:



Finished render:



48. **Cylinder Rigidity.** You can use a long cylinder as a stand-in for a rope, chain, or other object; however, I have found they tend to collapse in on themselves. This was a study of how resolution and buckling stiffness and ratio can overcome this.

   a. Create a long thin cylinder 3 feet long and 3 inches in diameter. Set axis to X-positive and use 10 segments and 8 sides.

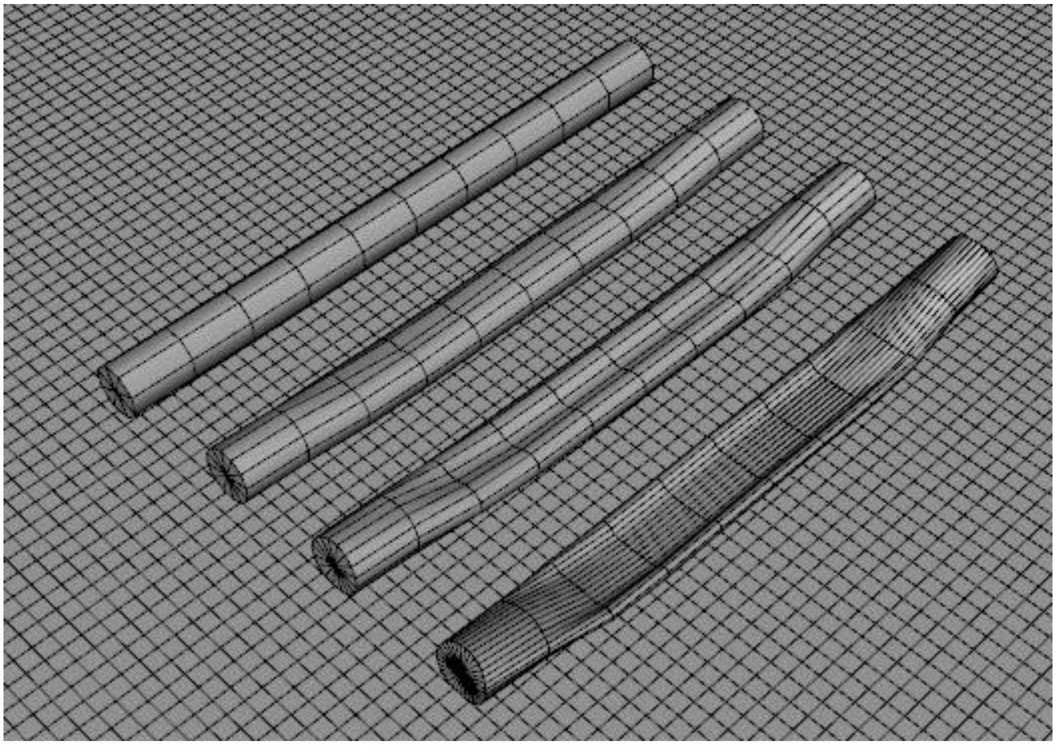
   b. Create 3 more cylinders, this time using 12, 16 and 32 sides.

   c. Apply a dynamic dForce modifier to each cylinder.

   d. Create a plane to act as a collision object.

   e. Line up the cylinders at an angle so one end hits the ground first.

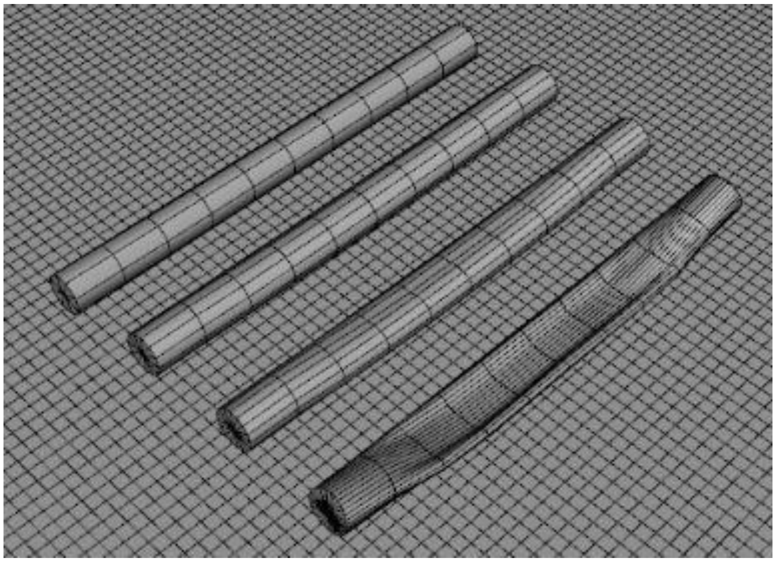
   f. Run a simulation with the default surface properties. You'll find the eight-sided cylinder retains its shape, the 12-sided shows slight flattening in the middle, the 16-sided shows a lot of collapse, and the 32-sided collapses completely.



   g. Use the following settings for the Simulation surface properties. These were determined by trial and error to minimize collapse and not have the object fly away.

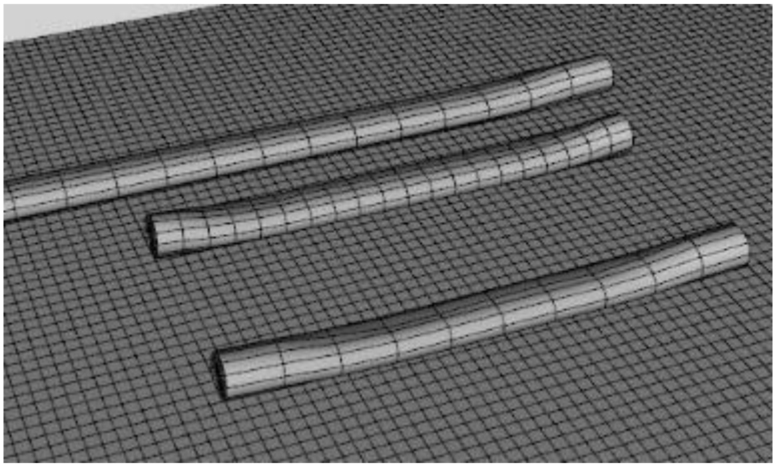
Sides	Buckling Stiffness	Buckling Ratio
8	5% (default)	70% (default)
12	35%	70%
16	62%	0.25%
32	10%	0.1%

With these new surface settings, the 12-sided retains its shape and the 16-sided has only slight flattening. Nothing helped the 32-sided cylinder.



h. Create two more 12-sided cylinders but with 20 segments instead of 10. For one of these cylinders, set Y-scale to 200%. Position them parallel to the 12-sided cylinder created in step b. Use default simulation surface properties for all three.

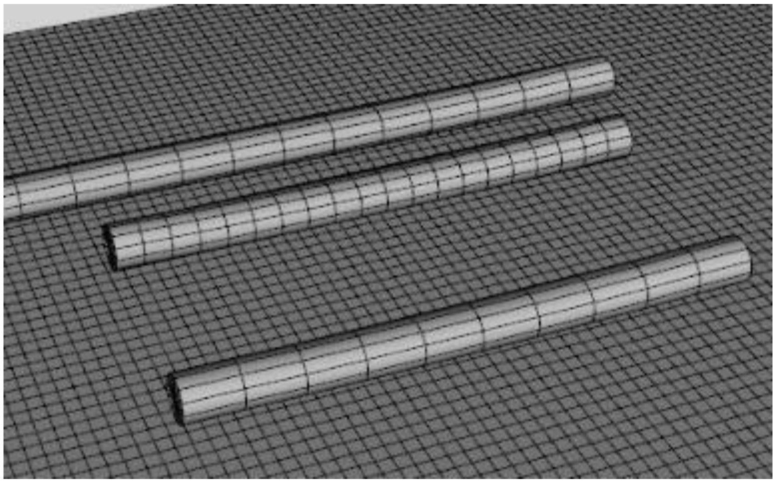
i. Run the simulation. You'll find some flattening for all three cylinders.



