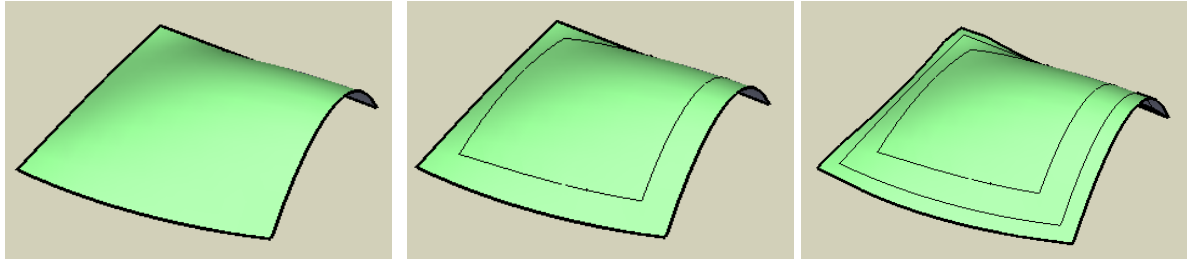


OFFSET CONTOUR ON SURFACE

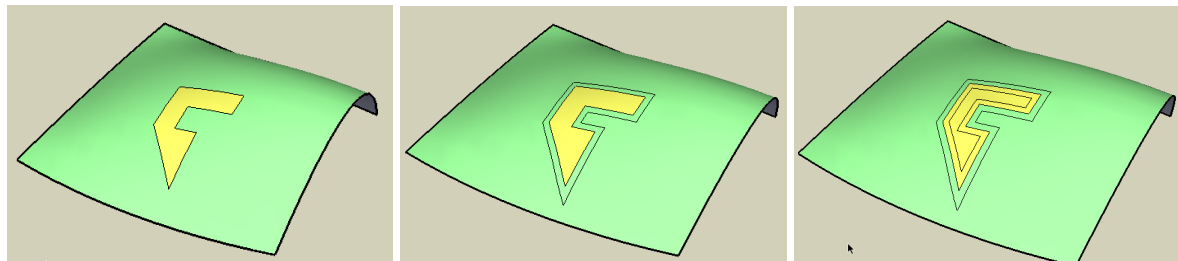
OffsetOnSurface.rb

Foreword - Installation

Offset on Surface tries to reproduce the behavior of the native Sketchup tool Offset on Face, but on a Surface shaped in 3D. Offset direction can be inside or outside.



It also works for inside surfaces on a surface:



Note that there is no mathematical solution for offsetting a contour on a surface in the general case, because there is no strict concept of distance on a surface. So the script works with some approximations and does not support all cases of topology. In practice, it is intended to work on regular and smooth surfaces.

To install the script, you should unzip¹ the file you downloaded (*OffsetOnSurface.zip* in principle) into the Sketchup Plugins folder. This operation should do the following:

- **Copy 3 script files to the Sketchup Plugins folder:**
 - *OffsetOnSurface.rb*, the main script
 - *LibTraductor.rb*, to manage language translation. Make sure you have the latest version (20 Jan 08), also shared with *BezierSpline.rb* and *JointPushPull.rb*.
 - *LibTraductor.def* a configuration file that can be edited to force a particular language
- **Create a sub-folder **OFS_DIR**** (if not already existing), where are stored icons and cursor files (*OFS_....png*) as well as the documentation in PDF format.



The macro will insert 1 new menu item “*Offset On Surface*” in the menu ‘*Tools*’ and create a toolbar with the corresponding icon (actually use the one for Joint Push Pull).

Compatibility: the macro works with **Sketchup v6** and should work in **v5 as of v5.0.260**. I tested it on Windows XP and Windows Vista. It should normally work on Mac, but I did not test it myself.

¹ In Winzip, make sure you do a Select All, and then an extract in the Plugins folder, so that you create or update the subfolderOFS_DIR.

1. Using *OffsetOnSurface*

I tried to mimic as much as possible the behavior of the native Sketchup Offset tool, with its 2 modes of selection, Explicit and Implicit, and some visual feedback. I added however a few options: Isolate Surface (F7), Generate in a Group (F6), Generate Faces when possible (F8), Generate new contours as Curves (F9).

The *OffsetOnSurface* script acts as a classic Sketchup tool, remaining active until you invoke another tool.

