Part 3

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

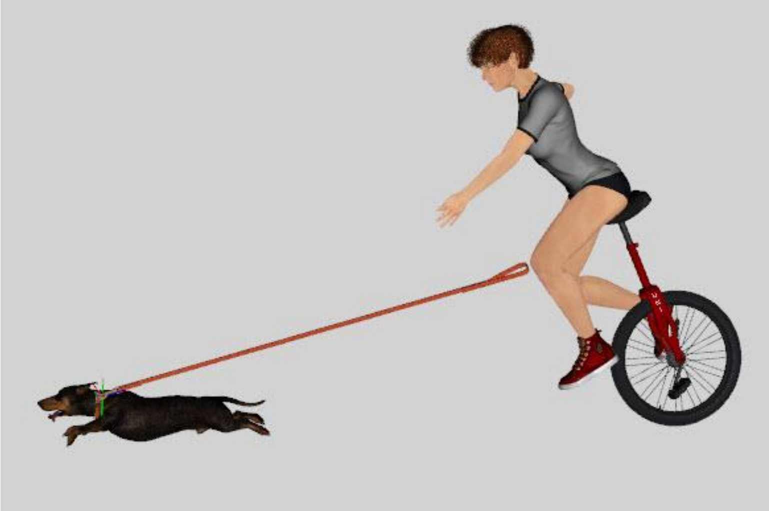
Copyright 2017-19 by Rich Schafermeyer
(RGcincy on Daz Forums)

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.

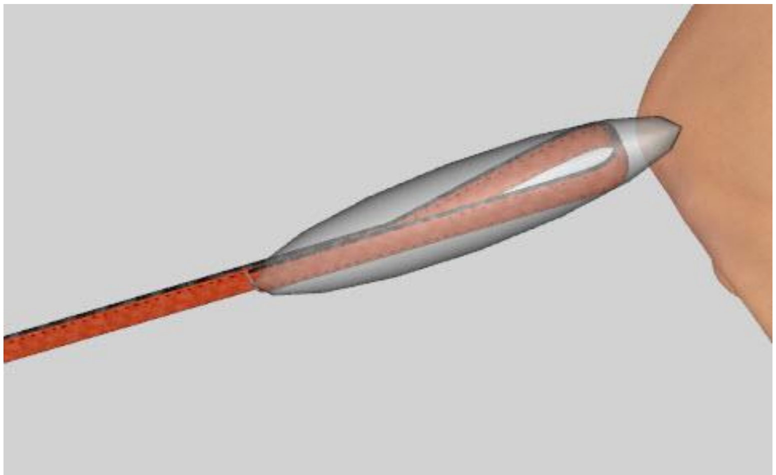
Index

51. **Dog Leash.** I picked up the new Daz Dog 8 as well as Collars and Leashes for Daz Dog(s) 8. The leash has 19 bones with parameters to control the positioning but it can be a lot of work to adjust multiple bones to get the leash in the right spot. Here I used a helper ellipsoid to move the leash into position.

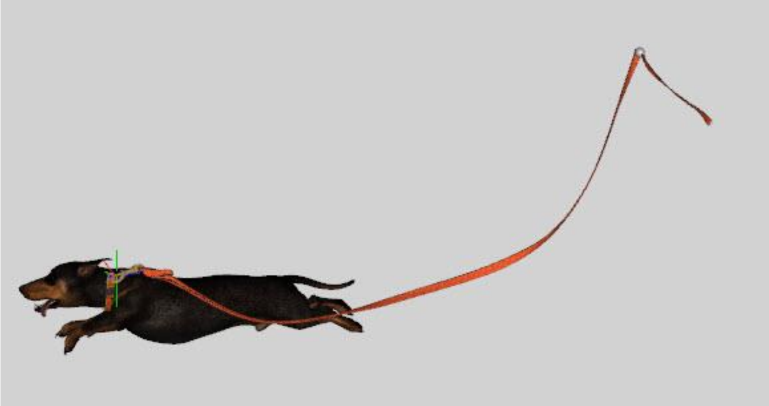
    a. Load the dog and the collar/leash as well as other scene elements. Rotate the leash as needed so it is near the figure but not inside the hand.

    b. Create a primitive sphere of 16 segments and 16 sides. Elongate it along the Z axis. Scale it so the ellipsoid covers the handle. This is the helper that will move the leash to the figure's hand.

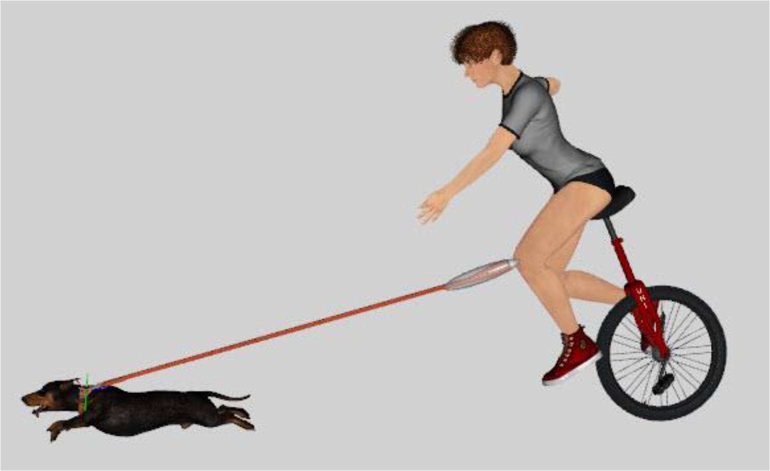
    

    The reason to cover the handle is to keep it from coming apart during simulation. If only a small sphere is used, you'll get the result shown in the figure to the right.

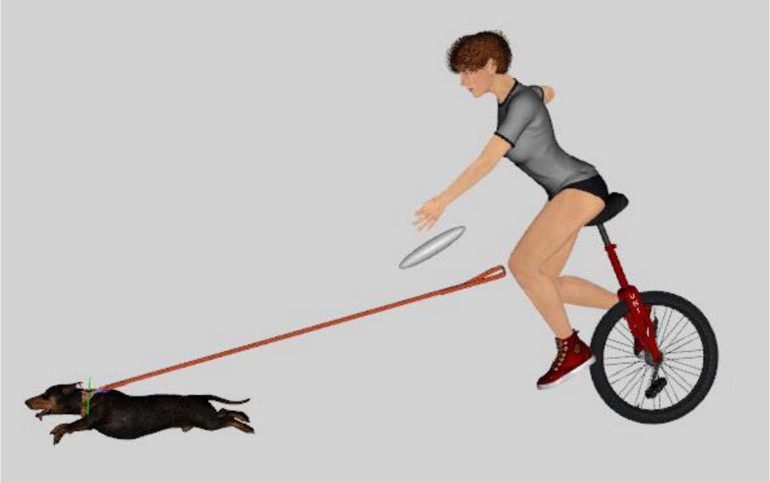
    

    c. Set up an animation with the ellipsoid moving to a position below the hand at frame 8 and into the hand at frame 15. At frame 15, rotate the ellipsoid so it is parallel with the forearm of the figure.

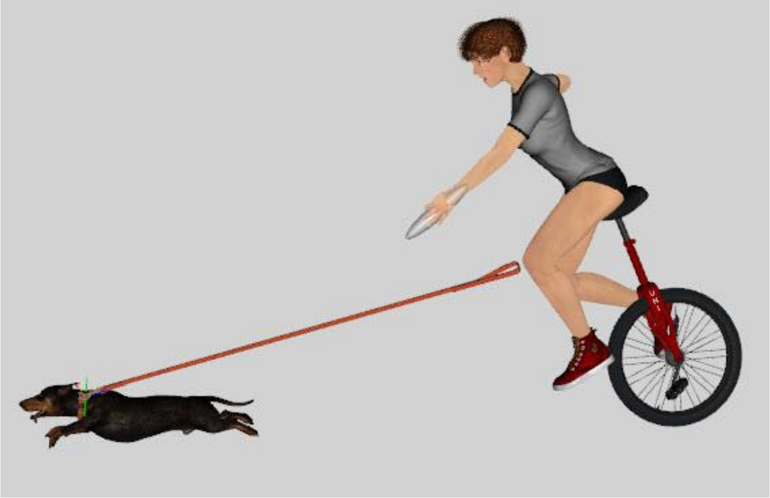
Frame 0:



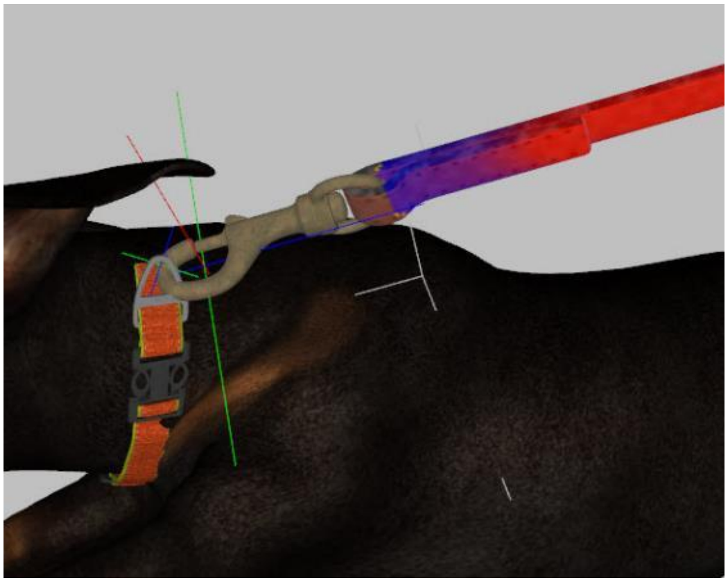
Frame 8:



Frame 15:

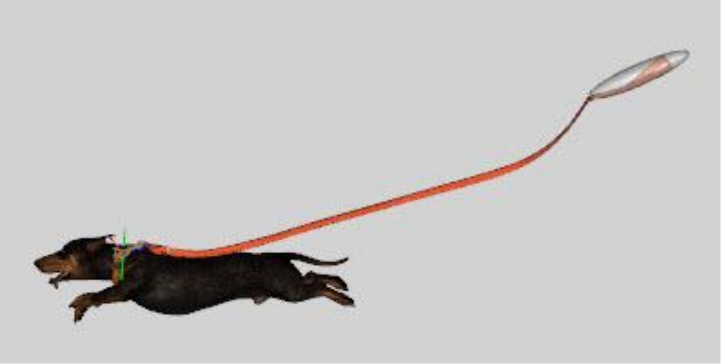


d. Add a dForce dynamic modifier to the leash and then add a dForce modifier weight node. Paint a weight map so the influence is removed from the clasp next to the collar. See image.

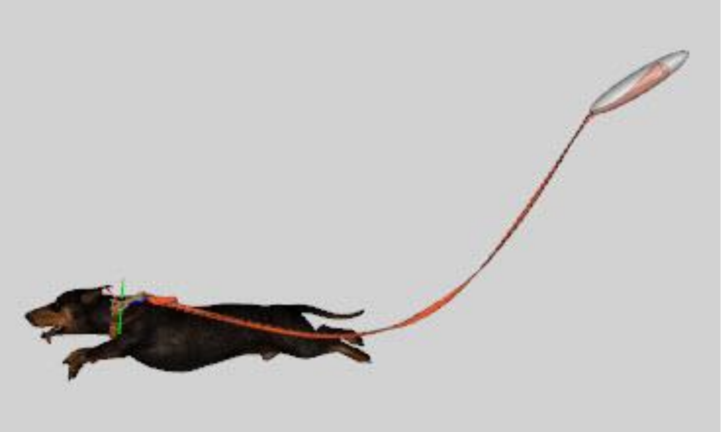


e. Hide the figure so it doesn't interfere with the leash's movement. Run the simulation. You will see a result like this:

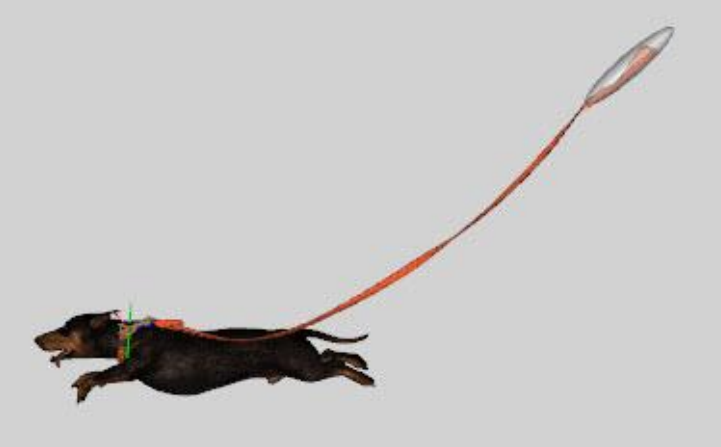
Frame 4:



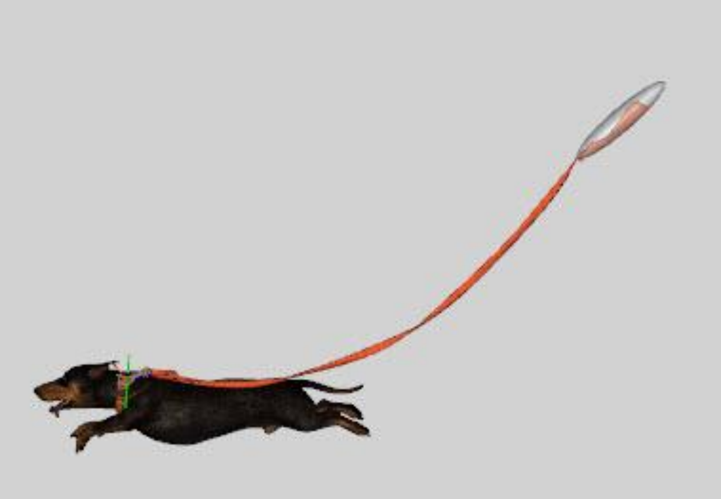
Frame 8:



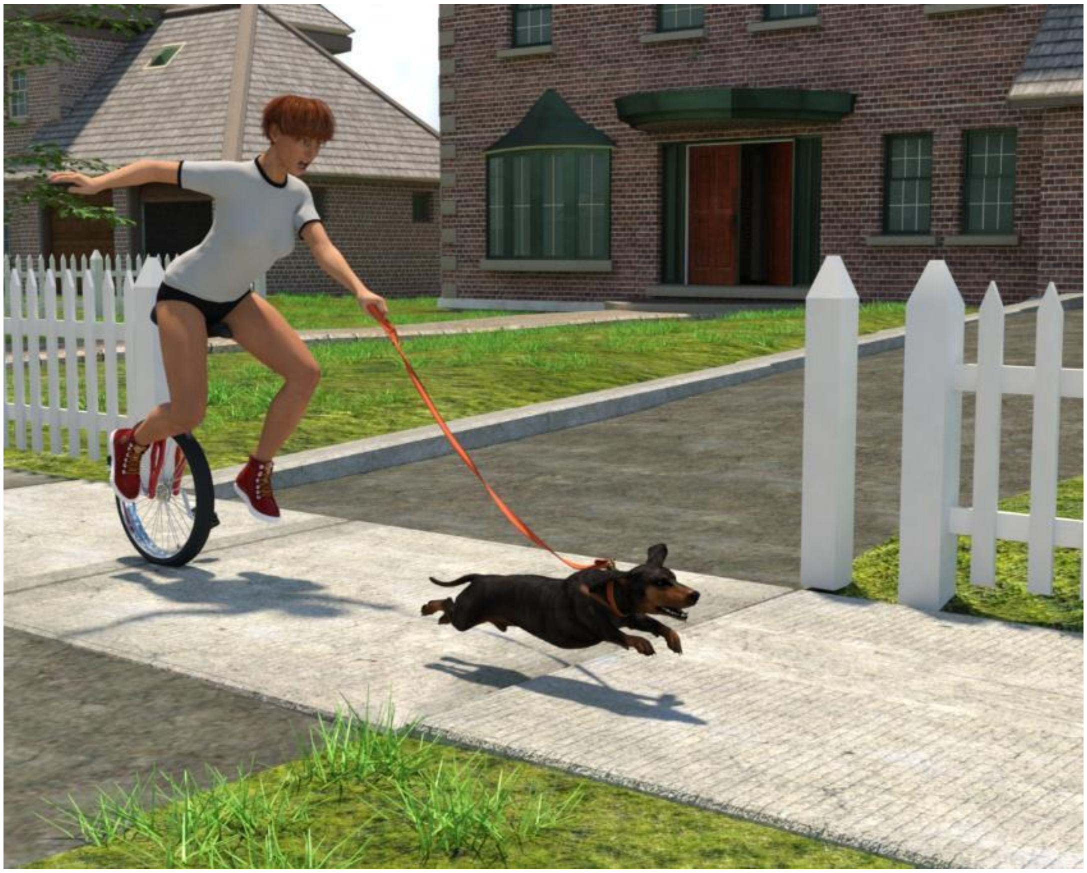
Frame 15:



Frame 30:



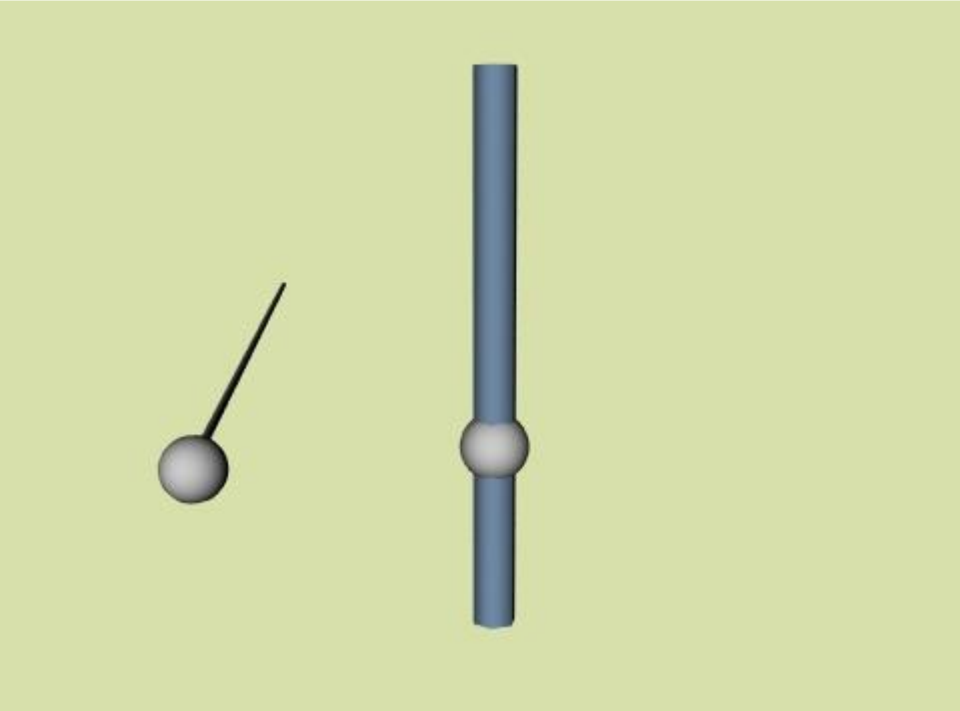
f. Finished renders with the dForce leash:



52. **Effect of Density on Rotating and Horizontal Ropes.** I've conducted a lot of experiments, some work out and some don't. This one was to see if you could use a helper sphere to swirl a rope around a central point. I always start with default surface settings to see what happens and then look for adjustments. In this case, it turns out super low density values are the key.

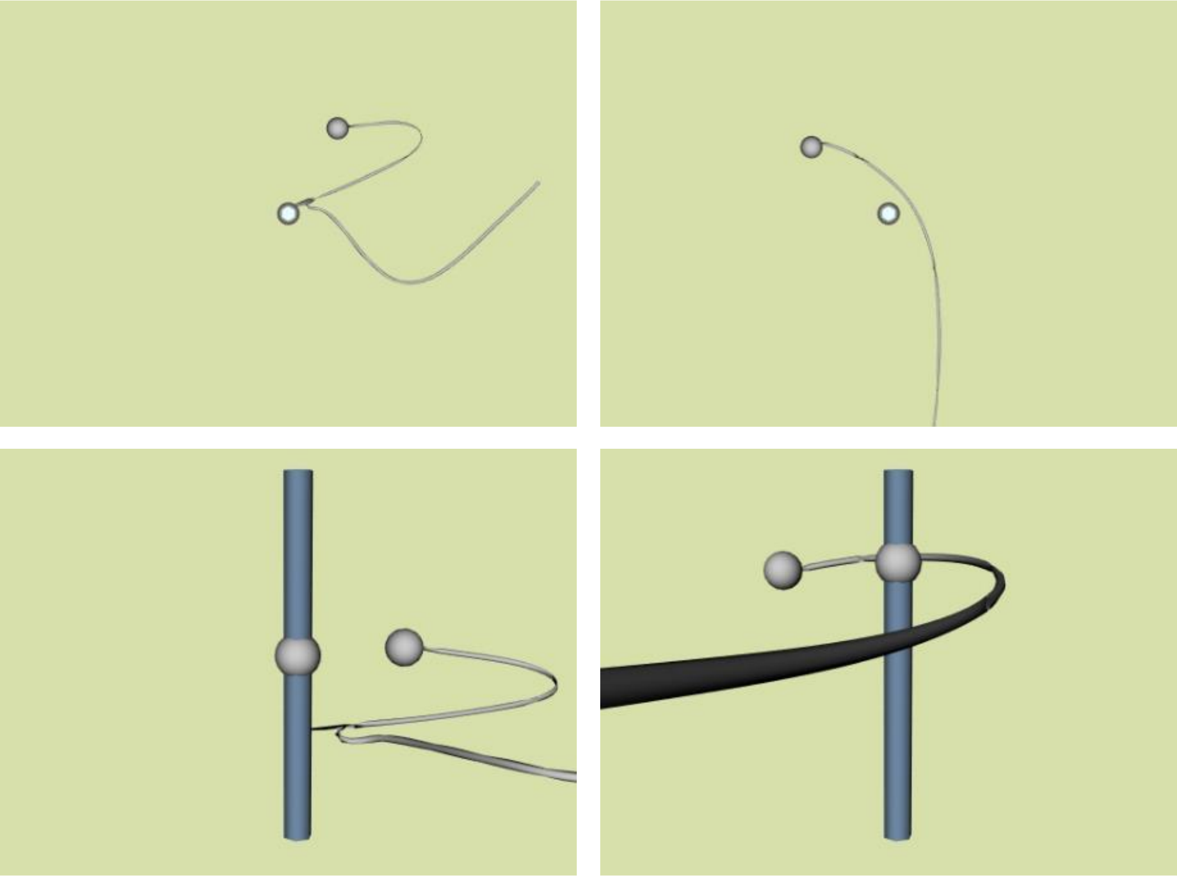
   a. Create a central pole with a cylinder of 4 feet length and 4 inch diameter. Create two spheres of 6 inch diameter, 16 segments, and 16 sides. Put one on the pole near its bottom and the other away from the pole. Parent the sphere that is away from the pole to the one on the pole.

   b. Create a long cylinder that is 100 feet long, 1 inch diameter, with 100 segments and 6 sides. Position one end of the cylinder into the sphere as shown below. Do not parent the cylinder to the sphere. Add a dForce dynamic modifier to the cylinder. Turn self-collide off in the Surfaces pane.

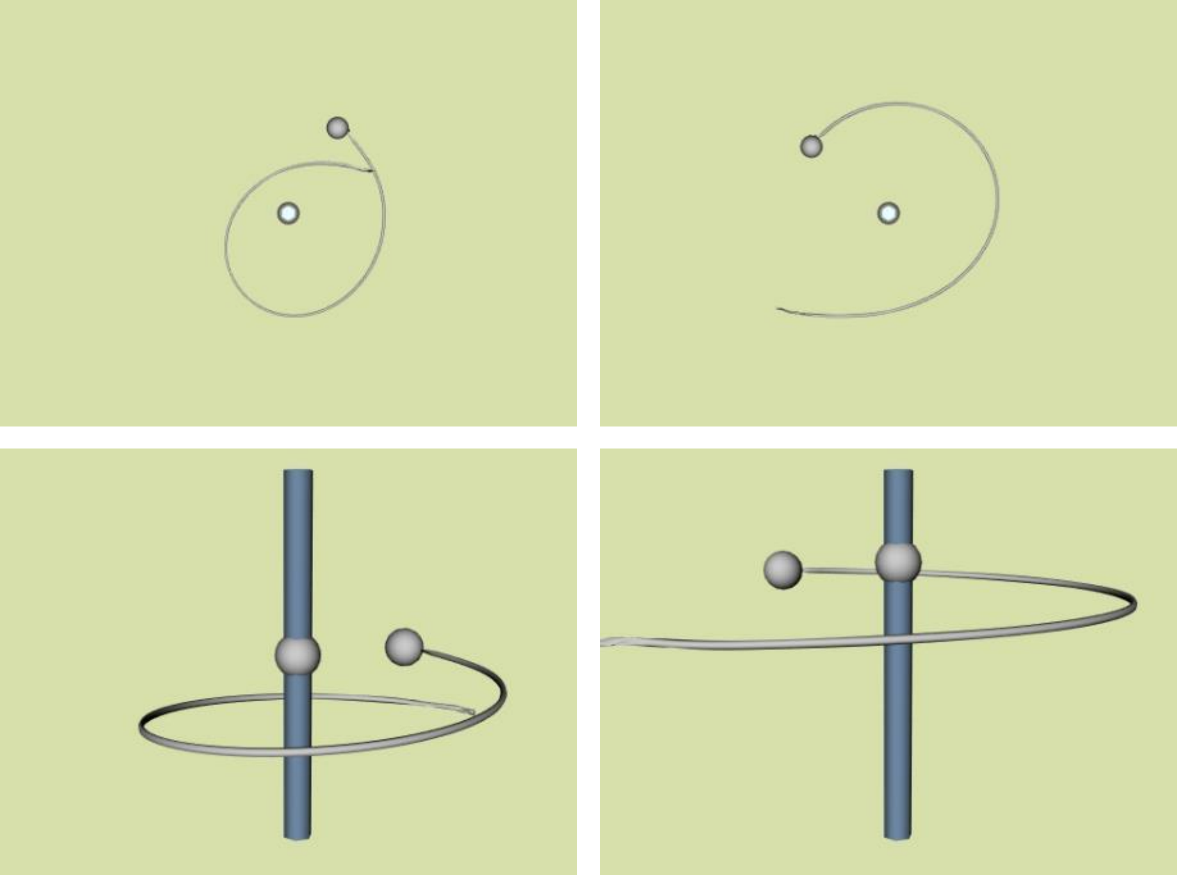


   c. Set up an animation timeline with 32 frames. Set a key frame for the sphere on the pole at frame 0. At frame 32, move the sphere up the pole and set Y-rotate to 1440 (which will give you 4 complete rotations). As the inner sphere rotates, the outer sphere will sweep in a wide arc around the pole, pulling the cylinder (rope) behind it.

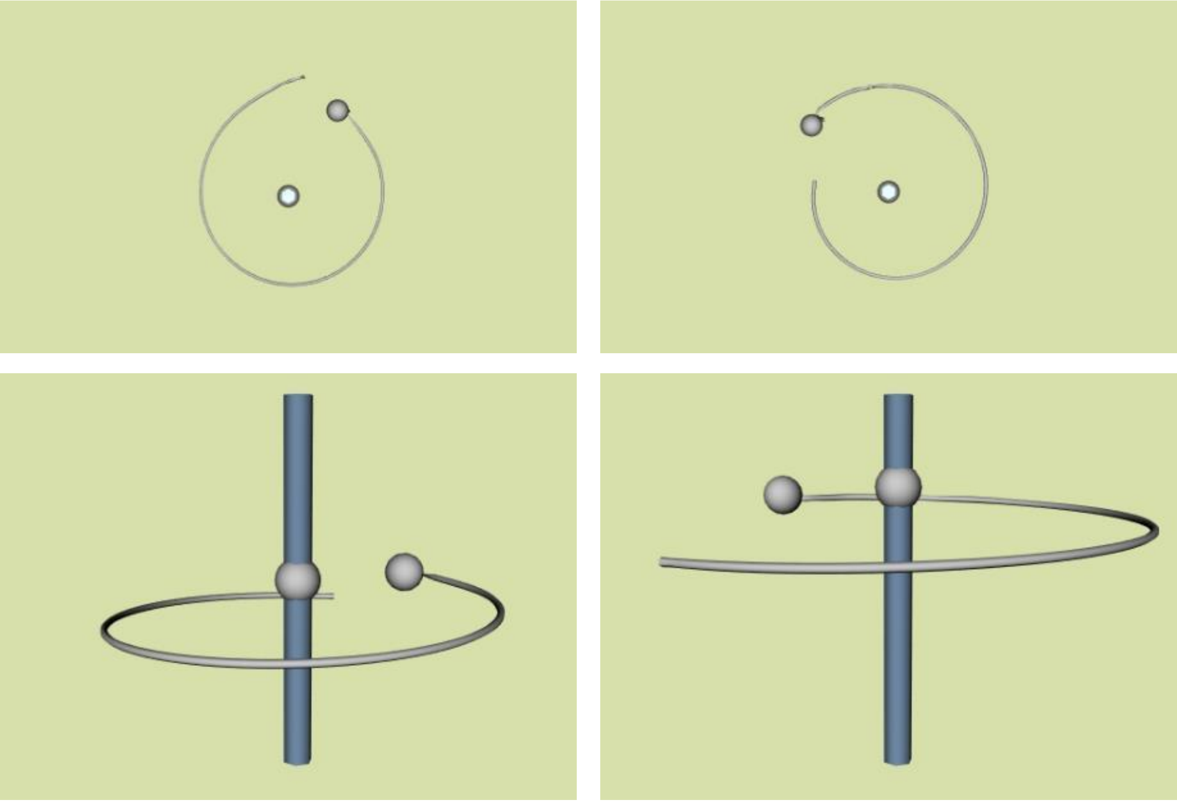
   d. Run the dForce simulation with default density (180). You will find the rope misbehaves, sticking to the center pole and wrapping up and over itself. The images below show frame 14 and 28 from both a top and side view. You can go here to see an animated version of these simulations.



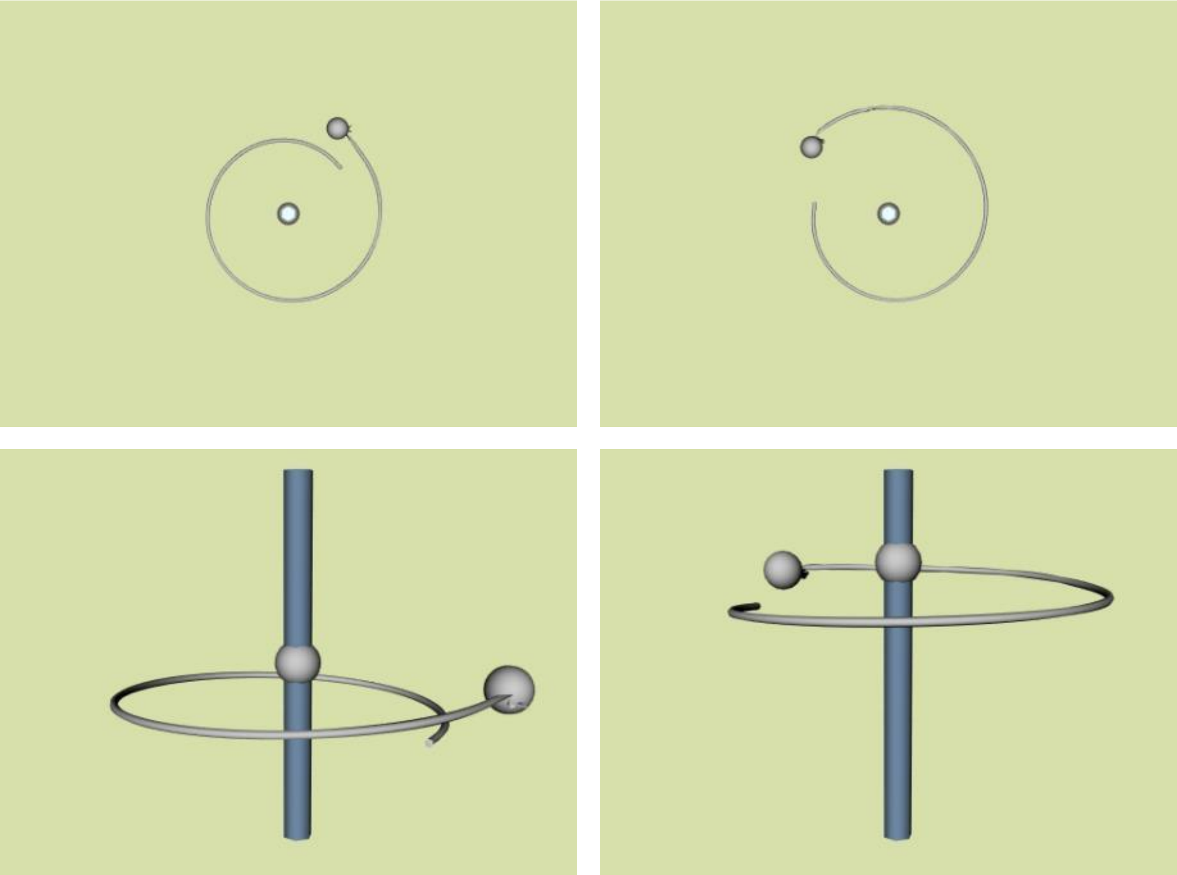
e. Density 25. This was the first density where the rope didn’t strike the central pole. The rope swirls around the pole but has an elongated shape.



f. Density 5. The rope mostly follows a circular path.



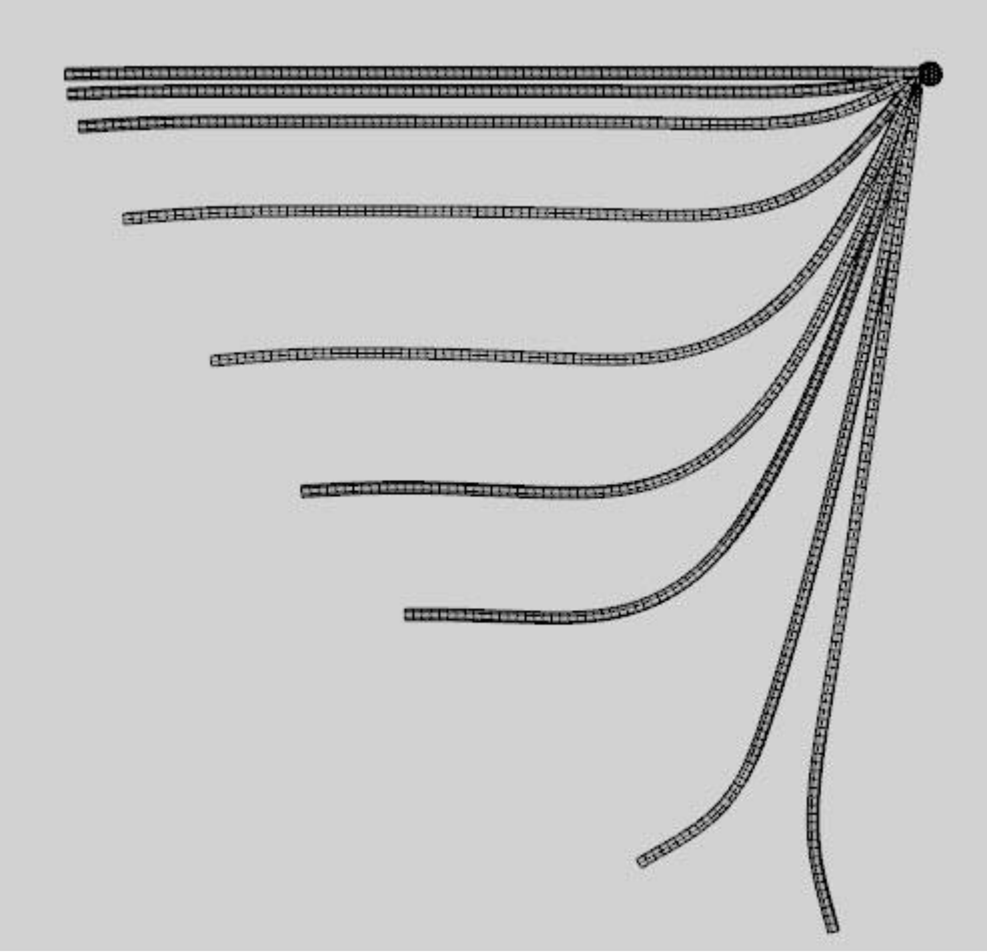
g. Density 1. The rope follows along a path that’s even tighter than a circle.



h. A density value between 1 and 5 seems to be best. These are much lower values than I would normally expect to use.

i. I set up another scene with just one sphere and the rope embedded in it. I duplicated the rope multiple times and changed the density for each one. Below is an image showing the result of single frame simulations using default gravity and default surface settings, other than Density, which runs from 2 for the top rope (stayed horizontal), to 5, 10, 25, 50, 75, 100, 150, and 180 (default) for the bottom rope.

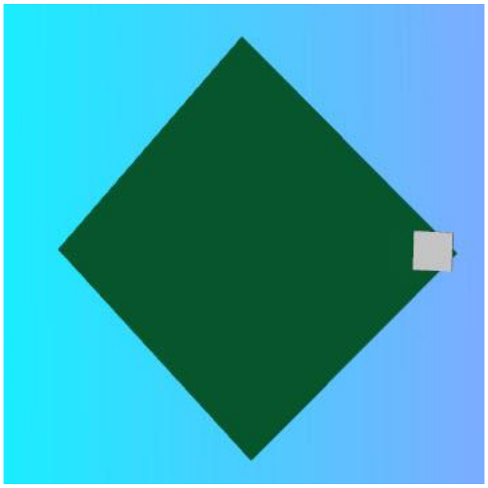
As you can see, the lowest density value of 2 makes the rope fully rigid. Steadily increasing density values cause the rope to sag more near the point of attachment. I've noticed this sagging pattern before, where instead of a smooth drape from end to end, you get a lot of draping near the attachment point and a mostly horizontal section away from it. Since these were all done for the same time duration, it's likely the horizontal section would fall further with added time. This also shows that dForce simulation appears to be working from one end to the other, rather than everywhere at once.



53. **Effect of Density on a Rotating Cloth.** Before I did the rotating rope discussed in section 52, I had started with a rotating cloth but had difficulty with it collapsing into a thin streamer. After discovering that very low density values helped stiffen the rope, I tried them on the cloth and got better results.

a. Create a 2 foot diameter plane with 50 divisions. Rotate it so it appears like a diamond. Add a dForce dynamic modifier to it.

b. Create a 3 inch cube with 20 divisions. Position it at one corner of the plane. Change the X-scale to be slightly wider than the plane.

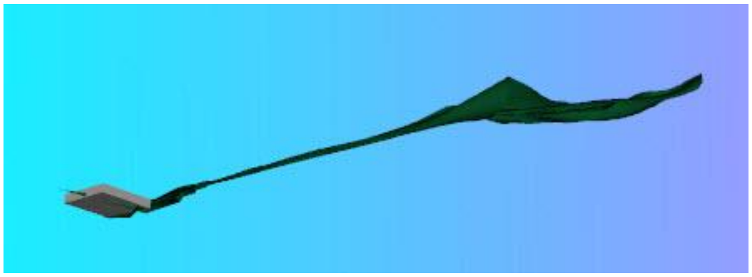


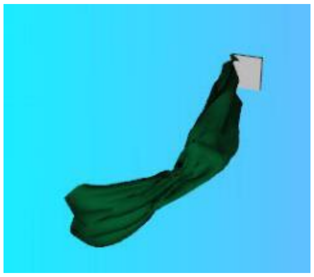
c. Add a figure with a dancing pose and fingers pinching together. Parent the cube to the thumbs of the figure. Do not parent the plane to the cube.

d. Set up an animation timeline with the figure doing a 360 degree rotation around the Y-axis. I also moved the arm up at the 180 degree mark and back down at 360.

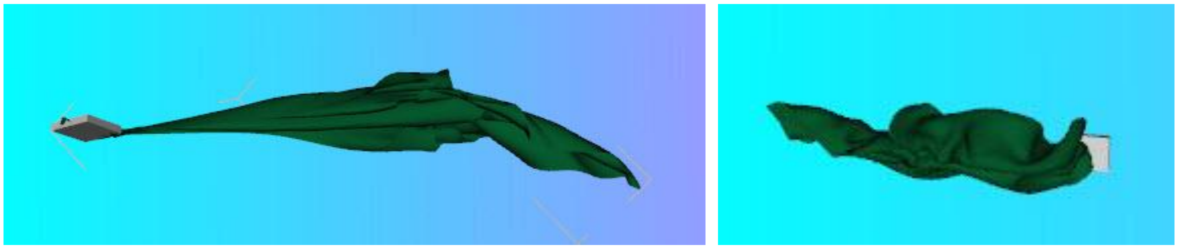
One thing that affects dForce animations is the speed of movement. If instead of the 360 degrees I used with the figure I used 1400 degrees, as I used with the rope in section 52, the cloth plane stretched and exploded no matter the density settings.

e. Run a dForce simulation at default density (180). The cloth stretches into a thin band (images show frame 16 and 32 of 32).

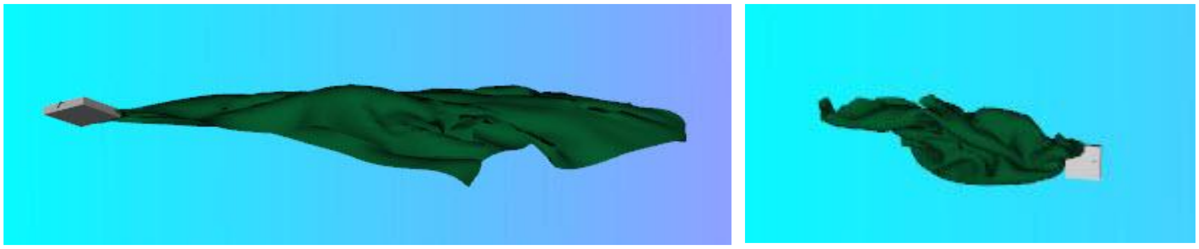




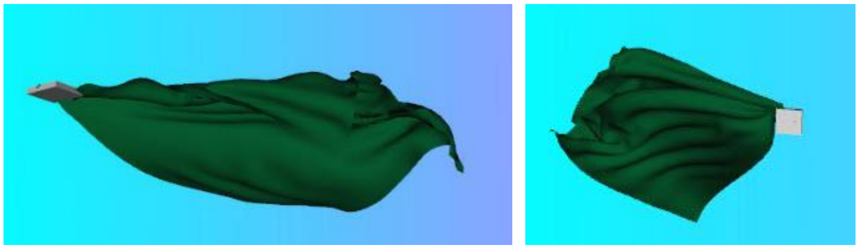
f. Density 50:



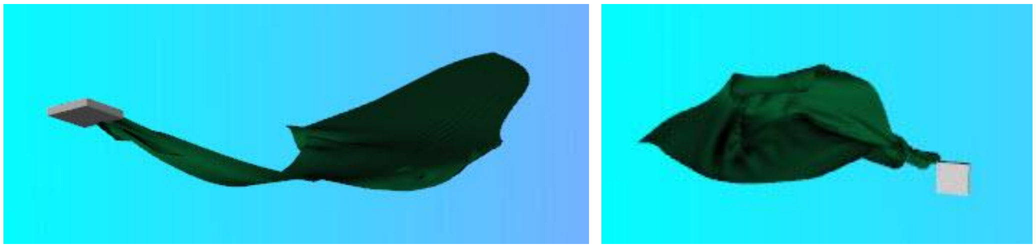
g. Density 25:



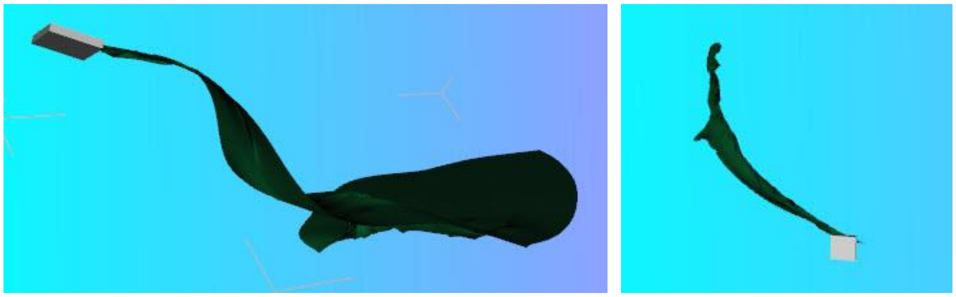
h. Density 10:



i. Density 2:



j. Density 1:



k. As you can see, lower density values result in a cloth that stays more spread out, but you can go too far as shown with density value of 1. Best results are between 2 and 10. One place where this could be applied is to create a flowing cape.

An animated GIF is available online that shows the flowing behavior at a density of 2.

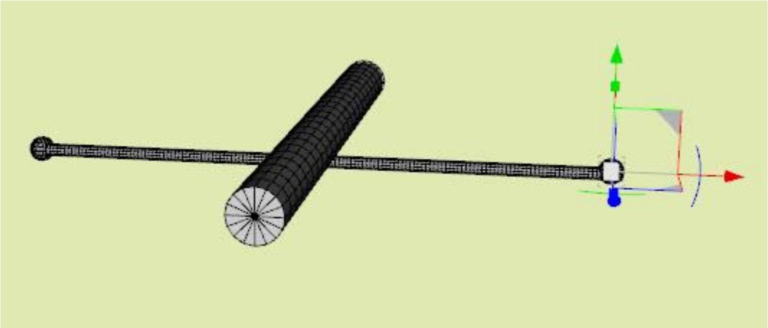
54. **Tie a String.** I wondered if one could use helper objects to tie a string in dForce. Turns out you can as I show below! This is an experiment that might not have much use but it was fun to work out.

   a. Create a primitive cylinder 12 inches long, 0.25 inches in diameter, X-positive, with 150 segments and 12 sides. I used 150 segments so there would be enough flex in the string and enough vertices so they would grab and not pass through. Add a dynamic modifier.

   b. Create a second primitive cylinder 12 inches long, 1 inch in diameter, Z-positive, with 20 segments and 16 sides. Move this cylinder so it is just above and not touching the first cylinder.

   c. Create two primitive spheres, 0.5 inch in diameter, X-positive, with 16 segments and 16 sides. Move and embed one sphere on either end of the long cylinder. Do not parent the cylinder to the spheres.

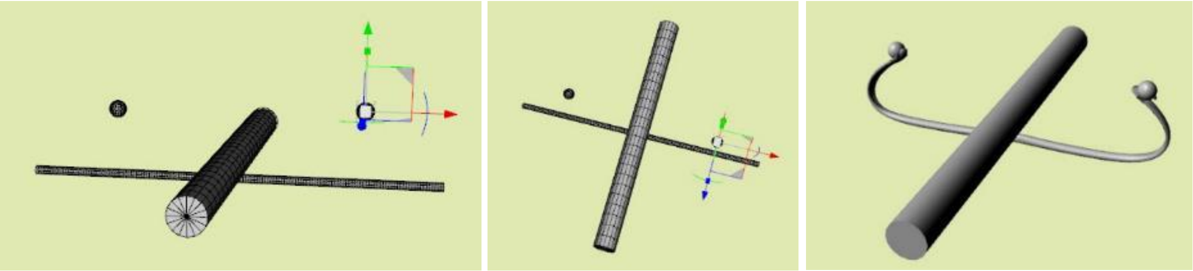
   d. Your scene should look something like this:



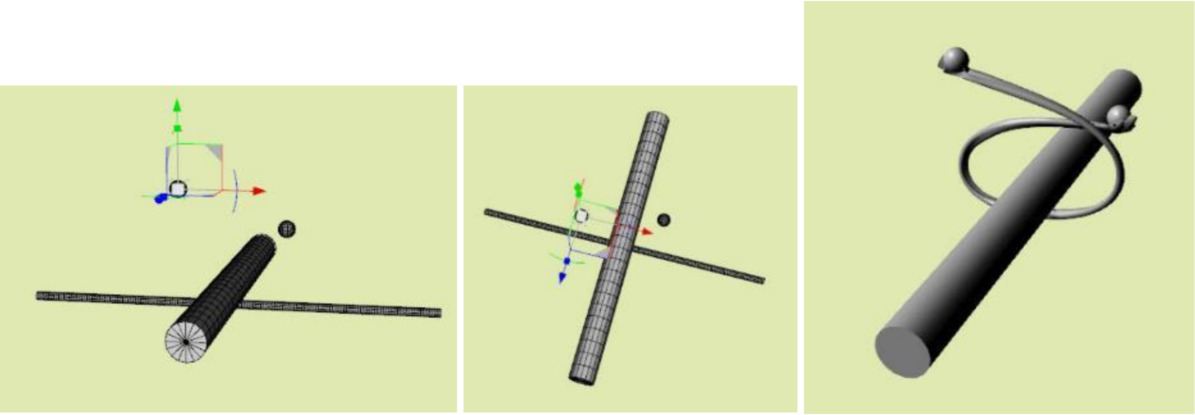
   e. Go to the Surfaces pane and set density to 2. This is important, as otherwise the string has too much flex and will catch and distort.

   f. Set up an animation timeline with 32 frames. Just like tying a knot, the two spheres will cross past each other, than one will wrap around, dip down and back through a loop formed by the crossed string. The following series of images show the setup from the front and then the top view, followed by the simulation of the string.

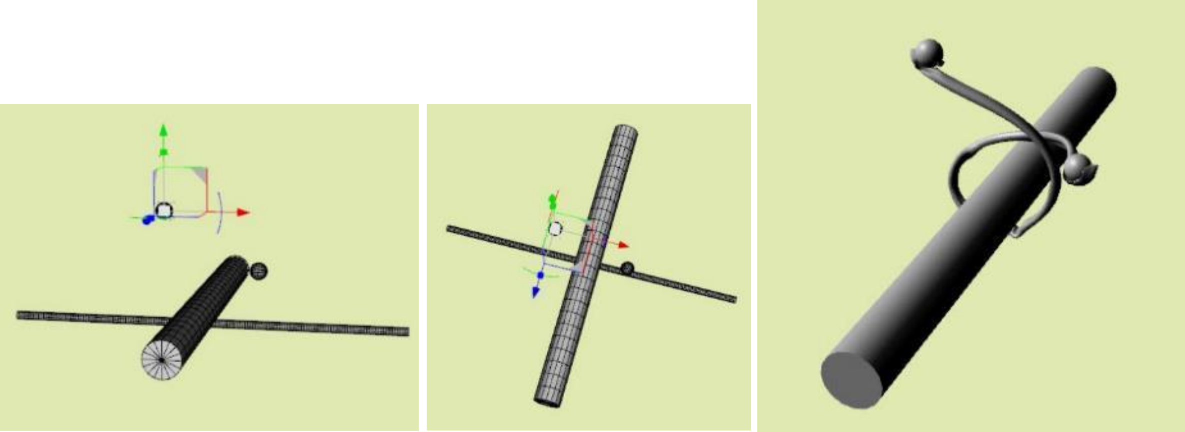
   Frame 4:



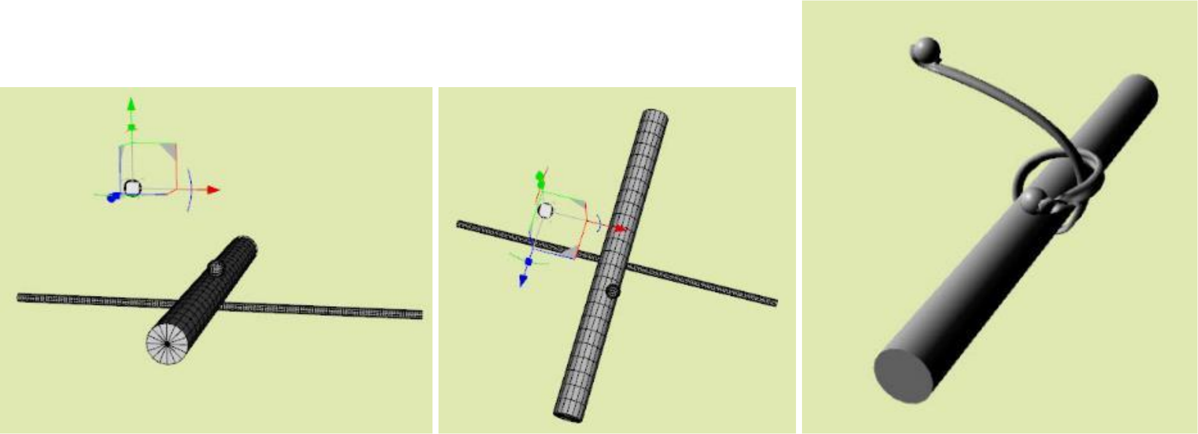
   Frame 8:



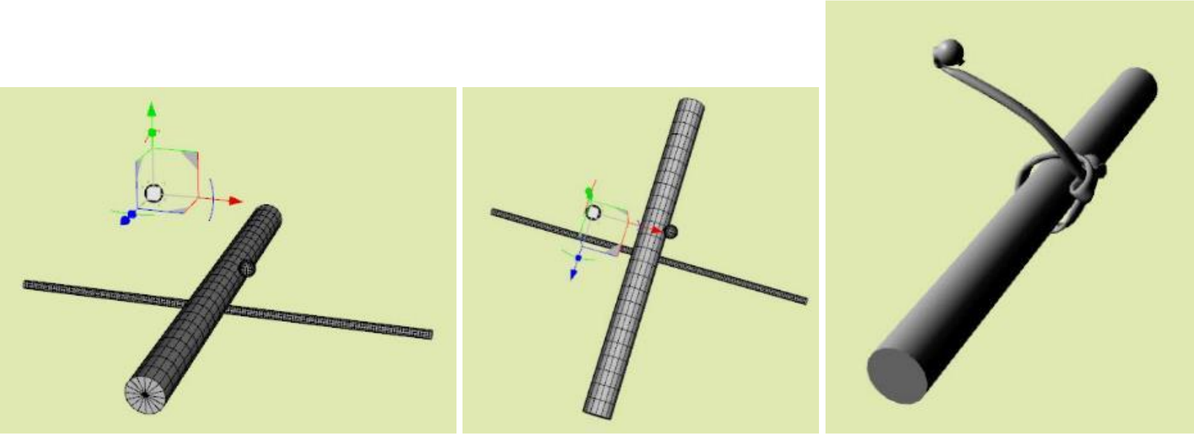
Frame 12:



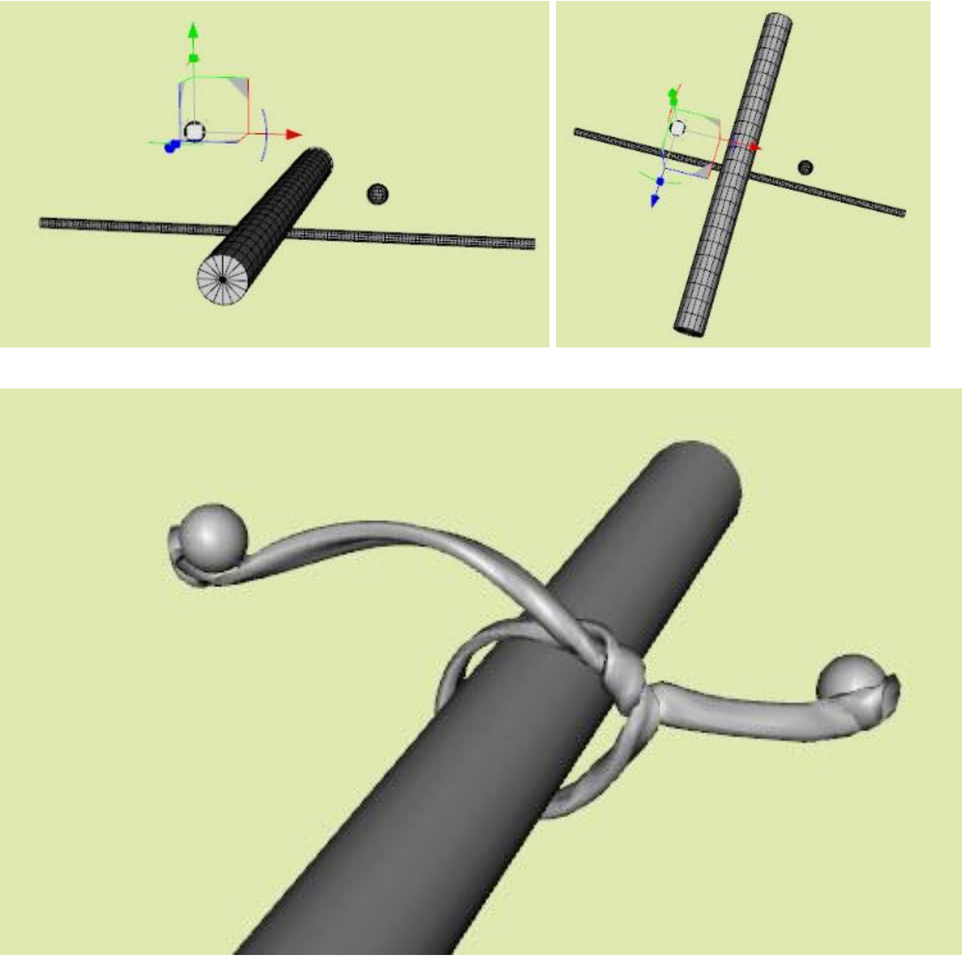
Frame 16:



Frame 20:



Frame 24:



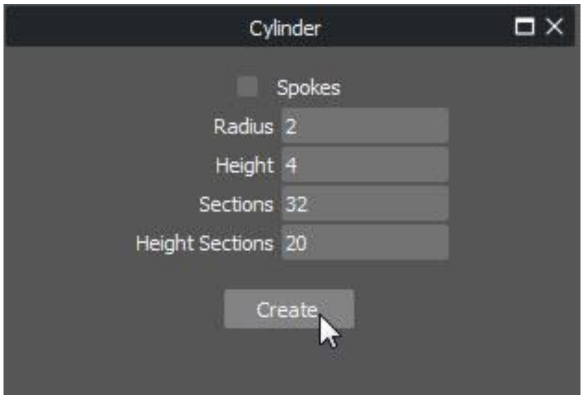
g. It’s probably hard to envision the movement, so there is a GIF available so you can see it in motion.

55. **Using a Dynamic Surface Add-on.** There are three dForce modifiers: dynamic, static, and dynamic add-on. The last one is not explained anywhere and it was only through the postings by Mada and barbult on the Daz forums that I learned how to use it - a big thanks to both!

    For a detailed list of steps using Blender and a wardrobe item (shirt in her case), see **barbult’s post**. Since clothing has more complex geometry that may hinder understanding how the add-on works, this tutorial section uses a simple cylinder made in Silo and imported into Daz Studio (other modelers should have similar tools although they may use different names).

    Silo Steps

    a. Choose Create/Cylinder/Opt from main menu of Silo. Set parameters as shown. Left-click on create.



Cylinder

□ Spokes

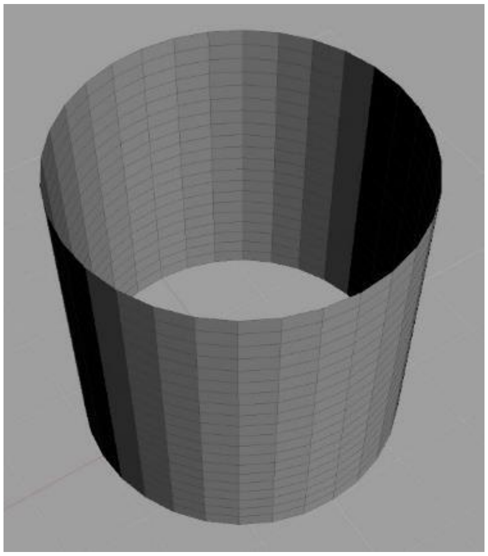
Radius 2

Height 4

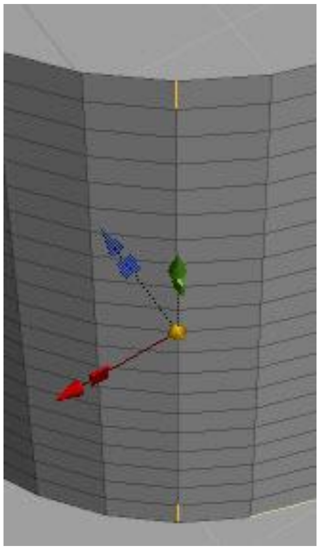
Sections 32

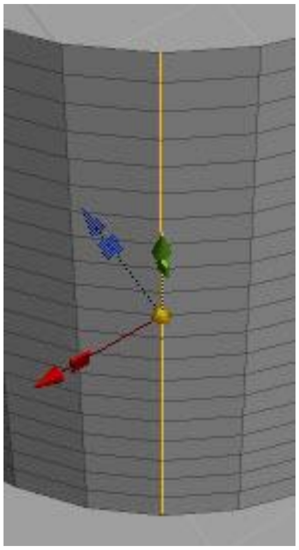
Height Sections 20

b. Using the face selection mode, click on the end faces and delete both.



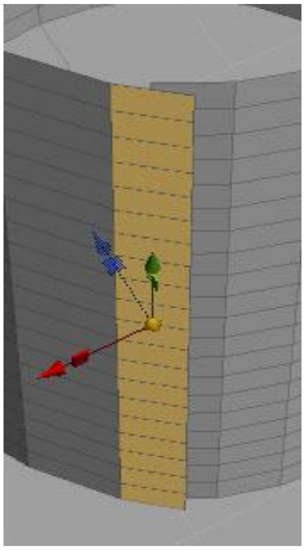
c. Use the edge selection mode and choose a vertical line in the top row of the cylinder and then holding down the Shift key, do the same in the bottom row, directly below the first edge. Select the path (Alt+T).





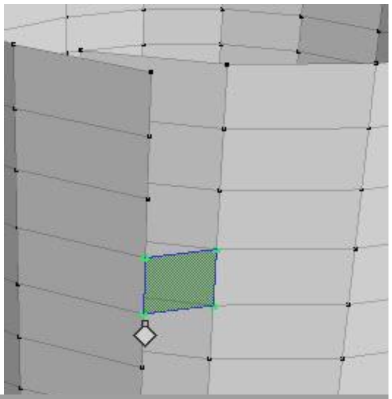
d. From the main menu, choose Modify/Break to split the cylinder down the path.

e. Use the face selection mode and choose the faces to the left of the split. Move them slightly forward and then to the right so they overlap the right faces but with a small gap.

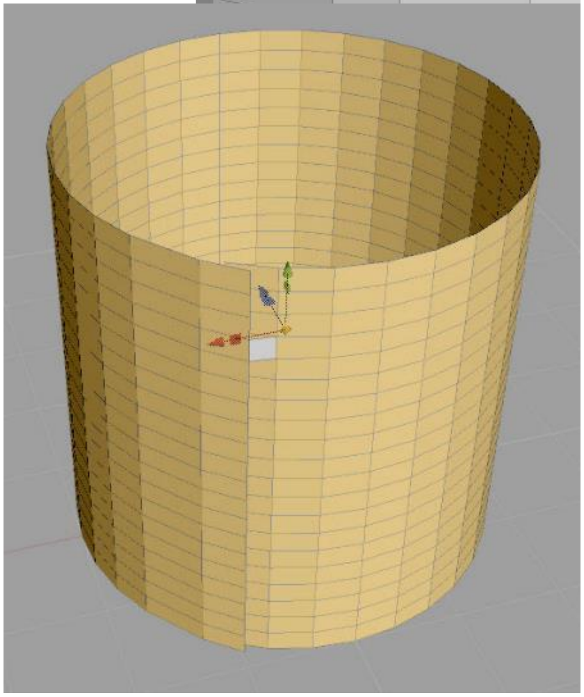


f. Select the cylinder and save it by going to the main menu and choosing File/Save Selected Objects…

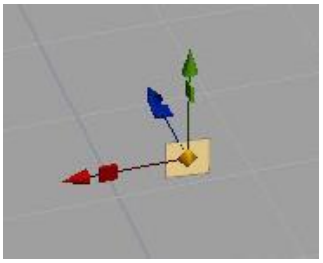
g. Zoom in towards the upper part of the cylinder. Choose the polygon tool (menu Create/Polygon Tool). The mouse icon will change to a diamond. Click on a vertex (the 4th one down) on the left edge, then click on a vertex on the right side, then one in the 5th row on the right side, and then the fifth one down on the left edge (see green area in image). Hit Enter to create the polygon and Esc to exit the tool. The newly created polygon is now a part of the cylinder object.



h. Use the face selection mode to select the just created polygon. Then go to the menu and choose Selection/invert Selection to deselect the polygon and select the cylinder.

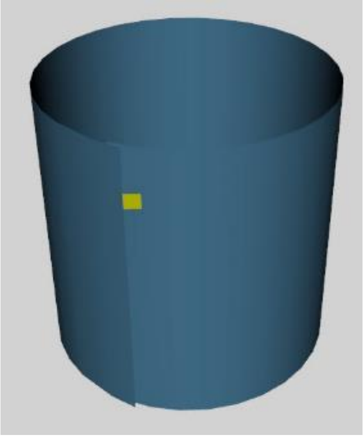


i. Hit the delete key. You will be left with only the lone polygon. Save it by going to the main menu and choosing File/Save Selected Objects…
(If you want to create another polygon, type Ctrl+z to undo and restore the cylinder).

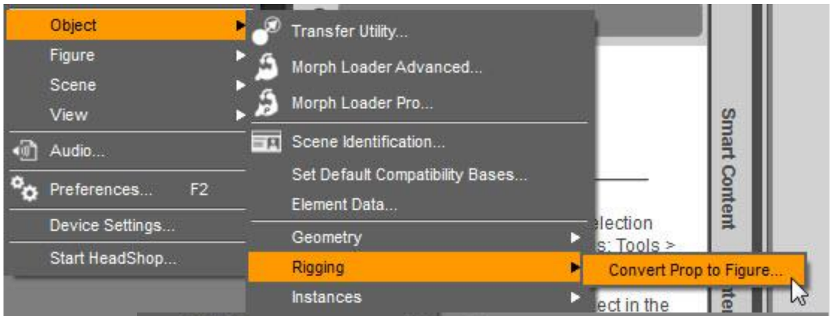


<u>Daz Studio Steps</u>

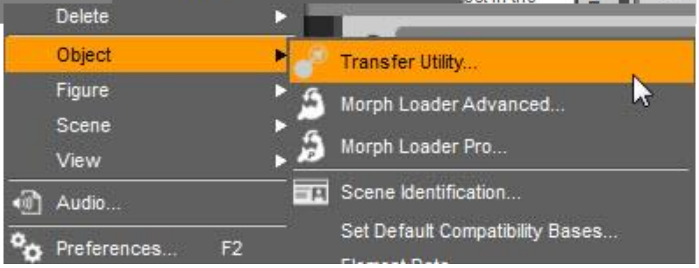
j. Import the cylinder into Daz Studio (use From: Silo to import it with the right dimensions). Repeat for the polygon.



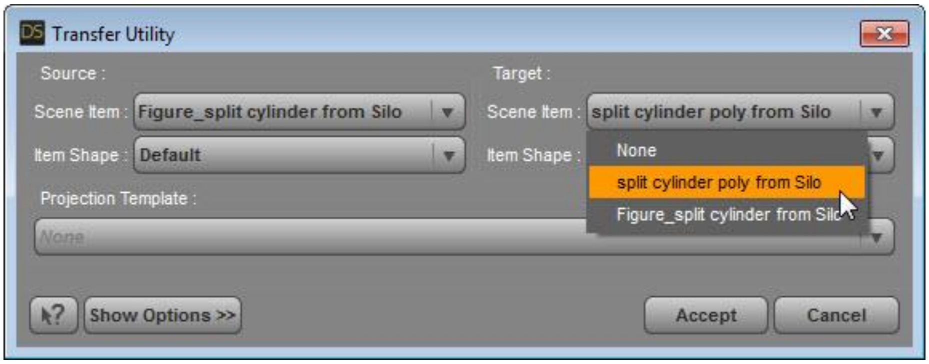
k. With the cylinder selected, go to the main menu and choose Edit/Object/Rigging/Convert Prop to Figure.



l. Go to the main menu and choose Edit/Object/Transfer Utility…



m. In the Source drop down box, choose the figure we just created. In the Target drop down box, choose the polygon. This will make the polygon part of the figure.


Transfer Utility

Source :

Scene Item : Figure_split cylinder from Silo

Item Shape : Default

Projection Template :

None

Target :

Scene Item : split cylinder poly from Silo

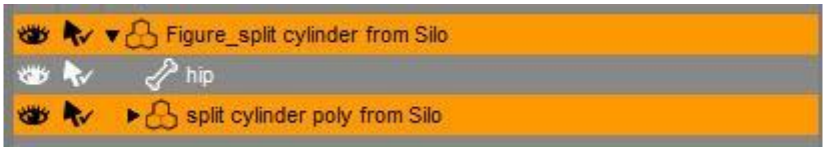
Item Shape :

None

split cylinder poly from Silo

Figure_split cylinder from Silo

The scene pane will show the following structure:

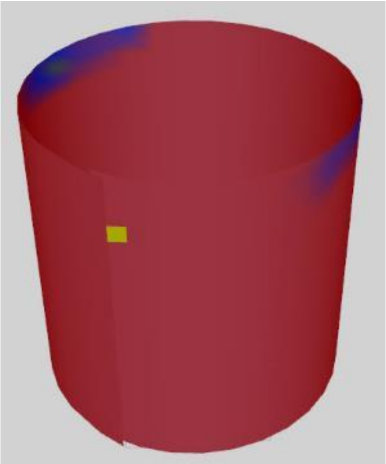

Figure_split cylinder from Silo

hip

split cylinder poly from Silo

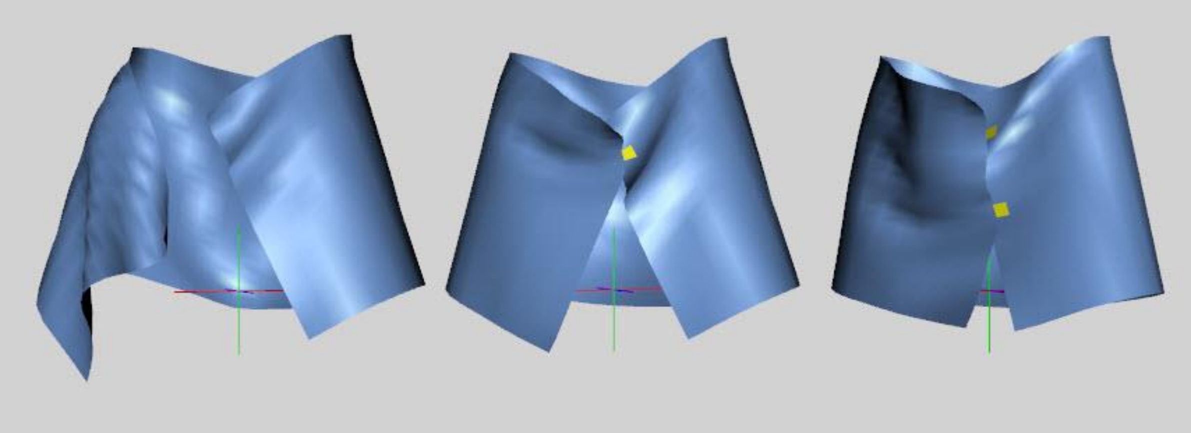
n. Select the cylinder and add a dForce dynamic modifier. Select the polygon and add a dForce dynamic add-on modifier.

o. From the main menu, choose Create/New dForce Modifier Weight Node. Go to the Node Weight Map Brush in the Tool Settings pane. Add the influence weight map. Paint a bit of blue on the upper left and right edges as shown in the image (remember to hold down the Alt key to remove the full-strength red). This will be enough to hold the cylinder in space during simulation.



p. Run the simulation. You'll see the split open up but it will be held in place by the add-on polygon. After the simulation you can hide the polygon or change its opacity to 0.

   In the image below, there are 3 different setups. The left cylinder shows what happens when there is no add-on, the middle one shows the result with one polygon, and the right one shows it with two polygons. You can see that if you were making a shirt, you could add a polygon at every button so the shirt would move in-between in a more realistic fashion.



q. You may find you can skip steps k to m (making the prop a figure and using the transfer utility). For the figure above, just making the polygon a child of the cylinder was sufficient for it to work, but other times it fails. Not using the transfer utility also means the polygons may appear in a different position before the simulation than they will appear during the simulation, which can be confusing and lead to unexpected results. With what I know now, I'd recommend sticking to all the steps.

r. For those without a modeler, I have figured out how to get the coordinates of vertices from within Daz Studio. An upcoming section will describe how to use that approach.

56. **Finding Vertex Coordinates within Daz Studio.** Many people have access to a 3D modeling application. If you do, the easiest way to create a polygon to use as an add-on is to create it in the modeler (like I did with Silo in section 55). If you don't have access to a modeler, all is not loss as you can hand-code a polygon.

   a. Go here and download and save the Geometry_Info script to your Daz Library Script folder.

   b. Open the Script IDE pane from the main menu by selecting Window/Panes (Tabs)/Script IDE.

   c. From the Script IDE menu, choose File/Open Script… and click on Geometry_Info.dsa

   d. We are going to modify this script, so choose File/Save Script As… and give it the name Geometry_Info_Vertex_Coordinates.dsa.

   e. First delete lines 140 through 170.

f. Next inset at line 140 the following:

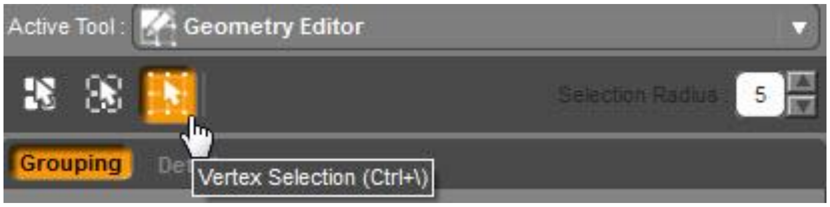
```
var oVertex;
// Get the number of vertices
var nVertices = oMesh.getNumVertices();
// Iterate over the vertices
for( var i = 0; i < nVertices; i += 1 ){
    // Get the 'current' vertex
    oVertex = oMesh.getVertex( i );
    print( String("Vertex #%1").arg(i) );
    print( String("\t %1 %2 %3")
        .arg( oVertex.x )
        .arg( oVertex.y )
        .arg( oVertex.z )
    );
```

g. Choose File/Save Script. You only have to do these steps one time.

h. To test the script and make a polygon, create a primitive cylinder in Daz Studio.

i. Using the Geometry Editor, select the top faces. Right-click in the viewport and hide them (Geometry Visibility), then delete them (Geometry Editing).

j. Change to the Geometry Editor vertex selection tool (the 3rd icon in the top row of the pane).

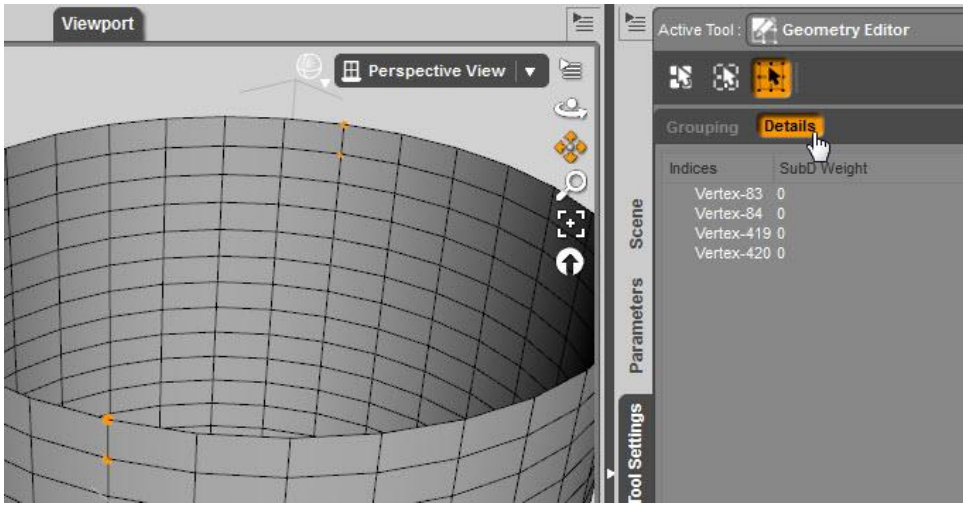


Active Tool : Geometry Editor

Selection Radius 5

k. Click on a vertex in the top row of the cylinder. Holding down the Ctrl key, select the vertex directly below. While still holding down Ctrl, select a vertex in the second row down on the opposite side of the cylinder, then choose the vertex directly above. You will have 4 vertices selected.

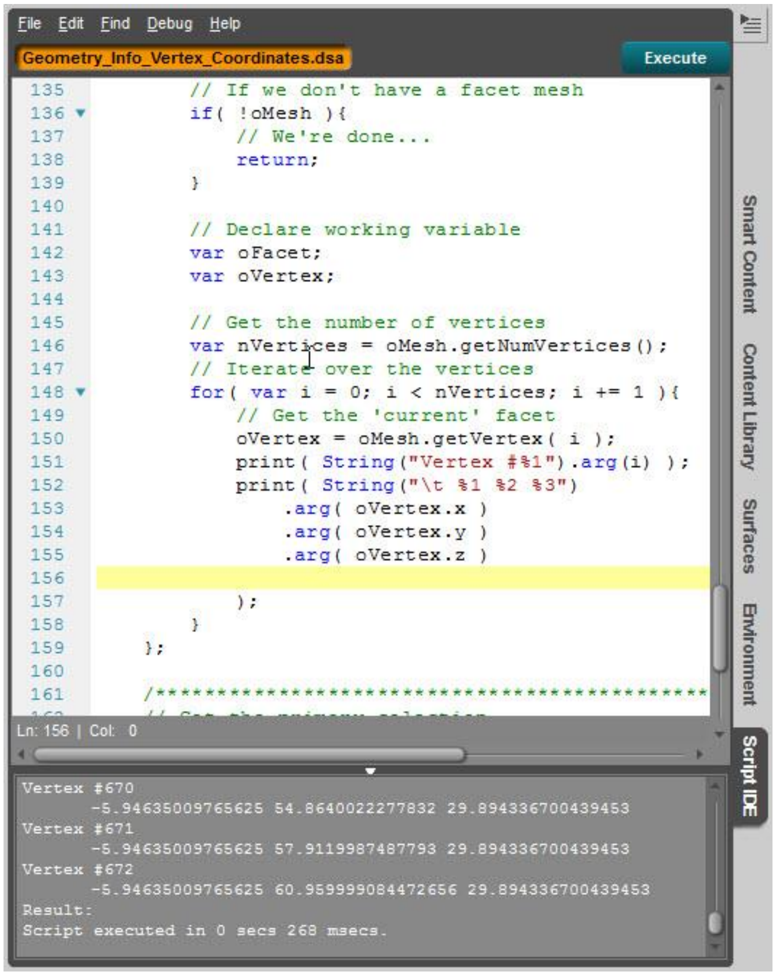
l. To see which vertices you've chosen, click on the Details button in the tab right below the icons in the geometry editor. Make note of the numbers. Daz sorts the vertices in numerical order which is not necessarily the order you selected them in.



Active Tool : Geometry Editor

Indices	SubD Weight
Vertex-83	0
Vertex-84	0
Vertex-419	0
Vertex-420	0

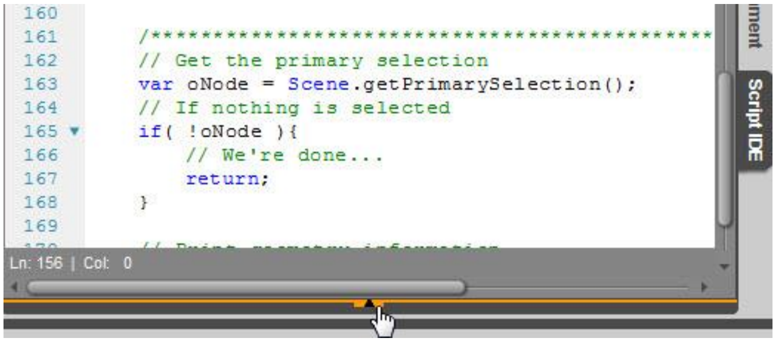
m. Now run the script we modified before by clicking the Execute button. The results will be in the gray output box at the bottom of the IDE:



```
        // If we don't have a facet mesh
        if( !oMesh ){
            // We're done...
            return;
        }

        // Declare working variable
        var oFacet;
        var oVertex;

        // Get the number of vertices
        var nVertices = oMesh.getNumVertices();
        // Iterate over the vertices
        for( var i = 0; i < nVertices; i += 1 ){
            // Get the 'current' facet
            oVertex = oMesh.getVertex( i );
            print( String("Vertex #%1").arg(i) );
            print( String("\t %1 %2 %3")
                .arg( oVertex.x )
                .arg( oVertex.y )
                .arg( oVertex.z )

            );
        }
    };

    /*************************************************
```

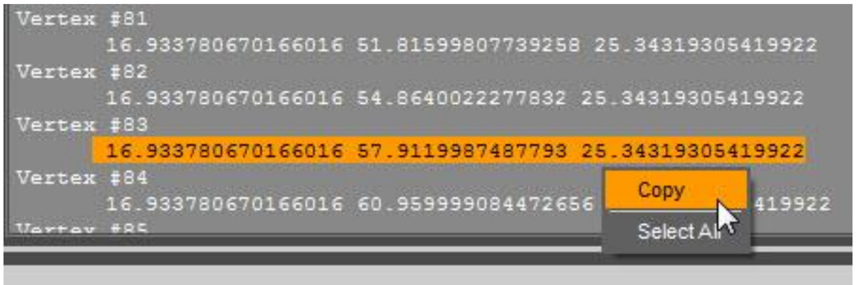
```
Vertex #670
        -5.94635009765625 54.8640022277832 29.894336700439453
Vertex #671
        -5.94635009765625 57.9119987487793 29.894336700439453
Vertex #672
        -5.94635009765625 60.959999084472656 29.894336700439453
Result:
Script executed in 0 secs 268 msecs.
```

(If this box is not showing, click on the small triangle at the bottom of the pane.)



```
    /*************************************************
    // Get the primary selection
    var oNode = Scene.getPrimarySelection();
    // If nothing is selected
    if( !oNode ){
        // We're done...
        return;
    }

```

n. Scroll through the result box and find the line that matches one of your vertex numbers. Highlight the row of numbers, right click and choose copy.



```
Vertex #81
        16.93378067016016 51.81599807739258 25.34319305419922
Vertex #82
        16.93378067016016 54.8640022277832 25.34319305419922
Vertex #83
        16.93378067016016 57.9119987487793 25.34319305419922
Vertex #84
        16.93378067016016 60.959999084472656             419922
```

o. In a text editor (like Notepad for Windows), type the letter v and then paste the numbers you copied.

```
v 16.933780670166016 57.9119987487793 25.34319305419922
```

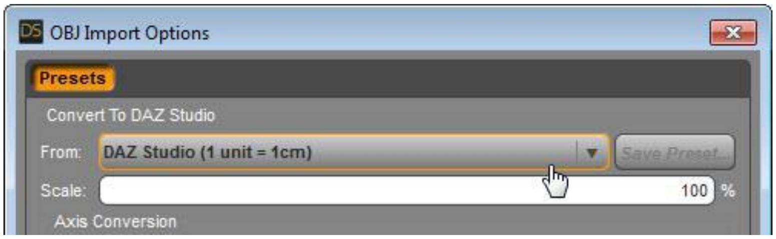
p. Repeat for the other 3 vertices.

q. Next add f 2 1 3 4 to the text file. You should have something that looks like this:

```
v 16.933780670166016 57.9119987487793 25.34319305419922
v 16.933780670166016 60.959999084472656 25.34319305419922
v -16.93378257751465 57.9119987487793 -25.343191146850586
v  -16.93378257751465 60.959999084472656 -25.343191146850586
f 2 1 3 4
```

r. Save the text file with .obj. You've just hand-coded a polygon!

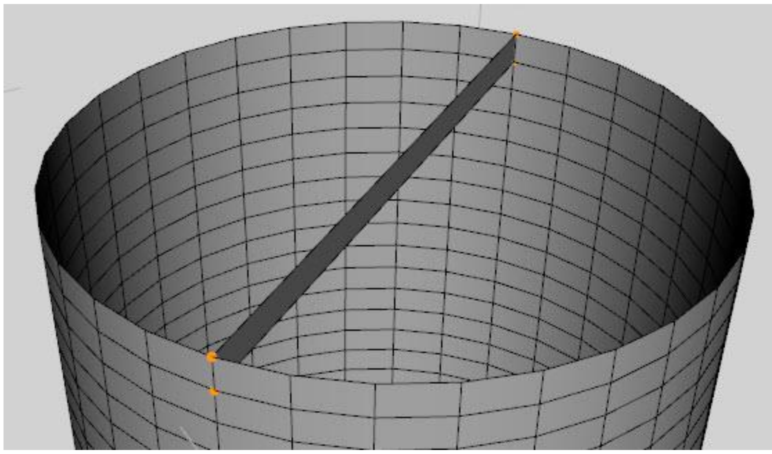
s. Import the polygon into Daz Studio. From the main menu, select Import and select the .obj file you just created. In the dialog box, be sure From: Daz Studio and Scale 100% are chosen.



From: DAZ Studio (1 unit = 1cm)

Scale: 100 %

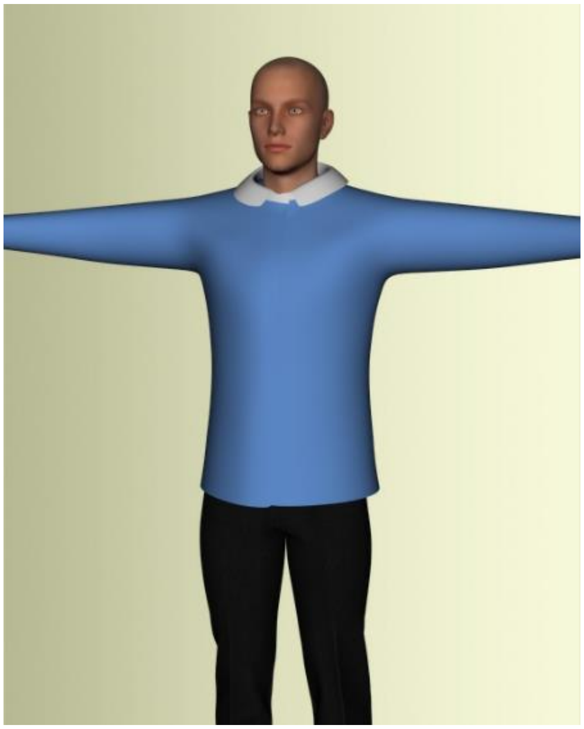
t. You should see the polygon show up in the correct position. If not, you likely did not copy the right vertex information. If the polygon seems to be twisted, swap the first two numbers in the "f" line.



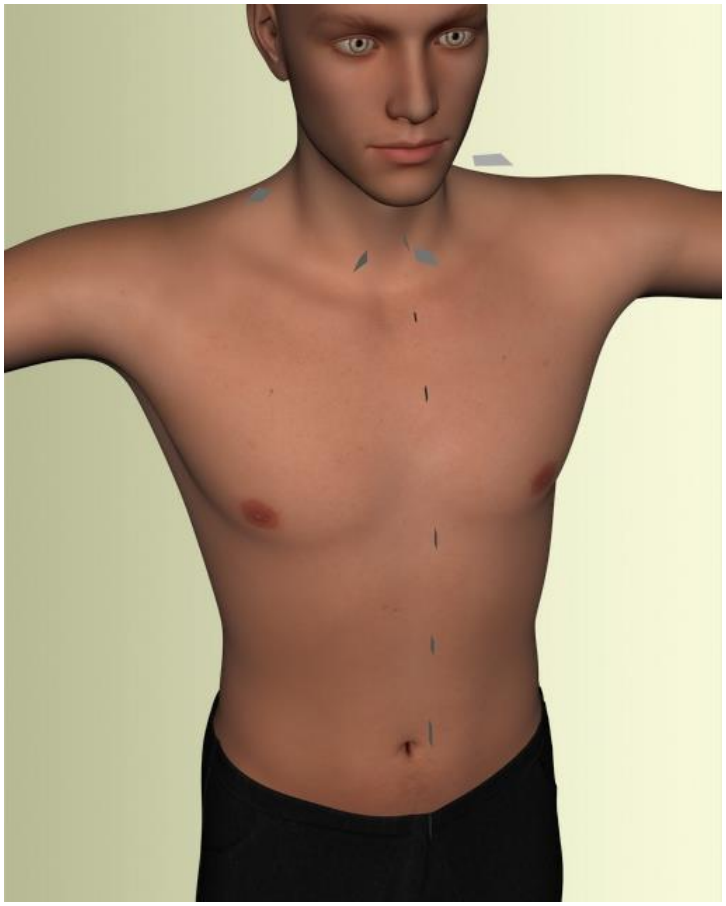
u. You can now continue with the steps j to p in section 55.

57. **Button and Collar Add-ons.** To continue exploring add-ons, I modified a shirt I previously made in Silo. Now I have a much deeper appreciation for those who create clothes for 3D figures. Whew, not easy! I took the shirt and split it down the front, then overlapped the edges and added a collar. For the front gap, I added 7 polygons where buttons would be. For the collar, I added 5 polygons (1 at each collar tip, 1 on each side, and 1 in the back). I then ran frame 0 and animated dForce simulations.

   a. Here is a view of the shirt at frame 0.

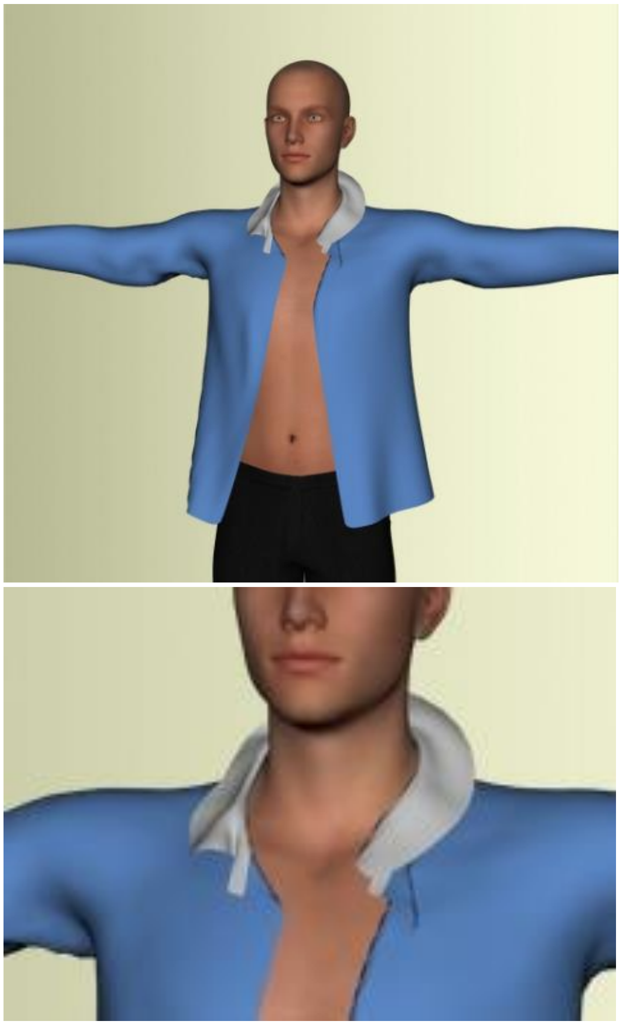


   b. Here is where the polygons are located. They are two different objects, so you can hide either or both.

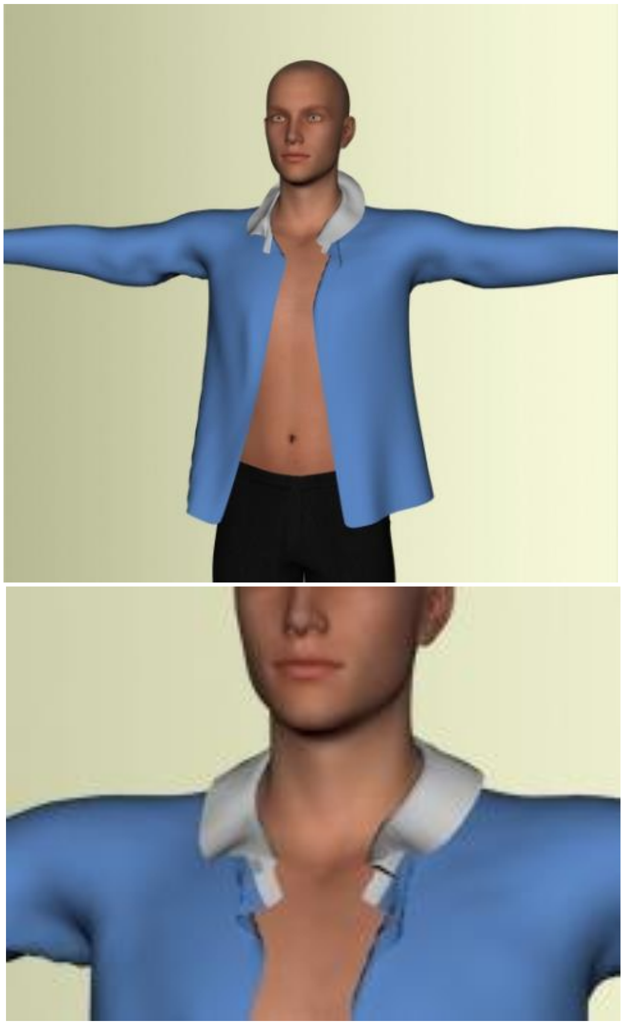


c. Frame 0 Simulations. The shirt drapes open and the collar turns up without add-ons. With the collar add-on, it stays down.

