Chapter - 6 (8 Marks)

Explain the importance of polygon table, plane equation and polygon mesh in surface modeling. [3+3]

What is polygon table? List the rules for making error free polygon table. How do you calculate the spatial orientation of a polygon? [3+2+4]

What is Wire-frame model and why do we need polygon data table? Explain with examples? [5]

b

Why do we use geometric tables and attribute tables for defining a polygon surface? How do you calculate the spatial orientation of a polygon? [3+3]

How can we model cone or cylindrical like surfaces using boundary representation and technique? [6]

6. How can a 3D-Dimensional object be modelled? How a normal to a plane of this object is calculated? [3+3]

How the geometric and attribute information of a 3-D objects are stored for the object representation? Explain with examples. [5]

b

How Geometric tables are used to represent a 3D object? Explain with example. Give conditions to generate error free table. [8]

b

Explain boundary representation technique to represent three dimensional objects with suitable example. [8]

Explain how the geometric and attribute information of a three dimensional objects are stored for the object representation? What are the conditions for error free generation of polygon table? [4+4]

b

Explain the significance of spatial orientation of a surface and polygon tables. Explain with example. [8]

Explain boundary representation technique to represent the 3D object with suitable example. How can you find the spatial orientation of a surface? [8+2]

b

Explain how the 3D object is represented using polygon table representation technique? Explain any one technique to calculate the spatial orientation of the individual surface component of 3D object. [4+4]

Explain boundary representation technique to represent the 3D object with suitable example. How can you find the spatial orientation of a surface? [8+2]

Describe polygon, Vertex and Edge table of polygon. How these terms are important in computer graphics. [8]

Explain how the geometric and attribute information of a three dimensional objects are stored for the object representation? What are the conditions for error free generation of polygon table? [5+3]

Surface modeling involves defining the surfaces of a 3D object. This method focuses on representing the outer shell or boundary of an object, rather than its interior. Surface modeling is widely used in computer-aided design (CAD) and computer graphics because it allows for a high level of detail and smoothness in the representation of complex shapes.

How do you represent an object in 3D? Explain the steps to find surface normal vector of a surface represented by $Ax + By + Cz + D = 0$. [4+4]

Three-Dimensional Object Representations

Three-dimensional object representations in computer graphics are methods used to depict the shape and form of objects in a 3D space. These methods define how the data of 3D objects is stored and manipulated. Common 3D representations include:

- **Wireframe Models**: A skeletal representation of an object using lines and curves, showing only the edges of the object. This is one of the simplest forms and is often used in initial stages of modeling or for quick visualization.

- **Solid Models**: Represent the volume of the object, including both its surface and interior. Solid modeling techniques are more complex and are used for applications where accurate volume and mass properties are required, like in engineering.
- **Surface Models**: As discussed earlier, these focus only on the object's surfaces without considering its volume. This is often sufficient for visual purposes where internal properties are not of interest.

Boundary Representations (B-Rep)

Boundary representation (B-rep) is a method for representing the shape of a 3D object using its boundaries, which include surfaces, edges, and vertices. B-reps describe a solid object by its spatial boundary, which is a collection of surfaces that enclose the volume.

Space Partitioning Representations

Space partitioning representations involve dividing a 3D space into smaller, manageable regions to represent an object. These methods are useful for efficiently managing and querying spatial data, particularly in computer graphics, gaming, and computational geometry.

- **Key Method**:
  - **Octrees**: A hierarchical tree structure where each node represents a cubic region of space. Each node can be subdivided into eight smaller cubes, which allows for efficient representation and manipulation of spatial data.

Polygon Surfaces (Boundary Representation)

Polygon surfaces are a fundamental method for representing 3D objects in computer graphics. A polygon surface is made up of flat, two-dimensional shapes (polygons) that are connected to form the surface of a three-dimensional object. The most common type of polygon used is the **triangle**, due to its simplicity and computational efficiency, although other polygons like quadrilaterals (quads) are also used.

Key Characteristics of Polygon Surfaces:

1. **Vertices**: The corner points of a polygon. Each vertex has coordinates in 3D space (x, y, z).
2. **Edges**: The straight lines that connect pairs of vertices, forming the sides of the polygon.
3. **Faces**: The flat surfaces enclosed by edges. A polygon face is defined by the sequence of its edges or vertices.
4. **Normal Vectors**: A vector perpendicular to the surface of the polygon, used in shading and lighting calculations to determine how light interacts with the surface.