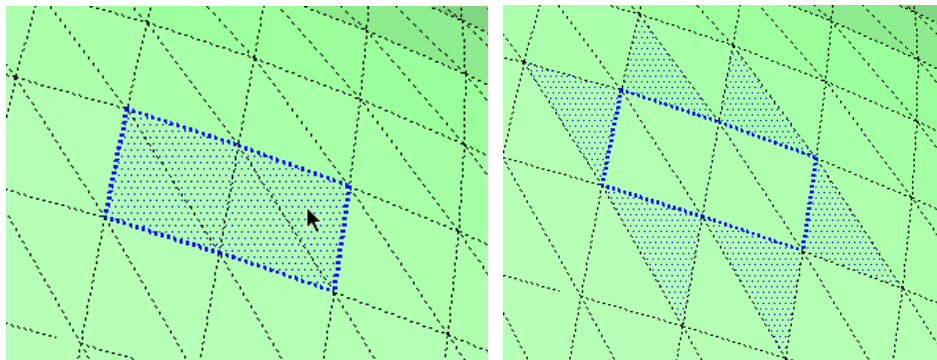
For convenience, the script supports 2 modes of selection:

- 1) **Implicit Selection** → you start the tool with no active selection
- 2) **Explicit Selection** → you make a Face / Edge selection first, and then call the tool.

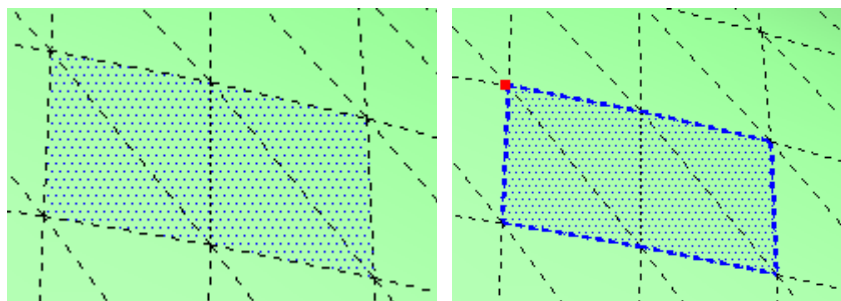
Note however that selection only applies to the active entities of the model. You cannot select a group to mean “all faces of the group”. You must open it first.

1) **Explicit Selection** (Select first, then start the tool)

The script works on a **selection of Edges and Faces** arranged so that you can determine where the inside and the outside are. This is why in practice you would rather select the faces bordering the edges to remove ambiguity. For instance, the 2 selections below define the same contour shape, but oriented reversely.



But if you select just the 4 inside faces, the script will automatically find the corresponding edges:



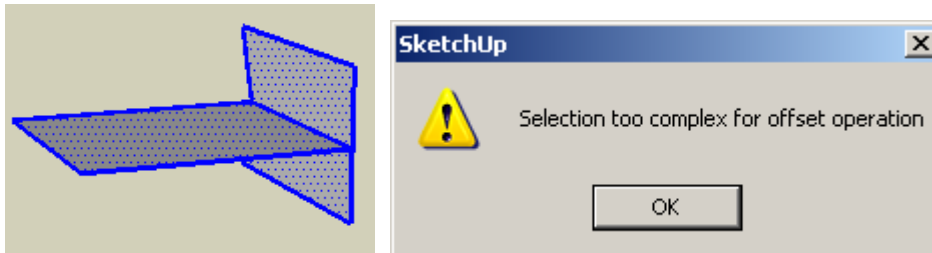
Note that the script will simply ignore all other entities which are not face or edges, that is construction lines and points, groups and components when not open, images, etc...

A selected edge is kept in the selection if and only if it is **bordered by one or two faces** (i.e. not zero, and not 3 or more)

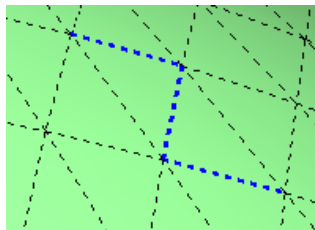
- If it is bordered by one face, then it is not mandatory to select the face
- If it is bordered by 2 faces, then you must select one of the two faces only

If you don't respect the rules relating edges and faces to waive any ambiguity, the script will unselect the edge.

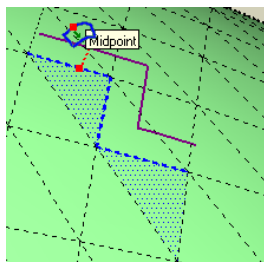
It will also apologies by a message box when the topology is too complex to calculate. This happens in particular when edges are common to more than 2 faces.