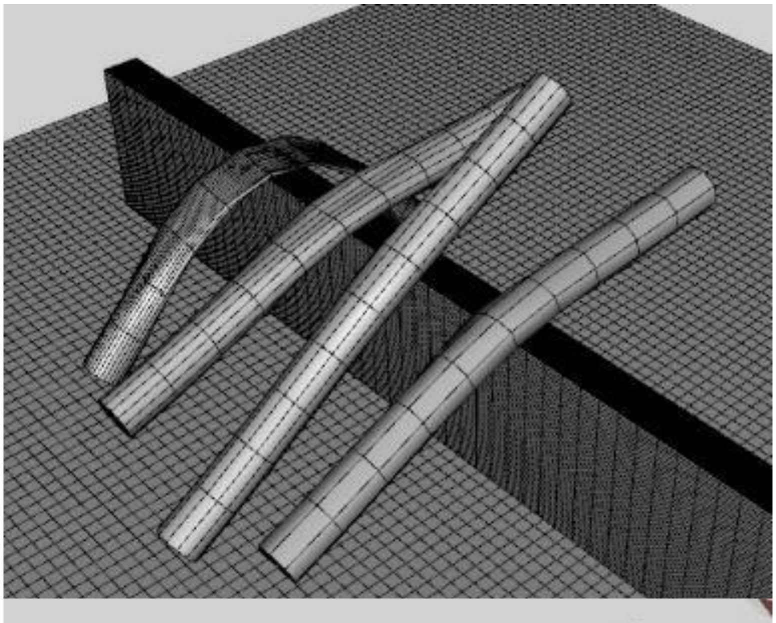
j. Use the following settings for the Simulation surface properties. These were determined by trial and error to minimize collapse.

Segments - Yscale	Buckling Stiffness	Buckling Ratio
10 – 100%	35%	70% (default)
20 – 100%	50%	70%
20 – 200%	35%	70%

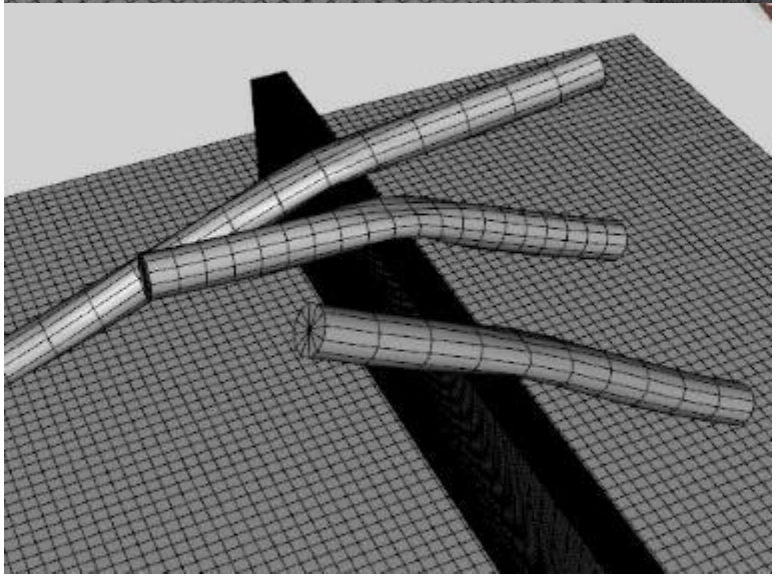
All 3 retain their shape. When there was more sides in the same length, we needed to increase the buckling stiffness. But if we increased the length so the segments were of the same size (10 segments and 20 segments with 200% Y-scale) they can use the same value.



k. Create a small cube and change its scaling to be a beam. Position it so the cylinders fall down across it. As you can see, the 8-, 12-, and 16-sided cylinders mostly retain their shape while the 32-sided cylinder collapses across the beam and into itself. Fine-tuning of the buckling stiffness and ratio would likely help the 16-sided cylinder which has some flattening.



Doing the same with the three 12-sided cylinders.

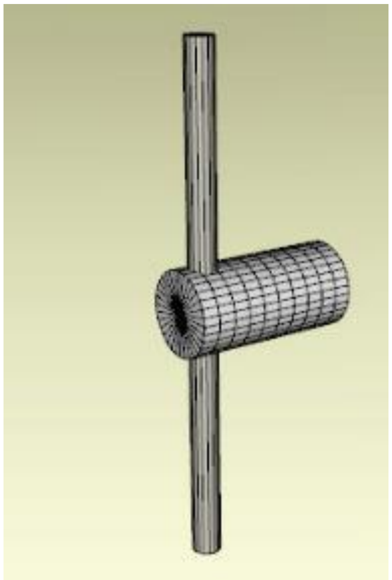


In conclusion, if you want a rope or chain to not flatten in on itself, it’s best to use fewer sides.

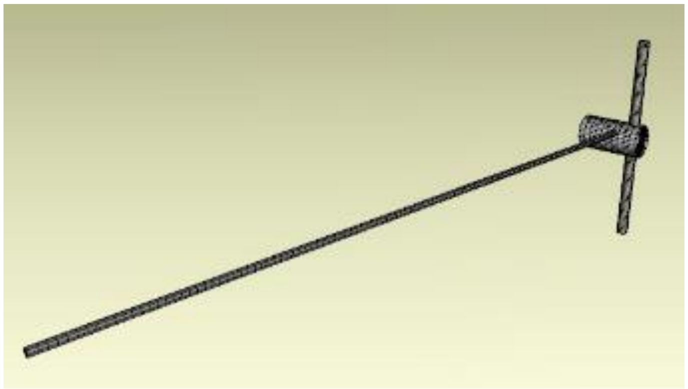
49. **Rotating Cylinder Helper – Wrap up a Rope.** Instead of a translating a cylinder, in this case we rotate a cylinder to wrap up a rope made from a long, thin cylinder.

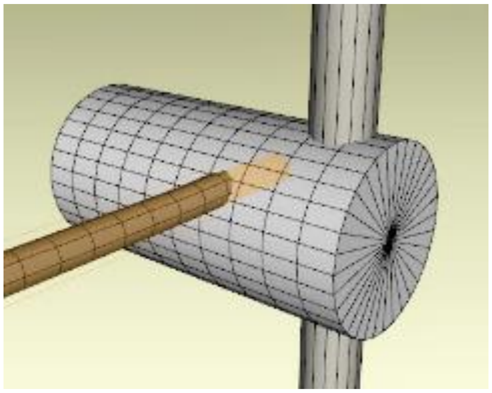
    a. Create a cylinder with 1 foot length, 6 inch diameter, 10 segments, 32 sides, and a Z-positive axis.

    b. Create a cylinder with 3 foot length, 2 inch diameter, 1 segment, 8 sides, and a Y-positive axis. Parent this cylinder to the first one so it makes a handle. See image at right.



    c. Create a cylinder with 10 foot length, 1 inch diameter, 120 segments, 8 sides, and an X-positive axis. Do not parent this cylinder. Position it so one end is embedded into the cylinder that was made in step a.



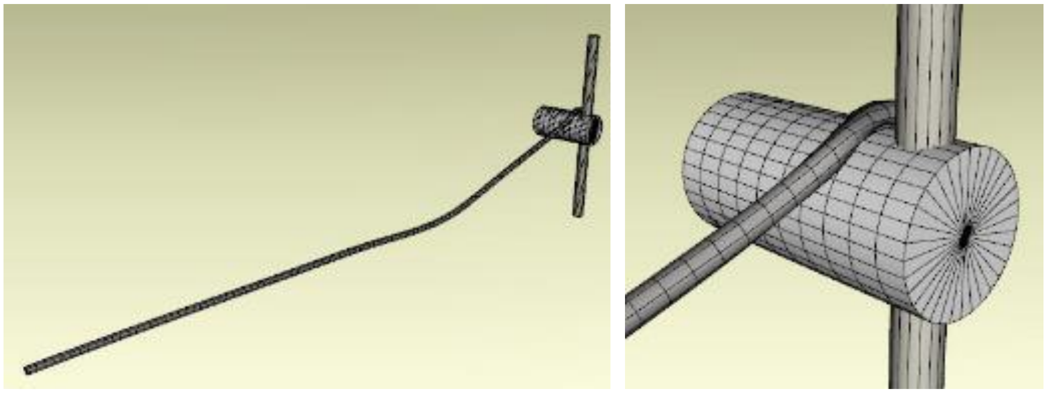


    d. Add a dynamic dForce modifier to the long cylinder made in step c. Set Buckling Stiffness to 70% and Buckling Ratio to 10%.

