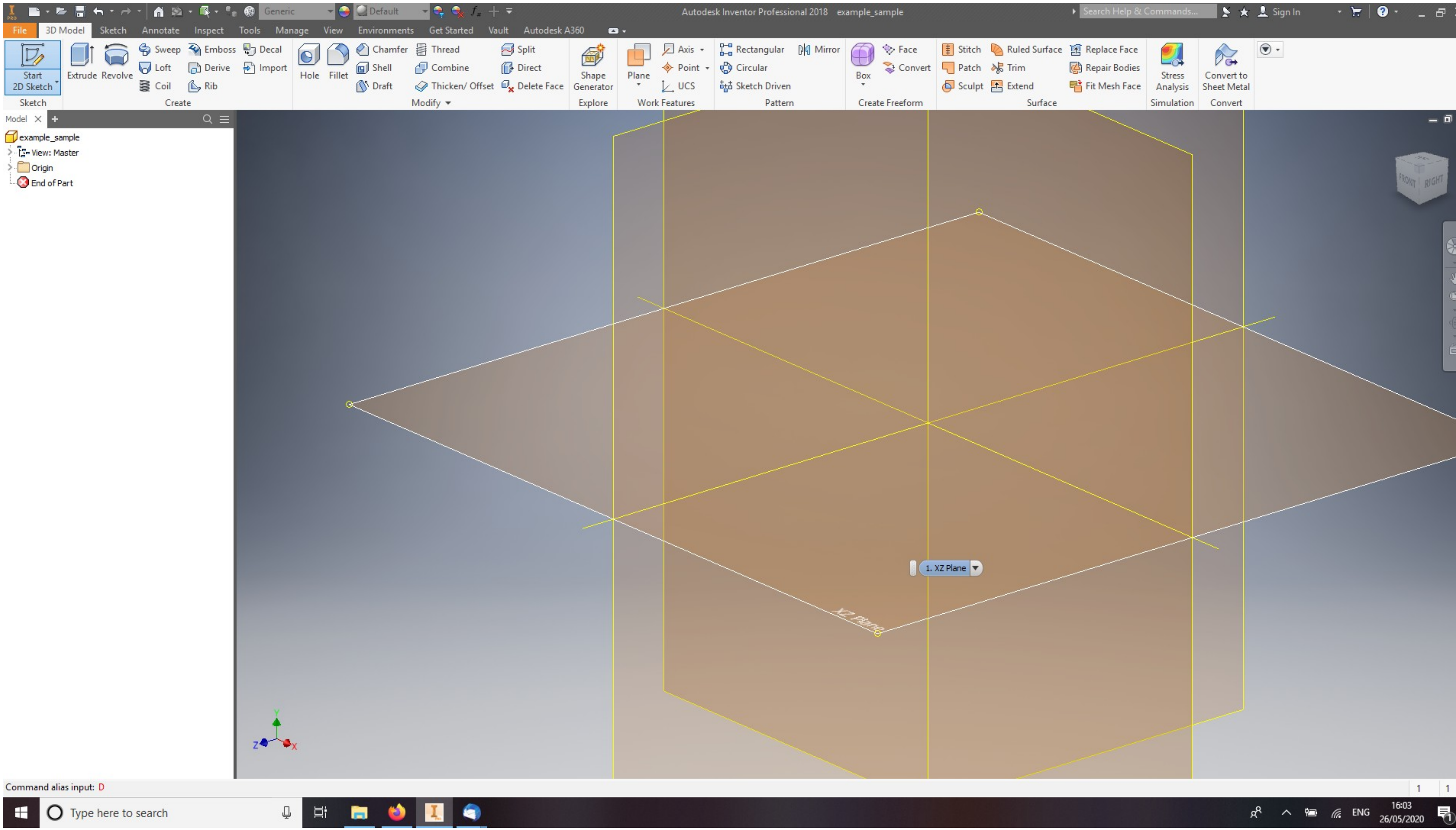
Making samples for SHeM ray tracing simulations

