THE UNIVERSITY OF RHODE ISLAND

# EGR 106
# Foundations of Engineering II

## Lecture 11 – Part B
## Design Project  - Week 2

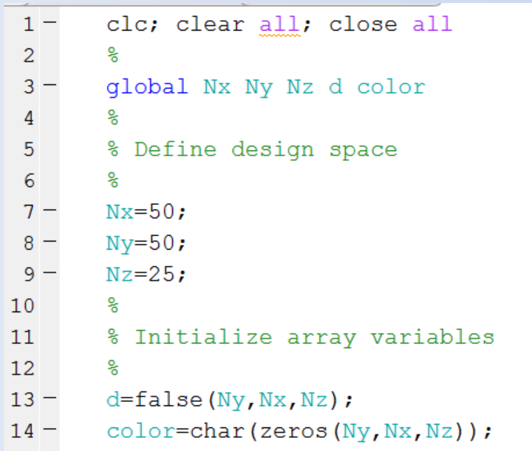
THINK BIG WE DO℠

# Design Project (Week 2)

- Main program structure
  - Initialize arrays
  - Calls to functions to define shapes
  - Calls to functions for creating 3D models
- Functions for creating basic 3D shapes
  - Box, sphere, cylinder_x, cylinder_y, cylinder_z, segment
- Creating 3D models
  - model_gen, model_animate
- Examples
  - truss, lamp, eight_balls
- Viewing 3D models in MeshLab

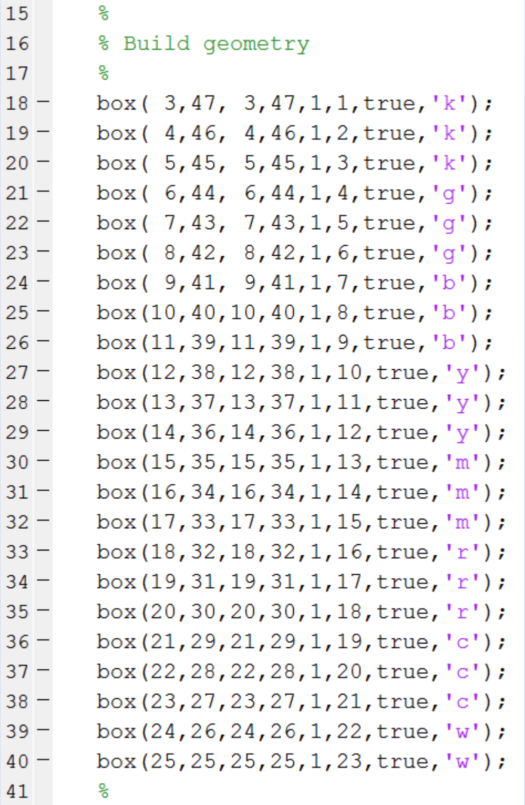
# Main program structure
# (ex. last week: eight_color_demo.m)

## Initialize arrays



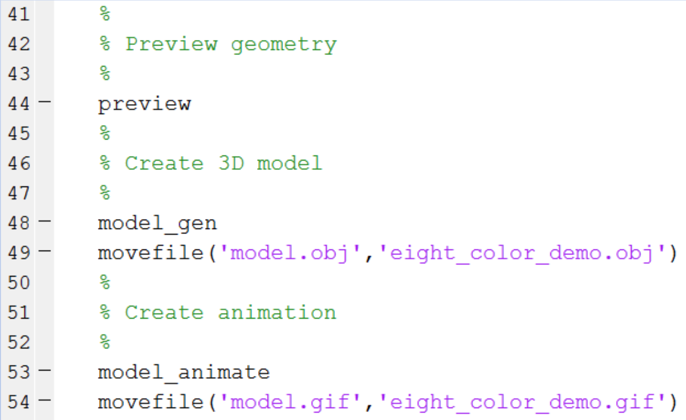
```
clc; clear all; close all
%
global Nx Ny Nz d color
%
% Define design space
%
Nx=50;
Ny=50;
Nz=25;
%
% Initialize array variables
%
d=false(Ny,Nx,Nz);
color=char(zeros(Ny,Nx,Nz));
```

## Define 3D shapes



```
%
% Build geometry
%
box( 3,47, 3,47,1,1,true,'k');
box( 4,46, 4,46,1,2,true,'k');
box( 5,45, 5,45,1,3,true,'k');
box( 6,44, 6,44,1,4,true,'g');
box( 7,43, 7,43,1,5,true,'g');
box( 8,42, 8,42,1,6,true,'g');
box( 9,41, 9,41,1,7,true,'b');
box(10,40,10,40,1,8,true,'b');
box(11,39,11,39,1,9,true,'b');
box(12,38,12,38,1,10,true,'y');
box(13,37,13,37,1,11,true,'y');
box(14,36,14,36,1,12,true,'y');
box(15,35,15,35,1,13,true,'m');
box(16,34,16,34,1,14,true,'m');
box(17,33,17,33,1,15,true,'m');
box(18,32,18,32,1,16,true,'r');
box(19,31,19,31,1,17,true,'r');
box(20,30,20,30,1,18,true,'r');
box(21,29,21,29,1,19,true,'c');
box(22,28,22,28,1,20,true,'c');
box(23,27,23,27,1,21,true,'c');
box(24,26,24,26,1,22,true,'w');
box(25,25,25,25,1,23,true,'w');
%
```

## Create and view 3D models



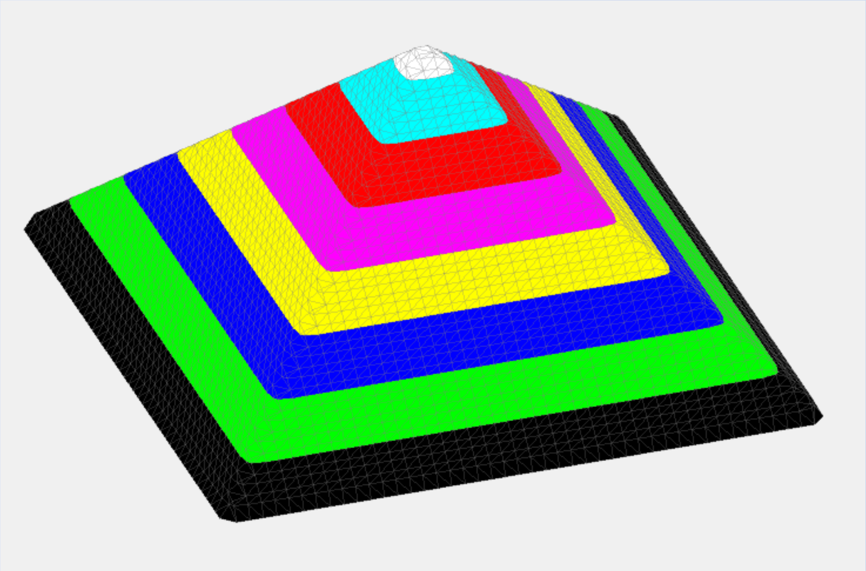
```
%
% Preview geometry
%
preview
%
% Create 3D model
%
model_gen
movefile('model.obj','eight_color_demo.obj')
%
% Create animation
%
model_animate
movefile('model.gif','eight_color_demo.gif')
```

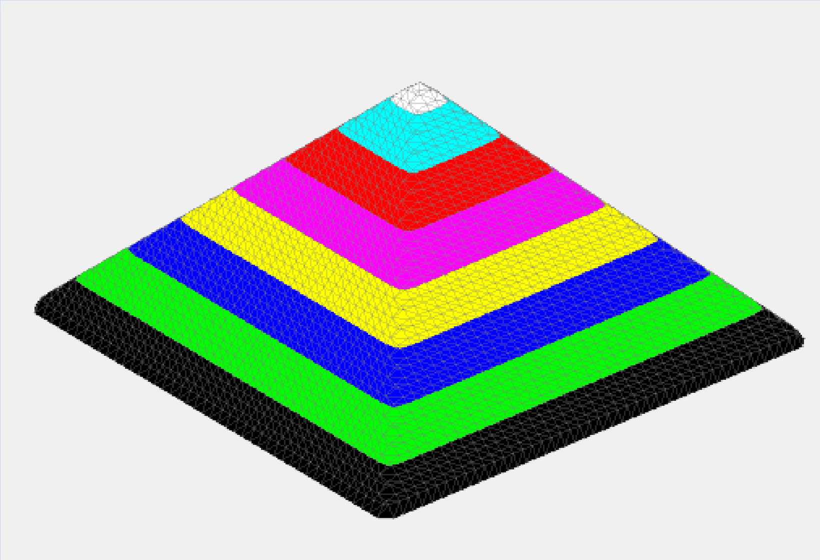
# eight_color_demo.m – Result with model_gen

Matlab Figure Window

Animated gif

(cropped at ezgif.com/crop)





# Functions for Creating 3D Shapes (provided on Brightspace as p-files*)

box

sphere

cylinder_x

cylinder_y

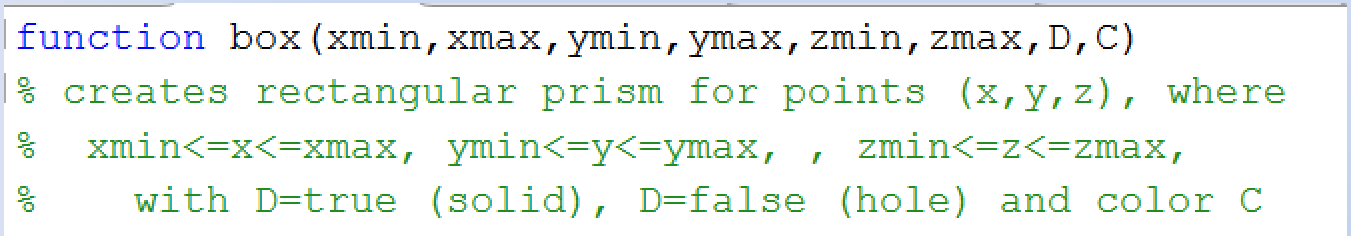
cylinder_z

segment

*p files – protected mode Matlab function scripts (can be run but source code not viewable, cannot be edited)