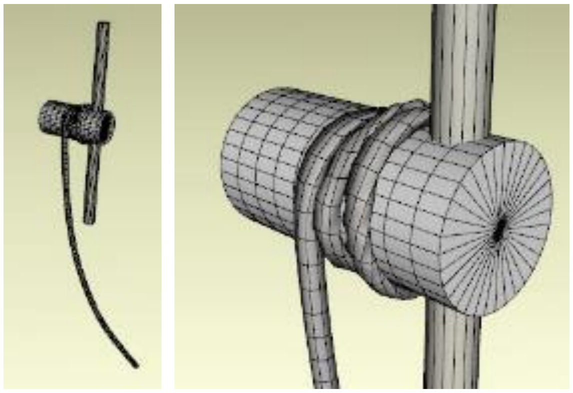
    e. Setup an animation timeline of 110 seconds at 30 fps. Every 8 frames, change the Z-rotation of the cylinder made in step a by -180. Since the cylinder is 6 inches in diameter, it has a circumference of 1.57 feet. To wrap up the 10 foot cylinder (rope), it should take 6.36 rotations to fully wrap it.

    f. Set Collision mode to Best in the Simulation Settings pane and run a simulation. You will see the rope begin to wrap up around the rotating cylinder, and as it proceeds, you will see multiple wraps of rope around it. I found the wrapping ended a little early, after about 5.5 rotations. You will also see that the unattached rope end falls as the wrap proceeds.

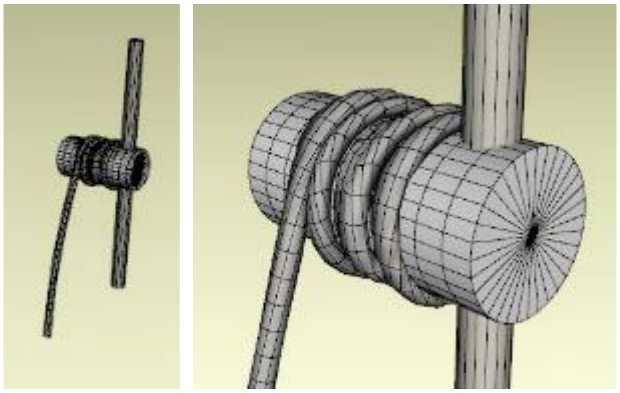
Frame 8:



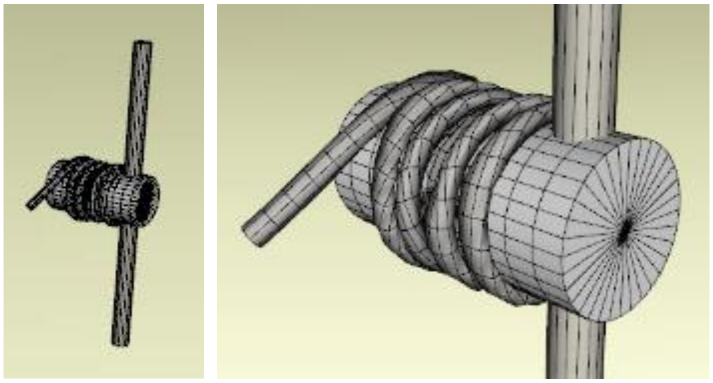
Frame 40:



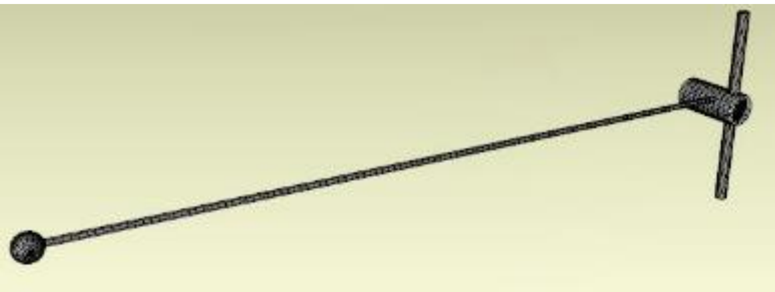
Frame 72:

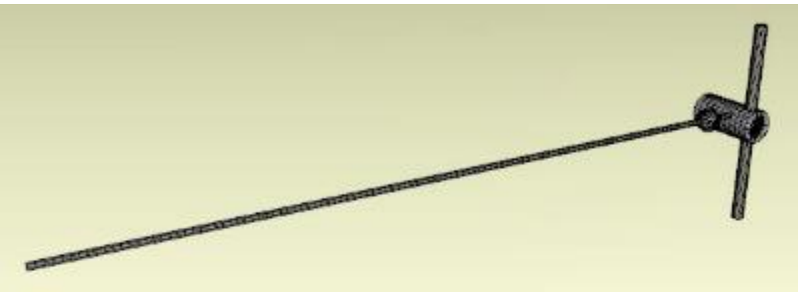


Frame 88:



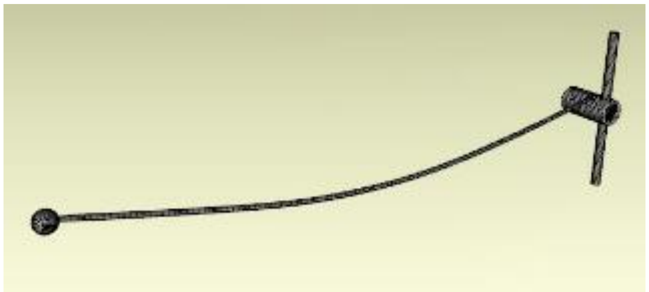
g. **Using a sphere to keep the rope's end from falling.** Next create a small sphere of 4 inch diameter and 16 segments and sides. Position it at the free end of the rope. At frame 0, with the sphere selected, add a key frame. Then at frame 88, slide the sphere next to the wrap-up cylinder.



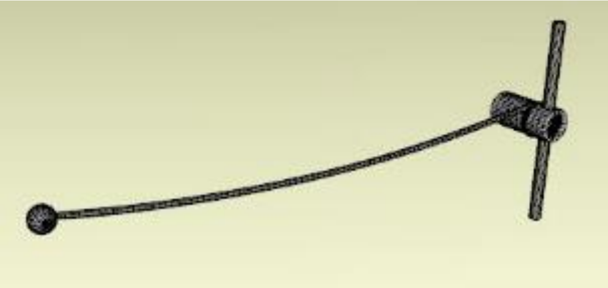


h. Run the simulation. Because the sphere is moving along with the end of the rope, the rope does not fall away as it did with the non-sphere simulation. It does droop, so the rope is being wrapped up slower than the sphere is moving. After the simulation, you can hide the sphere and just run the animation to see the rope move without the sphere attached.

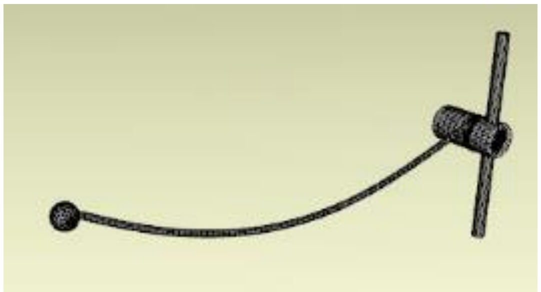
Frame 8:



Frame 32:

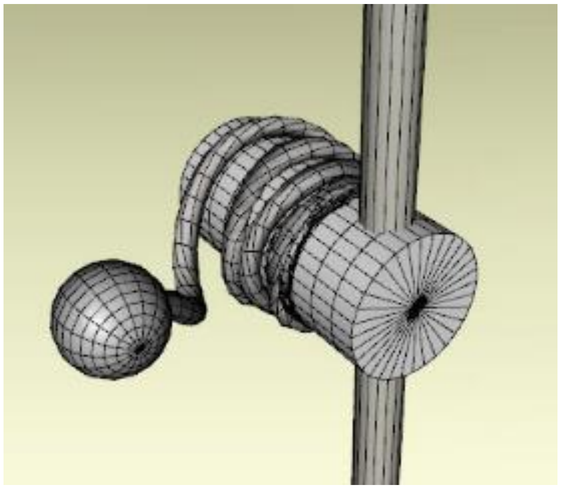


Frame 40:

