STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, Month 2024, pp 0–18.
Published online in International Academic Press (www.IAPress.org)

Spotting, Tracking algorithm and the remoteness: The-Point-in-Polygon-Problem (P.I.P.)

Aziz Arbai [1, *], Abounaima Mohammed Chaouki [2], Amina Bellekbir [3]

[1] *ENSIT - Ecole des nouvelles sciences d'ingénierie - Le Laboratoire Systèmes, Contrôle et décision (LSCD), Tanger, Maroc*
[2] *Faculty of Sciences and Technology, Sidi Mohammed Ben Abdellah University, Fes, Maroc*
[3] *École Nationale des Sciences Appliquées de Tanger, Abdelmalek Essaadi University, Tetouan, Maroc*

Abstract On this paper we present a solution to detect and know if a point M is inside a polygon $(A_k)_{k\in\{1,...,n\}}$ or outside. In the case where the point M is at outside the polygon, a simple optimization method will be applied to determine the distance between the point M and the polygon $A_1, ..., A_n$, that is to say between the point M and the point P of the border of the polygon closest to M: The Neighboring Point. The introduction of complex numbers in algebra and geometry is very useful. We therefore proceed with our purely geometric methods as well as with complex numbers, using the triangulation of a convex polygon and a very specific complex application that we have just constructed. So our contributions are: A very simple and fast tracking algorithm, and therefore an extremely low cost per iteration (An Algorithm for Judging Points Inside or Outside a Polygon), The neighboring Point and The spacing in the case where the point M is at outside the polygon, so without going through the quadratic optimization, we will give the simplest and fastest way to calculate the distance d between the point M and the polygon $(A_m)_{m\in\{1,...,n\}}$ and determine the point P of the boundary of the closest (neighboring) polygon to M . Above all it is an article full of examples for all the cases and methods and which are ready to be programmed and applied.

AMS 2010 subject classifications 65D99, 65F05

DOI: 10.19139/soic-2310-5070-1893

1. Introduction

The question of whether a point lies inside a polygon is a fundamental problem in computational geometry and has applications in various fields, including computer graphics, geographic information systems (GIS), and robotics. Here are some relevant works and algorithms related to this topic:

Ray Casting Algorithm: The Ray Crossing (ray intersection) Casting algorithm is One of the primary methods to solve the PIP problem (the Point-in-Polygon Problem) and a widely used method to determine whether a point is inside a polygon. This involves casting a ray from the test point and counting the number of times it intersects the edges of the polygon. If the count is odd, the point is inside the polygon; if it is even, the point is outside. The point of the ray that hits the polygon is called the "witness point", it was derived by Nordbeck and Rystedt in 1967 [9], [10], [12]. There are many works that have designed algorithms for the point in a polygon problem based on the ray casting algorithm like in [8], [11] using equations from vector geometry or other modifications and approximations.

*Correspondence to: A. Arbai (Email: arbaiaziz555@gmail.com) ENSIT, LSCD, Tanger

ISSN 2310-5070 (online) ISSN 2311-004X (print)

There is also a method to test if a point is inside a polygon by summing the oriented angles relative to that point. If it's zero, we're outside the polygon. When a point is inside a polygon, the sum of the angles between this point and the points constituting the polygon is equal to 360°.

The introduction of complex numbers in algebra and geometry is very useful see [1], [2], [3], [5].... That is, if a polygon has for summit $(A_k)_{k\in\{1,\dots,n\}}$ of affixes $(z_k)_{k\in\{1,\dots,n\}}$ and the point concerned is M with affix z then,

$$M \text{ is inside the polygon} \Longleftrightarrow \sum_{k=1}^{n} \left| \arg\left(\frac{z_{k+1}-z}{z_k - z} \right) \right| = 2\pi$$

with $z_{n+1} = z_1$.

Another similar method using the areas (or surfaces) of the triangles formed by this point and two successive extremities ordered in either a clockwise direction or viceversa, and then we compare the signs of the areas or just compare the sum of the surfaces (absolute values of the areas) with the total surface of the polygon.

Also, we can do a complex analysis:

Let z_0 be the affix of the point to be tested. We integrate $\frac{1}{2\pi i(z-z_0)}$ along the curve crossing the polygon. If the point is inside, the integral is 1 or -1 (depending on the chosen orientation), otherwise it is 0.

To see an overview of previous algorithms we just have to see the [7] paper

In reality, the problem is a bit more complicated.

The reason is that all computer calculations are performed with a finite number of significant digits. In this field, we cannot test the equality of two real numbers, because a real number does not exist in computer science. We can only test the equality between two floating points, or at least compare the difference of two floating points to a given small epsilon... Without forgetting the fact that it is necessary to avoid the cases where the intersections are vertices of the polygons which one cannot control.

In all cases, calculations of intersections, sum of areas or angles (arguments), and complex integrals (along the curve crossing the polygon) must be stopped at all costs.

We, in this paper, we will proceed by our purely geometric method, using the triangulation of a convex polygon and a very specific application that we have just constructed.

The most appropriate method would be that of traingulization followed by a search for the point on all the triangles (one by one) until falling on the triangle which contains the said point otherwise it is outside the polygon. Except for us, we are going to use (and this is our first contribution) another original method which requires less calculation to be carried out.

And we will finish at the end with another contribution that of finding the Closest Point on a Polygon using a simple optimization method

The triangulation of a convex polygon is trivial and is calculated in linear time, for example starting from a vertex and adding edges to all the other vertices. In 1991, *Bernard Chazelle* [6] showed that any simple polygon can be triangulated in linear time. The proposed algorithm is however very complex, and simpler algorithms are still being researched.

For this we will need a convex polygon (*convex polyhedron*) of vertices $(A_k)_{k\in\{1,\dots,n\}}$ ordered monotonically (increasing or decreasing) clockwise.

Before going any further, we will continue in our introduction indicating the classic and intuitive methods and calculations to solve the Point-in-Polygon Problem (*P.I.P.*): The Order of points in a polygon, Spotting & Tracking a point, The Closest Point on a Polygon and The Quadratic optimization.

1.1. Order of points in a polygon: The scheduling

Let $(A_m)_{1\leq m\leq n}$ be a polygon with affix $(z_m)_{1\leq m\leq n}$.

Problem 1. *How we can order the points* $\{A_m, 1\leq m\leq n\}$ *in* $\left\{A_{(m)}, 1\leq m\leq n\right\}$ *or simply in* $\{B_m, 1\leq m\leq n\}$ *of a monotonously (increasing or decreasing) clockwise?*

Let be $z_m = x_m + iy_m$ with $(x_m, y_m)\in IR^2$.

We therefore build $\{B_m, 1\leq m\leq n\}$ from the following algorithm:
k=1 : $z_{(1)} = z_j$ (& $B_1 = A_j$) such that $x_j = Min\{x_m/1\leq m\leq n\}$;
k=2 : $z_{(2)} = z_j$ such that
$\left|z_{(1)} - z_j\right| = Min\left\{\left|z_{(1)} - z_m\right| \neq 0 \ / \ 1\leq m\leq n \ \& \ y_m\leq y_{(1)}\right\}$
& $B_2 = A_j$ such that
$d(B_1, A_j) = Min\left\{d(B_1, A_m)\neq 0 \ / \ 1\leq m\leq n \ \& \ y_m\leq y_{(1)}\right\}$;
.
.
k : $(3\leq k\leq n)$ $z_{(k)} = z_j$ (& $B_k = A_j$) such that $\gamma_k = \beta_{(k)} = \beta_j = \arg\left(\frac{z_j - z_{(1)}}{z_{(2)} - z_{(1)}}\right)$.
.
.
n: ...
$\beta_m = \widehat{B_2 B_1 A_m} = \arg\left(\frac{z_m - z_{(1)}}{z_{(2)} - z_{(1)}}\right)$
and $\gamma_m = \beta_{(m)}$ (by ordering the β_m $\{A_i, 1\leq i\leq n\}$ according to the trigonometric-sense).

