Cross product

From Wikipedia, the free encyclopedia

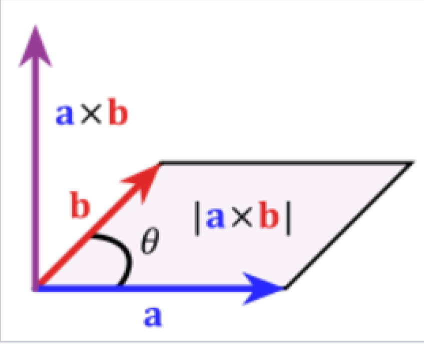
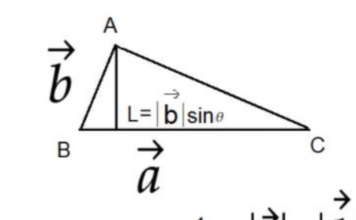


Figure 1. The area of a parallelogram as the magnitude of a cross product

$$\mathbf{a} \times \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \sin(\theta) \, \mathbf{n}$$



$$\text{area of } \triangle = \tfrac{1}{2} \times |\vec{a}| \times |\vec{b}| \times sin\theta$$
$$= \tfrac{1}{2} \times [\vec{a} \times \vec{b}\,]$$

Example 5: Finding the Area of a Triangle Given Its Three Vertices

Find the area of a triangle ABC, where $A(-8, -9)$, $B(-7, -8)$, and $C(9, -2)$.

Answer

The magnitude of the cross product of two vectors is equal to the area of the parallelogram spanned by them. The area of the triangle ABC is equal to half the area of the parallelogram spanned by two vectors defined by its vertices:

$$\text{the area of } ABC = \frac{1}{2}\left\|\overrightarrow{AB} \times \overrightarrow{AC}\right\| = \frac{1}{2}\left\|\overrightarrow{BA} \times \overrightarrow{BC}\right\| = \frac{1}{2}\left\|\overrightarrow{CB} \times \overrightarrow{CA}\right\|.$$

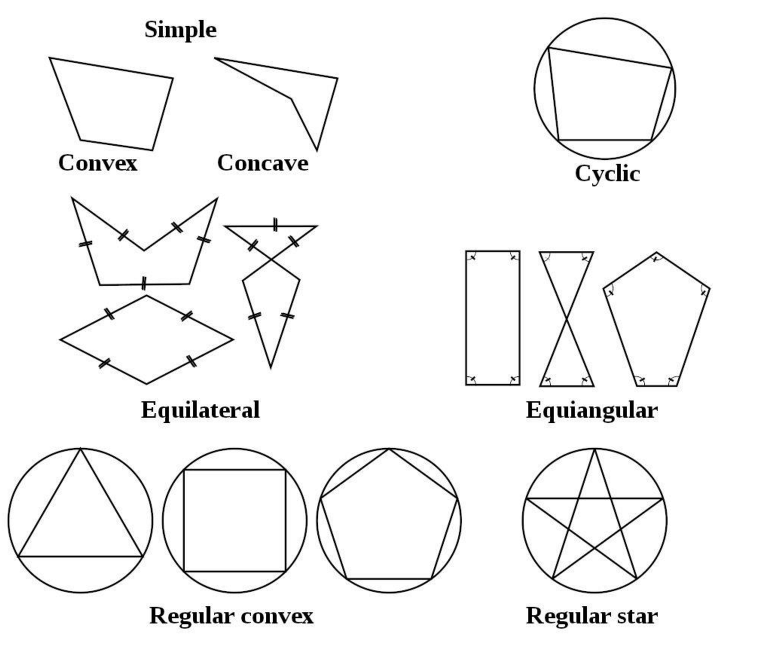
Polygon

From Wikipedia, the free encyclopedia

Convexity and non-convexity

Polygons may be characterized by their convexity or type of non-convexity:

- Convex: any line drawn through the polygon (and not tangent to an edge or corner) meets its boundary exactly twice. As a consequence, all its interior angles are less than 180°. Equivalently, any line segment with endpoints on the boundary passes through only interior points between its endpoints.
- Non-convex: a line may be found which meets its boundary more than twice. Equivalently, there exists a line segment between two boundary points that passes outside the polygon.
- Simple: the boundary of the polygon does not cross itself. All convex polygons are simple.
- Concave: Non-convex and simple. There is at least one interior angle greater than 180°.



