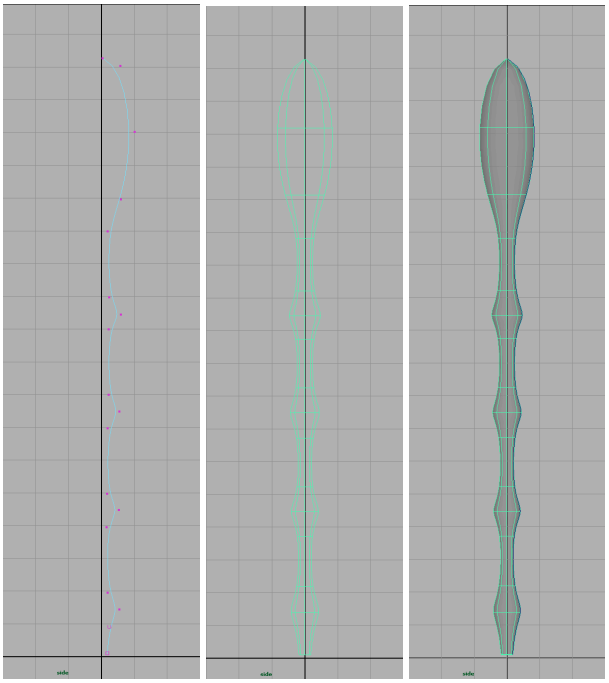
Maya 103: Animation

During this tutorial you will be creating a low resolution 'stand-in' model of the reovirus sigma 1 protein (the tentacle-like trimeric protein that extends out of the virus' turrets). The 3 monomers are thought to be in a flexible conformation during the early phases of the viral life cycle – once the surface of the virus loses its sigma3 coating (upon cleavage by chymotrypsin in the gut lumen), the sigma1 trimer transitions to a coiled-coil conformation and rigidifies. Therefore, we would like to animate each monomer as a flexible/trembling arm initially, and then have it erect and coil as a triple helix (with the other 2 rigid monomers).

Technically, in Maya, we can achieve this by creating a skeleton or joint-chain within the structure, and then bind (or 'skin') that skeleton to the geometry. We will then apply inverse kinematics to this skeleton in order to simplify its animation – we will 'key' the position of the inverse kinematic handle (IK handle) over time and then explore how to loop that motion cyclically in the graph editor. To achieve a 'trembling' motion on the sigma1 tentacles, there are several options to explore: 1) we will apply a simple expression to the x, y, z translate values of the IK handle, and 2) we will use values sampled from a fractal texture (yes, a texture!) to drive the movements of the IK handle. Finally, when the flailing monomers are ready to rigidify and twist around one another, we will apply a twist deformer to the trimer geometry.

Creating the tentacle

Switch your perspective viewport to the side orthographic view. Go to **Create -> CV Curve Tool** and begin placing points according to the image on the right. When you have placed your last point, remember to hit Enter to complete the curve creation process. With the curve selected, switch to component mode (F8) if you wish to edit any of the points.



Creating a low-res sigma1 monomer using a revolved curve

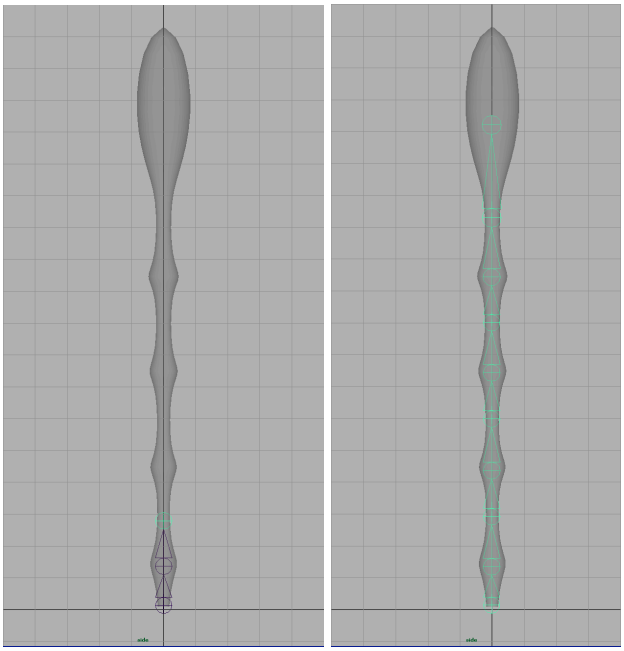
With the curve selected (and back in object mode), select the "Surfaces" master menu set and go to **Surfaces -> Revolve**. Activate smooth shaded mode (**press 5**), and at the top left of your side *viewport* (not the main drop-down menus), go to **Shading -> X-Ray**. This mode of representation will make it easier to work with joints for the remainder of the tutorial. You can also select the curve and hide it (**Ctrl-H**) so it doesn't get in the way.

Now to properly save your file, go to **File -> Project -> New**, select an existing directory in the "Location" box (My Documents for ex) within which to create a new project directory (call it **tutorial_04** – in the "Name" box), and click on Use Defaults (bottom center of the panel) to populate all the remaining fields. Then click Accept. Now **Save** your file **"sigma1_rig"** and it should automatically place it in the "scenes" folder of your newly-created directory.

Creating the rig

While still in side orthographic viewport, go to the **Animation** menu set and select **Skeleton -> Joint Tool | options**. In the panel's upper left menu, go to **Edit -> Reset Settings** (if the tool settings appear in place of the Attribute editor on the right of the Maya interface, simply click the "Reset Tool" button at the top of that panel).

