**Modelling 1 (implicit)**

**2.1 Introduction to Modelling**

Modelling discusses the definition and representations of virtual objects inside the computer.

For an object we have:

1. Geometric Shape
2. Material Properties - Mass (weight/hardness)
3. Texture – Colours, pattern

Geometric Modelling

For example, to represent a sphere we use the following equation:

This will represent the boundary of a sphere, and we’ll need:

1. Centre point of the sphere, (x0, y0, z0)
2. Radius, r

The equation above is known as an implicit representation of a sphere as it doesn’t explain to us the relation between x,y and z.

An explicit representation is known as a function like f(x,y) = z. However, x,y is a 2D plane, which for **any closed object like a sphere** **won’t be able to define a simple function** as there will be two points that go through it.

**Method (implicit to explicit)**

Rearrange the implicit expression making z the subject of the formula.

We will then find points that lie on the surface and use it to create a picture, we create a picture by connecting 3 points (graphically) to create a triangle (triangle since it will always remain a triangle, where as a square can form different shape).

**2.2 Affine Space**

To represent the geometry of an object we need to define the position of its **points**.

Reminder:

Let

For example: ||v|| = (2,3, -√3) = √ ((22) + (3)2 + (-√3)2) = 4

The issue with vector, is there they don’t have a position or a **point**, in order to represent an object, we must have a position.

Points doesn’t have a direction or size, it’s just a location, and a point doesn’t require coordinate system (coordinate system is a type of **affine** space)

By point 3 we can interoperate a vector v as connecting two points this means:

Points Vs Vector

* A point is defined by its position in the Affine space (one attribute)
* A Vector has a magnitude ||v|| (NORM) and unit direction v/||v|| (no position)
* You can add a vector to a point in A or a vector to a Vector, but not a point to a point.

Example:

If you’re given a spanning set V, and a point p over a Euclidean space then to find the points for which the spanning set can “reach”.

**2.3 Coordinate System**

They are define using an origin point and a set of axis. A point **represents a specific position in any space regardless of the coordinate system**.

**A point won’t move if the coordinate systems moves, only the representation of its coordinate under the chosen system will change.**

So we first define an origin point P0, and every point from the origin p, the difference between these points = VECTOR, and then we also need a base of the vector space.

**Method – finding angle between two vectors:**

In 3D (3,2,2) can represent a vector in the direction (3/√17, 2/√17, 2/√17) – (√17 comes from the normalisation calculation, and the vector shown is its unit vector)

In 3D can represent a point whose position to some origin point is (3,2,2).

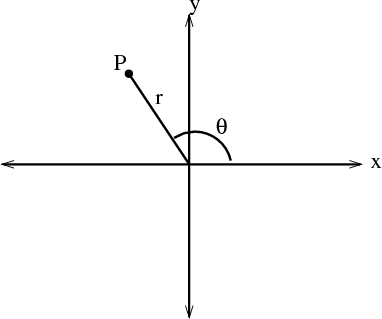
Affine combination occurs when .

This means that given a set of n points p1, … pn in A, every point in A can be defined by a unique set of an affine coefficients whose sum is by the equation.

Coordinate System in Graphics/Modelling

Coordinate Polar

|  |  |  |
| --- | --- | --- |
| y |  |  |
|  |  |  |
|  |  | x |



To add vectors 🡪 cartesian

To Measure Angles between vectors 🡪 Polar

To Measure Lengths of vectors 🡪 Polar

**Method:** Converting between 2D polar and 2D coordinate

**Given** **Formula** **Description**

|  |  |  |
| --- | --- | --- |
| **Given Cartesian (x, y)** | **Formula** | **Description** |
| r  (radius) | r = | Distance from the origin |
| (angle) |  | Angle in radians (adjust for quadrants |

|  |  |  |
| --- | --- | --- |
| **Given Polar (r, )** | **Formula** | **Description** |
| x |  | Convert radius and angle to x-coordinate |
| y |  | Convert radius and angle to y-coordinate |

|  |  |  |
| --- | --- | --- |
| **Quadrant** | **Condition** | **Correction for** |
| **I** (x > 0, y > 0) | No change |  |
| **II** (x < 0, y > 0) | Add |  |
| **III** (x < 0, y < 0) | Add |  |
| **IV** (x > 0, y < 0) | Add  if needed | or |

If you’re provided with polar coordinate and need to do an addition – it’ll be fast to add them as polar coordinates than doing the conversion computation and conversion back.

**Method:** Converting between 3D polar and 3D coordinate