Remark 1. $z_{(1)}$ *(so and* $B_1(z_{(1)})$*) is arbitrary. We can take any* $z_{(1)}$.

B_m is the point with affix $z_{(m)}$.

1.2. Spotting & Tracking a point:

Problem 2. *How to know if the point* $M(z)$ *is inside or is outside the polygon* $(A_m)_{1\leq m\leq n}$ *with affix* $(z_m)_{1\leq m\leq n}$*?*

we will study case by case to finally be able to see the robustness of the algorithm deduced by too many calculations and formulas, especially when n is very large.

1.2.1. Case for n=3

Let be $z = x + iy$, $z_m = x_m + iy_m$ $\forall m\in\{1,2,3\}$ and (A_1, A_2, A_3) a triangle such $A_m(z_m)$ $\forall m\in\{1,2,3\}$ and $M(z)$.
We know that:
M is inside the triangle (A_1, A_2, A_3) if and only if
$\exists\,(\alpha_1, \alpha_2, \alpha_3)\in[0,1]^3$ such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$ & $\alpha_1\overrightarrow{MA}_1 + \alpha_2\overrightarrow{MA}_2 + \alpha_3\overrightarrow{MA}_3 = \overrightarrow{0}$.
$\Leftrightarrow \alpha_1(z_1 - z) + \alpha_2(z_2 - z) + \alpha_3(z_3 - z) = 0$
$\Leftrightarrow$

$$\begin{array}{rcl} z & = & \alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 \\ \& \ (\alpha_1, \alpha_2, \alpha_3) & \in & [0,1]^3 \ / \ \alpha_1 + \alpha_2 + \alpha_3 = 1 \end{array}$$

$\Leftrightarrow z = \alpha_1 z_1 + \alpha_2 z_2 + (1 - \alpha_1 - \alpha_2) z_3$ & $(\alpha_1, \alpha_2,) \in [0,1]^2$
$\Leftrightarrow$

$$\begin{aligned} z - z_3 &= \alpha_1 (z_1 - z_3) + \alpha_2 (z_2 - z_3) \\ \& \ (\alpha_1, \alpha_2,) &\in [0,1]^2 \end{aligned}$$

- Let be $\begin{cases} Z_1 = z_1 - z_3 = |Z_1| e^{i\theta_1} = X_1 + iY_1 \\ Z_2 = z_2 - z_3 = |Z_2| e^{i\theta_2} = X_2 + iY_2 \\ Z = z - z_3 = |Z| e^{i\theta} = X + iY \end{cases}$

$\Longrightarrow Z = \alpha_1 Z_1 + \alpha_2 Z_2$
$\Longrightarrow \begin{cases} X = \alpha_1 X_1 + \alpha_2 X_2 \\ Y = \alpha_1 Y_1 + \alpha_2 Y_2 \end{cases}$
$\Longrightarrow \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} X_1 & X_2 \\ Y_1 & Y_2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}$
$\Longrightarrow \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \frac{1}{X_1 Y_2 - X_2 Y_1} \begin{pmatrix} Y_2 & -X_2 \\ -Y_1 & X_1 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$
$\Longrightarrow \begin{cases} \alpha_1 = \frac{XY_2 - X_2 Y}{X_1 Y_2 - X_2 Y_1} \\ \alpha_2 = \frac{X_1 Y - X Y_1}{X_1 Y_2 - X_2 Y_1} \end{cases}$
$\Longrightarrow \begin{cases} \alpha_1 = \frac{|Z| \sin(\theta_2 - \theta)}{|Z_1| \sin(\theta_2 - \theta_1)} \\ \alpha_2 = \frac{|Z| \sin(\theta - \theta_1)}{|Z_2| \sin(\theta_2 - \theta_1)} \end{cases}$
since
$XY_2 - X_2 Y = |Z| \cos\theta \, |Z_2| \sin\theta_2 - |Z_2| \cos\theta_2 \, |Z| \sin\theta$
$XY_2 - X_2 Y = |Z| \, |Z_2| \sin(\theta_2 - \theta)$
$X_1 Y - X Y_1 = |Z| \, |Z_1| \sin(\theta - \theta_1)$
$X_1 Y_2 - X_2 Y_1 = |Z_1| \, |Z_2| \sin(\theta_2 - \theta_1)$
$(\alpha_1, \alpha_2) \in [0,1]^2 \Longleftrightarrow \begin{cases} 0 \leq \frac{\sin(\theta_2 - \theta)}{\sin(\theta_2 - \theta_1)} \leq \frac{|Z_1|}{|Z|} \\ 0 \leq \frac{\sin(\theta_1 - \theta)}{\sin(\theta_1 - \theta_2)} \leq \frac{|Z_2|}{|Z|} \end{cases}$
with
$\theta_2 - \theta = \arg\left[\frac{z_2 - z_3}{z - z_3}\right]$
$\theta_1 - \theta = \arg\left[\frac{z_1 - z_3}{z - z_3}\right]$
$\theta_2 - \theta_1 = \arg\left[\frac{z_2 - z_3}{z_1 - z_3}\right]$
$|Z_1| = |z_1 - z_3|$
$|Z_2| = |z_2 - z_3|$
$|Z| = |z - z_3|$

- Or in another way

$\Leftrightarrow \exists (\alpha_1, \alpha_2, \alpha_3) \in [0,1]^3 \ /$
$\begin{cases} \alpha_1 (x - x_1) + \alpha_2 (x - x_2) + \alpha_3 (x - x_3) = 0 \\ \alpha_1 (y - y_1) + \alpha_2 (y - y_2) + \alpha_3 (y - y_3) = 0 \\ \alpha_1 + \alpha_2 + \alpha_3 = 1 \end{cases}$
$\Leftrightarrow \exists (\alpha_1, \alpha_2, \alpha_3) \in [0,1]^3 \ / \begin{cases} (\alpha_1 + \alpha_2 + \alpha_3) x - (\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3) = 0 \\ (\alpha_1 + \alpha_2 + \alpha_3) y - (\alpha_1 y_1 + \alpha_2 y_2 + \alpha_3 y_3) = 0 \\ \alpha_1 + \alpha_2 + \alpha_3 = 1 \end{cases}$
$\Leftrightarrow \exists (\alpha_1, \alpha_2, \alpha_3) \in [0,1]^3 \ / \begin{cases} \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 = x \\ \alpha_1 y_1 + \alpha_2 y_2 + \alpha_3 y_3 = y \\ \alpha_1 + \alpha_2 + \alpha_3 = 1 \end{cases}$

$$\Leftrightarrow \exists \alpha \in [0,1]^3 \ / \ \Phi.\alpha = b$$

with $\alpha = (\alpha_1, \alpha_2, \alpha_3)'$ $b = (x, y, 1)'$ and $\Phi = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}$

$\det(\Phi) = \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}$

$\Rightarrow \det(\Phi) = (x_1y_2 + x_2y_3 + x_3y_1) - (x_1y_3 + x_2y_1 + x_3y_2)$

$\Rightarrow \det(\Phi) = (x_1 - x_2)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_2)$

$\Rightarrow \det(\Phi) = \begin{vmatrix} x_1 - x_2 & x_2 - x_3 \\ y_1 - y_2 & y_2 - y_3 \end{vmatrix}$

$$\det(\Phi) = 0 \Longleftrightarrow (A_2A_1) \shortparallel (A_3A_2) \Longleftrightarrow (A_1, A_2, A_3) \text{ are aligned}$$

(A_1, A_2, A_3) a regular triangle $\Rightarrow \det(\Phi) \neq 0$

$\Rightarrow \Phi$ is invertible

$\Longrightarrow$

$$\alpha = \Phi^{-1}.b$$

with $\Phi^{-1} = \frac{1}{\det(\Phi)} [Adj(\Phi)]^t$

$$Adj(\Phi) = \begin{pmatrix} \begin{vmatrix} y_2 & y_3 \\ 1 & 1 \end{vmatrix} & -\begin{vmatrix} y_1 & y_3 \\ 1 & 1 \end{vmatrix} & \begin{vmatrix} y_1 & y_2 \\ 1 & 1 \end{vmatrix} \\ -\begin{vmatrix} x_2 & x_3 \\ 1 & 1 \end{vmatrix} & \begin{vmatrix} x_1 & x_3 \\ 1 & 1 \end{vmatrix} & -\begin{vmatrix} x_1 & x_2 \\ 1 & 1 \end{vmatrix} \\ \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} & -\begin{vmatrix} x_1 & x_3 \\ y_1 & y_3 \end{vmatrix} & \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} \end{pmatrix}$$

$\Rightarrow$

$$\Phi^{-1} = \frac{1}{|\Phi|} \begin{pmatrix} y_2 - y_3 & x_3 - x_2 & x_2y_3 - x_3y_2 \\ y_3 - y_1 & x_1 - x_3 & x_3y_1 - x_1y_3 \\ y_1 - y_2 & x_2 - x_1 & x_1y_2 - x_2y_1 \end{pmatrix}$$

$|\Phi| = (x_1 - x_2)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_2)$

Conclusion:

$\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in [0,1]^3 \Leftrightarrow M$ is inside the triangle (A_1, A_2, A_3)

$\Updownarrow$

$$\frac{1}{|\Phi|} \begin{pmatrix} y_2 - y_3 & x_3 - x_2 & x_2y_3 - x_3y_2 \\ y_3 - y_1 & x_1 - x_3 & x_3y_1 - x_1y_3 \\ y_1 - y_2 & x_2 - x_1 & x_1y_2 - x_2y_1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in [0,1]^3$$

with $|\Phi| = (x_1 - x_2)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_2)$

$\Updownarrow$

$$\begin{cases} 0 \leq \frac{x(y_2-y_3)+y(x_3-x_2)+(x_2y_3-x_3y_2)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_3-y_1)+y(x_1-x_3)+(x_3y_1-x_1y_3)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_1-y_2)+y(x_2-x_1)+(x_1y_2-x_2y_1)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \end{cases}$$

1.2.2. Example: Case of the triangle (OIJ):

Let $(A_1A_2A_3)$ be the triangle such that $A_1 = O\,(0)$, $A_2 = I\,(1)$ & $A_3 = J\,(i)$

so

$x_1 = 0$

$y_1 = 0$

$x_2 = 1$

$y_2 = 0$

$x_3 = 0$

$y_3 = 1$

$M\,(x + iy)$ is inside $(A_1A_2A_3)$ if and only if

$$\begin{cases} 0 \leq \frac{x(y_2-y_3)+y(x_3-x_2)+(x_2y_3-x_3y_2)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_3-y_1)+y(x_1-x_3)+(x_3y_1-x_1y_3)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_1-y_2)+y(x_2-x_1)+(x_1y_2-x_2y_1)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \end{cases}$$

$\Updownarrow$

$$\begin{cases} 0 \leq -x - y + 1 \leq 1 \\ 0 \leq x \leq 1 \\ 0 \leq y \leq 1 \end{cases}$$

$\Longleftrightarrow$

$$\begin{cases} 0 \leq x + y \leq 1 \\ 0 \leq x \leq 1 \\ 0 \leq y \leq 1 \end{cases}$$

So, $M\,(x + iy)$ is inside the triangle (OIJ) if and only if

$$\begin{cases} x + y \leq 1 \\ 0 \leq x \\ 0 \leq y \end{cases}$$

indeed it is the interior of the triangle (OIJ)

1.2.3. The general case ($n \geq 4$)

Let be $M = (x, y)$ and $(A_k)_{1\leq k\leq n}$ a polygon such that $\{A_k, 1 \leq k \leq n\}$ is monotonically ordered according to the trigonometric-sense.

The known method consists of partitioning the polygon $\{A_k, 1 \leq k \leq n\}$ into a union of disjoint triangles Δ_k such that

$\Delta_k = (A_1A_kA_{k+1})\ \forall k \in \{2, 3, ..., n-1\}$, what it means

M is inside the polygon $\{A_k, 1 \leq k \leq n\} \iff$

$$\exists k \in \{2, 3, ..., n-1\} \text{ such that } \begin{pmatrix} x_1 & x_k & x_{k+1} \\ y_1 & y_k & y_{k+1} \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in [0, 1]^3$$

$\Updownarrow$

$\exists k \in \{2, 3, ..., n-1\}$ such that

$$\begin{cases} 0 \leq \frac{x(y_k-y_{k+1})+y(x_{k+1}-x_k)+(x_ky_{k+1}-x_{k+1}y_k)}{(x_1-x_k)(y_k-y_{k+1})-(x_k-x_{k+1})(y_1-y_k)} \leq 1 \\ 0 \leq \frac{x(y_{k+1}-y_1)+y(x_1-x_{k+1})+(x_{k+1}y_1-x_1y_{k+1})}{(x_1-x_k)(y_k-y_{k+1})-(x_k-x_{k+1})(y_1-y_k)} \leq 1 \\ 0 \leq \frac{x(y_1-y_k)+y(x_k-x_1)+(x_1y_k-x_ky_1)}{(x_1-x_k)(y_k-y_{k+1})-(x_k-x_{k+1})(y_1-y_k)} \leq 1 \end{cases}$$

which is the intersection of three strips (*Ribbons*).

If not M is outside of the polygon $(A_i)_{1\leq i\leq n}$.

1.2.4. Case for n=4

Let A_1, A_2, A_3, A_4 be a quadruplet and $M \begin{pmatrix} x \\ y \end{pmatrix}$.

problem status :
How to know if M is inside the quadruplet A_1, A_2, A_3, A_4?

The solution:
Order the points A_1, A_2, A_3, A_4 according to the method cited in the second section;
We divide the quadruplet $A_{(1)}, A_{(2)}, A_{(3)}, A_{(4)}$ into two triangles $\left(A_{(1)}A_{(2)}A_{(3)}\right)$ and $\left(A_{(1)}A_{(3)}A_{(4)}\right)$, and
We apply the method for $n = 3$ to these two triangles or the last section The general case for $n = 3$.

Conclusions: Assuming that A_1, A_2, A_3, A_4 are ordered according to The scheduling method seen previously, then

$$\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in [0,1]^3$$

or $\begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in [0,1]^3$

then M is inside the quadruplet A_1, A_2, A_3, A_4

$\Updownarrow$

$$\begin{cases} 0 \leq \frac{x(y_2-y_3)+y(x_3-x_2)+(x_2y_3-x_3y_2)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_3-y_1)+y(x_1-x_3)+(x_3y_1-x_1y_3)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_1-y_2)+y(x_2-x_1)+(x_1y_2-x_2y_1)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \end{cases}$$

or

$$\begin{cases} 0 \leq \frac{x(y_2-y_4)+y(x_4-x_2)+(x_2y_4-x_4y_2)}{(x_1-x_2)(y_2-y_4)-(x_2-x_4)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_4-y_1)+y(x_1-x_4)+(x_4y_1-x_1y_4)}{(x_1-x_2)(y_2-y_4)-(x_2-x_4)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_1-y_2)+y(x_2-x_1)+(x_1y_2-x_2y_1)}{(x_1-x_2)(y_2-y_4)-(x_2-x_4)(y_1-y_2)} \leq 1 \end{cases}$$

And, if

$$\begin{cases} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \notin [0,1]^3 \\ \text{and} \\ \begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \notin [0,1]^3 \end{cases}$$

then M is outside the quadruplet A_1, A_2, A_3, A_4.

Example $\mathbf{n = 4}$:
$A_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $A_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $A_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $M = \begin{pmatrix} x \\ y \end{pmatrix}$
How to know if M is inside the quadruplet A_1, A_2, A_3, A_4?
We divide the quadruplet $A_{(1)}, A_{(2)}, A_{(3)}, A_{(4)}$ into two triangles $\left(A_{(1)}A_{(2)}A_{(3)}\right)$ and $\left(A_{(1)}A_{(3)}A_{(4)}\right)$, and
We apply the method for $n = 3$ to these two triangles.
A_1, A_2, A_3, A_4 are ordered according to The scheduling method, then (using the part "Case for $n = 4$")

we have
$x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1, y_1 = 0, y_2 = 1, y_3 = 1, y_4 = 0$
and,

$$\begin{cases} 0 \leq \frac{x(y_2-y_3)+y(x_3-x_2)+(x_2y_3-x_3y_2)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_3-y_1)+y(x_1-x_3)+(x_3y_1-x_1y_3)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \\ 0 \leq \frac{x(y_1-y_2)+y(x_2-x_1)+(x_1y_2-x_2y_1)}{(x_1-x_2)(y_2-y_3)-(x_2-x_3)(y_1-y_2)} \leq 1 \end{cases}$$

$$\text{or} \begin{cases} 0 \leq \frac{x(y_3-y_4)+y(x_4-x_3)+(x_3y_4-x_4y_3)}{(x_1-x_3)(y_3-y_4)-(x_3-x_4)(y_1-y_3)} \leq 1 \\ 0 \leq \frac{x(y_4-y_1)+y(x_1-x_4)+(x_4y_1-x_1y_4)}{(x_1-x_3)(y_3-y_4)-(x_3-x_4)(y_1-y_3)} \leq 1 \\ 0 \leq \frac{x(y_1-y_3)+y(x_3-x_1)+(x_1y_3-x_3y_1)}{(x_1-x_3)(y_3-y_4)-(x_3-x_4)(y_1-y_3)} \leq 1 \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} 0 \leq \frac{-y}{-1} \leq 1 \\ 0 \leq \frac{x-y}{-1} \leq 1 \\ 0 \leq \frac{-x}{-1} \leq 1 \end{cases} \text{ or } \begin{cases} 0 \leq \frac{x-1}{-1} \leq 1 \\ 0 \leq \frac{-y}{-1} \leq 1 \\ 0 \leq \frac{-x+y}{-1} \leq 1 \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} 0 \leq y \leq 1 \\ 0 \leq y - x \leq 1 \\ 0 \leq x \leq 1 \end{cases} \text{ or } \begin{cases} 0 \leq x - y \leq 1 \\ 0 \leq y \leq 1 \\ 0 \leq x \leq 1 \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} 0 \leq x \leq 1 \ \& \ 0 \leq y \leq 1 \\ x - 1 \leq y \leq x \text{ or } x \leq y \leq 1 + x \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} 0 \leq x \leq 1 \ \& \ 0 \leq y \leq 1 \\ x - 1 \leq y \leq 1 + x \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} 0 \leq x \leq 1 \ \& \ 0 \leq y \leq 1 \\ |y - x| \leq 1 \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} 0 \leq x \leq 1 \\ 0 \leq y \leq 1 \end{cases} \text{ (obviously)}$$

$$\iff M \begin{pmatrix} x \\ y \end{pmatrix} \text{ is inside the quadruplet } A_1 \begin{pmatrix} 0 \\ 0 \end{pmatrix}, A_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}, A_3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}, A_4 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

1.3. Closest Point on a Polygon:

In the case where the point M is outside a polygon, the closest point of this polygon is the point on the perimeter (within the boundaries of the polygon) that is closest to a given reference point (often called a target point or query point). This concept is often used in computational geometry and computer graphics for various purposes, such as collision detection, path finding, or spatial analysis.

Problem 3. *In the case where the point $M_0 \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ is outside the polygon $A_1, ..., A_n$, it would be interesting to know how far the point M_0 moves away from the polygon $A_1, ..., A_n$ and to know where is the closest point in the polygon?*

1.3.1. Quadratic optimization:

Each polygon $(A_m)_{1 \leq m \leq n}$ is a *convex polyhedron* of the form

$$\Phi \cdot \begin{pmatrix} x \\ y \end{pmatrix} \leq c \in IR^p$$

with Φ a matrix of dimension $p \times 2$.
The distance between the point M_0 and the *polyhedron* is the minimum of
$d\left(M_0\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, M\begin{pmatrix} x \\ y \end{pmatrix}\right)$
with $M\begin{pmatrix} x \\ y \end{pmatrix}$ is a point of the polygon (*polyhedron*), who means such $\Phi \cdot \begin{pmatrix} x \\ y \end{pmatrix} \leq c$.
But $d\left(M_0\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, M\begin{pmatrix} x \\ y \end{pmatrix}\right) = \sqrt{(x-x_0)^2 + (y-y_0)^2}$
it is therefore necessary to minimize

$$\begin{aligned} d(M_0, M) &= (x-x_0)^2 + (y-y_0)^2 \\ &= x^2 + y^2 - 2x_0x - 2y_0y + x_0^2 + y_0^2 \end{aligned}$$

Hence, we must minimize the quadratic function (since $x_0^2 + y_0^2$ no depend on x and y)
$f(x,y) = \frac{1}{2}X^tBX - b^tX = x^2 + y^2 - 2x_0x - 2y_0y$
under the constraints $\Phi.\begin{pmatrix} x \\ y \end{pmatrix} \leq c$
with $B = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = 2I_2$ and $b = \begin{pmatrix} 2x_0 \\ 2y_0 \end{pmatrix}$

Problem:

$$Min\left(\frac{1}{2}X^tBX - b^tX\right)$$
$$\Phi.X \leq c$$

which is a known problem under *The quadratic optimization***.**

1.3.2. Example $n = 4$*:*

In the case where the point $M_0\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ is outside the polygon A_1, A_2, A_3, A_4 with
$A_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $A_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $A_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $M = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$

$$Min\left(\frac{1}{2}X^tBX - b^tX\right)$$
$$\begin{cases} 0 \leq x \leq 1 \\ 0 \leq y \leq 1 \end{cases}$$

with $B = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = 2I_2$ and $b = \begin{pmatrix} 2x_0 \\ 2y_0 \end{pmatrix}$
$\frac{1}{2}X^tBX - b^tX = x^2 + y^2 - 2x_0x - 2y_0y$ $(X = \begin{pmatrix} x \\ y \end{pmatrix})$
The problem:
$\min(x^2 + y^2 - 2x_0x - 2y_0y)$ on

$\begin{cases} 0 \leq x \leq 1 \\ 0 \leq y \leq 1 \end{cases}$
once $(x, y) = (x_1, y_1)$ found, the distance between point M_0 and polygon A_1, A_2, A_3, A_4 will be

$$d = x_1^2 + y_1^2 - 2x_0x_1 - 2y_0y_1 + x_0^2 + y_0^2$$

1.3.3. Remark:

1. The problem can be solved by the following methods:

- Activation algorithm
- Conjugate gradient algorithm (from extensions)
- Projected gradient algorithm
- Augmented Lagrangian algorithm
- Interior points algorithm
- Nelder-Mead method.

2. All these alternative methods are robust and lead to long, complex

and expensive algorithms.

3. It therefore remains, how to find this point (x, y)? hence our contribution see the last section.

2. Contributions

As we said, the introduction of complex numbers in algebra and geometry is very useful see [1], [2], [3], [5].... We therefore proceed with our purely geometric methods as well as complex numbers, using the triangulation of a convex polygon and a very specific application that we have just built.

