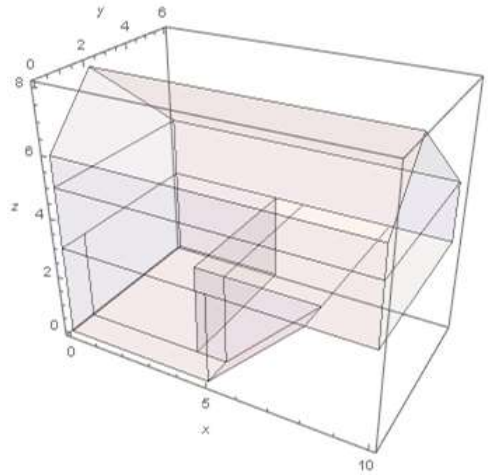
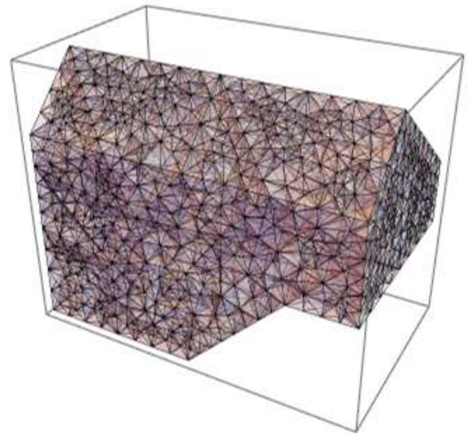




```
48 207
```

Setup and solve the eigenvalue problem

We'll solve for the first `nmodes` modes, and report the compute time (in seconds).

In [17]:
```
nmodes = 20;
AbsoluteTiming[{lambda, eigfuns} = NDEigensystem[{-Laplacian[{u[x, y, z]}, {x, y, z}]}, u[x, y, z],
                                                 Element[{x, y, z}, mesh], nmodes];
              ]
```

Out[17]:
```
{42.2551, Null}
```

Convert eigenvalues into mode frequencies.

In [19]:
```
Grid[Transpose[{Range[nmodes], freqs = c * Sqrt[lambda] / (2 * Pi)}], Alignment -> Right]
```

Out[19]:

1	5.60269×10^{-6}
2	15.1726
3	26.1368
4	28.5475
5	33.1735
6	34.307
7	39.1793
8	41.6354
9	44.4054
10	45.9726
11	48.6021
12	49.4015
13	53.6681
14	55.9967
15	56.5588
16	58.1418
17	59.453
18	61.7291
19	62.7865
20	64.7603

Visualize the room modes

Plot one of the mode shapes (mode number 12) using 3D density plot, and report the compute time.

This actually takes significantly more computation time than solving the eigenvalue problem 😬

In [20]:
```
mode = 12;
```

In [21]:
```
AbsoluteTiming[denplot = DensityPlot3D[eigfuns[[mode]], Element[{x, y, z}, model],
                                       ColorFunction -> ColorData["RedGreenSplit"],
                                       ViewVector -> {15, -15, 15}];
              ]
```

