# 1 Intersection of a ray with a triangle

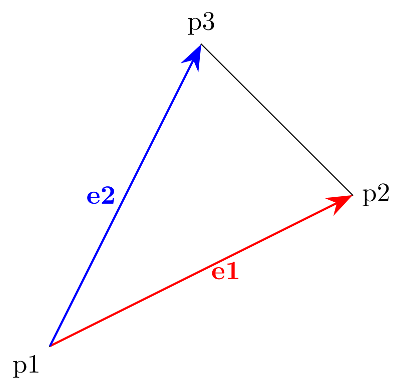
This implementation uses the Möller–Trumbore intersection algorithm.

Given a ray defined by its origin **o** and direction **d**, and a triangle defined by its vertices p1, p2, and p3:

## 1.1 Compute the edge vectors

$$\mathbf{e1} = p2 - p1$$

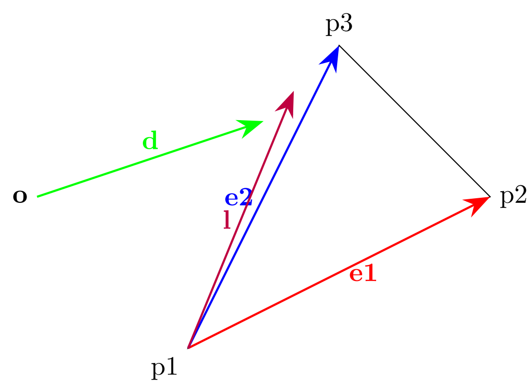
$$\mathbf{e2} = p3 - p1$$




## 1.2 Compute the determinant w

$$\mathbf{l} = \mathbf{d} \times \mathbf{e2}$$

$$\mathbf{w} = \mathbf{e1} \cdot \mathbf{l}$$



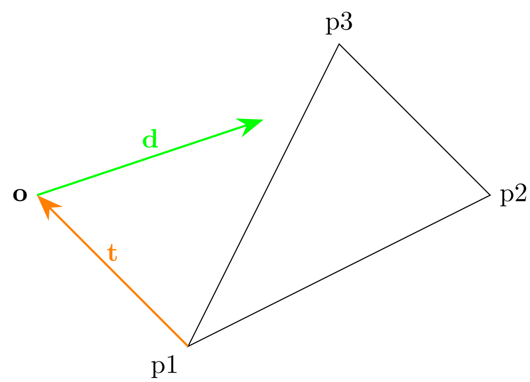

If **w** is near zero, the ray lies in the plane of the triangle.

## 1.3 Compute the inverse determinant i

$$i = \frac{1}{\mathbf{w}}$$

## 1.4 Compute the distance vector

$$\mathbf{t} = \mathbf{o} - p1$$




## 1.5 Compute the $u$ parameter

$$u = (\mathbf{t} \cdot \mathbf{l}) \times i$$

## 1.6 Compute the $v$ parameter

$$\mathbf{q} = \mathbf{t} \times \mathbf{e1}$$

$$v = (\mathbf{d} \cdot \mathbf{q}) \times i$$

## 1.7 Compute the intersection distance

$$t = (\mathbf{e2} \cdot \mathbf{q}) \times i$$

This algorithm efficiently determines if a ray intersects with a triangle by using barycentric coordinates and vector mathematics. It ensures that the intersection point lies within the triangle and not just on the plane of the triangle.