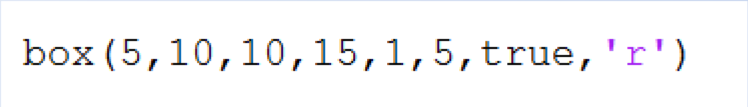
# box.p (introduced last week)

## Function to create a 3D box

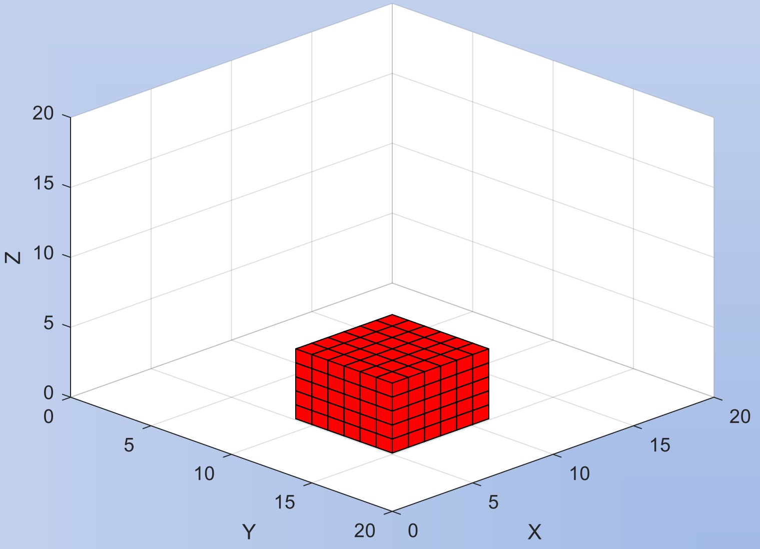


```
function box(xmin,xmax,ymin,ymax,zmin,zmax,D,C)
% creates rectangular prism for points (x,y,z), where
%  xmin<=x<=xmax, ymin<=y<=ymax, , zmin<=z<=zmax,
%    with D=true (solid), D=false (hole) and color C
```

## Example:

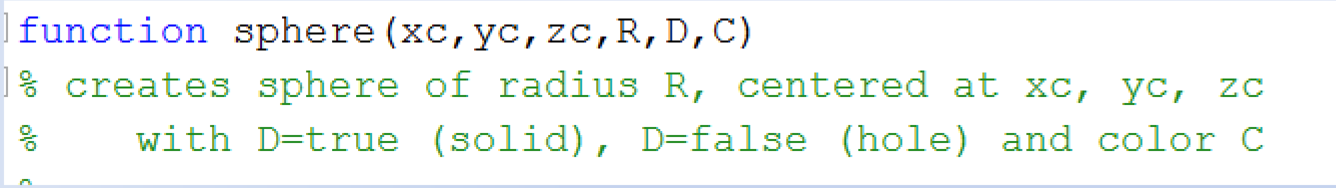


```
box(5,10,10,15,1,5,true,'r')
```



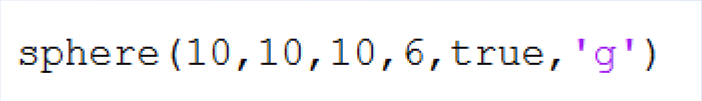
# sphere.p

## Function to create a 3D sphere

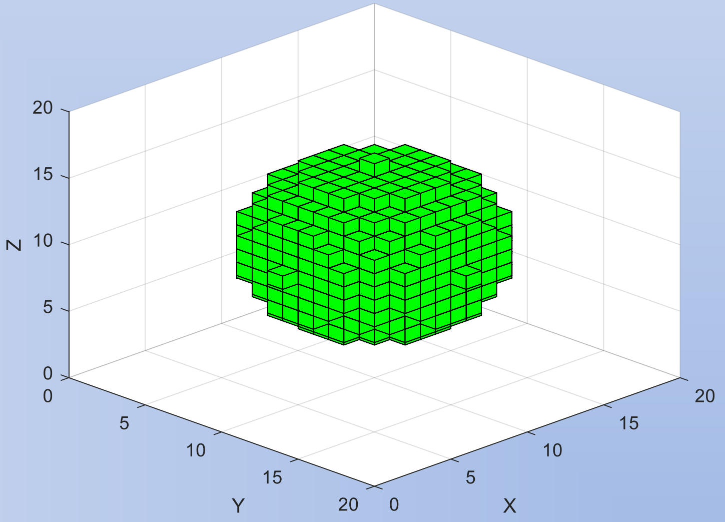


```
function sphere(xc,yc,zc,R,D,C)
% creates sphere of radius R, centered at xc, yc, zc
%    with D=true (solid), D=false (hole) and color C
```

## Example:

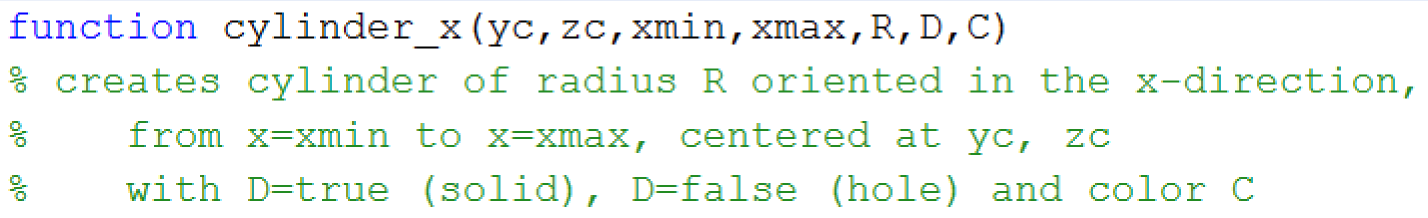


```
sphere(10,10,10,6,true,'g')
```



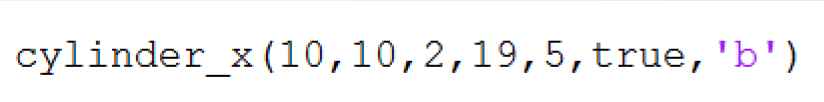
# cylinder_x.p

Function to create a circular cylinder parallel to x-axis

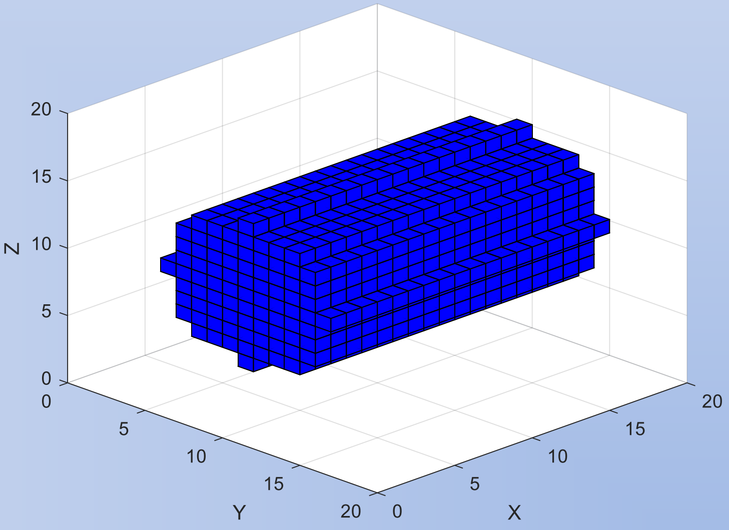


```
function cylinder_x(yc,zc,xmin,xmax,R,D,C)
% creates cylinder of radius R oriented in the x-direction,
%    from x=xmin to x=xmax, centered at yc, zc
%    with D=true (solid), D=false (hole) and color C
```

Example:

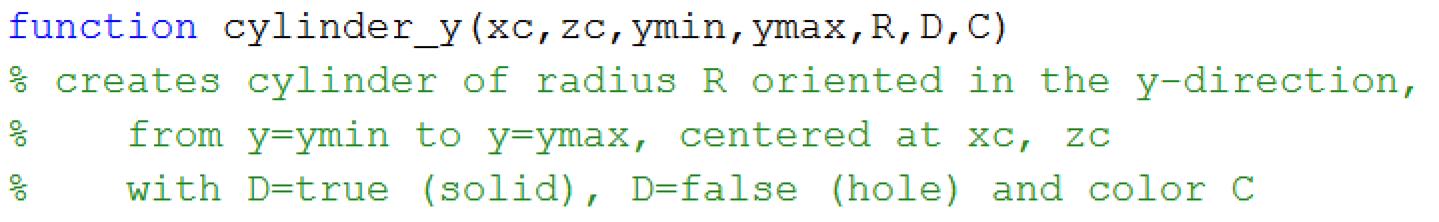


```
cylinder_x(10,10,2,19,5,true,'b')
```



