Intro to Rhinoceros and Grasshopper

1 Rhinoceros

You can download a copy of Rhinoceros 8 from `http://www.rhino3d.com/download/` (Mac and Windows versions are available) and use the 90 day free trial license. You can download this document and example grasshopper files at `https://math.okstate.edu/people/segerman/talks/workshop_files.zip`.

1.1 Navigating the viewpoints

- Open the Rhinoceros program and start a new model. Four viewpoints are shown: three orthogonal views, the top, front, and right views, and a perspective view.
- On a Mac with a touchpad, on any window: pinch with two fingers to zoom. Drag with two fingers to rotate the perspective view, or Shift+drag with two fingers to translate.
- On both Windows and Mac with a mouse: Shift+hold right click and drag on any of the windows to translate that view. Shift+Ctrl+hold left click does the same thing. Scroll wheel on any of the windows zooms the view. Hold right click and drag to rotate the perspective view. Ctrl+hold left click does the same thing.

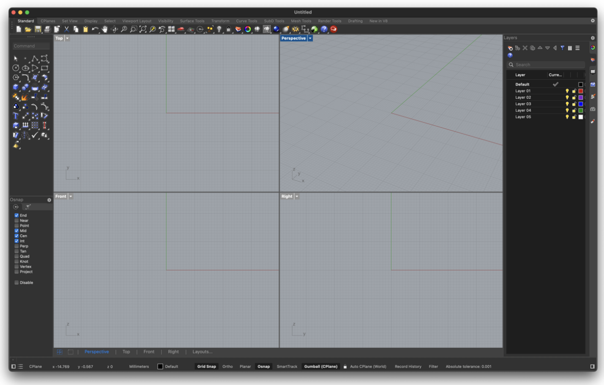

Figure 1: Default Rhinoceros screen.

1.1.1 Menus and Options

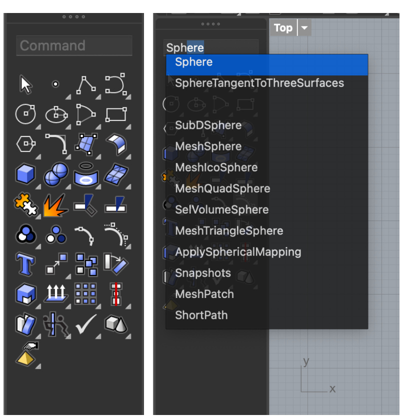

Figure 2: GUI commands can be accessed by clicking or by typing.

There are many options for customizing windows in Rhino. You should have a panel of tools on the upper left that has buttons with many built-in functions (and/or menus if you click and hold). You can also start typing the name of a function and the autofill will show a list of suggested functions. In the example, “sph” autofills `Sphere`, but also brings up other function suggestions that contain the letters “sph”.

The “Object Snaps” panel on the lower left has a list of different objects the cursor can snap to when you are selecting points. This helps when you are designing parts because you can rely on parts of existing objects (such as the end of a curve, the midpoint of a line or the center of a circle) to build new structures off of. The ones I usually have on are in the accompanying figure, but you may find another set work for you.

There is a side panel on the right that can show various pieces of information. On the far right you can click different icons to see different panels. The ones I use most are the “Properties” (the colorful circle icon) and “Layers” (the slice of cake icon) panels. The Properties panel is context sensitive. When no object is selected, the “Viewport” properties will show up. If an object is selected, then its properties will show in the panel. The most important of these is the “Type”, which will say something like “closed surface”, “open curve”, “open polysurface”, or “closed extrusion”. This

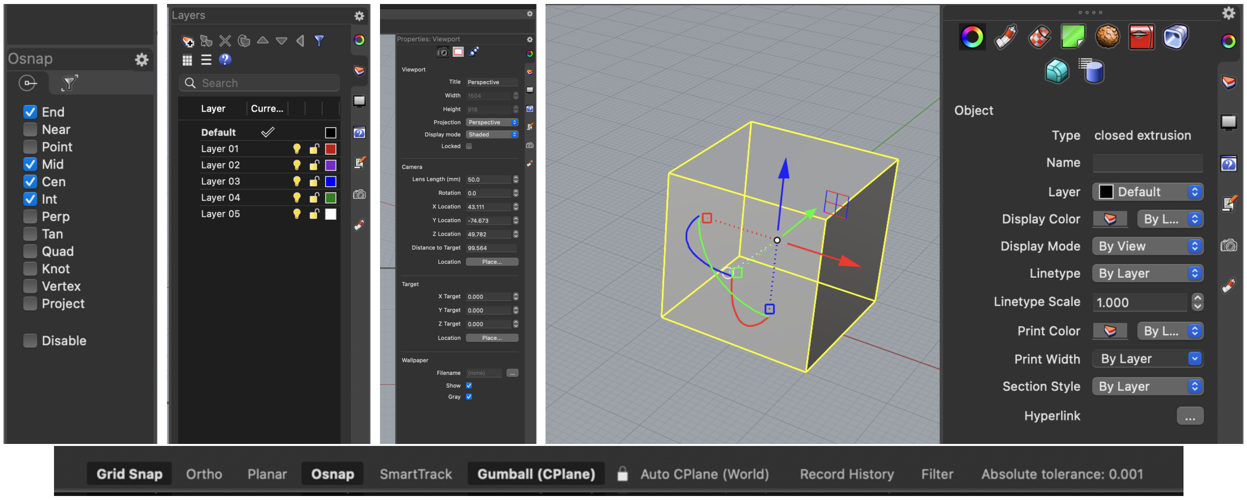

Figure 3: Various panels.

information is very useful both for exporting objects for 3D printing and combining them together. For example, you need to have closed objects to use Boolean operations, but open objects to join them together into a closed curve or closed polysurface. 3D printer slicer software cannot reliably deal with objects that are not closed.

Each object lives on a *layer*. In the Layers panel, you can choose to show or hide each layer, and assign colors to the objects on a layer. This is very useful for complex designs. To move or copy an object to a layer, select the object you want to move object then right-click the name of the layer you would like to move it to. Select `Move objects to this layer` if you'd like to move them and `Copy objects to this layer` if you'd like to duplicate them.

At the bottom of the screen is a horizontal bar of toggles – options that you can turn on or off. These include `OSnap` (which turns Object Snaps on or off). You will want to make sure `GridSnap` is on. (Some people may find `Gumball` useful as well.) I leave `GridSnap` on almost 100% of the time. Between this and the `Object Snaps` menu, you will be able to create very precise objects.

1.2 Basic Objects and Booleans

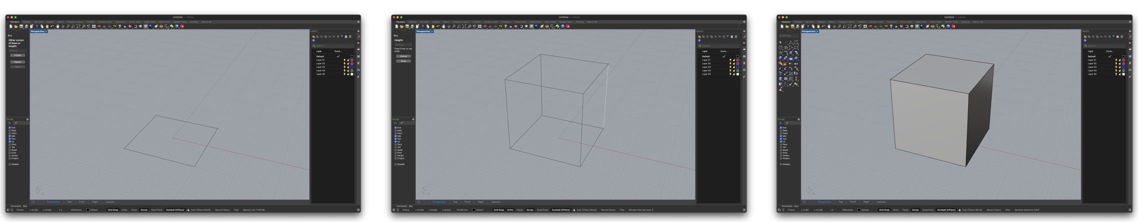

Figure 4: Building a cube.

- Let's begin by making a cube.
- Set your view to "Shaded". You can select this from the dropdown Views menu or on a Mac the using the hotkeys Command+Ctrl+S.
- Click on the cube shaped icon in the panel on the upper left. This function is called `Box: Corner to Corner, Height`. (Many objects have different ways to create them, e.g. `Circle: center, radius`, `Circle: tangent, tangent, radius` or `Circle: 3 points`. These are often indicated by the a little triangle on the bottom right corner of the button.)

- As with most Rhino functions, the panel on the upper left will be replaced by a list of steps to guide you through implementing the function. (On Windows, these steps appear in a horizontal bar near the top of the screen.) Here you will select the one corner of the base, then the diagonal corner of the base and lastly you will click the height. Your cube will appear as a wireframe until you make it a complete solid object, then it will display with the faces shaded in.

- For any inputs that take numbers or coordinates, you can type values directly into the dialog box rather than clicking the screen.

- You can change between views to make sure your selection is where you want it to be. As you would expect, the Perspective view shows a perspective view of the object. The other three views, Top, Front and Right are orthographic, meaning that everything is orthogonally projected onto the view plane.

- If you select the cube after you have created it and switch the right panel to show properties (the colorful circle icon), the panel will say `Type: closed extrusion`.