In this paper, we will share some of our last original contribution:

- A very simple and fast tracking algorithm, and therefore an extremely low cost per iteration: An Algorithm for Judging Points Inside or Outside a Polygon (We use the basic notions of algorithmic see [4]).
- The neighboring Point and the spacing in the case where the point M is outside the polygon, without going through the quadratic optimization, we will give the simplest and fastest way to calculate the distance d between the point M and the polygon $(A_m)_{\{1 \leq m \leq n\}}$ and determine the point P of the boundary of the closest (neighboring) polygon to M.
- Above all it is an article full of examples for all the cases and methods and which are ready to be programmed and applied.

This work will be devoted to the study of the following outlines:

- A very specific application,
- Tracking algorithm,
- The neighboring Point and the spacing “the remoteness”.

2.1. THE application

We define the geometric application in the Euclidean plane $\mathcal{P}$ with an orthonormal frame $\mathcal{R}(O,I,J)$ with I of affix 1 and J of affix i

$$\begin{array}{rcl} F & : & \mathbb{R}^2 \longrightarrow \mathbb{R}^2 \\ & : & M \longmapsto M' \end{array}$$

such as

$$z' = az + b\overline{z}$$

z is the affix of M and $z\prime$ is that of M'.
So we will associate to F the linear map f

$$\begin{array}{rcl} f & : & \mathbb{C} \longrightarrow \mathbb{C} \\ & : & z \longmapsto az + b\overline{z} \end{array}$$

with $(a,b) \in \mathbb{C}^2$
Our map F can transform any triangle (OAB) into any other triangle (OCD) with specific a and b. Since it transforms any segment $[M,N]$ into another segment $[M',N']$.
So it suffices to find a and b such that

$$\begin{cases} F(A) = C \\ F(B) = D \end{cases}$$

Proposition 1.

$$P \in [M,N] \Longrightarrow F(P) \in [F(M), F(N)]$$

Proof
Let be $M(z_1)$, $N(z_2)$, $F(M) = M'(f(z_1))$, $F(N) = N'(f(z_2))$, $P(z) \in [M,N]$ et $P'(z') = F(P)$
We know that
$P(z) \in [MN] \Longrightarrow \exists(\alpha,\beta) \in \mathbb{R}_+^2/\alpha+\beta = 1$ and $z = \alpha z_1 + \beta z_2$
$F(M) = M'(f(z_1)) \Longrightarrow f(z_1) = az_1 + b\overline{z_1}$,
$F(N) = N'(f(z_2)) \Longrightarrow f(z_2) = az_2 + b\overline{z_2}$
$P'(z') = F(P) \Longrightarrow z' = f(z) = az + b\overline{z}$
$\Longrightarrow z' = f(z) = a(\alpha z_1 + \beta z_2) + b(\alpha\overline{z}_1 + \beta\overline{z}_2)$ (because $(\alpha,\beta) \in \mathbb{R}_+^2$)
$\Longrightarrow z' = f(z) = \alpha(az_1 + b\overline{z}_1) + \beta(az_2 + b\overline{z}_2)$
$\Longrightarrow z' = f(z) = \alpha f(z_1) + \beta f(z_2)$
$\Longrightarrow P' \in [M'N']$
hence the transformation of any segment $[MN]$ is another segment $[M'N']$. □

Proposition 2. *(OAB) is a triangle $\Longrightarrow$ $(O, F(A), F(B))$ is a triangle.*
And $M \in \overbrace{(OAB)} \Longrightarrow F(M) \in \overbrace{(O, F(A), F(B))}$.
with $\overbrace{(OAB)}$*: all inside the triangle.*

Proof

Let be $M \in \overbrace{(OAB)}$.

If M, A and B have the affix z, z_1 and z_2 respectively,
so $F(M)$, $F(A)$ and $F(B)$ are respectively of affix $f(z)$, $f(z_1)$ and $f(z_2)$,
with $f(z) = az + b\overline{z}$, $f(z_1) = az_1 + b\overline{z_1}$ and $f(z_2) = az_2 + b\overline{z_2}$

$M \in \overbrace{(OAB)}$ & O have the affix $0 \Longrightarrow \exists\,(\alpha,\beta,\gamma) \in \mathbb{R}_+^3 \;/\; \alpha+\beta+\gamma = 1$ and $z = \alpha z_1 + \beta z_2 + \gamma 0$
$\Longrightarrow z = \alpha z_1 + \beta z_2 \Longrightarrow \overline{z} = \alpha\overline{z_1} + \beta\overline{z_2}$ (as $(\alpha,\beta) \in \mathbb{R}^2$)
$\Longrightarrow az + b\overline{z} = a(\alpha z_1 + \beta z_2) + b(\alpha\overline{z_1} + \beta\overline{z_2})$
$\Longrightarrow f(z) = \alpha f(z_1) + \beta f(z_2) + \gamma f(0)$ (since $f(0) = 0$)
$\Longrightarrow F(M) \in \overbrace{(O, F(A), F(B))}$. □

Proposition 3. $F(M) \notin \overbrace{(O, F(A), F(B))} \Longrightarrow M \notin \overbrace{(OAB)}$.

Proof

The contrapositive of the previous proposition. □

Remark 2. *F-Type maps preserve only the shape of polygons (not for circles) and don't necessarily preserve distances and angles. But they remain interesting and useful in our case, which we will see below.*

2.2. Transformation of any triangle

Let a point M and a any triangle (EFG) such that M, E, F and G have the affix z, z_E, z_F and z_G respectively.

Firstly we transforme (EFG) in (OAB) using a esay translation, and secondly the map F transforming (OAB) into (OIJ).

$$\begin{array}{cccccc} & T & & F & \\ E & \to & O & \to & O \\ F & \to & A & \to & I \\ G & \to & B & \to & J \\ M & \to & M_1 & \to & M_2 \end{array}$$

T is the translantion such $T(E) = O$, $T(F) = A$ and $T(G) = B$ so
$A(z_A) = F - E$
$B(z_B) = G - E$
$M_1(z_1) = M - E$
and

$$\begin{aligned} z_A &= z_F - z_E \\ z_B &= z_G - z_E \\ z_1 &= z - z_E \end{aligned}$$

Let the map F which transforms the triangle (OAB) into a triangle (OIJ) with I and J have for affix respectively 1 and i. So it suffices to find the complex numbers a and b such that

$$\begin{cases} F(A) = I \\ F(B) = J \end{cases}$$

since $F(O) = O$.
Hence, if A and B have the affix z_A and z_B respectively, then
$\begin{cases} F(A) = I \\ F(B) = J \end{cases} \Leftrightarrow \begin{cases} f(z_A) = 1 \\ f(z_B) = i \end{cases}$
$\Leftrightarrow \begin{cases} az_A + b\overline{z_A} = 1 \\ az_B + b\overline{z_B} = i \end{cases} \Leftrightarrow$

$$\begin{pmatrix} z_A & \overline{z_A} \\ z_B & \overline{z_B} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ i \end{pmatrix}$$

$\begin{pmatrix} z_A & \overline{z_A} \\ z_B & \overline{z_B} \end{pmatrix}$ is an invertible matrix because
(OAB) is a triangle $\Longrightarrow O, A, B$ are not aligned $\Longrightarrow z_A\overline{z_B} \notin \mathbb{R}$
$\Rightarrow z_A\overline{z_B} \neq \overline{z_A}z_B \Longrightarrow \begin{vmatrix} z_A & \overline{z_A} \\ z_B & \overline{z_B} \end{vmatrix} = z_A\overline{z_B} - \overline{z_A}z_B \neq 0$
$\Longrightarrow$

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{z_A\overline{z_B} - \overline{z_A}z_B} \begin{pmatrix} \overline{z_B} & -\overline{z_A} \\ -z_B & z_A \end{pmatrix} \begin{pmatrix} 1 \\ i \end{pmatrix}$$

