Fractal Program

Variable: N = the number of iterations to create points of the fractal

Two initial shapes set to S0:

- Square – If using this S0, uncomment out A4, B4, W4, and Next.append(W4)
- Triangle – If using this S0, comment out A4, B4, W4, and Next.append(W4)

Variable: a = scale of the scaling matrices

A1, A2, A3, A4 = $\begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}$

Vectors to change the position of the scaled point that can be altered and adjusted:

B1, B2, B3, B4

Two matrices, one must be commented out:

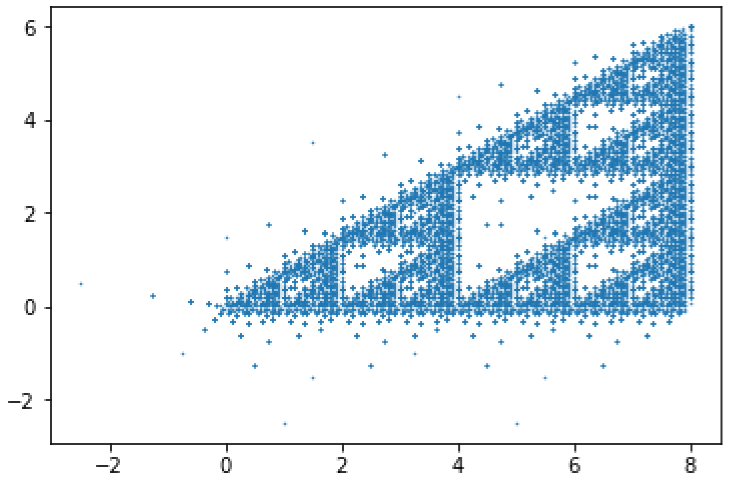
R = a rotational matrix = $\begin{bmatrix} cos(135°) & -sin(135°) \\ sin(135°) & cos(135°) \end{bmatrix}$

R = identity matrix = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

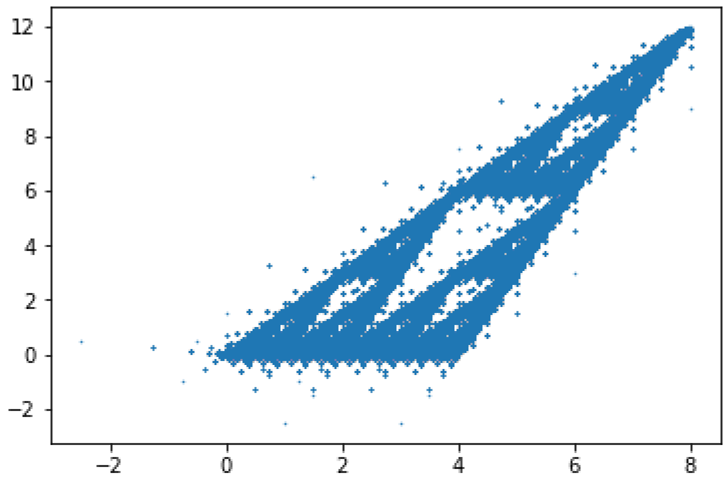
```
S =  copy of S0
Initialize the list Next to hold the new vectors
Initialize n to 0
While n is less than N:
    For each vector in the list S:
        Initialize c and b to be empty lists
        Set v to be a vector from S
        W1 = (R @ A1 @ v) + B1
        W2 = (R @ A2 @ v) + B2
        W3 = (R @ A3 @ v) + B3
        W4 = (R @ A4 @ v) + B4 (commented in or out)
        Append W1, W2, W3, and W4 (if commented in) to the list Next
    For each new vector in the list Next:
        Appending the x-values of the vectors to c
        Appending the y-values of the vectors to b
    Resetting the list S to be equal to the list Next
    Incrementing n by 1
Plotting the points from the final loop from lists c and b
```

For the triangle with the points:

(8,6), (2,-5), (-5,1), (3,-3), (-1.5,-2), (0,3)