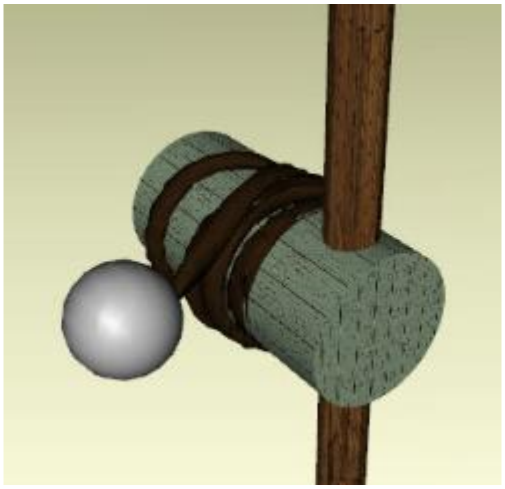


Frame 72:

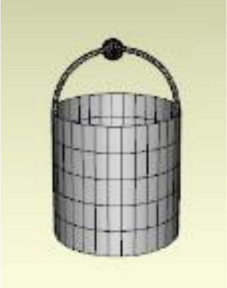
Frame 88:



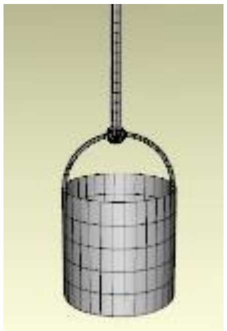
Frame 96:

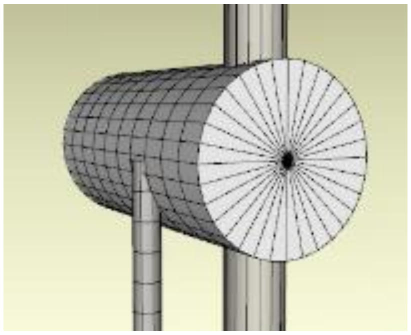


i. **Adding a bucket to the rope.** Now we are going to make a bucket. Create a 1 foot tall, 1 foot diameter, Y-positive axis cylinder. Use the Geometry Editor to select the top faces, hide them, and then delete them. Create a 1 foot major diameter, 0.5 inch minor diameter, Z-positive axis torus. Use the Geometry Editor to select the lower half faces, hide them, and then delete them. Create a 2 inch diameter sphere. Parent the bucket and the handle to the sphere. Assemble the pieces as show to the right.

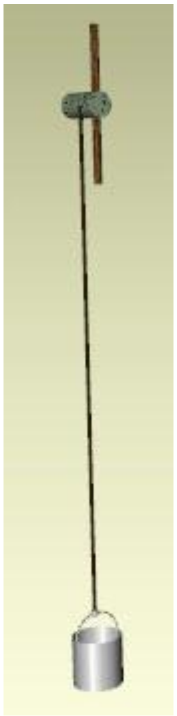


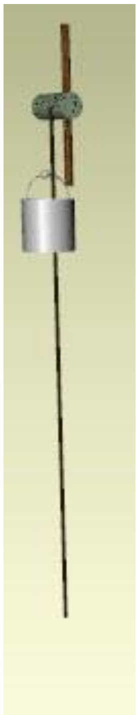
j. Rotate the rope we used before by 90 degrees on the Z-axis. Position one end embedded in the sphere from step I and the other end in the cylinder from step a.





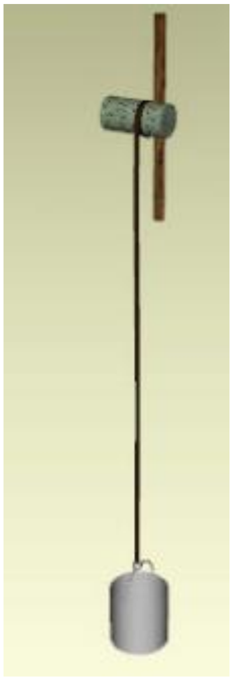
k. On the animation timeline, select the sphere and add a key frame at time 0. Move the sphere and attached bucket up the rope so it hangs below the wrap-up cylinder at frame 88.



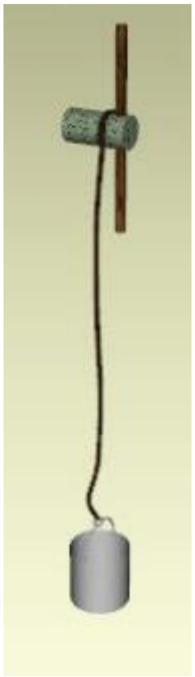


l. Run the simulation. The rope will wrap around the rotating cylinder and lift the bucket. It gets out of sync about halfway through and slack appears in the rope, whereas in real life that would not happen.

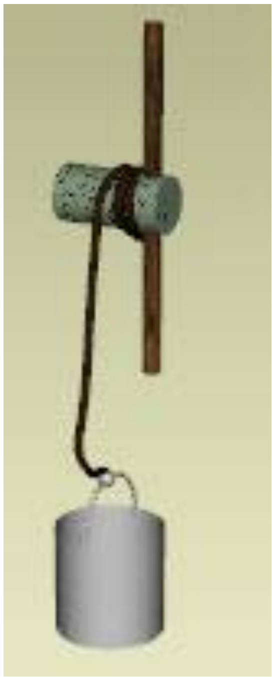
Frame 32:



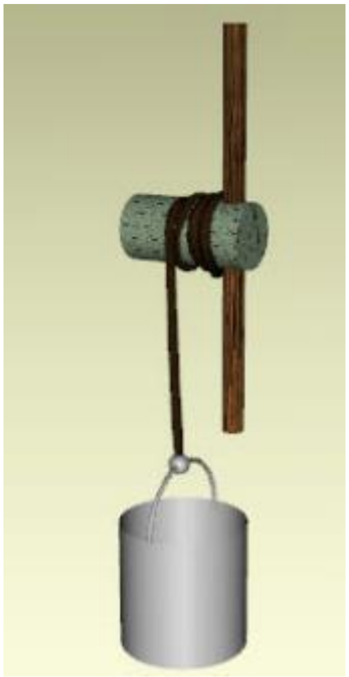
Frame 40:



Frame 72:

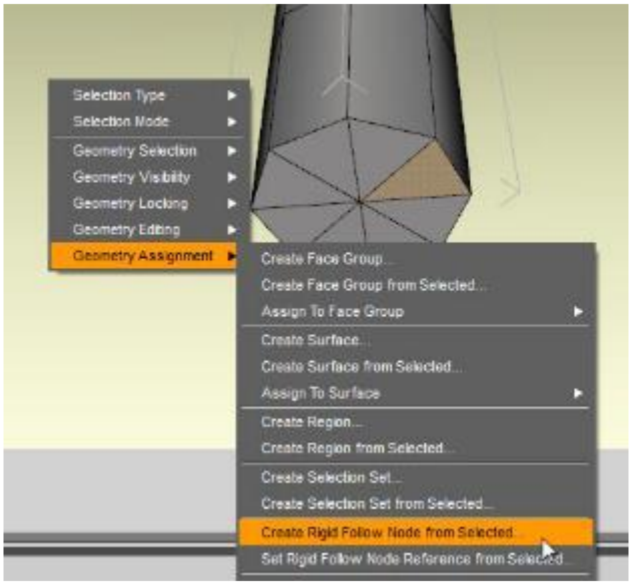


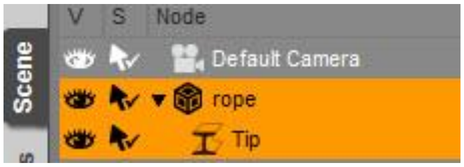
Frame 88:



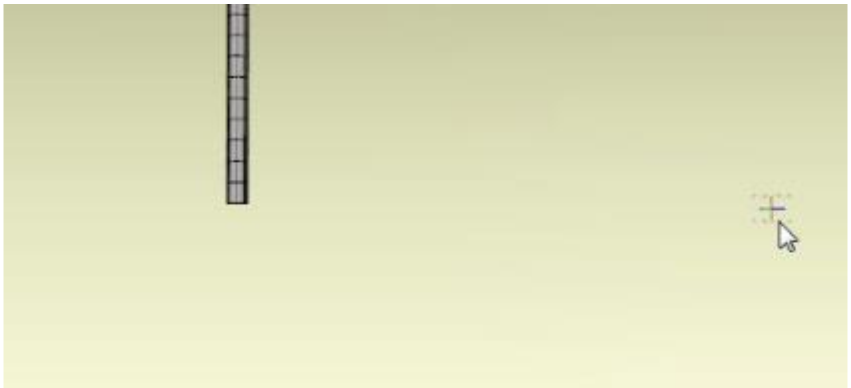
m. **Using a rigid follow node to attach the bucket.** Instead of using an animated sphere on the bucket to raise it with the rope, we use a rigid follow node.