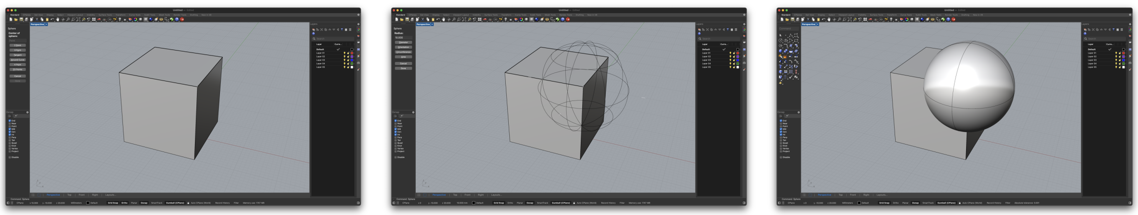

Figure 5: Add a sphere with radius half the side length of your cube, centered at a vertex.

- Next we will add a sphere to the corner of the cube.

- In the "Object Snaps" window, make sure that "End" and "Midpoint" are selected.

- Begin typing "sph". Select `Sphere` from the options that appear in the `Command` dialog box. This will default to the option "center, radius".

- Select the center point of the sphere by holding your cursor over one of the vertices of the cube. One the outline of one face of the cube will thicken and be visible through the rest of the cube. A text box will appear that says "End". Click.

- Next you will need to select the radius for your sphere. Hold your cursor over one of the edges adjacent to the center of the cube. Again cube face will be highlighted, and the text box will read "Midpoint". Click.

- The sphere will now go from wireframe to shaded. If you select the sphere and look at the Properties panel, the Type will be "closed surface".

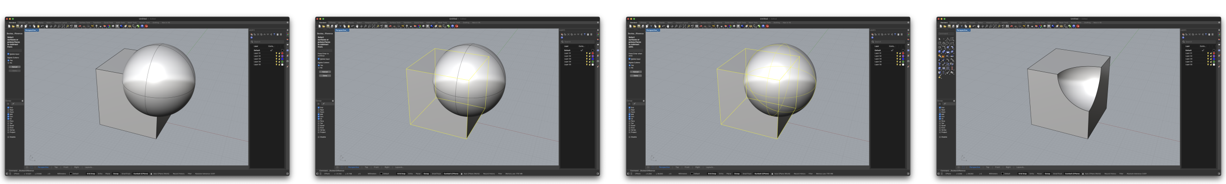

Figure 6: Use Boolean Difference to cut a bite out of your cube.

- From the top left tool panel, click and hold the button with two overlapping spheres (this is the icon for `Boolean Union`). A new grid of tools comes up – select the second tool on the top row (it looks like a solid sphere with a darker "negative" sphere above it). This is the `Boolean Difference` tool. You can also get to this command from the menu at the top. It is under the "Solid" menu, and there is called "Difference". (From now on we will use the shorthand Solid → Difference to denote a series of menu choices like this.)
- The panel will ask you to select the surfaces to subtract from. Click the cube. Hit the "Enter" key to finish selecting objects to subtract from. (If you had the cube selected before starting the Boolean Difference then Rhino assumes you want to subtract from the cube and this first step will be skipped.)
- The next step is to select the objects that you would like to subtract with. Now click the sphere. Hit the "Enter" key.
- You should see that your cube has a bite taken out of it.
- Now that you have an object, try switching to `Rendered` view style (Command+Ctrl+R). Try out the other viewing styles from the View menu (from the bar at the top of the screen).
- Other things to try: Transform → Move, Transform → Rotate, Transform → Scale → Scale 3-D. Follow the instructions in the pop-up window that appears.

1.3 Exercise: Building a right angle tetrahedron

- File → New or Command+N starts a new file.
- We will be building the edges of a tetrahedron with vertices at $(0, 0, 0), (20, 0, 0), (0, 20, 0), (0, 0, 20)$.
- If it isn't already on, turn on Grid Snap, using the button on the bar at the bottom of the screen.
- Build the three lines along the axes first, using the orthogonal viewpoints. Use Curve → Line → Single Line, then left click on one of the viewpoints at the start and endpoints to define the line. (Space bar repeats the last used command, which may be useful here.)
- For the other three lines, use the perspective view, and make sure that Object Snaps ("OSnap") is turned on, and snapping to "End" is enabled. When drawing the new lines, the mouse will snap to the endpoints of the existing lines.
- To correct an error, either Edit → Undo, or Command+Z to undo, or alternatively left click to select an object and Backspace to delete. Left click and drag a box to select multiple objects. Shift+left click adds objects to the selection and Command+left click removes objects from the selection.

We now need to thicken these lines up so that we can print the tetrahedron.

- Edit → Select Objects → All Objects or Command+A to select all of the lines.
- Solid → Pipe then Enter to make pipes around the lines. The default pipe radius is 1 (millimeter), but this can be altered in the dialog at the top left of the window. A radius of around 2 millimeters or more should print well.
- (Optional) use Solid → Sphere → Center, Radius to put a sphere at each of the vertices of the tetrahedron, to round off the corners.

- Select all with Command+A.
- Use Solid → Union to boolean union the pipes and spheres together into a single, non-overlapping object.

1.4 Converting to a mesh and exporting for printing

Rhino uses *NURBS surfaces* as well as meshes to describe surfaces. ("NURBS" stands for "Non-uniform rational B-spline". NURBS surfaces are generalisations of Bézier curves.) Both of the designs we have made so far (the cube with a bite and the piped tetrahedron) are *polysurfaces* made from NURBS surface patches. We (most likely) need to convert it from this parametric format into a mesh before sending the data to a 3D printer.

- Select all with Command+A.
- Do Mesh → From NURBS Object, then Enter to convert the polysurface to a mesh.
- The mesh and the polysurface now occupy the same space. To see only the mesh, select it, and then either move it away from the polysurfaces, or put it on a different layer. For either strategy, use Edit → Select Objects → Polygon Meshes (or similar) to easily select one kind of object.
- To move selected objects, either use Transform → Move, or turn on Gumball, using the button "Gumball" from the bar at the bottom of the screen, then use the handles that appear when you select an object.
- To export the mesh as an STL file, select only the mesh, then File → Export Selected. Change the file format at the bottom of the save dialog to STL. Choose a filename, and then click Export. You should then be able to open the STL file in the slicing software for your 3D printer.

2 Rhinoceros and Grasshopper

Grasshopper is the visual programming system built into Rhino. It allows you to wire together components to generate shapes, just like we have done using the Rhino user interface so far. Almost all Rhino commands have a corresponding Grasshopper component. Grasshopper also allows you to use mathematical expressions and incorporate scripts in Python, C#, and Visual Basic.

2.1 Make a parametric curve

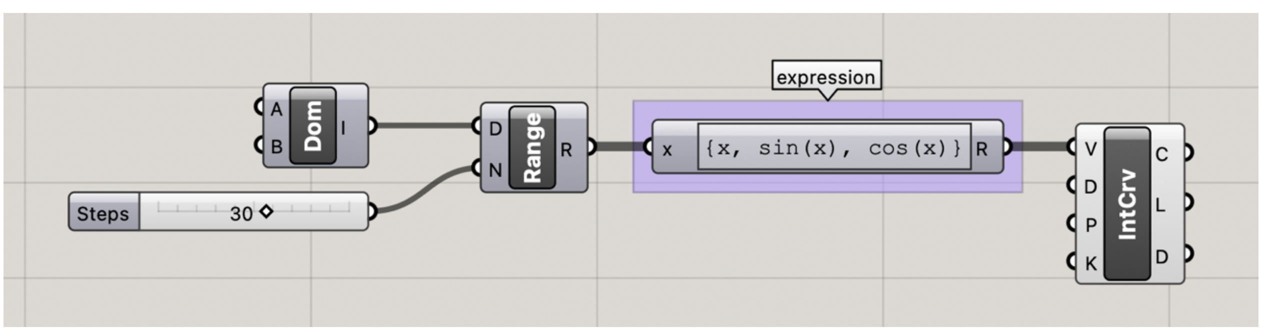

Figure 7: Grasshopper script that produces a helix.

- Command+N to start a new file.