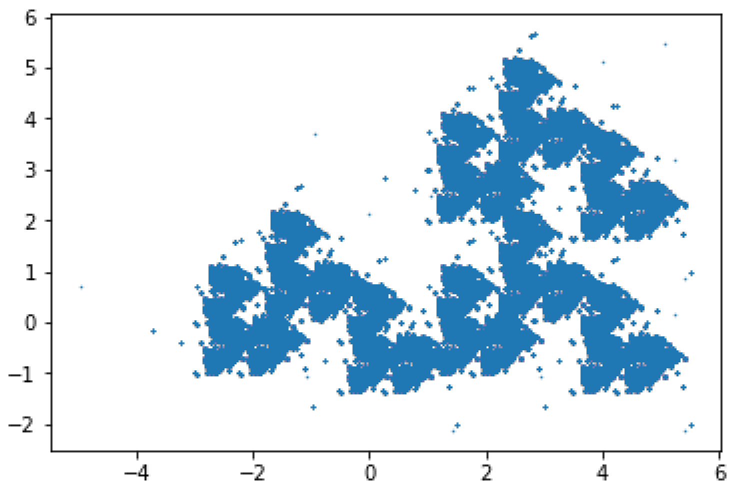


1. It was scaled by ½ and used the vectors $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}4\\0\end{bmatrix}, \begin{bmatrix}4\\3\end{bmatrix}$. The vectors are half the x and y lengths of the triangle. It created a shape similar to the Sierpinski Triangle, but there were quite a few discrepancies with the outlying dots that surrounded the triangle. The left most vertex is placed at (0,0) due to the scaling that scales the dots towards the origin. It has self-similarity.

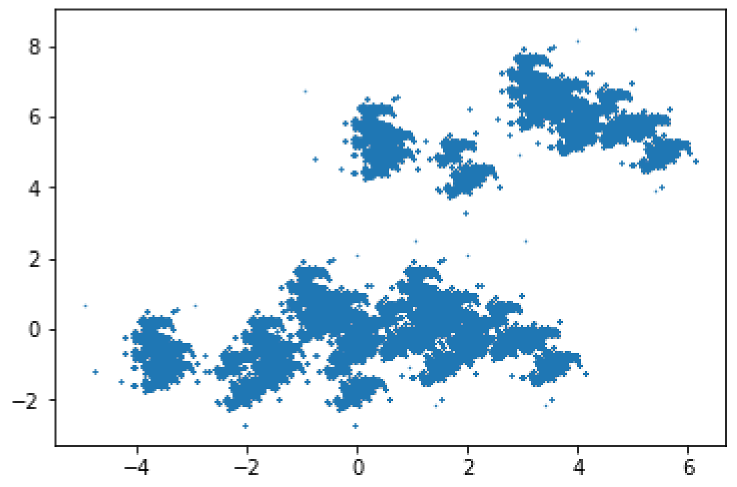


2. It was scaled by ½ and used the vectors $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}4\\6\end{bmatrix}$. The shape is similar to the first triangle but is skewed quite a lot by the vectors. It has self-similarity.

With the rotation:



3. It was scaled by ½ and used the vectors $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}4\\0\end{bmatrix}, \begin{bmatrix}4\\3\end{bmatrix}$. There are small triangular shapes created in the image, but the gaps and triangles do not line up nicely. It has some self-similarity.



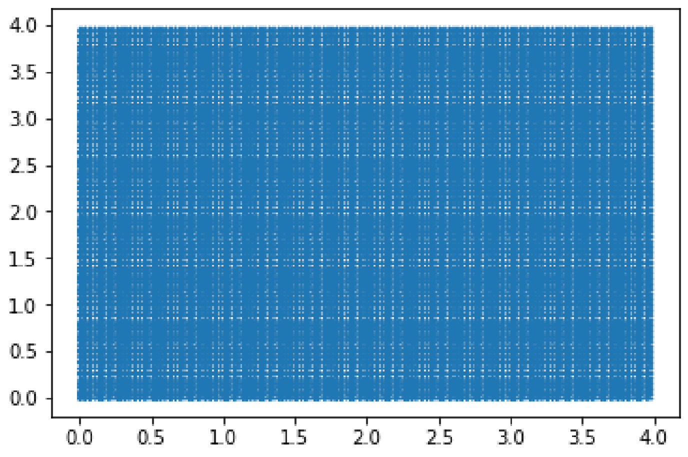
4. It was scaled by ½ and used the vectors $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}4\\6\end{bmatrix}$. It does not have self-similarity or any real pattern.