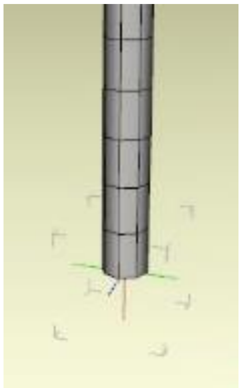
n. Select the rope, zoom in on the free end, and using the Geometry Editor, pick 1 face from the end. Right-click in the viewport and choose Geometry Assignment/Create Rigid Follow Node from Selected. A child with an I-beam icon will be created under the rope cylinder.



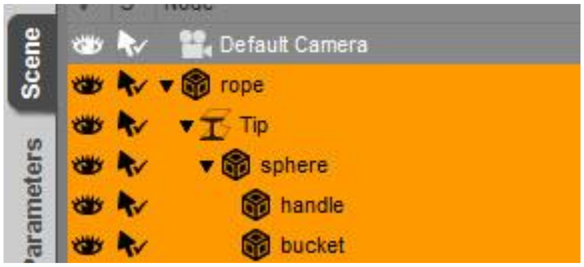


o. For some reason, the node is not located at the end of the rope. I used the X, Y, Z translate dials to move it into location.



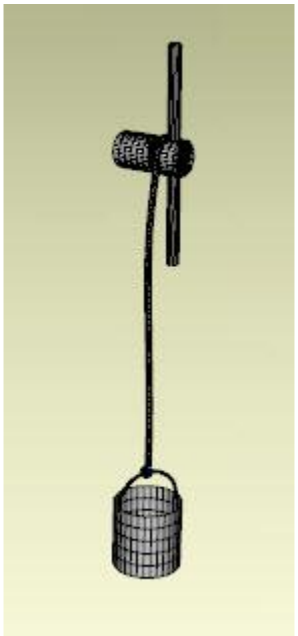


p. Run the dForce simulation with the bucket assembly not attached to the node. After the simulation is done, parent the bucket assembly to the node.

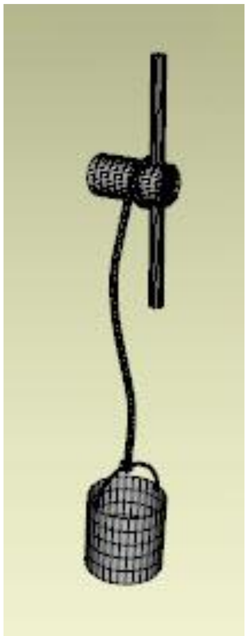


Now when you run the simulation, you will see the bucket rise. There is some turning and twisting as the bucket follows the action of the rope end.

Frame 56:

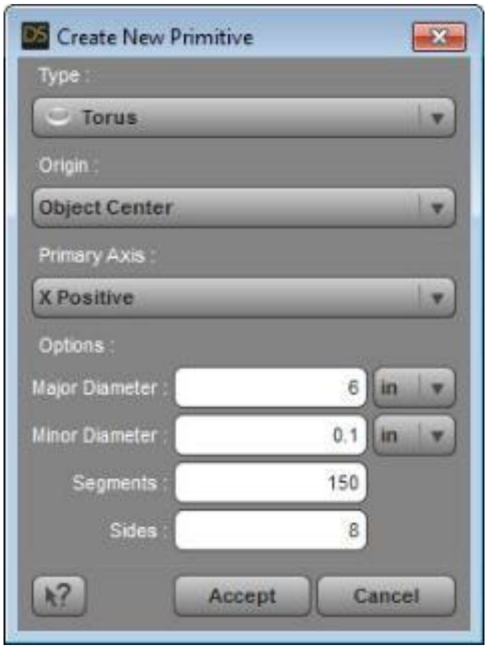


Frame 64:



50. **Creating a Necklace.** This uses part of a jewelry product purchased from Daz combined with a primitive torus to make a necklace.

   a. Create a primitive torus using the values in the image. Add a dFforce dynamic modifier to it. Set Self Collide to Off in the Surfaces pane.



Create New Primitive

Type : Torus

Origin : Object Center

Primary Axis : X Positive

Options :

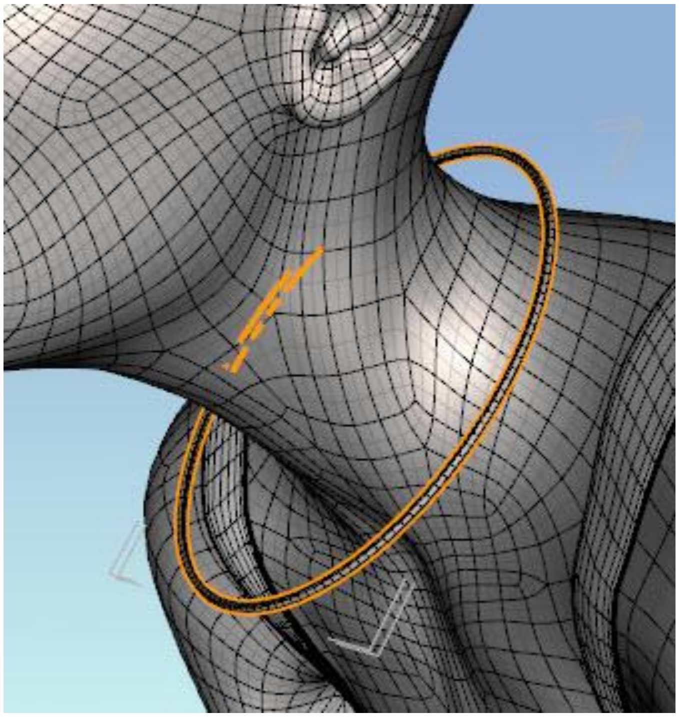
Major Diameter : 6 in

Minor Diameter : 0.1 in

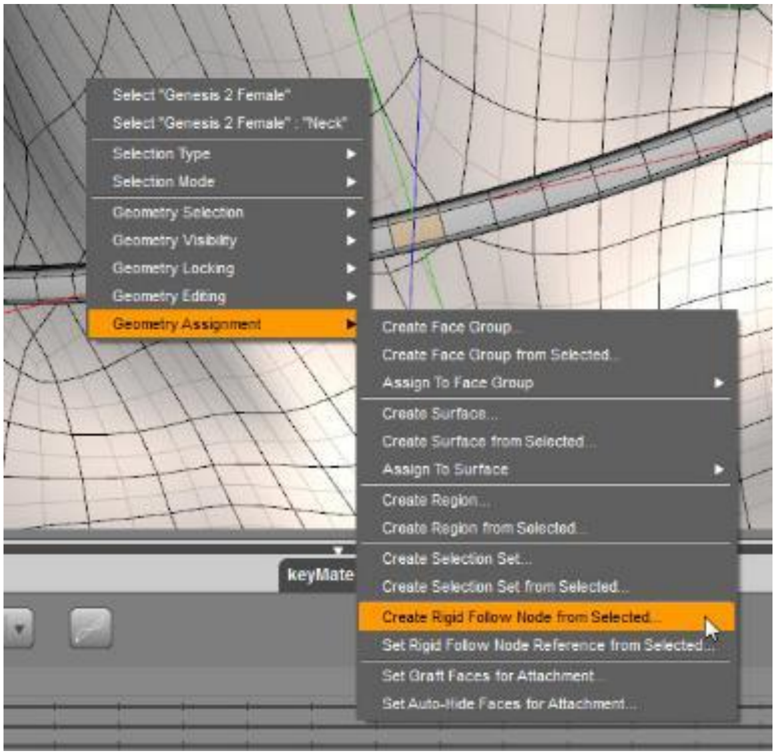
Segments : 150

Sides : 8

   b. Position the torus around the figure’s neck.