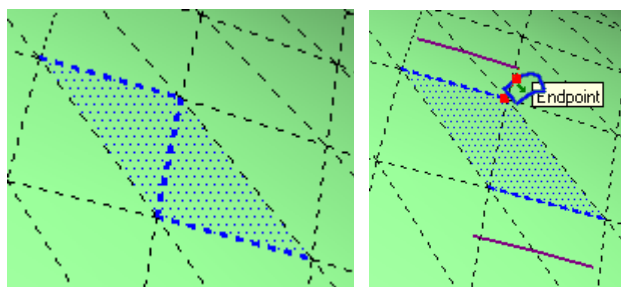
In the Explicit selection Mode you are not obliged to select closed contour. The script should normally works on a few edges. But keep in mind the rule about the ambiguity of edge selection. If you only select edges as below, this will not work:



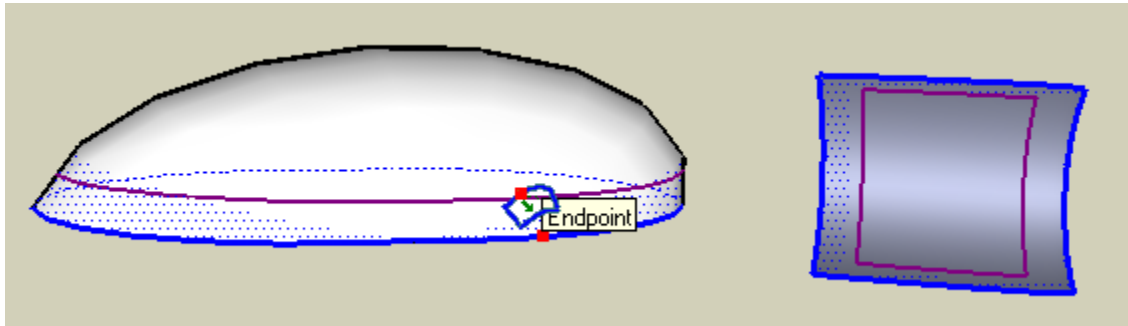
You must select faces to remove ambiguity:



In the case below however, the edge bordered by 2 selected faces will be unselected:



Finally, the selection does not need to contain contiguous contours. If it makes sense for you to select surfaces in different parts of the model, the script will handle them in parallel.

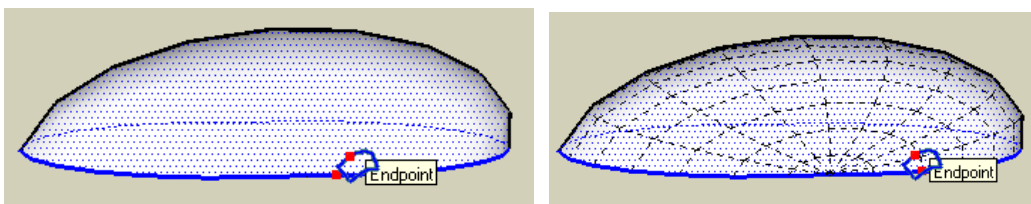


2) Implicit Selection (Start the tool with NO selection)

After starting the tool, you move the cursor over the model to select the surfaces, which will be highlighted.

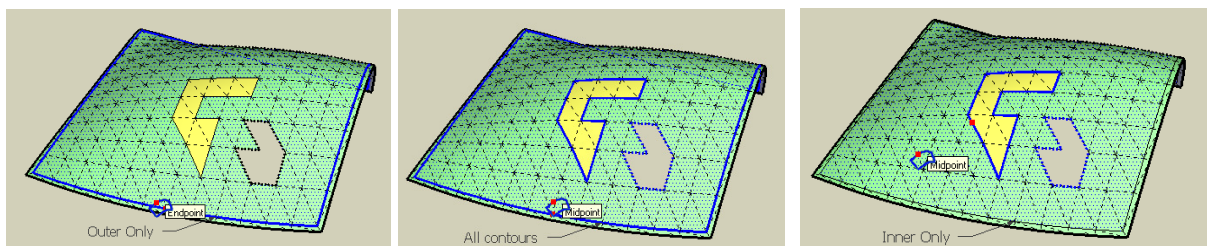
A surface is defined as a set of contiguous faces where Edges are **SOFT or HIDDEN**, as in Sketchup. Edges which are just smoothed will count as borders.

This Implicit selection mode will work the same whether you are or not in the mode “*View Hidden Geometry*”.



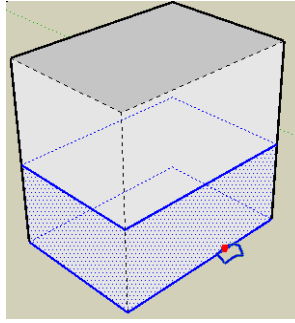
In the Implicit Selection mode too, there are ambiguities because a surface can have actually one outer contour and one or several inner contours (usually because of holes). As there is no way to decide what you wish to do, the macro provides 3 modes of automatic selection, accessible via the **contextual menu** or the **toggle key F5**. The current mode is shown in the status bar:

- 1) **Outer contour** → this is the default
- 2) **All contours**
- 3) **Inner contour(s) only**



Of course, you can always switch to the Explicit selection mode to select the exact contour you want.

Note that in some very specific cases, it is not possible to decide whether a contour is 'inner' or 'outer'. If this happens, then the script selects both contours.



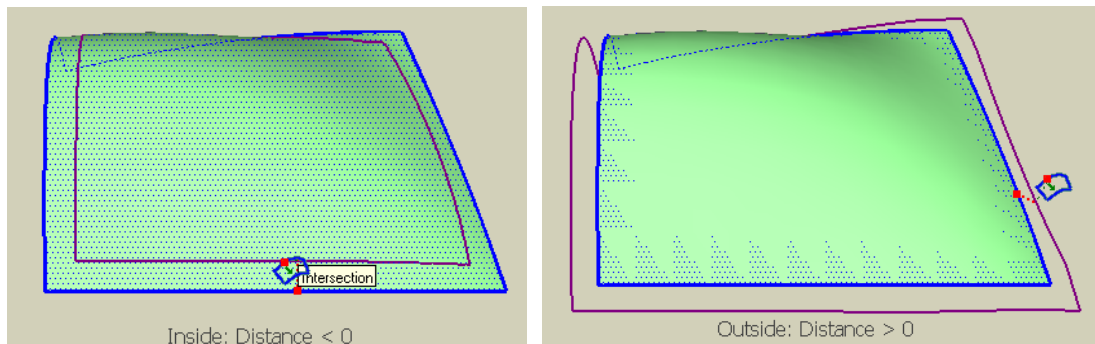
3) Distance of Offset

When you have a surface selected, resulting either from an implicit or explicit selection, you will see a **Red Dot** on the contour. By **clicking and dragging** (or clicking, releasing and moving), you can offset the contour with visual feedback. Note that the path of the Red Dot (in red dashed) will follow the surface. By pressing **Escape**, you would go back to the surface selection. By keeping **Shift** depressed while dragging, you will skip inferences.

The main parameter of the script is the **Distance of Offset**, which by convention² is:

- **Negative** when you go toward the **inside** of the contour
- **Positive** when you go toward the **outside** of the contour

Note that the convention Inside / Outside is irrespective of the orientation of faces.



At any time, you can type the value of the distance in the VCB. The convention is the following:

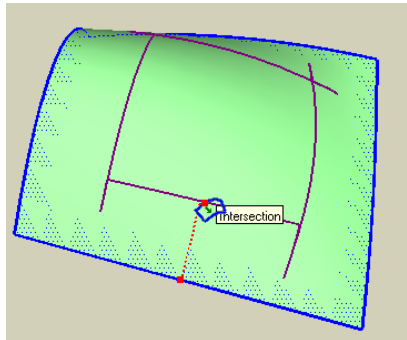
- **If you did not move the Red Dot yet**, then the sign of the value will tell whether the offset is performed toward the Inside (negative distance) or toward the Outside (positive value).
- **If you moved the Red Dot in a particular direction**, then a positive value will change the distance but keep the orientation, whereas a negative value will also reverse the direction.

In all cases, the offset operation will execute after dragging the contour or typing a value in the VCB (as in Sketchup). So no double-click here!

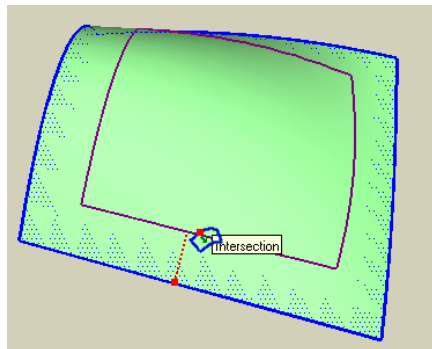
² The native Sketchup Offset has rather strange convention, where the sign of the distance entered in the VCB is relative to the offset already entered.

4) Simplifying Contours

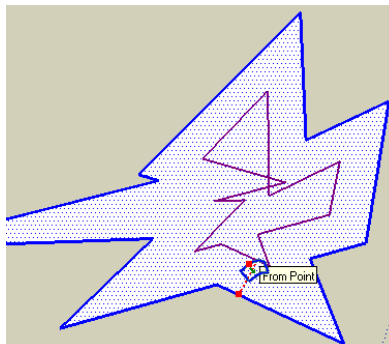
One of the real problem of an offset operation is that, if you want to be 'geometrically correct', then in most cases you will not obtain the desired result, as in the example below.