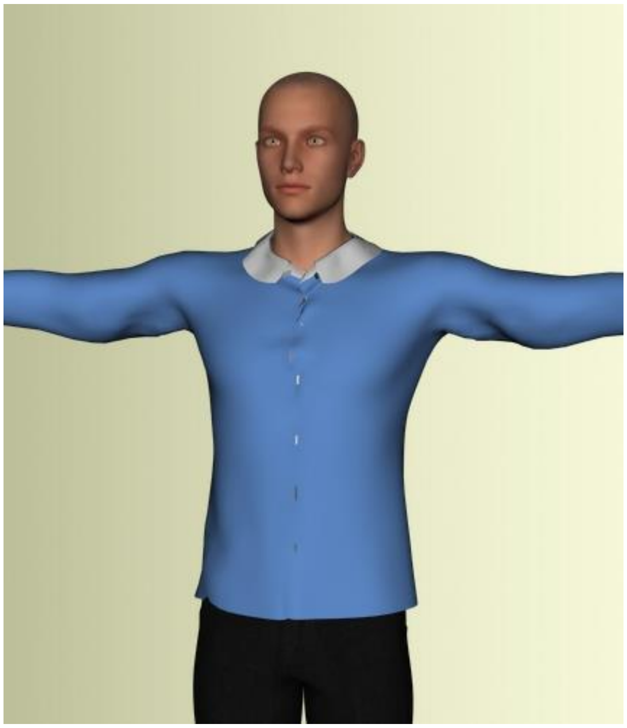
No Add-ons:



With Collar Add-on:

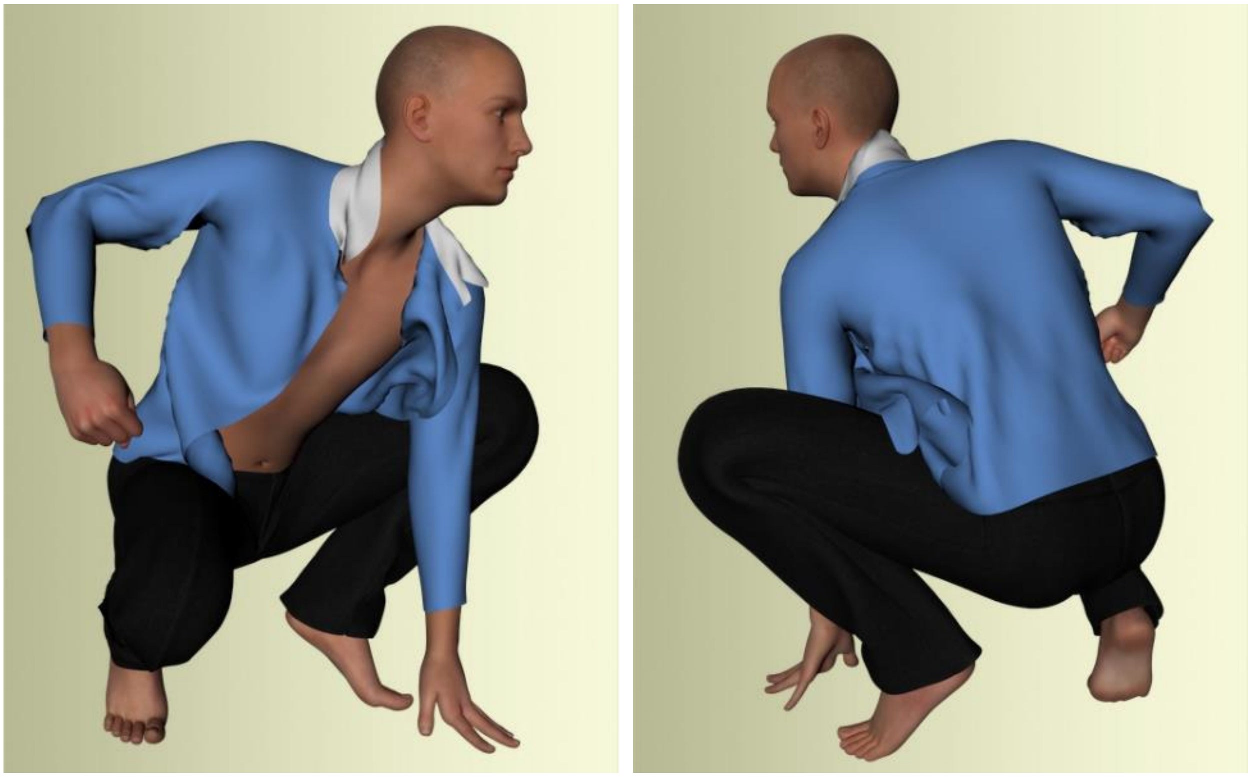


With Collar and Shirt Gap Add-ons:

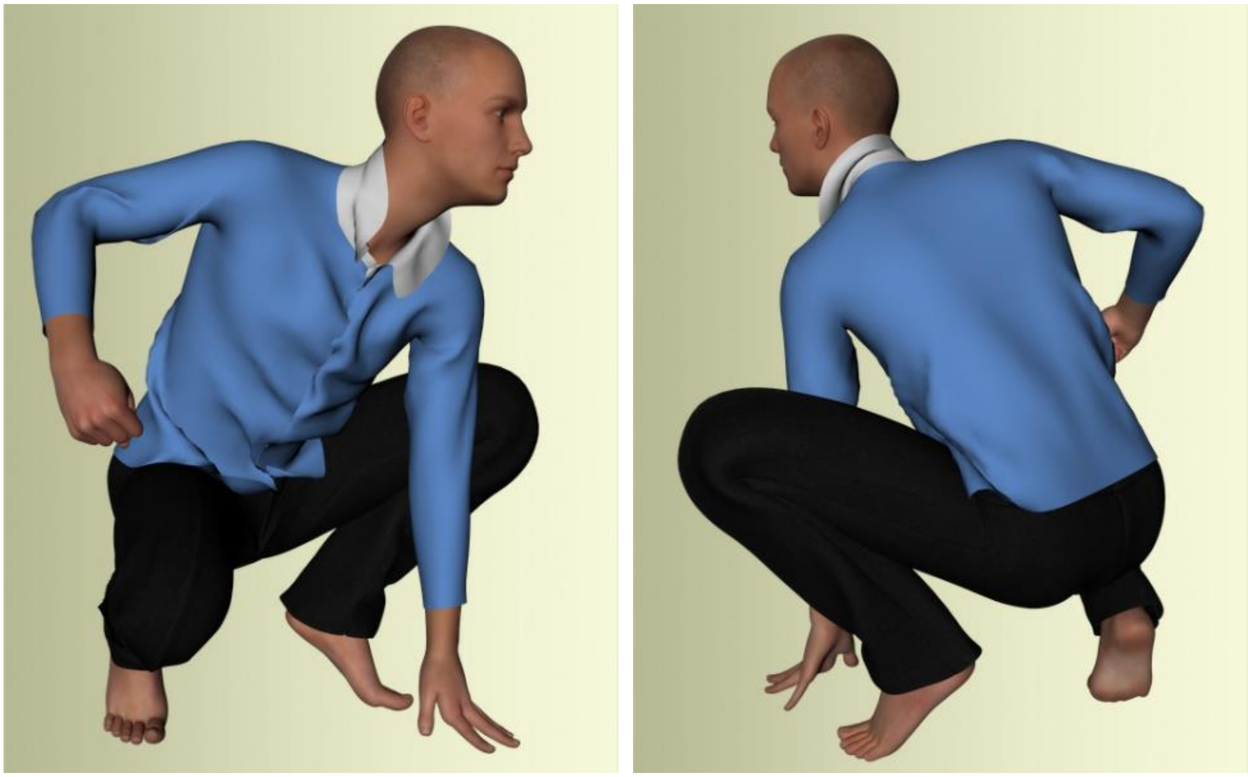


d. Frame 30 Simulations. An extreme pose showing a lot more folding in the shirt.

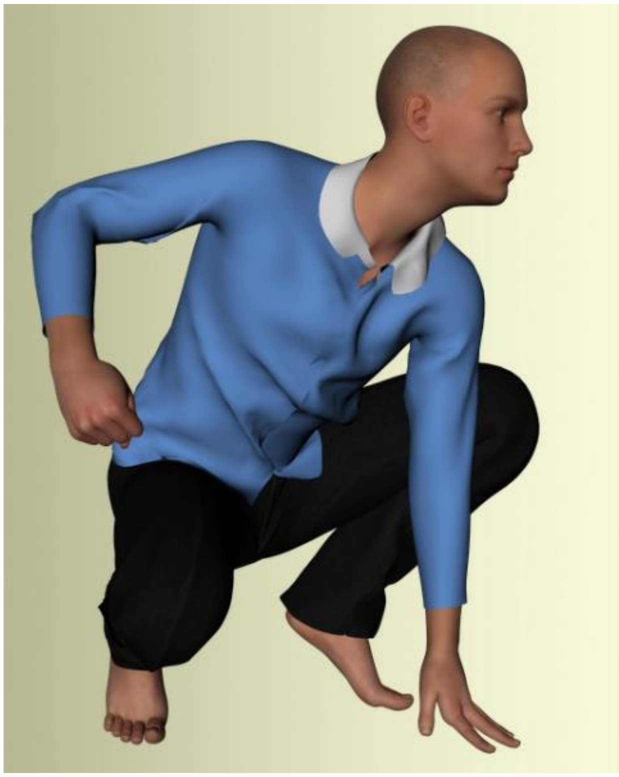
No Add-ons:

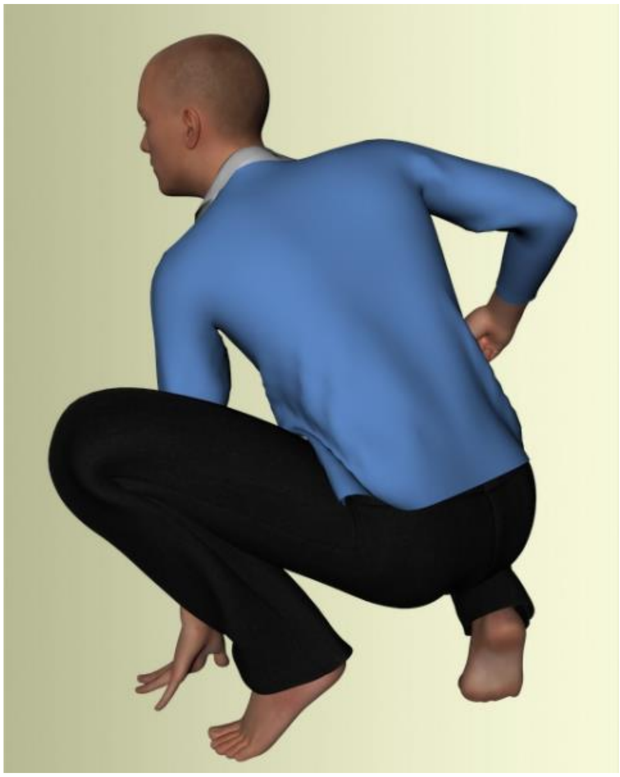


Shirt Gap Add-on:

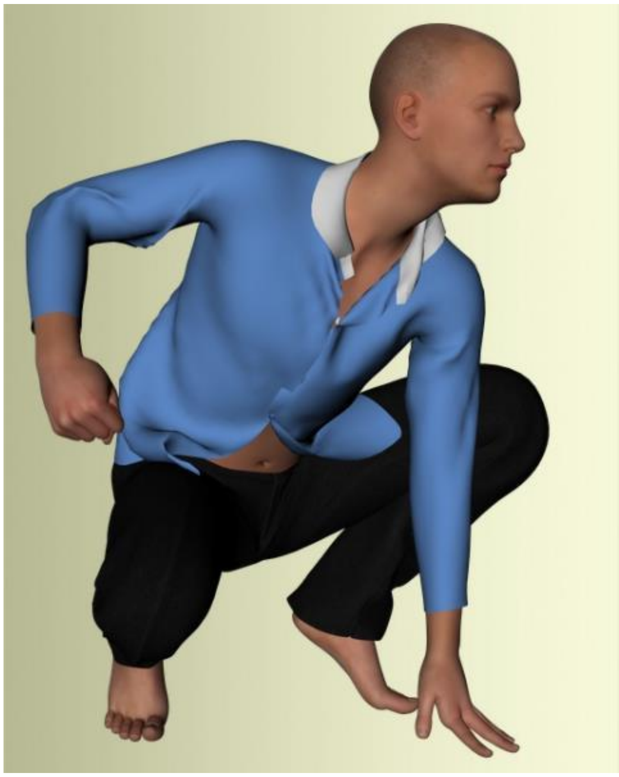


Collar and Shirt Gap Add-ons: With the more extreme pose, the polygon at the collar was not enough to hold the gap together.



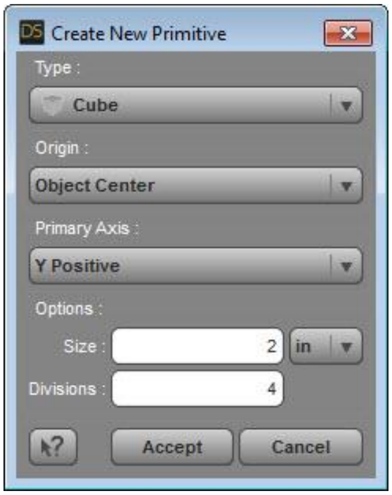


e. Finally I used the Geometry Editor to hide the top 2 and bottom two polygons of the shirt gap add-on to give it a more open look.



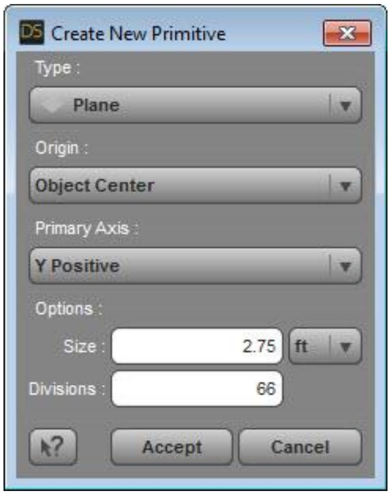
58. **Creating a Seat Belt.** Someone on the Daz Forum mentioned the fact that most 3D car models lack seat belts. Turns out dForce can create a seat belt that will wrap around a sitting passenger. My first attempt used a single long plane with 3 helper cylinders - it worked but was a little hard to setup. A second attempt using two planes was much easier to setup and simulate, so that is the approach I will discuss here.

a. Create a 2 inch primitive cube with 4 divisions. Change it to a color of your choice. This will serve as a reference for trimming larger planes to size.



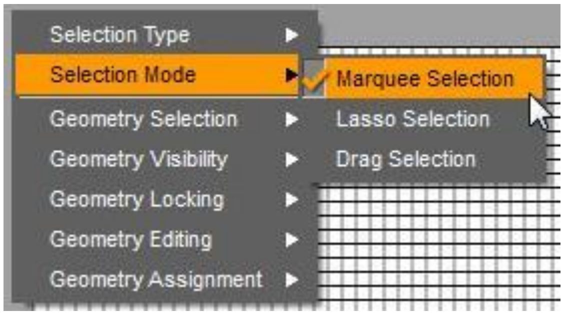
Type : Cube
Origin : Object Center
Primary Axis : Y Positive
Options :
Size : 2 in
Divisions : 4

b. Create a 2.75 foot primitive plane, y-positive, with 66 divisions. This is for the chest seat belt – depending upon the car model, it may need to be a bit longer.

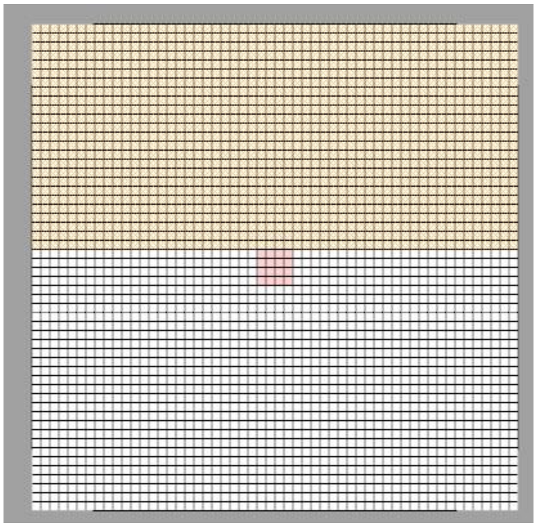


Type : Plane
Origin : Object Center
Primary Axis : Y Positive
Options :
Size : 2.75 ft
Divisions : 66

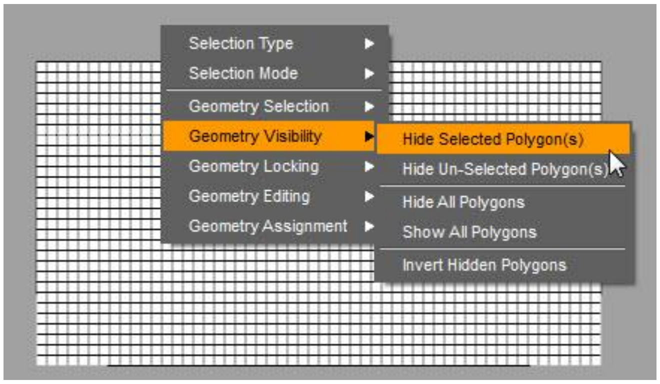
c. Open the Geometry Editor from the Tools Settings pane. Right click in the viewport and set Selection Mode to Marquee Selection.



d. Hold down the left mouse button and drag from the upper left to mid-right. Make sure all the squares above the guide cube are selected (orange).

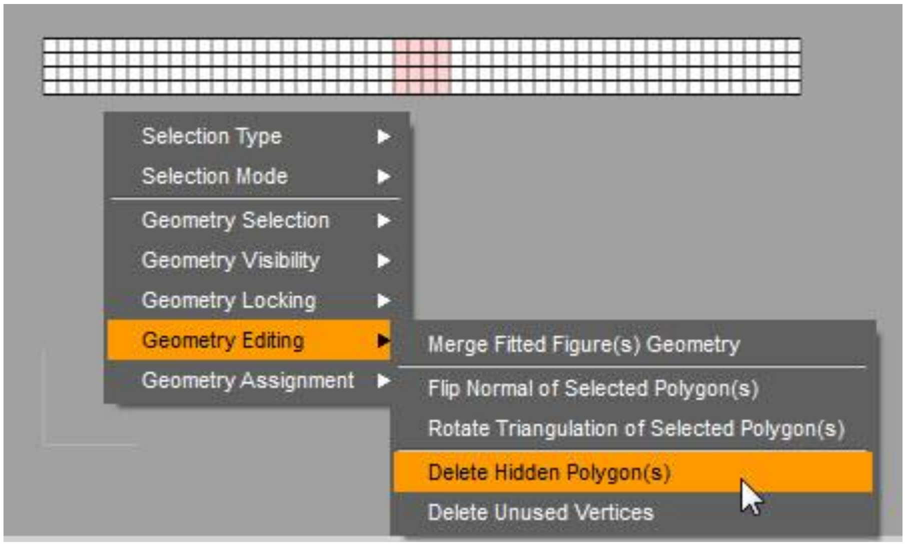


e. Right click in the viewport and select Geometry Visibility/Hide Selected Polygons.



f. Repeat step d and e for the lower half.

g. Right click in the viewport and select Geometry Editing/Delete Hidden Polygon(s). This will permanently eliminate the hidden part of the plane leaving just the thin belt.



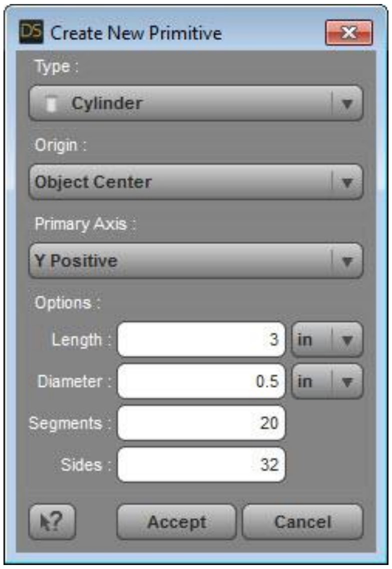
h. You may wonder if you could just scale down the plane, and you can, but the behavior of the belt during simulation leaves a lot to be desired.

i. Now create a 2.25 foot primitive plane, y-positive, with 54 divisions. This is for the lap seat belt – depending upon the car and figure models, it may need to be a longer. Follow steps d to g for the second plane.

j. Add a dForce dynamic modifier to both planes (Edit/Object/Geometry/Add dForce Modifier: Dynamic Surface).

k. Set the Density parameter on the Surfaces Pane to 2 for both belts.

l. Create 3 helper cylinders that are 3 inches long, 0.5 inches in diameter, 20 segments and 32 sides (see image).


Create New Primitive

Type : Cylinder

Origin : Object Center

Primary Axis : Y Positive

Options :

Length : 3 in

Diameter : 0.5 in

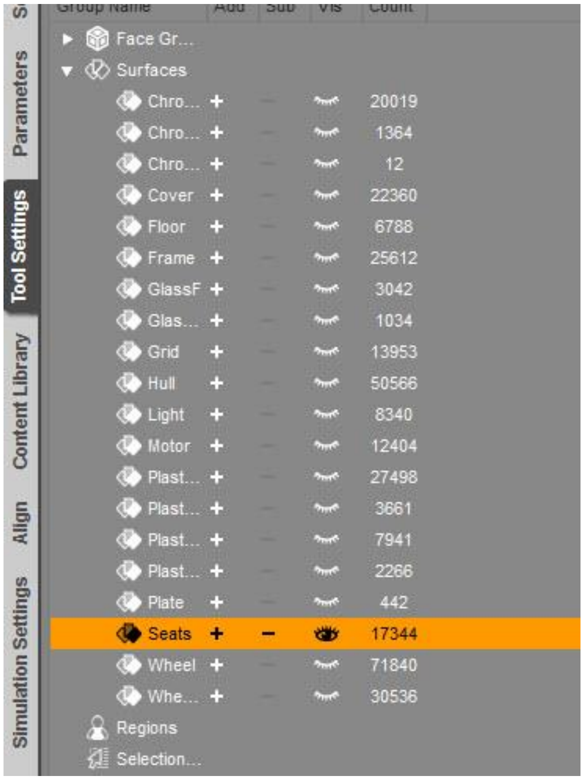
Segments : 20

Sides : 32

m. For the car model, I chose Car Ranger from Daz 3D. For the driver, I chose a Genesis 3 female. Pose the figure and make whatever adjustments are needed for her to sit in the driver seat with the hands in whatever final position you want them in.

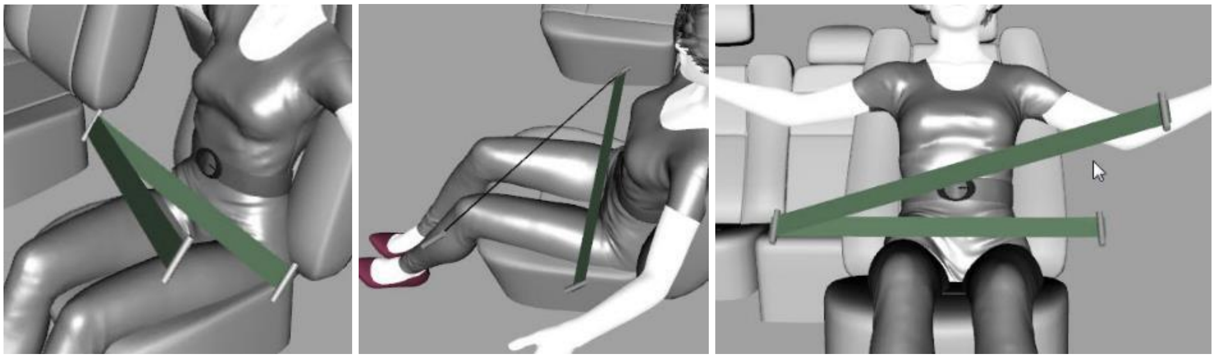
n. This next step will depend upon the car model you are using. Most of the geometry will interfere with and slow dForce simulation, so we want to temporarily hide it. Choose the car in the Scene pane. Select the Geometry Editor from the Tool Settings pane. Under Surfaces, click on each open eye and close it except for the Seats.

Depending upon the speed of your computer, it may also be helpful to temporarily hide each of the other seats (in this model there are 4 others).



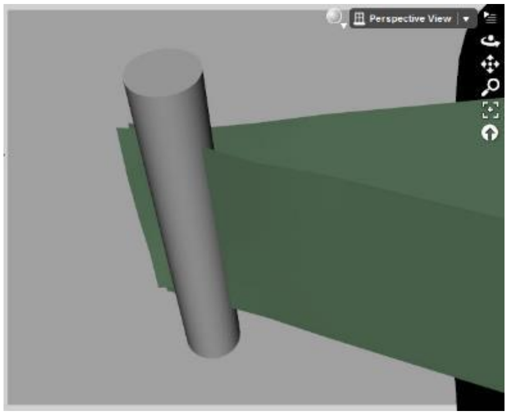
Group Name	Add	Sub	Vis	Count
▶ Face Gr...				
▼ Surfaces				
Chro...	+			20019
Chro...	+			1364
Chro...	+			12
Cover	+			22360
Floor	+			6788
Frame	+			25612
GlassF	+			3042
Glas...	+			1034
Grid	+			13953
Hull	+			50566
Light	+			8340
Motor	+			12404
Plast...	+			27498
Plast...	+			3661
Plast...	+			7941
Plast...	+			2266
Plate	+			442
Seats	+	–		17344
Wheel	+			71840
Whe...	+			30536
Regions				
Selection...				

o. Next we need to position the objects we created. Position the lap belt across the figure's lap, raised and slightly in front of the hips. Position the chest belt at an angle in front of the abdomen. Have the console end of the belts overlapping each other. I found it helpful to increase the length of the belts to 10% x-scale.

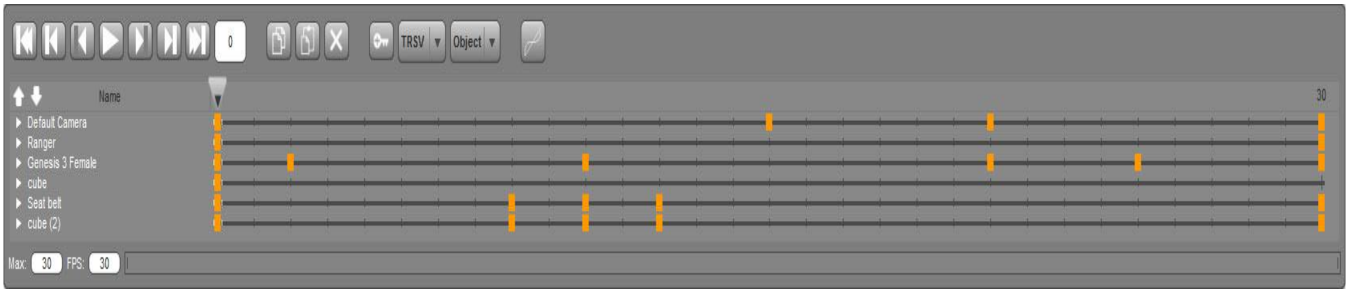


p. Position a helper cylinder at either end of the lap belt, with the first row of the belt's polygons embedded in the cylinder. Do the same for the door side of the chest belt (see image above).

The image to the right shows how the belts are embedded at the console end.



q. We will be doing an animated simulation. Setup a timeline as follows:



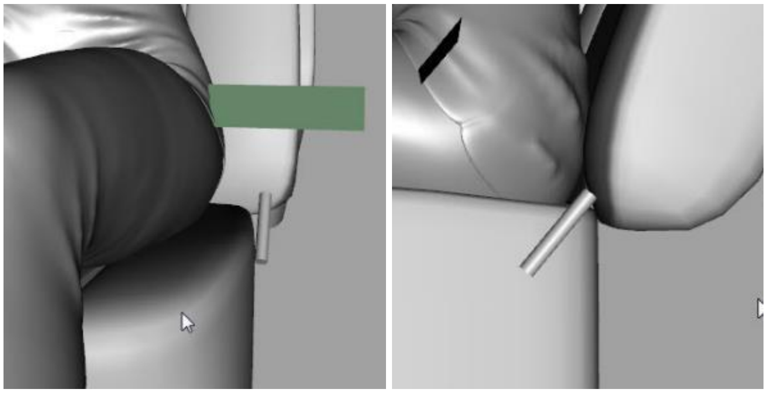
Max: 30 FPS: 30

i. Select the G3F figure, go to frame 30, and click on the set keyframe icon. This is to keep the figure's final pose set, as we will need to move the arms out of the way.

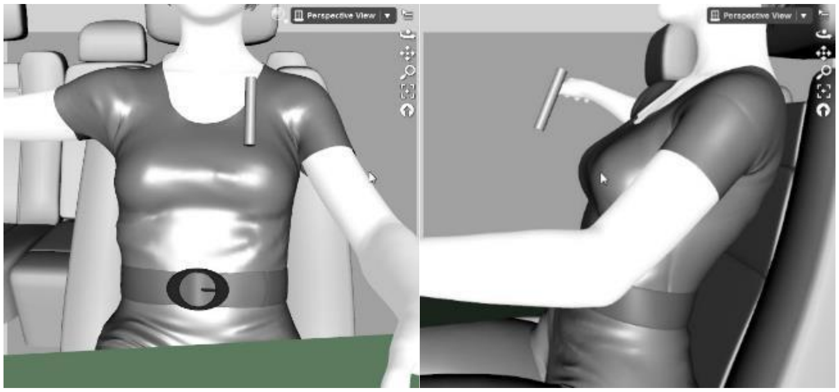
ii. At frame 2, set another keyframe for the G3F figure.

iii. At frame 0, move the figure down and back. This will allow the figure to rise up into the belt as the belt moves down towards the seat. Also move the figure's arms up and out to the side so the moving belts won't hit into them.

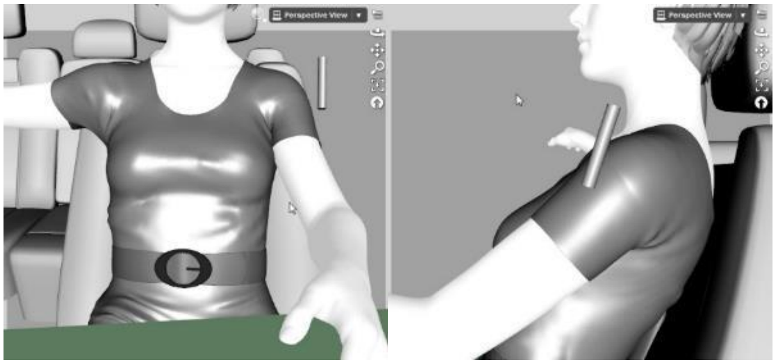
iv. At frame 8, move both the cylinders for the lap belt down and to the back and in towards the seat. This will be the final position of the belt end. (only one of the two cylinders are shown, but do the same for both).



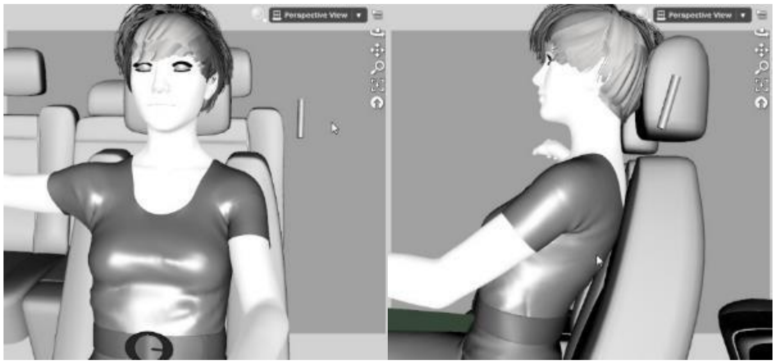
v. At frame 8, move the free end of the chest belt to the left and up, positioning it in front of the figure's left breast.



vi. At frame 10, move the free end of the chest belt to the left and up, positioning it near the shoulder.

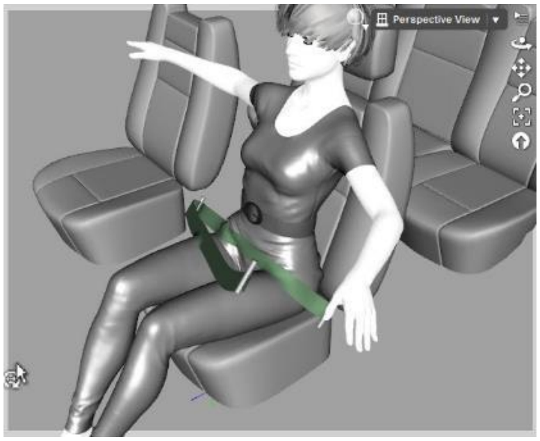


vii. At frame 12, move the free end of the chest belt to the left and up, positioning it near the door jamb where a seat belt would attach.

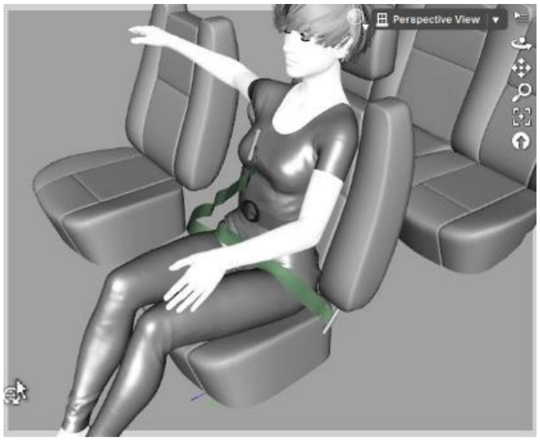


r. Run the dForce simulation. You should get the following results:

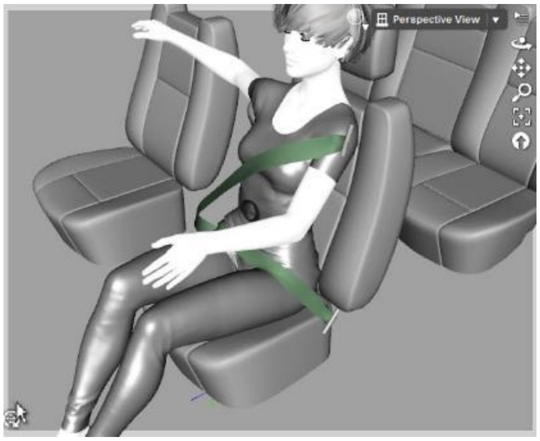
Frame 2:



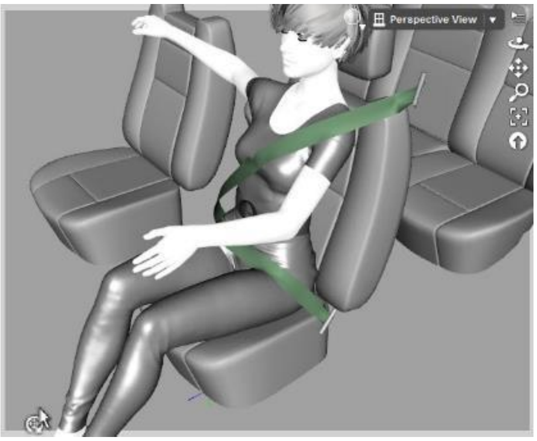
Frame 8:



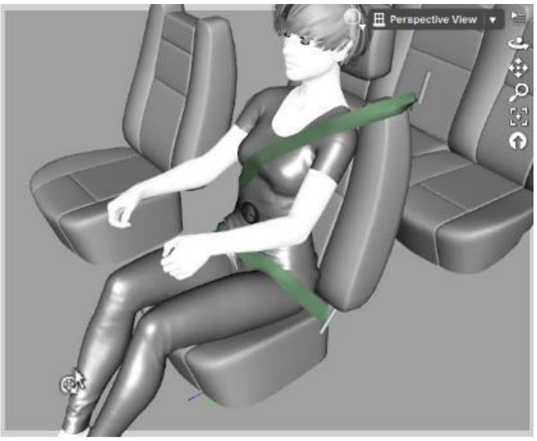
Frame 10:



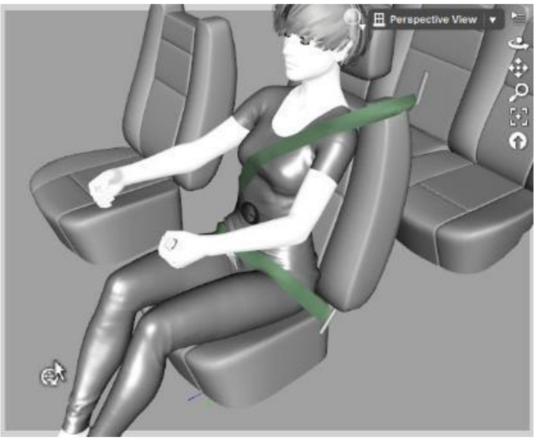
Frame 12:



Frame 25:

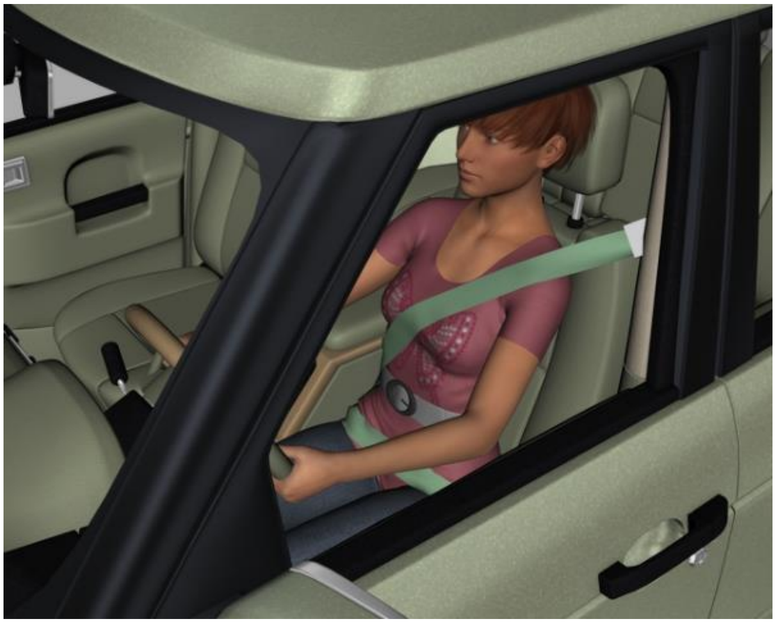


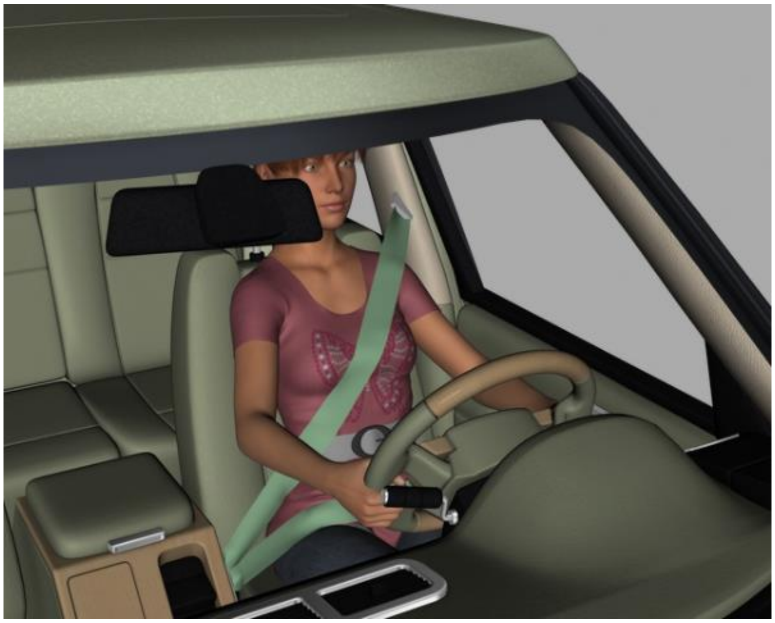
Frame 30:



s. Go back and unhide all the surfaces you hid in step n. Choose the car in the Scene pane. Select the Geometry Editor from the Tool Settings pane. Under Surfaces, click on each closed eye to open it. If you hid parts of the seat, a quick way to restore them is to click on the seat's open eye to close it then click again to reopen it.

t. Add a texture to the belt planes. Hide the cylinders if they are visible in your view. I also created a small cube to act as the door jamb anchor for the belt. Here are two final renders of the belt in use.





59. **Polyline Dynamic Add-On.** A new feature coming in Daz Studio 4.11 is the use of polylines. These are linear, non-rendering geometry which can be used as a dForce dynamic add-on. Unlike the polygon add-ons, these do not have to be hidden at render time to make them invisible.

    a. To use polylines, you must first install the Daz Studio Beta product, 4.11.0.231 or later. The beta does not replace your current Daz Studio general release (4.10 or earlier), both can coexist on the same computer.

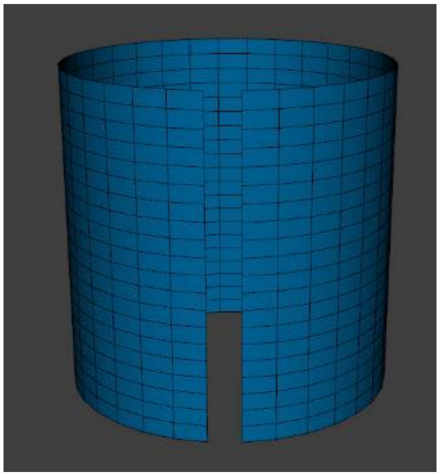
    b. Daz has released a script that makes adding polylines much easier. You can find it here. Save it to My Daz3D Library/Scripts Folder. You can also create polylines in some 3D modelers or by hand, but using the script is the easiest.

    c. If the Script IDE pane is not already open, select from the main menu Window/Panes (Tabs)/Script IDE.

    d. From the Script IDE pane’s menu, choose File/Open Script… and choose the downloaded script: Generate_Polyline_Dynamic_Surface_AddOn.dsa

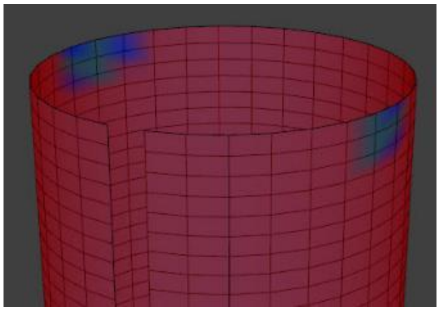
    e. Create a primitive cylinder in Daz Studio: two foot length, two foot diameter, 20 segments, 32 sides. Y-positive.

    f. In the scene pane, select the cylinder. From the Tool Settings pane, select the Geometry Editor and the Polygon selection tool (left icon in upper row of the pane). Select the top faces then right-click in the viewport and hide them (Geometry Visibility), repeat for the bottom faces. Next select a column of faces on the front and hide them. Then delete all the hidden faces (Geometry Editing). You should end up with something like this:

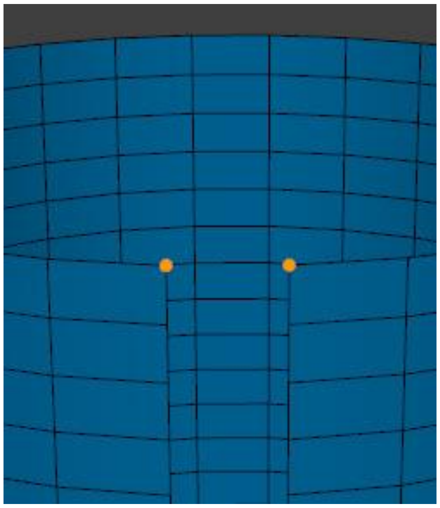


    g. Select the cylinder and from the main menu choose Edit/ Object/Geometry/ Add dForce Modifier: Dynamic Surface.

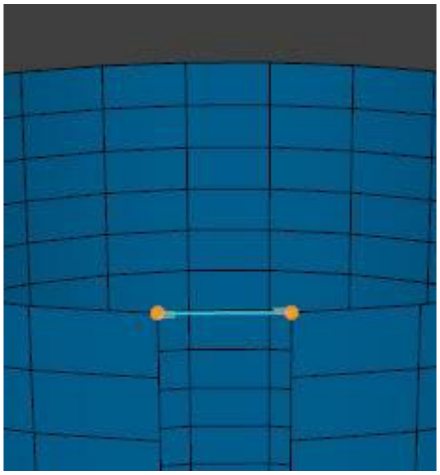
    h. From the main menu, choose Create/New dForce Modifier Weight Node. Go to the Node Weight Map Brush in the Tool Settings pane. Add the influence weight map. Paint a bit of blue on the upper left and right edges (remember to hold down the Alt key to remove the full-strength red). This will be enough to hold the cylinder in space during simulation.



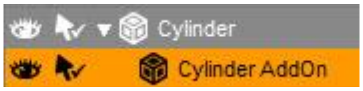
i. Continuing to use the Geometry Editor, choose the third icon at the top of the pane, the Vertex Selection tool. In the viewport, select two vertices on the prop. I picked one on either side of the split on the top row. Select one with the mouse, then hold down the Ctrl key and select the second.



j. Click on the Execute button at the top of the Script IDE pane. You will find that Daz Studio has added a new line between the two vertices.

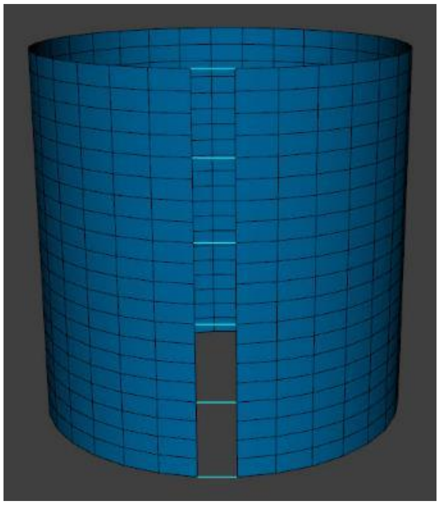


There will also be a new item with the word AddOn parented to the object whose vertices you selected.

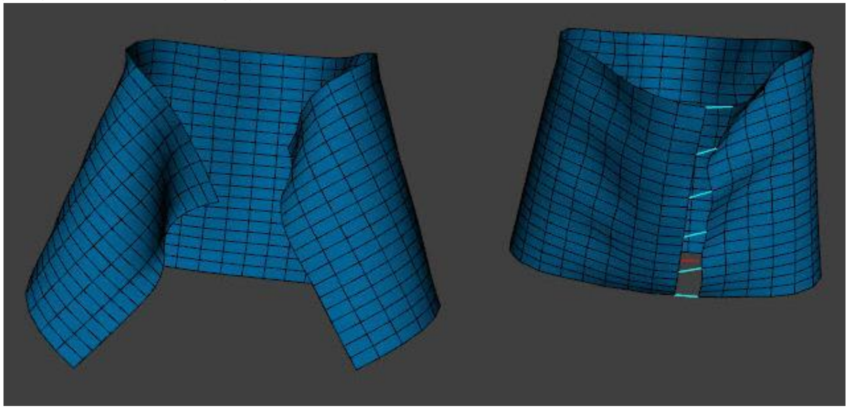


Cylinder
Cylinder AddOn

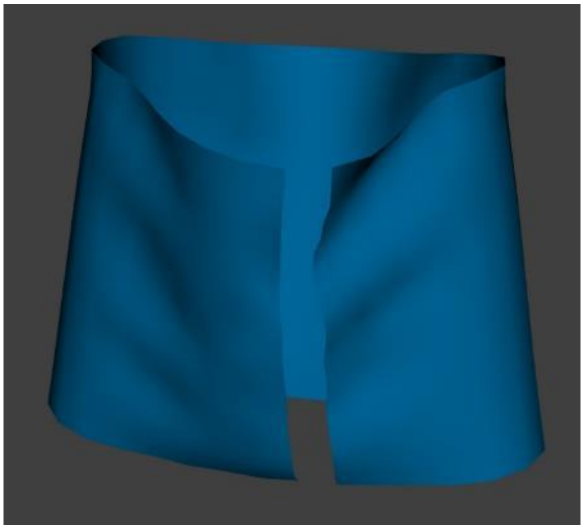
k. You can stop here, having made one polyline. Or you can select two new vertices 4 rows down, click on the Execute button again, and add another polyline to the addon. You will not see any change in the scene pane, as the new line will be added as part of the same object. You can repeat this as often as you like, adding as many polylines as needed.



l. After creating the polyline, you need to select the Cylinder AddOn in the scene pane and from the main menu choose Edit/Object/Geometry/Add dForce Modifier: Dynamic Surface Add-On. After doing this, run the simulation. You will find the polylines hold the gap closed (see below).



m. Although you can see the polylines in the viewport, they will be invisible in the render. That can be an advantage or a disadvantage, depending upon what you are trying to do.