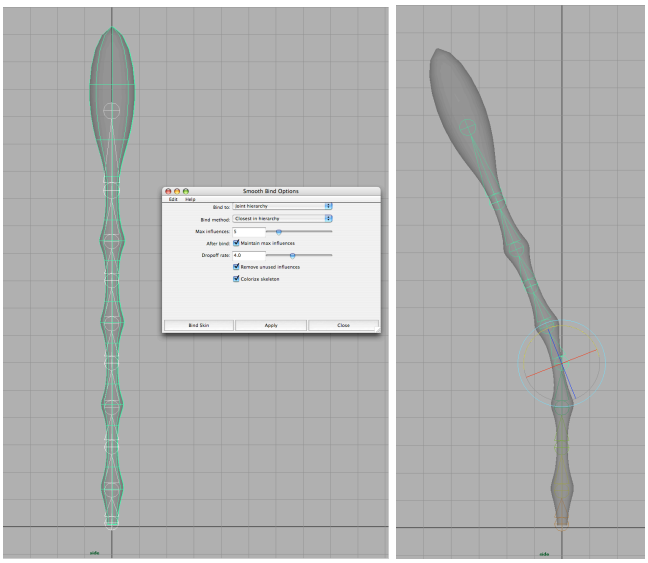
Your mouse will now become a 'crosshair' cursor – clicking in the viewport will create a set of hierarchically-connected joints. Start at the bottom of the sigma1 monomer – **shift-click** a joint very close to the origin and then move upwards (keeping shift down as you click – this will ensure that the joints are all created in a straight line above the first one). Try to place a joint roughly at the center of each vertical constriction/expansion of the geometry (see images to the right). As with curve creation, hit **Enter** to complete the joint chain.



Creating a joint chain within the sigma1 geometry

In the outliner, you'll notice that a **joint1** node has been created – **shift click** this node to fully expand its contents/children. Notice that every joint is a child of the one above it – in the viewport try selecting different joints in the chain by clicking on them and watch how they are being highlighted in the Outliner. You'll notice that joint1 is at the base of your sigma1 monomer – it is the **root joint** (first one you created). Indeed, if you select the root joint, all joints below it in the Outliner are also selected and if you rotate it, all children are also rotated. Now try this on a joint in the middle of the chain and notice that only its children are affected by rotation.

Undo any joint rotations you've applied. It's time to connect your geometry to the skeleton: **select the root joint**, **shift-click your geometry** and under the Animation menu set go to **Skin -> Bind Skin -> Smooth Bind | options**. As before go to **Edit -> Reset Settings** and click on **Bind Skin**. To check that the binding worked, select a middle joint in your chain and rotate it slightly – the geometry should now deform with it! Reset the joint rotation to its default position.



Binding/skinning your geometry to the skeleton & testing

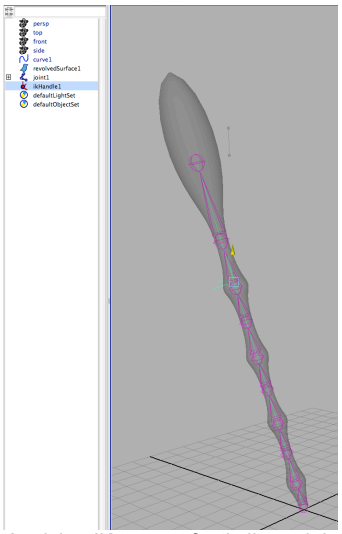
You could animate your geometry simply by selecting joints one by one and moving/rotating them – this is called forward kinematics. However, depending on the motion you are trying too achieve, this could become very

tedious… too many joints and too many keys to set on their x y z translate values! Instead we will create an *inverse kinematic handle* between the first and last joint in the chain to control the motion of the entire chain with one 'controller.'

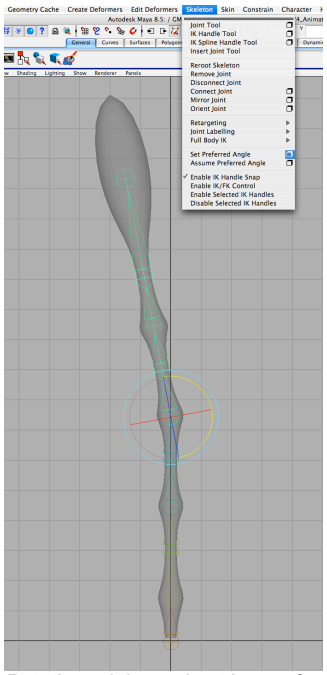
Go to **Skeleton -> IK handle Tool | options**. Reset Settings as before (whether in the floating window or panel on the right) and **switch from ikSCsolver** to **ikRPsolver**. Once you activate the tool, it works interactively in that you need to click (or marquee-select) the first joint and then click (or marquee-select) the last joint that you want controlled by IK. In this case you will first **select your root joint** and then your **last joint** in the chain. If you forget how to use the tool, keep in mind that Maya provides you with step-by-step instructions in the lower left-hand corner of your interface. You exit the tool after the second click - your joint chain turns purple. The outliner now shows a new node called **ikHandle1** – if you select it and activate your **move tool (w)**, you will see in the perspective view that the handle is positioned at the very end of your joint chain (basically overlapping with the position of the last joint).

Try to **move** this **ikHandle** around in your perspective viewport… it actually appears to be 'stuck' on the axis of the joint chain, and the sigma1 monomer is moving around as a straight rod – not the bending behavior we are looking for. This is happening because we created a perfectly linear joint chain (remember that we shift-clicked the joints perfectly one above the other when we created the skeleton in side orthographic view). As a result, the IK doesn't know which way to bend the chain. One way around this is to tell Maya what direction we want our skeleton to bend... to do this we first need to **delete** the **ikHandle** we created (select it in the Outliner and delete).