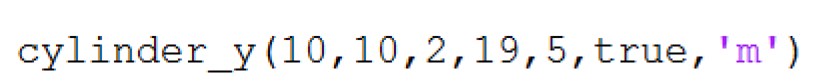
# cylinder_y.p

Function to create a circular cylinder parallel to y-axis

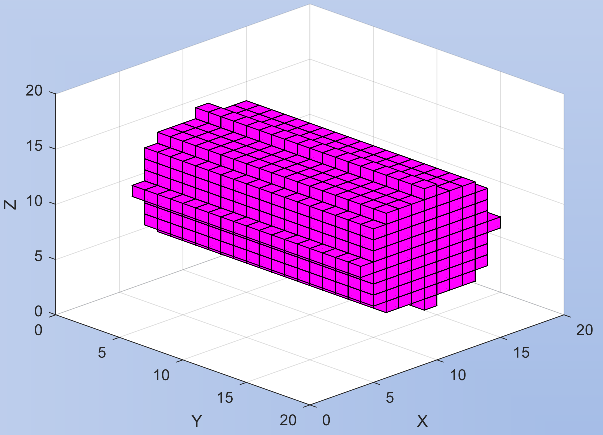


```
function cylinder_y(xc,zc,ymin,ymax,R,D,C)
% creates cylinder of radius R oriented in the y-direction,
%    from y=ymin to y=ymax, centered at xc, zc
%    with D=true (solid), D=false (hole) and color C
```

Example:

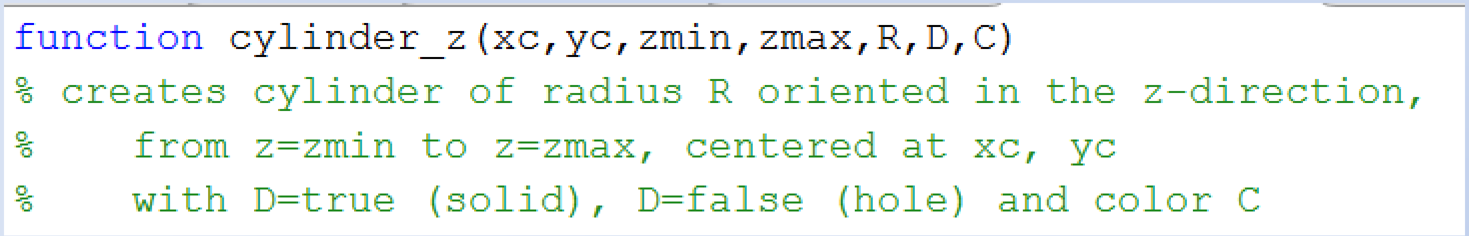


```
cylinder_y(10,10,2,19,5,true,'m')
```



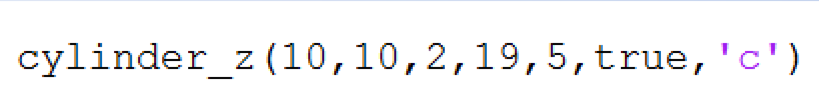
# cylinder_z.p

Function to create a circular cylinder parallel to z-axis

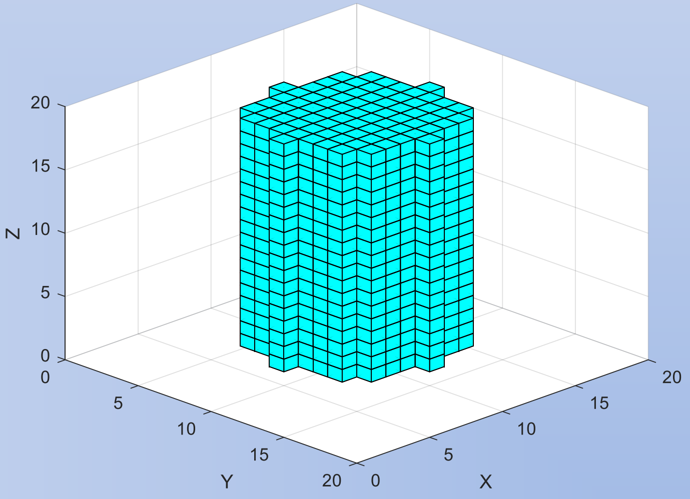


```
function cylinder_z(xc,yc,zmin,zmax,R,D,C)
% creates cylinder of radius R oriented in the z-direction,
%     from z=zmin to z=zmax, centered at xc, yc
%     with D=true (solid), D=false (hole) and color C
```

Example:

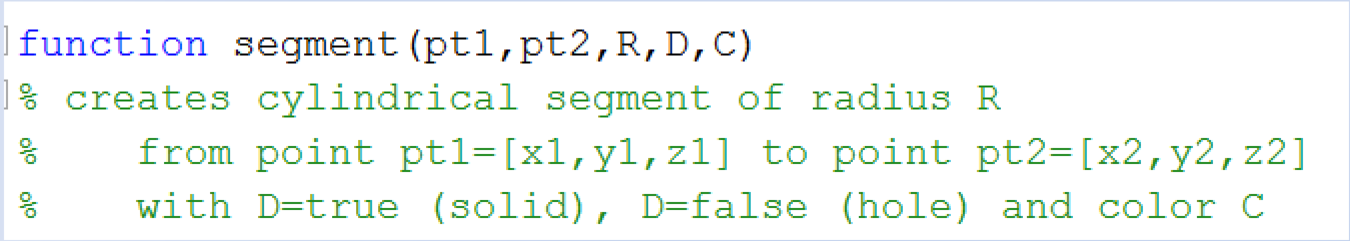


```
cylinder_z(10,10,2,19,5,true,'c')
```



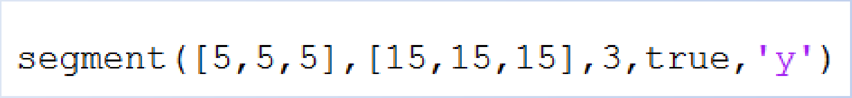
# segment.p

## Function to create a line segment

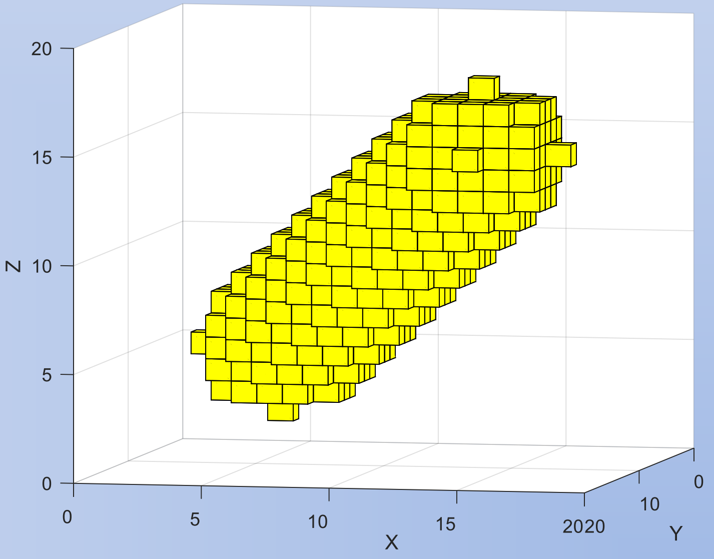


```
function segment(pt1,pt2,R,D,C)
% creates cylindrical segment of radius R
%     from point pt1=[x1,y1,z1] to point pt2=[x2,y2,z2]
%     with D=true (solid), D=false (hole) and color C
```

## Example:



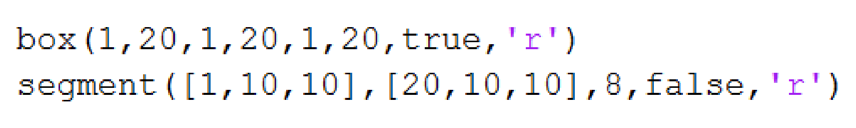
```
segment([5,5,5],[15,15,15],3,true,'y')
```



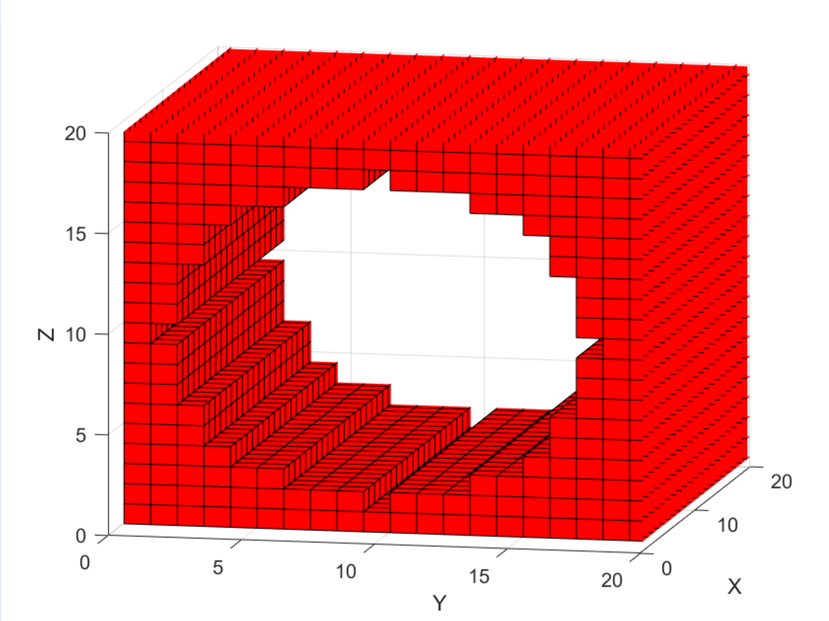
# Creating holes

Use ’false’ to create holes

Example:



```
box(1,20,1,20,1,20,true,'r')
segment([1,10,10],[20,10,10],8,false,'r')
```