By scaling and moving points, it creates a fractal of the triangle. But, by adding a rotation to the creation of new points, a nice fractal is lost and is more random.

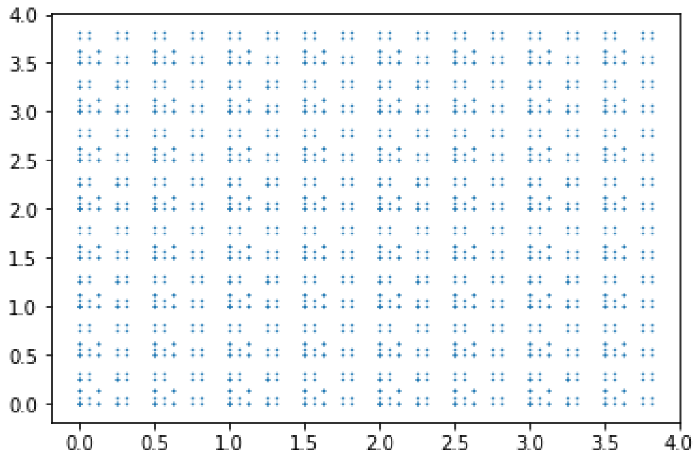
For the square with the points:

(0,0), (0,1), (1,0), (1,1)

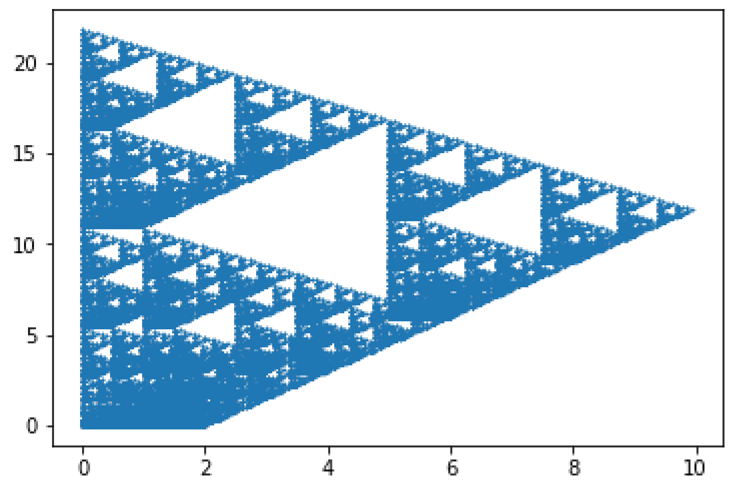
```
Iterations:  7
```



```
In [55]: runfile('C:/Users/lille_000/Documents/Math 361
Documents/Math 361B/FinalProject')
Iterations:  4
```



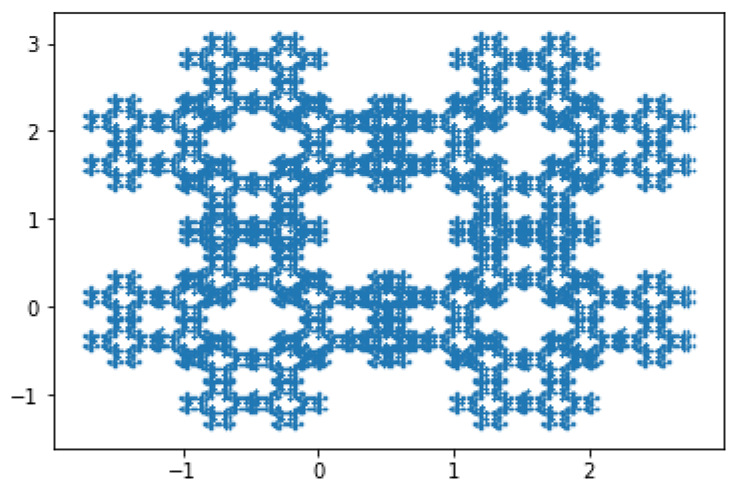
1. It was scaled by ½ and used the vectors $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}$. The iterations created a lot of smaller squares that could only be seen in the first few iterations because with more iterations it turned into a hash-like pattern. As can be seen in the difference of 4 iterations and 7.



