IB Mathematics: Analysis and Approaches HL

Internal Assessment

Approximating Curved Surface Areas of 3D Shapes Using Triangular Meshes and Vector Cross Products

I) Introduction

In many real-world contexts, we can only work with digital or physical models of curved objects rather than with their exact continuous surfaces, which makes numerical approximation of surface area an essential tool. Applications such as Computer graphics and 3D modelling software rely on subdividing curved surfaces into simpler elements whose areas are easier to compute and sum, so understanding how such subdivision behaves mathematically can clarify when these approximations are reliable and how fine a mesh is needed for a given accuracy.

In this investigation, I approximate each smooth surface with a triangular mesh, since triangles are the simplest 3D polygons and their areas can be computed directly using vector cross products without advanced surface integrals.

Aim

This investigation approximates the curved surface areas of a sphere, a cylinder, and a torus using triangular mesh subdivisions, calculating total area via vector cross products. The accuracy of these approximations is evaluated against exact surface area formulas to analyse how increasing mesh refinement reduces error.

Parameters and Limits

For consistency and comparability across shapes, the following dimensional parameters are fixed throughout this investigation:

- Sphere: $R_{Sphere} = 6$
- Cylinder: $r = 4$, $h = 9$
- Torus: Major radius $R_{Torus} = 8$, Minor radius $r_{Torus} = 2$

A torus is a three-dimensional donut-shaped surface formed by rotating a circle of radius r around an axis in the same plane at distance R; it has no singular points and can be smoothly parameterised using two angular variables, making it particularly suitable for triangular mesh approximations.

In each case, the curved surface is represented by a two-parameter vector-valued function **r(u, v)** that assigns each ordered pair **(u, v)** in a rectangular parameter domain to a unique point **(x, y, z)** in three-dimensional space on the surface. For the sphere, the parameters are two angular variables; for the cylinder, an angular variable and a linear height coordinate; and for the torus, two angular variables.

This ensures that every point on each surface corresponds uniquely to a single pair of parameter values within the specified ranges.

Three levels of mesh refinement are applied to each shape:

• Coarse mesh: Low grid resolution

• Medium mesh: Moderate grid resolution

• Fine mesh: High grid resolution

For each shape, the parameter domain is subdivided into a regular **m × n** grid, where **m** denotes the number of equal divisions in the first parameter and **n** denotes the number of equal divisions in the second parameter.

Each rectangular grid cell is then divided into two triangles using a fixed diagonal, meaning that the total number of triangles depends directly on the values of **m** and **n**. The specific values of **m** and **n** corresponding to the three levels of mesh refinement are summarised in Table 2, and these parameters determine both the structure of the mesh and the total number of triangles used in the subsequent calculations.

II) Mathematical Background

Vector Cross Product

The cross product of two vectors in three-dimensional space produces a third vector perpendicular to both. Given two vectors v_1 and v_2, their cross product $v_1 \times v_2$ has magnitude equal to the area of the parallelogram spanned by the two vectors (Stewart 805).

For vectors $v_1 = (a_1, b_1, c_1)$ and $v_2 = (a_2, b_2, c_2)$, the cross product is computed as:

$$v_1 \times v_2 = \begin{pmatrix} b_1c_2 - c_1b_2 \\ c_1a_2 - a_1c_2 \\ a_1b_2 - b_1a_2 \end{pmatrix}$$

The magnitude of this vector is:

$$\|v_1 \times v_2\| = \sqrt{[(b_1c_2 - c_1b_2)^2 + (c_1a_2 - a_1c_2)^2 + (a_1b_2 - b_1a_2)^2]}$$

Triangle Area Justification:

Consider a triangle with vertices P_1, P_2, and P_3 in three-dimensional space.

Two edge vectors can be defined as:

$v_1 = P_2 - P_1$

$v_2 = P_3 - P_1$

The area of the triangle is exactly half the area of the parallelogram formed by v_1 and v_2:

A_parallelogram = $\|v_1 \times v_2\|$

A_triangle = ½ $\|v_1 \times v_2\|$

This formula is geometrically justified because the cross product magnitude gives the parallelogram area, and a triangle occupies half that area (Larson and Edwards 782).

