An introduction to loadawobj

Table of Contents

Some examples of usage of loadawobs.m and companion files loadawmtl.m drawaw.m

This file can be `published' with publish('demoloadawobj','pdf')

Simple files

loadawobj, at its simplest, will load a file and draws it on the current figure, e.g. loadawobj('icosahedron.obj').

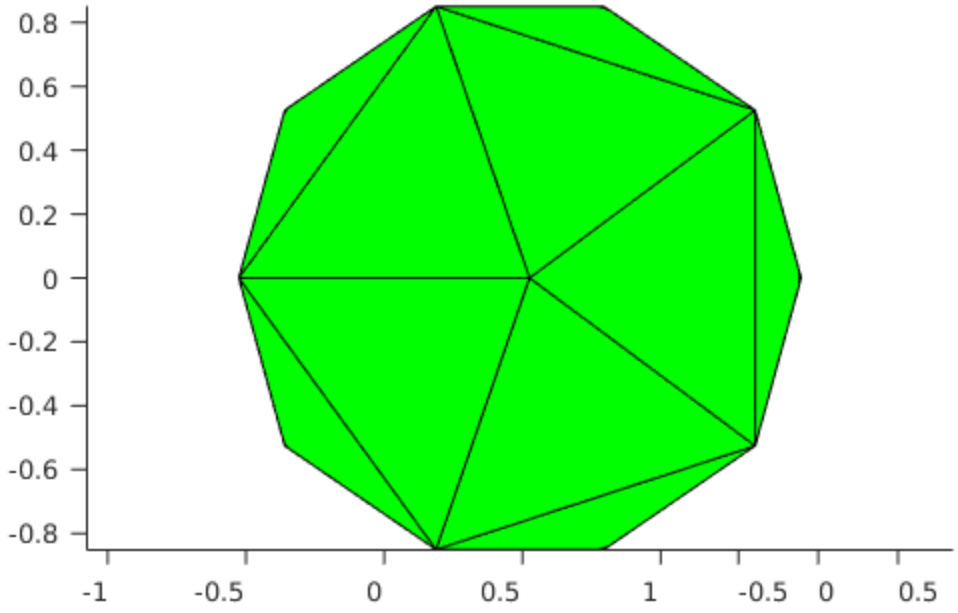
But more control is possible by extracting the vertices and faces information from the file

note

1. Since the icosahedron only has triangles, only F3 needs to be in the output list,

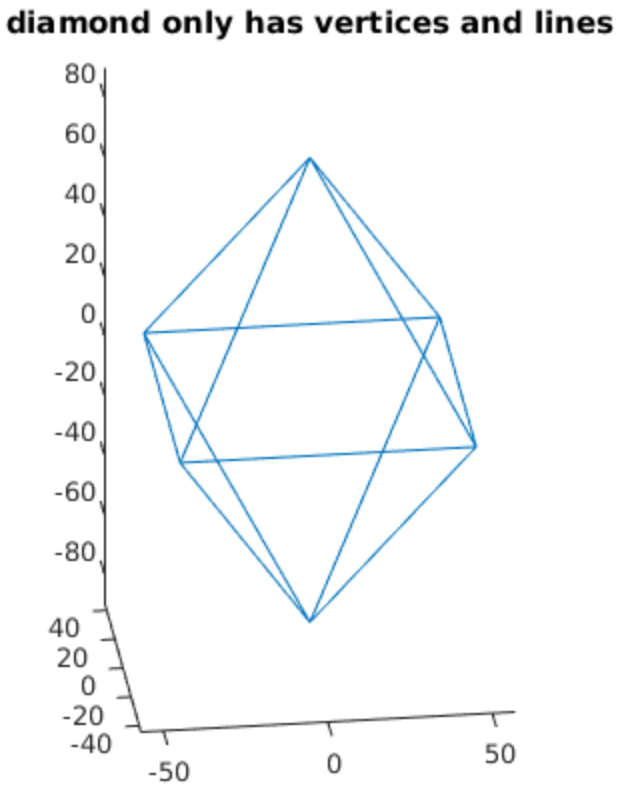
2. the transpose of the vertices and faces matrix is given to the patch command.

```
fig=figure; % open a new figure
modelname='icosahedron.obj';
[v,F3]=loadawobj('icosahedron.obj');
patch('Vertices',v','Faces',F3','FaceColor','g');
view(30,0);
axis('equal');
snapnow
pause(1)
```



line drawings

```
clf
loadawobj('obj/diamond.obj');
view(-7,26);
title('diamond only has vertices and lines');
snapnow
pause(1)
```



S=loadawobj('file.obj')

If there is a single output variable, loadawobj will return a structure with more details extracted from the obj file.

The structure fields are

```
version
v : vertices
f3,f4,f5,f6 : A list of faces with 3,4,5, and 6 vertices.
g : Group names
g3 g4 : The index indicates the group by faces
l : Lines (see diamond.obj)
umat3 umat4 : The index indicates to which group a face belongs
mtllib : The material library (load with loadawmtl)
usemtl : Material names
vt : Vertex textures
vn : Vertex normals
tc3 : texture coorndate for 3 face polygons
tc4 : texture coorndate for 4 face polygons
vn3 : texture coorndate for 3 face polygons
vn4 : texture coorndate for 4 face polygons
```

Note

g3 and g4 are the same length as f3 and f4 and indicate to which group a face set belongs.