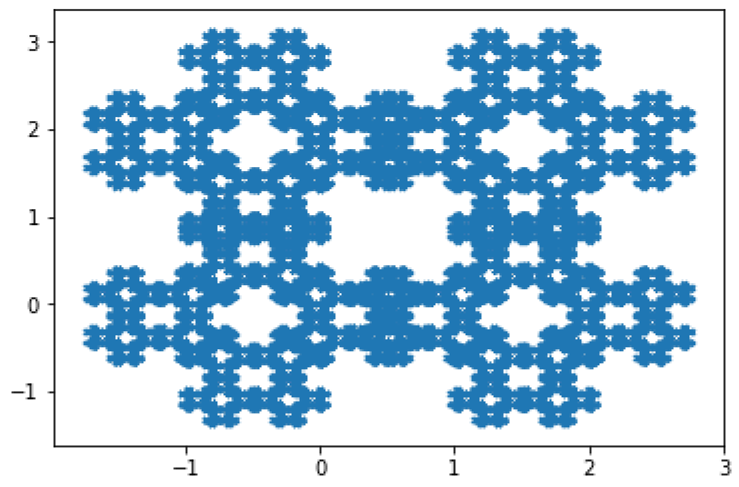
2. It was scaled by ½ and used the vectors $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}5\\6\end{bmatrix}, \begin{bmatrix}0\\11\end{bmatrix}$. The image created is similar to the Sierpinski Triangle but has four sides and skewed triangles. There is no self-similarity to the square, most likely because the squares were changed into quadrilaterals. But, it is interesting that it created triangular gaps.

With the rotation:

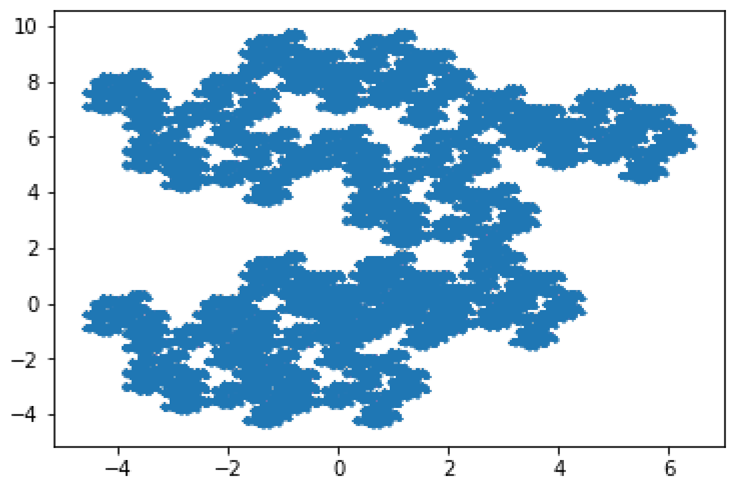
```
Iterations:  6
```



```
In [49]: runfile('C:/Users/lille_000/Documents/Math 361
Documents/Math 361B/FinalProject')
Iterations:  7
```



3. It was scaled by ½ and used the vectors $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}$. The rotation causes a fractal image to be created that appears to be self-similar to the square and there appears to be self-similarity with the shapes created.



4. It was scaled by ½ and used the vectors $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}4\\5\end{bmatrix}, \begin{bmatrix}5\\7\end{bmatrix}$. Changing the vectors disrupts the self-similarity that was created by the rotation. It's mostly random, but there appear to be small clusters of points that are spread out in a non-symmetrically way.