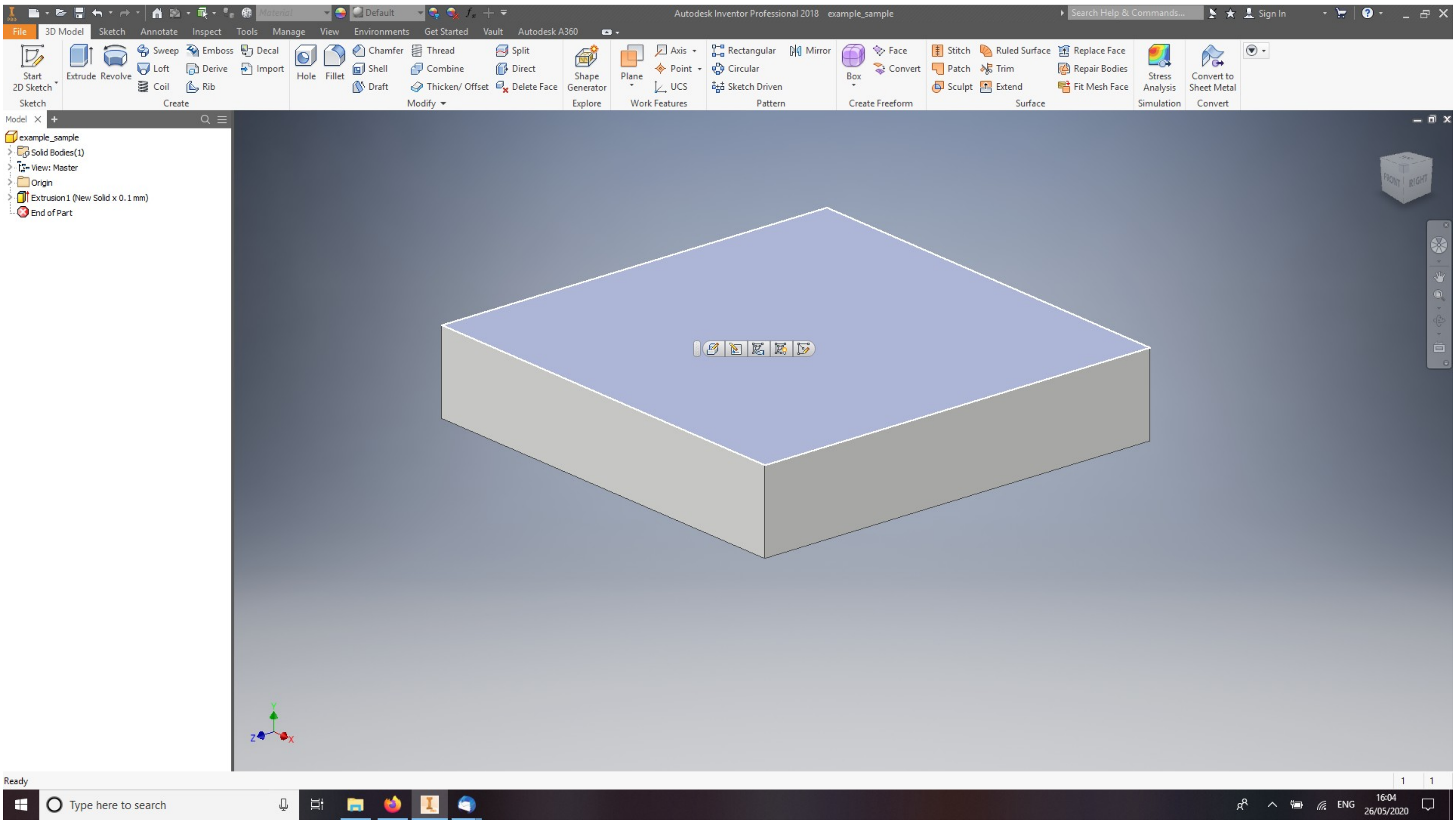
Samples need to made in a particular way for use with the ray tracing simulation. Instructions here outline how to use Autodesk Inventor to create a sample and use it in a simulation, other 3D CAD software can be used provided that an `.stl` file output can be generated. Regardless of what software package is used the following points should be the same as shown here:

1) The sample should be drawn on the appropriate plane, the xz plane with the sample extruded in the positive y direction.
2) The sample should have a base with a flat bottom.
3) The output file should be a *binary STL* file.
4) Care must be taken with the units, the ray tracing works in mm and if you want to preserve the size of your sample the output from CAD should also be in mm.

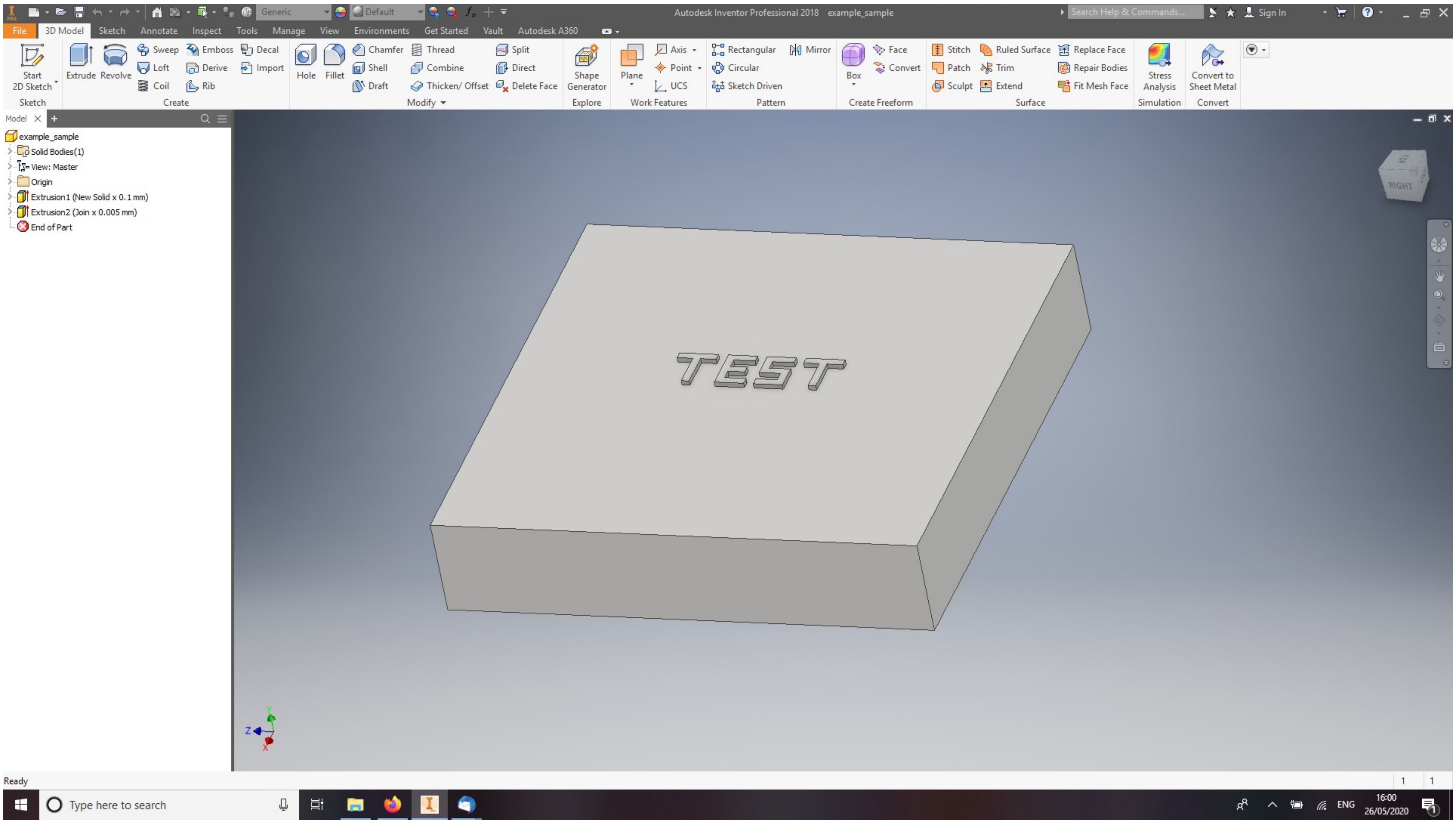
Choose the x-z plane to draw the base of the sample on, this is important, if another plane is chose the sample will not be imaged with the simulation.



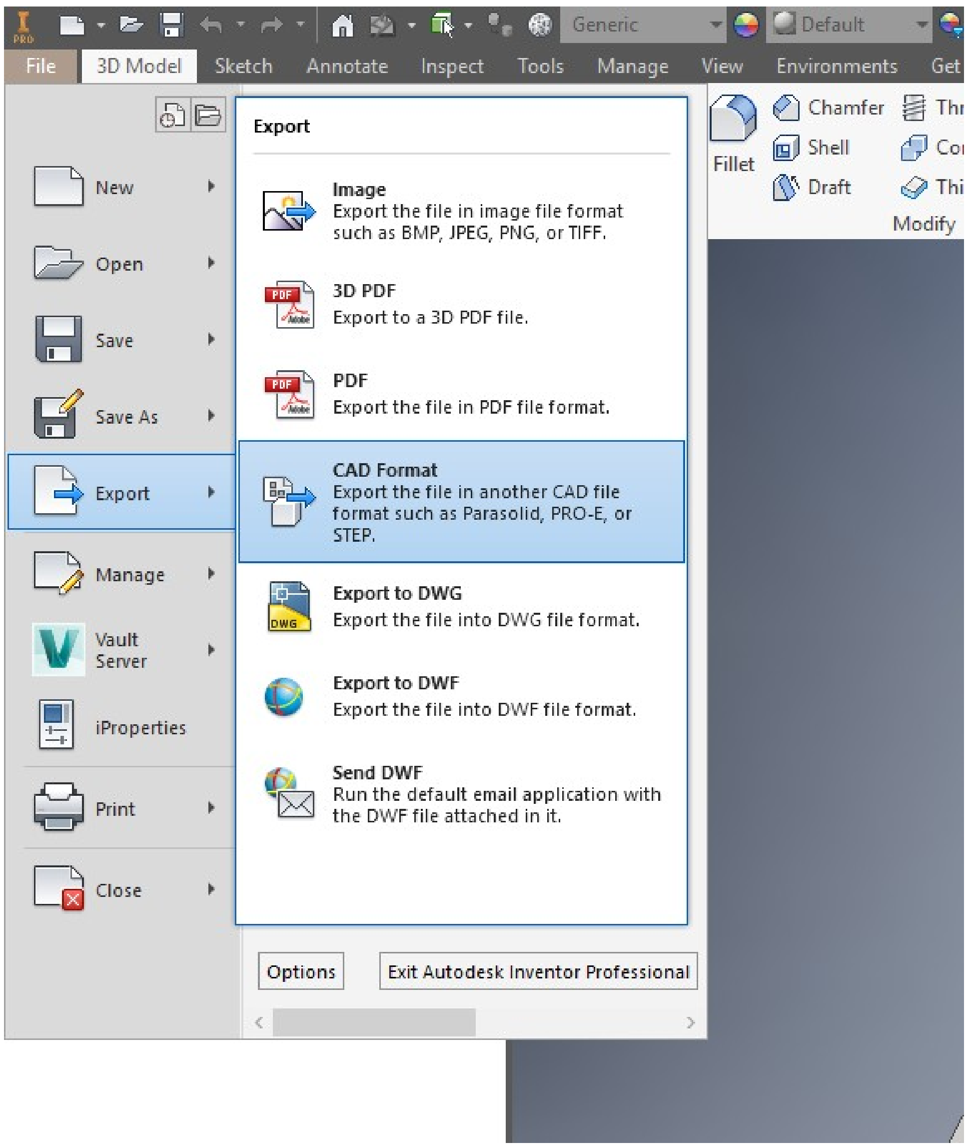
The base should be thicker the depth of the deepest feature on the sample. If the shape of the base is unimportant then creating a square base (over a circular one) will results in faster simulations. The face pointing in the *positive* y direction should be chosen to draw your sample on. It is important that the base is a simple extruded solid if the automatic sample placement it to be used.