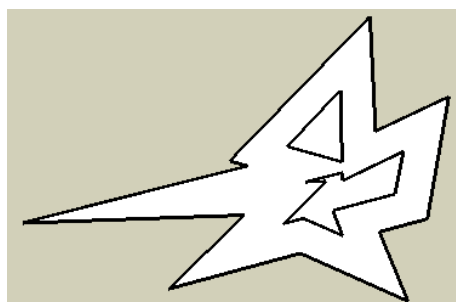
This actually also happens on plane surface with the native Sketchup Offset tool. The script applies a simplification algorithm to make sure that vertices are eliminated when appropriate. The same offset would then give the following result:



However, the script does not handle (yet) situations where the generated contour crosses itself, as in the following case:

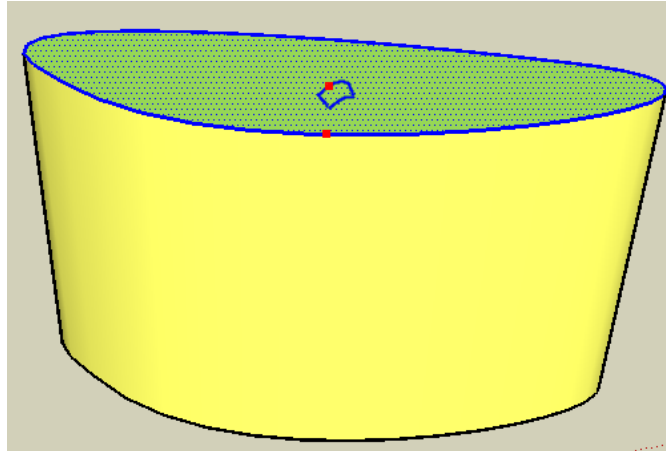


The correct result is shown below. You can of course correct it by hand, noting that erasing the edges of the generated contour will normally not erase the underlying faces.

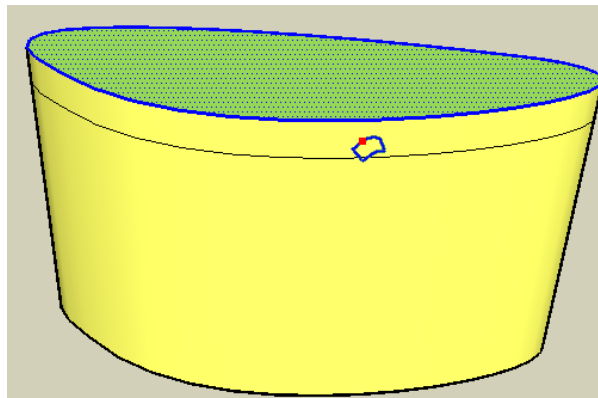


5) Isolating Contours on Surface

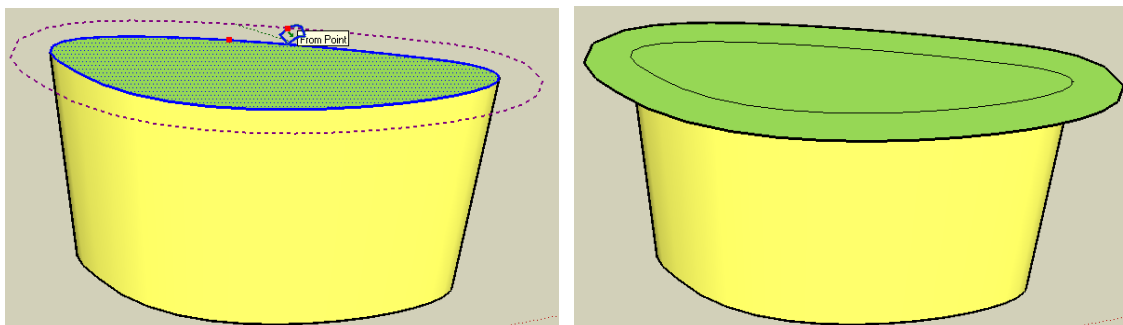
When you offset a contour which is partially or totally embedded within a larger surface, you have the option to ignore the rest of the model. This option may not be obvious to you in the first place, but consider the following case, where you select the top face:



If you offset the contour toward the Outside, it will actually spread on the lateral faces (which may be what you want by the way!):



To avoid this behavior, you can instruct the script to ignore the rest of the model and do as if the face was 'standalone'.

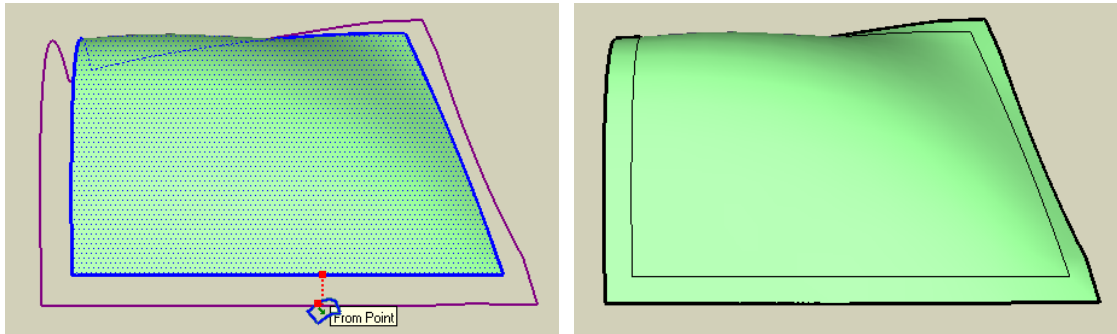


Note that the visual feedback of the offset in 'standalone' mode is in **dashed dark green** (instead of solid purple when this mode is not activated).