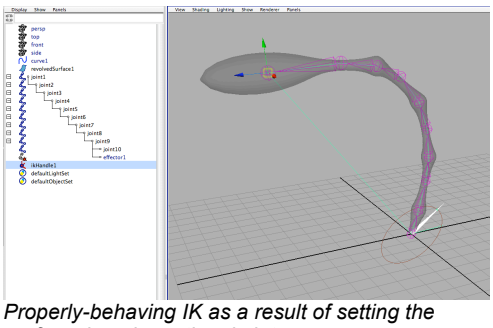
Return to the side orthographic viewport, **select a joint in the middle** of the skeleton, and **rotate** slightly using the green control handle (you can also just plug in a value into its Rotate Y attribute in the Channel Box). With that joint rotated and still selected, go to **Skeleton -> Set Preferred Angle | options**. Make sure '**recursive'** is selected and click **SET**. Return the joint's rotation Y value to 0, and let's repeat the ikHandle creation process: **Skeleton -> IK handle Tool | options** (select **ikRPsolver**), select your root and then last joint. Back to



Applying IK to a perfectly linear joint chain leads to problems



Rotating a joint and setting preferred angle

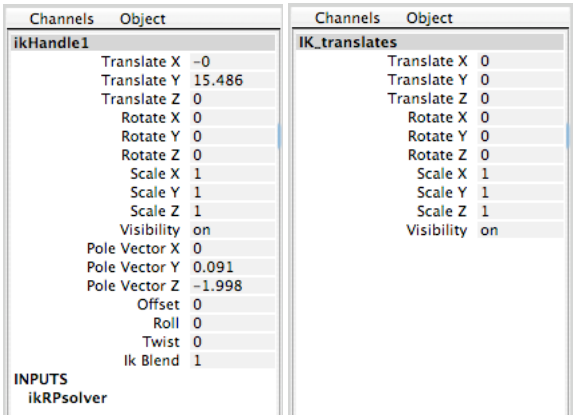


Properly-behaving IK as a result of setting the preferred angle on the skeleton

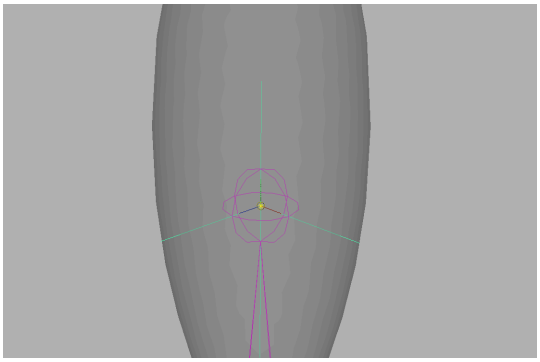
perspective view, select the ikHandle and move it around – tentacle unleashed!
A good practice when dealing with IK is to group the handle to itself so that you are not animating/keying the handle itself. For example, select the ikHandle1 node in the Outliner, and check out its Translate x y z values in the Channel Box… they are not 0 (even at its default, 'resting' state). Doing a 'Freeze Transform' to reset those values to 0 0 0 for this default pose would not work on an ikHandle. The solution is to **group the ikHandle to itself**. **Select the ikHandle1** node in the Outliner and go to **Edit -> Group** – rename the group "**IK_translates**." Notice that the group's Translate x y z values are all 0. To make the animating more intuitive, we'll want to move the pivot of this group to the location of the ikHandle – select the group in the Outliner, activate the move tool (**w**) and press **insert** to edit pivot. Now **activate 'snap to points**' mode:

Point snap mode (top of the Maya interface)

Move the group's pivot to the location of the ikHandle at the very end of the joint chain – should snap pretty easily. **Exit the edit pivot** mode by pressing **insert again** and make sure to **turn off point snap mode** as well. Now select the group in the Outliner and move it around in the perspective viewport to make sure it works (i.e. is moving the joint chain in the same way that the ikHandle was). From now on, make sure ***NOT*** to select/move the ikHandle when animating your skeleton (this would change the ikHandle's position relative to its group)!!!



ikHandle and 'IK_translates' Translate x y z values

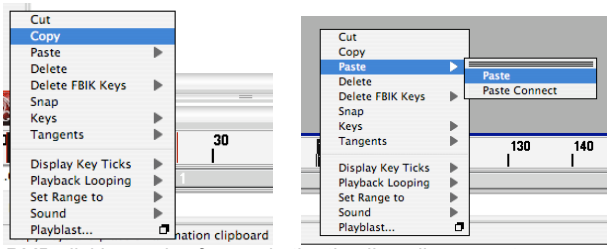


Group's pivot is snapped to the location of the ikHandle

Animating your rig

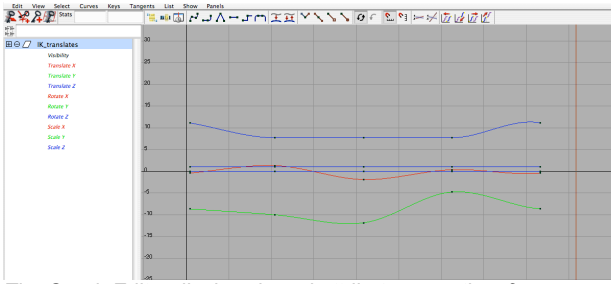
It's time to set your rig in motion by setting keys on the position of the IK_translates group node. Begin by extending the number of frames in your timeline – set it **1-300 frames**. If you recall the overall sequence we are trying to create, we want the tentacle to move around for a while and then rigidify and twist at the end. So we'll plan to have the tentacle flail around between frames 1 and 200 and then twist between frames 200 and 300. In perspective view, **select the IK_translates** node and activate **move (w**) – make sure you are on frame 1 and then move