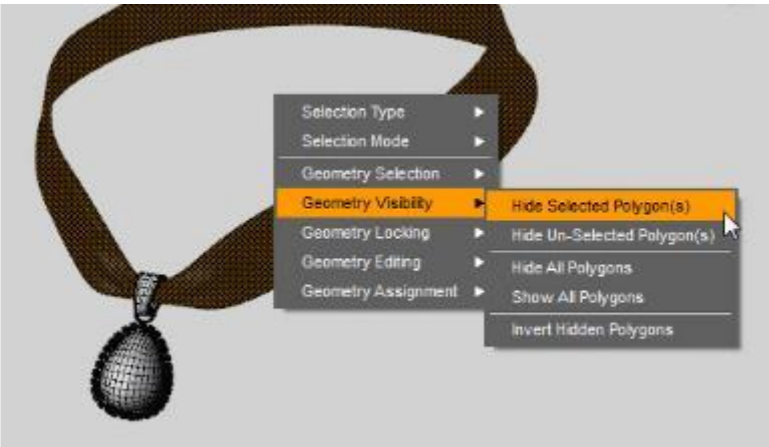


c. Use the Geometry Editor to select a single face in the lower edge of the torus in the center of the neck. Right click in the viewport and select Geometry Assignment/Create Rigid Follow Node from Selected… An i-beam icon will appear as a child under the Torus in the Scene pane.

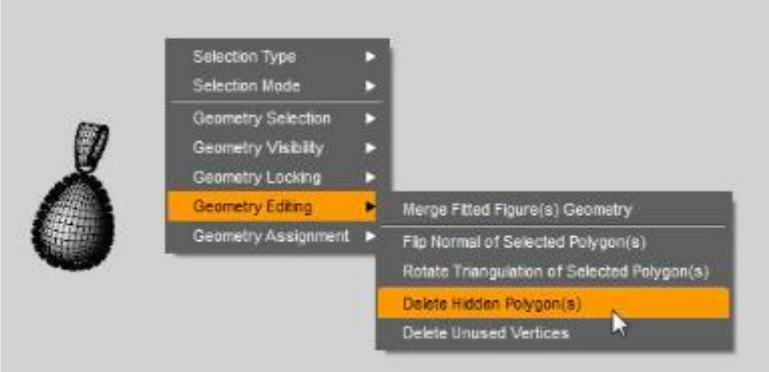

Select "Genesis 2 Female"
Select "Genesis 2 Female" : "Neck"
Selection Type
Selection Mode
Geometry Selection
Geometry Visibility
Geometry Locking
Geometry Editing
Geometry Assignment
Create Face Group...
Create Face Group from Selected...
Assign To Face Group
Create Surface...
Create Surface from Selected...
Assign To Surface
Create Region...
Create Region from Selected...
Create Selection Set...
Create Selection Set from Selected...
Create Rigid Follow Node from Selected...
Set Rigid Follow Node Reference from Selected...
Set Graft Faces for Attachment
Set Auto-Hide Faces for Attachment...
keyMate

d. For the necklace's pendant, I used the product C-Elegance Jewelry. Although no longer sold at Daz, the steps would be the same for similar products.

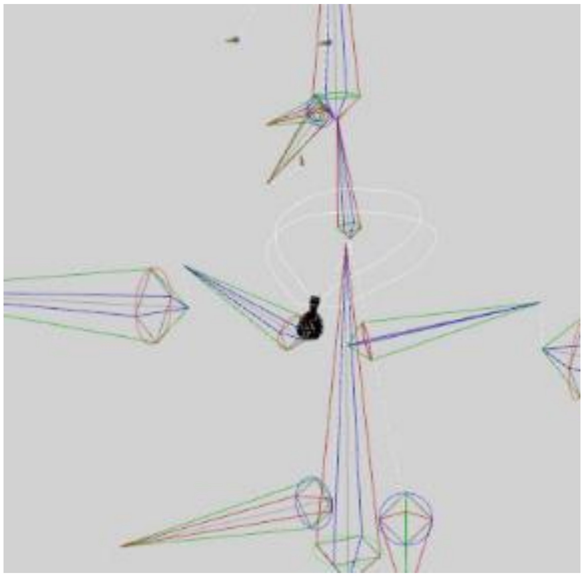
e. Load the necklace. Using the Geometry Editor, select one face on the ribbon (or chain or any other part of the jewelry you need to remove). Then type Ctrl+* to select all connected faces. Then right click in viewport and choose Geometry Visibility/Hide Selected Polygon(s).



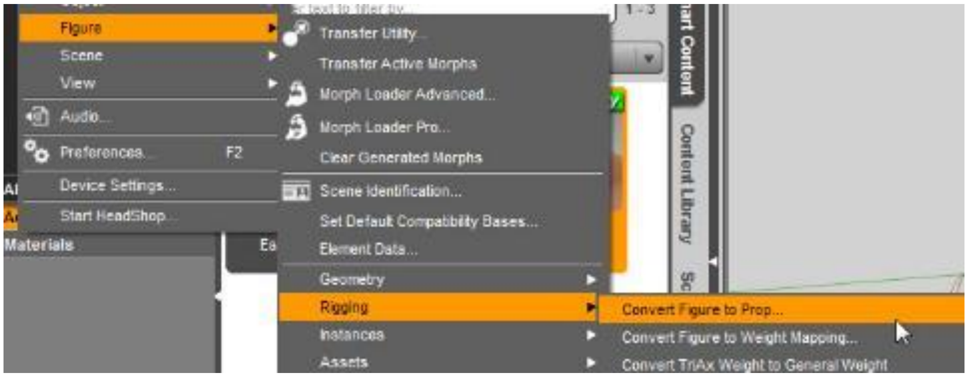
f. Now right click in viewport and choose Geometry Editing/Delete Hidden Polygons. This eliminates the ribbon so we can use the torus as the chain.



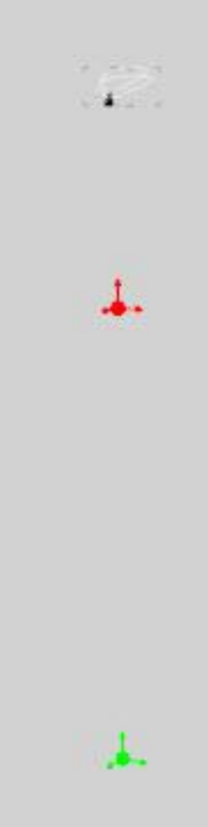
g. From the Tool Settings pane, choose the Joint Editor. You will see a lot of bones indicating the product was rigged as a figure. (If your jewel is already a prop, you can skip steps g-i).

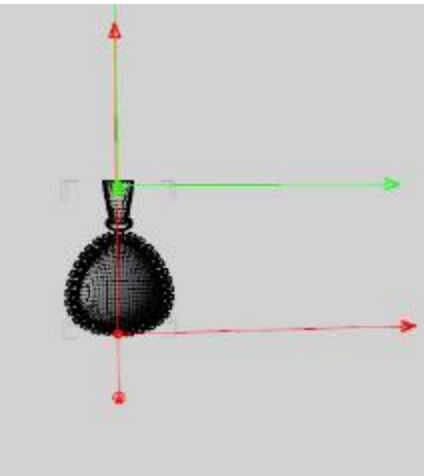


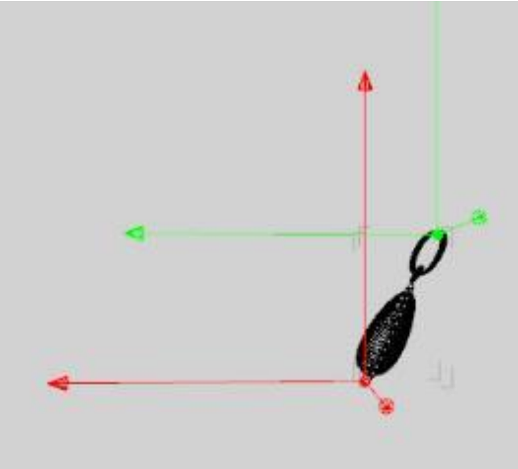
h. Select the necklace and then Edit/Figure/Rigging/Convert to Prop. This removes the bones.



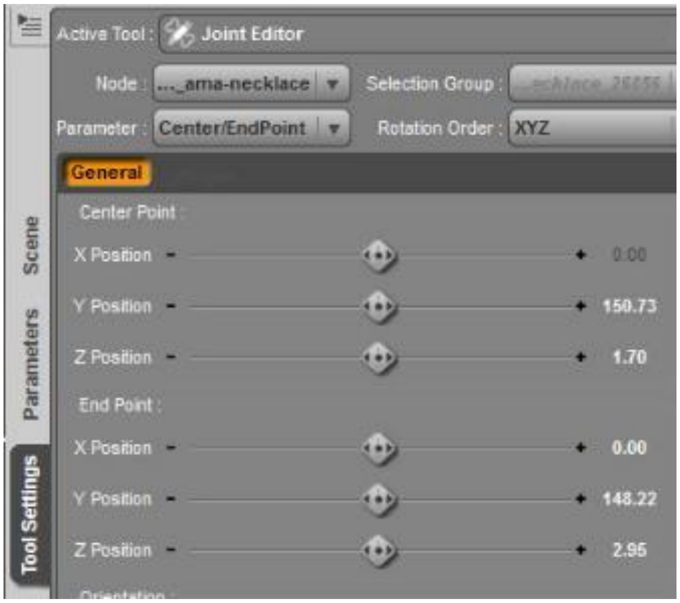
i. With the Joint Editor active, zoom out and you will find the Center (green) and End (red) points far from the pendant. Drag the red and green dots up until they are next to the pendant. Then zoom in and adjust so the green dot is at the center top of the bail and the red dot is at the center bottom of the pendant. After these adjustments, the center of rotation of the pendant is the green dot.