# Creating and Animating 3D models

model_gen.p

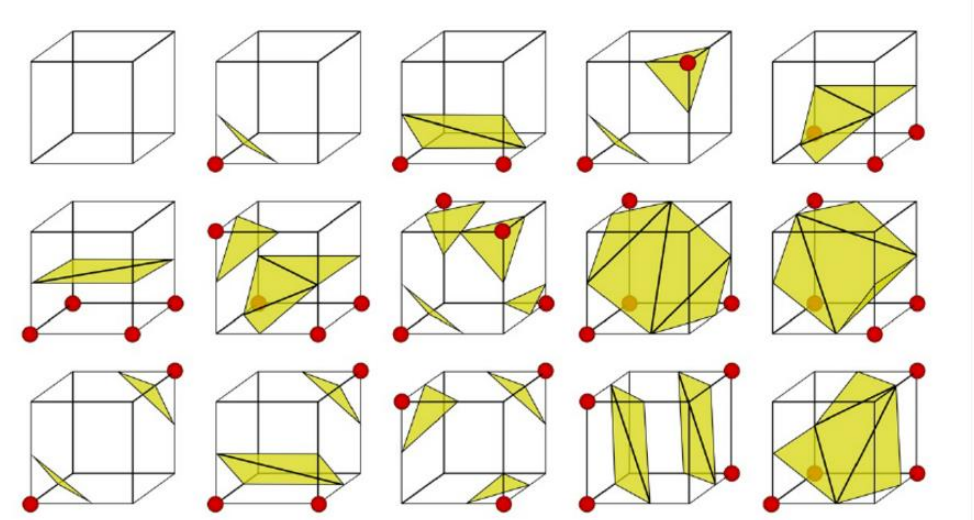
- Smooths the surface
- Generates CAD (.obj) files

model_animate.p

- Provides rotating image
- Saves  animated gif file
  - (can be cropped at ezgif.com/crop)

# model_gen.p –Algorithm

Marching cube algorithm – defines triangulated surface between solid and void regions



To use, issue command: model_gen
(be sure model_gen.p is in your current directory)

# model_animate.p code

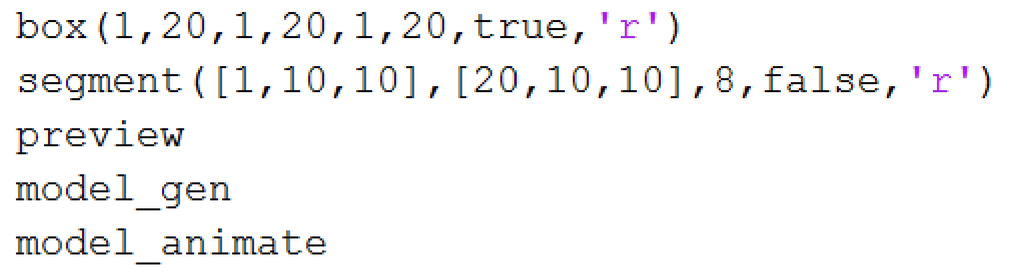
In a ‘for’ loop, uses Matlab functions:

- view
- getframe
- frame2im
- imwrite

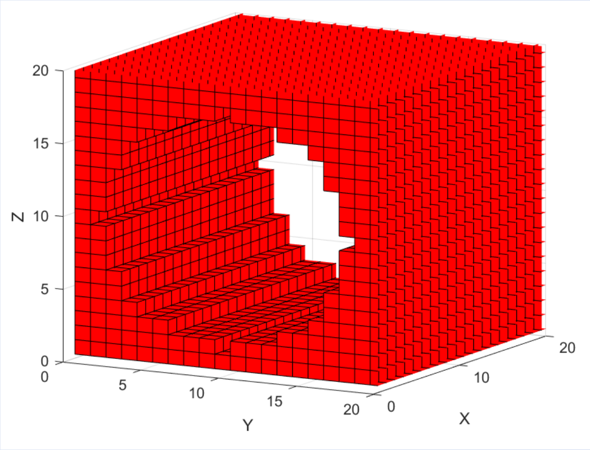
To use, issue command: model_animate

(be sure model_animate.p is in your current directory)

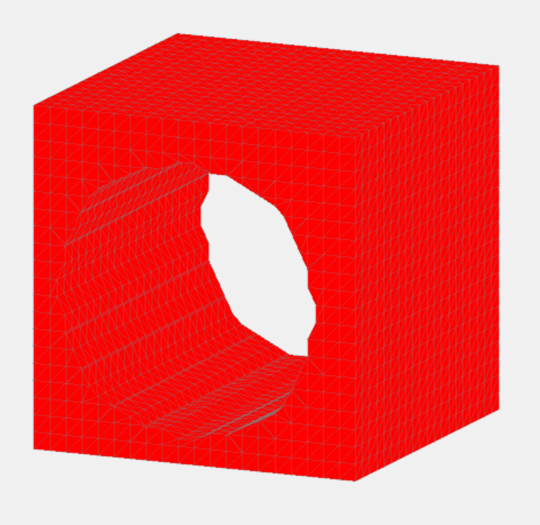
# Example (box with circular hole)



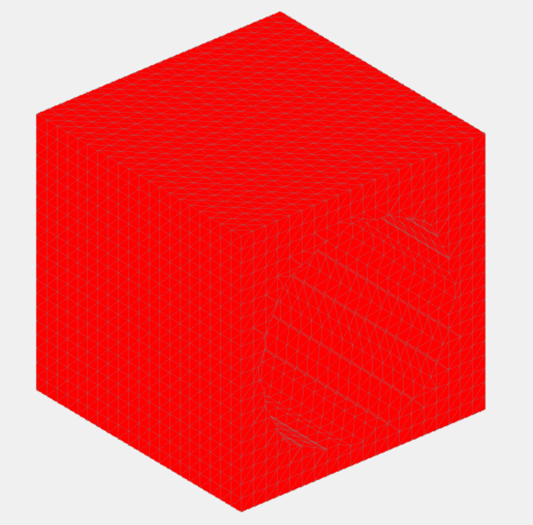
```
box(1,20,1,20,1,20,true,'r')
segment([1,10,10],[20,10,10],8,false,'r')
preview
model_gen
model_animate
```



preview



model_gen



model_animate

# Examples

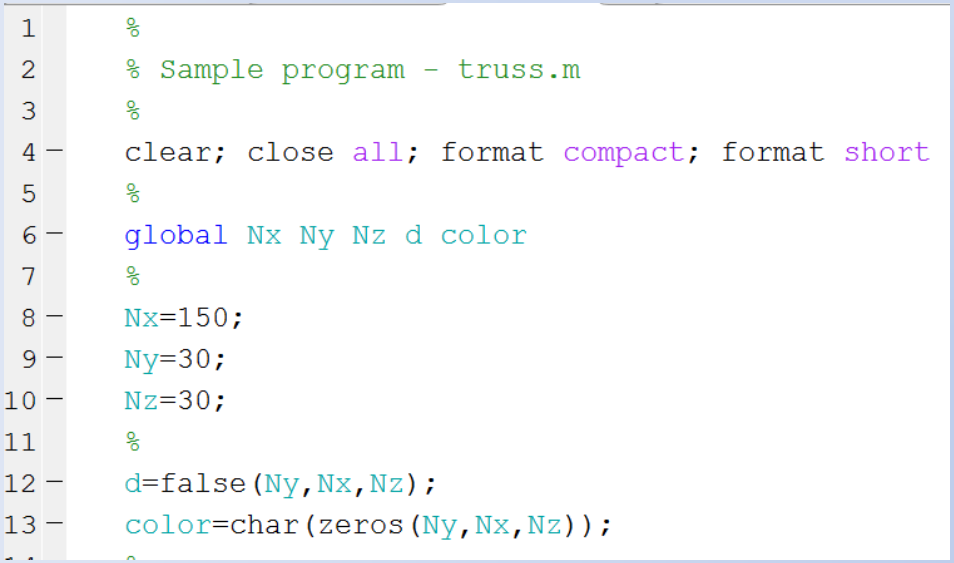
Truss (segment & box)

Lamp (cylinder_z)

Eight balls (sphere)

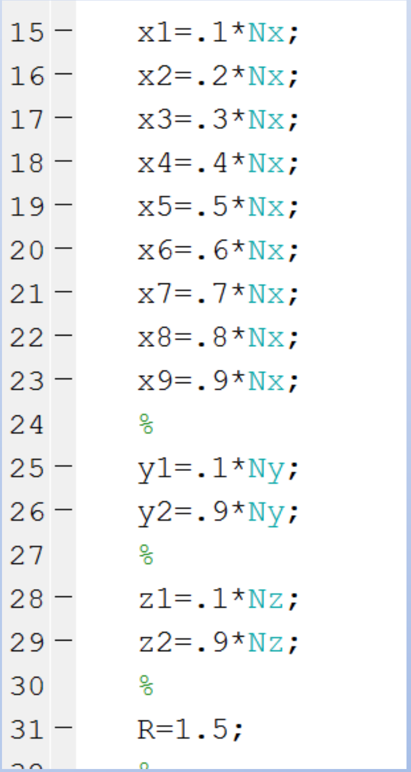
# Example 1 – truss.m

## Initialization



```
%
% Sample program - truss.m
%
clear; close all; format compact; format short
%
global Nx Ny Nz d color
%
Nx=150;
Ny=30;
Nz=30;
%
d=false(Ny,Nx,Nz);
color=char(zeros(Ny,Nx,Nz));
```

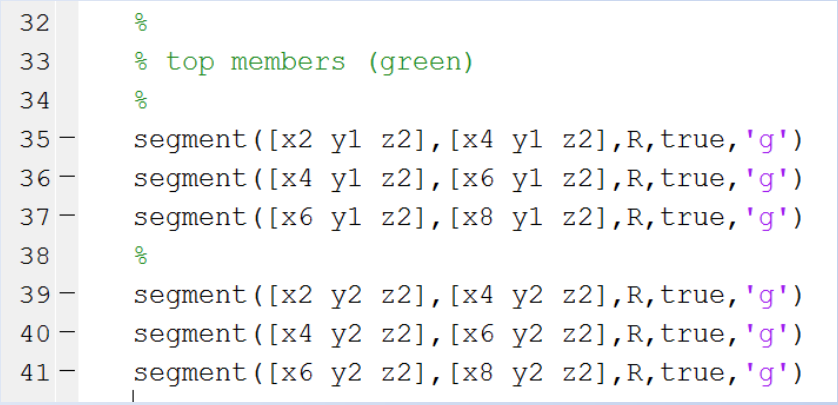
## Define key points



```
x1=.1*Nx;
x2=.2*Nx;
x3=.3*Nx;
x4=.4*Nx;
x5=.5*Nx;
x6=.6*Nx;
x7=.7*Nx;
x8=.8*Nx;
x9=.9*Nx;
%
y1=.1*Ny;
y2=.9*Ny;
%
z1=.1*Nz;
z2=.9*Nz;
%
R=1.5;
```

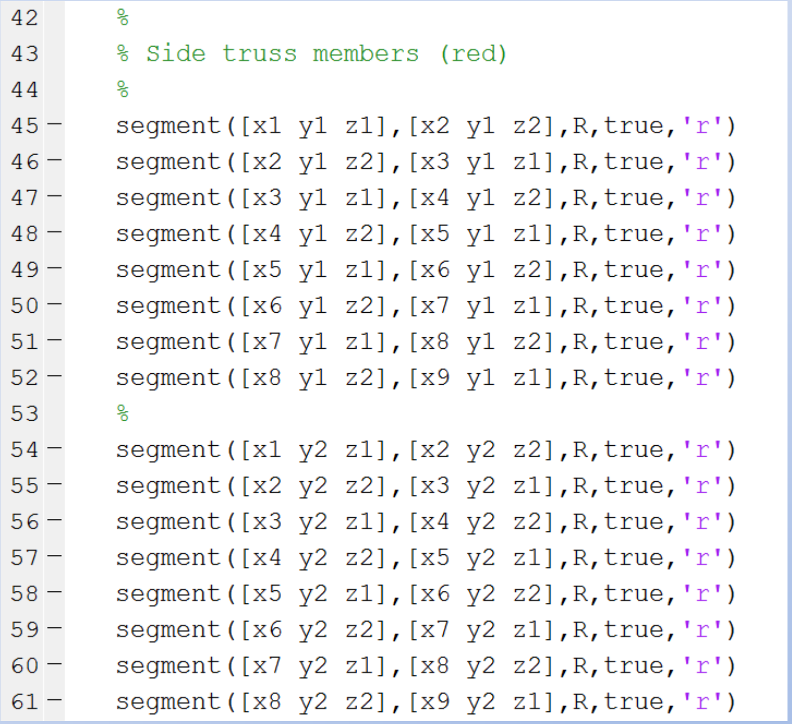
# Example 1 – truss.m (cont.)

Define top members



```
%
% top members (green)
%
segment([x2 y1 z2],[x4 y1 z2],R,true,'g')
segment([x4 y1 z2],[x6 y1 z2],R,true,'g')
segment([x6 y1 z2],[x8 y1 z2],R,true,'g')
%
segment([x2 y2 z2],[x4 y2 z2],R,true,'g')
segment([x4 y2 z2],[x6 y2 z2],R,true,'g')
segment([x6 y2 z2],[x8 y2 z2],R,true,'g')
```

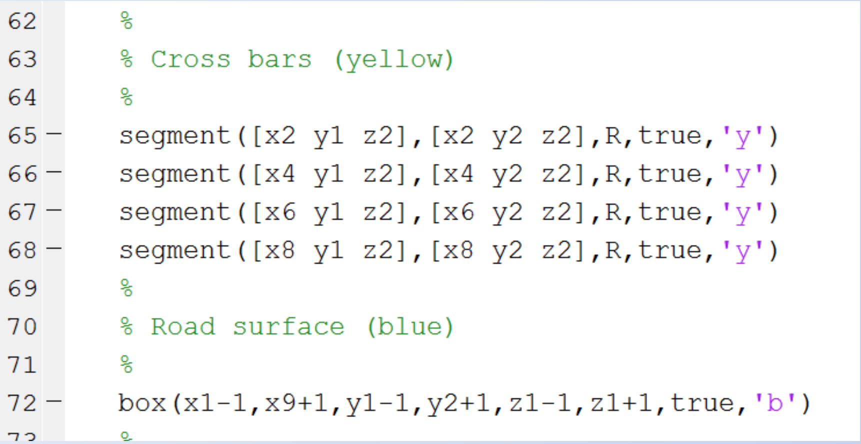
Define side truss members



```
%
% Side truss members (red)
%
segment([x1 y1 z1],[x2 y1 z2],R,true,'r')
segment([x2 y1 z2],[x3 y1 z1],R,true,'r')
segment([x3 y1 z1],[x4 y1 z2],R,true,'r')
segment([x4 y1 z2],[x5 y1 z1],R,true,'r')
segment([x5 y1 z1],[x6 y1 z2],R,true,'r')
segment([x6 y1 z2],[x7 y1 z1],R,true,'r')
segment([x7 y1 z1],[x8 y1 z2],R,true,'r')
segment([x8 y1 z2],[x9 y1 z1],R,true,'r')
%
segment([x1 y2 z1],[x2 y2 z2],R,true,'r')
segment([x2 y2 z2],[x3 y2 z1],R,true,'r')
segment([x3 y2 z1],[x4 y2 z2],R,true,'r')
segment([x4 y2 z2],[x5 y2 z1],R,true,'r')
segment([x5 y2 z1],[x6 y2 z2],R,true,'r')
segment([x6 y2 z2],[x7 y2 z1],R,true,'r')
segment([x7 y2 z1],[x8 y2 z2],R,true,'r')
segment([x8 y2 z2],[x9 y2 z1],R,true,'r')
```

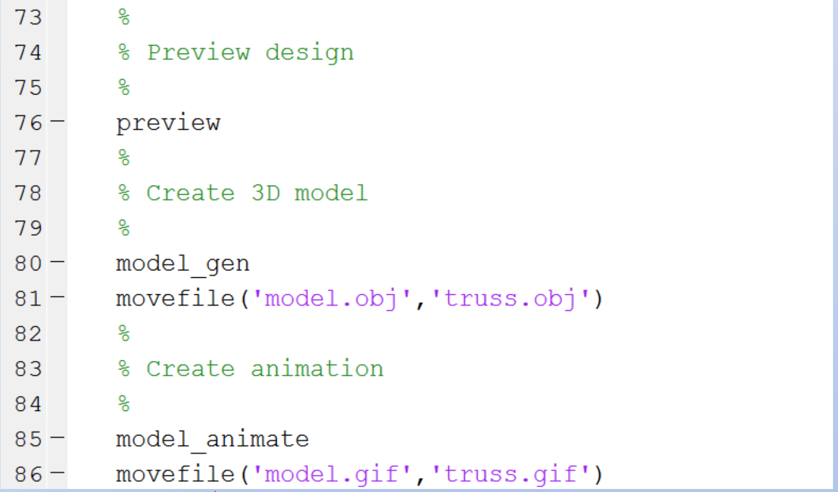
# Example 1 – truss.m (cont.)

Define cross bars and road surface



```
%
% Cross bars (yellow)
%
segment([x2 y1 z2],[x2 y2 z2],R,true,'y')
segment([x4 y1 z2],[x4 y2 z2],R,true,'y')
segment([x6 y1 z2],[x6 y2 z2],R,true,'y')
segment([x8 y1 z2],[x8 y2 z2],R,true,'y')
%
% Road surface (blue)
%
box(x1-1,x9+1,y1-1,y2+1,z1-1,z1+1,true,'b')
%
```