$\Updownarrow$

$$\begin{cases} a = \frac{\overline{z_B} - i\overline{z_A}}{z_A\overline{z_B} - \overline{z_A}z_B} \\ b = \frac{-z_B + iz_A}{z_A\overline{z_B} - \overline{z_A}z_B} \end{cases} \quad (1)$$

So, if $M_2 = F(M_1)$, so

$$\begin{aligned} z_2 &= az_1 + b\overline{z_1} \\ z_2 &= \frac{\overline{z_B} - i\overline{z_A}}{z_A\overline{z_B} - \overline{z_A}z_B} z_1 + \frac{-z_B + iz_A}{z_A\overline{z_B} - \overline{z_A}z_B}\overline{z_1} \\ z_2 &= \frac{\left(\overline{z_B} - i\overline{z_A}\right) z_1 - \left(z_B - iz_A\right)\overline{z_1}}{z_A\overline{z_B} - \overline{z_A}z_B} \end{aligned}$$

We have:
1- M is inside the triangle $(EFG) \Leftrightarrow M_1$ is inside the triangle $(OAB) \Leftrightarrow M_2$ is inside the triangle (OIJ) and
2- $M_2(x + iy)$ is inside the triangle (OIJ) if and only if

$$\begin{cases} x + y \leq 1 \\ 0 \leq x \\ 0 \leq y \end{cases}$$

$x + iy = z_2 \Longrightarrow x = \frac{z_2 + \overline{z_2}}{2}$ and $y = \frac{z_2 - \overline{z_2}}{2i} = i(x - z_2)$
with $z_2 = \frac{\left(\overline{z_B} - i\overline{z_A}\right)z_1 - \left(z_B - iz_A\right)\overline{z_1}}{z_A\overline{z_B} - \overline{z_A}z_B}$

$\Longrightarrow$

$$\begin{aligned}
x &= \frac{\frac{(\overline{z_B}-i\overline{z_A})z_1-(z_B-iz_A)\overline{z_1}}{z_A\overline{z_B}-\overline{z_A}z_B}+\overline{\left(\frac{(\overline{z_B}-i\overline{z_A})z_1-(z_B-iz_A)\overline{z_1}}{z_A\overline{z_B}-\overline{z_A}z_B}\right)}}{2}\\
x &= \frac{\frac{(\overline{z_B}-i\overline{z_A})z_1-(z_B-iz_A)\overline{z_1}}{z_A\overline{z_B}-\overline{z_A}z_B}+\left(\frac{(z_B+iz_A)\overline{z_1}-(\overline{z_B}+i\overline{z_A})z_1}{\overline{z_A}z_B-z_A\overline{z_B}}\right)}{2}\\
x &= \frac{\frac{(\overline{z_B}-i\overline{z_A})z_1-(z_B-iz_A)\overline{z_1}}{z_A\overline{z_B}-\overline{z_A}z_B}-\frac{(z_B+iz_A)\overline{z_1}-(\overline{z_B}+i\overline{z_A})z_1}{z_A\overline{z_B}-\overline{z_A}z_B}}{2}\\
x &= \frac{z_1\overline{z_B}-\overline{z_1}z_B}{z_A\overline{z_B}-\overline{z_A}z_B}
\end{aligned}$$

and

$$\begin{aligned}
y &= i\left(x-z_2\right)\\
y &= i\left(\frac{z_1\overline{z_B}-\overline{z_1}z_B}{z_A\overline{z_B}-\overline{z_A}z_B}-\frac{\left(\overline{z_B}-i\overline{z_A}\right)z_1-\left(z_B-iz_A\right)\overline{z_1}}{z_A\overline{z_B}-\overline{z_A}z_B}\right)\\
y &= i\frac{iz_1\overline{z_A}-i\overline{z_1}z_A}{z_A\overline{z_B}-\overline{z_A}z_B}\\
y &= \frac{\overline{z_1}z_A-z_1\overline{z_A}}{z_A\overline{z_B}-\overline{z_A}z_B}
\end{aligned}$$

So,

$$M \text{ is inside } (EFG) \Longleftrightarrow \begin{cases} x+y\leq 1\\ 0\leq x\\ 0\leq y\end{cases}$$

$$\Longleftrightarrow \begin{cases} \frac{z_1\overline{z_B}+\overline{z_1}z_A-\overline{z_1}z_B-z_1\overline{z_A}}{z_A\overline{z_B}-\overline{z_A}z_B}\leq 1\\ 0\leq \frac{z_1\overline{z_B}-\overline{z_1}z_B}{z_A\overline{z_B}-\overline{z_A}z_B}\\ 0\leq \frac{\overline{z_1}z_A-z_1\overline{z_A}}{z_A\overline{z_B}-\overline{z_A}z_B}\end{cases} \qquad (2)$$

such $\begin{cases} z_A=z_F-z_E\\ z_B=z_G-z_E\\ z_1=z-z_E\end{cases}$, $E\left(z_E\right)$, $F\left(z_F\right)$, $G\left(z_G\right)$ and $M\left(z\right)$

2.3. Tracking algorithm

Claim 1. Our method *consists of transforming each triangle* $\Delta_k=(A_1A_kA_{k+1})$ *obtained by the* triangulation *of the polygon into the triangle* (OIJ)*, which (this last triangle) gives us easy and fast calculations to detect whether a point is inside or outside a polygon.*

So
1- We transforme A_1 in O : By translation of landmark (change of coordinates)
2- $(A_1A_kA_{k+1})\longrightarrow(OC_kC_{k+1})$ and $M\left(z\right)\longrightarrow M_1\left(z-z_1\right)$ such that $A_1\left(z_1\right)$ $C_k\left(z_k-z_1\right)$, $C_{k+1}\left(z_{k+1}-z_1\right)$
3- $(OC_kC_{k+1})\longrightarrow(OIJ)$ by F_k and f_k such $f_k\left(z\right)=az+b\overline{z}$ and $M_1\left(z-z_1\right)\longrightarrow M_k\left(t_k\right)$, with

$$t_k=f_k\left(z-z_1\right)=a\left(z-z_1\right)+b\left(\overline{z-z_1}\right)$$

$$\begin{cases} a = \frac{\overline{z_B} - i\overline{z_A}}{z_A\overline{z_B} - \overline{z_A}z_B} \\ b = \frac{-z_B + iz_A}{z_A\overline{z_B} - \overline{z_A}z_B} \end{cases}$$

$A = C_k$, $B = C_{k+1}$, $z_A = z_k - z_1$, $z_B = z_{k+1} - z_1$
so

$$\begin{cases} a = \frac{\overline{(z_{k+1}-z_1)} - i\overline{(z_k-z_1)}}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)} \\ b = \frac{-(z_{k+1}-z_1) + i(z_k-z_1)}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)} \end{cases}$$

$$t_k = f_k(z - z_1) = \frac{\left[\overline{(z_{k+1}-z_1)} - i\overline{(z_k-z_1)}\right](z-z_1) - \left[(z_{k+1}-z_1) - i(z_k-z_1)\right]\overline{(z-z_1)}}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)}$$