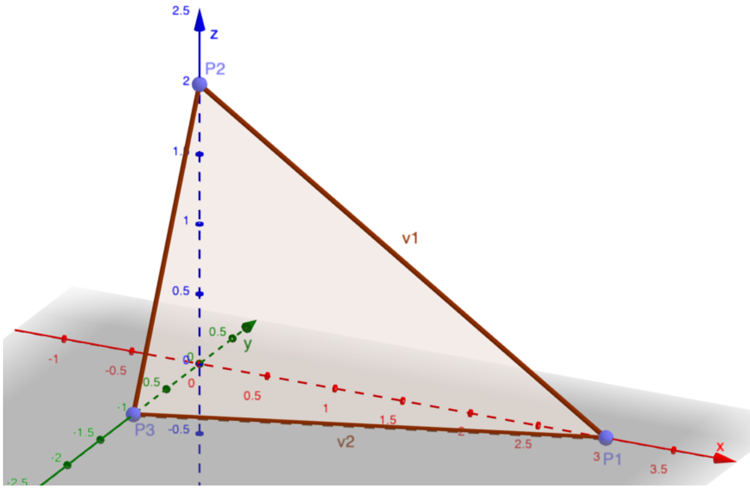

Figure 1: Triangle Area via Cross Product

Exact Surface Area Formulas

The exact surface areas for the three shapes under investigation are given by well-known geometric formulas (Clapham and Nicholson 512):

Sphere:

$$S = 4\pi(R_{Sphere})^2$$

For R_{Sphere} = 6:

$$S = 4\pi(6)^2 = 144\pi \approx 452.389 \text{ (3 decimal places)}$$

Cylinder (curved surface area):

$$S = 2\pi rh$$

For r = 4, h = 9:

$$S = 2\pi(4)(9) = 72\pi \approx 226.195 \text{ (3 decimal places)}$$

Torus:

$$S = 4\pi^2 R_{Torus}\, r_{Torus}$$

For R_{Torus} = 8, r_{Torus} = 2:

$$S = 4\pi^2(8)(2) = 64\pi^2 \approx 631.655$$

These exact values serve as benchmarks against which the triangular mesh approximations are compared.

Subdivision Concept

The fundamental approach is to approximate a smooth curved surface by covering it with a mesh of flat triangular facets. Each triangle lies in a plane tangent or nearly tangent to the surface at that region. As the mesh becomes finer, containing more triangles of a smaller size, the approximation converges toward the true surface area. This links the vector cross-product formula for the area of a single triangle to a global approximation method for curved surfaces, where the total approximated area is the sum of the areas of all triangles in the mesh.

The construction process involves:

1. Parameterisation: Express the surface using two parameters (u, v) that map to three-dimensional coordinates (x, y, z).
2. Grid Generation: Divide the parameter space into a regular rectangular grid.
3. Vertex Placement: Compute the 3D coordinates of grid vertices using the parameterization.
4. Triangulation: Split each rectangular grid cell into two triangles by adding a diagonal.
5. Area Calculation: Use the vector cross product formula to compute the area of each triangle.
6. Summation: Add all triangular areas to obtain the total approximated surface area.

Refinement Levels

Three levels of mesh refinement are defined by the number of divisions along each parameter:

Table 1: Parameter definitions

Shape	**Parameter 1**	**Range**	**Parameter 2**	**Range**	**Description**
Sphere	θ	$0 \le \theta \le 2\pi$	$\emptyset$	$0 \le \emptyset \le \pi$	θ: Rotation around vertical axis $\emptyset$: Angle from positive z-axis
Cylinder	θ	$0 \le \theta \le 2\pi$	y	$0 \le y \le h$ (h: height of cylinder)	θ: Rotation around axis y: Vertical height from base of cylinder
Torus	θ	$0 \le \theta \le 2\pi$	$\emptyset$	$0 \le \emptyset \le 2\pi$	θ: Rotation around central axis $\emptyset$: Rotation around tube

Table 2: Mesh refinement values

Shape	**Coarse mesh (divisions)**	**Medium mesh (divisions)**	**Fine mesh (divisions)**
Sphere	12 x 6	24 x 12	48 x 24
Cylinder	12 x 6	24 x 12	48 x 24
Torus	12 x 12	24 x 24	48 x 48

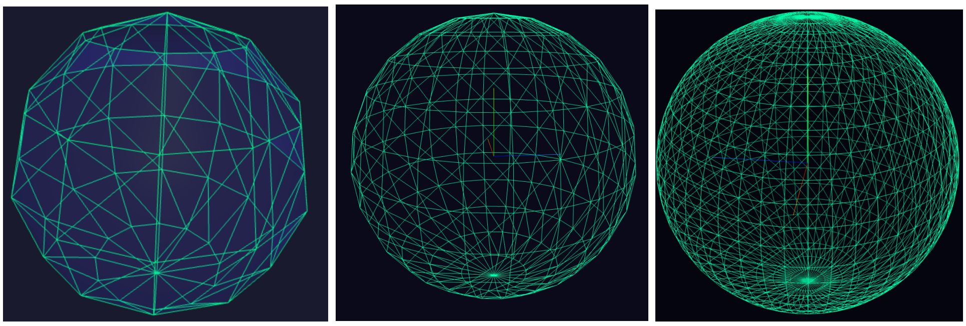

Figure 2: Spheres (coarse → medium → fine)