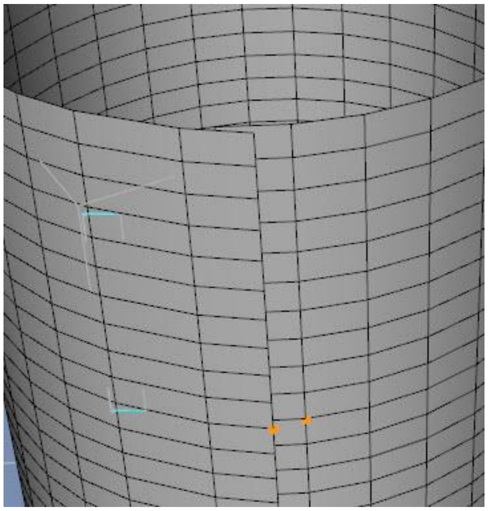


n. As another example, I will use the split cylinder I made in Silo for section 55. See that section for the initial steps.
   i. Import the cylinder into Daz Studio (use From: Silo to import it with the right dimensions).
   ii. Select the cylinder and add a dForce dynamic modifier.
   iii. Select the cylinder and add a dForce weight node. Add the dynamic strength map. Paint a bit of blue on the upper left and right edges (remember to hold down the Alt key to remove the full-strength red).

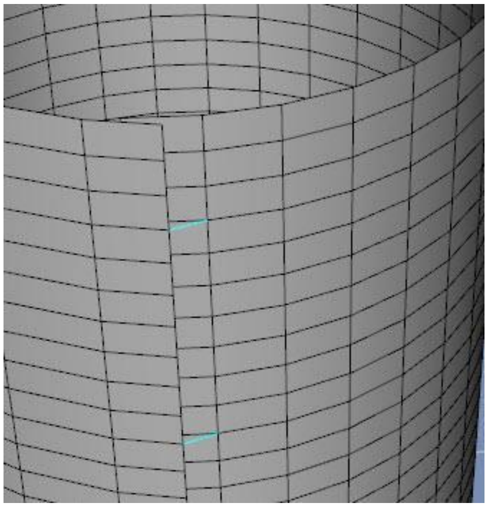
o. This cylinder shows up one oddity when creating the addon. The split I made in Silo is on the side of the cylinder. To center it in the scene, I rotated the cylinder 28 degrees. When you add the polylines, they do not appear where the selected vertices were located. Instead, they are rotated 28 degrees whether you parent them in place or not.

p. In spite of the misposition, you can go ahead and go to the main menu and choose Edit/Object/Geometry/Add dForce Modifier: Dynamic Surface Add-On. Run a simulation and the polyline(s) will pop into place and act as desired. Or do as I did, and rotate the AddOn 28 degrees. (on some other models, you may find you need to zero put X, Y, and Z translate as well).

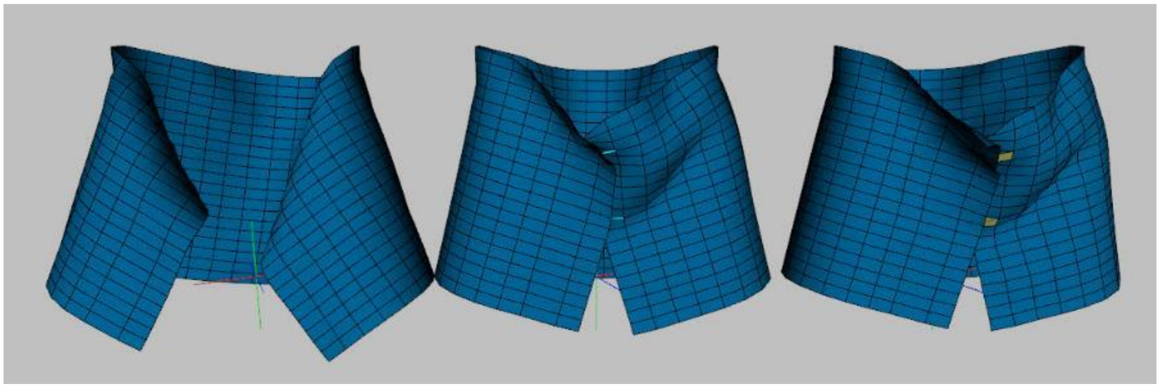
After creation of polylines:



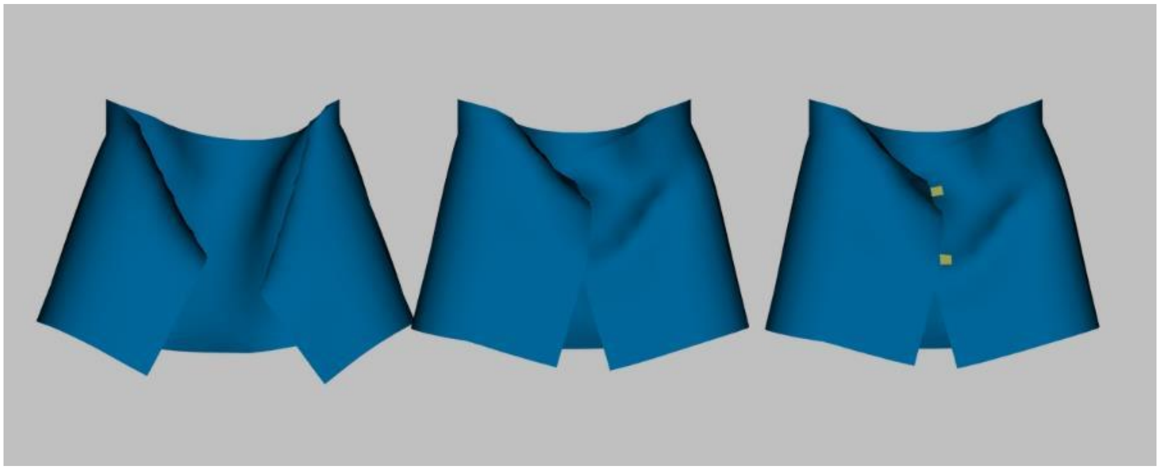
After rotation:



q. Below is the viewport after simulation, with the split cylinder without any add-on at the right, the polyline add-on in the center, and a polygon add-on at the left. The polylines and the polygon give similar results as they are in the same position.



After rendering, you can still see the polygons (until you hide them), whereas the polyline is not visible.



r. In section 55, I showed how you used the Transfer Utility after creating and before using the Add-on. I have found this was not really necessary. You can still do so, but the dForce add-on will work without it.

60. **Script to Quickly Add Dynamic Modifier to an Object.** If you do many simulations, the sequence of menu selections to add a dForce modifier can get tiresome. Here's a modified Daz script that allows you to do it in one step.

a. Download the basic script from here.

b. Save it to your My Daz3D Library/Scripts folder.

c. If the Script IDE pane is not already open, choose Window/Panes (Tabs)/Script IDE from the main menu.

d. Using the Script IDE menu, select File/Open Script and open the downloaded script.

e. Near the bottom of the script you will find the lines (currently around line 73 but that could change if the script is ever updated):

```
// Define the classname of the action we want to invoke the execution of
var sAction = "DzRestoreShapeAction";
```

f. Replace DzRestoreShapeAction in the last line with DzAddDForceModifierDynamicSurfaceAction

g. Do the same replacement 4 lines lower.

h. Save the modified script with a name like Add dForce Dynamic Modifier.

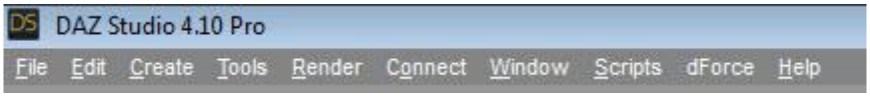
i. Now when you want to add a dynamic modifier, select an object in the Scene pane, execute the script, and the modifier is added.

j. To make it even easier, go to the Content Library pane, navigate to the Scripts folder, find the icon for the newly saved script, right click and choose Create CustomAction. Now instead of using the Script IDE, go to the main DS menu, choose Scripts and click on the Add dForce Dynamic Modifier item.

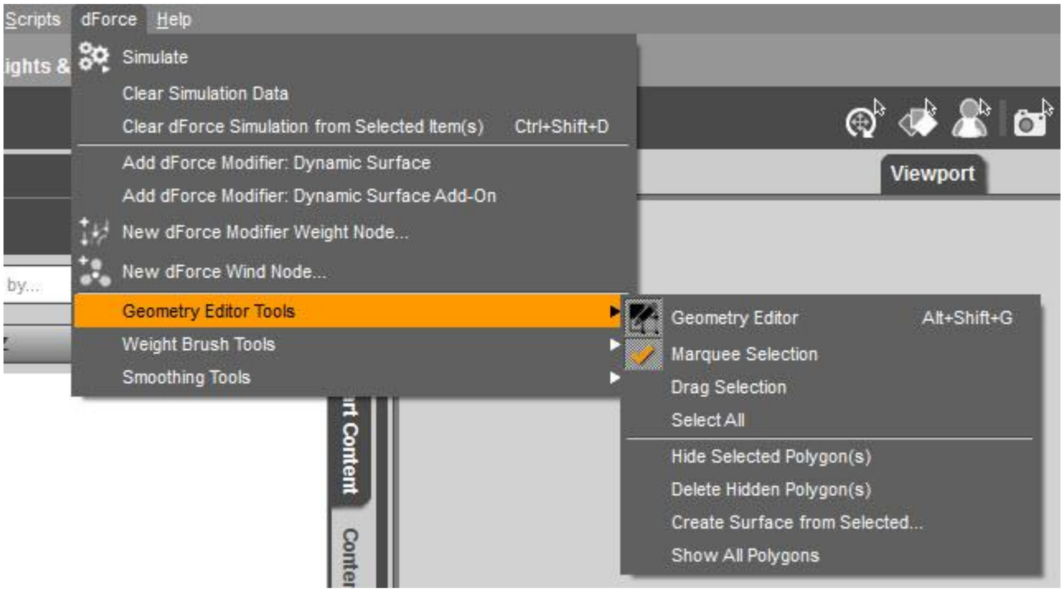
k. Right after doing step j, close Daz Studio. That will save the new custom action in case DS crashes. It will then be readily available every time you use DS.

61. **dForce Menu Addition.** Building upon a comment made by barbult on the Daz Forum, I modified the main Daz Studio menu to include a personalized dForce menu. Instead of remembering the many locations and menu/submenu combinations needed for various actions, I have those I use most often in one easy to access spot.

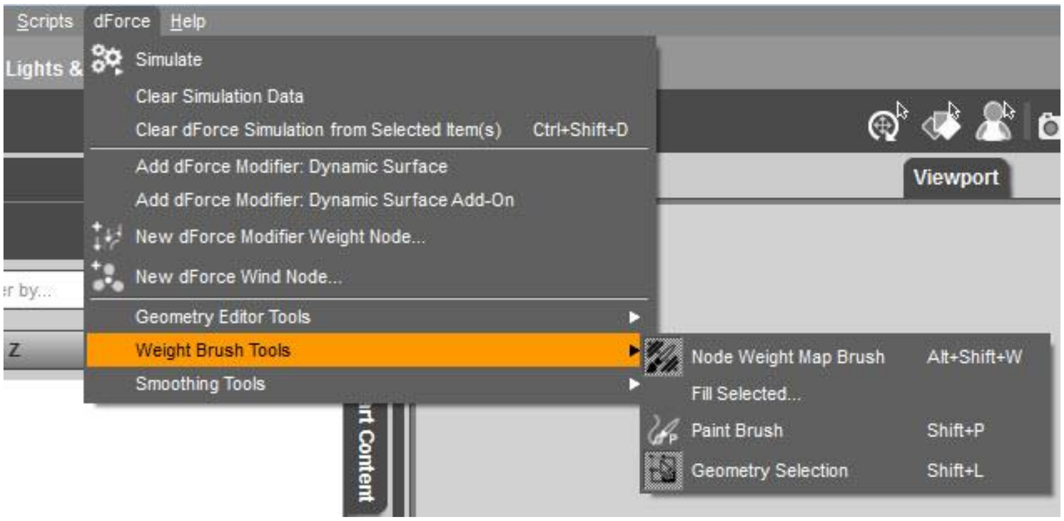
a. Below shows the modified main menu. The dForce choice is second from the right.


DAZ Studio 4.10 Pro
File Edit Create Tools Render Connect Window Scripts dForce Help

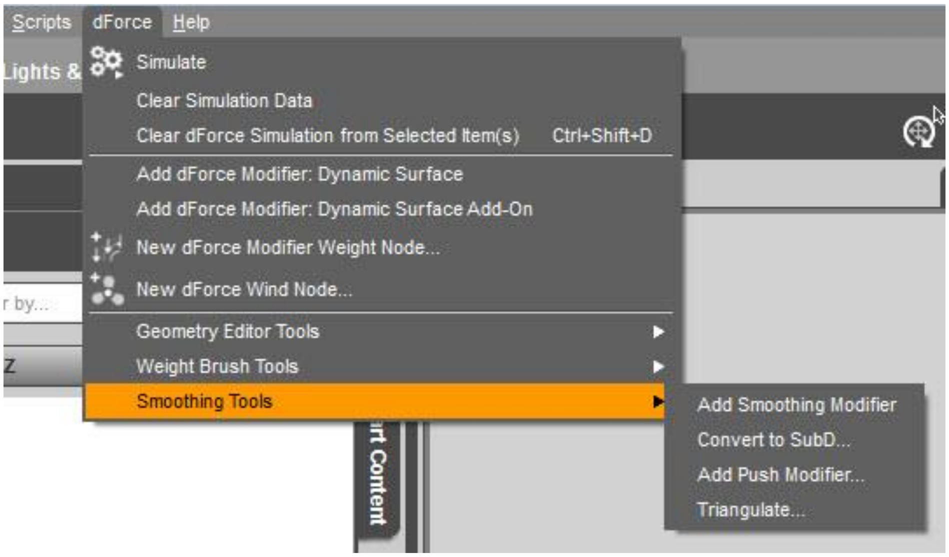
b. This is the structure I set up. The first three items are so I don't have to open the Simulation Settings pane so often, the next four consolidate several Edit and Create Menu actions, then the last three are submenus for tools I frequently use together with dForce.



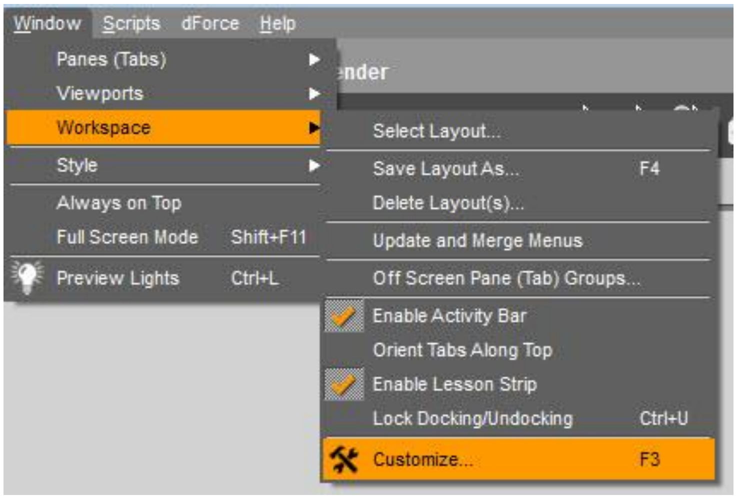
The image above also shows my most often used Geometry Editor actions. The image below shows the same for the Weight Brush tool. Although I can call the Weight Brush tool from this menu, I still have to go to the Tool Settings pane once to create the map.


Scripts
dForce
Help
Lights &
Simulate
Clear Simulation Data
Clear dForce Simulation from Selected Item(s) Ctrl+Shift+D
Add dForce Modifier: Dynamic Surface
Add dForce Modifier: Dynamic Surface Add-On
New dForce Modifier Weight Node...
New dForce Wind Node...
Geometry Editor Tools
Weight Brush Tools
Smoothing Tools
Node Weight Map Brush Alt+Shift+W
Fill Selected...
Paint Brush Shift+P
Geometry Selection Shift+L
Viewport
rt Content

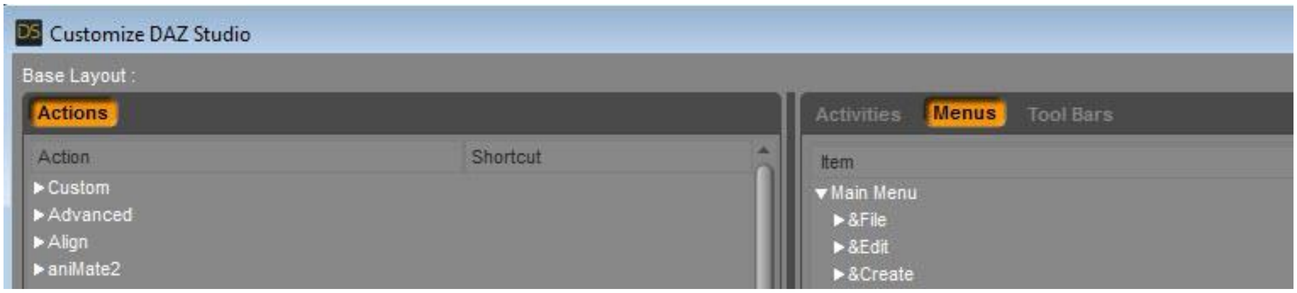
This last image shows the Smoothing tools. This allows quicker access then using the typical menu/submenu navigation.



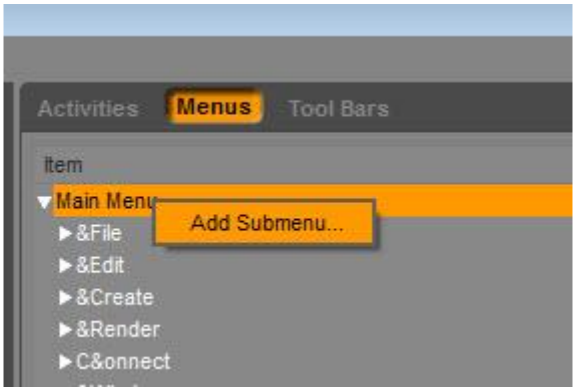
c. To create a similar menu for yourself, you have to first open the Customize tool. From the main menu, choose Window/Workspace/Customize.



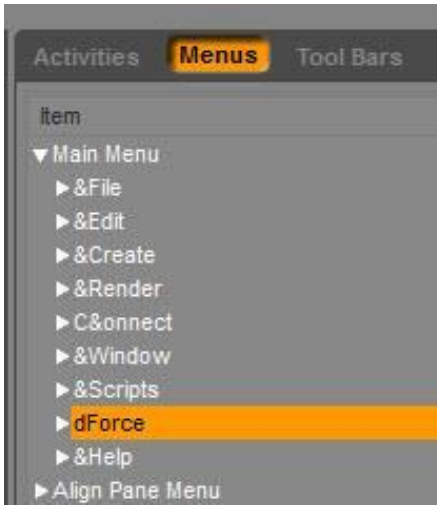
d. You’ll see a pop-up that looks similar to below. In the right pane at the top, click on the Word Menus.



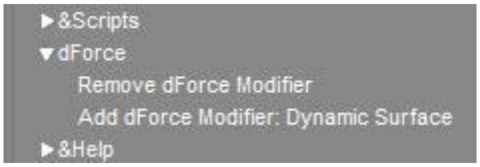
e. Right click on Main Menu in the right hand pane and Select Add Submenu. Type dForce.

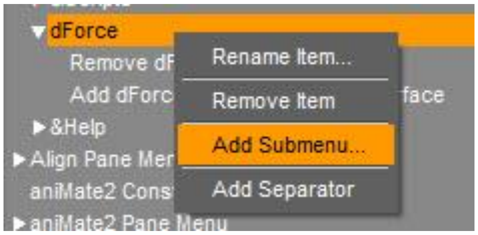


f. Drag the word dForce down to your preferred location in the main menu.



g. To populate your menu, select actions in the Left Hand Actions pane and drag and drop them onto the dForce menu item in the right pane. The most recent added action will be at the top. You can later drag it down into position. You can also add a separator or sub-menu by right-clicking on the word dForce and selecting the desired action.

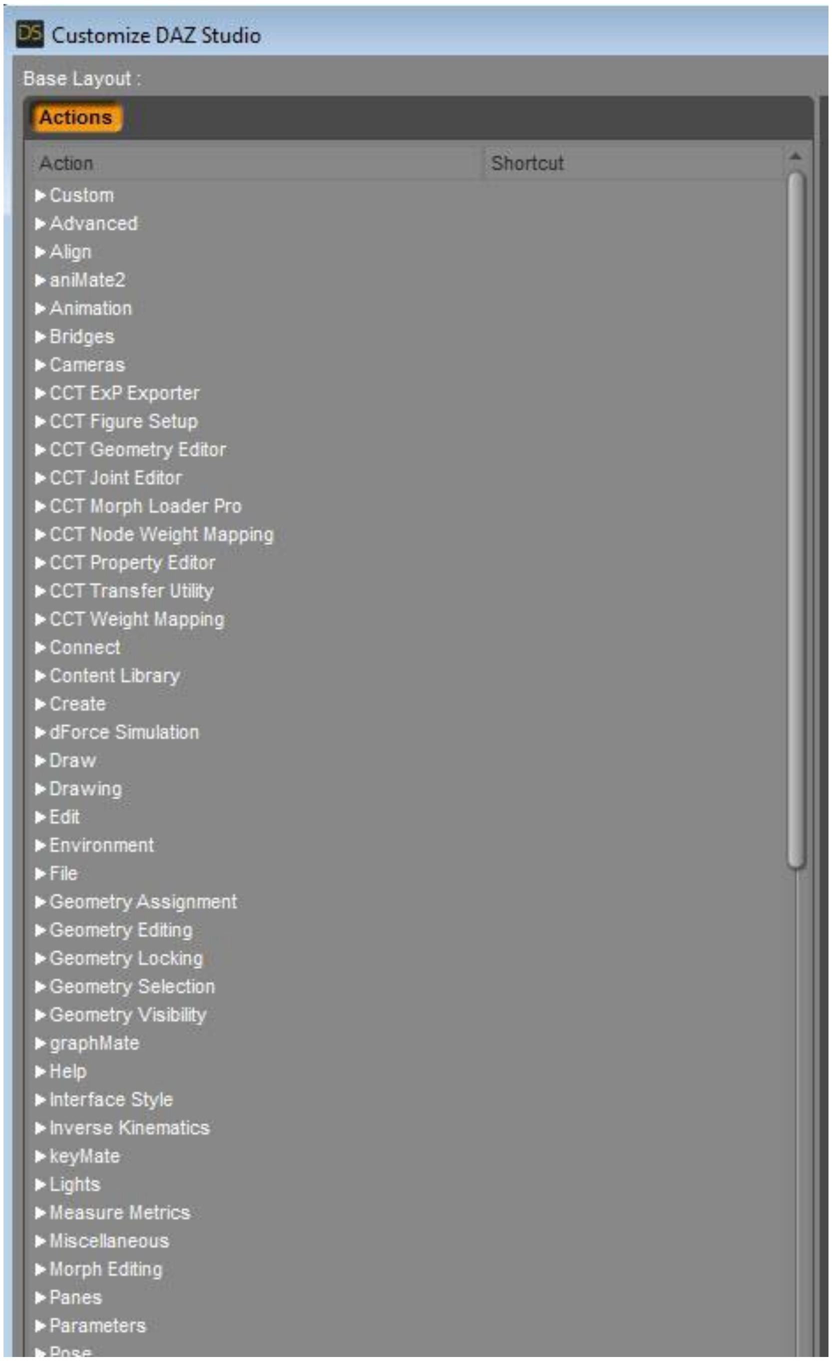




h. The hardest part in constructing the menu is finding the Actions in the left hand pane. The ones I used were found in:

- dForce Simulation
- Geometry Assignment
- Geometry Editing
- Geometry Selection
- Geometry Visibility
- Scene Hierarchy
- Simulation
- Viewport Tools
- Weight Map Brush

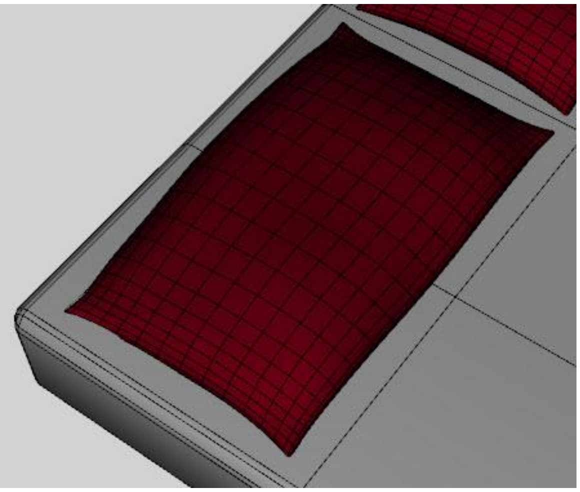
i. After adding and moving the actions into position, accept the result. The new menu structure will now be in effect. You will lose this new menu if Daz Studio were to crash, so close DS and reopen before proceeding. Alternatively, you can save your layout by choosing from the main menu Window/Workspace/Save Layout As...



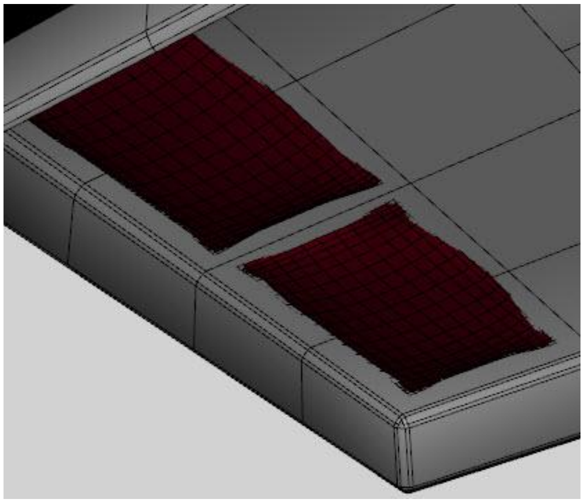
62. **Pillows**. Almost everyone lets their head sink into a pillow at night but citizens of Dazland often find their pillows hard and stiff, unyielding to their heads. Here are a few examples of pillows that work.

a. One I’ve already shown early in this series of tutorials is from Modern Room Bedroom (see Getting a Figure’s Head to Sink into a Pillow). I was lucky I used that one as it dForced readily for two reasons:

(1) It has quad faces (see image to right)

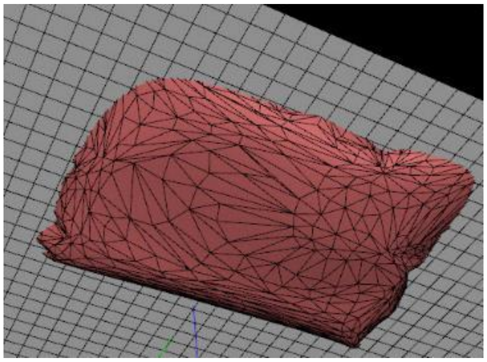


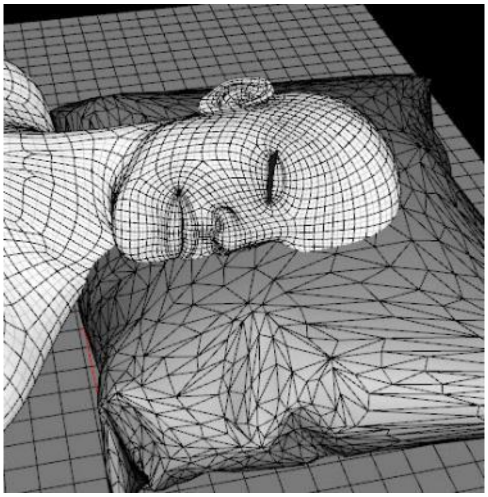
(2) The lower polygons of the pillow are embedded below the surface of the mattress which locks them in place when negative gravity is applied (see image below). Too bad not all pillows are made like this. You can load these pillows as a prop, so they can be used in other scenes.



b. A second pillow is from The Breakfast Nook. Like many others, it has triangular faces which causes issue with dForce. It has also been somewhat deformed by the vendor who wanted them to look like they had been “in use”. Here’s what to do to use it with dForce:

i. Create a plane or cube to serve as a mattress. Load the pillow onto it and lower it so it penetrates into the plane or cube. Add a dForce dynamic modifier to the pillow.



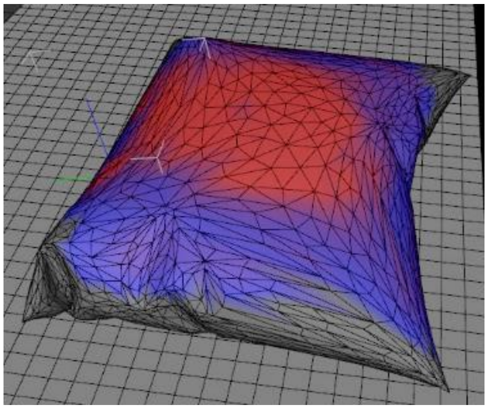


ii. Next load and position a figure. You will see the pillow cuts right into the face. On the animation timeline, set the figure so the upper body is raised at frame 0. At frame 8, have the body lowered and turn the head to the side as shown.

iii. Add a dForce weight node to the pillow: choose Create/New dForce Modifier Weight Node. A new child weight node will appear in the Scene pane under the selected object. Select the new weight node object, and from the Tool Settings pane, select Node Weight Map Brush in the Active Tool drop-down box. In the lower section of the pane, choose dForce Simulation::Influence Weights in the unused Maps drop-down box then click on the Add Map button.

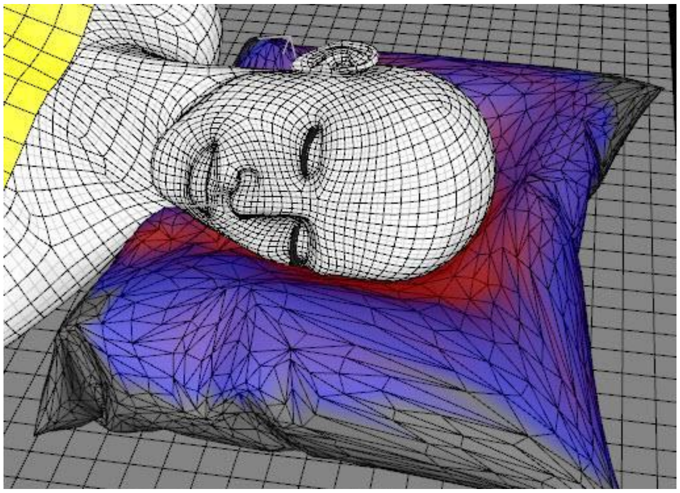
iv. The object will be red showing that the weights on the map are at the full strength of 1. Right click in the viewport and choose Geometry Selection/Select All. Then right click in the viewport and choose Weight Editing/Fill Selected and enter 0% to clear the weight.

v. Choose the paint brush mode (4th icon from right at the top of the pane, has a small P in lower right corner). Start to paint around the head until it becomes red again. Paint less on the surrounding area so it is blue. Use the smooth brush mode (3rd icon from right at the top of the pane, has a small S) along the colored edges to make the transitions smoother.

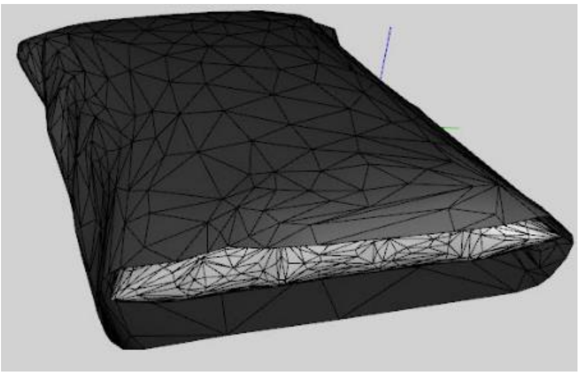


vi. The triangular faces will cause the mesh to explode. In the Surfaces pane, set Bend or Stretch Stiffness to 0.3. In the Simulation Settings pane set gravity to -0.1 then run the simulation.

vii. The head will have depressed the center of the pillow where the red was painted. There will be less deformation as you move towards the gray areas.

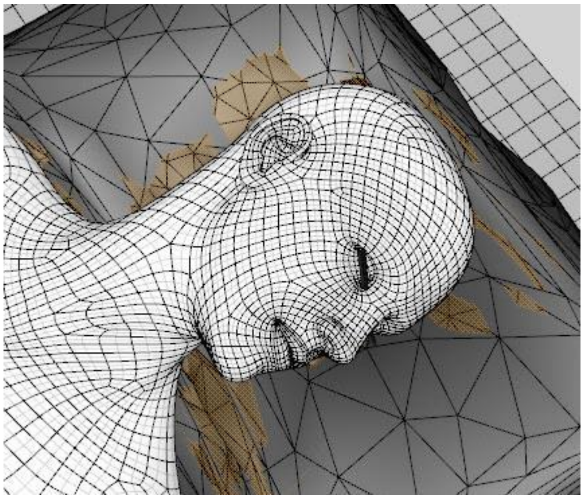


c. A third pillow is from Day in Bed. This one is tricky because although it is one object, it has a pillowcase around the pillow. There's a mesh within the mesh that will create issues. It also has the triangular faces that cause problems with dForce.

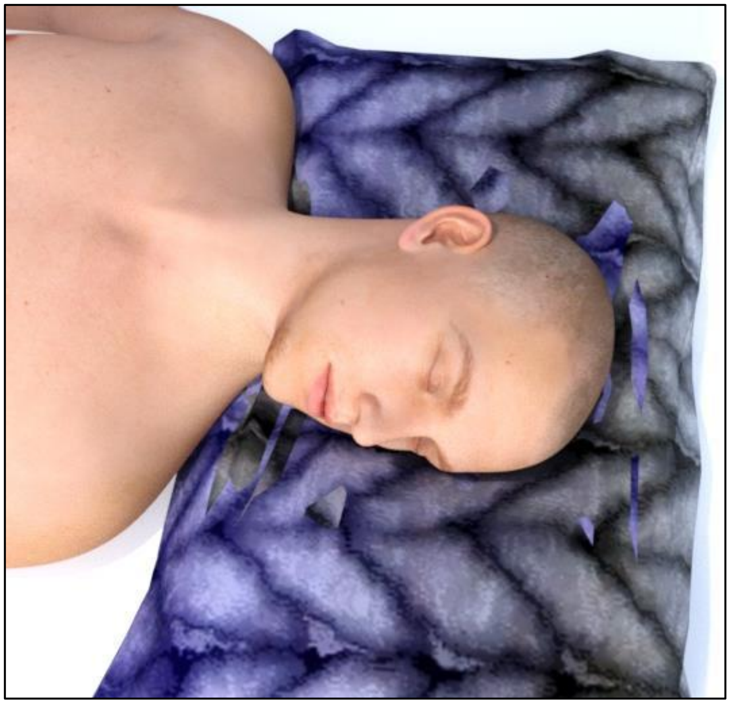


i. Repeat all the steps in section b for this new pillow.

ii. After simulation, you will find that the interior part of the mesh (orange) has protruded through the exterior mesh. The interior has a different mesh design, in particular, smaller faces. Because the meshes have different size faces they behave differently. Plus when painting the weight map, it was applied to both interior and exterior as there is just one surface.



You may get away with this behavior if you have a single-color shader or material, but for most shaders you'll see the interference as in the image.



iii. To correct this, we'll first create a second surface:

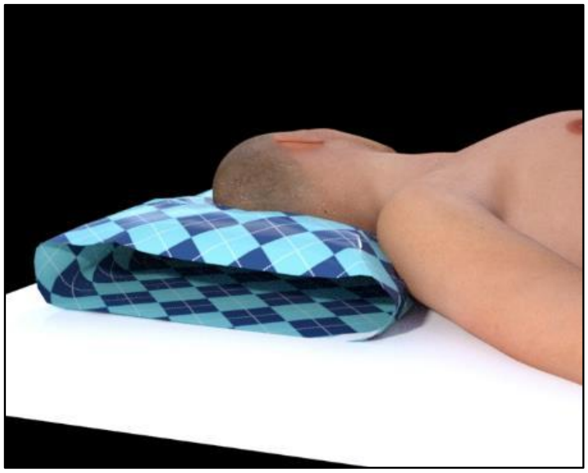
1. Select the pillow in the Scene pane.
2. From the Tool Settings pane, select Geometry Editor in the Active Tool drop-down box. Right click in the viewport and choose the Selection Type/Polygon Selection and drag Selection Mode.
3. Click one face of the interior pillow (you can access it from the side). Right click in the viewport and choose Geometry Selection and click on Select Connected.
4. Right click in the viewport again and choose Geometry Assignment/Create Surface from Selected and give the new surface the name pillow.
5. The new surface won't show the Simulation property group in the Surfaces pane until you re-add the dForce dynamic modifier to the pillow.

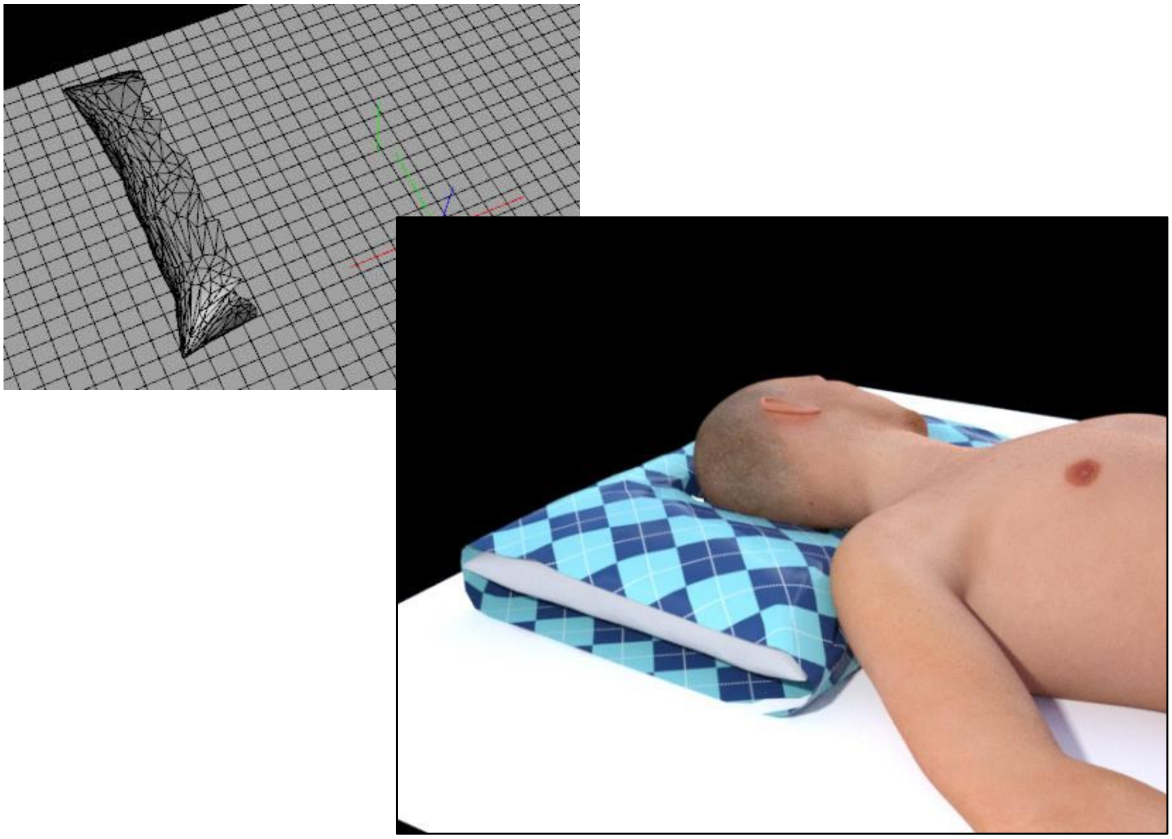
iv. We could stop here if you set the camera so you don’t see into the gap on the left side.

1. Right click in the viewport and choose Geometry Selection/Select By/Surfaces/ Pillow.
2. Right click in the viewport again and choose Geometry Visibility/Hide Selected Polygon(s).
3. Now that the pillow is hidden, leaving only the pillowcase, you can do a simulation and get good results.

v. If your camera does need to include the left side, you’ll see that the pillow is missing. You can change this by hiding most of the interior:

1. Right click in the viewport and choose Geometry Selection/Select By/Surfaces/Sheet then Geometry Visibility/Hide Selected Polygon(s). This hides the pillowcase so you can do the next step.
2. Next use the Marquee selection tool to select the left hand portion of the interior mesh then Geometry Visibility/Hide Un-Selected Polygon(s) to hide the rest.
3. In the Geometry Editor click on the eye icon for the Sheet surface to turn it back on. Run the dForce simulation and render the results.



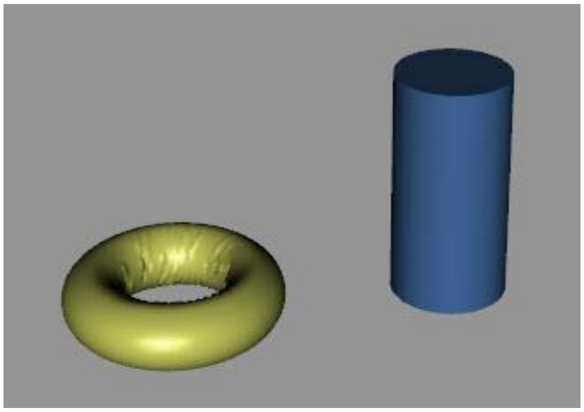


63. **Momentum Transfer**. Some have asked if dForce can be used to cause a ball to bounce. It can't (at least so far) but it can transfer momentum and cause an object to fly away. I first saw hints of this while testing a particular scene and then I changed simulation settings and surface settings to enhance it. The animated gif shows what happens.

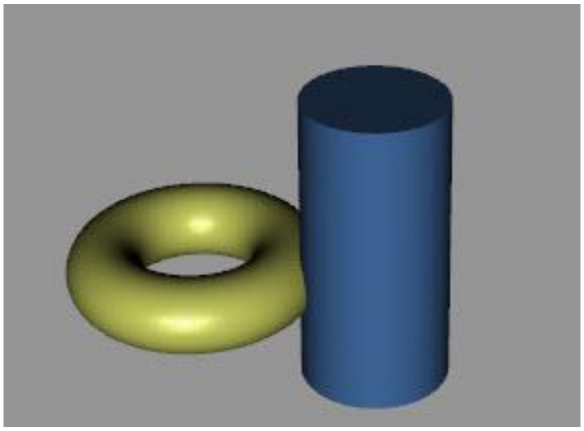
   a. Create a torus of 1.5 inch major diameter and 1 inch minor diameter, Y-positive, 100 segments and 32 sides. Add a dForce dynamic modifier to it.

   b. Create a cylinder 4 inches high, 2 inch diameter, 20 segments and 32 sides.

   c. Set up a 60-frame animation so at frame 0 the cylinder is about a torus diameter away.



   d. At frame 10, have the cylinder move forward and collide with the torus off-center.



   e. Set gravity to 0. This means the torus will neither rise nor fall unless it undergoes a collision.

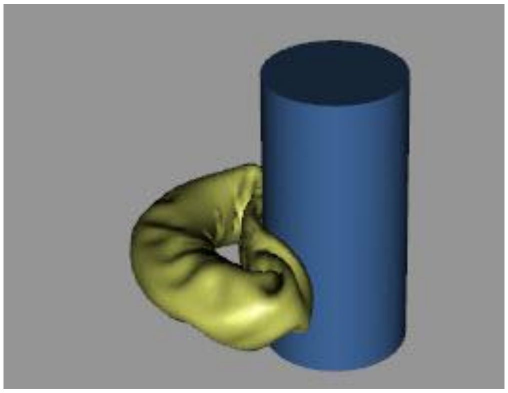
   f. Set air resistance to 0. This removes drag and allows the torus after the collision to move farther and faster.

   g. Set Buckling Stiffness to 100% and Buckling Ratio to 0. This will mostly keep the torus from collapsing in on itself when hit.

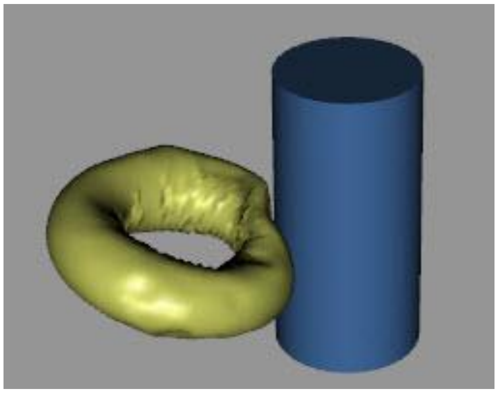
   h. Set density to 10. This will make the torus surface stiffer.

   i. Run the simulation. You will see the torus partially collapse when struck by the cylinder. Momentum is transferred to the torus, the torus's surface will rebound, and it will fly away.

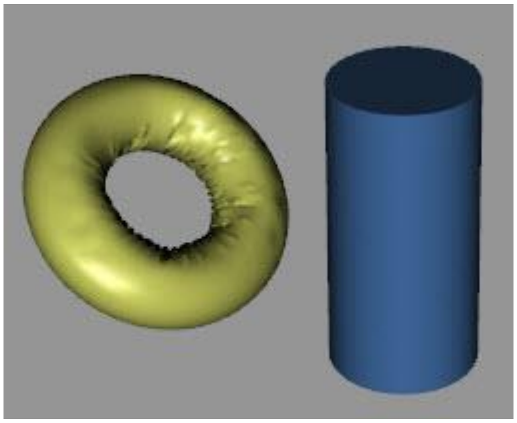
Frame 10:



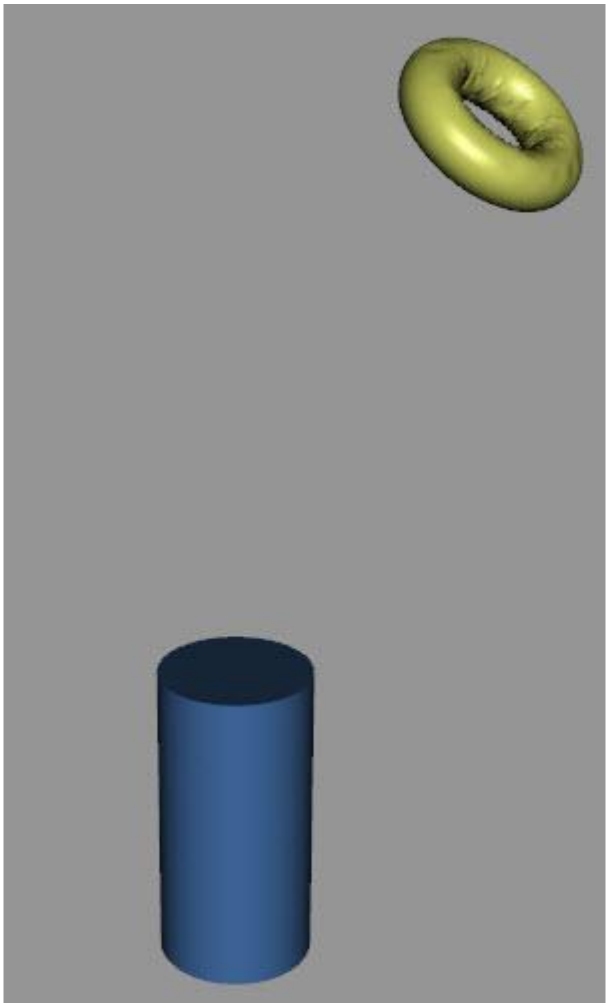
Frame 16:



Frame 30:



Frame 60:



j. What is this good for? I don't know but it suggests there may be a way to get dForce objects to bounce off surfaces.