the tentacle around. To key that position, simply **press s** – notice that all the channels for that node become orange in the Channel Box. Now **move to frame 25**, **move the tentacle** a bit, and **key its new position (s)**. Repeat this process for frame 75 (i.e. go to frame, move tentacle, key position). To ensure that the position of the tentacle at frame 100 is the same as that on frame1 (for looping purposes), simply **RMB-click over the timeline on frame1** and select **copy**. Move the playhead to **frame 100** in the timeline and **RMB-click to Paste -> Paste**.

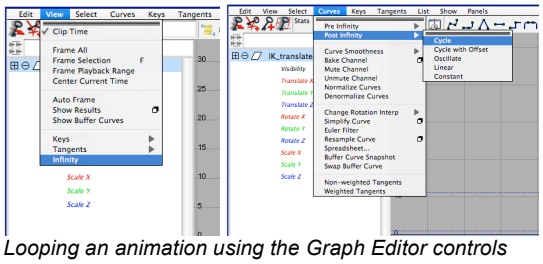


RMB clicking on keyframes in the timeline allows you to copy and paste them.

With the IK_translates node still selected, go to the **Graph Editor** (select one of the interface preset buttons on the left of the Maya interface). Press **F** to frame the view – you should see a graphical representation of all the keyed attributes you set for the IK_translates node over time (frame numbers are on the bottom of the editor).

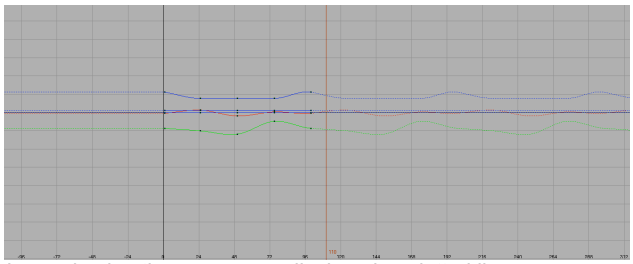

The Graph Editor displays keyed attributes over time for any selected node (in this case the IK_translates node).

If you wanted to loop this bit of animation, here's what you would do… go to the top of the graph editor panel and select **View -> Infinity** (straight dotted lines should appear on either side of your existing curves). Now go to **Curves -> Post Infinity -> Cycle**. Notice what happens to the dotted lines in the graph editor – play your animation and you'll notice that it is now cycling.

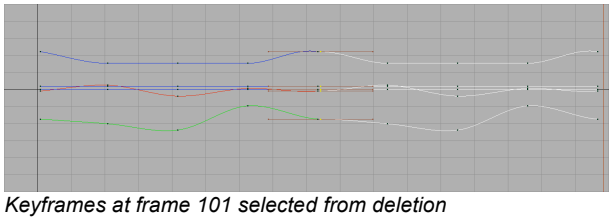


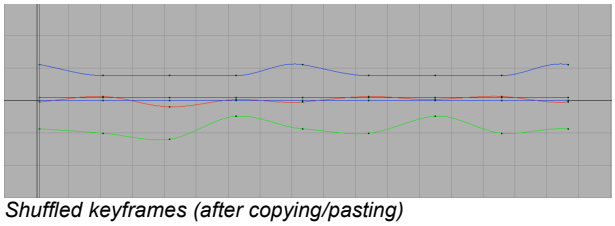
Looping an animation using the Graph Editor controls

For this particular animation, we don't want it to cycle indefinitely (because we want to control its twisting starting around frame 200). Instead, we will copy all the existing keyframes and shuffle them a bit to add variety to the motion of the tentacle before the twisting begins.


Looped animation curves are displayed as dotted line

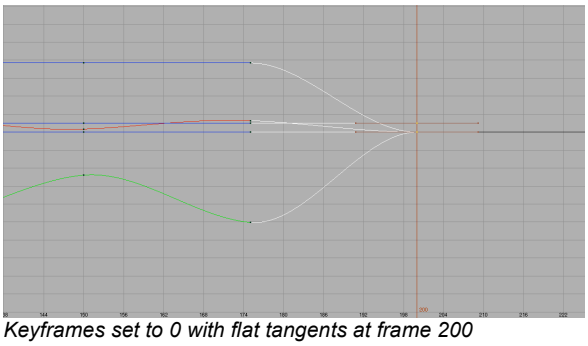
Go back **Curves** and select **Post Infininty -> Constant** and then go to **View -> Infinity** (to toggle it off). **Now marquee-select over the all the keyframes** in the graphic editor – **copy** them (using the keyboard shortcut or RMB-clicking in the interface and selecting Copy). Move the playhead to **frame 101** and **paste**. To clean things up, marquee-select the keyframes at **frame 101** and **delete** them. Let's now shuffle some keyframes around for variety (and practice)… In the timeline, RMB-click over red tick mark at frame 75 and paste it on frame 150. Repeat this by copying frame 25 and paste it at frame 175. Now go to **frame 200**, and in the Channel Box enter values of **0 0 0** for the **Translate X Y Z** of the IK_translates node


Keyframes at frame 101 selected from deletion


Shuffled keyframes (after copying/pasting)

(which should already be selected). To smooth the motion of the tentacles as they approach frame 200, let's **marquee-select** them at **frame 200** in the Graph Editor and set **the tangents to flat**.