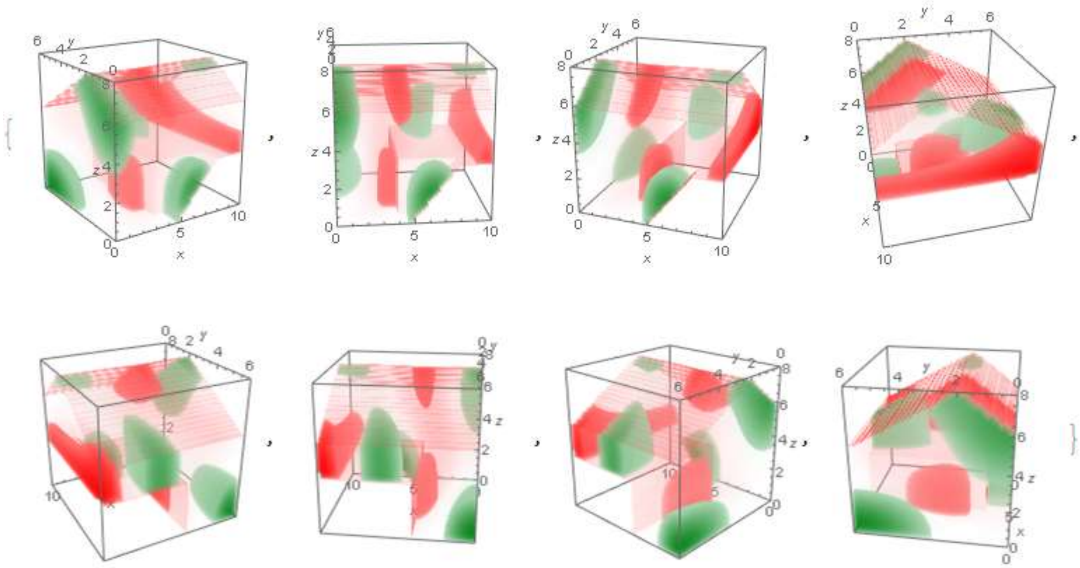
Out[21]:
```
{317.41, Null}
```

If we are using Mathematica, the 3D density plot generated is a 3D object and we can spin it around.
However, Wolfram on Jupyter notebook only supports static pictures, but we can generate multiple views by specifying a series of view point locations.

In [22]:
```
r = 30;
theta = Pi/3;
```

```
Table[Show[denplot, AxesLabel -> {x, y, z},  ViewVector -> FromSphericalCoordinates[{r, theta, phi}]],
      {phi, -3*Pi/4, Pi, Pi/4}]
```

Out[22]:



Other methods to present the results — slice density plots.

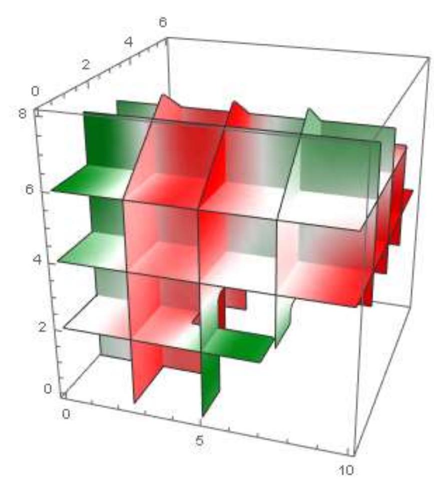
In [25]:

```
slice1 = SliceDensityPlot3D[eigfuns[[mode]], {"XStackedPlanes"}, Element[{x, y, z}, model],
                             ColorFunction -> ColorData["RedGreenSplit"],
                             ViewVector -> {15, -15, 15}];
slice2 = SliceDensityPlot3D[eigfuns[[mode]], {"YStackedPlanes"}, Element[{x, y, z}, model],
                             ColorFunction -> ColorData["RedGreenSplit"],
                             ViewVector -> {15, -15, 15}];
slice3 = SliceDensityPlot3D[eigfuns[[mode]], {"ZStackedPlanes"}, Element[{x, y, z}, model],
                             ColorFunction -> ColorData["RedGreenSplit"],
                             ViewVector -> {15, -15, 15}];
```

In [28]:

```
Show[slice1, slice2, slice3]
```

Out[28]:



A series of slices. Unfortunately, Wolfram doesn't offer slice plots with transparency.

In [29]:

```
Table[SliceDensityPlot3D[eigfuns[[mode]],
                          {{"XStackedPlanes", {0}}, {"YStackedPlanes", {ycoord}}, {"ZStackedPlanes", {0}}},
                          Element[{x, y, z}, model], ColorFunction -> ColorData["RedGreenSplit"],
                          ViewVector -> {15, -15, 15}],
      {ycoord, 0, 6}]
```

Out[29]: