Unit- 5

Curves and Fractals

What are Fractals?

- Fractals are very complex pictures generated by a computer from a single formula.
- They are created using iterations.
- This means one formula is repeated with slightly different values over and over again, taking into account the results from the previous iteration.

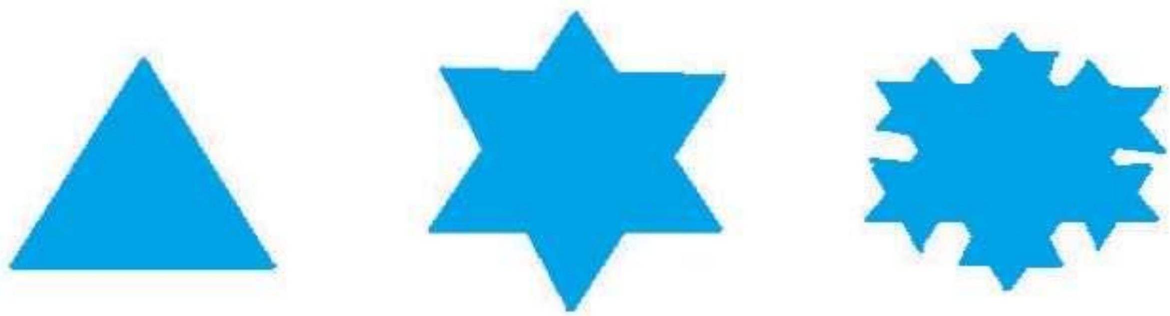
Fractals are used in many areas such as –

- **Astronomy –** For analyzing galaxies, rings of Saturn, etc.
- **Biology/Chemistry –** For depicting bacteria cultures, Chemical reactions, human anatomy, molecules, plants,
- **Others –** For depicting clouds, coastline and borderlines, data compression, diffusion, economy, fractal art, fractal music, landscapes, special effect, etc.

Generation of Fractals:

- Fractals can be generated by repeating the same shape over and over again as shown in the following figure.
- In figure aa shows an equilateral triangle.
- In figure bb, we can see that the triangle is repeated to create a star-like shape.

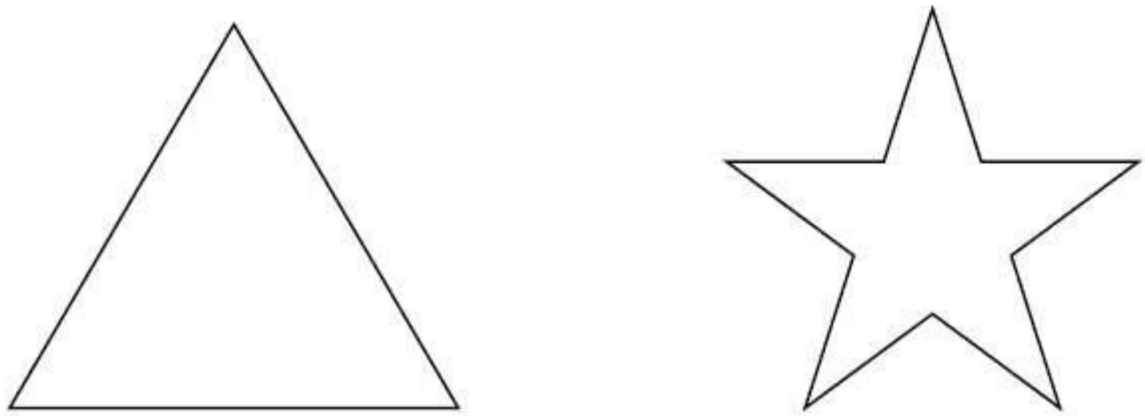
- In figure cc, we can see that the star shape in figure bb is repeated again and again to create a new shape.
- We can do unlimited number of iteration to create a desired shape.
- In programming terms, recursion is used to create such shapes.



(a) Zeroth Generation (b) First Generation (c) Second Generation

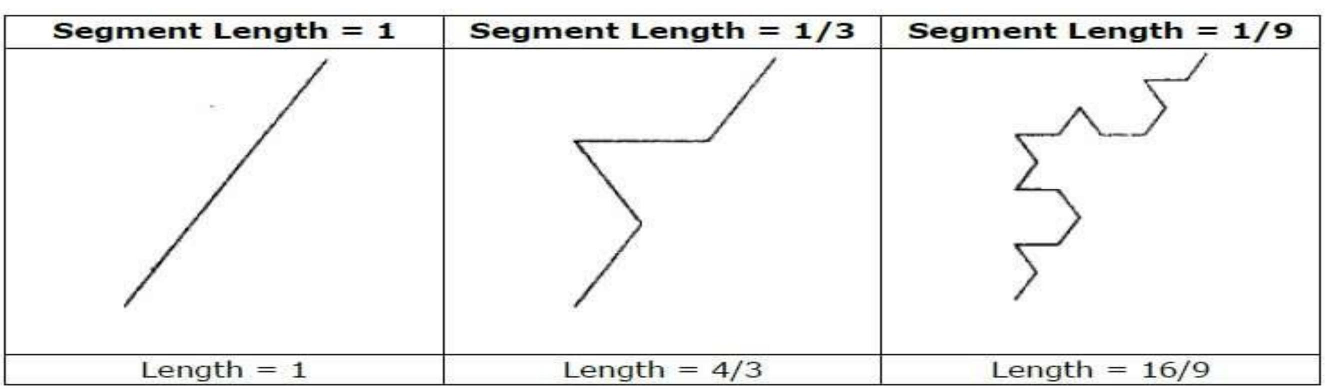
Geometric Fractals

- Geometric fractals deal with shapes found in nature that have non-integer or fractal dimensions.
- To geometrically construct a deterministic non random non random self-similar fractal, we start with a given geometric shape, called the initiator.
- Subparts of the initiator are then replaced with a pattern, called the generator.



Initiator Generator

- As an example, if we use the initiator and generator shown in the above figure, we can construct good pattern by repeating it.
- Each straight-line segment in the initiator is replaced with four equal-length line segments at each step.
- The scaling factor is 1/3, so the fractal dimension is D = ln 4/ln 3 ≈ 1.2619.
- Also, the length of each line segment in the initiator increases by a factor of 4/3 at each step, so that the length of the fractal curve tends to infinity as more detail is added to the curve as shown in the following figure –



Curves:

- In computer graphics, we often need to draw different types of objects onto the screen.
- Objects are not flat all the time and we need to draw curves many times to draw an object.

Types of Curves

- A curve is an infinitely large set of points.
- Each point has two neighbors except endpoints.
- Curves can be broadly classified into three categories − explicit, implicit, and parametric curves.

Implicit Curves

- Implicit curve representations define the set of points on a curve by employing a procedure that can test to see if a point in on the curve.
- Usually, an implicit curve is defined by an implicit function of the form −

$$fx, yx, y = 0$$

- It can represent multi valued curves multiple y values for an x value multiple y values for an x value.
- A common example is the circle, whose implicit representation is

$$x2 + y2 - R2 = 0$$

Explicit Curves

- A mathematical function y = fxx can be plotted as a curve.
- Such a function is the explicit representation of the curve.
- The explicit representation is not general, since it cannot represent vertical lines and is also single-valued.
- For each value of x, only a single value of y is normally computed by the function.

Parametric Curves

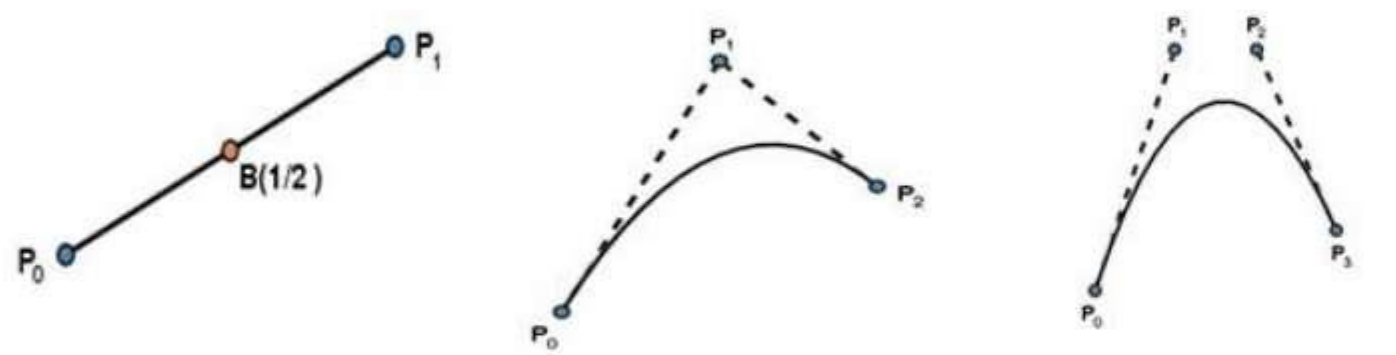
- Curves having parametric form are called parametric curves.
- The explicit and implicit curve representations can be used only when the function is known.
- In practice the parametric curves are used.
- A two-dimensional parametric curve has the following form −

  P t t = f t t, g t t or P t t = x t t, y t t
- The functions f and g become the x,yx,y coordinates of any point on the curve, and the points are obtained when the parameter t is varied over a certain interval [a, b], normally [0, 1].

Bezier Curves

- Bezier curve is discovered by the French engineer Pierre Bézier.

- These curves can be generated under the control of other points.
- Approximate tangents by using control points are used to generate curve.
- Where n is the polynomial degree, i is the index, and t is the variable.
- The simplest Bézier curve is the straight line from the point P0P0 to P1P1.
- A quadratic Bezier curve is determined by three control points.
- A cubic Bezier curve is determined by four control points.



Simple Bezier Curve — Quadratic Bazier Curve — Cubic Bazier Curve

Properties of Bezier Curves

Bezier curves have the following properties −

- They generally follow the shape of the control polygon, which consists of the segments joining the control points.

- They always pass through the first and last control points.
- They are contained in the convex hull of their defining control points.
- The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- A Bezier curve generally follows the shape of the defining polygon.
- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.
- They are invariant under an affine transformation.
- Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
- A given Bezier curve can be subdivided at a point t=t0 into two Bezier segments which join together at the point corresponding to the parameter value t=t0.

B-Spline Curves:

The Bezier-curve produced by the Bernstein basis function has limited flexibility.

- First, the number of specified polygon vertices fixes the order of the resulting polynomial which defines the curve.
- The second limiting characteristic is that the value of the blending function is nonzero for all parameter values over the entire curve.

The B-spline basis contains the Bernstein basis as the special case. The B-spline basis is non-global.

Properties of B-spline Curve

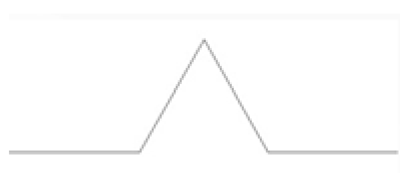
B-spline curves have the following properties −

- The sum of the B-spline basis functions for any parameter value is 1.
- Each basis function is positive or zero for all parameter values.
- Each basis function has precisely one maximum value, except for k=1.
- The maximum order of the curve is equal to the number of vertices of defining polygon.
- The degree of B-spline polynomial is independent on the number of vertices of defining polygon.
- B-spline allows the local control over the curve surface because each vertex affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.

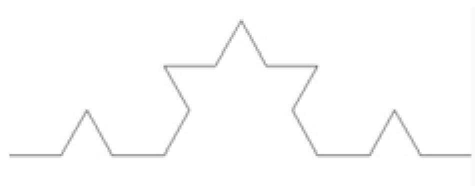
- The curve exhibits the variation diminishing property.
- The curve generally follows the shape of defining polygon.
- Any affine transformation can be applied to the curve by applying it to the vertices of defining polygon.
- The curve line within the convex hull of its defining polygon.

Koch Curve

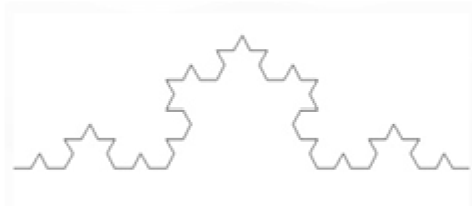
1. The Koch snowflake can be constructed by starting with an equilateral triangle, then recursively altering each line segment as follows:
    - Divide the line segment into three segments of equal length.
    - Draw an equilateral triangle that has the middle segment from step 1 as its base and points outward.



- Remove the line segment that is the base of the triangle from step 2.



- After one iteration of this process, the resulting shape is the outline of a hexagram.



1. The Koch snowflake is the limit approached as the above steps are followed over and over again.
2. The Koch curve originally described by Koch is constructed with only one of the three sides of the original triangle.
3. In other words, three Koch curves make a Koch snowflake.

<u>Hilbert Curve:</u>

- An efficient algorithm for the generation of Hilbert's space-filling curve is given.
- The algorithm implements a recursive procedure that involves simple integer operations and quickly converges to the set of points that make the Hilbert curve.
- The algorithm is elegant, short, and considerably easier to implement than previous recursive and non recursive algorithms and can be efficiently implemented in all programming languages that have integer operations and allow recursion.
- The fundamental Hilbert shape (a line joining the four corners of a square) is represented by two variables with values of either 0 or 1.

- This coding technique could be successfully applied to the generation of other regular space-filling curves, such as the Peano curve.