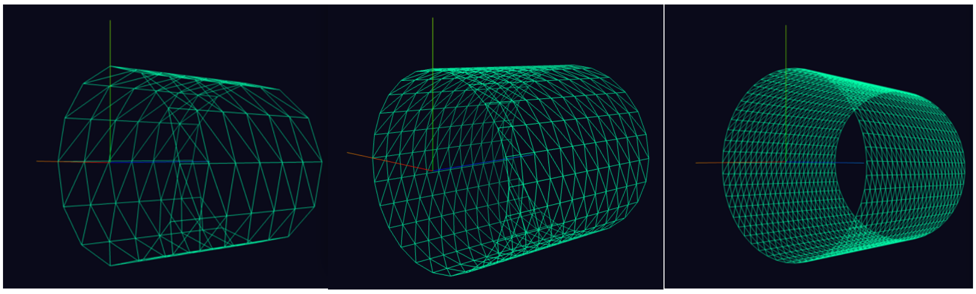

Figure 3: Cylinders (coarse → medium → fine)

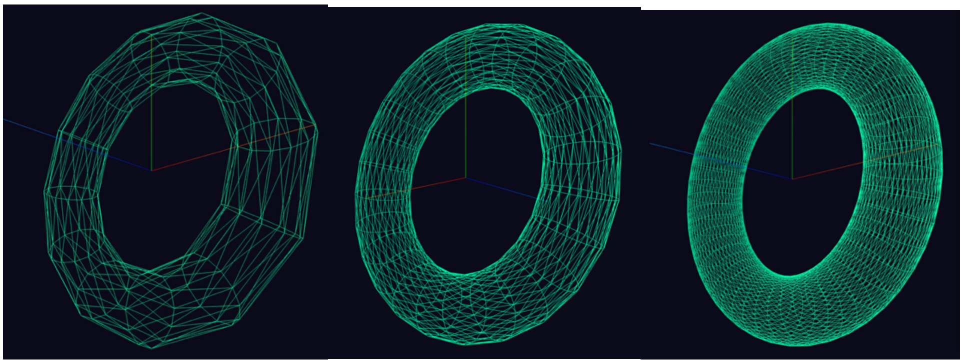

Figure 4: Torus (coarse → medium → fine)

<u>Sphere</u>:

m = divisions in θ (longitude), n = divisions in $\emptyset$ (latitude)

<u>Cylinder</u>:

m = divisions in θ (around the axis), n = divisions in y (height)

<u>Torus</u>:

m = divisions in θ (around the central axis), n = divisions in $\emptyset$ (around the tube)

For the sphere, the parameter grid forms m segments around θ. However, at $\emptyset = 0$ and $\emptyset = \pi$, the grid collapses to single points (the poles of the sphere), so the usable "bands" between poles are $n - 1$. Each band has m cells, and each cell is split into two triangles:

$$N_{\Delta,\text{sphere}} = 2m(n - 1)$$

Considering the cylinder, the grid forms m subdivisions around the circumference and n subdivisions along the height, giving mn rectangular cells. Each cell is split into two triangles:

$$N_{\Delta,\text{cylinder}} = 2mn$$

The surface of a torus is periodic in both parameters, so the rectangular parameter grid wraps around in two directions. This produces mn rectangular cells, each split into two triangles:

$$N_{\Delta,\text{torus}} = 2mn$$

<u>Calculation of total triangles:</u>

Sphere:

Coarse mesh: $N = 2m(n-1) = 2(12)(6-1) = 2(12)(5) = 120$

Medium mesh: $N = 2(24)(12-1) = 2(24)(11) = 528$

Fine mesh: $N = 2(48)(24-1) = 2(48)(23) = 2208$

Cylinder:

Coarse mesh: $N = 2mn = 2(12)(6) = 144$

Medium mesh: $N = 2(24)(12) = 576$

Fine mesh: $N = 2(48)(24) = 2304$

Torus:

Coarse mesh: $N = 2mn = 2(12)(12) = 288$

Medium mesh: $N = 2(24)(24) = 1152$

Fine mesh: $N = 2(48)(48) = 4608$

<u>Explanation of shape appearances:</u>

Parameterisation is the process of representing a curved surface using two variables that uniquely determine the three-dimensional coordinates of each point on the surface, allowing the surface to be systematically divided into a grid for approximation and calculation.

Sphere:

The sphere is parameterised using two angular parameters, longitude $\theta \in [0,2\pi]$ and latitude $\phi \in [0,\pi]$, which map each point on the surface to three-dimensional Cartesian coordinates. The parameter space is divided into a rectangular $m \times n$ grid, where m represents divisions in the longitudinal direction and n represents divisions in the latitudinal direction. Each rectangular grid cell between adjacent latitude bands is subdivided into two triangles using a consistent diagonal. Special consideration is required at the north and south poles, where lines of longitude converge to a single point. These regions are handled using triangular fan structures, ensuring that no degenerate rectangular cells are formed. Figure 2 illustrates how increasing values of m and n produce progressively finer triangular approximations of the spherical surface.

Cylinder:

The curved lateral surface of the cylinder is parameterised using an angular coordinate $\theta \in [0,2\pi]$ around the circular cross-section and a linear height coordinate $y \in [0,h]$. Together, these parameters map each point on the surface to three-dimensional Cartesian coordinates. The parameter domain is divided into a rectangular $m \times n$ grid, where m denotes the number of divisions in the angular direction and n denotes the number of divisions along the height. Each rectangular grid cell between adjacent height levels is subdivided into two triangles using a consistent diagonal, producing a uniform triangular mesh over the entire lateral surface. Because the cylinder has no singular points on its curved surface, the same triangulation rule applies throughout, and the mesh wraps seamlessly when θ reaches 2π. Figure 3 shows how increasing values of m and n generate progressively finer triangular approximations of the cylindrical surface.

Torus:

The surface of the torus is parameterised using two angular variables, $\theta \in [0,2\pi]$ and $\phi \in [0,2\pi]$, where θ represents rotation about the central axis of the torus and ϕ represents rotation around the circular cross-section of the tube. The parameter domain forms a rectangular grid with m divisions in the θ-direction and n divisions in the ϕ-direction. Each rectangular grid cell in the (θ, ϕ) parameter space is subdivided into two triangles using a consistent diagonal, producing a uniform triangular mesh over the entire toroidal surface. Because the torus is periodic in both parameters, the grid wraps seamlessly in both directions, and no boundaries or singular points require special treatment. Figure 4 illustrates how the mesh becomes progressively denser as the grid resolution is refined.

IV) Modelling & Calculations

Before modelling each individual shape, it is useful to describe the common structure of the meshes used. Each mesh is generated from a rectangular grid in the surface's parameter space: I define *m* as the number of equal divisions in the first parameter and *n* as the number of equal divisions in the second parameter. A mesh described as *m* x *n* therefore has *m* segments in the first parameter direction and *n* segments in the second, and each rectangular parameter cell is split into two triangles using a consistent diagonal. This construction determines both the total number of triangles and the approximate shape and distribution of the triangular facets for all three surfaces.

4.1 **Sphere**

<u>Parameterisation of the Sphere</u>

The spherical surface is represented using the vector-valued function

$$\mathbf{r}(\theta, \phi) = (R \sin\phi \cos\theta\,,\ R \sin\phi \sin\theta\,,\ R\cos\ \phi),$$

where

$$\theta \in [0,2\pi], \phi \in [0, \pi].$$

For every ordered pair (θ, ϕ), the function $\mathbf{r}(\theta, \phi)$ outputs the Cartesian coordinates (x, y, z) of a point on the surface of a sphere with radius R. The parameter θ controls rotation around the vertical axis, while ϕ controls the vertical position on the sphere from the top pole to the bottom pole. By varying both parameters across their ranges, the entire curved surface of the sphere is generated.

For this investigation, the radius of the sphere is fixed as $R_{sphere} = 6$.

<u>Vertex Generation</u>

To construct a triangular mesh, the parameter ranges are divided into equal intervals. For a mesh defined by m divisions in θ and n divisions in ϕ, the step sizes are

$$\Delta\theta = \frac{2\pi}{m}, \Delta\phi = \frac{\pi}{n}.$$

Vertices are generated at parameter values

$$\theta_i = i\Delta\theta (i = 0,1, \dots, m-1),$$

$$\phi_j = j\Delta\phi (j = 0,1, \dots, n).$$

Each vertex on the surface is therefore given by

$$P_{i,j} = \mathbf{r}(\theta_i, \phi_j).$$

Since all longitude lines converge at $\phi = 0$ and $\phi = \pi$, these values correspond to the north and south poles of the sphere and require special handling during triangulation.

<u>Triangulation Rule</u>

Between adjacent latitude bands ϕ_j and ϕ_{j+1}, rectangular parameter cells are formed. Each rectangular cell is subdivided into two triangles using a consistent diagonal. At the poles, where rectangular cells collapse into triangular regions, triangular fan structures are used to connect each latitude ring to the pole without creating degenerate triangles. For the sphere, the parameter grid forms **m** segments in the θ direction. At $\phi = 0$ and $\phi = \pi$, all lines of longitude converge to a single point, corresponding to the poles of the sphere. As a result, there are $n-1$ usable bands between the poles, with each band containing **m** rectangular cells. Since each rectangular cell is subdivided into two triangles, the total number of triangles in the spherical mesh is

$$\mathrm{N}_{\Delta,sphere} = 2m(n-1)$$

Worked Example: One Triangle (Coarse Mesh)

For the coarse mesh, the values $m = 12, n = 6$ are used.

The corresponding step sizes are

$$\Delta\theta = \frac{2\pi}{12} = \frac{\pi}{6}, \Delta\phi = \frac{\pi}{6}.$$

Consider a triangle formed between the parameter points:

$$(\theta_0, \phi_1), (\theta_1, \phi_1), (\theta_0, \phi_2).$$

Substituting values:

$$\theta_0 = 0, \theta_1 = \frac{\pi}{6}, \phi_1 = \frac{\pi}{6}, \phi_2 = \frac{\pi}{3}.$$

The Cartesian coordinates of the vertices are:

$$P_1 = \mathbf{r}(0, \frac{\pi}{6}) = (6\sin\frac{\pi}{6}, 0, 6\cos\frac{\pi}{6}) = (3, 0, 3\sqrt{3}),$$

$$P_2 = \mathbf{r}(\frac{\pi}{6}, \frac{\pi}{6}) = \left(3\cos\frac{\pi}{6}, 3\sin\frac{\pi}{6}, 3\sqrt{3}\right) = \left(\frac{3\sqrt{3}}{2}, \frac{3}{2}, 3\sqrt{3}\right),$$

$$P_3 = \mathbf{r}(0, \frac{\pi}{3}) = (6\sin\frac{\pi}{3}, 0, 6\cos\frac{\pi}{3}) = (3\sqrt{3}, 0, 3).$$

Two edge vectors are then defined:

$$\mathbf{v}_1 = P_2 - P_1, \mathbf{v}_2 = P_3 - P_1.$$

The area of the triangle is calculated using the cross product formula:

$$A_{\Delta} = \frac{1}{2} \| \mathbf{v}_1 \times \mathbf{v}_2 \|.$$

This method is applied identically to all triangles in the mesh.

Total Area Summation

The approximated surface area of the sphere for a given mesh level is obtained by summing the areas of all individual triangles:

$$S_{\text{approx}} = \sum A_{\Delta}.$$

This summation is performed computationally using Python with NumPy to ensure numerical accuracy. The exact surface area of the sphere is given by

$$S_{\text{exact}} = 4\pi R^2 = 144\pi.$$

Table 3a: Approximated Surface Area of the sphere

Mesh level	Triangles	Approximated area (units²)	Exact area (units²)
Coarse	120	427.006	144π
Medium	528	445.959	144π
Fine	2208	450.776	144π

Error Calculations:

The accuracy of the approximation is measured using absolute and percentage error:

$$E_{\text{abs}} = | S_{\text{approx}} - S_{\text{exact}} |,$$

$$E_{\text{rel}} = \frac{E_{\text{abs}}}{S_{\text{exact}}} \times 100\%$$

Table 3b: Approximate Error for the Sphere

Mesh level	Absolute error	Percentage error (%)
Coarse	25.383	5.61
Medium	6.431	1.42
Fine	1.613	0.36

4.2 **Cylinder** (Curved Surface Only)

Parameterisation of the Cylinder

The curved (lateral) surface of the cylinder is represented using the vector-valued function

$$\mathbf{r}(\theta, y) = (r\cos\theta\,,\, r\sin\theta,\, y),$$

where

$$\theta \in [0, 2\pi], y \in [0, h].$$

This expression is read as follows: for each ordered pair (θ, y), the function $\mathbf{r}(\theta, y)$ outputs the Cartesian coordinates (x, y, z) of a point on the curved surface of the cylinder. The parameter θ controls the position around the circular cross-section, while the parameter y determines the vertical position along the height of the cylinder. Varying both parameters across their ranges generates the entire lateral surface, excluding the top and bottom circular faces.

For this investigation, the dimensions of the cylinder are fixed as $r = 4, h = 9$.