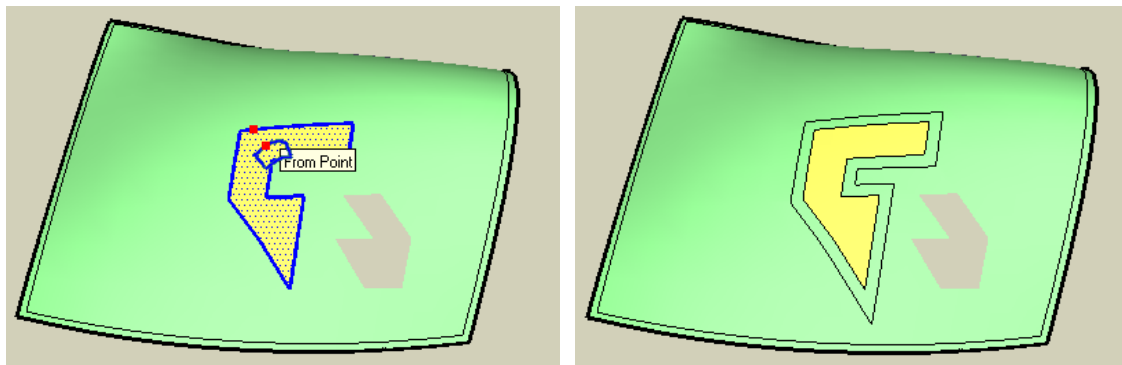
By default the script treats all selected surfaces in a non-isolated mode. You can however activate the standalone mode via the contextual menu or the **Toggle Key F7**.

6) Generate Surfaces

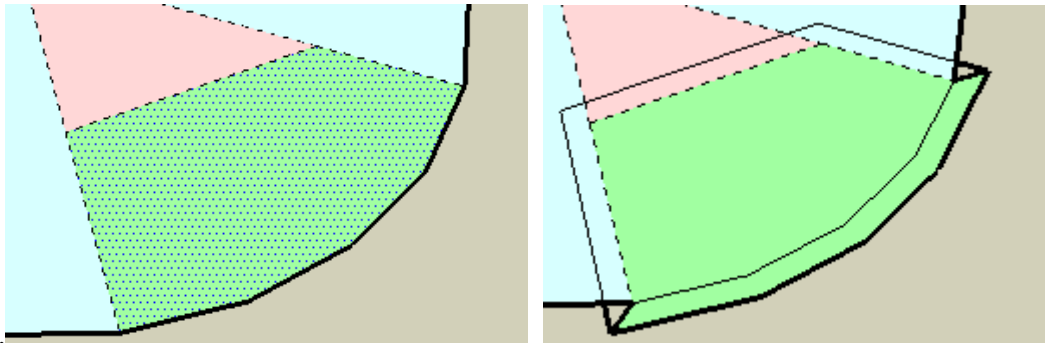
When you enlarge the contour (i.e. distance > 0), the space between the original and generated contours can be filled by a surface (as in the native Sketchup Offset tool).



This will happen only for edges that are truly at the border of the surfaces, that is, bordered by only one face. If the surface is itself within a larger surface, there will be no face generation.



As usual, you may find more complex topologies where you'll get a little bit of both!

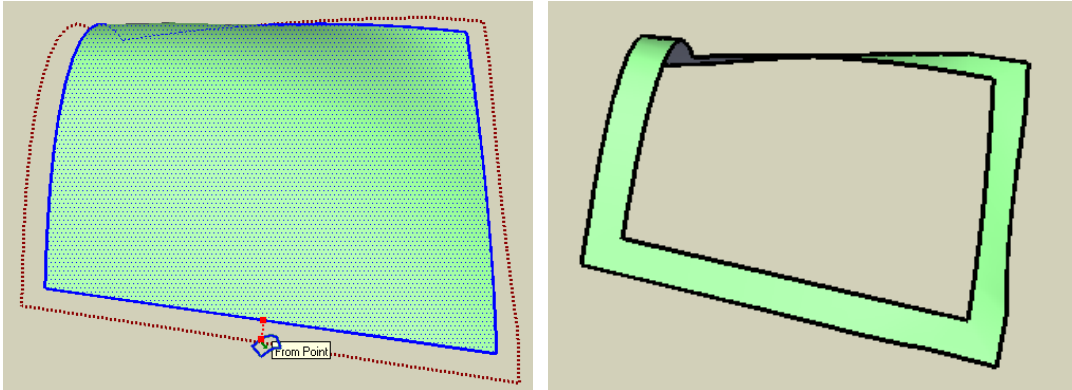


Note that the script will try to remove coplanar edges between faces (unless materials are different). This sometimes does not work correctly, as Sketchup is very sensitive to exact co-planarity.

By default the generation of surface is *On*. You can however deactivate it via the contextual menu or the **Toggle Key F8**.

7) Generate as a Group

For convenience, the offset contour and the generated faces can be generated as a Group, which you can then manipulate independently or explode. Note that the visual feedback of the offset is in **dotted dark orange** (whereas it is in solid purple when no group is generated).



By default the generation as Group is *Off*. You can however activate it via the contextual menu or the **Toggle Key F6**.

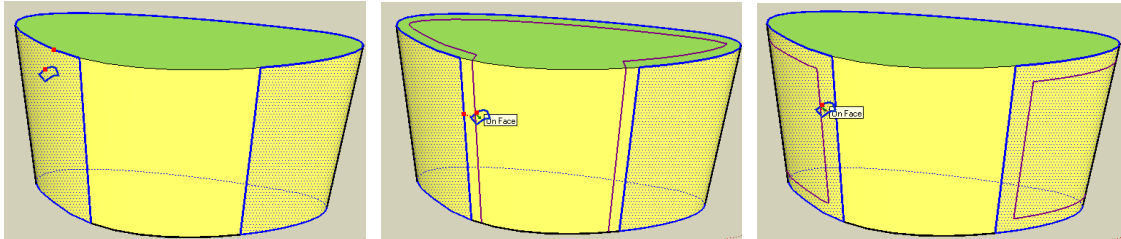
8) Generate Contour as Curves

The offset contour(s) can be generated as welded curves, which are sometimes easier to manipulate. Note that the visual feedback of the offset is in **dashed dark orange** (whereas it is in solid purple when no group is generated).

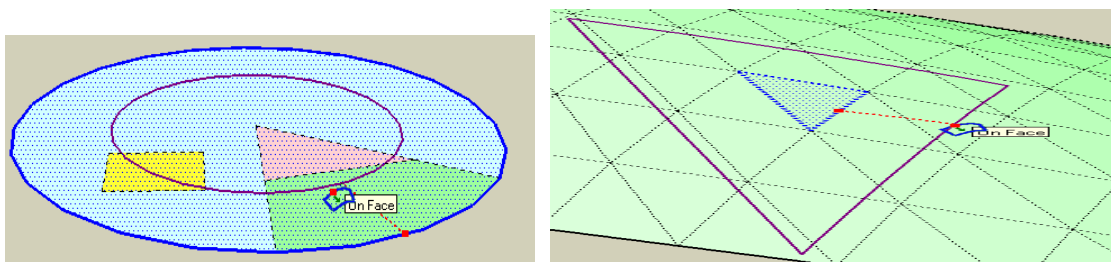
By default the generation as curves is *On*. You can however deactivate it via the contextual menu or the **Toggle Key F9**.

2. A few examples

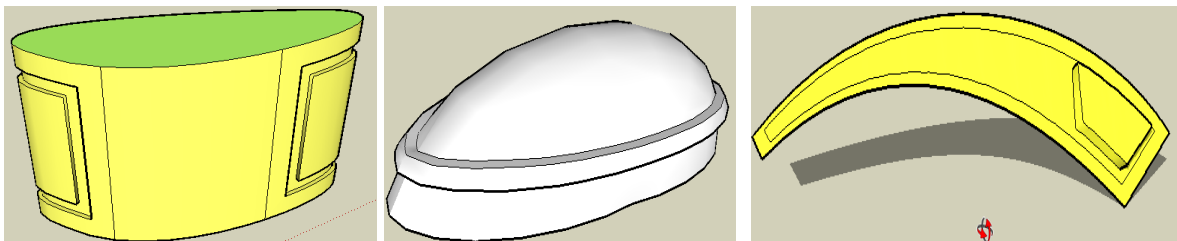
Here are a few examples of what the script can cover



Even of a plane surface, the Tool will allow you to perform Offset operations where Sketchup will refuse, because of the multiple faces.



You can of course combine the Offset On Surface tool with Joint Push Pull.



3. Known Problems, Limitations, Caveats

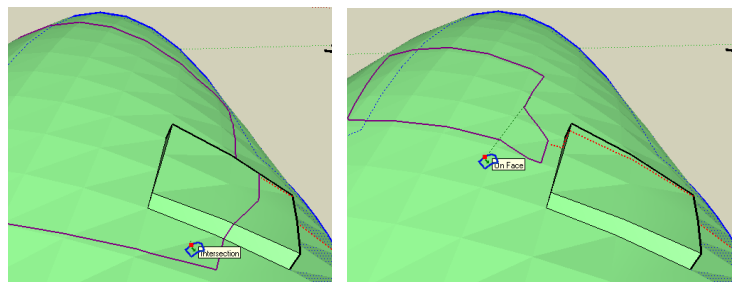
Now the bad news!