64. **Chair Cushions**. Here's an example showing how to get a chair cushion to deform from a figure sitting in it. I used a chair from Back Door Walkway. As with clothing, some chairs simulate well and others will explode or crash DS. The only way to know is to give it a try and use a weight map to remove as much of the unneeded geometry from the simulation.

a. Load a chair. Set the viewport to Wire Shaded and check for trigons (which make objects harder to dForce. The chair I selected had quads as shown in the image).

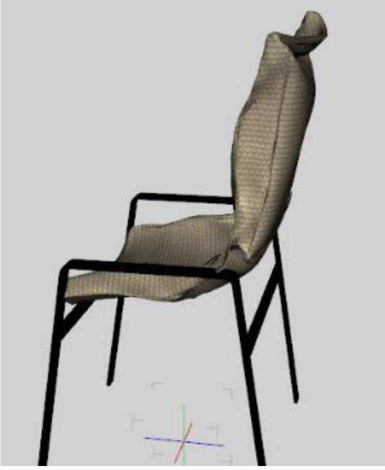
b. Select the chair in the Scene pane and go to the Surfaces pane and see how many surfaces are present. If only one, you will need to use the Geometry Editor to select all the polygons associated with the frame and assign them to a new surface. If more than one, your chair model likely has that done already.

c. Add a dForce dynamic modifier to the chair.

d. Go to the Surfaces pane and for any surfaces related to the chair's frame, set Visible in Simulation to Off.

e. For cushion surfaces, I would set Density to 25 as this will give a stiffer surface.

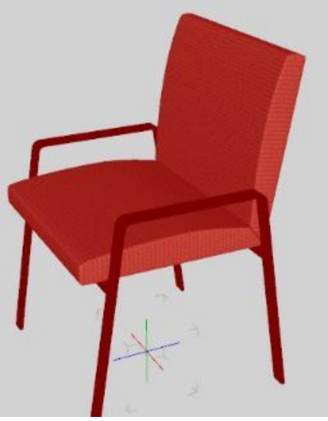
f. In the Simulation Settings pane, set Gravity to -0.2 as this will let the cushion move up and around the figure. Set Frames to Simulate to animated timeline.



g. Load a figure that is standing in front of the chair in frame 0 and set a keyframe on the animation timeline. At frame 10, apply a sitting pose on the figure. Adjust the figure so the buttocks are embedded somewhat down into the cushion and the figure's back is embedded in the seat back cushion if there is one.

h. Run a simulation to see how the cushion deforms. You will likely find the cushions deform more than you want (as is the case for this chair, see image. Note that I've hidden the sitting figure). Since there's a lot of deformation around the sides and top, I added a dForce modifier weight node.

i. Select the added node and choose the Node Weight Map Brush tool in the Tool Settings pane. First add an influence weight map then use the brush or other tools to select all the polygons associated with the underside or backside of the cushions. Once selected, right click in the viewport and choose Weight Editing/Fill Selected and set to 0. This will keep these polygons from changing during the simulation. The first image show all the cushion and frame with an influence weight of 1 (it doesn't matter that the frame is at one as we turned its surface Off for Visible to Simulation). The second



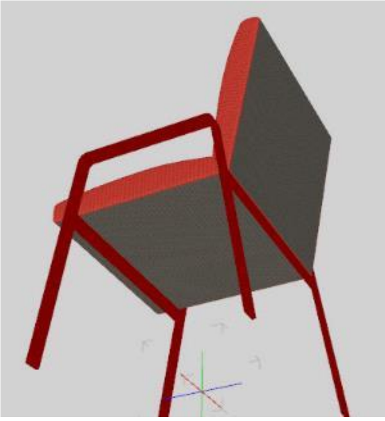
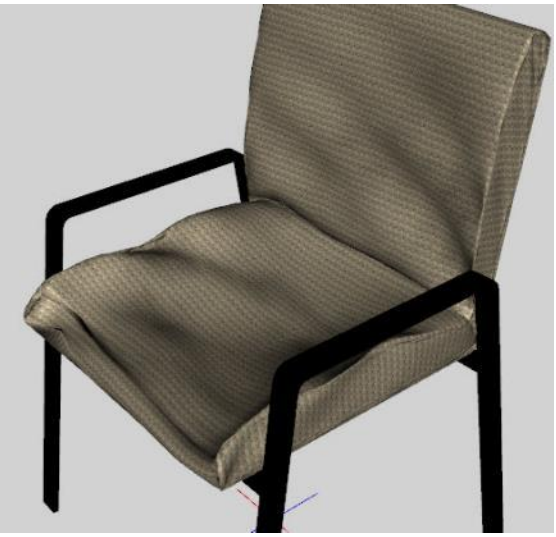


image shows the underside and backside polygons are free of red, so the influence weight in those areas is 0.

j. Run the simulation and you should see the cushion move up and around the figure. How it moves depends upon how the mesh is structured, the shape of the figure, its pose, and other factors. The left image shows the figure in the chair. In the right image I hid the figure so you can see how the chair deformed. If this is more deformation than you like, you can choose an earlier frame or set density even lower.



65. **Men's Underwear**. I've started digging into applying dForce to existing clothing. As most have experienced, it's hit or miss. Now that there's an ever increasing amount of dForce-designed clothing available in the stores, you could rely on that but most users will have an extensive back catalog of clothing they may want to use. I decided to start with some of the items that may be overlooked. In this section, I'll look at underwear.

a. The first one is Basic Wear Boxer Briefs that comes with Genesis 3 Starter Essentials. These are tight fitting conformed clothing that will move with the figure's pose.

b. Load a Genesis 3 male and the briefs.

c. Go to the Parameters pane and turn on smoothing for the briefs.

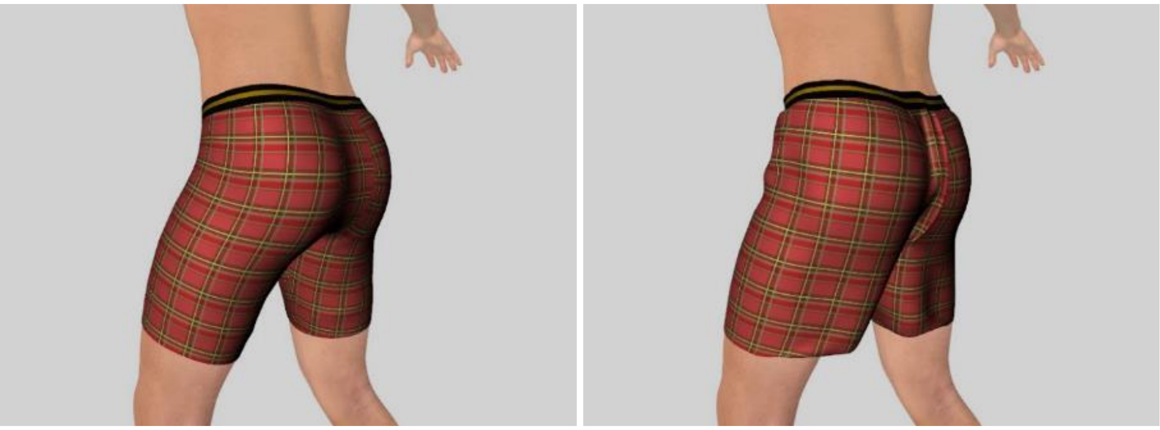
d. Set up an animated timeline with an exaggerated crouching pose at frame 12. I used Crouch Ready from DA Lucian 7 Shadow Pose Set.

e. Add a dForce dynamic modifier and run with default surface settings. It works but you'll see some separation from the waistband and the main body of the briefs. They will also be very tight fitting.

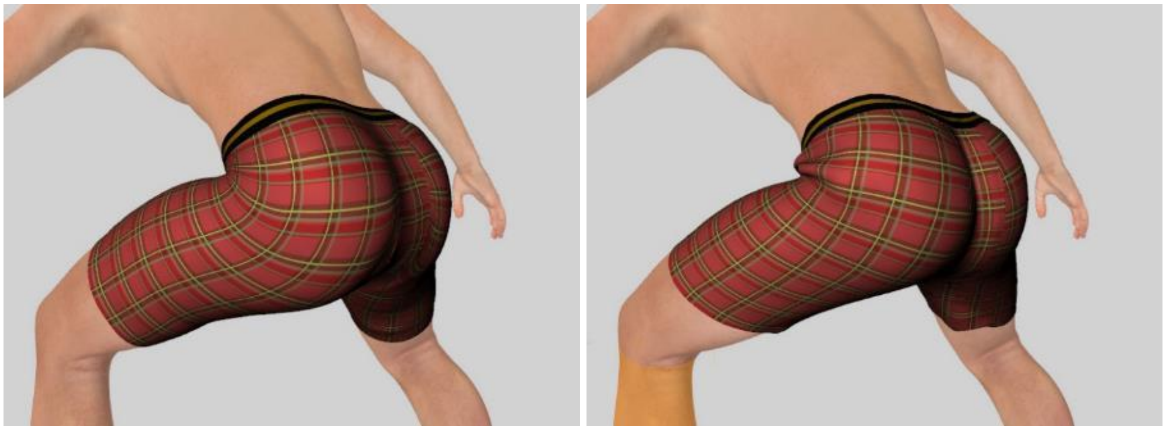
f. I kept the waistband surface at default settings but changed the main surface to Stretch Stiffness 0.3, Bend Stiffness 0.2 and Contraction-Expansion Ratio to 110%. The latter gives a somewhat looser fit.

g. The following sequence of frames compares the non-dForce results (left) with the dForce results (right):

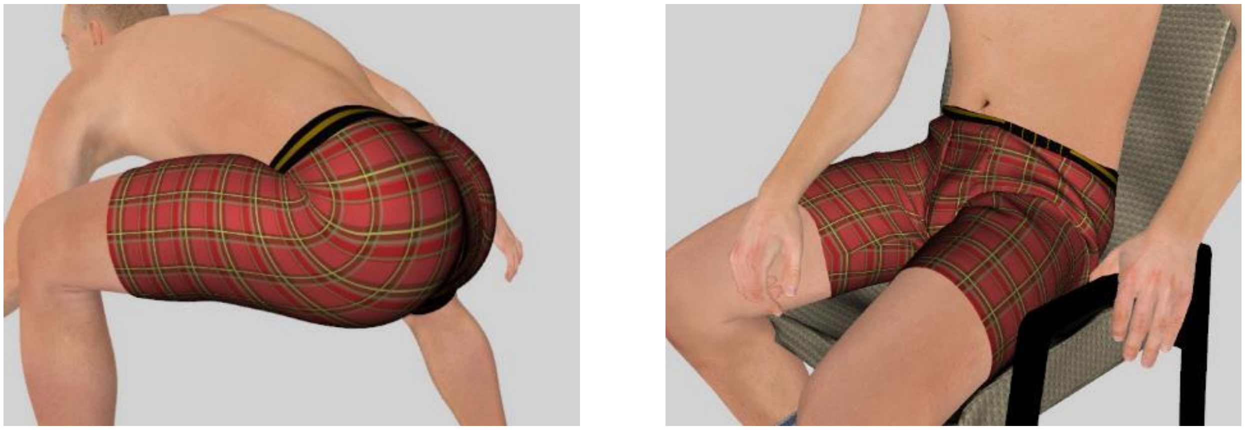
Frame 4: Looser fit with dForce



Frame 8: dForce allows some folding of fabric by the hips and a less stretched texture pattern

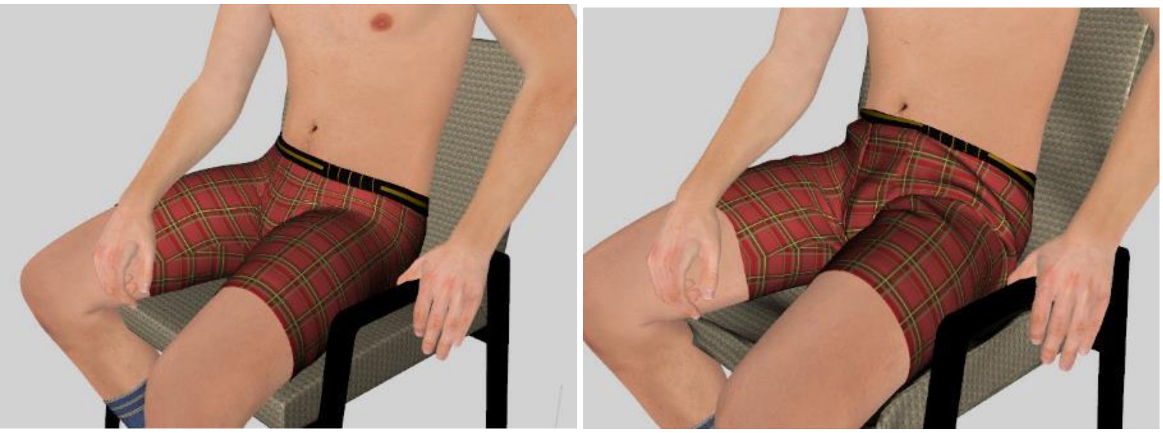


Frame 12: the full crouch



You can use more or less Contraction-Expansion to vary the tightness of fit.

h. Here are the same briefs but in a sitting pose. Contraction-Expansion Ratio was set to 105%. No dForce on left, dForce on right.

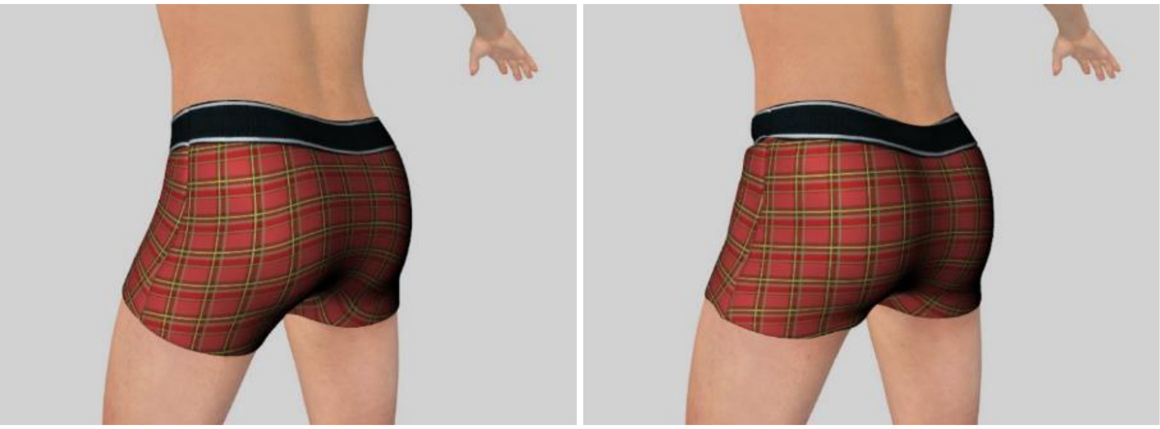


i. The next item tested is BV Underwear from [Beach Vibes Outfit for Genesis 3 Male(s)](#).

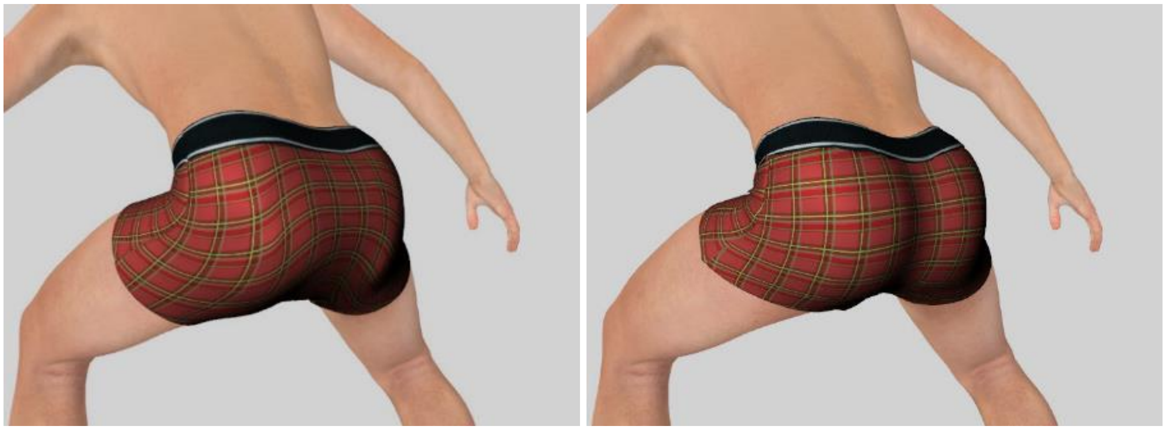
j. These exploded with default settings so I used dForce settings similar to the first example (step f) except Contraction-Expansion was 103%. You could get by just changing Bend Stiffness to 0.2.

k. The following sequence of frames compares the non-dForce results (left) with the dForce results (right):

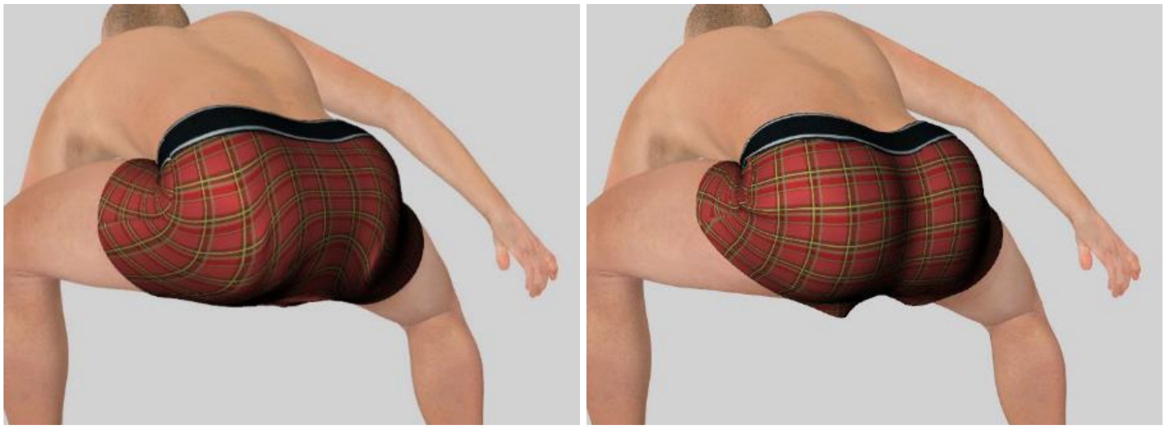
Frame 4:



Frame 8:



Frame 12:



66. Shorts & Swim Trunks. These are similar in construction to the underwear covered in section 65.

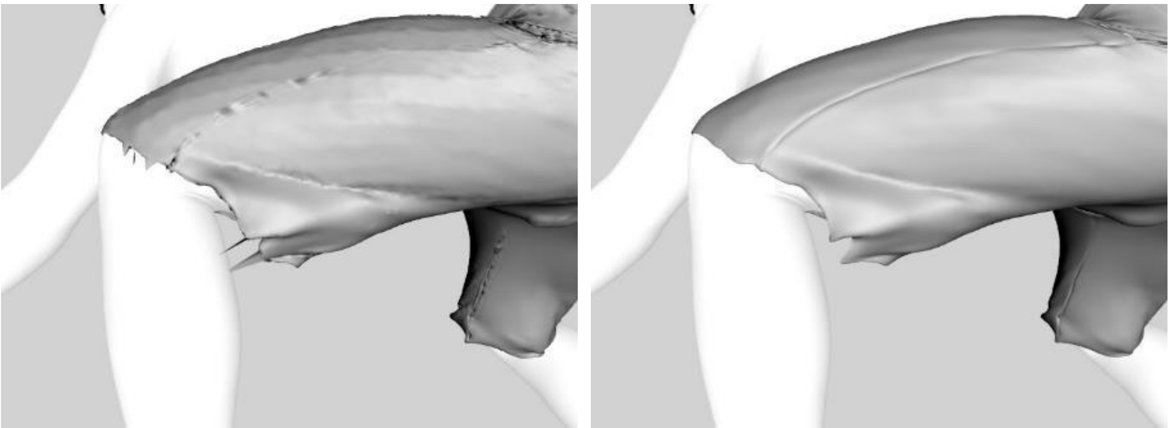
a. The first shorts I looked at are from Beach Vibes Outfit for Genesis 3 Male(s). There are five surfaces: Backpockets, BeltLoops, Button, Shorts, WaistBand.

b. Add a dForce dynamic modifier.

c. For all the surfaces except Shorts, set Dynamic Strength to 0.

d. Set Stretch Stiffness to 0.3 and Bend Stiffness to 0.2.

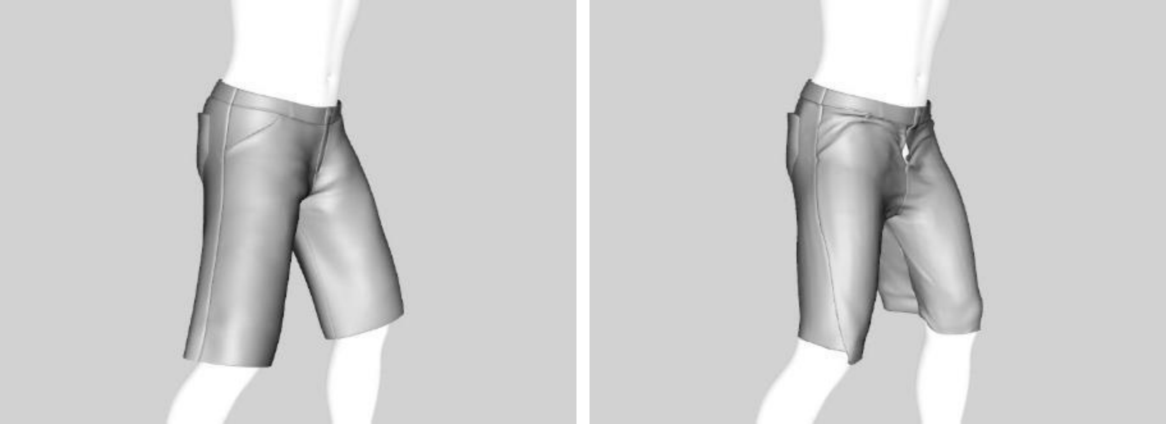
e. Run a simulation. I found that some vertices at the leg opening "stuck" and led to a few elongated polygons stuck to the leg (left image). Turning smoothing on for the shorts in the Parameters pane unsticks them but you can still see the points (right image), it's just not attached to the leg anymore.



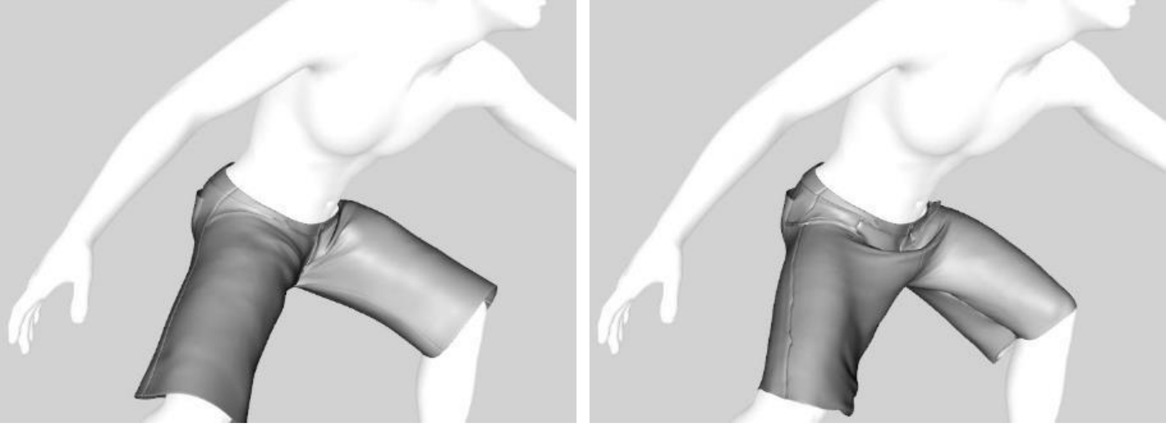
f. To correct this, add a static dForce modifier to the G3M figure and set friction to 0 and set friction to 0.1 on the shorts surface. This will allow the fabric to slide more smoothly. You can also set Shear Stiffness to 0.55 to counteract the cloth being wrapped around the left knee. Either one alone seems to work but I applied both in the following simulation results.

g. Here's the results comparing no dForce on left and dForce on right.

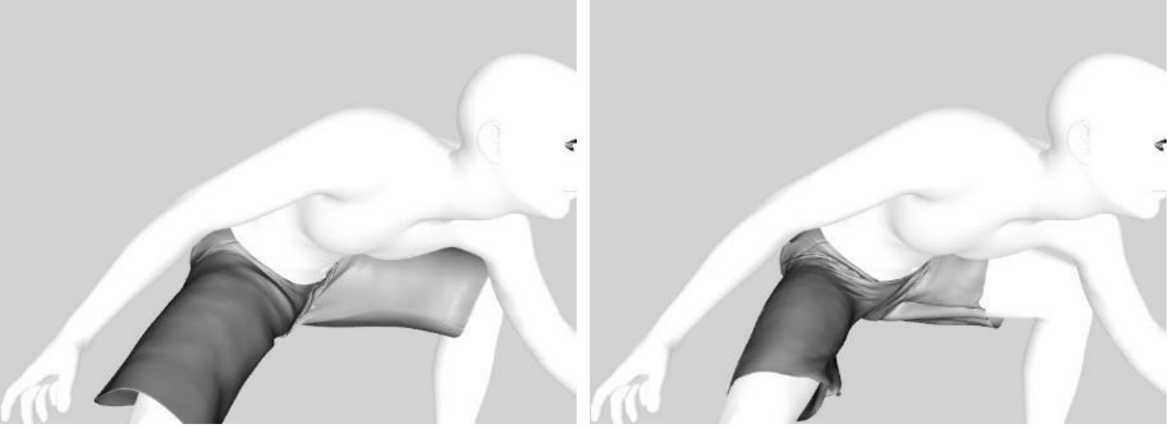
Frame 4:



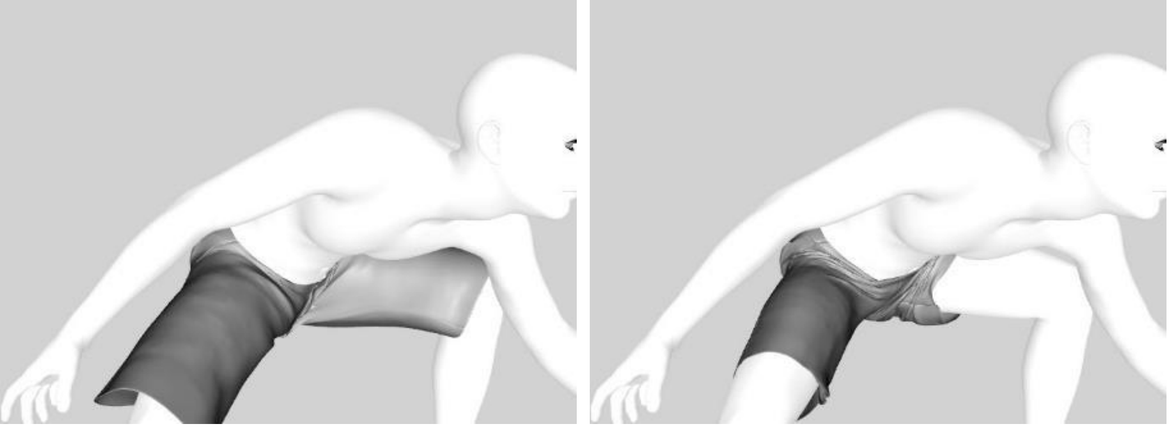
Frame 8:



Frame 12: the dForce version show the fabric being pulled by the leg movement

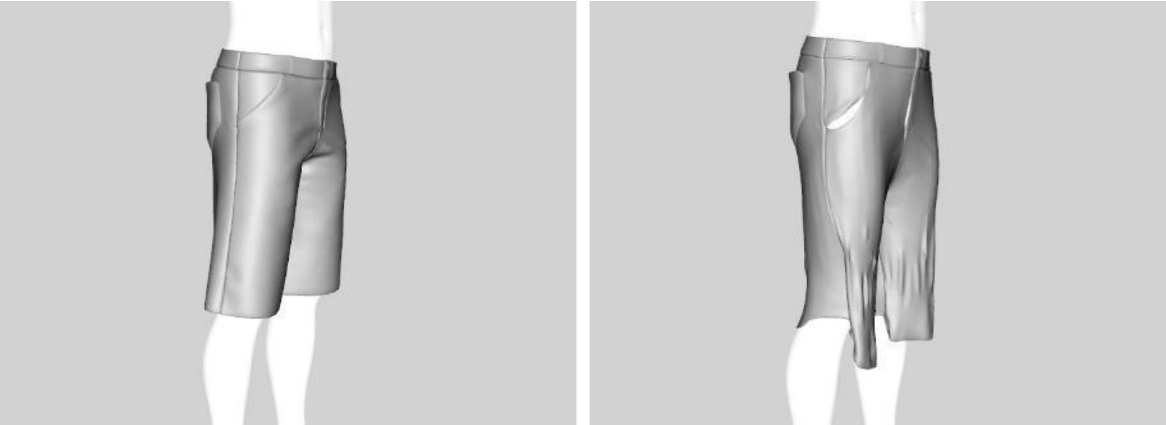


Frame 16: note how the left leg has pulled up towards the hip; this is caused by momentum from the figure’s posing. I would use frame 12 which was where the pose was set.

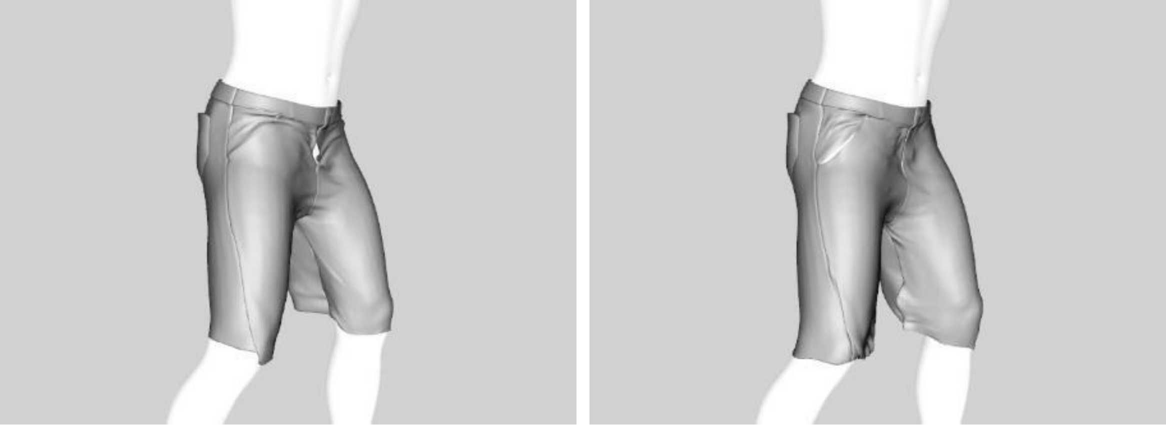


h. Since frame 16 showed the left leg of the shorts gathering at the hip, you can counteract this by setting Shear Stiffness to 0. This actually will cause the fabric to elongate which hurts the appearance in earlier frames but helps in later frames. Here’s a comparison of 0.55 Shear Stiffness (what was used above) on left and 0.001 Shear Stiffness on right.

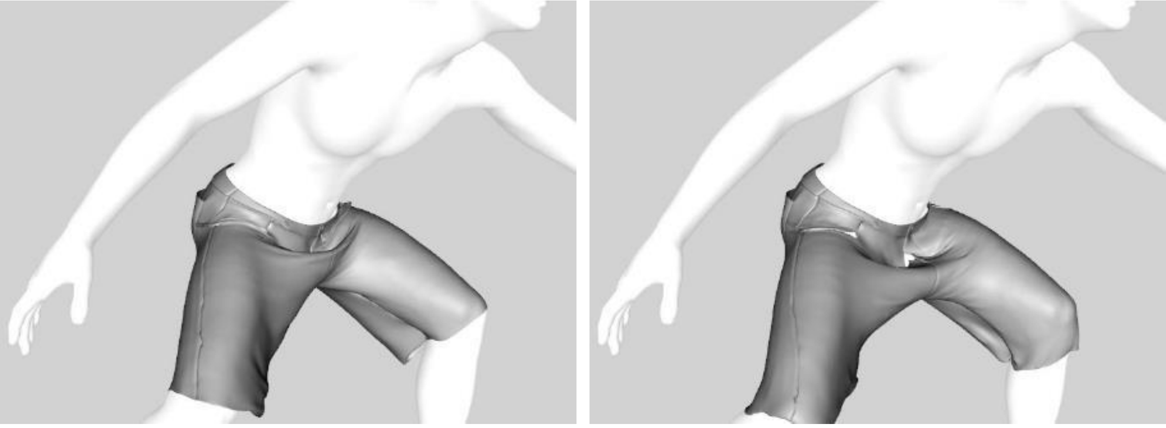
Frame 1: note the elongation with 0 Shear Stiffness. Note how some gaps appear at the pocket and the zipper.



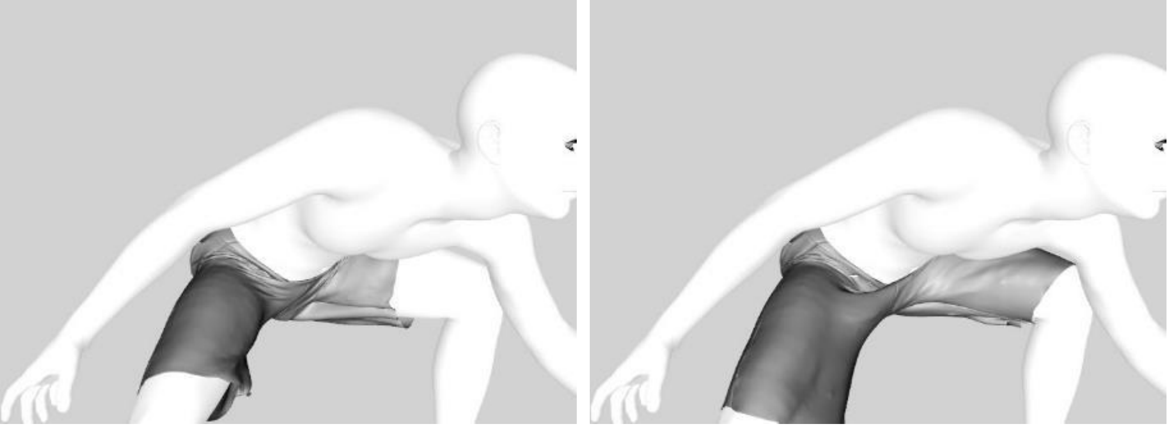
Frame 4:



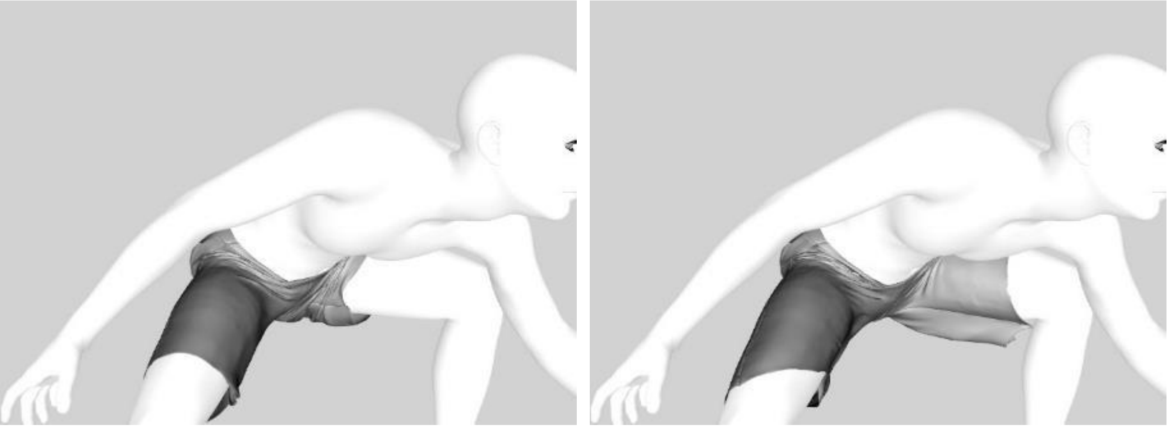
Frame 8: the lower Shear Stiffness keeps more of the cloth over the knee but also leads to bigger gaps and more stretch between the legs



Frame 12: the shorts remain longer as this style should



Frame 16: the zero Shear Stiffness doesn’t show the gathering by the hip



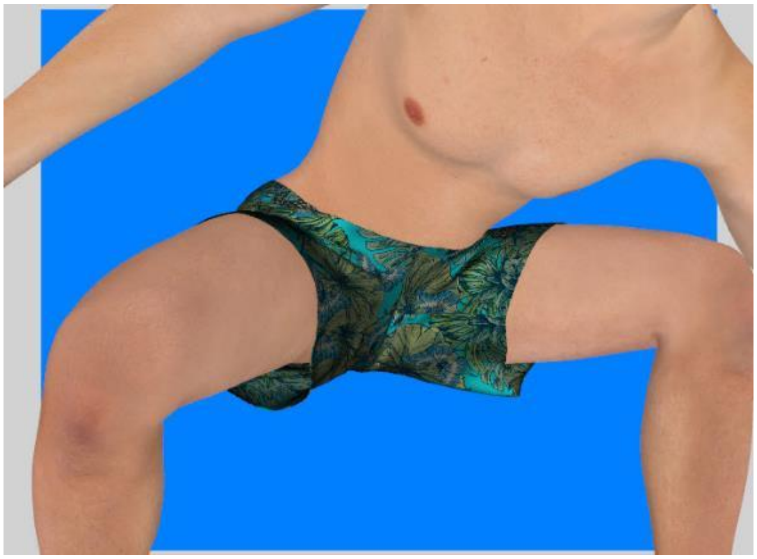
The moral of the story is the pose and posing dynamics (i.e., speed of change in the pose) will affect the results you obtain.

i. Instead of shorts, here’s some swim trunks. These are from [H&C Rash Guard Outfit for Genesis 3 Male](). Add the dynamic dForce modifier and set Stretch Stiffness to 0.3 and Bend Stiffness to 0.2 as we did for the underwear and shorts.

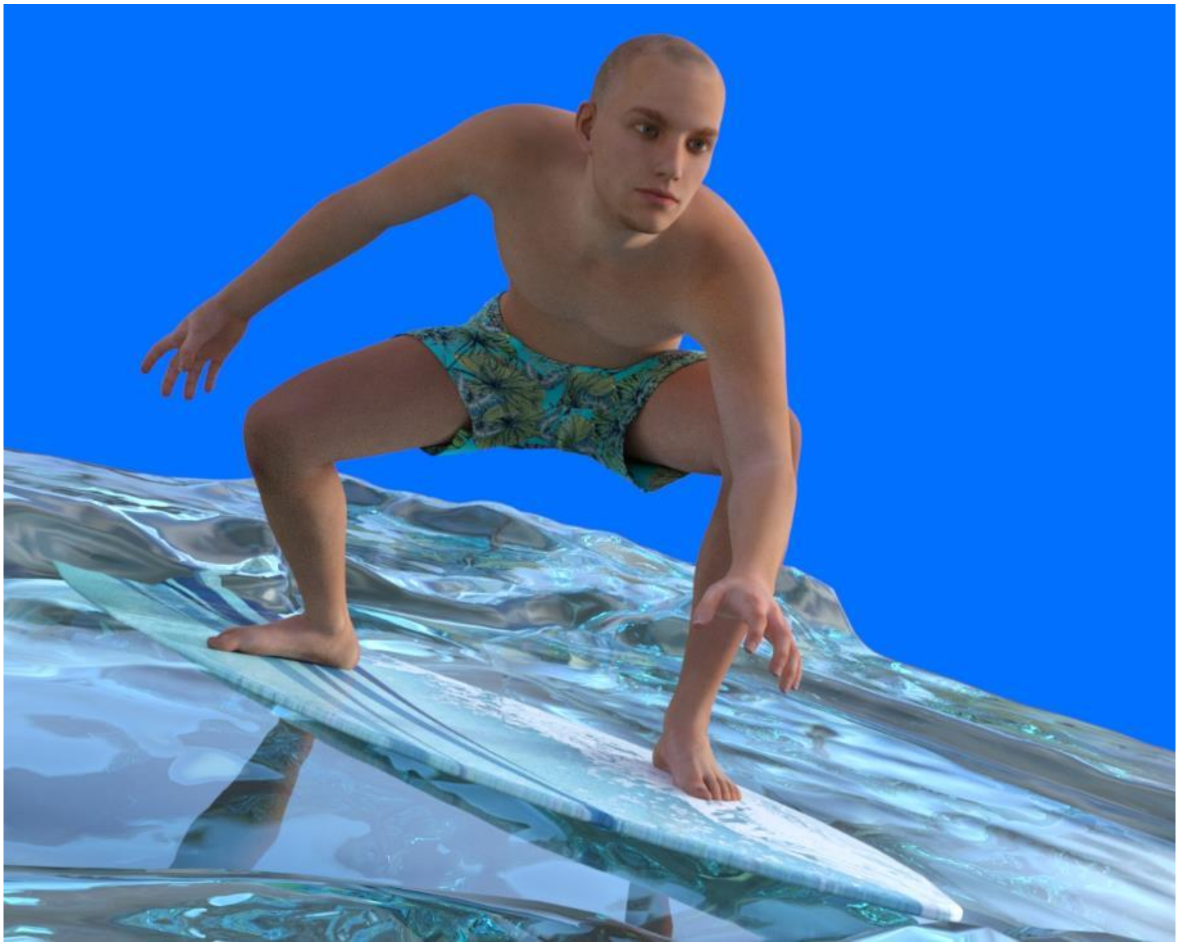
j. Using the same pose, you get something that looks like this at frame 16:

No dForce: baggier, a gap near crotch

dForce: more pulled in as you would expect from the legs spreading, right leg shows fabric hanging open rather than stuck up near leg



Rendered scene:



67. More Momentum Transfer. In section 63 I talked about momentum transfer using a torus. I have subsequently found that of the various primitives available (spheres, cylinders, cones, cubes) only the torus seems to show this behavior (perhaps I am missing something on the other ones?). I concluded section 63 with the words “What is this good for? I don’t know but it suggests there may be a way to get dForce objects to bounce off surfaces.” With just the torus as a possibility, use of this behavior seemed limited.

Praxis built upon my Daz forum post by showing a bouncing ball. Besides the bouncing ball, he has shown the ball being hit and bouncing around inside a box or off a vertical plane. To accomplish this he used the new Daz Studio beta 4.11 and a polyline mesh he modeled. Polylines are a new type of geometry coming to DS and can be used as a dForce add-on. His clever use of a polyline add-on is shared here.

Since my original post, I have found a way to use the torus as a helper object for other items. Create a rigid follow node on the inside of the torus, then after simulation, load a prop and parent it to the rigid node, hide the torus, and now the prop will assume the animated path the torus took.

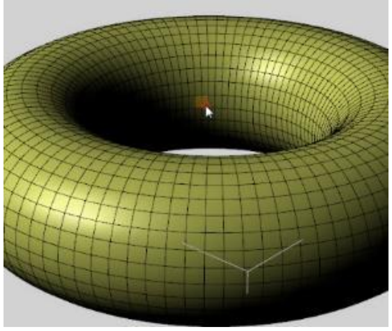
Here’s how to go about it:

a. First follow the steps in section 63.

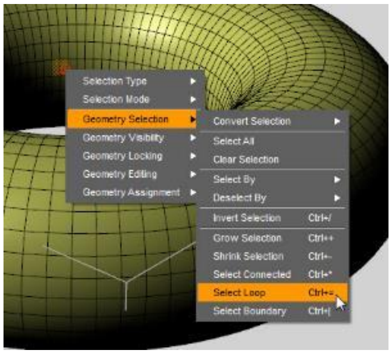
b. Run a simulation to test that the torus works as intended.

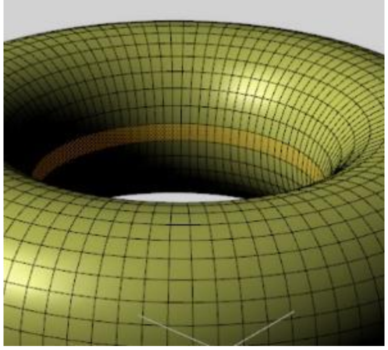
c. Clear the simulation. Select the torus in the Scene pane. Go to the Tool Settings pane and choose the Geometry Editor.

d. Right click in the viewport and choose Polygon as the Selection Type and Drag as the Selection Mode. Drag across two polygons on an inside loop of the torus midway down the ring.

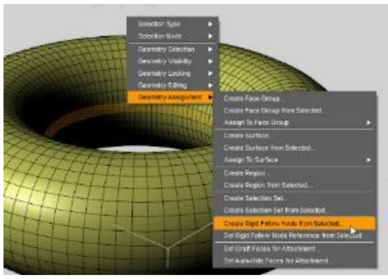


e. With the selection still active, right click in the viewport again and choose Geometry Selection/Select Loop.





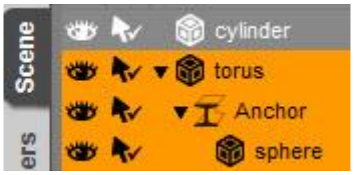
f. Right click in the viewport and choose Geometry Assignment/Create Rigid Follow Node and give it a name (I used Anchor). You will now see an I-beam icon parented to the torus. You can now parent other objects to the I-beam and they will follow the movement of the torus.


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...

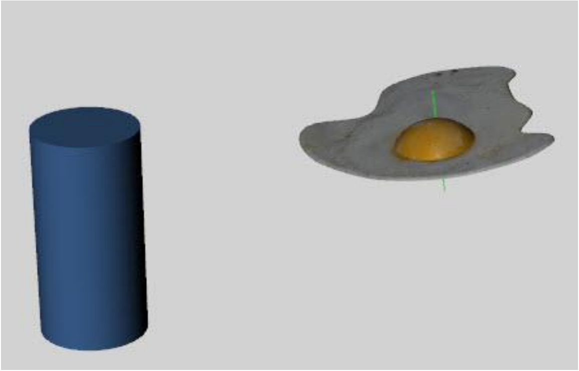
You can place the rigid node anywhere you want on the torus by selecting different polygons, but the inside loop will have the added prop follow the general translation and rotation of the torus.

g. Rerun the animated dForce simulation.