Preview design, create 3D model and animation

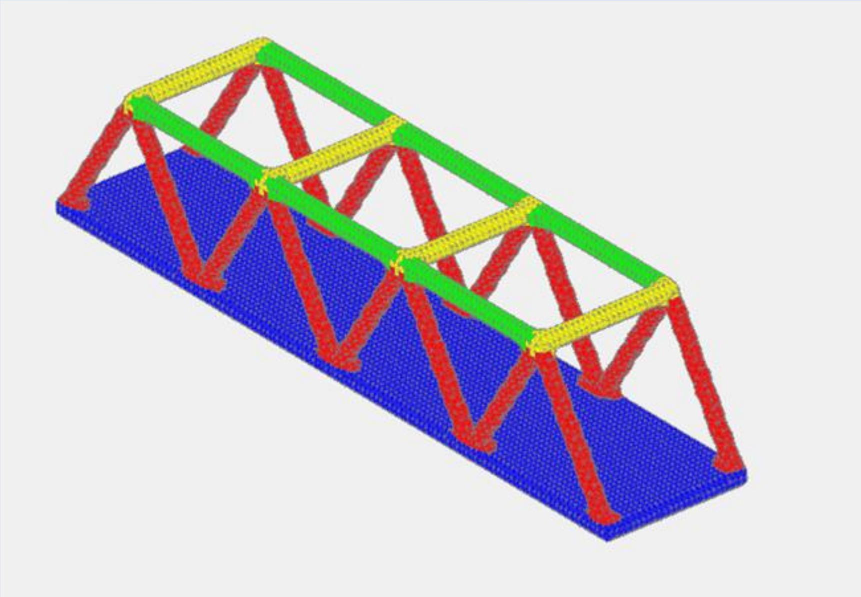


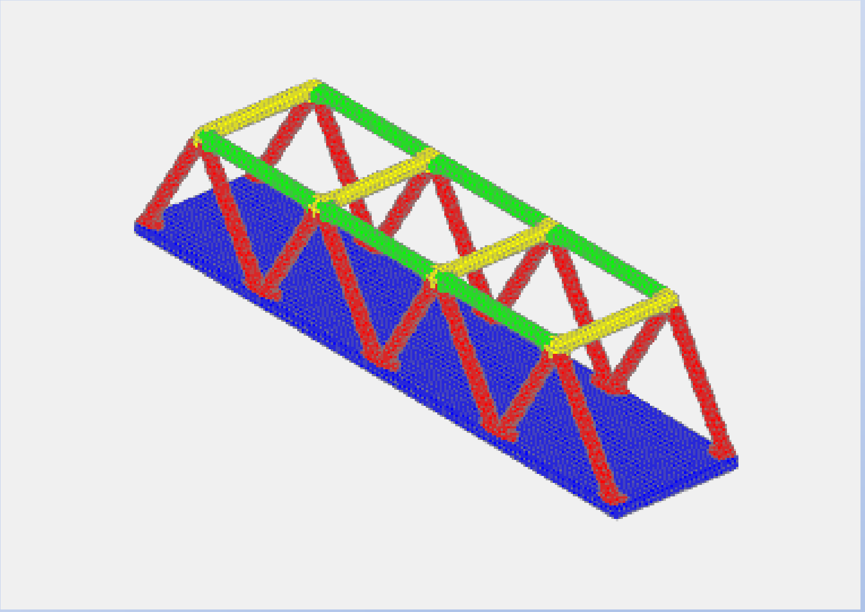
```
%
% Preview design
%
preview
%
% Create 3D model
%
model_gen
movefile('model.obj','truss.obj')
%
% Create animation
%
model_animate
movefile('model.gif','truss.gif')
```

Use movefile command to rename .obj and .gif files

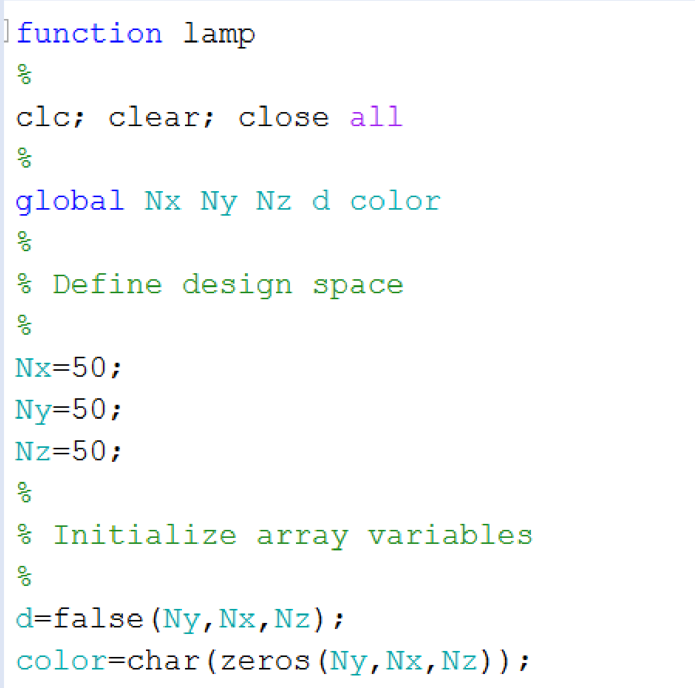
# Example 1 – truss.m (cont.)

Result

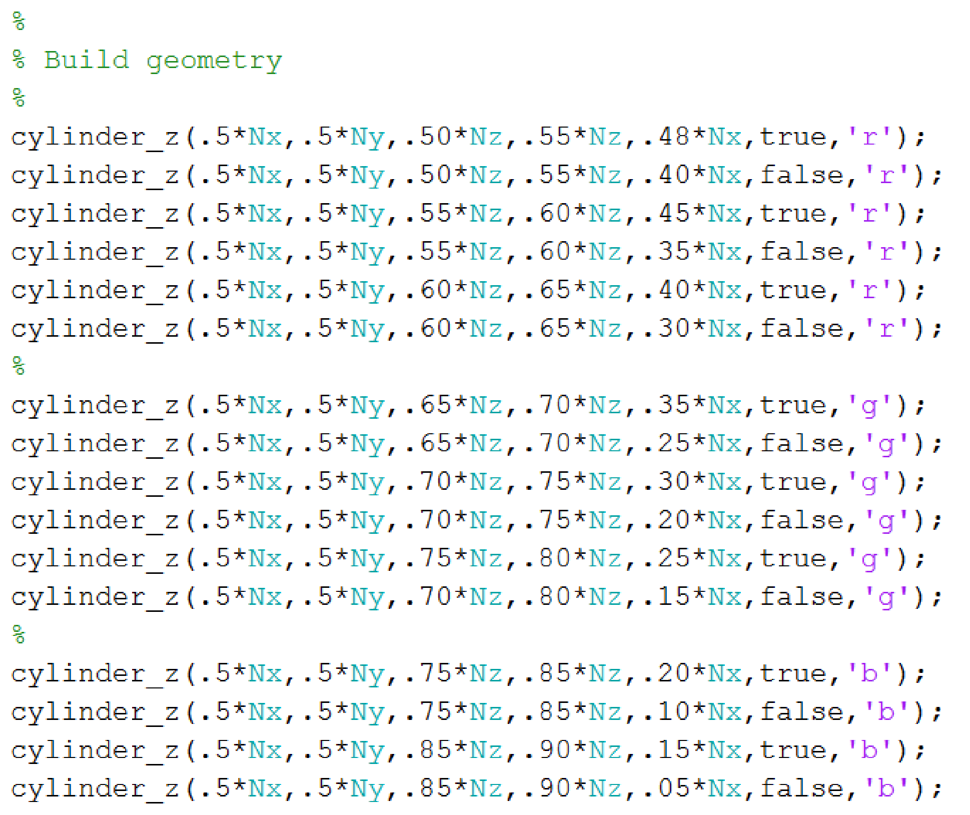




# Example 2 – lamp.m

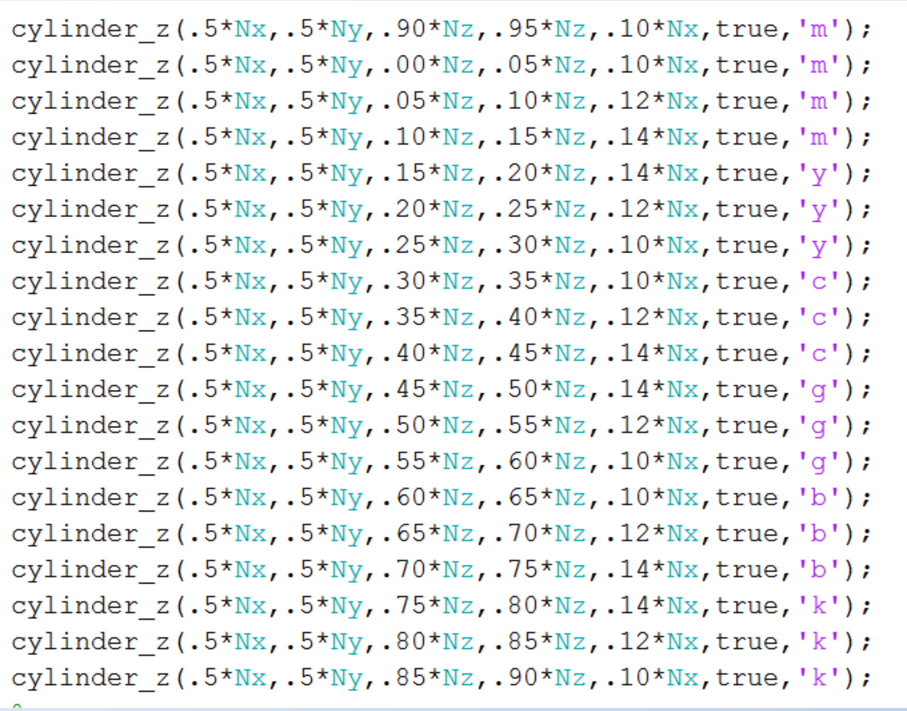


```
function lamp
%
clc; clear; close all
%
global Nx Ny Nz d color
%
% Define design space
%
Nx=50;
Ny=50;
Nz=50;
%
% Initialize array variables
%
d=false(Ny,Nx,Nz);
color=char(zeros(Ny,Nx,Nz));
```