There are many limitations and known problems with this version of the script. Not to mention the unknown ones and bugs

1) Path on Surface

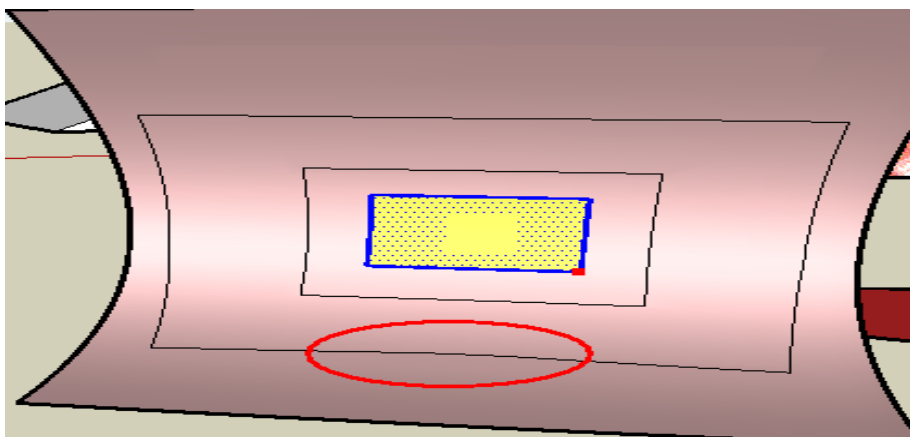
There is no real exact algorithm for offsetting edges on a surface. Actually, the script moves each vertex along the average vector of the normals to its edges with the given distance. So, the script physically reconstructs the path of each vertex along this direction on the surface, which indeed depends on which kind of 'hurdles' it will find on the path.

For instance, if your surface is not really regular, like the one below, some vertices will have their path impacted by the bumping square. It will become even worse when you pass the bumping square, as the script as not the same intuitive approach to 3D topology as we, human beings, have!



So, as a general rule, you should apply Offset On Surface on rather regular surfaces, which can be 'draped' by a plane.

The same happen for the deformation of geometric figures. In theory, it would be possible to preserve the angles from the original contour to the offset contour. In practice, it is difficult. The current script uses a simple algorithm which does not guarantee the preservation of angles. I have in stock a second algorithm that would (in most situations), but which is more complex and time-consuming. For instance, the rectangular shape is not exactly preserved when the offset distance is too large:



2) Performances

The script is actually sensitive to the number of edges in the faces of the model (i.e. not just the ones selected, but the one constituting the surface on which you offset the contour). The reason is that there is no other way than to explore the intersection of the vertex path with the edges on each face that comes up on the surface.

With surfaces made of triangles or quadrangles, the response time should be OK. However, take for instance a simple circle with 120 edges, and just offsetting its contour will slow down the visual feedback.

3) Limitations

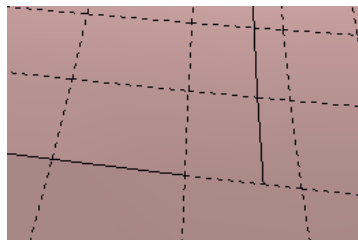
There are many.

I will complete the list in a later version of the documentation, based on the user feedback.

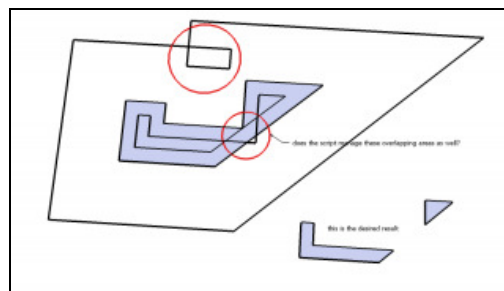
4) Remaining work

There are a number of features that could be added, but I am not quite sure this is needed.

1. **Input of distance AFTER offset operation**, as in the native Sketchup tool. In the current version, you have to Undo and start over. This is rather tricky, because when Sketchup performs an undo, the original faces are not all restored and new ones may be created. So it is not straightforward to find the selection.
2. **Improving Inference of the Red Point**, which today is only working when the surface is plane. Because the Red Point also moves onto the surface, the inference on remote point is complex to handle.
3. **Find a workaround for a Sketchup bug in overlaying edges**. In some cases, when the generated contour is exactly superimposed to an existing edge, Sketchup does not change the Soft property. This results in the fact that the generated contour is not closed. You have to find the edge rupture(s) and draw it by hand.



4. **Improve simplification of contours**, in particular to split generated contours in several portions as described by [a4chitect](#) in the Sketchup Community forum



5. **Improving the algorithm to better preserve angles**. Today I use a simplified version that does some approximations.