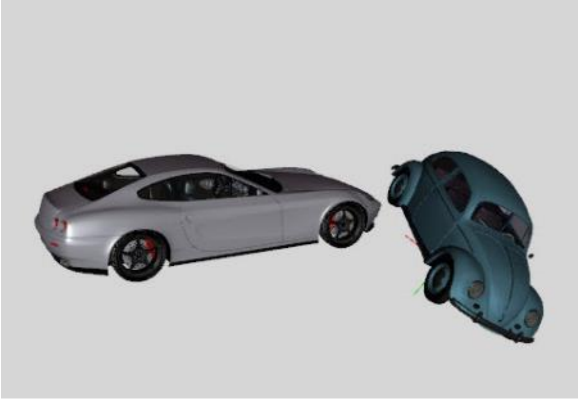
h. Create a sphere and parent it to the I-beam. Hide the torus and run the animation. Now instead of a torus moving, you will see a ball moving.



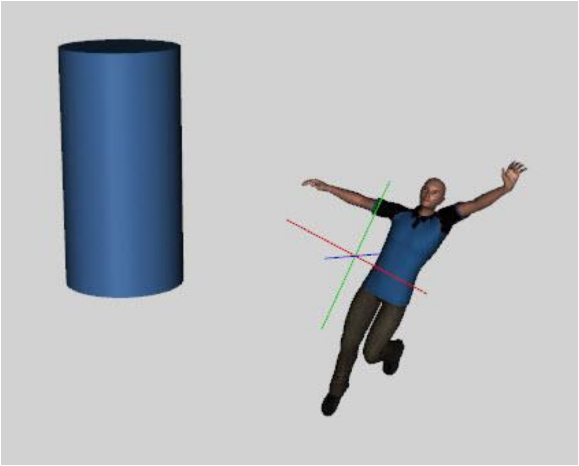
i. What else can be used? How about a fried egg?



An automobile skidding on the road? (Two autos, one parented to the cylinder, the other to the torus, cylinder and torus hidden.)

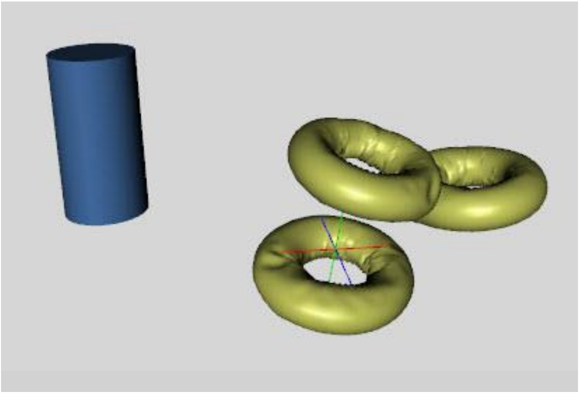


A man flying through the air?



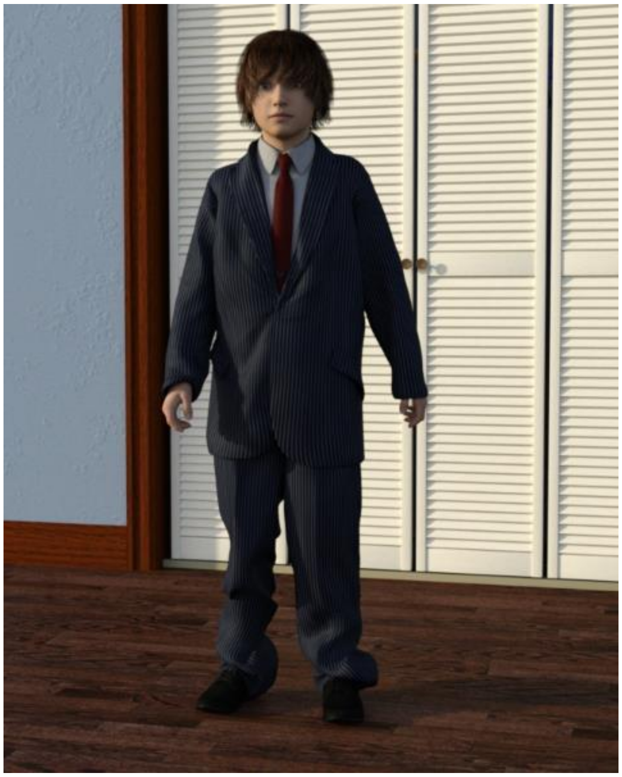
The possibilities are endless.

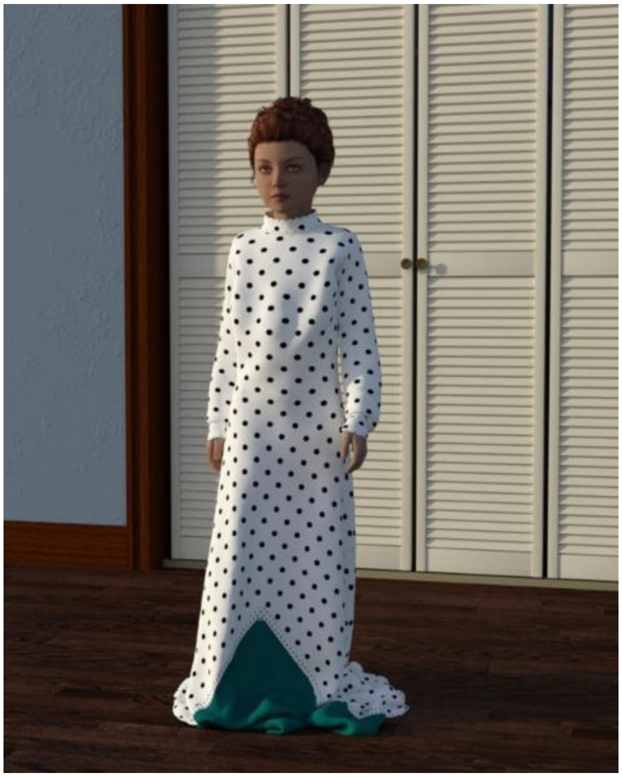
j. You can simulate more than 1 torus. Here are three set to collide with each other. Since the second two are at rest, it takes some time for them to receive the energy and begin to move.



k. These simulations/animations are not by any means perfected but rather illustrate what may be possible. DS also seems to insert some random 180 degree rotations along the way where the item flips around.

68. Loose and Rumpled Clothing. Barbult posted to the Daz Forum an interesting result she observed while running a dForce simulation with an adult figure morphing into a young growing up figure. The result was oversized clothing that made it look like the children were playing dress up. Her first example is here and the second is here. I've replicated the result so I can show a couple of examples:





I'm a little surprised the clothing didn't autofit but it doesn't during or after the dForce simulation. It's a stable result unless you clear the simulation when the child is small, then the clothing conforms as normal and you end up with the typical look of a child in children's clothing.

I wondered if you could use smaller scale changes to give a looser fit to clothing. I find 3D clothing to be a little too perfect many times. Below I explain how you can indeed loosen the fit or even make for a rumpled look by scale changes during the simulation.

a. Set up a scene with a standard figure and clothing item. I used G2F and the dress from Early Spring Outfit. Add a dynamic modifier to the dress.

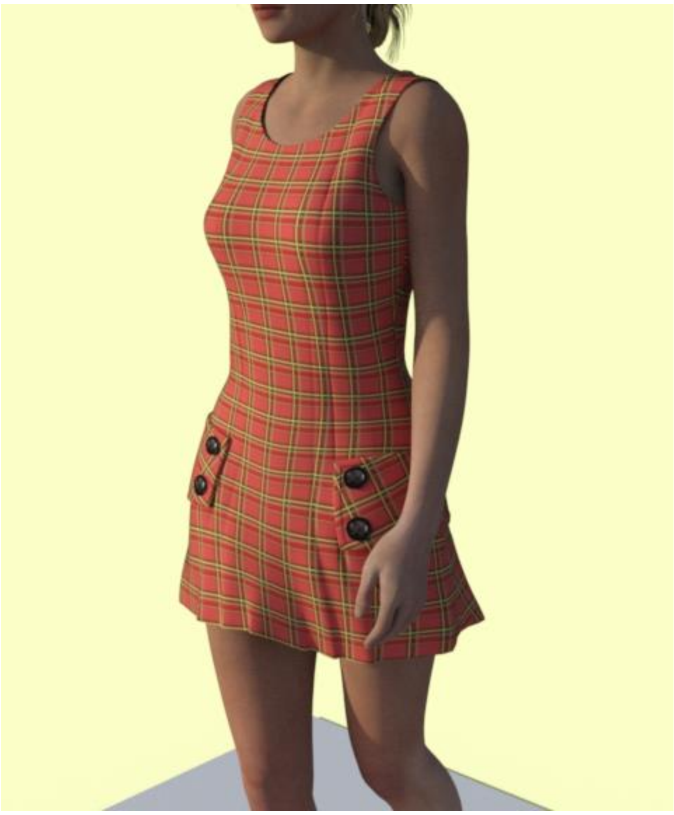
b. On the surfaces pane, select all five surfaces of the dress and set Dynamics Strength to 0. Then set the MainDress surface value back to 1. On the same surface, set Bend Strength to 0.2, Buckling Ratio to 55% and Density to 20.

c. Start with the T-pose at frame 0. At frame 30 pick a new pose (I used a walking pose).

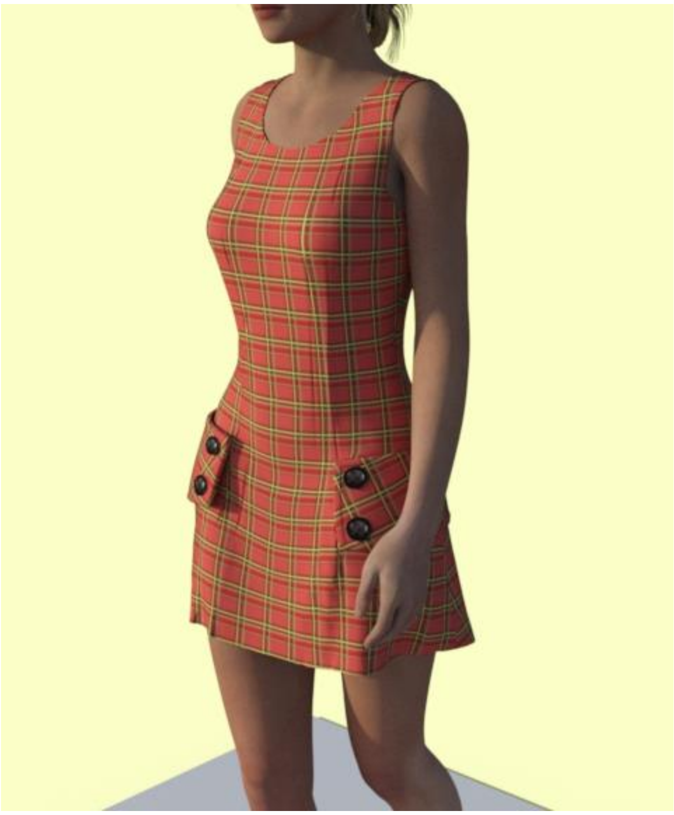
d. Change the figure's scale (not the dress) to 105% at frame 0. Set it back to 100% at frame 30. Run the simulation and render. Repeat using different combinations of scale. To change X, Y, and Z-scale of figure, use the Parameters pane submenu (the stack of papers icon at a top corner) and check Show Hidden Properties.

e. The following series of images show the results you get.

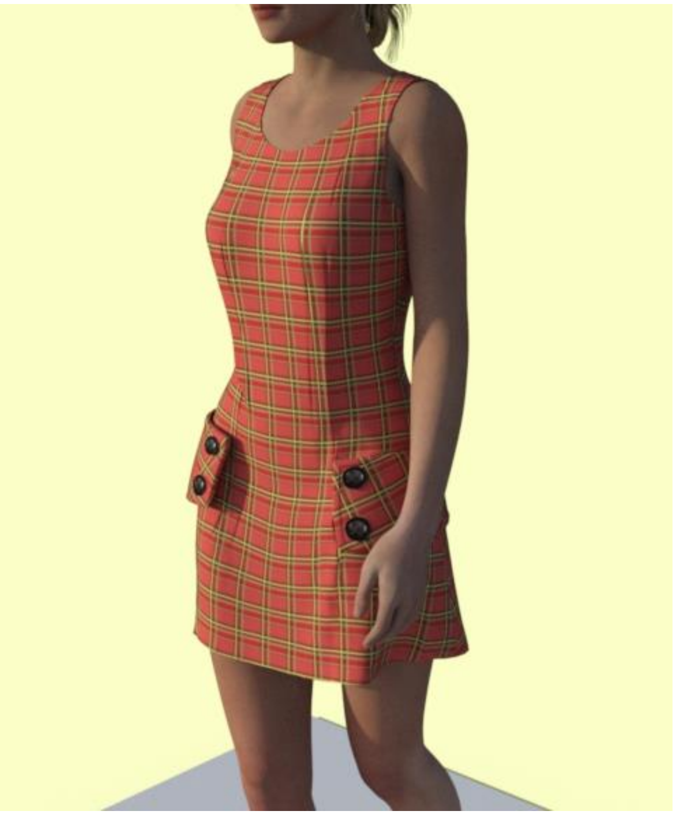
(1) Normal conforming result with no scaling changes and no dForce simulation:



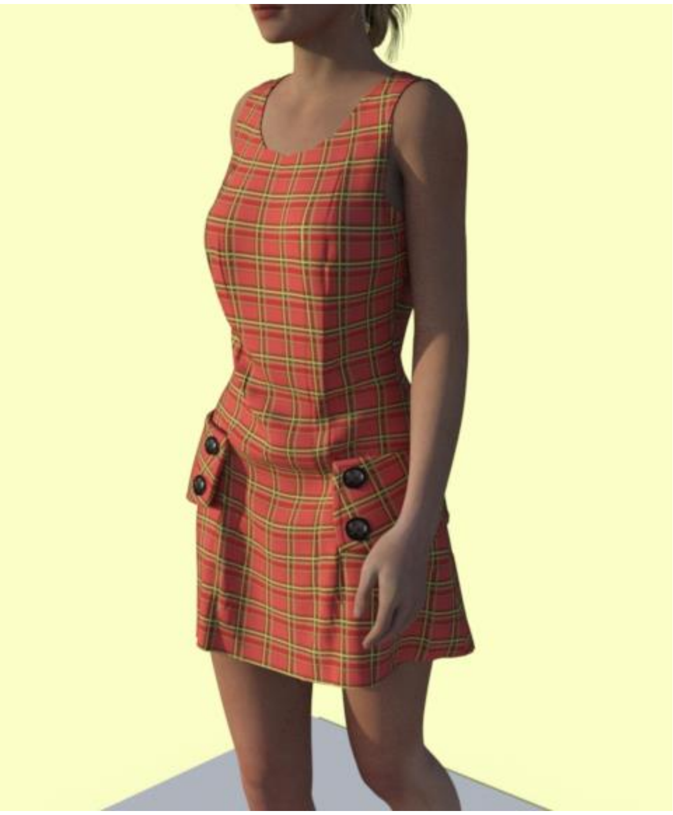
(2) Scale at 100% throughout. This shows the result from normal dForce simulation using the parameters mentioned in step b.



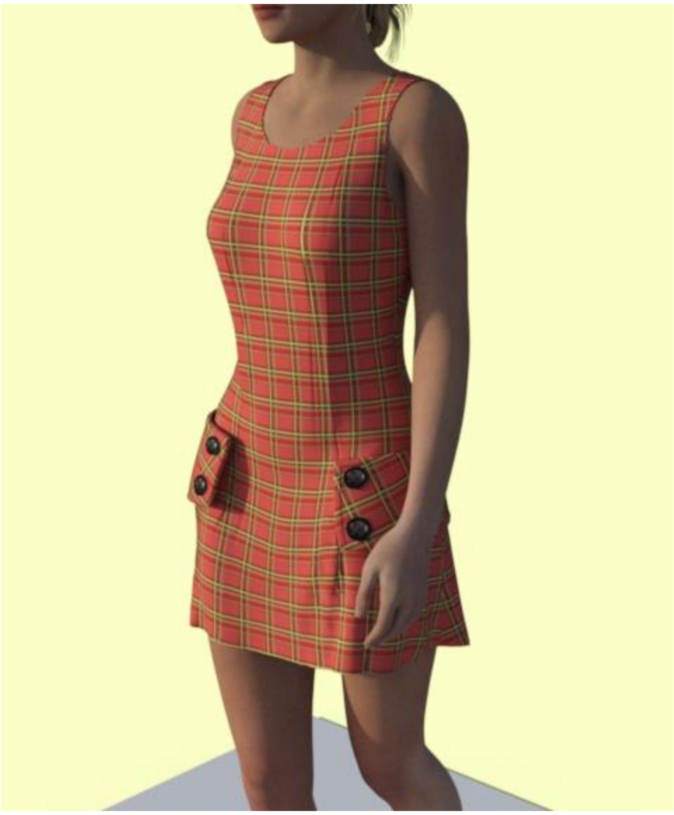
(3) Scale starts at 102%. Not much difference.



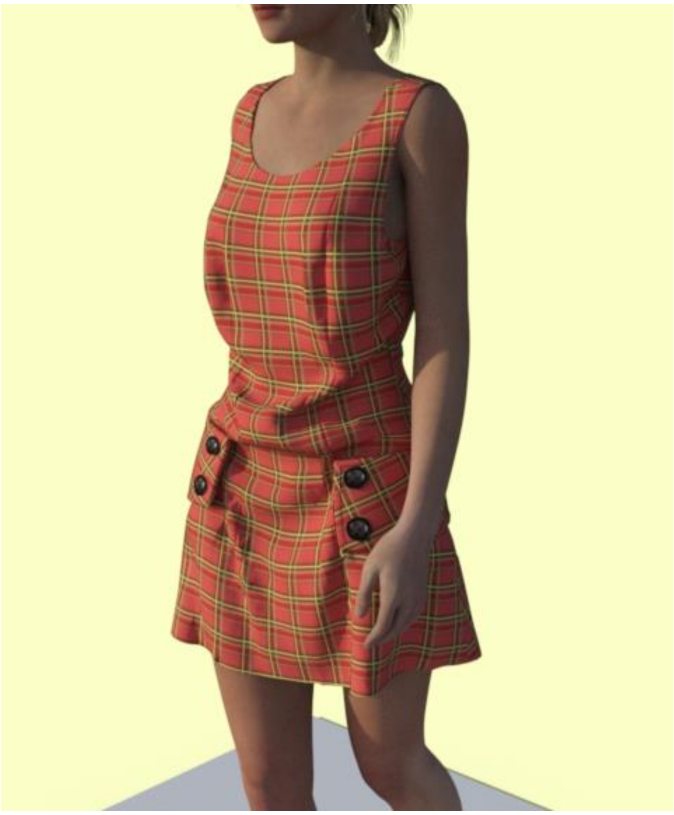
(4) Scale starts at 105%. Note the looser fit under the bodice, the bulges across the waist, and indentation of the panel at the front between the legs.



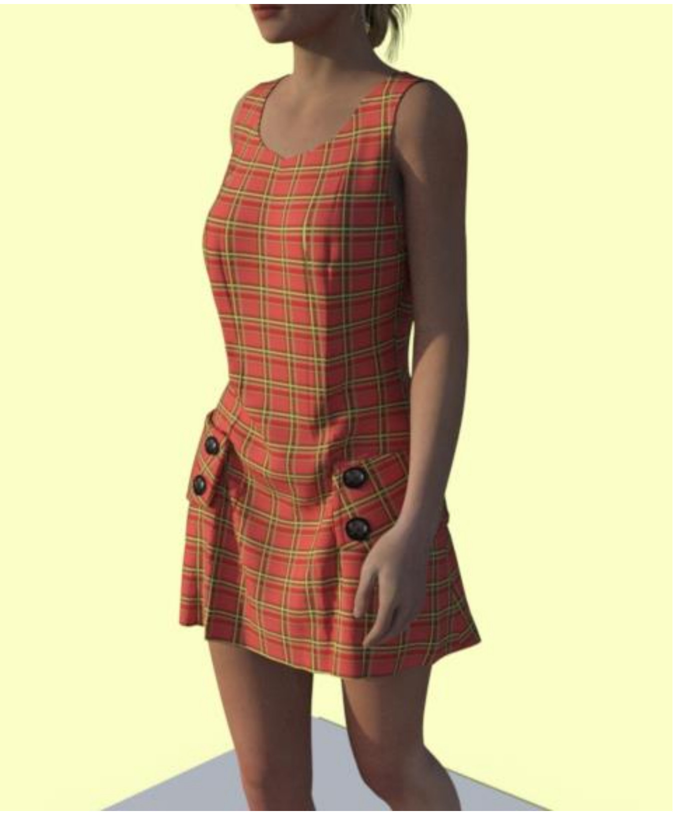
(5) Scale starts at 105% for X and Z, not Y. This was done as Y-scaling results in a longer dress than the starting version. The results is a look similar to dForce at 100% throughout (image 2) so most of the looseness came from changes in Y-scale.



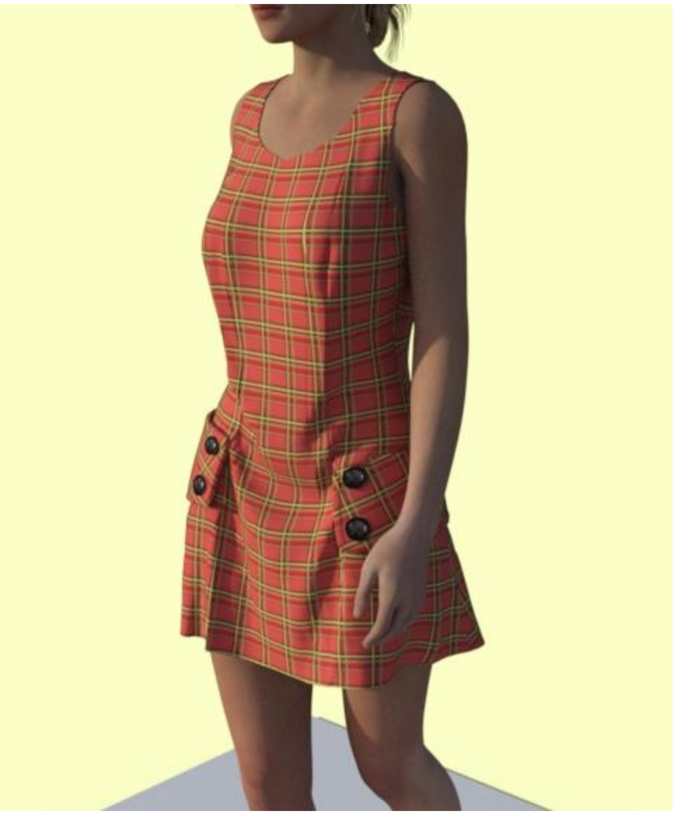
(6) Scale starts at 110%. Much looser, more rumpled.



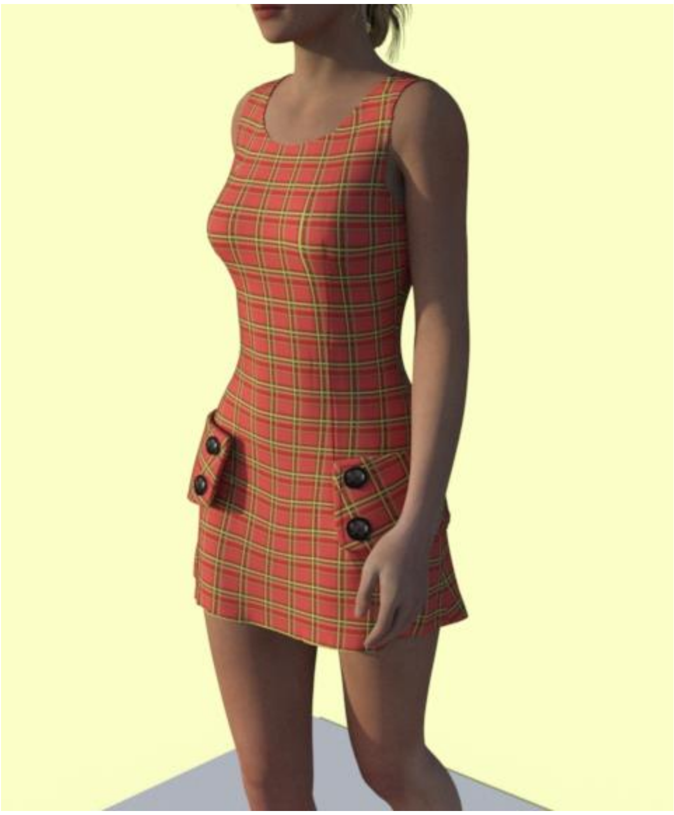
(7) Scale starts at 110% for X and Z, not Y. Now we see more variation than we did using the same approach at 105% scale (image5).



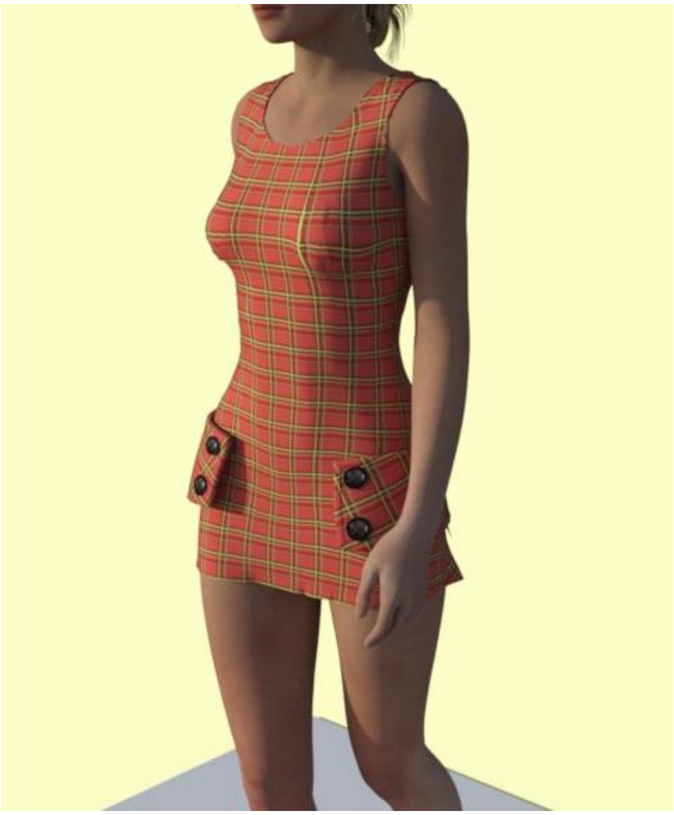
(8) Scale starts at 110% for X and Z, 102% for Y.



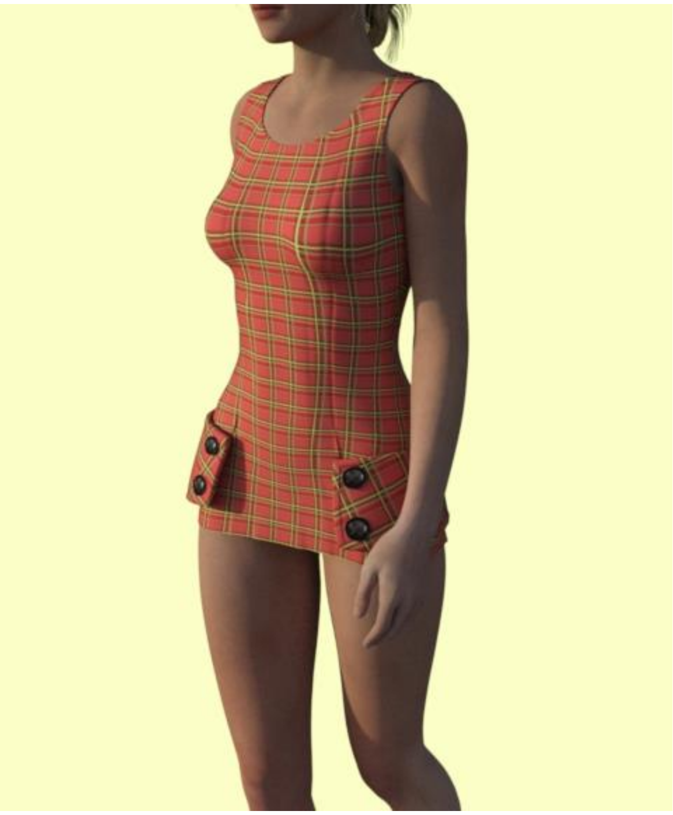
(9) Scale starts at 95%. You get a very taut looking result.



(10) Scale starting at 85%. Dress becomes mini-sized.



(11) And even smaller at 72% scale to start. At 50% it was more a shirt than dress.



f. In conclusion, you can using a reduction in scaling during a simulation to give a loose to rumpled look to clothing. The key driver is change in Y-scale but that also results in a slightly longer clothing item. You can get a shorter result by starting below 100% in scale and going large.

69. **dForce Magnets**: This is a new product that was recently released in the Daz store. I've given it a try and find it a useful addition to the dForce toolkit.

   a. The magnet is a small geometric shape with a dForce add-on modifier. First select a clothing item or create a primitive like a plane. Use the geometry editor to select a single vertex on mesh then run the magnet script. The magnet is added to your scene as a child of the selected item. You animate the magnet as you would any other item. Run the dForce simulation and you will find the magnet pulls the clothing or object as if you were holding it between your thumb and index finger.

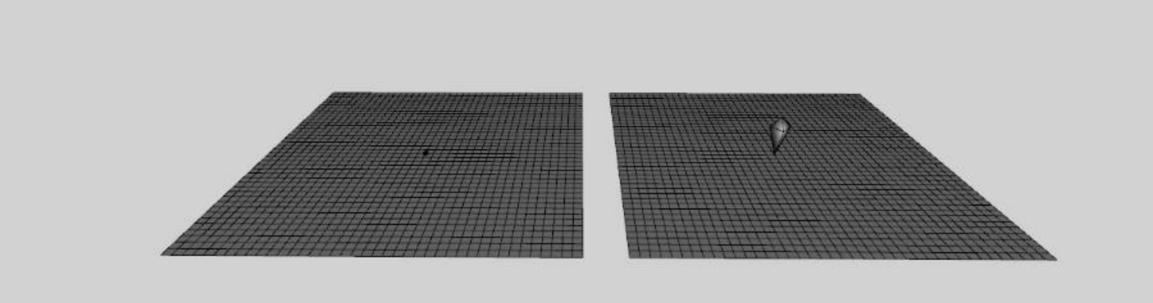
   b. The animated action is very similar to what you get using the helper objects I've described in several sections above. What's helpful about the magnet is you can select a single vertex. When trying to grab a single vertex with a primitive like a sphere, it can slip out during simulation. You typically need to use a larger object that covers several vertices which may then have a negative effect on the mesh it encloses.

c. The main advantages of magnets versus helper objects are:
   i. They can make tight pulls (like a strand of hair) or a pinch of cloth
   ii. You can select a single vertex. Helper objects need to grab several vertices otherwise they may slip away during simulation
   iii. It's easy to get accurate positioning as it will be focused on a single vertex. With a helper object, you have to visually adjust its location

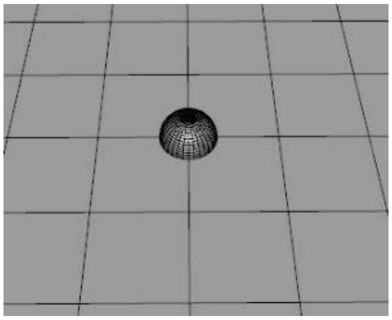
d. The main disadvantages of magnets versus helper objects are:
   i. The pinpoint grip when you want a broader grab. An easy solution is to parent a helper object to a magnet and extend the effect of the magnet
   ii. You cannot parent magnet(s) to anything else or group multiple magnets. If you do, they lose their connection(s) and you need to delete them and add new ones.

e. **Example 1** comparing magnets with helper objects. Set up two, 2-foot planes with 40 divisions. Add a dynamic modifier to each. Create a small sphere and center it on one plane. Do not parent the sphere to the plane. Pick the center vertex of the other plane with the Geometry Editor and add a dForce magnet. At frame 10, raise the sphere and magnet.

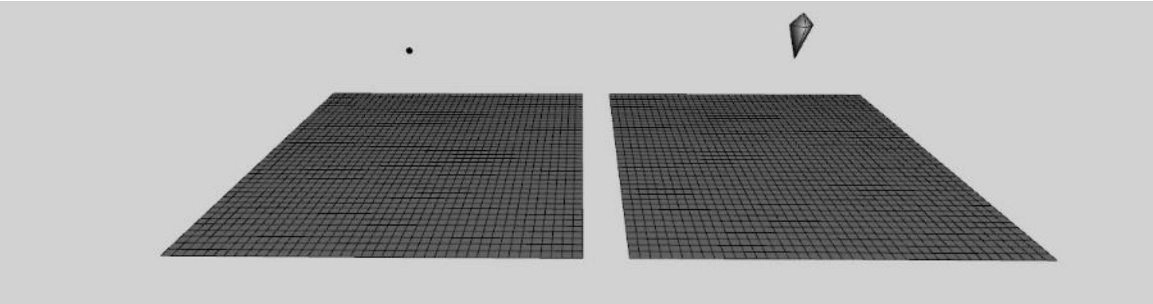
Frame 0 setup:



Sphere positioning:

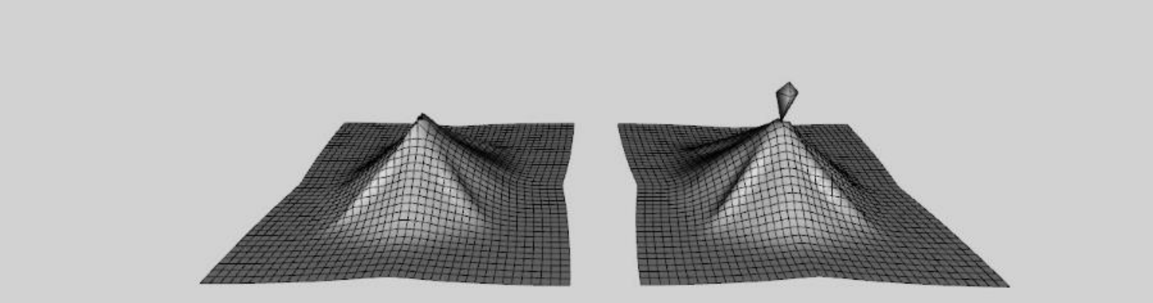


Frame 10 setup:

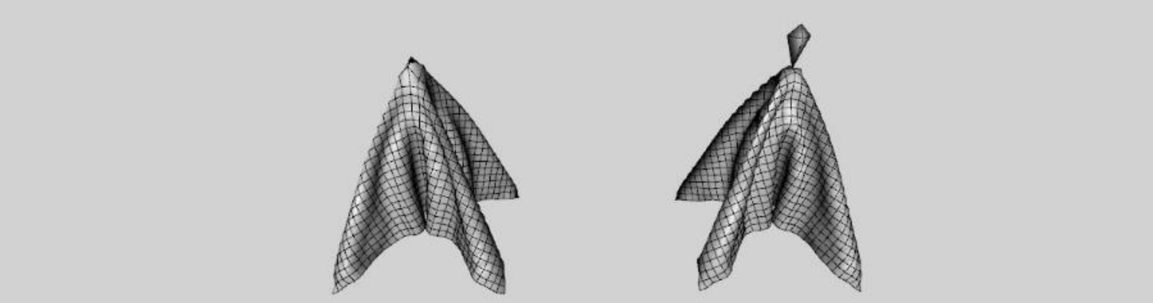


Run a 30 frame dForce simulation. Here are the results:

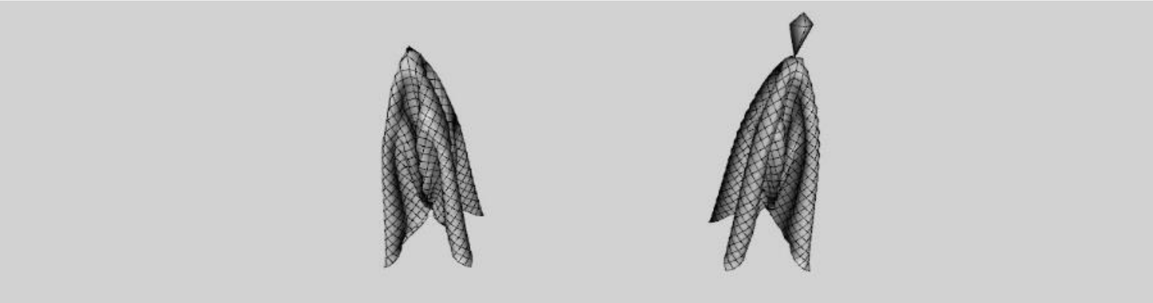
Frame 4:



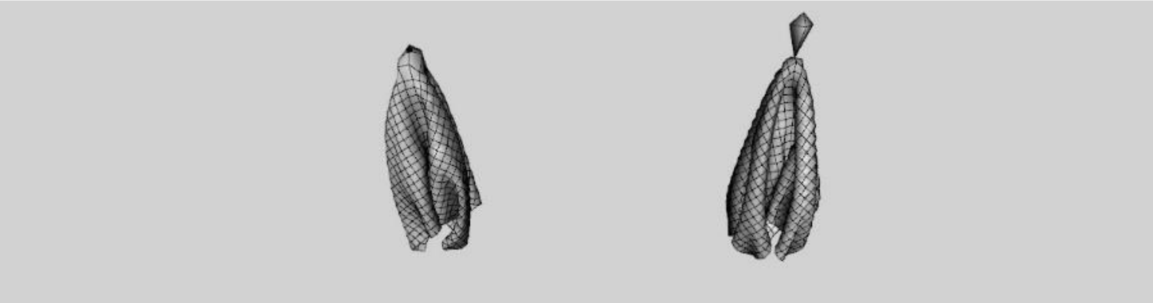
Frame 8:



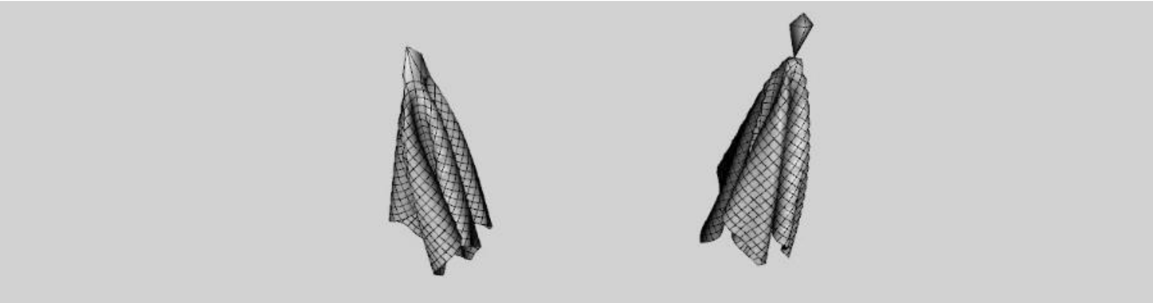
Frame 10:



Frame 16:



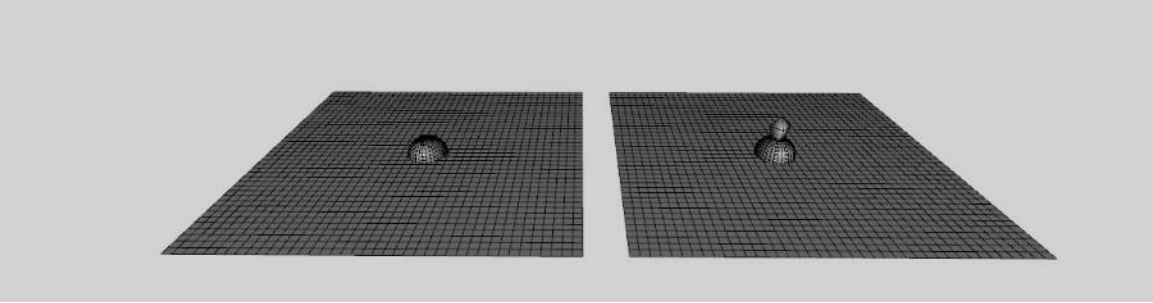
Frame 30:



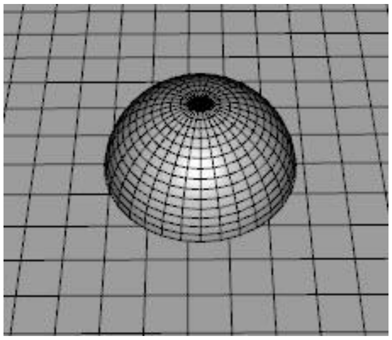
You get similar results using a helper sphere or a magnet, but note the stretched polygons at the top when using the sphere.

f. **Example 2**. Replace the small sphere with a larger sphere. Duplicate the sphere and parent the duplicate to the magnet.

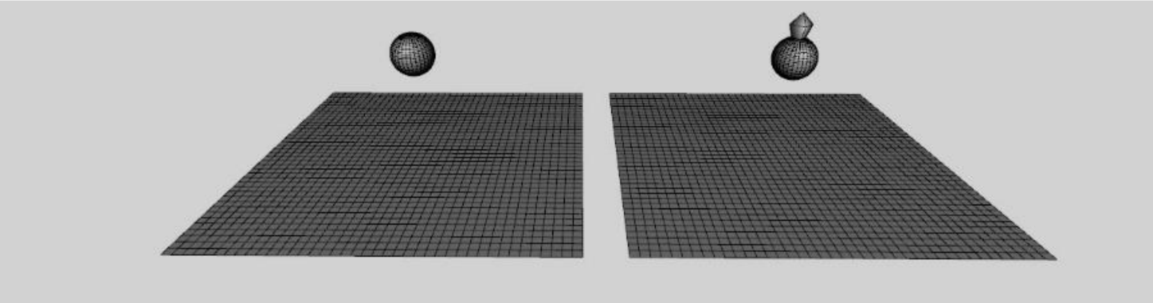
Frame 0 setup:



Sphere positioning:

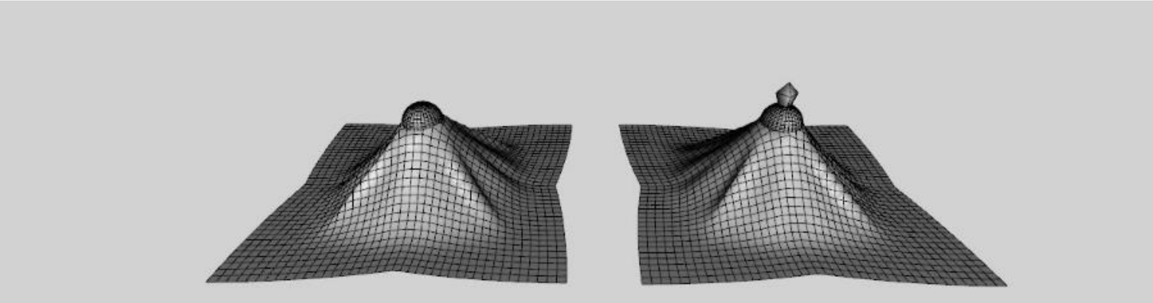


Frame 10 setup:

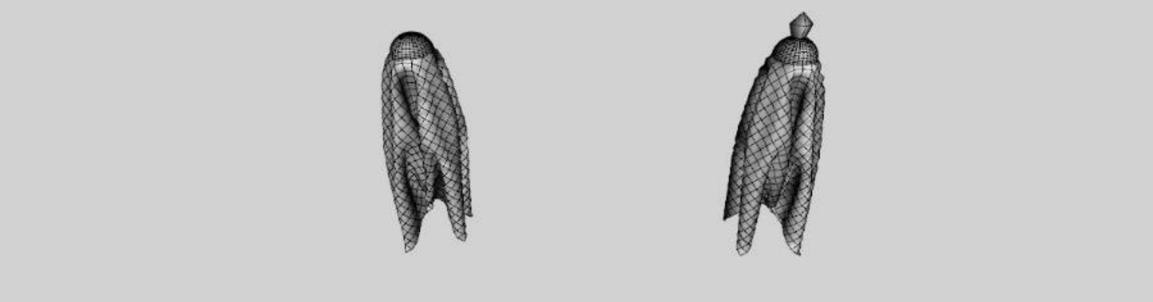


Run a 30 frame dForce simulation.

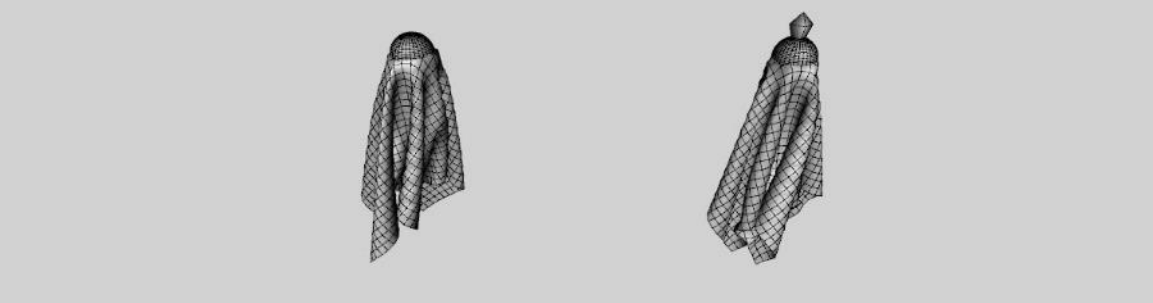
Frame 4:



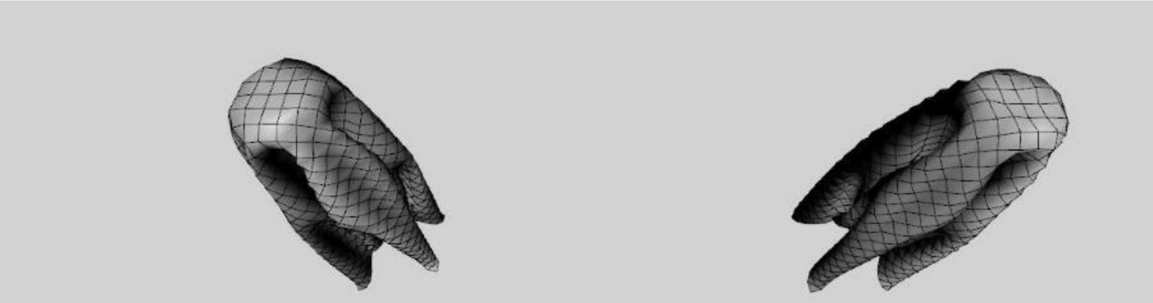
Frame 10:



Frame 30:



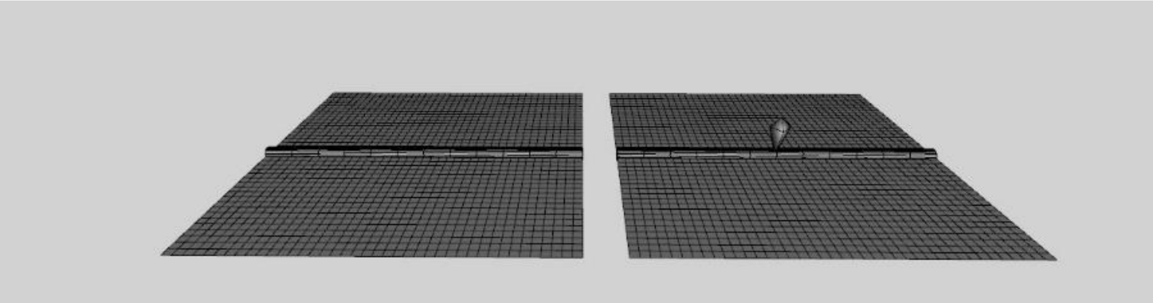
Spheres hidden:



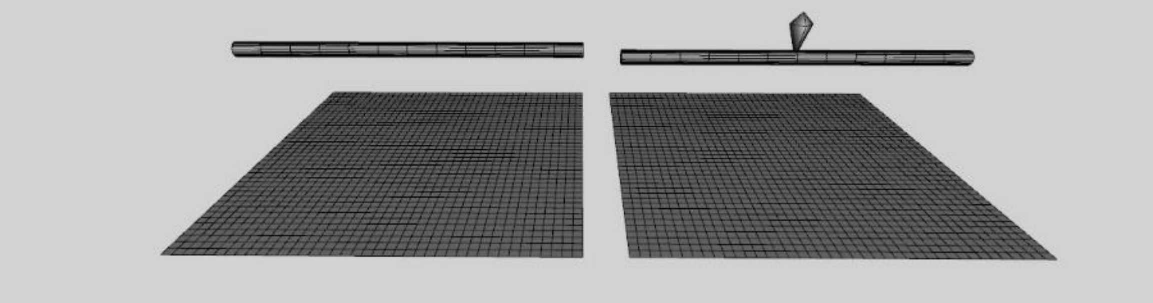
You get very similar results using a large sphere whether by itself or attached to a magnet.

g. **Example 3**. Create two, 2-foot long narrow cylinders. Parent one to the magnet and the other center on the plane. At frame 10, add keyframes to keep the cylinder's scale at 100%. At frame 20, shrink the cylinder to 35%.