- Tools → Grasshopper will start Grasshopper. Grasshopper appears as a separate window on your desktop. You will want to rearrange your windows so that you can see both the Grasshopper window and the Rhino window.
- File → Open Document, then navigate your computer's file system to find the example file "parametric_curve_example.gh" (in the files for this workshop) and open it. You should see a small red helix in the Rhino window.
- In the Grasshopper window, you should see four components and a slider labelled "Steps". To navigate the Grasshopper window, you can use the same translating and zooming commands as for any of the orthogonal views in Rhino. The component labelled "expression" is the formula for the helix. If you double click it, an editor will open up that allows you to modify the formula.
- The "parametric_curve_example_with_python.gh" file is almost identical, except that the expression component has been replaced by a Python component. Double clicking on that will open up a Python editor, in which you can do more complicated things than the expression component can handle. If you want to add more input variables to either the expression or the Python components, zoom in on the component in Grasshopper and you should see little "+" and "-" icons that will do this.
- The rightmost "IntCrv" component is the one that generates the curve in Rhino. If you left click that component, both the component and the curve in Rhino will be highlighted in green.
- The IntCrv component makes a curve by interpolating through a sequence of points. Those points come out of the expression component, but are not currently visible in Rhino. You can change this by right clicking the expression component and choosing "Preview". In Rhino you should then see the points that it generates.
- The "Range" component produces a list of input values for the expression. The "Steps" slider modifies how many values it produces (try moving the slider and see what happens).
- The "Dom" component determines the domain within which the Range component produces values. It has two inputs, labelled "A" and "B". If you hover your mouse over those inputs, you will see that they have values of 0 and 4.0 * pi. You can modify those values by right clicking on the A or B label and choosing "Set Number...".
- Alternatively, you can make a slider to input a value into one of those inputs. Double click on the Grasshopper background, and a box "Enter a search keyword..." appears. If you know the name of a component you want to add, you can type it in here and get the component. To make a slider, instead write "0.0..10.0". This produces a slider that can output values between 0.0 and 10.0. Left click and drag the dot at the right end of the slider and connect it up with the B input of the Dom component. Now moving the slider will change the end of the domain for the curve, which shrinks or grows the curve.
- You cannot interact with the curve in Rhino using Rhino tools since it is a Grasshopper object. You can convert it into a Rhino object by right clicking the IntCrv component and choosing "Bake...". It will then ask you which layer you want to bake the curve into.
- Curves are one-dimensional objects, but to 3D print something it must have some volume. To thicken up a curve, you can use the Pipe command in Rhino. Then you can Mesh and Export your design, as before. (You will likely want to scale up the design as well, because the default units are in millimeters. A pipe of radius 2mm or so should print well.)

- Alternatively, you can create the pipe in Grasshopper. Double click the Grasshopper background to bring up the search, and type in "Pipe". Then connect the output of IntCrv to the "C" input of the Pipe command. You may also want to set a radius for the pipe component (use a slider connected to the "R" input), and modify how the ends of the pipe are dealt with (right click the "E" input of the Pipe component, and choose between "None", "Flat", and "Round"). For printing, you will want to choose "Flat" or "Round" so that you get a closed surface. You can then Bake the pipe, then Mesh it in Rhino as before. (You could also Mesh it in Grasshopper using the "Mesh Brep" component.)

- If a NURBS object self-intersects (e.g. if you Pipe a self-intersecting curve), then Union will not do anything, and 3D printers may have trouble printing the resulting mesh. To solve this kind of problem, subdivide the curve into non-intersecting pieces before applying Pipe. Use the Round Cap option in the Pipe dialog – this will make the resulting NURBS objects overlap transversely, so that Union will work.

2.2 Make a parametric surface

- File → Open Document, then choose "parametric_curve_example.gh". This draws part of a hyperbolic parabola (scaled up by a factor of 10). Again you can modify the expression (this time broken up into the three individual coordinate expressions then combined) and bake the result into Rhino.

- To thicken the surface up to make it printable, use Solid → Extrude Surface. This will make a closed NURBS object, made out of six NURBS patches. In contrast, applying Pipe to a curve will usually produce a closed NURBS object made out of three NURBS patches: the two end caps and the tube around the curve.

2.3 Joining and exploding NURBS patches

- Delete the thickened hyperbolic parabola (or move it to a different layer). You should still have the two-dimensional hyperbolic parabola surface. Use Transform → Mirror to reflect the surface across the line $x = 10$, making a copy of it (you have to draw the mirror plane, use the top viewpoint to do this).

- Now select both NURBS surface patches and Edit → Join. This joins the two surfaces together along the edge they meet at. We can build complicated shapes up out of NURBS in this way. You may need to change the viewpoint from "Wireframe" to "Shaded" to see the surface properly.

- With the joined surface selected, do Surface → Edge Tools → Show Edges, then click the "Naked Edges" button in the pop-up window. This highlights the boundary ("naked") edges of the surface. To make a surface that we can turn into a mesh and then print, all edges or surface patches must be matched up properly to give a closed surface.

- With the joined surface selected, do Edit → Explode. This is the opposite of Join, and you should see that the edges between the two surface patches are now naked.

- When you Mesh a surface made from NURBS patches that are properly joined together, Rhino produces a mesh that is connected across the join. Without proper joins, the mesh surface will think it has non-empty boundary, and the 3D printer will not know what is inside and outside of the mesh. To check the status of a mesh, open the Properties panel on the right, then select the mesh. The Type of the mesh will tell you if it is closed or open. Surface → Edge Tools → Show Edges also works for meshes, to tell you where the mesh has boundary.