Vertex Generation

For a mesh defined by m divisions in θ and n divisions along the height, the parameter step sizes are

$$\Delta\theta = \frac{2\pi}{m}, \Delta y = \frac{h}{n}.$$

Vertices are generated at

$$\theta_i = i\Delta\theta(i = 0,1, \dots, m-1),$$

$$y_j = j\Delta y(j = 0,1, \dots, n).$$

Each vertex on the curved surface is given by

$$P_{i,j} = \mathbf{r}(\theta_i, y_j).$$

Since the surface wraps continuously around the circumference and has no singular points, the grid structure remains uniform across the entire surface.

Triangulation Rule

The cylindrical surface is parameterised by angular coordinate θ and linear height coordinate y. The surface is periodic in the θ-direction, so the first and last angular divisions are joined to form a closed circumference, while the height direction remains non-periodic. The parameter grid consists of m subdivisions around the circumference and $n$subdivisions along the height, producing mn rectangular cells.

$$N_{\Delta,\text{cylinder}} = 2mn.$$

Worked Example: One Triangle (Coarse Mesh)

For the coarse mesh,

$$m = 12, n = 6,$$

giving

$$\Delta\theta = \frac{\pi}{6}, \Delta y = \frac{9}{6} = 1.5.$$

Consider the triangle formed by the parameter points

$$(\theta_0, y_0),\ (\theta_1, y_0),\ (\theta_0, y_1),$$

where

$$\theta_0 = 0, \theta_1 = \frac{\pi}{6}, y_0 = 0, y_1 = 1.5.$$

The Cartesian coordinates of the vertices are:

$$P_1 = (4,0,0),$$

$$P_2 = \left(4\cos\frac{\pi}{6}, 4\sin\frac{\pi}{6}, 0\right) = (2\sqrt{3}, 2,0),$$

$$P_3 = (4,0,1.5).$$

Two edge vectors are defined as

$$\mathbf{v}_1 = P_2 - P_1, \mathbf{v}_2 = P_3 - P_1.$$

The area of the triangle is calculated using

$$A_\Delta = \frac{1}{2} \parallel \mathbf{v}_1 \times \mathbf{v}_2 \parallel.$$

This procedure is applied identically to all triangles in the cylindrical mesh.

The area of each triangle is calculated using

$$A_{\Delta} = \frac{1}{2} \parallel \mathbf{v}_1 \times \mathbf{v}_2 \parallel.$$

This calculation is applied to every triangle in the cylindrical mesh using the corresponding edge vectors, and the resulting triangle areas are summed to obtain the total approximated curved surface area.

Total Surface Area Approximation

The approximated curved surface area of the cylinder is obtained by summing the areas of all triangular facets:

$$S_{\text{approx}} = \sum A_{\Delta}.$$

This summation is performed computationally using Python and NumPy.

The exact curved surface area of the cylinder is

$$S_{\text{exact}} = 2\pi r h = 72\pi.$$

Table 4a: Approximated Curved Surface Area of the cylinder

Mesh level	Number of triangles	Approximated area	Exact area
Coarse	144	223.620	72π
Medium	576	225.549	72π
Fine	2304	226.033	72π

Error Calculations

The approximation accuracy is measured using

$$E_{\text{abs}} = |\, S_{\text{approx}} - S_{\text{exact}} \,|,$$

$$E_{\text{rel}} = \frac{E_{\text{abs}}}{S_{\text{exact}}} \times 100\%.$$

Table 4b: Approximated error of the cylinder

Mesh level	Absolute error	Percentage error (%)
Coarse	2.575	1.14
Medium	0.645	0.29
Fine	0.161	0.07

As the mesh resolution increases, the percentage error decreases, indicating improved approximation of the curved surface.

4.3 **Torus**

<u>Parameterisation of the Torus</u>

The torus is represented using the vector-valued function

$$\mathbf{r}(\theta, \phi) = ((R + r\cos\ \phi)\cos\ \theta,\ (R + r\cos\ \phi)\sin\ \theta,\ r\sin\ \phi),$$

Where $\theta \in [0,2\pi], \phi \in [0,2\pi]$.

This expression is read as follows: for each ordered pair (θ, ϕ), the function $\mathbf{r}(\theta, \phi)$ produces the Cartesian coordinates of a point on the surface of a torus. The parameter θ controls rotation around the central axis of the torus, while ϕ controls rotation around the circular cross-section of the tube. Varying both parameters across their ranges generates the entire closed surface.

For this investigation, the radii are fixed as $R = 8, r = 2$.