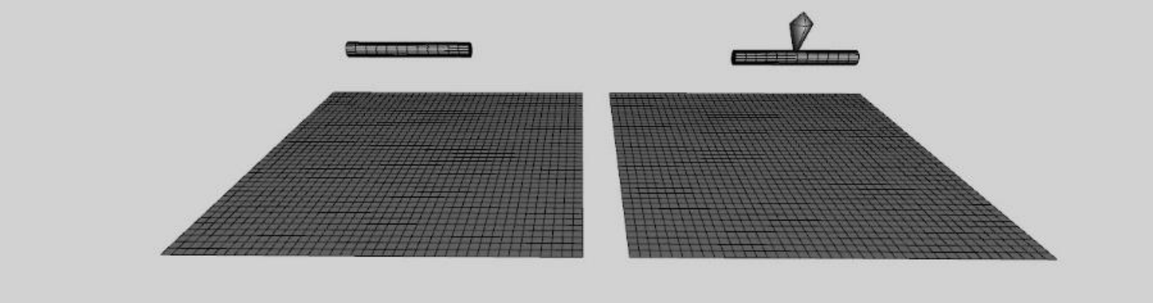
Frame 0 setup:



Frame 10 setup:

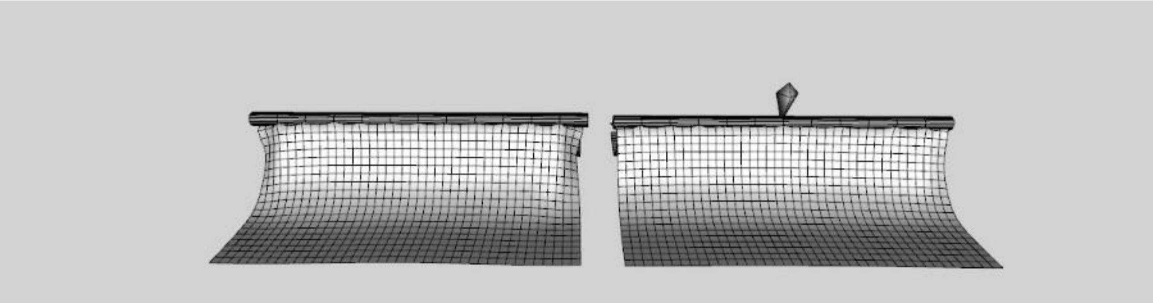


Frame 20 setup:

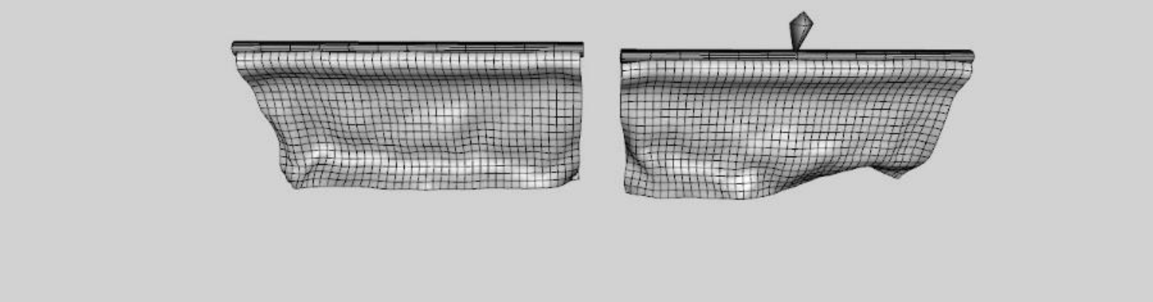


Run a 30 frame dForce simulation.

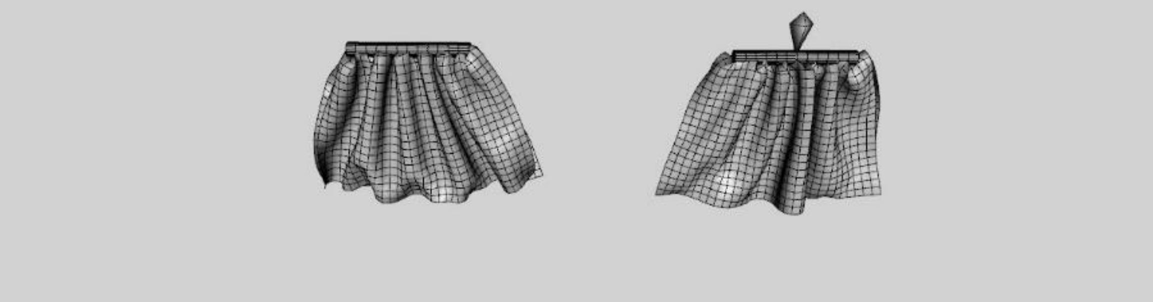
Frame 4:



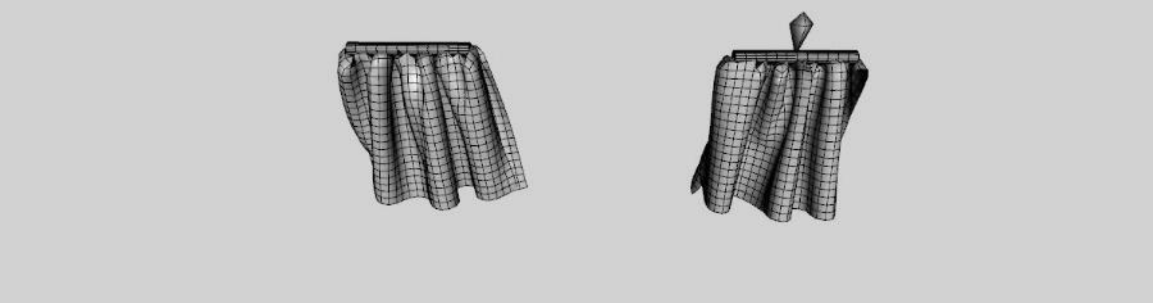
Frame 10:



Frame 20:



Frame 30:



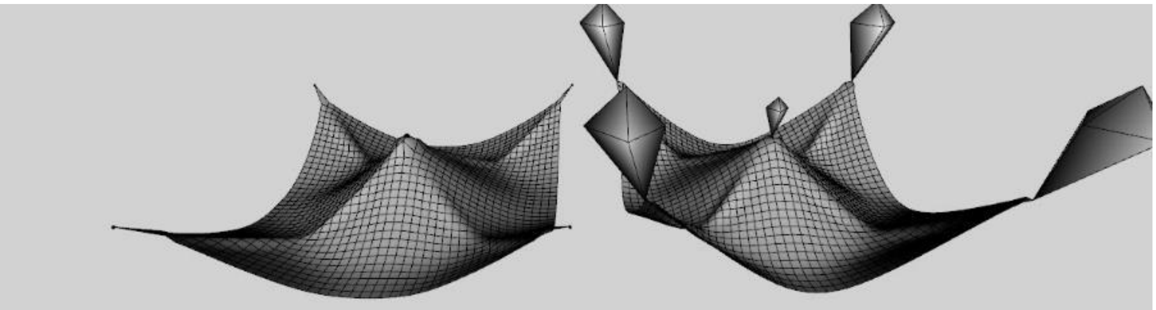
You get very similar results whether using the helper cylinder only or using the magnet with cylinder. You don't really need to magnet in this case, but using it does make for easier positioning of the cylinder.

h. **Example 4.** Add four magnets to each corner. You cannot duplicate or parent these. They need to be attached directly to the plane by selecting a vertex. Select the center sphere and type Ctrl+C. Select each of the other magnets and type Ctrl+V. This will copy the timeline actions to the 4 new magnets.

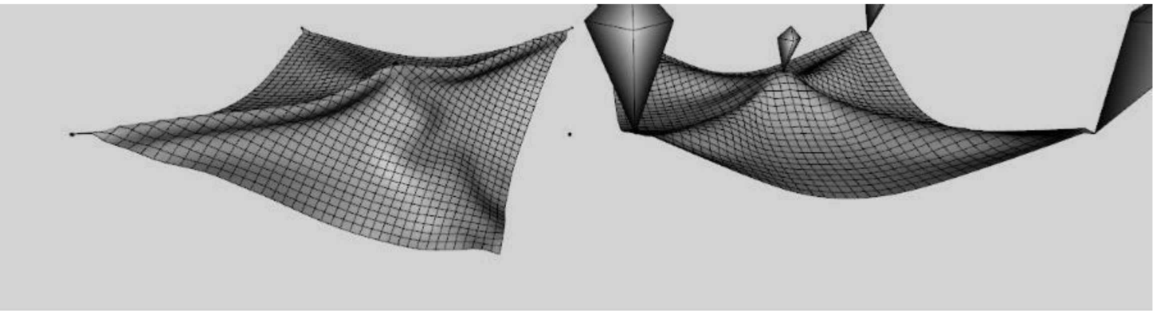
Duplicate the sphere 4 times and positon the new spheres at the corners of the plane.

Run the simulation.

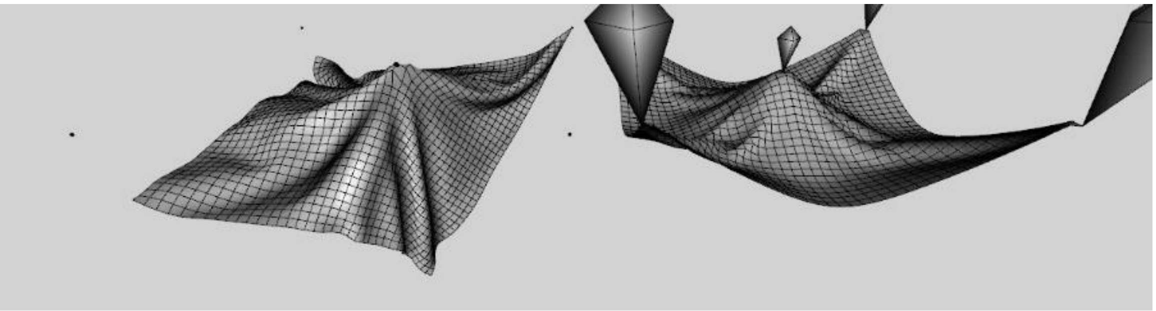
Frame 4:



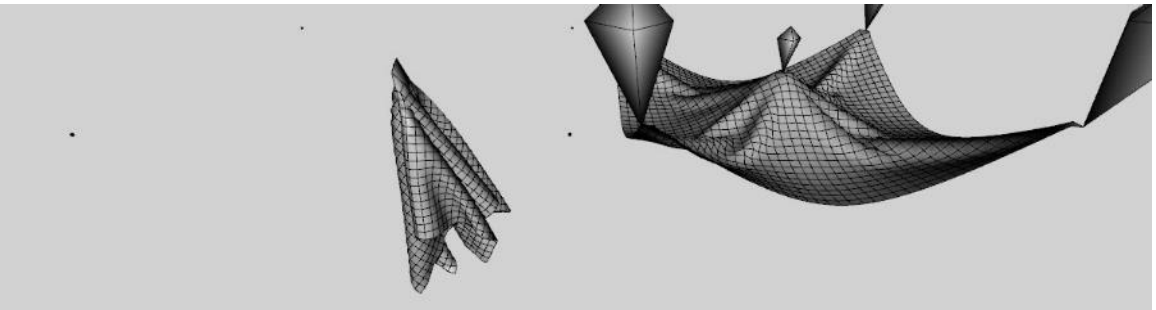
Frame 10:



Frame 16:



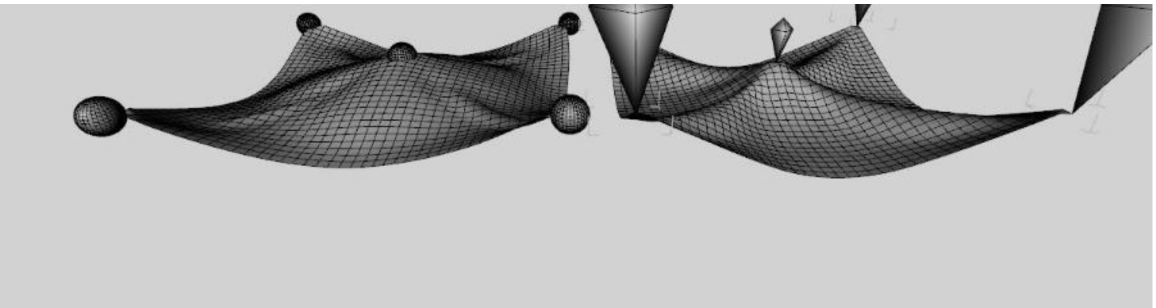
Frame 30:



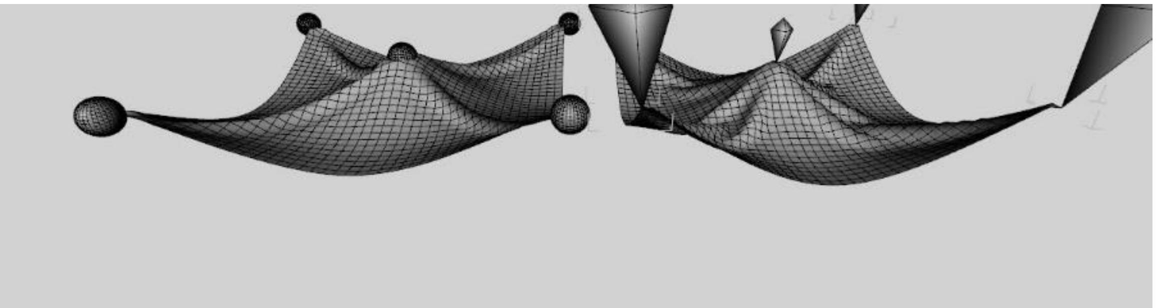
Note how the small spheres were no longer able to hang on and eventually the plane fell away from all 4 corner spheres. Much better to use the magnets.

You can make the spheres larger and in that case they hold:

Frame 10:

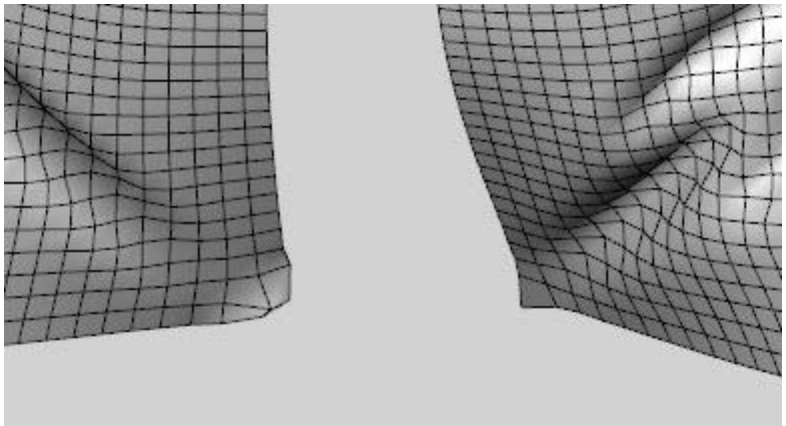


Frame 30:



Note the difference in draping, the magnets forming some additional ridges.

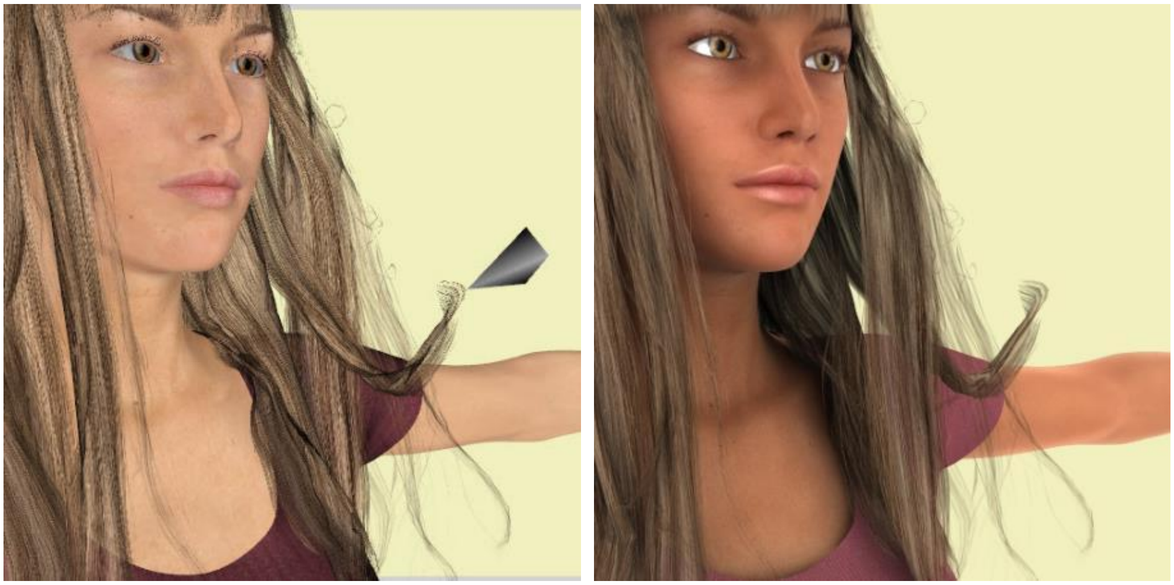
The larger spheres lead to some drooping of the tip (sphere was on the left, magnet was on the right).



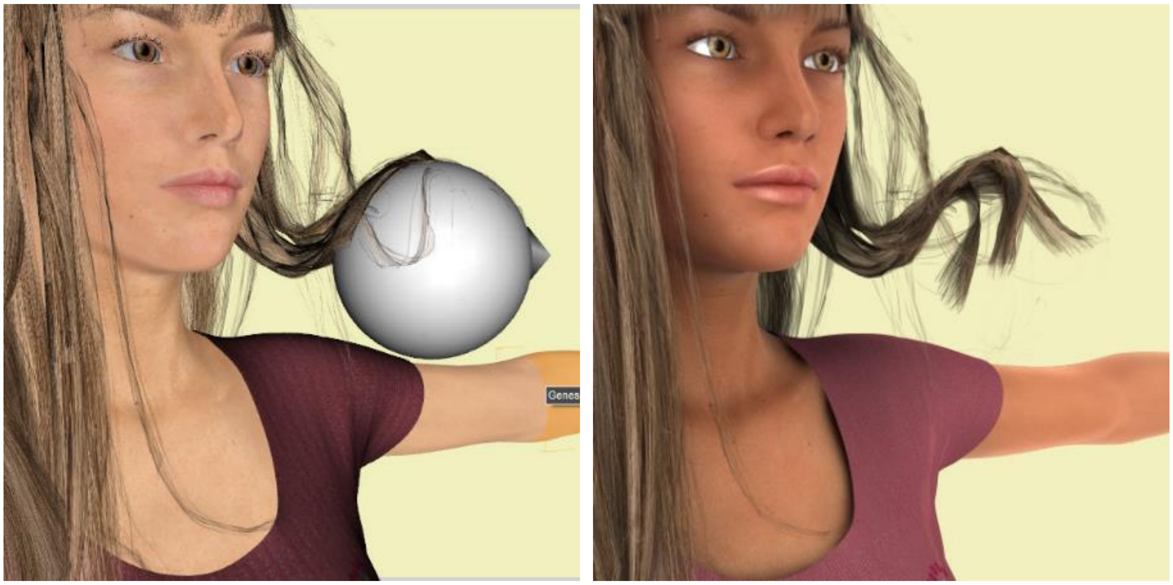
i. In my opinion, the dForce magnet product is a good addition to your dForce toolkit. Much of what they do you can accomplish with helper objects, but the ability to precisely attach them to a vertex of your choosing is quite handy. Plus magnets and helper objects work well together, especially if you want to extend the grip of the magnets.

j. **Example 5.** To illustrate the extended grip, here is an example on hair. To illustrate the extended grip, here is an example on hair. The first row shows the use of the magnet only, the second row shows using a sphere that was parented to the magnet, and the third shows a torus in use. The viewport screenshot at the left of each row shows the magnet and helper object, the image at the right shows a render with the helpers hidden. As you can see, a lot more hair is lifted with the helper objects.

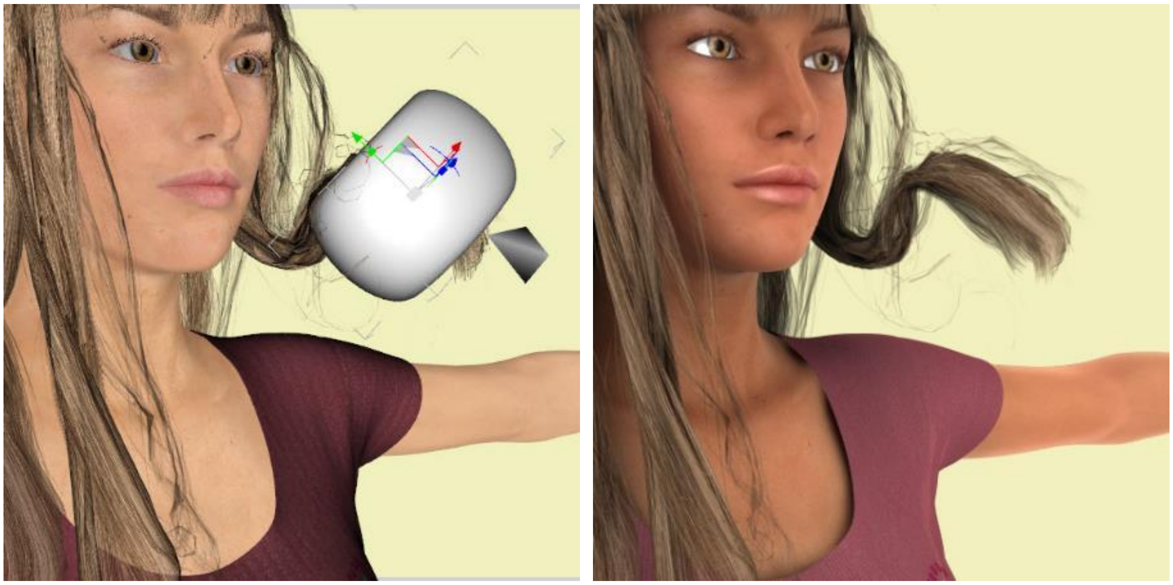
Magnet Only:



Sphere:



Torus:

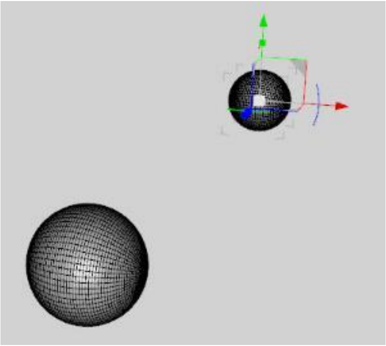


70. Making Caves. I was a big caver as a youth and still enjoy going in them today, so I'm always on the lookout for a way to model caves in 3D. A year and a half ago I posted a [tutorial on making a cave with push modifiers](), but a better solution is to use dForce.

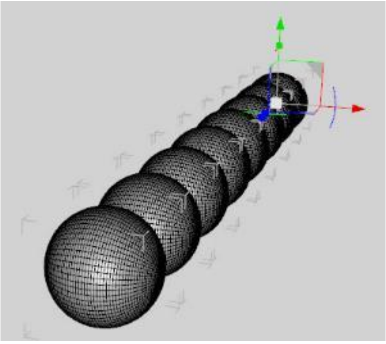
a. Create a single sphere of 2 foot diameter and 32-64 segments and sides.

b. From the main menu select Edit/Duplicate/Duplicate Node Hierarchies. Duplicate the sphere 7 times for a total of 8. Do not make instances as dForce ignores them.

c. Z-translate the 8th sphere a short distance.

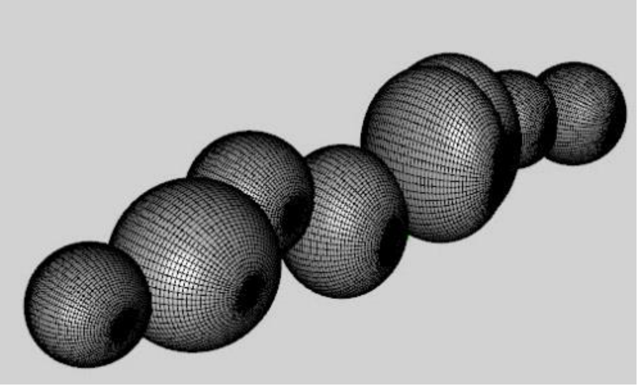


d. Open the Align pane Window/Panes (Tabs)/Align. Select all 8 spheres in the Scene pane. In the Align pane, for Z-axis choose Distribute : Centers and click on Apply.



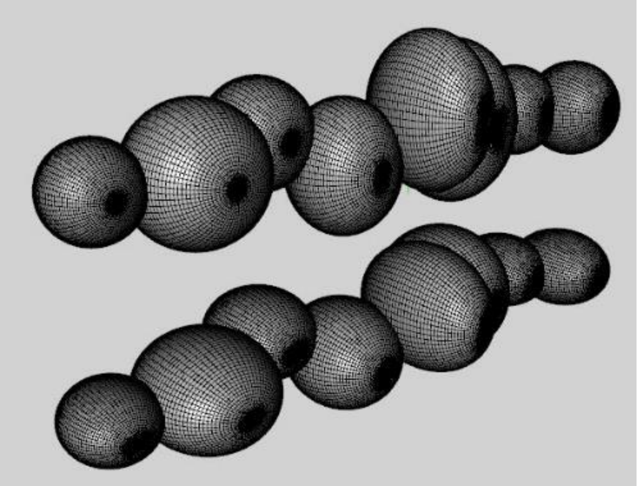
e. With the 8 spheres still selected, choose Create/New Group from the main menu and give the group the name Upper Spheres.

f. Now adjust the scale and position of the 8 spheres. Move some to the right or left, others up and down. Scale some bigger and some smaller, both overall and on different axes. Make sure they overlap otherwise your cave may come to a close.



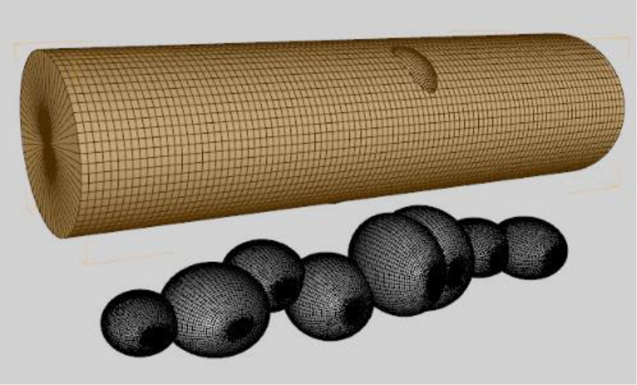
g. Select the group and choose Edit/Duplicate/Duplicate Node Hierarchies. Use the name Lower Spheres for the new group. This group will serve to support the dForce mesh so it doesn't hang down too far.

Optional: Adjust the scale and positioning of the lower spheres so the floor is different than the ceiling.



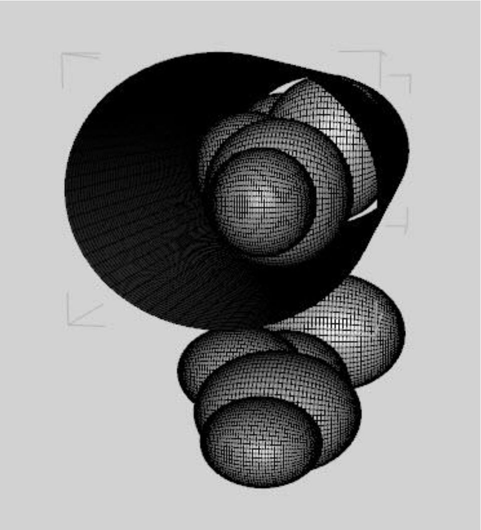
h. Create a Z-positive cylinder of length 8 feet, diameter 2 feet, 100 segments and 64 sides.

i. Adjust the overall, X, Y, and Z scale and rotation of the cylinder so the upper spheres are mostly or fully inside.

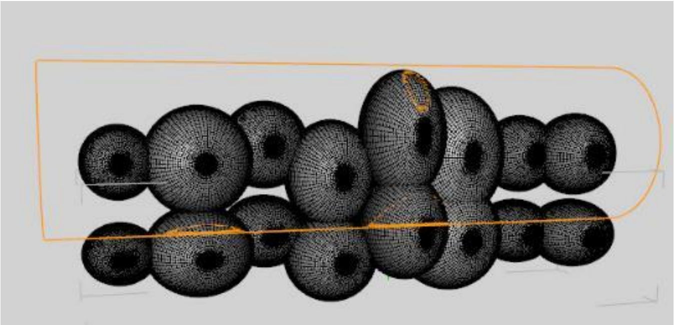


j. Use the Geometry Editor to select all the faces on each end of the cylinder. Right click in the viewport and choose Geometry Visibility/Hide Selected Polygon(s). Right click in the viewport and choose Geometry Editing/Delete Hidden Polygon(s).

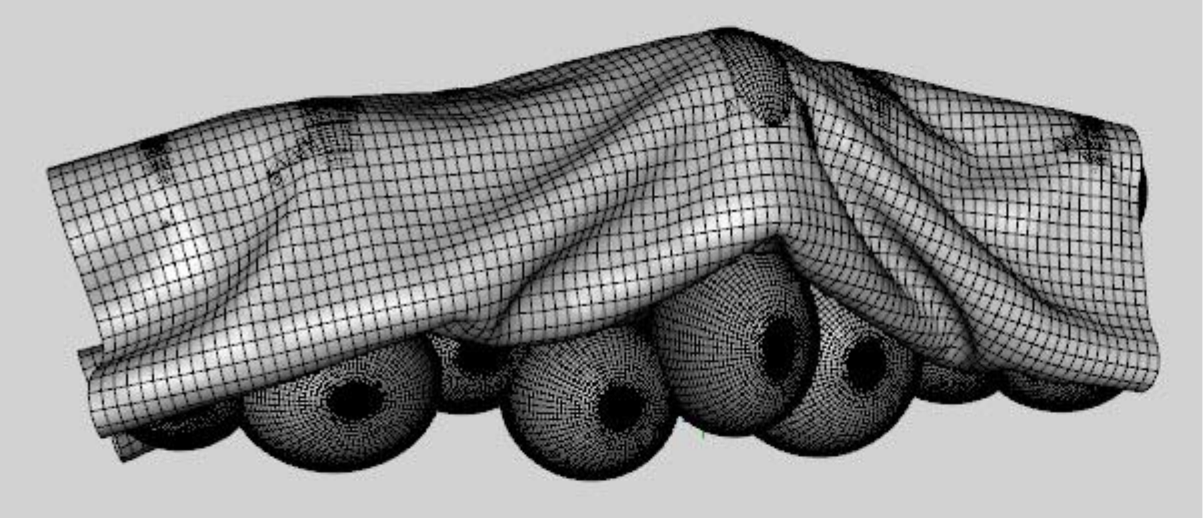
k. Add a dForce dynamic modifier to the cylinder.



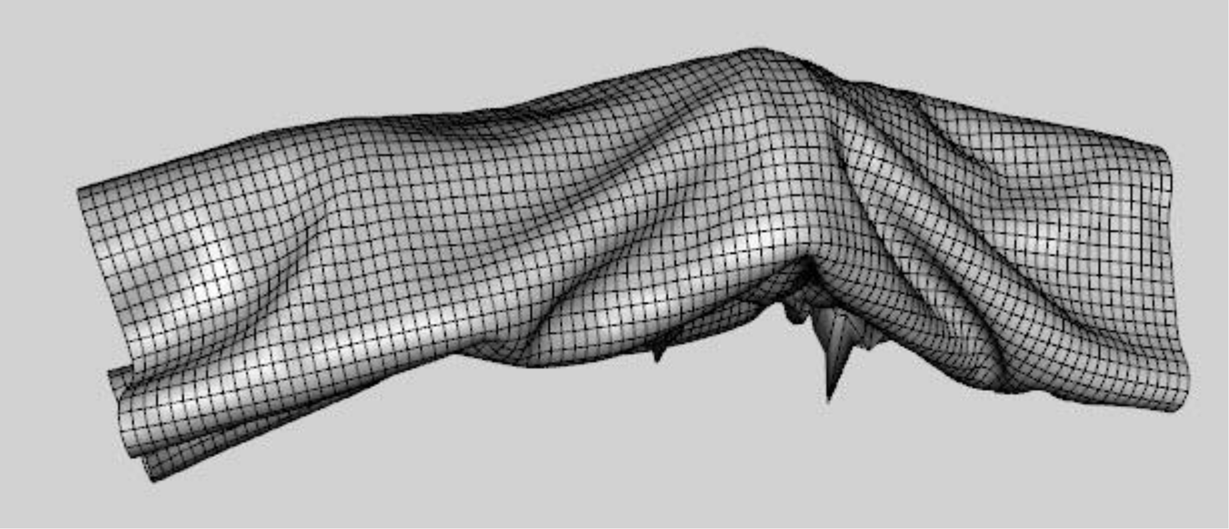
l. Set up an animation timeline. At frame 0 have the group Lower Spheres well below the cylinder. At frame 10, raise the group so it sits just below the group Upper Spheres (it's OK if there is some penetration of the two).



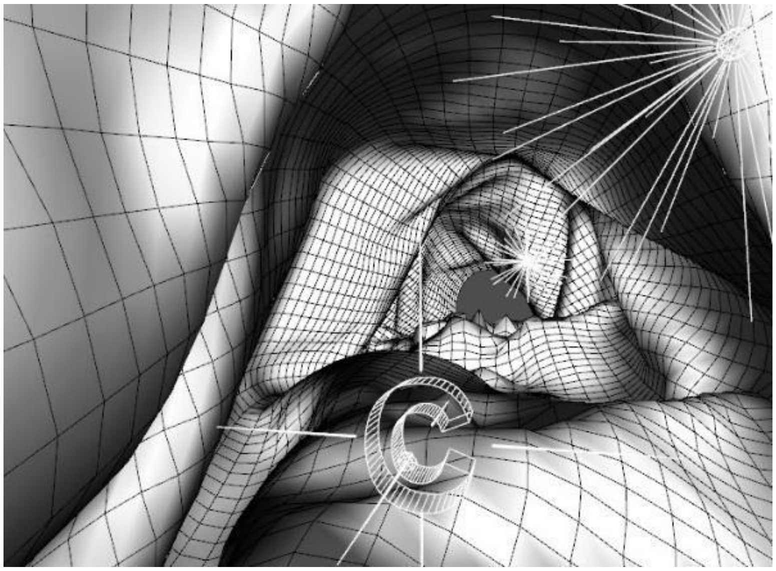
m. Run the simulation.



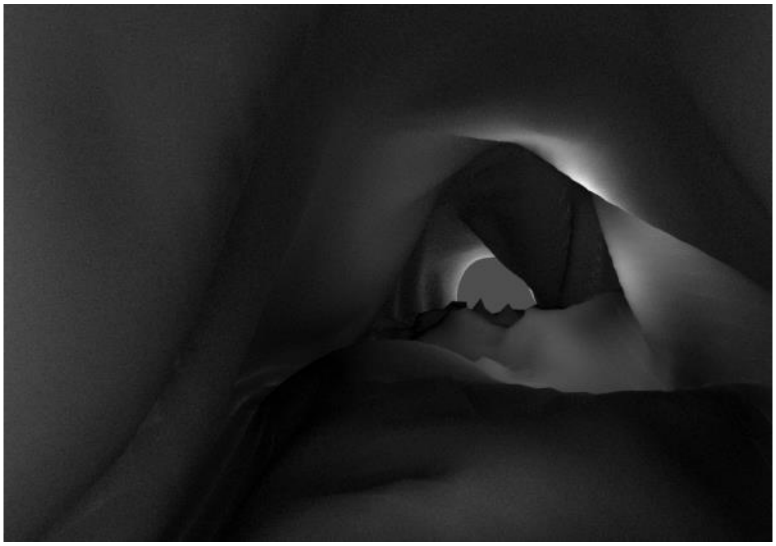
n. Hide the two sphere groups. Now you have a dForce cave! It's OK if you have some spiky spots as they can serve as cave formations (if you don't want them, adjust the height of the Lower Spheres group at frame 10).



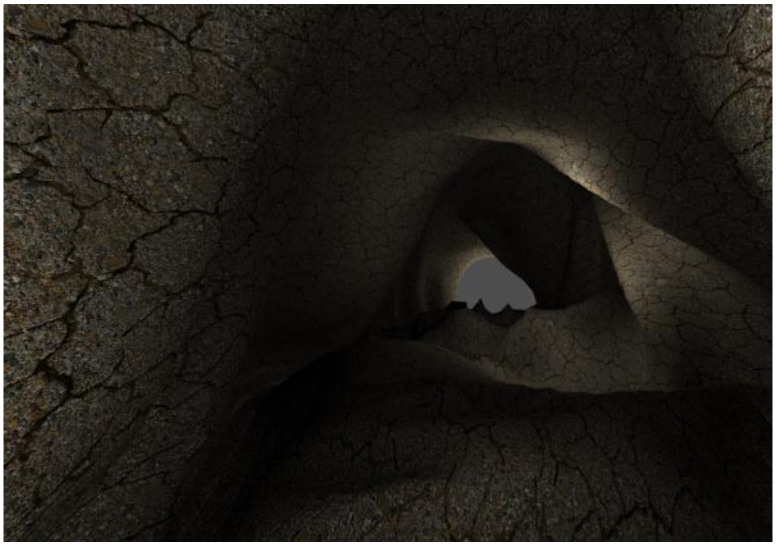
o. Position a camera within the cave and add some lights.



p. I rendered using 3Delight as it's easier to light interior spaces. You can also use Iray if you take steps to add enough light for rendering.



q. Add a rock texture, preferably one with some degree of bump and/or displacement.



r. You can also use the push modifier tutorial I mentioned at the start to create even more variation.

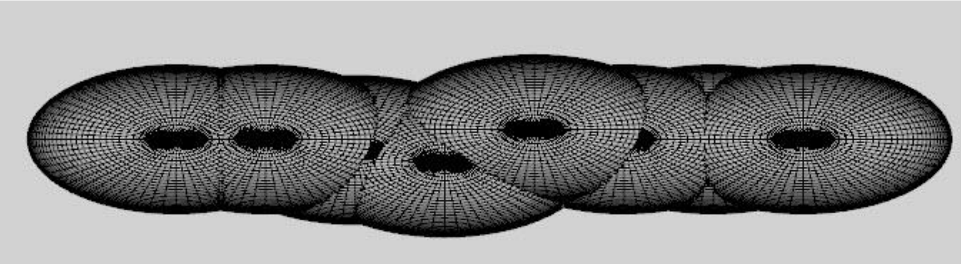
s. Another cave with a different look.

t. If you are going to include a figure in the cave, you'll want to either scale the cave after creation or the figure to fit, as I only used 2-foot diameter spheres. You can create much larger spheres to start but I find it less convenient to navigate the viewport with the larger spheres.

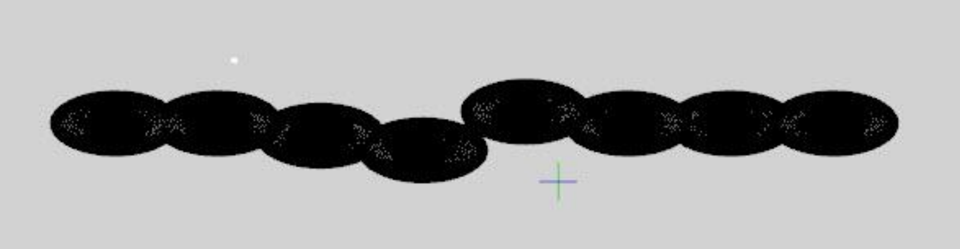
71. **Fantastic Voyage: Making Artery and Blood Cells.** After I posted section 70, dragonfly_2004 mentioned the movie "Fantastic Voyage". I took up the idea and made a scene with a miniature human diver amid blood cells in an artery. It's similar to the cave but with less sag between spheres. Plus I use dForce to create a red blood cell and then use UltraScatterPro to make multiple copies.

   a. Follow steps a to f of section 70.

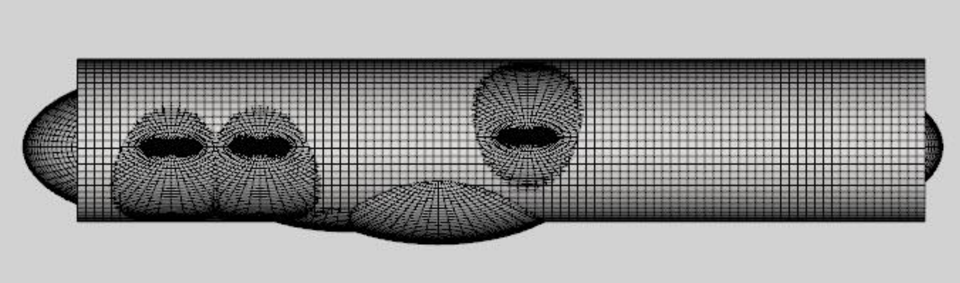
   b. For step f, you want larger more overlapped spheres to smooth the walls. At frame 10, I set overall scale of the spheres to 150 and Y-scale to 200.



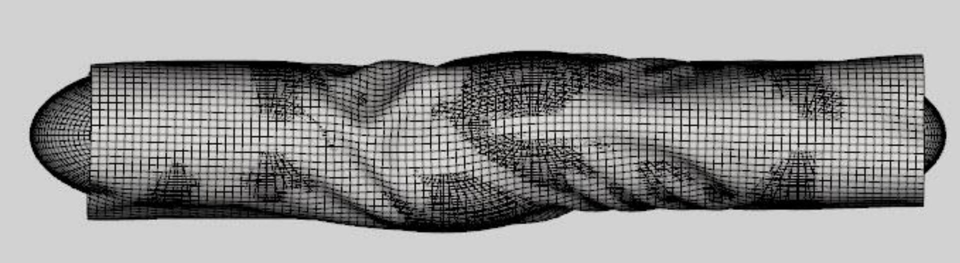
At frame 0 I set overall scale to 55 and kept Y-scale at 200. These smaller spheres will grow and push out on the cylinder's walls.



c. Create a cylinder as in steps h to k of section 70. Adjust cylinder size and position as shown below.

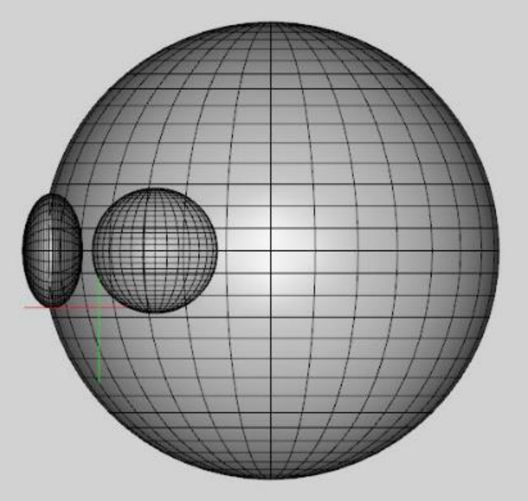


d. We are not using lower spheres this time. Instead, set dForce Simulation pane Gravity and Air Resistance to 0. The scaling between frame 0 and 10 will push out on the cylinder. Here is what it looks like after simulation:



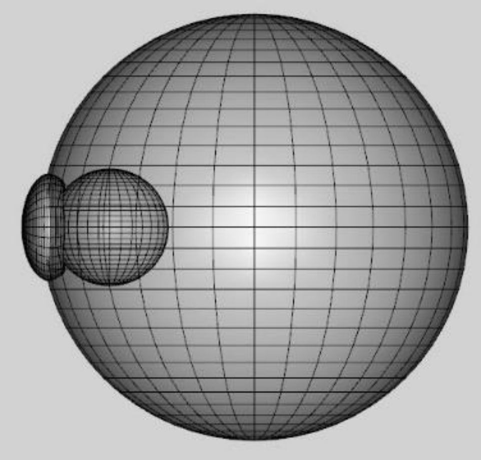
e. Select the cylinder in the Scene pane and in the Parameter pane set Freeze Simulation to On.

f. We next need to make a blood cell. Create a large sphere and two small spheres. Add a dForce modifier to one of the small spheres and flatten that same sphere along the X-axis. Position them as shown to the right (the sphere with the modifier is on the left). Set the Stretch and Bend Stiffness to 0.1 (this is done to manage the triangular polygons on either end of the sphere) and set Density to 2 (to make for a stiffer surface).

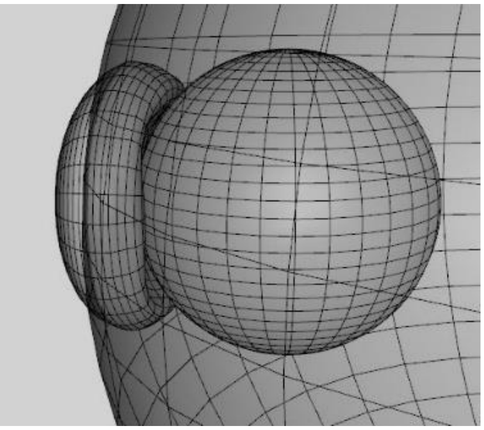


g. We are going to animate the small sphere on the right. Select it and add a keyframe at frame 0.

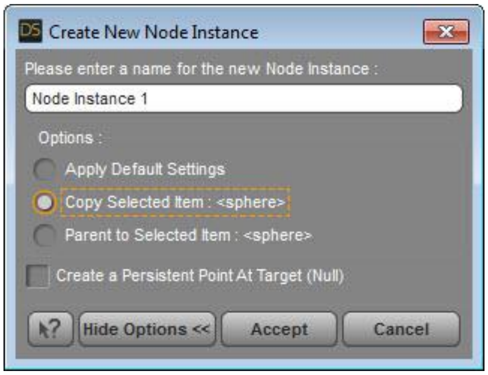
h. We want the movement to be slow to avoid momentum changes, so at frame 28 set the right sphere embedded into the left sphere as shown in the image.



i. Set Gravity and Air Resistance to 0. All the deformation will come from pressing one sphere into the other. Here's what it looks like after simulation:



Tip: It can be hard to see what’s going on with the sphere blocking the view. Create an instance of the left sphere and choose Copy Selected Item. Then you can see what's happening during the collision.