Polygon Table

A **polygon table** is a data structure used in computer graphics to store information about the polygons that make up a 3D object. It is used to efficiently manage and access the data needed for rendering and other operations. The polygon table contains various lists and arrays that store the attributes of polygons, such as vertices, edges, and other relevant properties.

polygon data tables are organized into two groups: geometric tables and attribute tables

1. Geometric Table

The geometric table in computer graphics is used to store information about the geometry of polygons, such as vertices, edges, and surfaces. It typically contains the following sub-tables:

a. Vertex Table

The vertex table stores the coordinates of each vertex in a 3D object. Vertices are points in space that define the corners of polygons.

- **Data stored:** Each entry in the vertex table contains the (x, y, z) coordinates of a vertex.

b. Edge Table

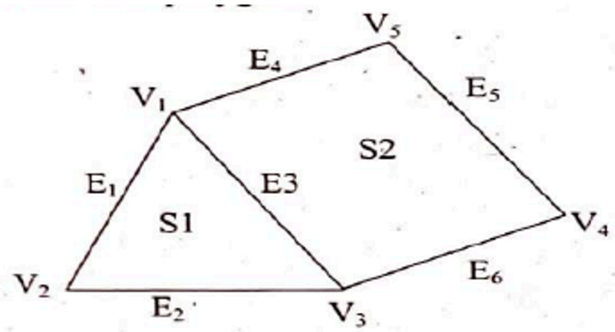
The edge table stores information about the edges of the polygons, which are the line segments connecting pairs of vertices.

- **Data stored:** Each entry in the edge table contains two indices that reference the vertices at the ends of the edge. Additionally, some implementations may store edge attributes like length, visibility, or connectivity information.

c. Polygon (Surface) Table

The polygon or surface table stores information about the polygons themselves, which are typically defined by a list of edges or vertices.

- **Data stored:** Each entry in the polygon table includes a list of indices that reference edges (or sometimes vertices) that form the boundary of the polygon. It may also store the polygon's normal vector, color, texture coordinates, and other attributes necessary for rendering.



Vertex Table
$V_1: x_1, y_1, z_1$
$V_2: x_2, y_2, z_2$
$V_3: x_3, y_3, z_3$
$V_4: x_4, y_4, z_4$
$V_5: x_5, y_5, z_5$

Edge Table
$E_1: V_1, V_2$
$E_2: V_2, V_3$
$E_3: V_3, V1$
$E_4: V_3, V_4$
$E_5: V_4, V_5$
$E_6: V_4, V_3$

Polygon- Surface Table
$S_1: E_1, E_2, E_3$
$S_2: E_3, E_4, E_5, E_6$

Listing the geometric data in three tables provides a convenient reference to the individual components (vertices, edges, and polygons) of each object.

2. Attribute Table

The attribute table in computer graphics is used to store non-geometric data related to the polygons, which may affect how they are rendered. These attributes can include material properties, colors, textures, and shading information.

Guidelines to Generate Error-Free Tables

1. **Ensure consistent indexing**: Verify all indices correctly reference existing vertices and edges.
2. **Maintain proper data formatting**: Use consistent data types and structures across all tables.
3. **Ensure polygons are closed**: Check that all polygons form complete loops with connected edges.
4. **Validate edge and vertex connectivity**: Ensure edges correctly connect vertices to maintain geometric integrity.
5. **Avoid redundant data**: Use indices to reference shared vertices, edges, or attributes, minimizing repetition.
6. **Implement error-checking routines**: Regularly validate data integrity after modifications or transformations.
7. **Keep normal orientations consistent**: Ensure all polygon normals are consistently oriented for correct rendering.

Plane Equations and Their Importance

To display a three-dimensional object, several steps must be undertaken to transform and render the object properly. These steps include:

1. **Transformation**: Convert the model and world coordinate descriptions to viewing coordinates, and then to device coordinates.

2. **Visible Surface Identification**: Determine which surfaces of the object are visible from the viewer's perspective.
3. **Surface Rendering**: Apply rendering techniques to display the object's surfaces correctly.

For these processes, particularly rendering and visible surface determination, we need to know the spatial orientation of each surface component of the object. This is done using the plane equations that describe these surfaces.

Plane Equation

A plane in three-dimensional space can be represented by the equation:

$$Ax + By + Cz + D = 0$$

- (x, y, z) is a point on the plane.
- A, B, C, and D are constants that describe the spatial properties of the plane.