<u>Vertex Generation</u>

For a mesh defined by m divisions in θ and n divisions in ϕ, the parameter step sizes are

$$\Delta\theta = \frac{2\pi}{m}, \Delta\phi = \frac{2\pi}{n}.$$

Vertices are generated at

$$\theta_i = i\Delta\theta (i = 0,1, \dots, m-1),$$

$$\phi_j = j\Delta\phi (j = 0,1, \dots, n-1).$$

Each vertex is given by

$$P_{i,j} = \mathbf{r}(\theta_i, \phi_j).$$

Due to periodicity in both parameters, the grid wraps seamlessly in both directions and contains no boundaries or singular points.

<u>Triangulation Rule</u>

The torus is parameterised by two angular variables, making the surface periodic in both directions. As a result, the rectangular parameter grid wraps seamlessly around the surface in both parameters, forming a fully closed mesh with no boundary edges. This grid produces mn rectangular cells distributed uniformly across the toroidal surface.

$$N_{\Delta,\text{torus}} = 2mn.$$

<u>Worked Example: One Triangle (Coarse Mesh)</u>

For the coarse mesh,

$$m = 12, n = 12,$$

giving

$$\Delta\theta = \frac{\pi}{6}, \Delta\phi = \frac{\pi}{6}.$$

Consider the triangle formed by the parameter points

$$(\theta_0, \phi_0), (\theta_1, \phi_0), (\theta_0, \phi_1),$$

where

$$\theta_0 = 0, \theta_1 = \frac{\pi}{6}, \phi_0 = 0, \phi_1 = \frac{\pi}{6}.$$

The Cartesian coordinates of the vertices are:

$$P_1 = (10,0,0),$$

$$P_2 = (10\cos\frac{\pi}{6}, 10\sin\frac{\pi}{6}, 0) = (5\sqrt{3}, 5, 0),$$

$$P_3 = (8 + 2\cos\frac{\pi}{6}, 0, 2\sin\frac{\pi}{6}) = (8 + \sqrt{3}, 0, 1).$$

Two edge vectors are defined as

$$\mathbf{v}_1 = P_2 - P_1, \mathbf{v}_2 = P_3 - P_1.$$

The area of the triangle is calculated using

$$A_\triangle = \frac{1}{2} \| \mathbf{v}_1 \times \mathbf{v}_2 \|.$$

Total Surface Area Approximation

The approximated surface area of the torus is obtained by summing the areas of all triangular facets:

$$S_{\text{approx}} = \sum A_\triangle.$$

This summation is performed computationally using Python and NumPy.

The exact surface area of the torus is

$$S_{\text{exact}} = 4\pi^2 Rr = 64\pi^2.$$

Table 5a: Approximated Curved Surface Area of the torus

Mesh level	Number of triangles	Approximated area	Exact area
Coarse	288	606.883	$64\pi^2$
Medium	1152	625.372	$64\pi^2$
Fine	4608	630.078	$64\pi^2$

Error Calculations

The approximation accuracy is measured using

$$E_{\text{abs}} = | S_{\text{approx}} - S_{\text{exact}} |,$$

$$E_{\text{rel}} = \frac{E_{\text{abs}}}{S_{\text{exact}}} \times 100\%.$$

Table 5b: Approximated error of the torus

Mesh level	**Absolute error**	**Percentage error (%)**
Coarse	24.772	3.92
Medium	6.283	0.99
Fine	1.576	0.25

As the mesh resolution increases, the percentage error decreases, demonstrating convergence of the triangular approximation.

V) Results

Summary Table: Approximated Surface Areas and Errors

Shape	**Refinement**	**Triangles**	**Approximated Area (sq units)**	**Exact Area (sq units)**	**Absolute Error (sq units)**	**Percentage Error (%)**
Sphere	Coarse	120	427.006	452.389	25.383	5.61
Sphere	Medium	528	445.959	452.389	6.431	1.42
Sphere	Fine	2208	450.776	452.389	1.613	0.36
Cylinder	Coarse	144	223.620	226.195	2.575	1.14
Cylinder	Medium	576	225.549	226.195	0.645	0.29
Cylinder	Fine	2304	226.033	226.195	0.161	0.07
Torus	Coarse	288	606.883	631.655	24.772	3.92
Torus	Medium	1152	625.372	631.655	6.283	0.99
Torus	Fine	4608	630.078	631.655	1.576	0.25

Observations:

• **Sphere**: The percentage error decreases substantially from 5.61% for the coarse mesh to 0.36% for the fine mesh. This indicates steady convergence of the triangular approximation as the number of surface facets increases.

•**Cylinder**: The cylinder exhibits the lowest percentage error at all refinement levels, decreasing from 1.14% to 0.07%. This suggests that the curved surface of a cylinder is approximated very efficiently by planar triangular facets.

• **Torus**: The torus shows a clear reduction in error from 3.92% to 0.25% as mesh resolution increases, demonstrating convergence despite its more complex surface geometry.

Comparative Analysis

Overall, all three shapes display a consistent trend of decreasing percentage error with increasing mesh refinement. However, the **rate of convergence differs between shapes**. The cylinder converges most rapidly, followed by the torus, while the sphere exhibits the slowest convergence. This suggests that the geometric properties of a surface influence how effectively it can be approximated using flat triangular elements.

VI) Reflection & Evaluation

Accuracy Improvement:

For all three shapes, increasing the number of triangular facets led to a clear reduction in percentage error, demonstrating improved accuracy with mesh refinement.

For the **sphere**, the percentage error decreased from 5.61% for the coarse mesh to 0.36% for the fine mesh. This represents a reduction of approximately 93.6% in error.

For the **cylinder**, the percentage error decreased from 1.14% to 0.07%**,** corresponding to a reduction of approximately 93.9%**.**

For the **torus**, the percentage error decreased from 3.92% to 0.25%**,** representing a reduction of approximately 93.6%**.**

Although the percentage reduction is similar across shapes, the **absolute error and rate of** convergence differ. The error reduction does not appear linear with respect to the number of triangles; instead, the error decreases more rapidly as the mesh becomes finer. This behaviour is consistent with numerical approximation theory, where polygonal surface approximations typically converge non-linearly as mesh resolution increases (Press et al.). No explicit convergence order was calculated, as this investigation focuses on numerical comparison rather than formal error modelling.

Straight-Edge vs. Curved Surface Error:

A fundamental limitation of this method is that **triangular facets are flat**, while the surfaces being approximated are curved. Each triangle approximates a small portion of the surface using straight edges, which introduces error whenever the surface has curvature.

The **sphere** has positive curvature at every point, meaning that no region of the surface is locally flat. As a result, planar triangles consistently underestimate the true surface area unless the mesh is sufficiently refined.

The **cylinder** has curvature in only one direction and can be unrolled into a flat rectangle without distortion. This geometric property explains why the cylinder exhibits the lowest approximation error at all mesh levels.

The **torus** has varying curvature across its surface, with regions of higher and lower curvature. This leads to intermediate error behaviour between the sphere and the cylinder.

These differences in geometric structure directly influence how effectively each surface can be approximated using flat triangular elements.

Grid Limitations:

The rectangular parameter grids used in this investigation impose several limitations. First, equal spacing in parameter space does not always correspond to equal spacing on the actual surface, particularly for the sphere and torus. Second, the triangulation topology is fixed, meaning that all grid cells are subdivided in the same manner regardless of local curvature. Third, boundaries or singular regions may introduce distortion when mapped onto a flat grid. These limitations could be addressed by using adaptive mesh refinement, where smaller triangles are placed in regions of higher curvature. However, such methods fall outside the scope of this investigation and the IB Mathematics AA HL syllabus.

Shape-Specific Issue: Sphere Pole Singularities:

For the sphere, the parameterisation introduces singular points at the north and south poles, where multiple grid lines converge. Near these regions, triangles become narrower and less uniform, which can reduce local approximation accuracy. This effect contributes to the relatively higher error observed for the sphere compared to the cylinder at equivalent mesh resolutions.

Shape Comparison

Among the three shapes, the **cylinder** was the easiest to approximate, exhibiting the smallest errors at all refinement levels. The **sphere** was the most difficult to approximate due to its constant curvature in all directions. The **torus** showed intermediate difficulty, as its curvature varies across the surface.

Ranking the shapes by approximation difficulty:

1. Cylinder (easiest)
2. Torus
3. Sphere (hardest)

This ranking aligns with the geometric properties of each surface and highlights the relationship between curvature and approximation accuracy.

VII) Conclusion

Across all three shapes, increased mesh refinement led to consistently lower percentage error, demonstrating convergence of the approximation method. The cylinder showed the lowest errors at all resolutions, the sphere the highest, and the torus intermediate values, reflecting their differing curvature properties: the cylinder's developable surface is easier to approximate than the sphere's fully curved surface, with the torus lying between. The investigation also confirmed the validity of using the vector cross product to compute surface area via triangulation, a method widely used in computer graphics and engineering. Limitations include uniform grids and fixed triangulation patterns, and further work could explore adaptive meshes or alternative parameterisations to improve accuracy.

Works Cited

Appendix