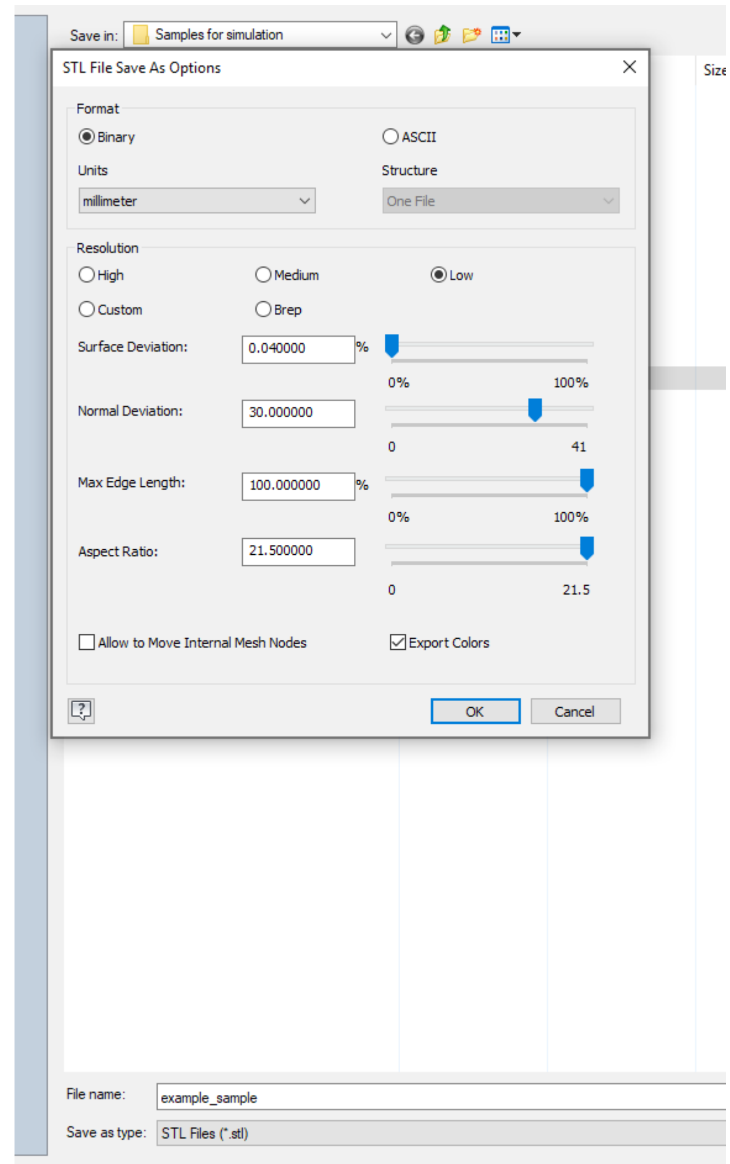


The sample may be drawn using any of the 3D tools, but it should be remembered that overly complex samples will result in slow computation and that curved surfaces will not be rendered accurately without using many triangles to represent them.



To export the sample for simulation (also save in the native CAD format for later editing) select an STL file and then selection ‘options’ and make sure a binary output with mm units is chosen. You may change the ‘resolution’ of the triangulation: this is relevant where curved surfaces are present, ‘Low’ is recommended for a first simulation.





Importing sample into the ray-tracing.

Once an STL file has been created it can be put in a directory close to the ray tracing source code. Then it must be specified in the `ray_tracing_parameters.txt` file, the path (relative or absolute) and file name should not include any spaces.

There are other parameters associated with the sample that need to be specified in the text file and there are further more in-depth modifications that can be made in the main simulation script .

In terms of sample placement the code places the ‘interesting’ part of the sample in the centre of the image. The ‘interesting part is all the vertices that are not part of the square base of the sample, this is usually a good starting point but you may wish to move your sample slightly, which can be done using a series of Matlab commands.

Within the parameter text file there will be a section entitled ‘Set up your sample’. The final parameter is the relative (or absolute) path to STL file. The second parameter sets the scattering distribution from the sample, there are currently 4 options. The first parameter must be set to ‘custom’ if you are to use your own sample. Finally the working distance for the sample must be set (in mm), this is the distance from the pinhole plate to the top most part of the sample; there is a separate ‘design working distance’ parameter which helps define the geometry of the simulated microscope, these two should probably be of a similar magnitude.

```
%% Set up your sample
% There are a few inbuilt samples or you can provide your own stl file of a
% sample, options are: 'flat', 'sphere', 'custom', 'photoStereo'
% The scattering from the sample must also be defined, options:
% 'cosine', 'uniform', 'specular', 'broad_specular'. The latter 2 require
% parameters.
%  - If you are using the 'sphere' option specify the radius of sphere.
%  - If using the 'flat' option specify the size of the flat square sample.
%  - If specifying your own sample specify the name and path to the stl file,
%    please use a file name and path without spaces
What type of sample are you useing: custom
Scattering: cosine
Reflectivity: 0.1
Standard deviation (deg): 90
Sample description: Test sample
Working distance to place the sample at: 2.121
Sphere radius (mm): 0.1
Length of side of flat sample: 1
Custom sample stl file: ../samples/example_sample.stl
```

By default the scan is centred on what the code picks up as the interesting part of the sample. If you want to shift that slightly to match your sample then edit the `xrange` and `yrange` parameters within `performScan.m`. The code also by default uses the same pixel separation in both `x` & `z` directions (either with or without scaling according to the incidence angle. The option to change this is next to the scan range parameters.

```
    raster_movment2D_x = pixel_seperation;
    raster_movment2D_z = pixel_seperation;
    xrange = [-range_x/2, range_x/2];
    zrange = [-range_z/2, range_z/2];
```

An important extra parameter is the ‘scale’ parameter which is defined in the `performScan.m` script. This parameter allows the sample to be drawn on a different scale to that desired: for example the sample could be drawn with the features mm sized rather than um sized. As CAD software is not usually designed for drawing things on the um scale this can be useful, or if the software being used does not export in mm. Note that the scale is the amount *smaller* you want the sample to be from the STL model.

```
% Sample scaling, for if the CAD model had to be made at a larger scale. 10 will
% make the model 10 times smaller (Inventor exports in cm by default...).
scale = 1;
```

Manipulating the sample in ray-tracing

Some basic manipulations can be added to the ‘custom’ case of the sample importing section of `performScan.m`. The two most likely you are to use are `moveBy()` and `rotateY()`, these are called as methods to the `TriagSurface` class thus:

```
sample_surface.moveBy([x, y, z]);
sample_surface.rotateY;
```

Which both do what they say (recall that the dimension from the sample to the pinhole plate is the y axis of the simulation), the rotation is 90deg and there are options for rotating about the other axes too.

```
    case 'custom'
        sample_surface = inputSample('fname', sample_fname, 'scattering', ...
            diffuse(1), 'plate_dist', dist_to_sample, 'scale', scale, ...
            'parameters', diffuse(2), 'working_dist', working_dist);
        make_sphere = 0;
        sample_surface.rotateY;
        sample_surface.rotateY;
        sample_surface.rotateY;
```

A further argument can be provided to the `inputSample` function: ‘dontMeddle’, which if set to true will not attempt to automatically place the sample. If this option is chosen then all sample manipulation will be have to be done manually.

A final useful command will plot a triangulated surface within Matlab (an optional true/false argument sets if a new figure is to be used):

```
sample_surface.patchPlot();
```

Basic outputs