Finding the Plane Coefficients (A, B, C, D)

To determine the values of A, B, C, and D, we need three non-collinear points (vertices) on the plane. These three points, (x_1, y_1, z_1), (x_2, y_2, z_2), and (x_3, y_3, z_3), allow us to set up a system of linear equations:

$$Ax_1 + By_1 + Cz_1 + D = 0$$
$$Ax_2 + By_2 + Cz_2 + D = 0$$
$$Ax_3 + By_3 + Cz_3 + D = 0$$

Let us solve the following simultaneous equations for ratios A/D, B/D, and C/D. You get the values of A, B, C, and D.

$$A/D\, x_1 + B/D\, y_1 + C/D\, z_1 = -1$$

$$A/D\, x_2 + B/D\, y_2 + C/D\, z_2 = -1$$

$$A/D\, x_3 + B/D\, y_3 + C/D\, z_3 = -1$$

To obtain the above equations in determinant form, apply Cramer's rule to the above equations.

$$A = \begin{bmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{bmatrix} B = \begin{bmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{bmatrix} C = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} D = -\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}$$

After expanding the determinants, the coefficients are computed as:

$$\begin{aligned} A &= y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2) \\ B &= x_1(z_2 - z_3) + x_2(z_3 - z_1) + x_3(z_1 - z_2) \\ C &= x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \\ D &= -[x_1(y_2 z_3 - y_3 z_2) + x_2(y_3 z_1 - y_1 z_3) + x_3(y_1 z_2 - y_2 z_1)] \end{aligned}$$

→ determinant nikalne in simple way

These equations allow us to calculate the plane coefficients A, B, C, and D using the vertex coordinates of the polygonal surface.

Normal Vector to the Plane

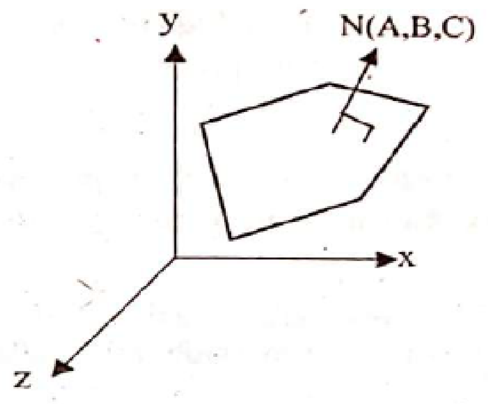


Figure 6.2: *The vector N, normal to the surface of a plane described by the equation $Ax + By + Cz + D = 0$, has Cartesian components (A, B, C)*

The orientation of the plane surface in space is described by the **normal vector**, which has Cartesian components (A, B, C) corresponding to the plane coefficients:

$$\mathbf{N} = (A, B, C)$$

The normal vector is perpendicular to the surface of the plane. In a right-handed coordinate system, if the polygon vertices are listed in a counterclockwise order when viewed from outside, the normal vector points from the inside to the outside of the plane.

Determining the Normal Vector

To determine the normal vector for a shaded surface:

1. Select three vertices along the polygon's boundary in counterclockwise order, as seen from outside the surface.
2. Use the vertices to compute the cross product, which gives the direction of the normal vector.

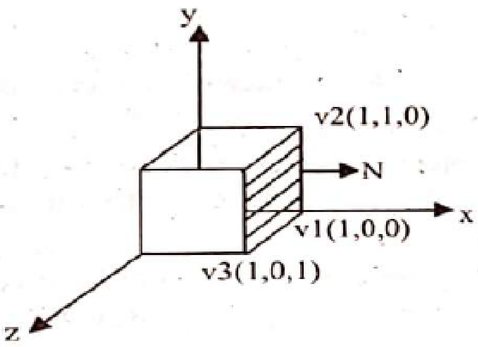


Figure 6.3: *The shaded polygon surface of the unit cube has plane equation x - 1 = 0 and normal vector N = (1, 0, 0)*

Now using the co-ordinate values for these vertices, we solve the equation for A, B, C and D are given.

A = 1, B = 0, C = 0, D = -1

This shows the normal vector is in positive x-direction. we can obtain the element of normal vector by calculating the vector cross product.

We again select any three vertices in counter clockwise direction while viewing from outside and calculate the normal vector as

$$\overrightarrow{N} = (\overrightarrow{v_2} - \overrightarrow{v_1}) \times (\overrightarrow{v_3} - \overrightarrow{v_1})$$

This will give us the direction of $\overrightarrow{N}$

Point Classification Relative to the Plane

Points not on the polygon surface do not satisfy the plane equation $Ax + By + Cz + D = 0$. Instead:

- If $Ax + By + Cz + D < 0$, the point (x, y, z) is **inside** the plane.
- If $Ax + By + Cz + D > 0$, the point (x, y, z) is **outside** the plane.

Alternate Representation of the Plane Equation

The plane equation can also be represented using a **position vector** P for any point on the plane and the normal vector $\mathbf{N}$:

$$\vec{\mathbf{N}} \cdot \vec{\mathbf{P}} = -D$$

Here, $\vec{\mathbf{P}} = (x, y, z)$ is any point on the plane, and $\vec{\mathbf{N}} = (A, B, C)$ is the normal vector.

Polygon Meshes

Polygon meshes are a common method for representing three-dimensional (3D) objects in computer graphics. They are collections of vertices, edges, and faces (polygons) that define the shape of a 3D object. Each polygon in a mesh is typically a triangle or a quadrilateral, but more complex polygons can also be used. Polygon meshes are widely used due to their simplicity and flexibility in modeling a wide variety of shapes and surfaces

Types of Polygon Meshes

1. **Triangle Meshes**:
    - **Description**: In this type of mesh, all the faces are triangles. Triangle meshes are the most common type of polygon meshes used in computer graphics because they are simple to compute and render.
    - **Advantages**: They are highly flexible and can approximate any surface with sufficient triangles. Triangles are also guaranteed to be planar (flat), which simplifies rendering and mathematical operations.
    - **Uses**: Used extensively in real-time graphics, such as video games and virtual reality, where performance is critical.

With n vertices, produce n-2 triangles

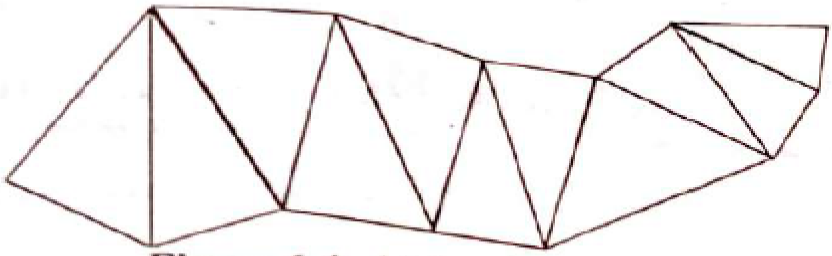


Figure 6.4: A triangle strip formed with 11 triangles 13 vertices

2. **Quadrilateral Meshes (Quads)**:
   - **Description**: In quad meshes, all the faces are quadrilaterals (four-sided polygons). Quads can be beneficial for certain types of modeling and animation because they can represent surfaces with fewer polygons compared to triangles.
   - **Advantages**: Easier to use in surface subdivision algorithms and provide better results in smooth surface modeling and animations, such as character rigs.
   - **Uses**: Common in applications that involve subdivision surfaces, like animation and film production, where smooth transitions are required.

$m = 5$

$n = 4$

Quadrilaterals $= (m-1)(n-1) = 12$

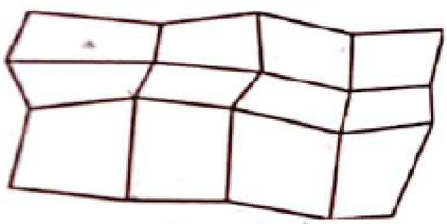


Figure 6.3: A quadrilateral mesh containing 12 quadrilaterals constructed from a 5 by 4 input vertex array

---

b

8. How do surface modeling techniques contribute to realistic rendering and visualization in computer graphics? Explain about blobby object representation. [8]

Surface modeling techniques play a crucial role in achieving realistic rendering and visualization in computer graphics by providing detailed, accurate representations of objects. These techniques define the shape and appearance of objects in a 3D space, allowing for the simulation of real-world surfaces. Here's an overview of how surface modeling techniques contribute to realism and a brief explanation of blobby object representation.

How Surface Modeling Techniques Contribute to Realistic Rendering:

1. **Accurate Shapes**: Surface modeling techniques help create precise shapes and curves that mimic real-world objects, making them look more realistic in a 3D space.
2. **Smooth Surfaces**: Techniques like splines and NURBS create smooth, continuous surfaces that are important for realistic shading and lighting effects, especially for organic shapes like faces or car bodies.
3. **Texture and Material Application**: These models allow for the application of textures (like wood grain or skin patterns) and materials (like metal or glass), which add detail and realism to objects.
4. **Efficient Rendering**: Advanced modeling methods can adjust the level of detail based on the viewer's distance, optimizing rendering time without losing realism.
5. **Realistic Light Interaction**: Surface models provide detailed geometry that helps in simulating how light reflects, refracts, or scatters, which is crucial for realistic effects.
6. **Natural Movement and Deformation**: Surface models can deform smoothly, making them ideal for animations where objects bend, twist, or stretch naturally, like in character movements.

Blobby Object Representation:

1. **What It Is**: Blobby object representation, or metaballs, is a modeling technique that creates smooth, organic shapes using mathematical functions. These objects are defined by fields that blend together seamlessly.

2. **Smooth Blending**: Blobby objects blend smoothly when they are close to each other, making them ideal for representing soft, merging shapes like liquids or muscle tissues.
3. **Uses**: This technique is often used in animations for fluid effects, character modeling for soft tissues, and scientific visualizations for things like molecules.
4. **Realism**: By allowing objects to merge and split in a smooth, natural way, blobby object representation helps create more lifelike animations and visual effects.

In summary, surface modeling techniques, including blobby object representation, are key to creating realistic graphics by accurately modeling shapes, applying textures, simulating light, and creating smooth deformations and animations.

In surface rendering, the polygon table, plane equations, and polygon meshes play crucial roles in defining how 3D objects are represented and displayed. Here's why each of these elements is important:

1. Polygon Table

Purpose:

- The polygon table is essential for managing and organizing the polygons that make up a 3D object's surface. It provides a structured way to reference which vertices and edges form each polygon, which is vital for rendering the object correctly.

Importance:

- **Efficient Rendering**: By having a clear record of which vertices form which polygons, rendering engines can efficiently process and draw each polygon.
- **Data Management**: Helps in organizing geometric data and ensures that all polygons are correctly defined.
- **Facilitates Operations**: Allows for easy updates and modifications to the object, such as transformations or optimizations.

2. Plane Equation

Purpose:

- The plane equation is used to mathematically describe the flat surface of a polygon. In 3D graphics, it is crucial for various calculations, including clipping, intersection, and lighting.

Importance:

- **Clipping**: Determines whether parts of a polygon are visible within a view frustum and should be rendered.
- **Intersection**: Helps in calculating intersections between rays (e.g., camera rays or light rays) and surfaces, which is fundamental for rendering algorithms like ray tracing.
- **Lighting Calculations**: Essential for computing how light interacts with surfaces, including determining surface normals and shading effects.

3. Polygon Mesh

Purpose:

- A polygon mesh is a collection of polygons that together form a 3D object. It represents the surface geometry of the object and is the fundamental structure used in most 3D modeling and rendering systems.

Importance:

- **Complexity and Detail**: Polygon meshes allow for the creation of detailed and complex surfaces by combining multiple polygons. This is essential for achieving realistic representations of objects.
- **Surface Approximation**: Meshes approximate curved surfaces using polygons, enabling the representation of complex shapes.
- **Rendering Efficiency**: Modern graphics hardware is optimized to handle polygon meshes efficiently, making them suitable for real-time rendering.

Spacial orientation of polygon can be calculated by ⇒

Vertex Ordering: The spatial orientation of a polygon depends on the order in which its vertices are defined. In a clockwise (CW) or counterclockwise (CCW) order, the orientation will differ, affecting how the polygon faces are rendered.

Surface Normal: The orientation can be determined by calculating the surface normal vector. This vector is perpendicular to the surface of the polygon and can be calculated using the cross product of two edges of the polygon. For a polygon with vertices V_1, V_2, V_3:

$$\text{Normal} = (V_2 - V_1) \times (V_3 - V_1)$$

Winding Order: The direction of the normal vector (pointing inward or outward) depends on the winding order of the vertices. For instance, in a right-handed coordinate system, if the vertices are ordered counterclockwise, the normal will point out of the surface.

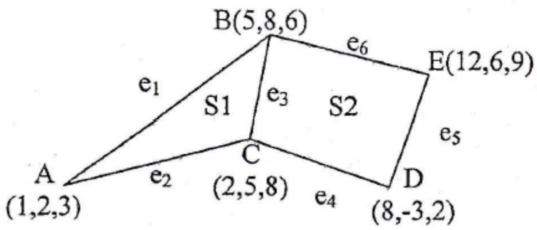
To summarize, spatial orientation is mainly calculated through the vertex order and the resulting surface normal vector, which determines how the polygon is oriented in 3D space.

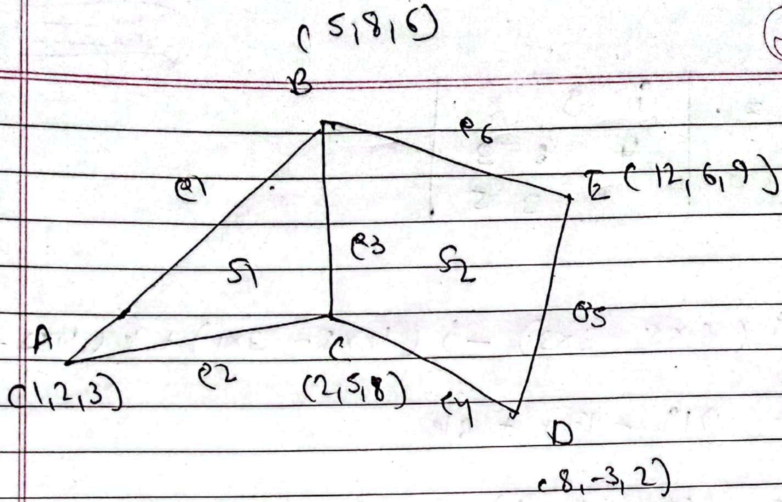
---

b

passes through (0,0,0) and (-2,1,1) and is controlled by (7,5,2) and (2,0,1). [2+6]

6. Represent the following surfaces by polygon table method and find the normal of surface S1. [2+5]





Vertex Table
A (1,2,3)
B (5,8,6)
C (2,5,8)
D (8,-3,2)
E (12,6,9)

Edge Table
E_1 : AB
E_2 : AC
E_3 : BC
E_4 : CD
E_5 : DE
E_6 : ~~or~~ BE

Surface Table
S_1 : $E_1 E_2 E_3$
S_2 : E_3, E_4, E_5, E_6

S_1 is defined by the points A (1,2,3) B (5,8,6) and C (2,5,8)

N to S_1 is :- $N = \vec{AB} \times \vec{AC}$

$\vec{AB} = \vec{B} - \vec{A} = (5-1, 8-2, 6-3) = (4, 6, 3)$

$\vec{AC} = \vec{C} - \vec{A} = (2-1, 5-2, 8-3) = (1, 3, 5)$

Date
Page

$$\vec{N} = \begin{vmatrix} i & j & k \\ 4 & 6 & 3 \\ 1 & 3 & 5 \end{vmatrix}$$

$N = i(6\times5 - 3\times3) - j(4\times5 - 3\times1) + k(4\times3 - 6\times1)$

$= 21i - 17j + 6k$

$\vec{N} = (21, -17, 6)$

to S1

(alternatively you ~~can~~ can take cross product of AB and BC or AC and BC)

Do you agree Polygon Descriptions are referred to as "Standard Graphics Object", If yes, Why? If you have three coordinates (X1, Y1, Z1), (X2, Y2, Z2) and (X3, Y3, Z3), then how do you find the coefficient of Surface Normal N(A, B, C)? [3+3]

Yes, polygon descriptions are commonly referred to as "Standard Graphics Objects" in computer graphics because they are fundamental building blocks for representing 3D shapes. Polygons, particularly triangles, are used to model surfaces in 3D space because they are simple to process and render.

To find the coefficients of the surface normal $N(A, B, C)$ given three coordinates $(X1, Y1, Z1)$, $(X2, Y2, Z2)$, and $(X3, Y3, Z3)$, you can use the cross product of two vectors that lie on the surface of the polygon.

1. First, form two vectors from the three points:

$$\text{Vector 1} = (X2 - X1, Y2 - Y1, Z2 - Z1)$$

$$\text{Vector 2} = (X3 - X1, Y3 - Y1, Z3 - Z1)$$

- Then, find the cross product of these two vectors to get the normal vector $\mathbf{N} = (A, B, C)$:

$$\mathbf{N} = \mathbf{v}_1 \times \mathbf{v}_2$$

$$\mathbf{N} = ((Y_2 - Y_1)(Z_3 - Z_2) - (Z_2 - Z_1)(Y_3 - Y_2),$$

$$(Z_2 - Z_1)(X_3 - X_2) - (X_2 - X_1)(Z_3 - Z_2),$$

$$(X_2 - X_1)(Y_3 - Y_2) - (Y_2 - Y_1)(X_3 - X_2))$$

- The coefficients A, B, and C correspond to the components of this normal vector.