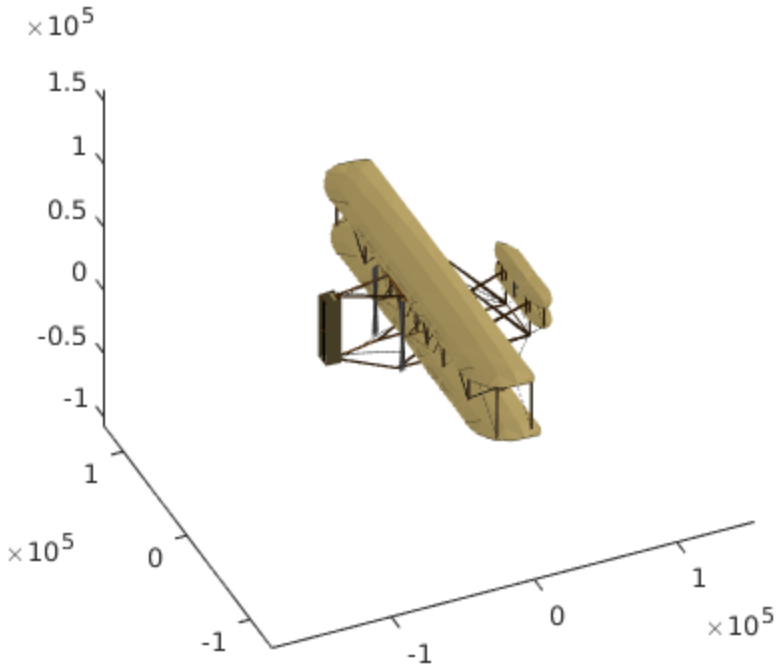
Only 3-6 faces are currently supported, it should be relatively easy to add more

plane.obj

Load and display a biplane Check the contents of drawaw for more details on how to handle Matlab®'s FaceVertexColorData

```
clf
modelname='obj/plane.obj';
S=loadawobj(modelname);
if isfield(S,'mtllib')
    mtl=loadawmtl(['obj/' S.mtllib]);
end
drawaw(S,mtl)
view(-24,36);
light
snapnow;
pause(1)
%

mtllib plane.mtl
# Max2Mtl Version 4.0 Mar 10th, 2001
#
# Multi/Sub Material__35 (5) to come
#
d 1.0
illum 2
#
d 1.0
illum 2
#
d 1.0
illum 2
#
d 1.0
illum 2
#
d 1.0
illum 2
#
# Multi/Sub Material__35 done
#
# EOF
```



functions for coordinate transforms

these can be used in place, i.e. S.v=tfverts(Rz180*Rx90,[10;0;0],S.v)

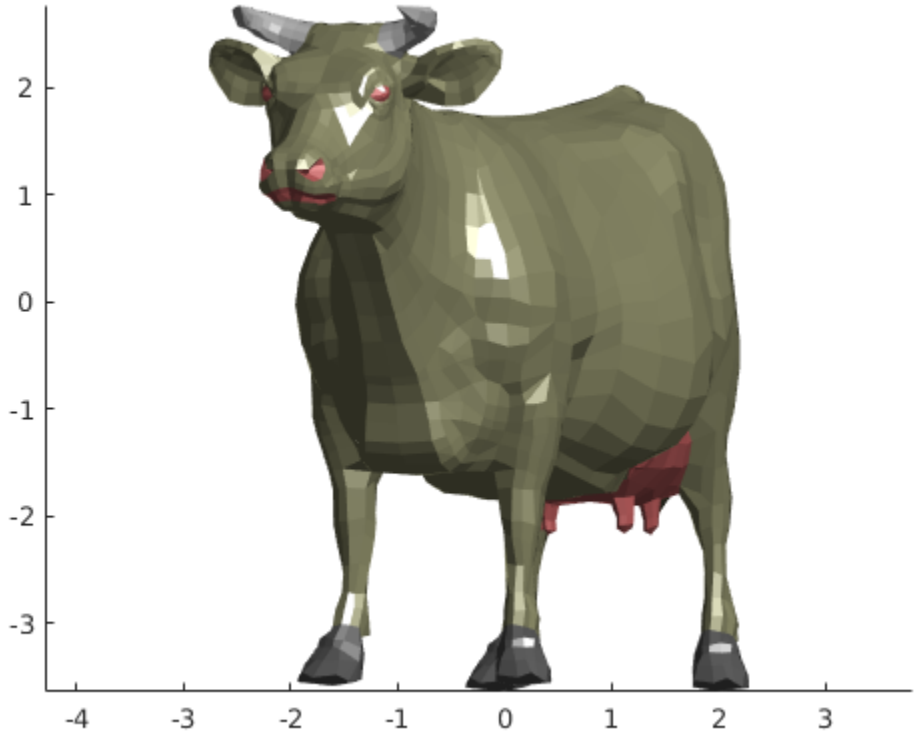
```
Rz=@(th)[cos(th) -sin(th) 0;sin(th) cos(th) 0; 0 0 1];
Rx=@(th)[1 0 0;0 cos(th) -sin(th);0 sin(th) cos(th)];
Ry=@(th)[cos(th) 0 sin(th);0 1 0;-sin(th) 0 cos(th)];
tfverts=@(R,p,v) R*v+p*ones(1,length(v));
```

so in-order to avoid looking at the back end of a cow we can rotate it around the y axis by 160 degrees

```
modelname='obj/cow.obj';
S=loadawobj(modelname);
S.v=tfverts(Ry(160*pi/180),[0 0 0]',S.v);
if isfield(S,'mtllib')
    mtl=loadawmtl(['obj/' S.mtllib]);
end
clf;drawaw(S,mtl)
camlight
snapnow
pause(1)
close(fig)
```

```
mtllib cow.mtl
unprocessed-#---
unprocessed-#-- materials for cow.obj model-
```

```
unprocessed-#---
unprocessed-#-- same as hide-
unprocessed-#-- should be "udder" not "utter"-
```



Published with MATLAB® R2015b