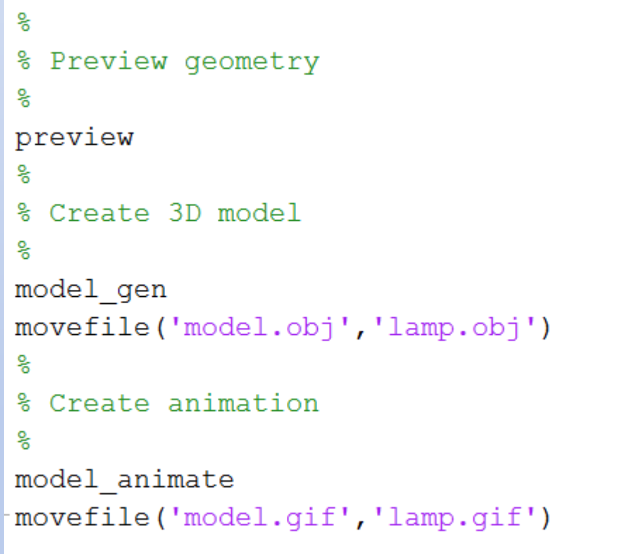


```
%
% Build geometry
%
cylinder_z(.5*Nx,.5*Ny,.50*Nz,.55*Nz,.48*Nx,true,'r');
cylinder_z(.5*Nx,.5*Ny,.50*Nz,.55*Nz,.40*Nx,false,'r');
cylinder_z(.5*Nx,.5*Ny,.55*Nz,.60*Nz,.45*Nx,true,'r');
cylinder_z(.5*Nx,.5*Ny,.55*Nz,.60*Nz,.35*Nx,false,'r');
cylinder_z(.5*Nx,.5*Ny,.60*Nz,.65*Nz,.40*Nx,true,'r');
cylinder_z(.5*Nx,.5*Ny,.60*Nz,.65*Nz,.30*Nx,false,'r');
%
cylinder_z(.5*Nx,.5*Ny,.65*Nz,.70*Nz,.35*Nx,true,'g');
cylinder_z(.5*Nx,.5*Ny,.65*Nz,.70*Nz,.25*Nx,false,'g');
cylinder_z(.5*Nx,.5*Ny,.70*Nz,.75*Nz,.30*Nx,true,'g');
cylinder_z(.5*Nx,.5*Ny,.70*Nz,.75*Nz,.20*Nx,false,'g');
cylinder_z(.5*Nx,.5*Ny,.75*Nz,.80*Nz,.25*Nx,true,'g');
cylinder_z(.5*Nx,.5*Ny,.70*Nz,.80*Nz,.15*Nx,false,'g');
%
cylinder_z(.5*Nx,.5*Ny,.75*Nz,.85*Nz,.20*Nx,true,'b');
cylinder_z(.5*Nx,.5*Ny,.75*Nz,.85*Nz,.10*Nx,false,'b');
cylinder_z(.5*Nx,.5*Ny,.85*Nz,.90*Nz,.15*Nx,true,'b');
cylinder_z(.5*Nx,.5*Ny,.85*Nz,.90*Nz,.05*Nx,false,'b');
```

# Example 2 – lamp.m (cont.)



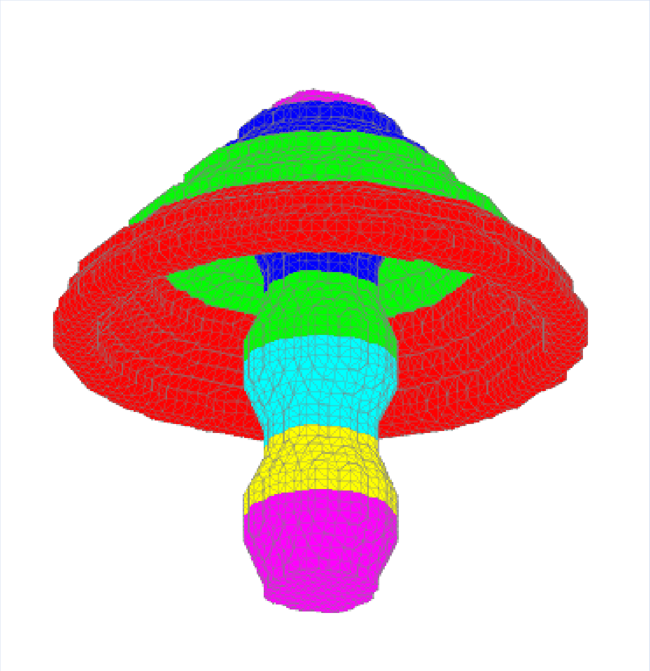
```
cylinder_z(.5*Nx,.5*Ny,.90*Nz,.95*Nz,.10*Nx,true,'m');
cylinder_z(.5*Nx,.5*Ny,.00*Nz,.05*Nz,.10*Nx,true,'m');
cylinder_z(.5*Nx,.5*Ny,.05*Nz,.10*Nz,.12*Nx,true,'m');
cylinder_z(.5*Nx,.5*Ny,.10*Nz,.15*Nz,.14*Nx,true,'m');
cylinder_z(.5*Nx,.5*Ny,.15*Nz,.20*Nz,.14*Nx,true,'y');
cylinder_z(.5*Nx,.5*Ny,.20*Nz,.25*Nz,.12*Nx,true,'y');
cylinder_z(.5*Nx,.5*Ny,.25*Nz,.30*Nz,.10*Nx,true,'y');
cylinder_z(.5*Nx,.5*Ny,.30*Nz,.35*Nz,.10*Nx,true,'c');
cylinder_z(.5*Nx,.5*Ny,.35*Nz,.40*Nz,.12*Nx,true,'c');
cylinder_z(.5*Nx,.5*Ny,.40*Nz,.45*Nz,.14*Nx,true,'c');
cylinder_z(.5*Nx,.5*Ny,.45*Nz,.50*Nz,.14*Nx,true,'g');
cylinder_z(.5*Nx,.5*Ny,.50*Nz,.55*Nz,.12*Nx,true,'g');
cylinder_z(.5*Nx,.5*Ny,.55*Nz,.60*Nz,.10*Nx,true,'g');
cylinder_z(.5*Nx,.5*Ny,.60*Nz,.65*Nz,.10*Nx,true,'b');
cylinder_z(.5*Nx,.5*Ny,.65*Nz,.70*Nz,.12*Nx,true,'b');
cylinder_z(.5*Nx,.5*Ny,.70*Nz,.75*Nz,.14*Nx,true,'b');
cylinder_z(.5*Nx,.5*Ny,.75*Nz,.80*Nz,.14*Nx,true,'k');
cylinder_z(.5*Nx,.5*Ny,.80*Nz,.85*Nz,.12*Nx,true,'k');
cylinder_z(.5*Nx,.5*Ny,.85*Nz,.90*Nz,.10*Nx,true,'k');
```

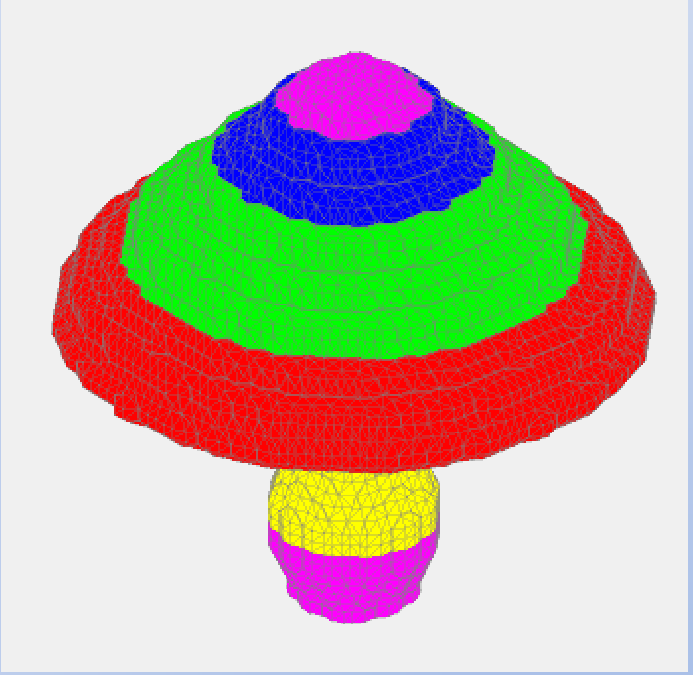


```
%
% Preview geometry
%
preview
%
% Create 3D model
%
model_gen
movefile('model.obj','lamp.obj')
%
% Create animation
%
model_animate
movefile('model.gif','lamp.gif')
```

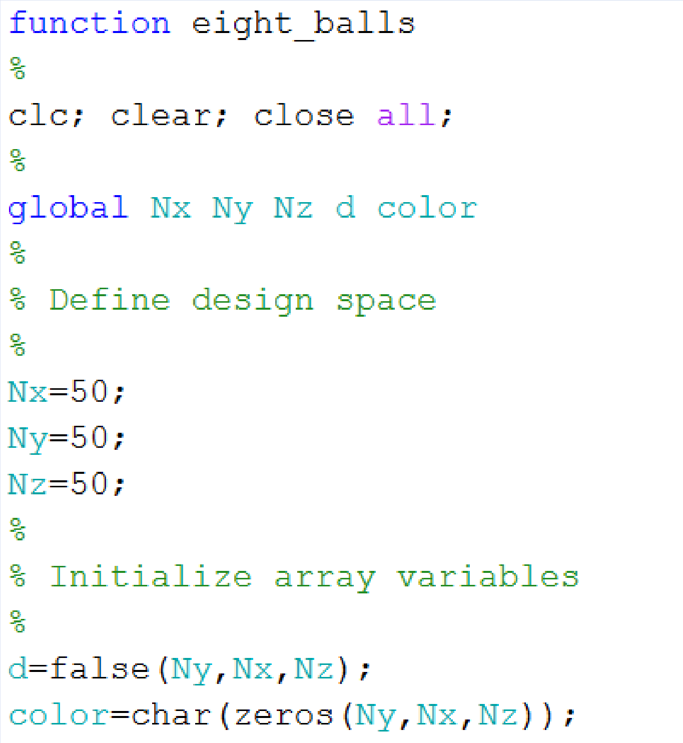
# Example 2 – lamp.m (cont.)

Result

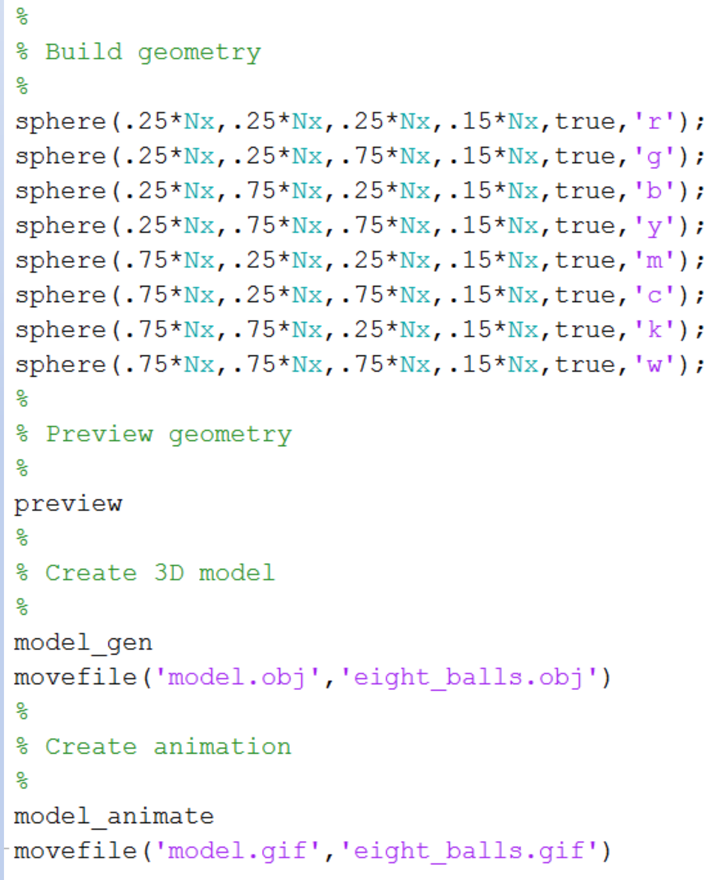




# Example 3 – eight_balls.m



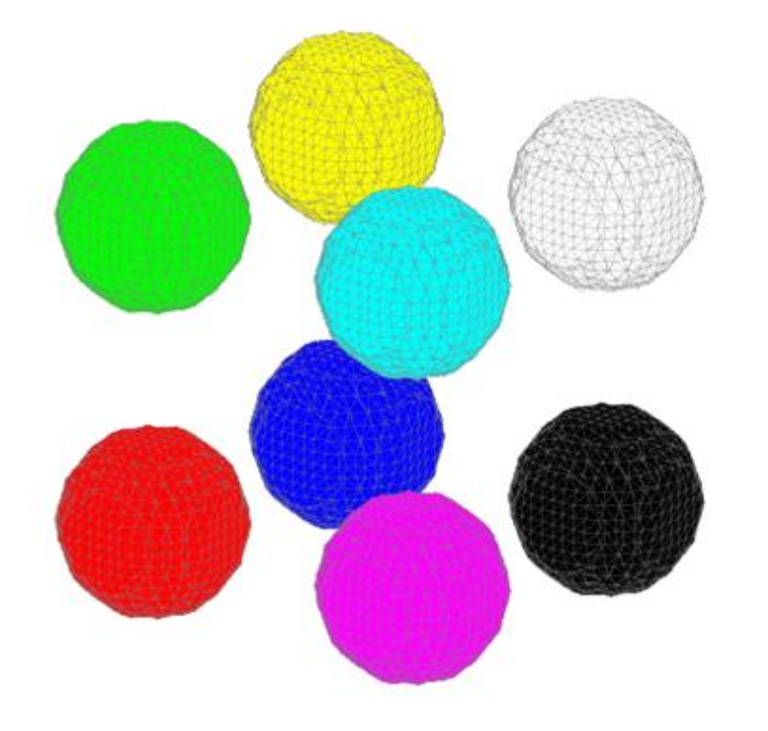
```
function eight_balls
%
clc; clear; close all;
%
global Nx Ny Nz d color
%
% Define design space
%
Nx=50;
Ny=50;
Nz=50;
%
% Initialize array variables
%
d=false(Ny,Nx,Nz);
color=char(zeros(Ny,Nx,Nz));
```

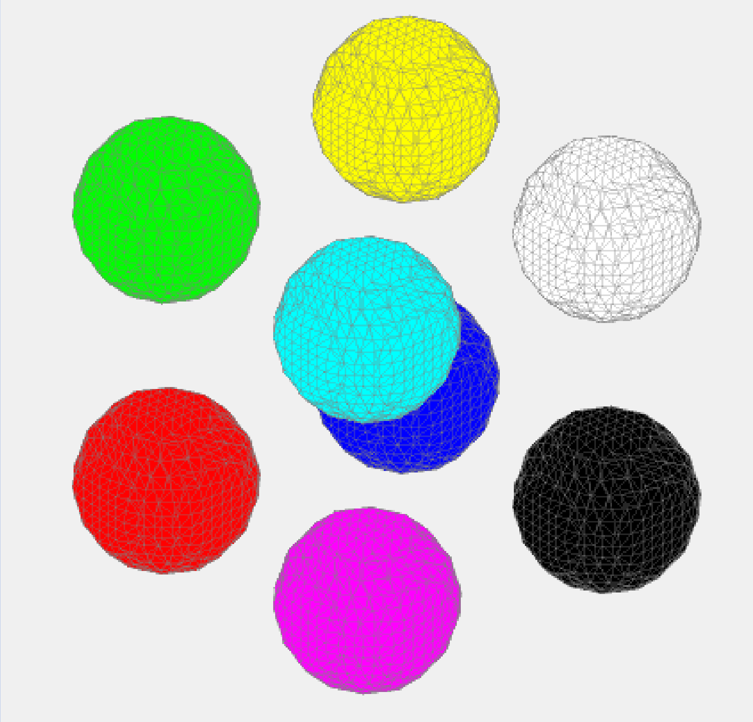


```
%
% Build geometry
%
sphere(.25*Nx,.25*Nx,.25*Nx,.15*Nx,true,'r');
sphere(.25*Nx,.25*Nx,.75*Nx,.15*Nx,true,'g');
sphere(.25*Nx,.75*Nx,.25*Nx,.15*Nx,true,'b');
sphere(.25*Nx,.75*Nx,.75*Nx,.15*Nx,true,'y');
sphere(.75*Nx,.25*Nx,.25*Nx,.15*Nx,true,'m');
sphere(.75*Nx,.25*Nx,.75*Nx,.15*Nx,true,'c');
sphere(.75*Nx,.75*Nx,.25*Nx,.15*Nx,true,'k');
sphere(.75*Nx,.75*Nx,.75*Nx,.15*Nx,true,'w');
%
% Preview geometry
%
preview
%
% Create 3D model
%
model_gen
movefile('model.obj','eight_balls.obj')
%
% Create animation
%
model_animate
movefile('model.gif','eight_balls.gif')
```

# Example 3 – eight_balls.m (cont.)

Result





# Viewing 3D models in MeshLab

Files needed:

- obj file (created by model_gen)
- Color definition files (available on Brightspace)
  - colors.mtl, red.jpg, green.jpg, blue.jpg, yellow.jpg
  - magenta.jpg, cyan.jpg, black.jpg, white.jpg

# Installing and Running Meshlab

Free download from: [http://www.meshlab.net/](http://www.meshlab.net/)

Available on ECC computers

To run:

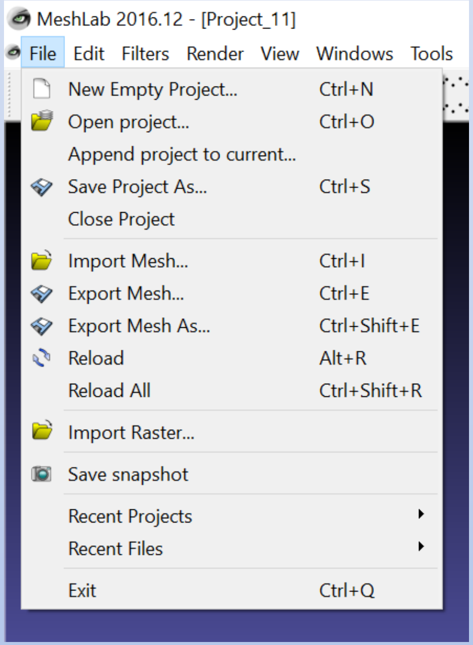
- File => New Empty Project
- File=> Import Mesh
- Navigate and select ‘obj’ file

If you don’t see colors , select

- View => Show Layer Dialog
- Select => Color => Face

To change background color

- Tools =>Options
- Background top => variable value => click => click = > select color
- Background bottom => variable value => click => click = > select color


MeshLab 2016.12 - [Project_11]
File Edit Filters Render View Windows Tools
New Empty Project... Ctrl+N
Open project... Ctrl+O
Append project to current...
Save Project As... Ctrl+S
Close Project
Import Mesh... Ctrl+I
Export Mesh... Ctrl+E
Export Mesh As... Ctrl+Shift+E
Reload Alt+R
Reload All Ctrl+Shift+R
Import Raster...
Save snapshot
Recent Projects
Recent Files
Exit Ctrl+Q

# MeshLab Demonstration

truss.obj

lamp.obj

eight_balls.obj

hw8.obj (last week's assignment)

# Files available on Brightspace (compressed in week_11.zip)

## m files

- truss.m
- lamp.m
- eight_balls.m

## p files*

- box.p
- sphere.p
- cylinder_x.p
- cylinder_y.p
- cylinder_z.p
- segment.p
- preview.p
- model_gen.p
- model_animate.p

## Color definition files

- colors.mtl
- red.jpg
- green.jpg
- blue.jpg
- yellow.jpg
- magenta.jpg
- cyan.jpg
- black.jpg
- white.jpg

*p files – protected mode Matlab function scripts (can be run but source code not viewable, cannot be edited)