Create New Node Instance

Please enter a name for the new Node Instance :

Node Instance 1

Options :

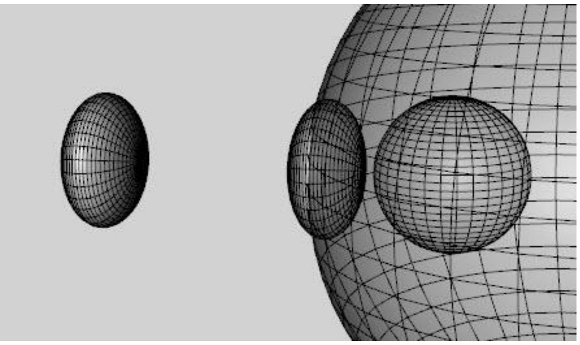
○ Apply Default Settings

● Copy Selected Item : <sphere>

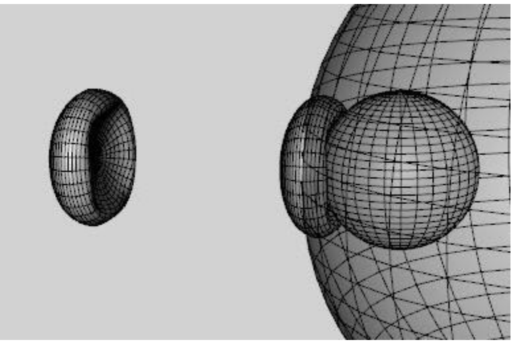
○ Parent to Selected Item : <sphere>

□ Create a Persistent Point At Target (Null)

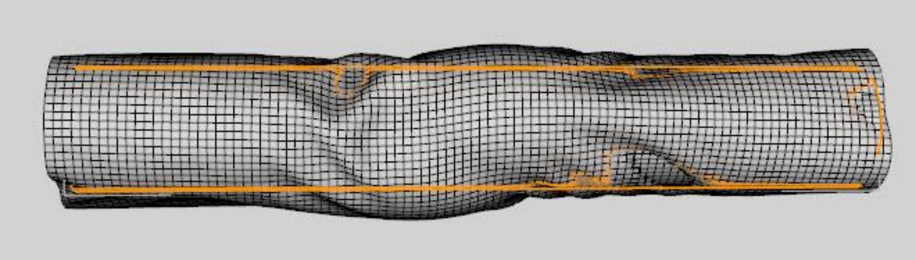
Before simulation (the instance is at the far left):

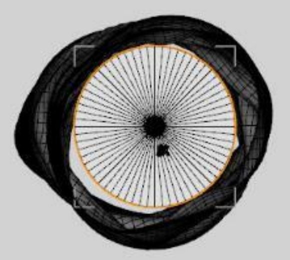


During Simulation:

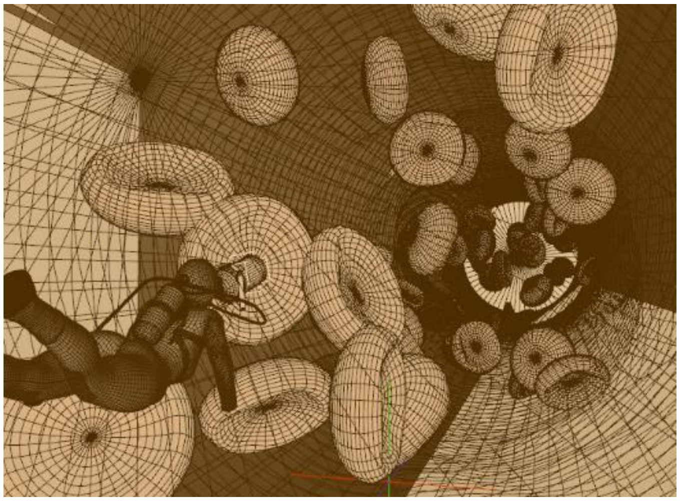


j. We want a bunch of blood cells, so I used UltraScatterPro to create instances as it has a volume fill option. I created a new cylinder smaller than the artery to be used as the target, as using the artery cylinder can lead to items embedded in the wall. Position the new cylinder so it is inside the artery (see orange highlight in images).

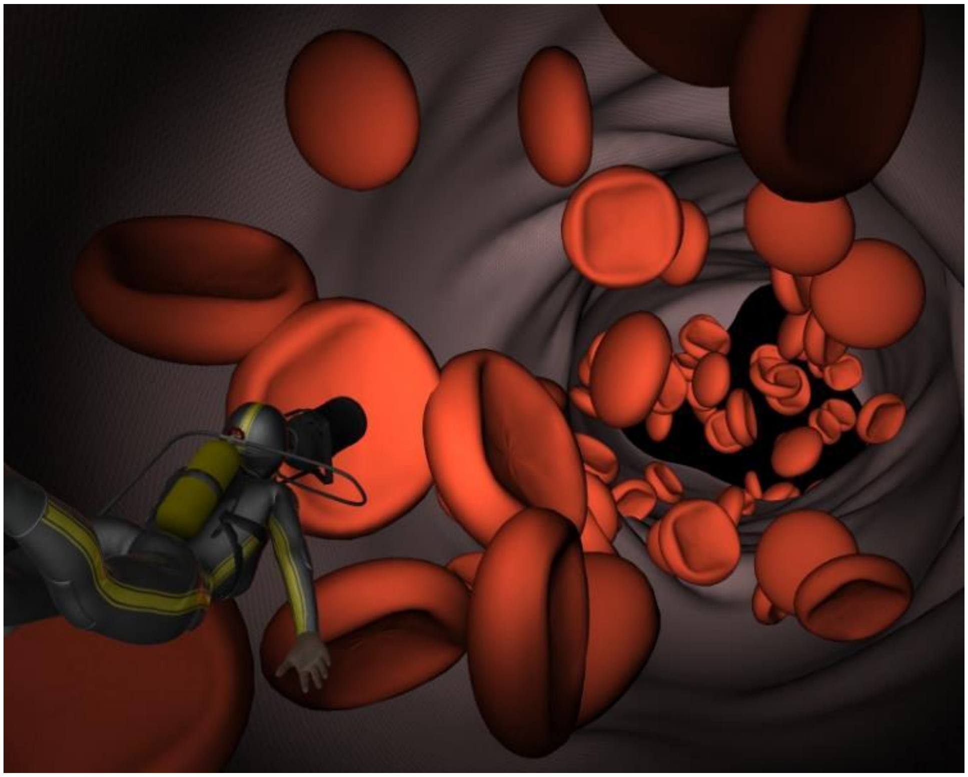




k. This is the result after scattering 70 blood cells.



l. The final render:

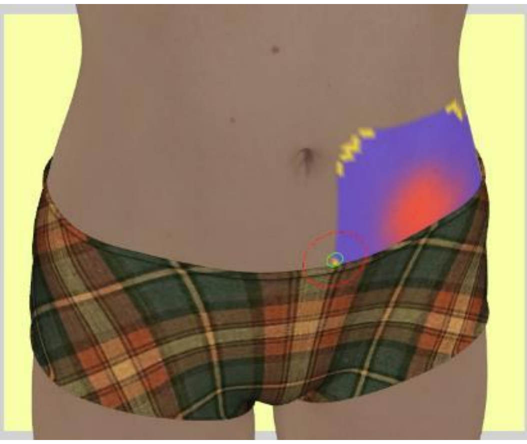


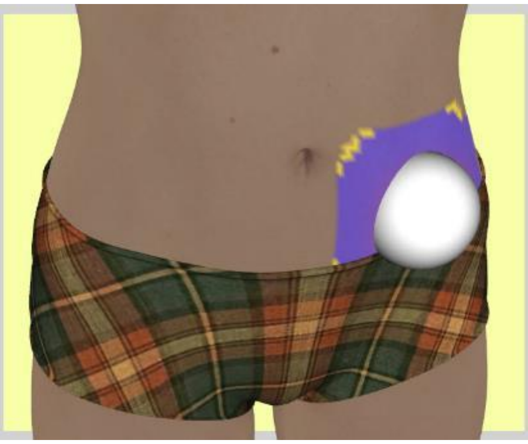
72. **Skin Indentation.** SnowSultan asked on the Daz forum how to simulate the effect of pressure from a hand or other object pressing on human skin. You can create such an effect using dForce although there are certain limitations as I'll mention below.

a. Load a figure and apply a dynamic modifier and a dForce weight node to it.

b. Create a sphere and position it near the stomach. Have it slightly away from the skin's surface at frame 0. At frame 10, have it half-way embedded into the skin.

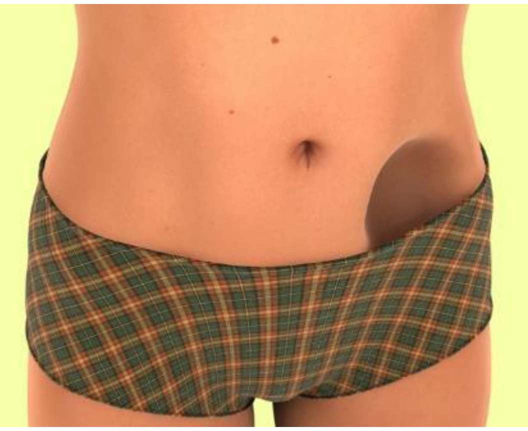
c. Add an influence weight map then right click in the viewport and select all, then right click again and select Weight Editing/Fill Selected and set the value to 0. This will remove all the red color. Next use the paint brush tool and paint a small area of red where the sphere collides with the skin.



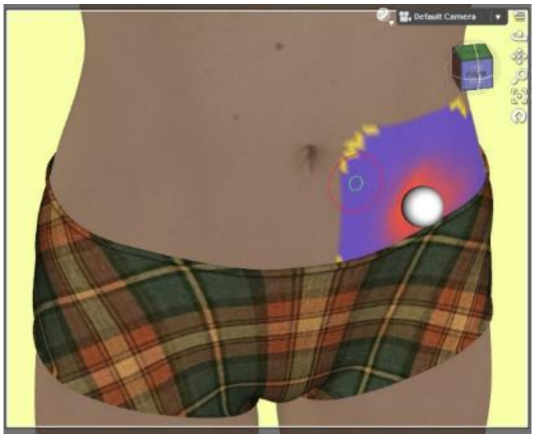


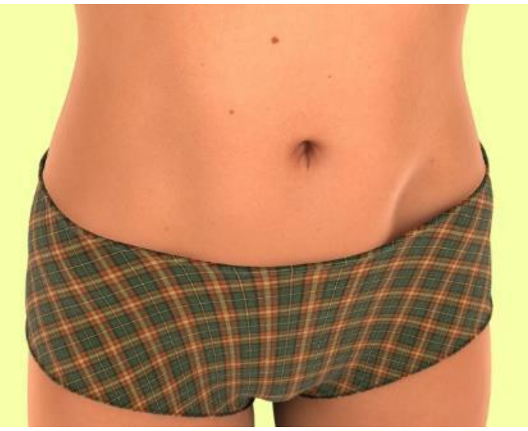
d. Set Gravity and Air Resistance to 0. This allows the changes to be driven by the movement of the sphere only.

e. Run the simulation. You will see the skin indented by the movement of the sphere. With this large sphere halfway embedded, the depression is quite deep.

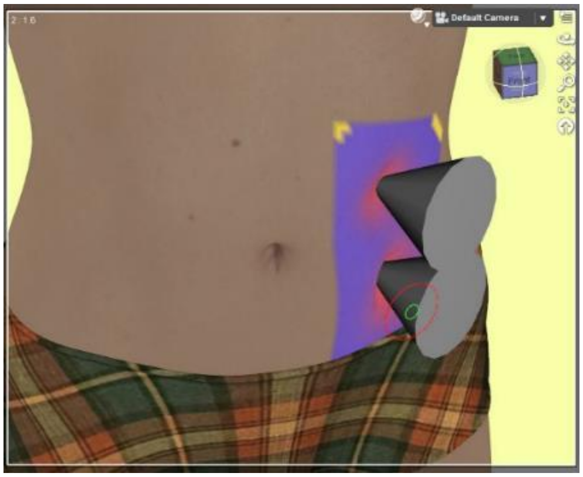


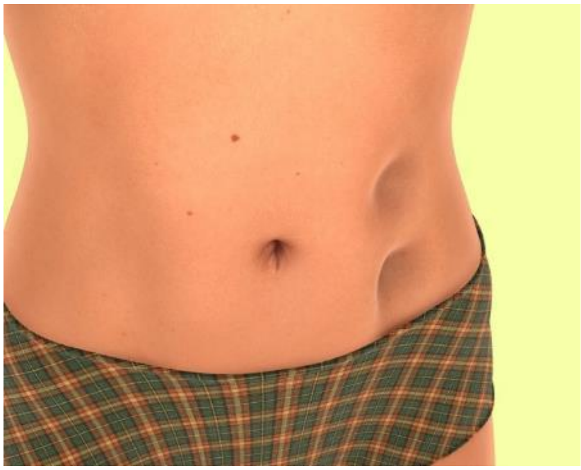
f. Repeat steps b - e with a smaller sphere. You can use the same weight map or repaint it to make the red area smaller. You will see a smaller indentation.



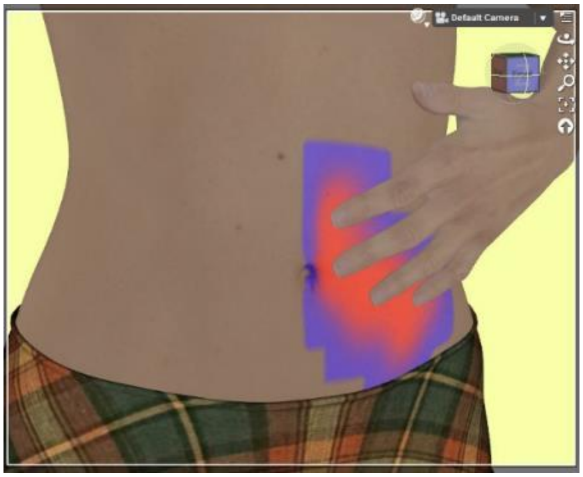


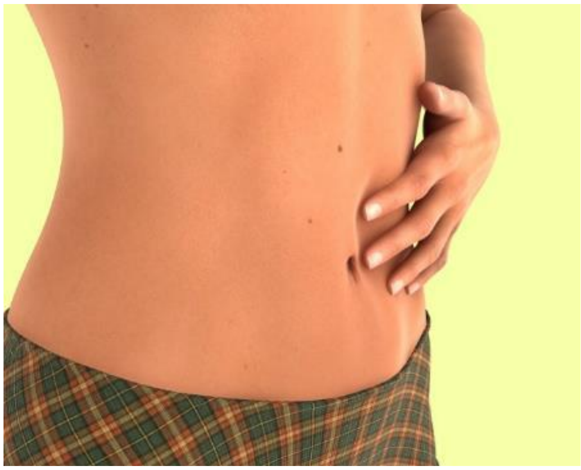
g. Create two cones and position them in the same area as the sphere had been located. Set them away from the skin at frame 0 and into the skin at frame 10. Repaint the weight map to have two areas of red. This will let you create two points of indentation.



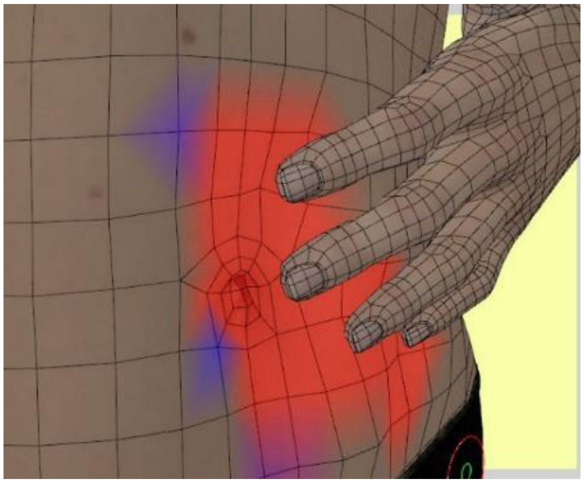


h. Remove the spheres and cones and animate the left arm so the hand moves into the stomach at frame 30. You will see the skin indent.

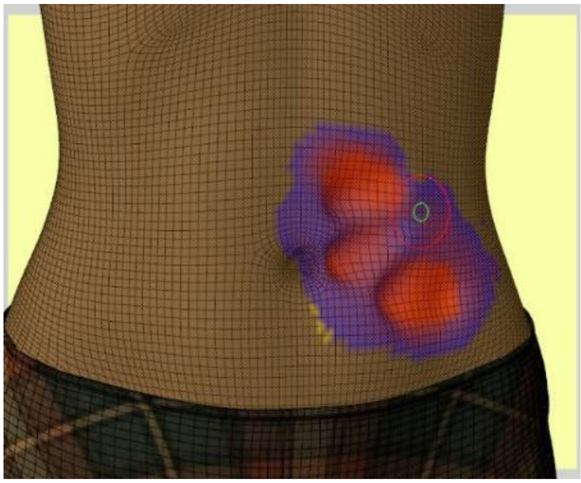




i. The hand shows one of the problems modeling skin indentation with dForce. Because the polygons in the base mesh of the figure are fairly large, you cannot make the map fine enough to show each finger's indentation.

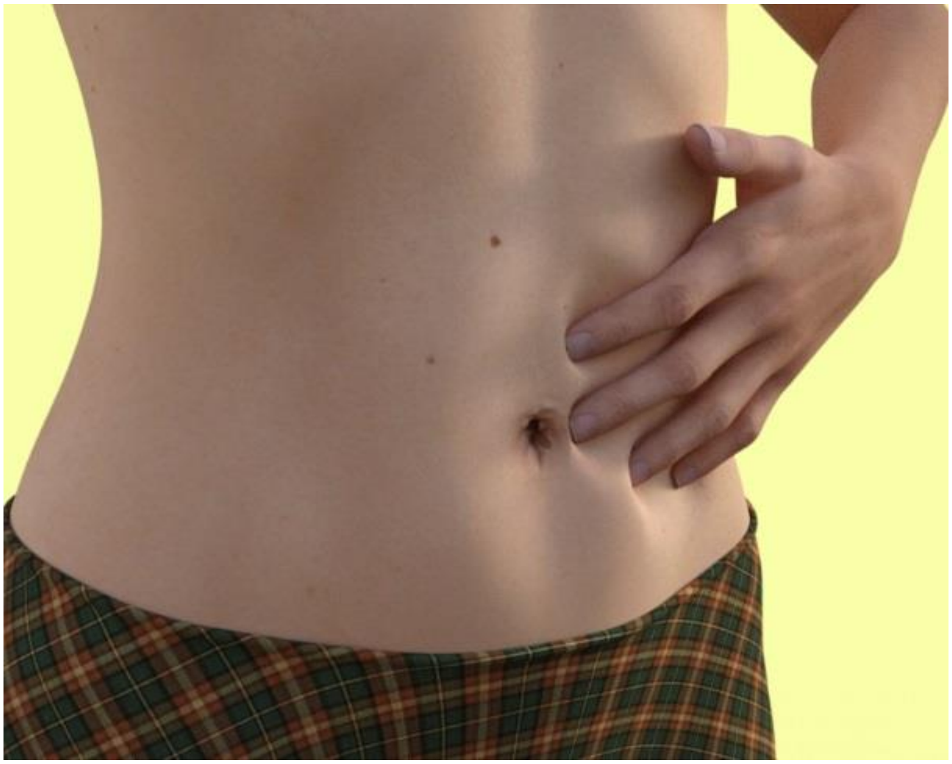


j. If you don't mind some extra work, you can hide everything but the figure, have it set to High Resolution, and export it as an obj file. Reimport the file and it will now have the finer mesh. You can't animate the arm but you can use the original's figure arm to do the pressing. Hide the unneeded body parts of the original mesh so it doesn't interfere. The biggest watch out with this approach is you have to have finalized the figure's pose before exporting.



k. Repaint the weight map with red around the finger tips and some blue in between.

l. Run the simulation. A render with the finer resolution mesh. Note the depressions by each finger now.



m. The other main problem trying to simulating skin indentation is that dForce distorts the surface skin but does not preserve volume. You push in and the skin depresses but the fat and muscles do not bulge out elsewhere. You can try and mimic this yourself by pushing from the inside with some form of primitive or by using a push modifier post-simulation, but then you've left the realm of physics and entered that of artistry.

73. **Blindfold.** A forum member asked for help on creating a blindfold. It took more tries than I expected but I figured out a method that is explained here.

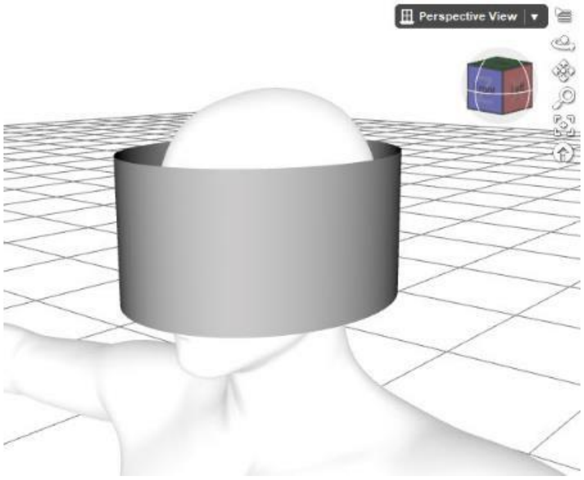
   a. Load a figure (I used G3M).

   b. Create a Y-positive primitive cylinder of diameter 9 inches and length 3 inches. Use 12 segments and 100 sides to give enough divisions for the mesh to transform.

   c. Using the Geometry Editor, select all the faces on the top of the cylinder and hide them. Do the same for the bottom of the cylinder. Then delete the hidden faces (these actions can all be accessed by right-clicking in the viewport when the Geometry Editor is active).

   d. Position the cylinder over the figure's face. I used Y-translate of 163.

   e. Adjust the scale of the cylinder so the figure's head is entirely inside (that includes nose and ears). I used 100, 167, and 116 for X, Y, and Z.



   f. Set up an animated timeline. Select the cylinder and set a key frame at frame 0. Go to frame 10 and change Y- and Z-scale to 100%.

   g. Add the dForce dynamic modifier from the Edit/Object/Geometry menu.

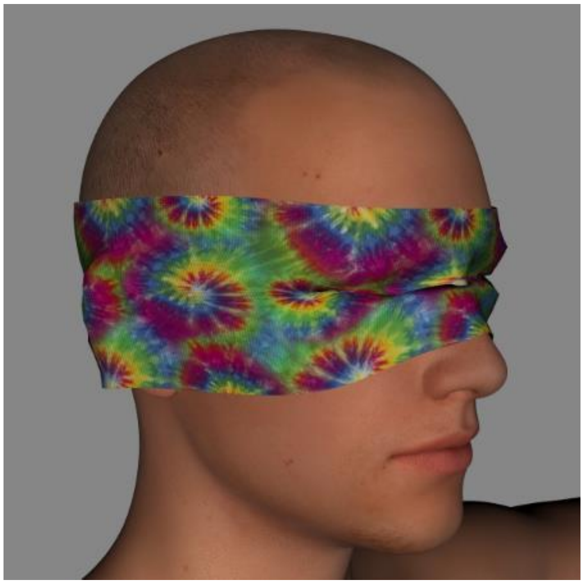
   h. Set gravity and air resistance to 0 in the Simulation Settings pane. This will keep the blindfold from falling down while simulating.

   i. Select the cylinder and in the Surfaces pane, set Contraction-Expansion Ratio to 75%. This will cause the cylinder to shrink tight against the head.

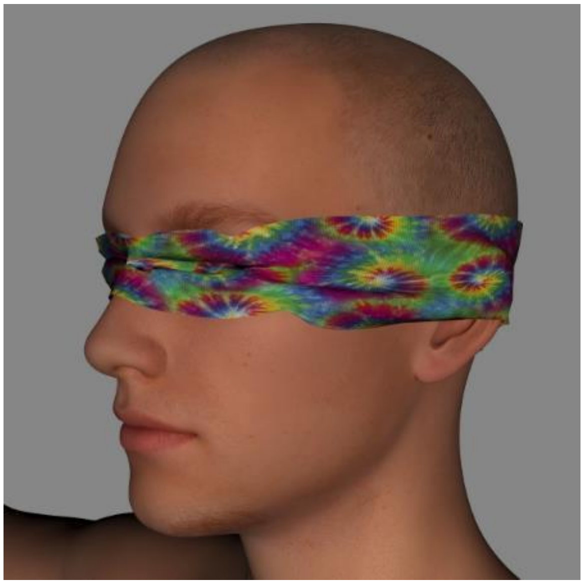
   j. You might experience some skin pokethrough, so add a smoothing modifier from the Edit/Object/Geometry menu. Turn ON smoothing and collision (set Collision Item to be your figure).

   k. Run a custom animation simulation. I used 14 frames.

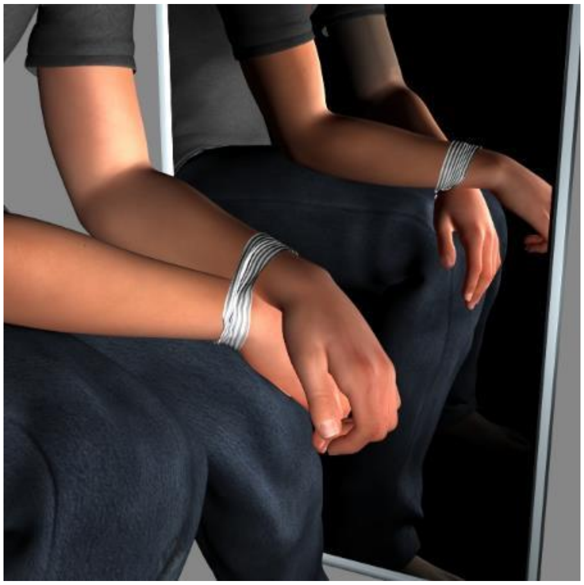
l. Here is the result:



m. For a narrower blindfold, set Y-scale to 125% and Y-translate to 164.5 at frame 0:



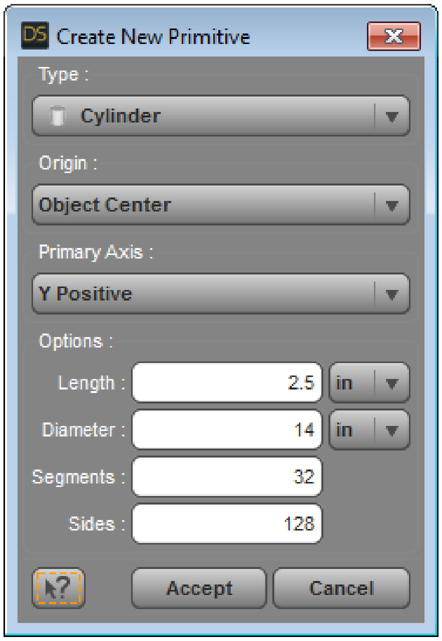
n. You can take a similar approach to wrap the band around the wrists. You'll need to adjust scale, translation, and rotation as needed to keep the band in frame 0 clear of the body parts.



74. Making a Breast Helper for Clothing. You can buy commercial products to do this but you can also make your own. There are two parts: Part 1 is making the helper and Part 2 is using the helper.

Part 1

a. Create a cylinder with the specs shown in the image to the right.



Create New Primitive

Type : Cylinder

Origin : Object Center

Primary Axis : Y Positive

Options :

Length : 2.5 in

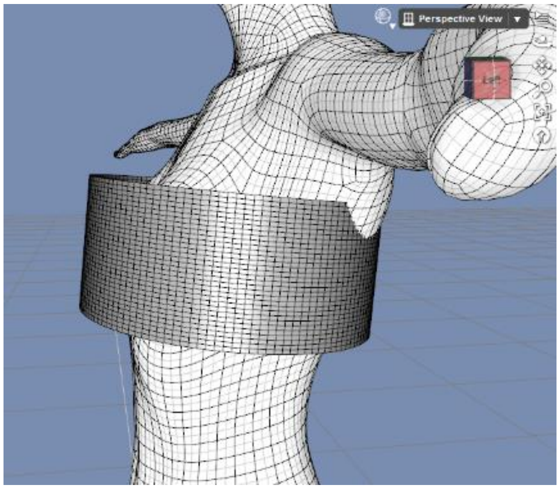
Diameter : 14 in

Segments : 32

Sides : 128

b. Go to Tool Settings pane and select the Geometry Editor. Select all the top faces of the cylinder, right click in the viewport, and choose Geometry Visibility/Hide Selected Polygon(s). Do the same for the lower faces. Then right click in the viewport and choose Geometry Editing/Delete Hidden Polygon(s). This will leave you with just the outside surface of the cylinder.

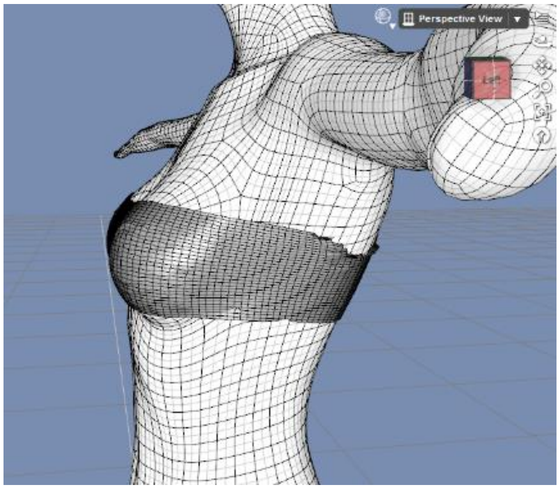
c. Raise the cylinder and position it around the breasts. Set Y-scale to 200% and Z-scale to 80%. Adjust X-rotate to -5.5%. Adjust Z-translate so cylinder clears all parts of the cylinder except leave a tiny bit embedded in the back corners of the figure.



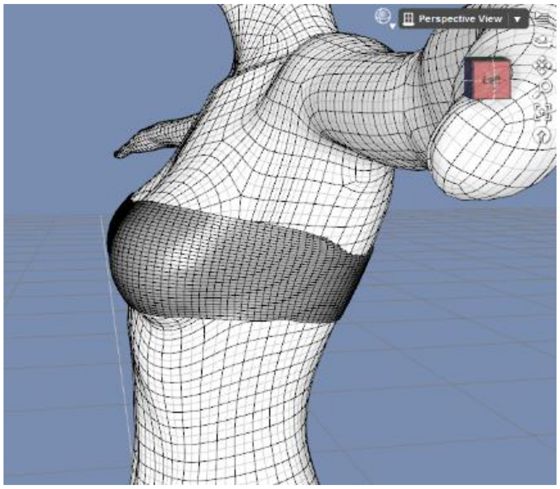
d. Add a dForce modifier to the cylinder. On the Surfaces pane, set Contraction-Expansion Ratio to 80%.

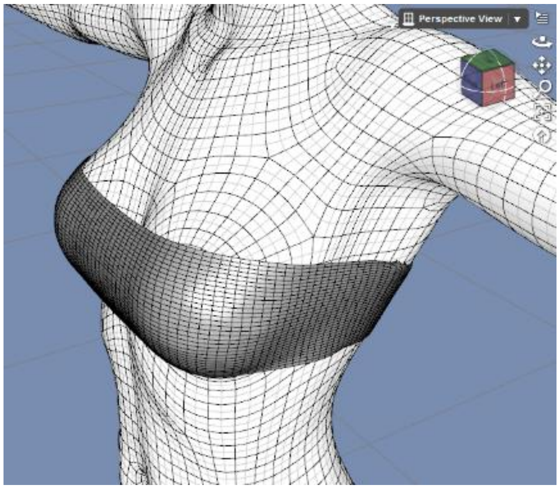
e. On the Simulation Settings pane, set Gravity and Air Resistance to 0. Set Duration to be Animated Custom with frames 1 to 15.

f. Run the Simulation.



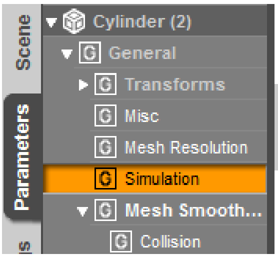
g. Add a Smoothing Modifier to the cylinder to smooth out wrinkles. This completes Part 1.

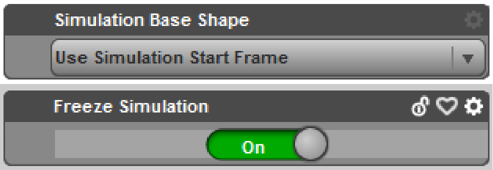




Part 2:

h. Before doing anything else, on the Parameter pane under the Simulation tab, set Freeze Simulation to On for the cylinder. This will preserve the current shape.





Simulation Base Shape

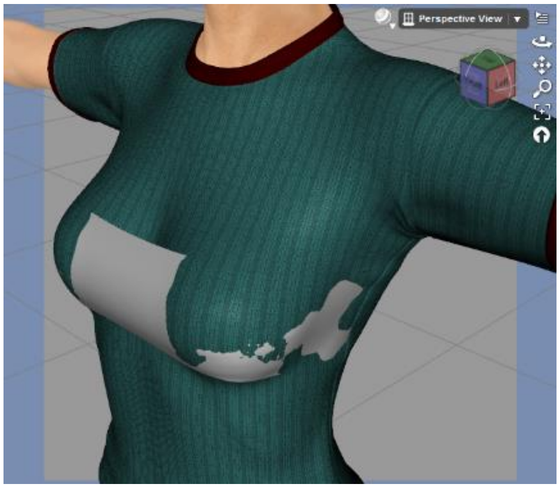
Use Simulation Start Frame

Freeze Simulation

On

i. After freezing the simulated cylinder, be sure to never hit the Clear button on the Simulation Settings pane as it will clear all simulations including those frozen and you will need to start over. Save your scene or scene subset so you can return to it if you accidently hit that button.

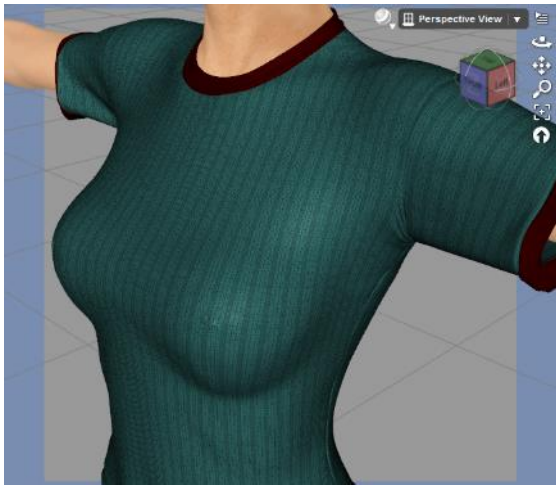
j. Add a shirt to your figure. I used 70's Gym Uniform for Genesis 3 Females. You'll find the shirt conforms to the figure with the helper partially visible.



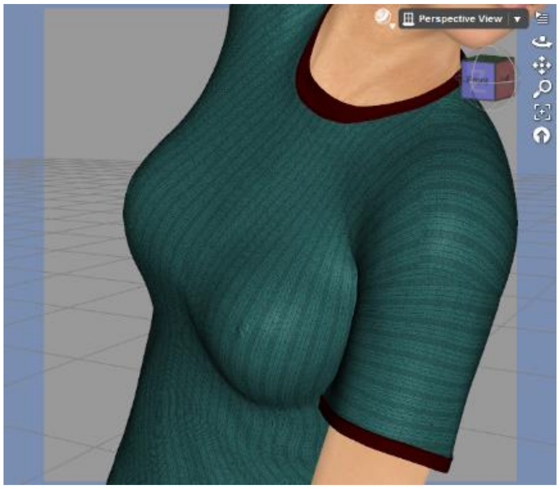
k. Add a dForce modifier to the shirt. This shirt can use the default surface properties but other clothing may require Bend Stiffness to be set to 0.2.

l. Select the cylinder in the scene pane, go to the animation timeline pane and set a keyframe at frame 10. Go to frame 0 and set the scale for the cylinder to 50%. Go back to frame 10 and be sure scale is back to 100%. This will hide the helper within the figure at first, then it will push out the shirt as the simulation proceeds.

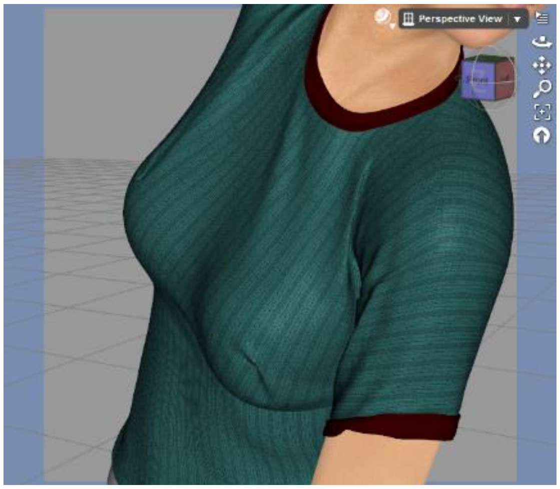
m. Set gravity to 1. Run the simulation. You will see the gap between the breasts has smoothed out.



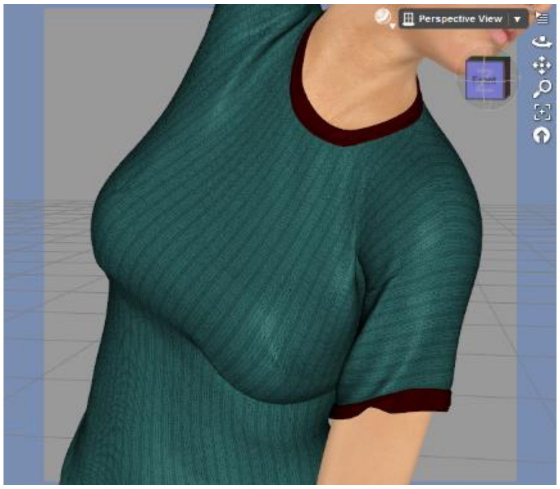
n. Hide the cylinder and repose the figure. Since the shirt was deformed already, the helper effect stays with the shirt.



o. This is what the shirt would look like for this pose with a dForce simulation on the shirt and no helper in use.



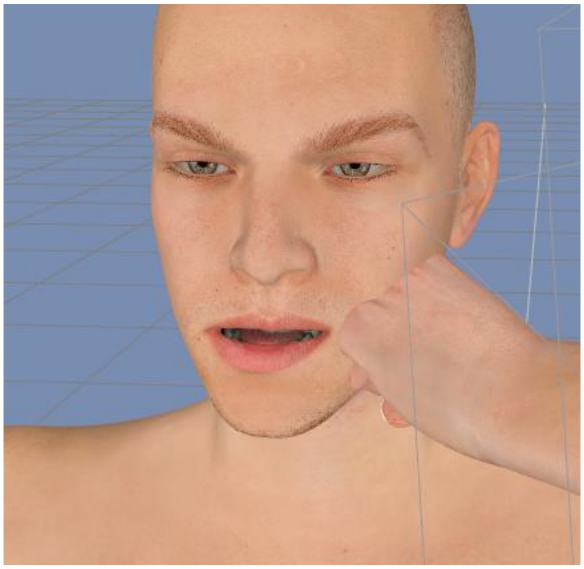
p. You can repeat Part 1 with the new pose (simulate the helper) then Part 2 (simulate the shirt) but I find the results shown in step n acceptable without the added work.



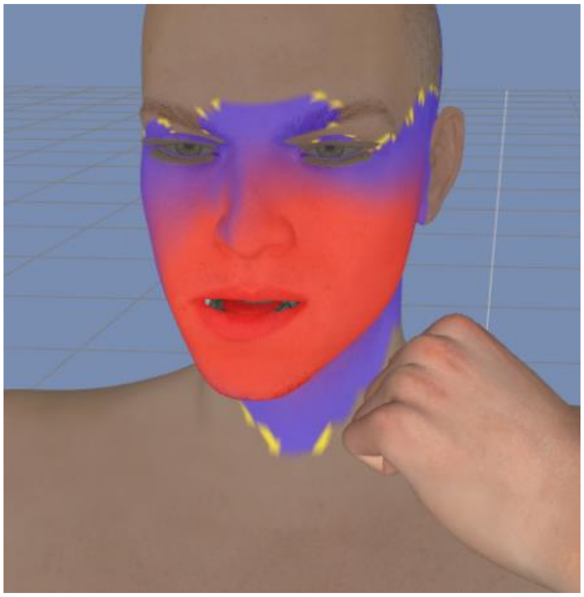
75. Skin Indentation: Punch to Face. Someone on the Daz forum asked about showing a face being hit. Here is how to simulate it.

a. Set up the pose so the fist is embedding into the jaw at frame 8.

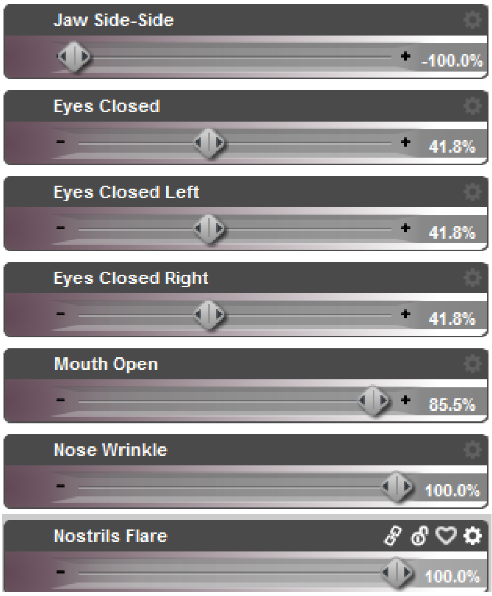
b. Add a dForce modifier and weight node to the figure being struck.



c. Create influence map and select all, set weight to 0, then paint influence onto the jaw, cheeks, lips, etc. Red indicates highest influence; lack of color indicates no influence.



d. dForce moves surfaces, not volume. To show a reaction to the punch, I set several face parameters to what's shown in the image at right.



Jaw Side-Side -100.0%

Eyes Closed 41.8%

Eyes Closed Left 41.8%

Eyes Closed Right 41.8%

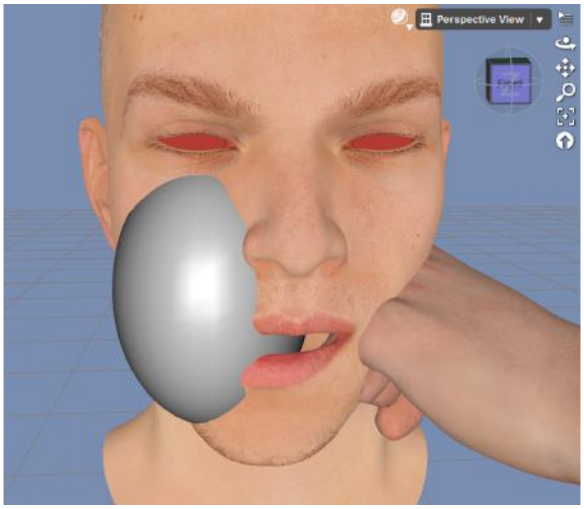
Mouth Open 85.5%

Nose Wrinkle 100.0%

Nostrils Flare 100.0%

e. Hide the upper and lower jaw and tongue to avoid an explosion.

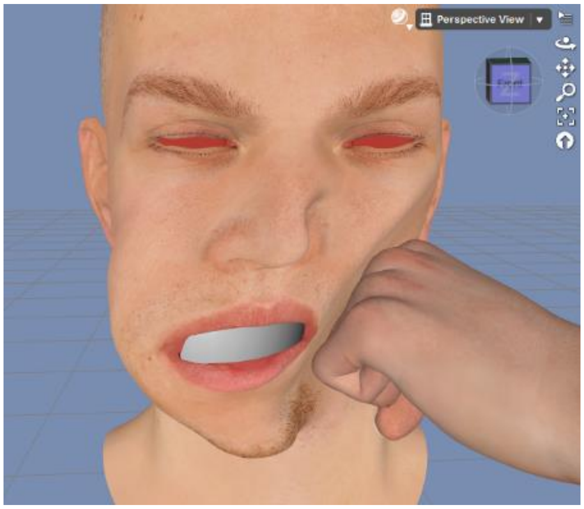
f. To show reaction on the opposite side of the face, I embedded a sphere within the head. At frame 0 it's hidden within the face. At frame 8 it's scaled so it protrudes outside the face.



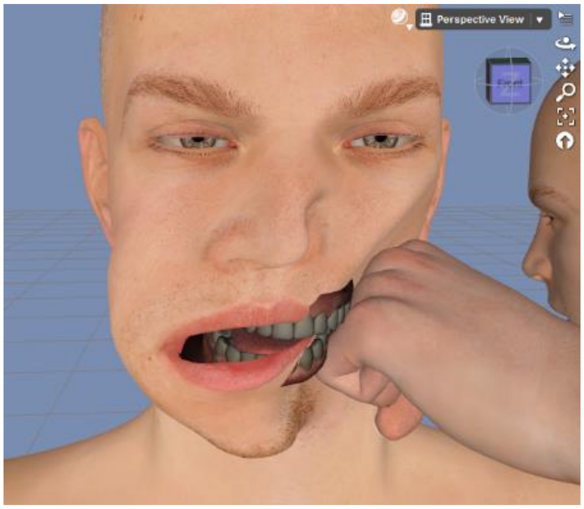
g. Set Gravity and Air Resistance to 0.

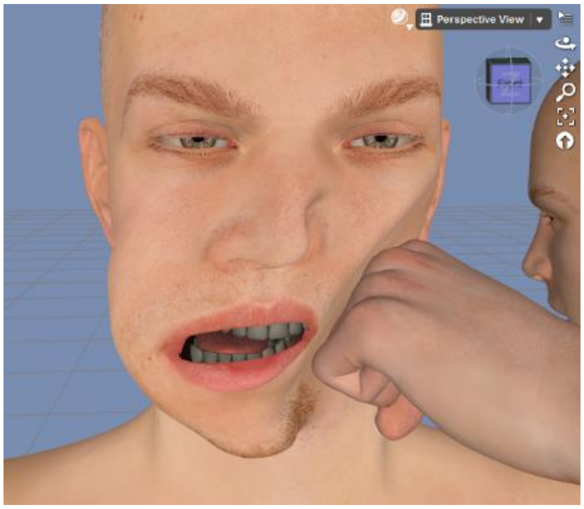
h. (Optional) To speed up simulation, hide as much of the two figures as you can, leaving only the head of the one and the punching arm of the other. **Tip:** hold down ctrl while clicking on an eye and it will hide/show all children.

i. Run the simulation and you will see the jaw and cheek deform under the punch.



j. After simulation the teeth may be jutting out of the jaw. I used translate parameters to move them (don't move the jaw, just the teeth).





k. There was one odd behavior while doing this: the jaws begins to deform before the fist is close. Not sure what's driving that but I presume it's the dForce parameters making adjustments. If after the sim the fist is not right at the skin, move the hand in closer before rendering.

l. You can adjust the size and position of the sphere and repaint the weight map as needed to get the look you want as I did for the render to the right.

Go to Menu