D2 Pascal’s Triangle and Sierpinski Gasket

D2.1 Pascal’s Triangle

Pascal’s triangle is created in the following way. The first row is all zeroes except for a single one. Then every entry in subsequent rows is the sum of the two entries above it. We suppress the zeroes when displaying it.

\[
\begin{array}{cccccccc}
1 \\
1 & 1 \\
1 & 2 & 1 \\
1 & 3 & 3 & 1 \\
1 & 4 & 6 & 4 & 1 \\
1 & 5 & 10 & 10 & 5 & 1 \\
1 & 6 & 15 & 20 & 15 & 6 & 1 \\
1 & 7 & 21 & 35 & 35 & 21 & 7 & 1 \\
1 & 8 & 28 & 56 & 70 & 56 & 28 & 8 & 1 \\
\end{array}
\]

There are many many interesting facts about Pascal’s triangle. The most obvious is that each entry is a binomial coefficient. Here we will just talk about the MATLAB code to generate it and one particular property. If you color in those entries in the triangle that are odd, then you get a fractal-like picture.

There is a MATLAB script named `ParityPascalTriangle.m`.
D2.2 The Sierpinski Gasket

The Sierpinski gasket is a region (or rather a curve) of the plane. It is created by starting with a filled equilateral triangle, and repeatedly performing the following: cut out from every filled triangle (the inside of) the equilateral triangle form by joining the midpoints of the sides. Note that the result has a vanishing small area: it’s area is said to be measure 0.

In theory we could approach drawing this in the same way as we did the Koch curve, by maintaining a list of triangles: each round replacing the one triangle by three smaller triangles. It is easier to define the Gasket by recursion.

In a recursive approach, we say that we cut out the middle triangle and then do the same in all three smaller triangles. Thus we get code like:

```matlab
function SierpinskiErase ( Triangle )
    find midpoints of Triangle and white-out triangle they induce
    for each of the three remaining triangles
        SierpinskiErase ( that triangle )
    end
end
```

In order to avoid the function going on forever, we draw an approximation. For this we go down a certain level. So we keep track of the level as one instance of the function makes use of another. The main MATLAB function for filling in polygons is the `fill` command.