Architectural Geometry — Worksheet 3

This worksheet is intended to help us understand deformations and offsets.

You are allowed (and indeed encouraged) to discuss the tasks with other participants, but you are not allowed to hand in solutions produced by somebody else. If you get stuck, please do not hesitate to contact me, and I will do my best to help you get past the obstacle.

Write up your solutions neatly in a document, and send it together with an archive (e.g. zipped) containing your code to me at `marten.wadenback@chalmers.se` at the latest on the 26th of July.

Task 1: Use the skeleton in `twistObject.m` to implement the twist deformation. You can use the provided function `rotm(v,phi)` to create a matrix which rotates vectors around $\boldsymbol{v}$.

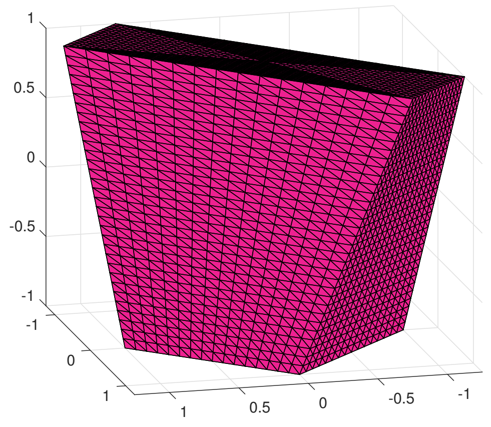
Hint: Plotting (a segment of) the axis of rotation may help you in verifying that your twist works as intended.

Please include the figures generated by the following commands:

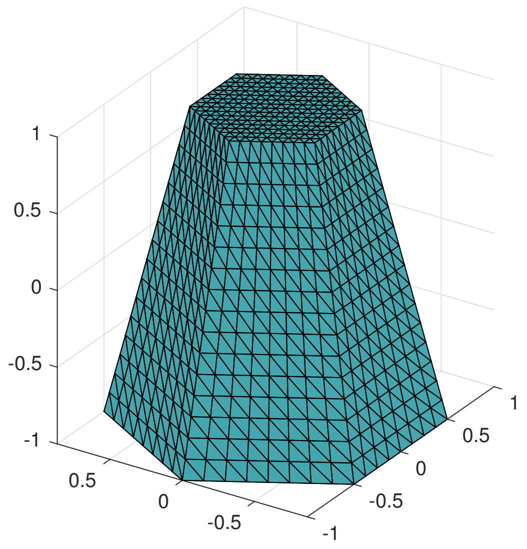
```
>> S = cylinder(1,3);
>> S = subdivide(S);
>> renderObject(twistObject(S,rotm([0 1 0]',-pi/4),zeros(3,1),pi/4));
>> figure;
>> renderObject(twistObject(S,rotm([1 0 0]',pi/2),zeros(3,1),pi/3));
>> figure;
>> renderObject(twistObject(S,eye(3),zeros(3,1),2*pi/3));
>> figure;
>> renderObject(twistObject(S,eye(3),[1 0 0]',pi));
```

Task 2: Use the skeleton in `taperObject.m` to implement the tapering deformation.

Verify that your tapering works, and use it to produce the two shapes from the slides:



(a)



(b)

Task 3: Use the skeleton in `shearObject.m` to implement the shear deformation.

Please include the figures generated by the following commands:

```
>> S = cylinder(1,6);
>> S = subdivide(S);
>> S = subdivide(S);
>> S = subdivide(S);
>> c = @(t) [2*t.*(t-1); zeros(size(t)); zeros(size(t))];
>> renderObject(shearObject(S,eye(3),zeros(3,1),c));
```

Task 4: Derive the offset surface for each of the following surfaces:

(a) $z = x + y$,

(b) $z = xy$, and

(c) $z = 0.3 \cdot (x^2 - y^2)$.

For each, produce a plot showing the original surface in one colour and the two parts of the offset surface, at distance $d = 0.1$, in another colour.

To plot the surface in e.g. (b), use

```
>> [X,Y] = meshgrid(-2:0.2:2,-2:0.2:2);
>> Z = 0.3*(X.^2-Y.^2);
>> surf(X,Y,Z,'FaceColor',[0.5 0.25 1]);
>> axis equal;
```