Flat tangents tool (Graph Editor interface)

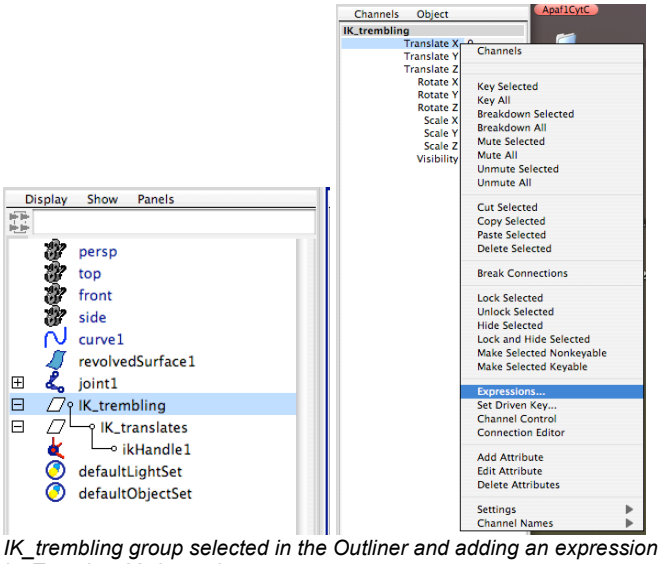


Keyframes set to 0 with flat tangents at frame 200

Now we will add a trembling/shaking motion to the tentacle using 2 different methods – first using expressions and then using an animated texture.

Trembling tentacles - Expressions

The Translate x y z channels of the 'IK_translates' node (which in turns controls the ikHandle1 node under it) already have keys set on them (they are orange in the Channel Box). To superimpose additional motion onto this rig, we will use the same trick as before – **select the 'IK_translates' node** and **group it to itself**. **Rename** that node '**IK_trembling**' (no need to pivot snap that one to the ikHandle1 position). By grouping the group to itself, all we have done is created another Maya transform node that can be manipulated independently.

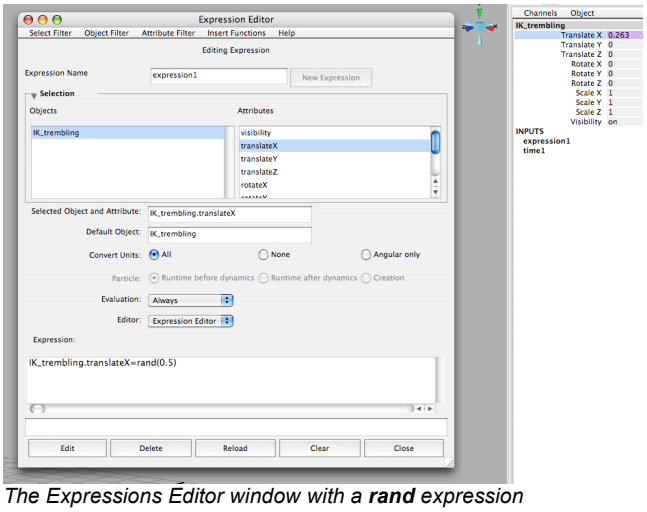


IK_trembling group selected in the Outliner and adding an expression to its Translate X channel.

Select the **IK_trembling** node and, in the Channel Box**, LMB-click the Translate X** channel name to select it and **RMB-click over it** – select '**Expressions'** towards the bottom of the pop-up menu. This will open up a floating window called the **Expressions Editor**.

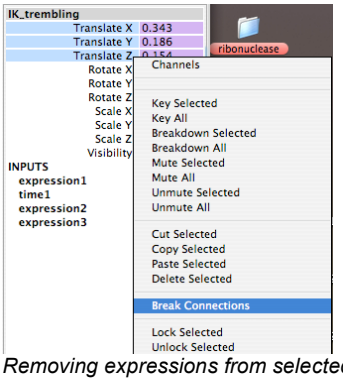
Notice that the window already has the IK_trembling node selected in the left-hand column and the translate X attribute selected on the right. An easy way to start writing an expression, is to select and copy the text in the "Selected Object and Attribute" box and paste it in the white area labeled "**Expression:**".

Type in the following text:
IK_trembling.translateX=rand(0.5)
And click Create at the bottom left. This expression sets the translateX attribute to a random value generated between 0 and 0.5. Play your animation and observe what happens. Now repeat the process to **apply the very same expression** to the **translateY** and **translateZ** attributes. We have some significant shaking going on now.



*The Expressions Editor window with a **rand** expression applied to the Translate X attribute*

Now change these expressions to:
IK_trembling.translateX=noise(time)
This applies a noise function to the frame number as the animation plays along the timeline – it creates a somewhat smoother vibration on the tentacle. You can create many other kinds of expressions using various mathematical operations available to you in Maya (in fact take a peek at the "**Insert Functions**" menu in the Expressions Editor menu set).



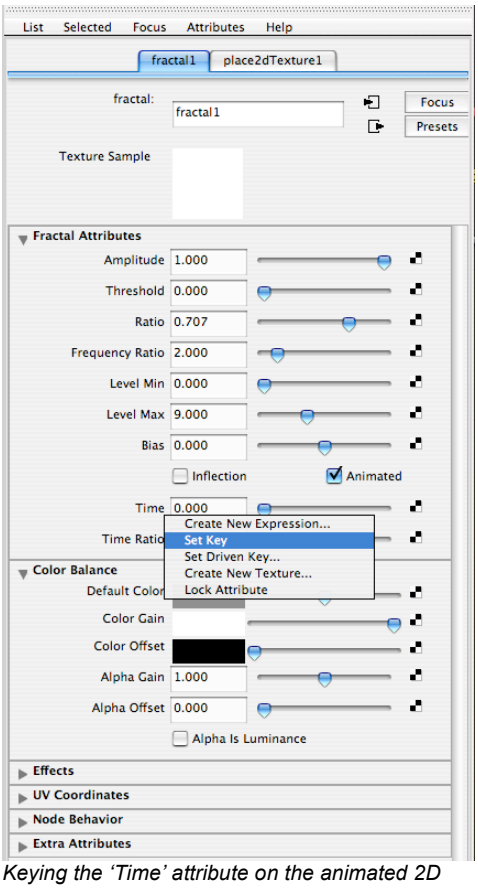
Removing expressions from selected attributes

Before we are ready to apply the next trembling method, we'll need to remove the expressions set on the translate attributes of IK_trembling. To do this, go to the **Channel box**, **LMB-select/drag over the 3 Translate attribute names**, and **RMB-click** – select '**Break Connections**' from the drop-down. Notice that the box values go from being **purple** (which indicates expression-driven values) back to the default **white**.

Trembling tentacles – Animated texture

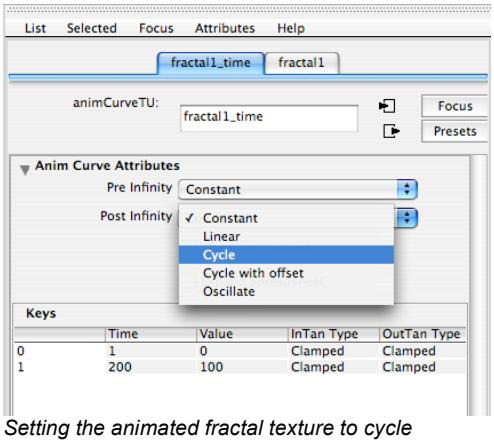
The idea behind this method is to use the great variety of greyscale values present in a texture (like a fractal) and pipe those into translate attribute values of a node. Remember that although these textures are typically used as surfacing tools, they are just numbers to Maya and if we can connect them appropriately to our geometry, they too can drive motion.

Open the **Hypershade** and create a **2D fractal** texture (using the vertical menu on the left). 2 nodes should appear in the work area of the Hypershade (if you only see one zoom out a bit – chances are the 2D placement node is off screen by default). **Double click on the fractal** node to open up its attributes in the Attribute Editor (on the right of the interface). The first thing we need to do is make the texture animated (i.e. the fractal pattern you see in the little preview box needs to change over time). Click on the '**Animated' checkbox** – this makes the Time and Time Ratio controls (just below) available to you. We will key the value of Time over the range of your animation – start at **frame 1** and **RMB-click** over the word **Time** in the Attribute Editor **-> Set Key** (the box turns orange). Now go to **frame 300** in the timeline and, back in the Attribute Edtior, move the **Time**



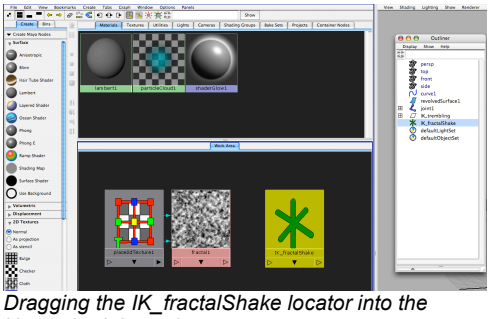
Keying the 'Time' attribute on the animated 2D Fractal texture in the Attribute Editor.

attribute slider all the way to the right (value of 100), and **RMB -> Set Key** again. In case we expand out timeline beyond 300 frames later on, it would be nice to know that the texture pattern will remain animated – to do this, look at the 'Time' attribute in the Attribute Editor and notice that the usual checkered square all the way to the right has turned into a different icon (**right-pointing arrow over the left side of box)**… for any attribute, this means that it is being driven by another node (said differently, that attribute has an input piping information into it – in this case, it's the fractal1_time node that was created the moment you keyed the Time attribute). **Clicking this little icon** takes you to the **fractal1_time** node attributes (could also have used the tab at the top of the Attribute Editor) – under **Post Infinity** select **Cycle** (this is the same idea as what we did in the Graph Editor).

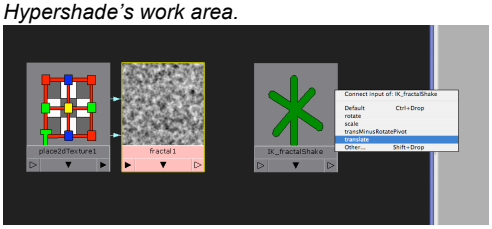


Setting the animated fractal texture to cycle

Now lets create a locator (**Create -> Locator**) and rename it '**IK_fractalShake**.' We will be transferring the fractal texture values into it and then parent the geometry (IK_translates) to the locator. Have both the Outliner and Hypershape windows open side by side – MMB drag the locator **IK_fractalShake from the Outliner to the Hypershade** work area. In the same way that we connected nodes in the Hypershade before to create shading networks, **MMB drag the fractal node** onto the **Locator node** and select '**translate'** from the menu that appears. By default this grabs the fractal's 'outColor' values and pipes them into the translate values of the Locator. Go to perspective viewport and play – close to the origin, you should see a locator (green cross) moving around erratically.

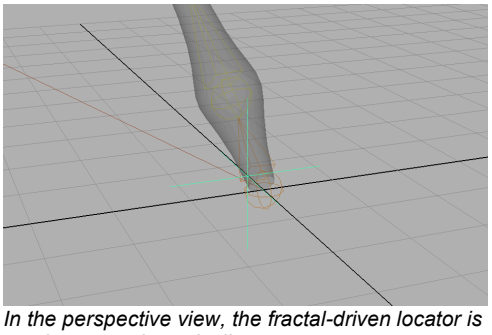


Dragging the IK_fractalShake locator into the Hypershade's work area.

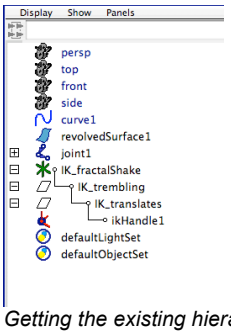


Connecting the fractal texture to the Locator's translate attributes

Now all we need to do is get the ikHandle of your skeleton to 'see' that locator and be driven by its motion. Just to review: the ikHandle is grouped (and driven) by the IK_translates node (onto which you placed a number of keys). After our last exercise, we grouped IK_translates and created yet another transform node above it in the hierarchy called IK_trembling (but we deleted the expression-driven motion from that group's translates). So now all we need to do is **MMB drag the IK_trembling group** onto the **IK_fractalShake locator node** in the Outliner. Now play the animation and… once again, you have a trembling sigma1 tentacle. However, this time you can edit your texture attributes to



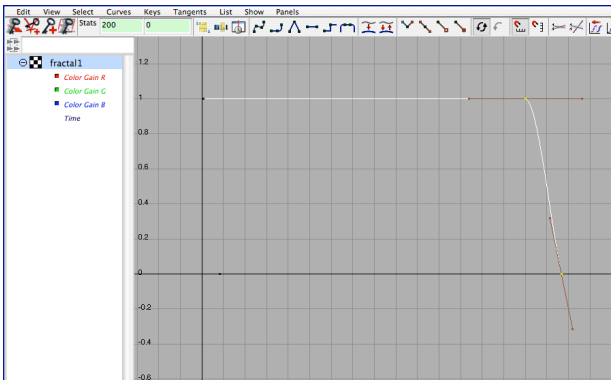
In the perspective view, the fractal-driven locator is moving around erratically



Getting the existing hierarchy to be driven by the locator

change the trembling properties – a much more powerful and controlled method.

While you keep your animation playing, go back to the **Hypershade** and **double-click the fractal** to gain access to its attributes. Now try sliding the Amplitude down to 0 and see how this affects the motion – move the Amplitude to 10! OK – back to 1. Experiment with the color gain value as well (in the Color Balance panel). In fact, it might be nice to reduce the amount of trembling as our tentacles prepare to rigidify and twist – so lets do that by setting some keys on the color gain attribute of the texture. Go to frame 1 and RMB-click on the word 'Color Gain' in the Attribute Editor -> Set Key. Now go to frame 180 (where we might want the trembling to start quieting down) and set another key. Finally, go to frame 200 (where the twisting will start), set the Color Gain to black and key again. So the trembling will remain constant between frames 1 and 180, decrease between frames 180 and 200, and stop on frame 200.

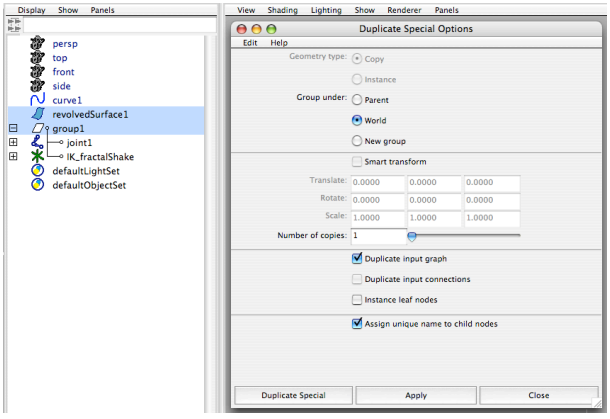


Keys set on the Color Gain attribute of the fractal texture will keep trembling constant up to frame 180 and then decrease to a stop on frame 200 when we will have the tentacle twisting begin

Duplicating/moving rigged geometry

Now we want to create 2 copies of our tentacle – move them around, and rotate them a bit… This can get pretty confusing – the basic concept to remember is that any moving of skinned geometry is now controlled by the skeleton it is bound to – so even if you want to move the position of the entire object (not the position of the joints within it) you need to select the skeleton (i.e. the root joint) and move that - the geometry will move along with it. To duplicate an entire rig with animation, we need to invoke the Duplicate Special command in Maya which allows us to duplicate objects with all inputs maintained. If you just selected your geometry and did a simple duplicate operation you would be left with an inanimate tentacle. We want to duplicate the geometry with its skeletal rig and all the great motion we have been applying to it!

Select the joint chain (simply by clicking on the root joint1 in the Outliner) **and shift select the IK_fractalShake group** – **group them**. Now select this new group and shift-select the geometry (still called revolvedSurface1). Now go to **Edit -> Duplicate Special | options**. Enter the parameters that are shown in the image on the right – among other settings, the key here is to have the check box "**Duplicate**



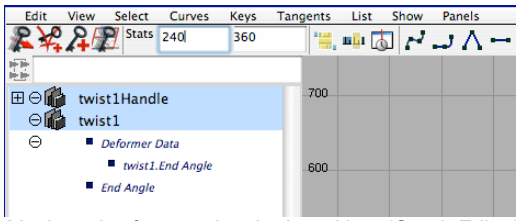
Duplicate Special options window

Input Graph" checked. This is what maintains all existing connections in the duplicated nodes. Click **Duplicate Special** – and look at what you have in the Outliner… 2 new nodes: a geometry revolvedSurface2 node and a new group containing all of the nested groups (ultimately controlling the motion of the ikHandle) and a new joint chain. **Select this new master group** and **move** the new tentacle to a different position on the xz plane (**rotate** it as well so that it is facing away from the other one).

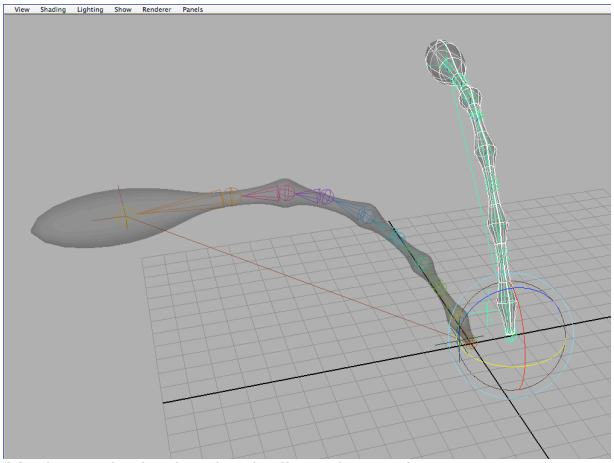
Repeat the selection and duplicate special process using the first tentacle to create a third – position and orient that one as well (use the image on the right that shows al three tentacles to guide you). Now play your animation and you should see all three tentacles moving around at first and then rigidifying around frame 200 and remain still until frame 300.

Now **select all three revolvedSurface** nodes in the Outliner and, go to **Create Defomers -> Nonlinear -> Twist**. Go to frame 200 in the timeline. Now select the twist1Handle in the Outliner – and in the Channel Box, click on the twist1 (last line under INPUTS) – this should expand and reveal the twist deformer keyable attributes. End Angle should be on 0 by default – **select the End Angle attribute** name (click on it) and **key its value (0) on frame 200** (RMB-click over the attribute name). Now go to **frame 300** in the timeline, and enter a value of **360** for **End Angle** – key this value.

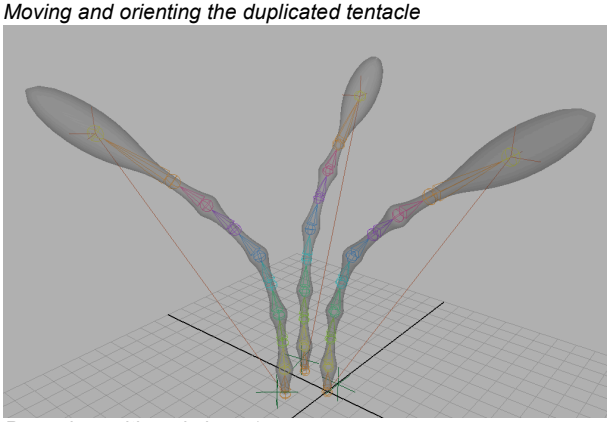
To smooth out the ending motion of the twist (i.e. to have an 'ease-out' effect applied to the motion) go the Graph Editor, **marquee-select the key at frame 300** and make its **tangent flat** (remember the icon we used at the top of page 6). Finally, it seems that the twist is happening a little slowly relative to the previous flailing motion so we might want to have the twist start at frame 200 but end sooner (say at frame 240). We can easily do this by **marquee-selecting the keyframe at 300** in the Graph Editor and now **type 240** in the **'Stats' box** in the upper left of the Graphic Editor interface (see image below). We're done!



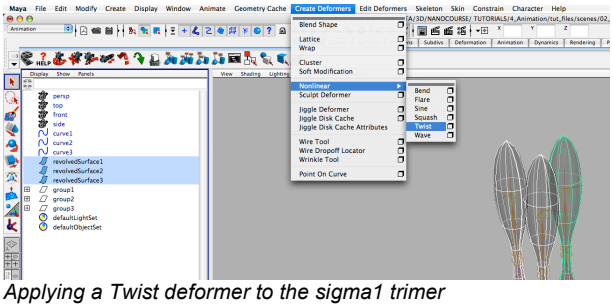
Moving a keyframe using the 'stats' box (Graph Editor)



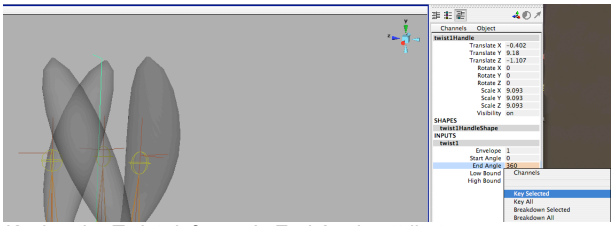
Moving and orienting the duplicated tentacle



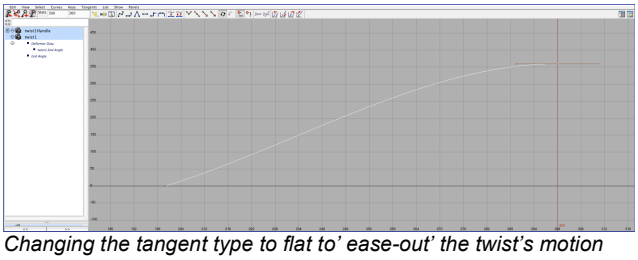
Properly positioned sigma1 monomers



Applying a Twist deformer to the sigma1 trimer



Keying the Twist deformer's End Angle attribute



Changing the tangent type to flat to' ease-out' the twist's motion