Here are the Joint Editor settings I used:



Active Tool : Joint Editor

Node : ..._ama-necklace

Selection Group : ...ecklace_25655

Parameter : Center/EndPoint

Rotation Order : XYZ

Center Point :

X Position 0.00

Y Position 150.73

Z Position 1.70

End Point :

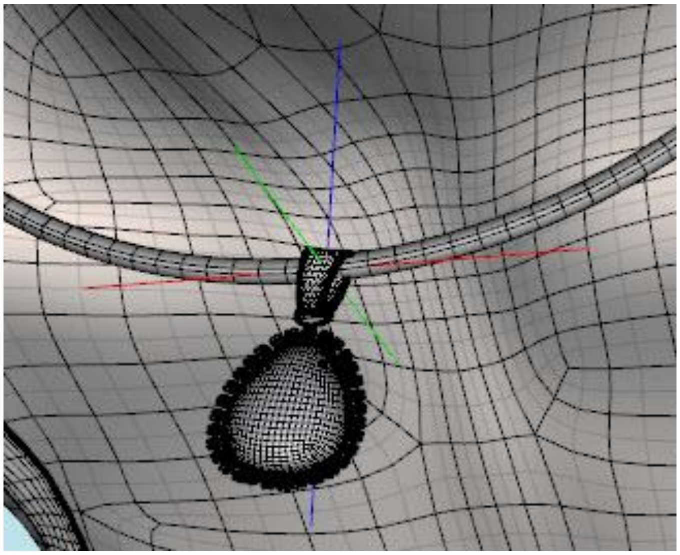
X Position 0.00

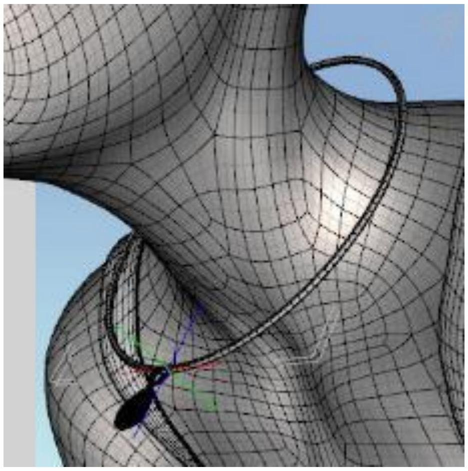
Y Position 148.22

Z Position 2.95

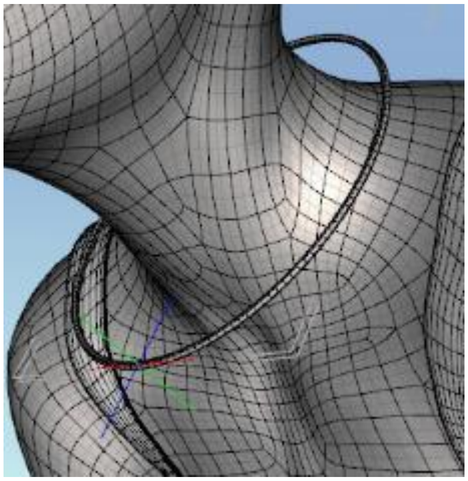
Orientation :

j. Add the pendant as a child of the rigid follow node. Using the universal tool, position the bail of the pendant around the torus at the point of the rigid follow node (the node location is indicated by a red-blue-green icon; see image at right).

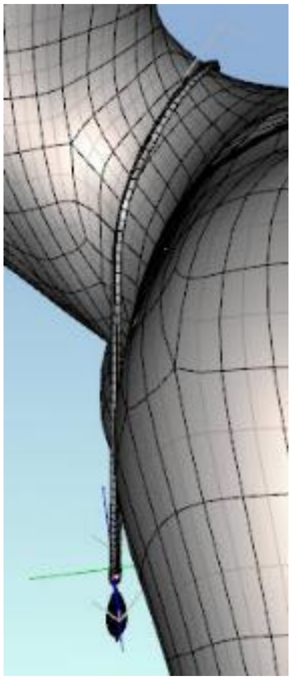


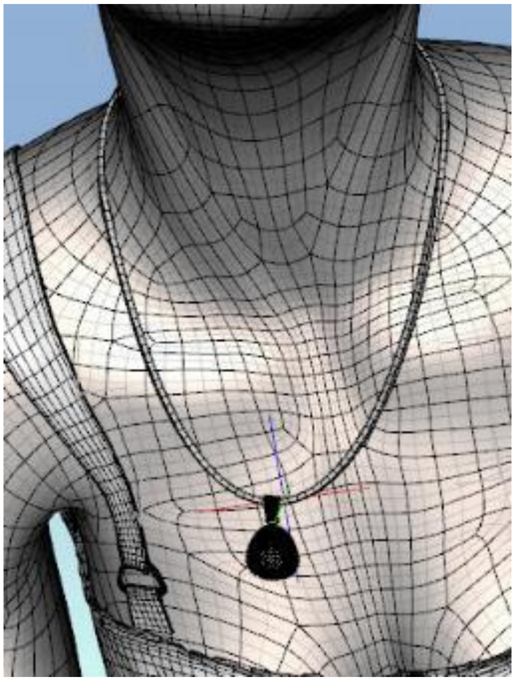


k. Hide the pendant by clicking on the eye in the Scene pane. If left visible, the pendant will interact with the chain and may cause it to explode.



l. Run an animated dForce simulation. Images show result after the pendant is turned back on. The pendant and chain drapes as you would expect from gravity.

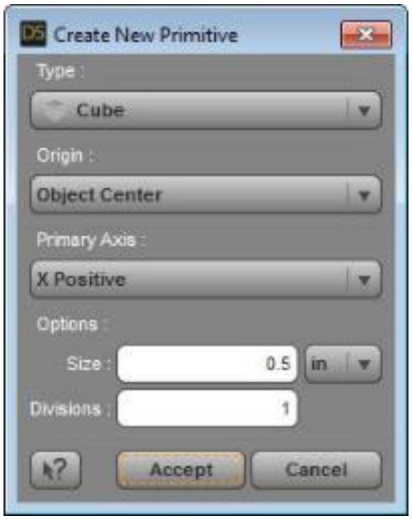




m. An iRay render is at right. I used a braided metal shader for the chain.

n. **Create a V in the chain.** You may have noticed the lower end of the chain is curved. For some types of chains (e.g., braided metal) that may be appropriate but for a light chain, the pendant would pull the chain into a V. I could not find a combination of dForce settings that would create the V but you can use a cube helper to do so.

o. Create a small primitive cube using the parameters shown in the image. Parent the cube to the torus and position it to be right over the location of the rigid follow node. Set overall scale to 50% and Y scale to 300% so you have a long rectangular box.