Then using The Example: Case of the triangle (OIJ)
So, $M(x+iy)$ is inside the triangle (OIJ) if and only if

$$\begin{cases} x + y \le 1 \\ 0 \le x \\ 0 \le y \end{cases}$$

$$x + iy = t_k = \frac{\left[\overline{(z_{k+1}-z_1)} - i\overline{(z_k-z_1)}\right](z-z_1) - \left[(z_{k+1}-z_1) - i(z_k-z_1)\right]\overline{(z-z_1)}}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)}$$

and $\left[\overline{(z_{k+1}-z_1)} - i\overline{(z_k-z_1)}\right](z-z_1) - \left[(z_{k+1}-z_1) - i(z_k-z_1)\right]\overline{(z-z_1)} =$
$= \left[\overline{(z_{k+1}-z_1)}(z-z_1) - (z_{k+1}-z_1)\overline{(z-z_1)}\right] + i\left[(z_k-z_1)\overline{(z-z_1)} - \overline{(z_k-z_1)}(z-z_1)\right]$
we have
$(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1) \in iIR$
$\overline{(z_{k+1}-z_1)}(z-z_1) - (z_{k+1}-z_1)\overline{(z-z_1)} \in iIR$
$(z_k-z_1)\overline{(z-z_1)} - \overline{(z_k-z_1)}(z-z_1) \in iIR$
so

$$x = \frac{\overline{(z_{k+1}-z_1)}(z-z_1) - (z_{k+1}-z_1)\overline{(z-z_1)}}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)}$$

$$y = \frac{(z_k-z_1)\overline{(z-z_1)} - \overline{(z_k-z_1)}(z-z_1)}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)}$$

$$\begin{cases} x + y \le 1 \\ 0 \le x \\ 0 \le y \end{cases} \iff \begin{cases} \frac{\overline{(z_{k+1}-z_1)}(z-z_1) - (z_{k+1}-z_1)\overline{(z-z_1)} + (z_k-z_1)\overline{(z-z_1)} - \overline{(z_k-z_1)}(z-z_1)}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)} \le 1 \\ 0 \le \frac{\overline{(z_{k+1}-z_1)}(z-z_1) - (z_{k+1}-z_1)\overline{(z-z_1)}}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)} \\ 0 \le \frac{(z_k-z_1)\overline{(z-z_1)} - \overline{(z_k-z_1)}(z-z_1)}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)} \end{cases}$$

$$\iff \begin{cases} \frac{\overline{(z_{k+1}-z_k)}(z-z_1) - (z_{k+1}-z_k)\overline{(z-z_1)}}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)} \le 1 \\ 0 \le \frac{\overline{(z_{k+1}-z_1)}(z-z_1) - (z_{k+1}-z_1)\overline{(z-z_1)}}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)} \\ 0 \le \frac{(z_k-z_1)\overline{(z-z_1)} - \overline{(z_k-z_1)}(z-z_1)}{(z_k-z_1)\overline{(z_{k+1}-z_1)} - \overline{(z_k-z_1)}(z_{k+1}-z_1)} \end{cases}$$

Our contribution consists of introducing a simple and effective algorithm to detect if the position of a point is inside or outside a convex polygon.

Let $(A_m)_{1\leq m\leq n}$ be a polygon with affixes $(z_m)_{1\leq m\leq n}$ and M a point with affix z.

*2.3.1. **Method & Tracking algorithm:***

1. Order the points of the polygon $(A_m)_{1\leq m\leq n}$ (with affixes $(z_m)_{1\leq m\leq n}$) in $(B_m)_{1\leq m\leq n}$ (with affixes $(z'_m)_{1\leq m\leq n}$): as explained in the 1st section $B_m = A_{(m)}$;
2. Change of frame by translation ($\overrightarrow{u} = \overrightarrow{B_1O}$) such that:

$$(B_m)_{1\leq m\leq n} \quad \overset{\overrightarrow{u}}{\longrightarrow} \quad (C_m)_{1\leq m\leq n}$$

$(C_m)_{1\leq m\leq n}$ with affixes $(z'_m - z'_1)_{1\leq m\leq n}$ and M_c with affix $z - z'_1$
$\Rightarrow C_1 = O$ (the new origin of the landmark)
Let $z"_m = z'_m - z'_1$
$\Rightarrow C_m$ with affix $z"_m$ (and M_c with affix $z - z'_1$);

3. *Algorithm:*

For $k = 2$ to $n-1$
$f_k : \mathbb{C} \longrightarrow \mathbb{C}$
$\quad z \longmapsto a_k z + b_k \overline{z}$
such that $(C_1C_kC_{k+1}) \longrightarrow (OIJ)$ with J the affix point i and I the affix point 1.

$$\Rightarrow \begin{cases} 1 = f_k(z"_k) = a_k z"_k + b_k \overline{z"_k} \\ i = f_k(z"_{k+1}) = a_k z"_{k+1} + b_k \overline{z"_{k+1}} \end{cases}$$

$$\Rightarrow \begin{cases} a_k = \frac{\overline{z"_{k+1}} - i\overline{z"_k}}{z"_k\overline{z"_{k+1}} - \overline{z"_k}z"_{k+1}} \\ b_k = \frac{-z"_{k+1} + iz"_k}{z"_k\overline{z"_{k+1}} - \overline{z"_k}z"_{k+1}} \end{cases} \quad \text{(according to our Transformation see (1))}$$

$M_k(t_k)$ such that $t_k = f_k(z_c) = a_k z_c + b_k \overline{z_c}$

$$\Rightarrow t_k = a_k(z - z'_1) + b_k\left(\overline{z} - \overline{z'_1}\right) = r_k + is_k$$

$$\begin{cases} r_k = \frac{(a_k + \overline{b_k})(z - z'_1) + (b_k + \overline{a_k})(\overline{z} - \overline{z'_1})}{2} \\ s_k = \frac{(a_k - \overline{b_k})(z - z'_1) - (\overline{a_k} - b_k)(\overline{z} - \overline{z'_1})}{2i} \end{cases}$$

If $M_k \in (IOJ)$ (if $r_k + s_k \leq 1$ & $0 \leq r_k$ & $0 \leq s_k$ see (2)
Then Fin: M is inside the polygon.
Else $k \longrightarrow k+1$
M is outside the polygon. ($k = n$)

2.4. The neighboring Point and the spacing: The remoteness

In the case where point M is outside the polygon, our contribution consists (without going through *quadratic optimization*) of giving the simplest and fastest way to calculate the distance d between point M and the polygon $(A_m)_{1\leq m\leq n}$ and to determine the point P of the boundary of the closest (neighboring) polygon to M.

Our method consiste:
Let $(C_m)_{1\leq m\leq n}$ a Ordoned polygone such $C_1 = O$ (the origin of the frame of reference).
- First, we detect the nearest polygon vertex C_{m_0} to the point M.
So, the point of the polygon closest to M is in $(C_{m_0-1}, C_{m_0}]$ or $[C_{m_0}, C_{m_0+1})$.
- Second, we determine the projectors points P_- of M in $(C_{m_0-1}, C_{m_0}]$ and P_+ of M in $[C_{m_0}, C_{m_0+1})$.

So our contribution here is to determine the projectors points of the point M in the straight (C_{m_0-1}, C_{m_0}) & in the straight (C_{m_0}, C_{m_0+1}) using the complex numbers and the optimization.