Equality and symmetry

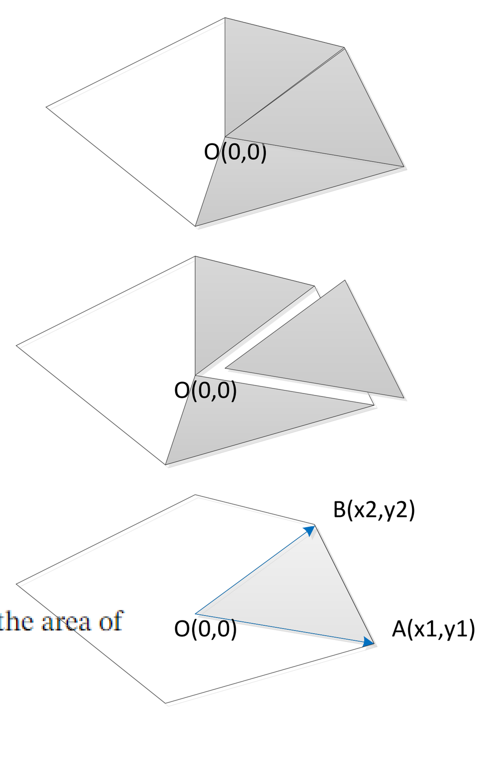
- Regular: the polygon is both *isogonal* and *isotoxal*. Equivalently, it is both *cyclic* and *equilateral*, or both *equilateral* and *equiangular*. A non-convex regular polygon is called a *regular star polygon*.
- Isogonal or vertex-transitive: all corners lie within the same symmetry orbit. The polygon is also cyclic and equiangular.
- Isotoxal or edge-transitive: all sides lie within the same symmetry orbit. The polygon is also equilateral and tangential.
- Cyclic: all corners lie on a single circle, called the circumcircle.
- Equilateral: all edges are of the same length. The polygon need not be convex.
- Equiangular: all corner angles are equal.
- Tangential: all sides are tangent to an inscribed circle.

Area

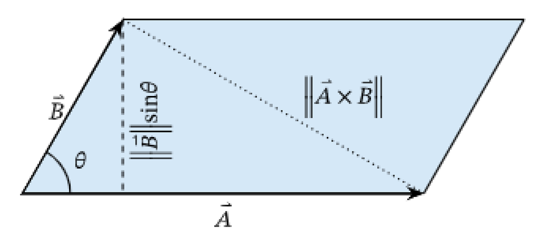
In this section, the vertices of the polygon under consideration are taken to be $(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$ in order. For convenience in some formulas, the notation $(x_n, y_n) = (x_0, y_0)$ will also be used.

If the polygon is non-self-intersecting (that is, simple), the signed area is

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i) \quad \text{where } x_n = x_0 \text{ and } y_n = y_0,$$



- The cross product is *distributive*: $(\vec{A} + \vec{B}) \times \vec{C} = \vec{A} \times \vec{C} + \vec{B} \times \vec{C}$.
- The cross product is *anticommutative*: $\vec{A} \times \vec{B} = -\vec{B} \times \vec{A}$.
- The cross product of two collinear vectors is zero, and so $\vec{A} \times \vec{A} = 0$.
- The area of the parallelogram spanned by $\vec{A}$ and $\vec{B}$ is given by $\left\|\vec{A} \times \vec{B}\right\|$. It follows that the area of the triangle with $\vec{A}$ and $\vec{B}$ defining two of its sides is given by $\frac{1}{2}\left\|\vec{A} \times \vec{B}\right\|$.



Centroid

Using the same convention for vertex coordinates as in the previous section, the coordinates of the centroid of a solid simple polygon are

$$C_x = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),$$

$$C_y = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i).$$

In these formulas, the signed value of area A must be used.

For triangles ($n = 3$), the centroids of the vertices and of the solid shape are the same, but, in general, this is not true for $n > 3$. The centroid of the vertex set of a polygon with n vertices has the coordinates

$$c_x = \frac{1}{n} \sum_{i=0}^{n-1} x_i,$$

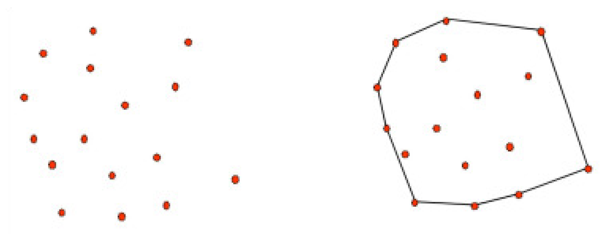
$$c_y = \frac{1}{n} \sum_{i=0}^{n-1} y_i.$$

```
//std::vector<KVector2>   g_vertices;
KVector2 geoCenter = KVector2::zero;
for (auto v : g_vertices)
{
    geoCenter += v;
}
geoCenter.x /= g_vertices.size();
geoCenter.y /= g_vertices.size();
for (auto& v : g_vertices)
{
    v -= geoCenter;
}
```

Convex hull

From Wikipedia, the free encyclopedia

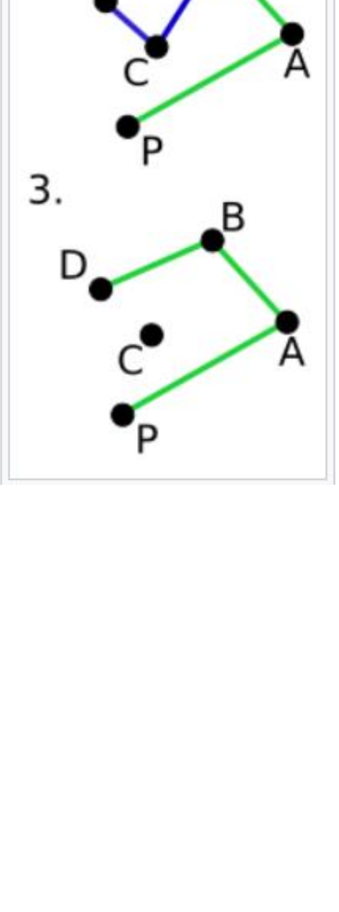
In geometry, the **convex hull** or **convex envelope** or **convex closure** of a shape is the smallest convex set that contains it. The convex hull may be defined either as the intersection of all convex sets containing a given subset of a Euclidean space, or equivalently as the set of all convex combinations of points in the subset. For a bounded subset of the plane, the convex hull may be visualized as the shape enclosed by a rubber band stretched around the subset.



Graham scan

From Wikipedia, the free encyclopedia

Graham's scan is a method of finding the convex hull of a finite set of points in the plane with time complexity $O(n \log n)$. It is named after Ronald Graham, who published the original algorithm in 1972.[1] The algorithm finds all vertices of the convex hull ordered along its boundary. It uses a stack to detect and remove concavities in the boundary efficiently.




```
let points be the list of points
let stack = empty_stack()

find the lowest y-coordinate and leftmost point, called P0
sort points by polar angle with P0, if several points have the same polar angle then only keep the farthest

for point in points:
    # pop the last point from the stack if we turn clockwise to reach this point
    while count stack > 1 and ccw(next_to_top(stack), top(stack), point) <= 0:
        pop stack
    push point to stack
end
```

Second moment of area

https://en.wikipedia.org/wiki/Second_moment_of_area

From Wikipedia, the free encyclopedia
(Redirected from Area moment of inertia)

This article is about the geometrical property of an area, termed the second moment of area. For the moment of inertia dealing with the rotation of an object with mass, see Mass moment of inertia.
For a list of equations for second moments of area of standard shapes, see List of second moments of area.

The **second moment of area**, or **second area moment**, or **quadratic moment of area** and also known as the **area moment of inertia**, is a geometrical property of an area which reflects how its points are distributed with regard to an arbitrary axis. The second moment of area is typically denoted with either an I (for an axis that lies in the plane) or with a J (for an axis perpendicular to the plane). In both cases, it is calculated with a multiple integral over the object in question. Its dimension is L (length) to the fourth power. Its unit of dimension, when working with the International System of Units, is meters to the fourth power, m^4, or inches to the fourth power, in^4, when working in the Imperial System of Units.

Definition [edit]

The second moment of area for an arbitrary shape R with respect to an arbitrary axis BB' is defined as

$$J_{BB'} = \iint_R \rho^2 \,\mathrm{d}A$$

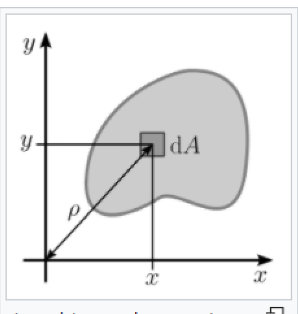
where

$\mathrm{d}A$ is the infinitesimal area element, and
ρ is the perpendicular distance from the axis BB'.[2]

For example, when the desired reference axis is the x-axis, the second moment of area I_{xx} (often denoted as I_x) can be computed in Cartesian coordinates as

$$I_x = \iint_R y^2 \,\mathrm{d}x\,\mathrm{d}y$$

The second moment of the area is crucial in Euler–Bernoulli theory of slender beams.



An arbitrary shape. ρ is the radial distance to the element dA, with projections x and y on the axes.

Parallel axis theorem [edit]

Main article: Parallel axis theorem

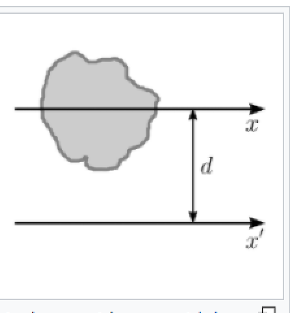
It is sometimes necessary to calculate the second moment of area of a shape with respect to an x' axis different to the centroidal axis of the shape. However, it is often easier to derive the second moment of area with respect to its centroidal axis, x, and use the parallel axis theorem to derive the second moment of area with respect to the x' axis. The parallel axis theorem states

$$I_{x'} = I_x + Ad^2$$

where

A is the area of the shape, and
d is the perpendicular distance between the x and x' axes.[4][5]

A similar statement can be made about a y' axis and the parallel centroidal y axis. Or, in general, any centroidal B axis and a parallel B' axis.



A shape with centroidal axis x. The parallel axis theorem can be used to obtain the second moment of area with respect to the x' axis.

Composite shapes [edit]

For more complex areas, it is often easier to divide the area into a series of "simpler" shapes. The second moment of area for the entire shape is the sum of the second moment of areas of all of its parts about a common axis. This can include shapes that are "missing" (i.e. holes, hollow shapes, etc.), in which case the second moment of area of the "missing" areas are subtracted, rather than added. In other words, the second moment of area of "missing" parts are considered negative for the method of composite shapes.

Rectangle with centroid at the origin [edit]

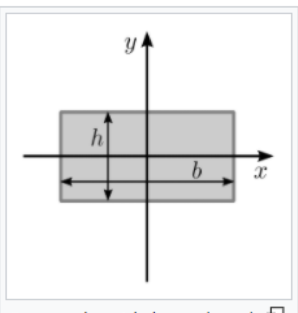
Consider a rectangle with base b and height h whose centroid is located at the origin. I_x represents the second moment of area with respect to the x-axis; I_y represents the second moment of area with respect to the y-axis; J_z represents the polar moment of inertia with respect to the z-axis.

$$I_x = \iint_R y^2 \,\mathrm{d}A = \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_{-\frac{h}{2}}^{\frac{h}{2}} y^2 \,\mathrm{d}y\,\mathrm{d}x = \int_{-\frac{b}{2}}^{\frac{b}{2}} \frac{1}{3}\frac{h^3}{4} \,\mathrm{d}x = \frac{bh^3}{12}$$

$$I_y = \iint_R x^2 \,\mathrm{d}A = \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_{-\frac{h}{2}}^{\frac{h}{2}} x^2 \,\mathrm{d}y\,\mathrm{d}x = \int_{-\frac{b}{2}}^{\frac{b}{2}} hx^2 \,\mathrm{d}x = \frac{b^3h}{12}$$

Using the perpendicular axis theorem we get the value of J_z.

$$J_z = I_x + I_y = \frac{bh^3}{12} + \frac{hb^3}{12} = \frac{bh}{12}\left(b^2 + h^2\right)$$



Rectangle with base b and

Any polygon [edit]

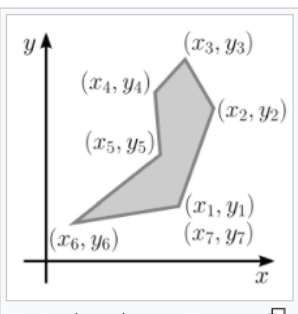
The second moment of area about the origin for any simple polygon on the XY-plane can be computed in general by summing contributions from each segment of the polygon after dividing the area into a set of triangles. This formula is related to the shoelace formula and can be considered a special case of Green's theorem.

A polygon is assumed to have n vertices, numbered in counter-clockwise fashion. If polygon vertices are numbered clockwise, returned values will be negative, but absolute values will be correct.

$$I_y = \frac{1}{12}\sum_{i=1}^{n}\left(x_i y_{i+1} - x_{i+1} y_i\right)\left(x_i^2 + x_i x_{i+1} + x_{i+1}^2\right)$$

$$I_x = \frac{1}{12}\sum_{i=1}^{n}\left(x_i y_{i+1} - x_{i+1} y_i\right)\left(y_i^2 + y_i y_{i+1} + y_{i+1}^2\right)$$

$$I_{xy} = \frac{1}{24}\sum_{i=1}^{n}\left(x_i y_{i+1} - x_{i+1} y_i\right)\left(x_i y_{i+1} + 2x_i y_i + 2x_{i+1} y_{i+1} + x_{i+1} y_i\right)$$



A simple polygon. Here, $n = 6$, notice point "7" is identical to point 1.