Type : Cube

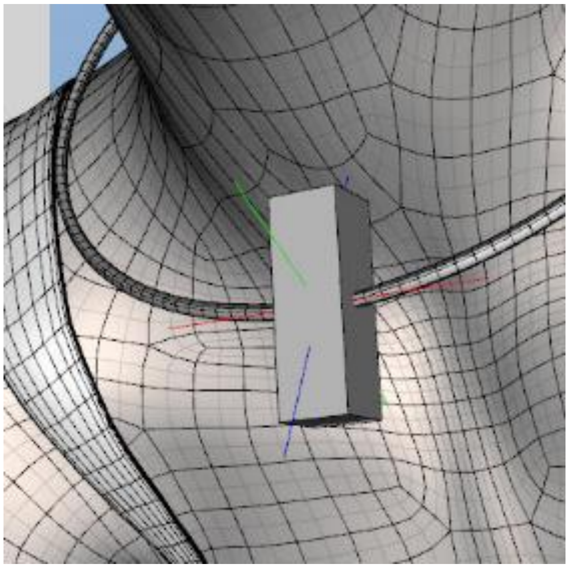
Origin : Object Center

Primary Axis : X Positive

Options :

Size : 0.5 in

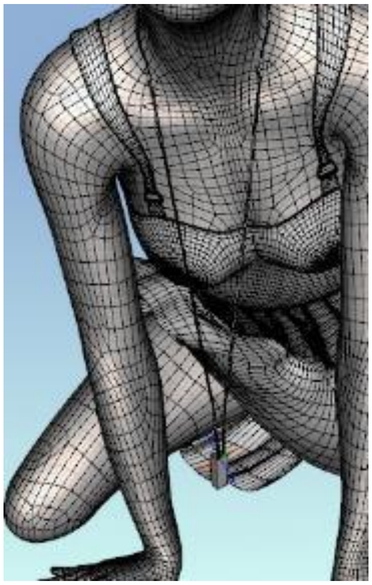
Divisions : 1



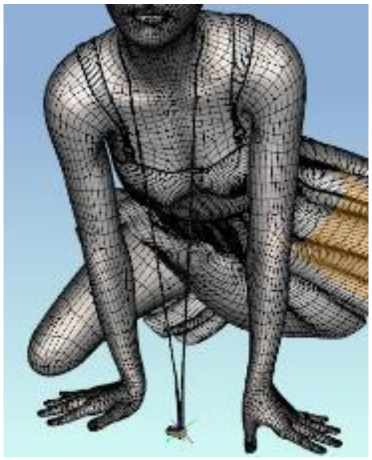
p. Add a Dforce modifier to the cube. Be sure the pendant is hidden otherwise it will interfere in the simulation and at least on my computer crashed Daz Studio. Run an animated simulation for 40 frames but no keyframes or positioning is needed.

q. With default surface parameters, the box will cause extreme elongation of the chain and eventually an explosion of the mesh. I found reducing the Dynamics Strength on the cube to between 0.85 and 0.90 works to suppress elongation and mesh explosion. Lower values (<0.85) give a shorter chain but the cube (and hence the location of the pendant) will deform the chain into a W as the cube doesn't fall far enough. Higher values (>0.90) result in too much stretching. My best value was 0.88.

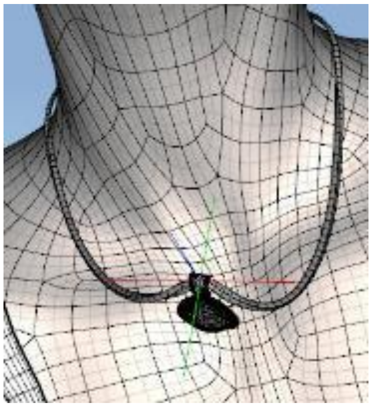
1.0 Dynamics Strength (frame before mesh explosion). You can see how the chain has stretched:



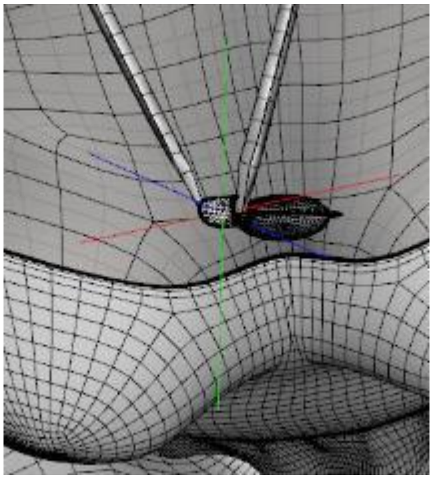
0.97 Dynamics Strength (frame before mesh explosion). You can see the chain has stretched to the floor:

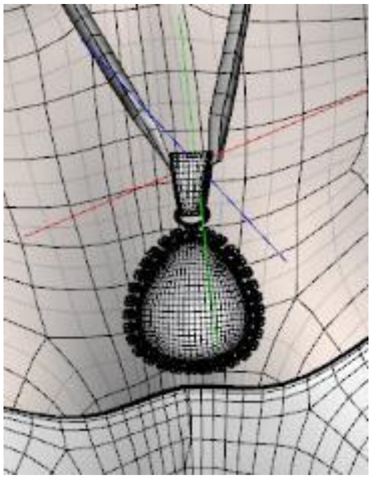


0.85 Dynamics Strength. The cube/pendant has twisted and formed a W in the chain.



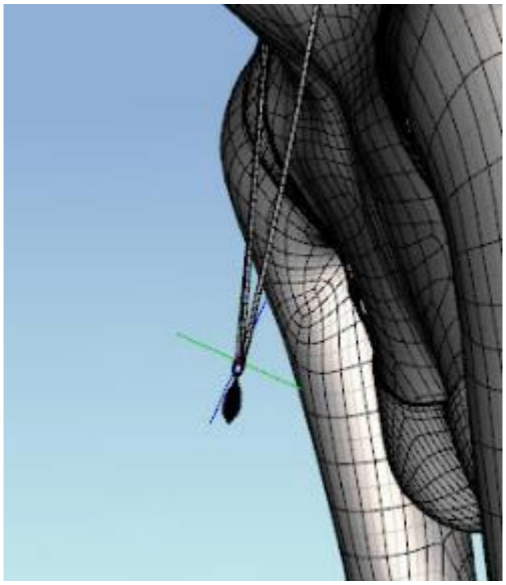
0.88 Dynamics Strength. The cube/pendant has twisted but has formed a V in the chain. Unlike simulation without the cube, the rigid follow node has twisted and I had to rotate the pendant after simulation so it would hang down.



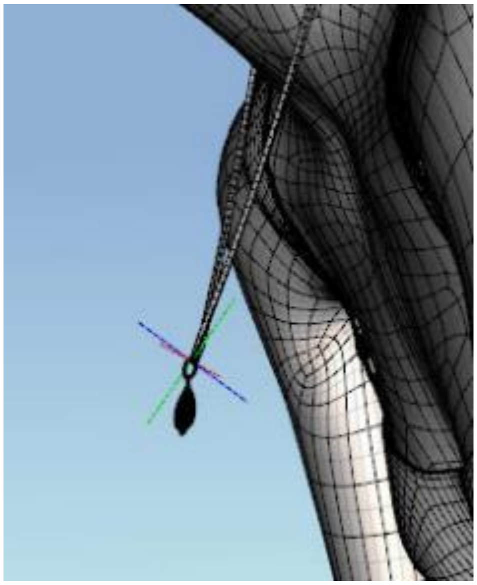


r. Another issue with the use of the helper cube is the chain hovers away from the chest. Frame 12 gives a somewhat better position but I could not achieve what was done without the cube. In both images below, the pendant has been rotated to be parallel with the chain.

Frame 12, Dynamics Strength 0.88:

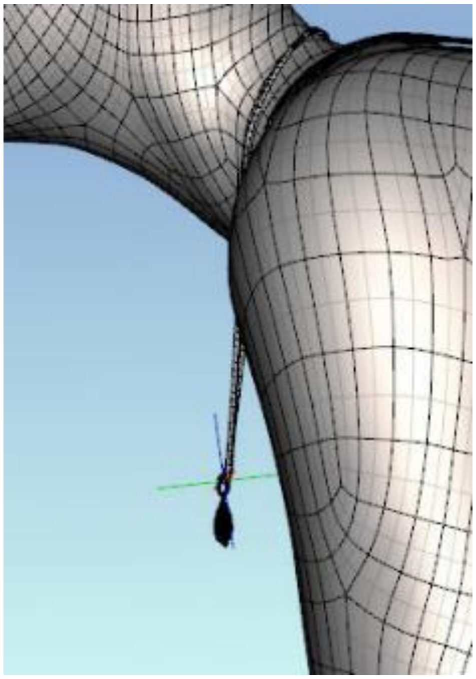


Frame 40, Dynamics Strength 0.88:



s. You can correct both the problems mentioned by running a simulation, going to frame 40 and moving the cube closer to the chest and rotating the pendant into position, then running the simulation a second time. Be sure to hide the pendant before resimulating to avoid a crash.

If you make a big change in the figure's pose, you'll want to delete the added key frame in frame 40 for the cube then do the double simulation (simulate-move cube-simulate).



A finished render of the V chain:

Go to Menu