File Edit Selection View Go Run Terminal Help ComputeMass.cpp - Visual Studio Code [Administrator]

ComputeMass.cpp × Settings Extension: C/C++

C: > Users > PC > Downloads > ComputeMass.cpp > b2PolygonShape::ComputeMass(b2MassData *, float32) const

```cpp
void b2PolygonShape::ComputeMass(b2MassData* massData, float32 density) const
{
    // Polygon mass, centroid, and inertia.
    // Let rho be the polygon density in mass per unit area.
    // Then:
    // mass = rho * int(dA)
    // centroid.x = (1/mass) * rho * int(x * dA)
    // centroid.y = (1/mass) * rho * int(y * dA)
    // I = rho * int((x*x + y*y) * dA)
    //
    // We can compute these integrals by summing all the integrals
    // for each triangle of the polygon. To evaluate the integral
    // for a single triangle, we make a change of variables to
    // the (u,v) coordinates of the triangle:
    // x = x0 + e1x * u + e2x * v
    // y = y0 + e1y * u + e2y * v
    // where 0 <= u && 0 <= v && u + v <= 1.
    //
    // We integrate u from [0,1-v] and then v from [0,1].
    // We also need to use the Jacobian of the transformation:
    // D = cross(e1, e2)
    //
    // Simplification: triangle centroid = (1/3) * (p1 + p2 + p3)
    //
    // The rest of the derivation is handled by computer algebra.

    b2Assert(m_count >= 3);

    b2Vec2 center; center.Set(0.0f, 0.0f);
    float32 area = 0.0f;
    float32 I = 0.0f;

    // s is the reference point for forming triangles.
    // It's location doesn't change the result (except for rounding error).
    b2Vec2 s(0.0f, 0.0f);

    // This code would put the reference point inside the polygon.
    for (int32 i = 0; i < m_count; ++i)
    {
        s += m_vertices[i];
    }
    s *= 1.0f / m_count;

    const float32 k_inv3 = 1.0f / 3.0f;

    for (int32 i = 0; i < m_count; ++i)
    {
        // Triangle vertices.
        b2Vec2 e1 = m_vertices[i] - s;
        b2Vec2 e2 = i + 1 < m_count ? m_vertices[i+1] - s : m_vertices[0] - s;

        float32 D = b2Cross(e1, e2);

        float32 triangleArea = 0.5f * D;
        area += triangleArea;

        // Area weighted centroid
        center += triangleArea * k_inv3 * (e1 + e2);

        float32 ex1 = e1.x, ey1 = e1.y;
        float32 ex2 = e2.x, ey2 = e2.y;

        float32 intx2 = ex1*ex1 + ex2*ex1 + ex2*ex2;
        float32 inty2 = ey1*ey1 + ey2*ey1 + ey2*ey2;

        I += (0.25f * k_inv3 * D) * (intx2 + inty2);
    }

    // Total mass
    massData->mass = density * area;

    // Center of mass
    b2Assert(area > b2_epsilon);
    center *= 1.0f / area;
    massData->center = center + s;

    // Inertia tensor relative to the local origin (point s).
    massData->I = density * I;

    // Shift to center of mass then to original body origin.
    massData->I += massData->mass * (b2Dot(massData->center, massData->center) - b2
}
```

⊗ 0 ⚠ 0 Ln 1, Col 1 Tab Size: 4 UTF-8 CRLF C++ Win32

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) \quad \text{where } x_n = x_0 \text{ and } y_n = y_0,$$

Centroid

Using the same convention for vertex coordinates as in the previous section, the coordinates of the centroid of a solid simple polygon are

$$C_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),$$

$$C_y = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i).$$

Any polygon [edit]

The second moment of area about the origin for any simple polygon on the XY-plane can be computed in general by summing contributions from each segment of the polygon after dividing the area into a set of triangles. This formula is related to the shoelace formula and can be considered a special case of Green's theorem.

A polygon is assumed to have n vertices, numbered in counter-clockwise fashion. If polygon vertices are numbered clockwise, returned values will be negative, but absolute values will be correct.

$$I_y = \frac{1}{12}\sum_{i=1}^{n}\left(x_i y_{i+1} - x_{i+1} y_i\right)\left(x_i^2 + x_i x_{i+1} + x_{i+1}^2\right)$$

$$I_x = \frac{1}{12}\sum_{i=1}^{n}\left(x_i y_{i+1} - x_{i+1} y_i\right)\left(y_i^2 + y_i y_{i+1} + y_{i+1}^2\right)$$

$$I_{xy} = \frac{1}{24}\sum_{i=1}^{n}\left(x_i y_{i+1} - x_{i+1} y_i\right)\left(x_i y_{i+1} + 2x_i y_i + 2x_{i+1} y_{i+1} + x_{i+1} y_i\right)$$