1. Let P the projector point of the point M in the straight (the line) (CD), so

$$\begin{aligned} d(P,M) &= \min_{N\in(CD)} d(N,M) \\ d(P,M)^2 &= \min_{N\in(CD)} d(N,M)^2 \end{aligned}$$

Then, if $P(t^*)$, $M(z)$, $C(z_1)$, $D(z_2)$ and $N(t)$, so $P(t^*)$ is such

$$|t^*-z|^2 = \min_{N(t)\in(CD)} \left[|t-z|^2\right]$$

$N(t)\in(CD) \Longleftrightarrow \exists\alpha\in IR \,/\, t-z_2=\alpha(z_2-z_1)$
$\Longrightarrow t=t(\alpha)=z_2+\alpha(z_2-z_1)$
$\Longrightarrow |t-z|^2=|z_2+\alpha(z_2-z_1)-z|^2=|(z_2-z)+\alpha(z_2-z_1)|^2$
$\Rightarrow f(\alpha)=|t-z|^2=|z_2-z|^2+\alpha^2|z_2-z_1|^2+\alpha(\overline{z_2}-\overline{z})(z_2-z_1)+\alpha(z_2-z)(\overline{z_2}-\overline{z_1})$
$\Rightarrow f'(\alpha)=2\alpha|z_2-z_1|^2+(\overline{z_2}-\overline{z})(z_2-z_1)+(z_2-z)(\overline{z_2}-\overline{z_1})$

$$\begin{aligned} f'(\alpha) &= 0 \Longleftrightarrow \alpha=-\frac{(\overline{z_2}-\overline{z})(z_2-z_1)+(z_2-z)(\overline{z_2}-\overline{z_1})}{2|z_2-z_1|^2} \\ &\Leftrightarrow \alpha=\frac{(\overline{z_2}-\overline{z})(z_1-z_2)+(z_2-z)(\overline{z_1}-\overline{z_2})}{2|z_1-z_2|^2} \end{aligned}$$

$\Rightarrow t^*=z_2+\frac{(\overline{z_2}-\overline{z})(z_1-z_2)+(z_2-z)(\overline{z_1}-\overline{z_2})}{2|z_1-z_2|^2}(z_2-z_1)$
$\Rightarrow$

$$t^*=\frac{z+z_2}{2}+\frac{(z_1-z_2)^2}{2|z_1-z_2|^2}(\overline{z}-\overline{z_2})$$

2. Let the nearest polygon vertex $C_{m_0}(z"_{m_0})$ to the point M and the projectors points $P_-\left(t^*_-\right)$ of M in the straight (C_{m_0-1},C_{m_0}) and $P_+\left(t^*_+\right)$ of M in the straight (C_{m_0},C_{m_0+1}). So, the point $P(p)$ of the boundary of the closest (neighboring) polygon to $M(z)$ is such

$$|z-p|=\min\left\{|z-z"_{m_0}|,\left|z-t^*_+\right|,\left|z-t^*_-\right|\right\}$$

$t^*_-=\frac{z+z"_{m_0}}{2}+\frac{\left(z"_{m_0-1}-z"_{m_0}\right)^2}{2\left|z"_{m_0-1}-z"_{m_0}\right|^2}\left(\overline{z}-\overline{z"_{m_0}}\right)$

and
$t^*_+=\frac{z+z"_{m_0+1}}{2}+\frac{\left(z"_{m_0}-z"_{m_0+1}\right)^2}{2\left|z"_{m_0}-z"_{m_0+1}\right|^2}\left(\overline{z}-\overline{z"_{m_0+1}}\right)$

The method:

1. Ordering the points of the polygon $(A_m)_{1\leq m\leq n}$ into $(B_m)_{1\leq m\leq n}$ (as explained in the section: The scheduling, $B_m=A_{(m)}$);
2. Changing of the frame of reference by translation $(\overrightarrow{u})$

$$(B_m)_{1\leq m\leq n} \overset{\overrightarrow{u}}{\longrightarrow} (C_m)_{1\leq m\leq n}$$

such that:

$(B_m)_{1\leq m\leq n}$ of affixes $(z'_m)_{1\leq m\leq n}$ and $\overrightarrow{u}(-z'_1)$
$(C_m)_{1\leq m\leq n}$ of affixes $(z'_m - z'_1)_{1\leq m\leq n}$ et M_c of affix $z - z'_1$
$\Rightarrow C_1 = O$ (the origin of the frame of reference)
Let $z"_m = z'_m - z'_1$
$\Rightarrow C_m$ of affix $z"_m$ (& M_c of affix $z - z'_1$);

3. Calculate $d_0 = \min\{d(M_c, C_m)/m \in \{1, ..., n\}\}$;
4. Determine m_0 & C_{m_0} such that $d_0 = d(M_c, C_{m_0})$;
   be

$P_+(t^*_+)$: The projection of M_c on the line (C_{m_0}, C_{m_0+1})
$P_-(t^*_-)$: The projection of M_c on the line (C_{m_0-1}, C_{m_0})
The point $P(p)$ of the boundary of the closest (neighboring) polygon to $M_c(z - z'_1)$ is such

$$|z - z'_1 - p| = \min\left\{|z - z'_1 - z"_{m_0}|, \left|z - z'_1 - t^*_+\right|, \left|z - z'_1 - t^*_-\right|\right\}$$

The point of the boundary of the closest (neighboring) polygon to $M(z)$ is Q such

$$\overrightarrow{OQ} = \overrightarrow{OP} - \overrightarrow{u}$$

since $\overrightarrow{u}(-z'_1)$, so

$$Q(p + z'_1)$$

and

$$d = d(M_c, P) = d(M, Q) = d\left(M, (A_m)_{1\leq m\leq n}\right)$$

$d(M_c, P) = d(M, Q)$ because translations preserve distances.

The method diagram:

$(A_m)_{1\leq m\leq n}$	The scheduling	$(B_m)_{1\leq m\leq n}$	$\overrightarrow{u}(-z'_1)$	$(C_m)_{1\leq m\leq n}$
$A_1(z_1)$	$\longrightarrow$	$B_1(z'_1) = A_{(1)}$	$\longrightarrow$	$C_1(z"_1) = O$
$A_2(z_2)$	$\longrightarrow$	$B_2(z'_2) = A_{(2)}$	$\longrightarrow$	$C_2(z"_2)$
$\vdots$		$\vdots$		$\vdots$
$A_m(z_m)$	$\longrightarrow$	$B_m(z'_m) = A_{(m)}$	$\longrightarrow$	$C_m(z"_m) = C_m(z'_m - z_1)$
$\vdots$		$\vdots$		$\vdots$
$A_n(z_n)$	$\longrightarrow$	$B_n(z'_n) = A_{(n)}$	$\longrightarrow$	$C_n(z"_n)$
$M(z)$			$\longrightarrow$	$M_c(z - z_1)$

$\hookrightarrow \min\{d(M_c, C_m)/m \in \{1, ..., n\}\} \rightarrow m_0$ & C_{m_0}
$\hookrightarrow \begin{cases} Proj_{(C_{m_0}, C_{m_0+1})}(M_c) = P_+\left(t^*_+\right) \\ Proj_{(C_{m_0-1}, C_{m_0})}(M_c) = P_-\left(t^*_-\right) \end{cases}$
$\hookrightarrow P(p)$ such $d(P, M_c) = \min\{d(M_c, C_{m_0}), d(M_c, P_+), d(M_c, P_-)\}$
$\hookrightarrow \begin{cases} \text{The point of the boundary: } Q(p + z'_1) \\ \text{The remoteness: } d = d(P, M_c) \end{cases}$

3. Conclusion

In this work, we shared our original contribution which is based on the development of a new method and a simple algorithm which will allow us to define whether a point is inside a polygon or not.
To sum up, we have seen:

- A method of ordering the points of a polygon and the triangulization of a polygon;
- A very simple and fast tracking algorithm using the map F which transforms the triangle (OAB) into a triangle (OIJ), therefore an extremely low cost per iteration, and we avoided all cases like calculations of intersections, sum of areas or angles (arguments), and complex integrals (along the curve crossing the polygon), which cause problems at the time to develop and execute a program or software;
- The neighboring Point and the spacing in the case where the point M is outside the polygon, without going through the quadratic optimization, we gave the simplest and fastest way to calculate the distance d between the point M and the polygon $(A_m)_{\{1\leq m\leq n\}}$ and determined the point P of the boundary of the closest (neighboring) polygon to M.
- In our next work, we will translate these equations into application, based on daily life. Our next paper will be dedicated solely to real case studies which we will reveal later.

REFERENCES

1. Arbai, A. A methode to solve the equations of the second degree in the general case (with complex coefficients) ; International Journal of Scientific Engineering and Applied Science Volume-2, Issue-12 (2016).
2. Arbai, A. A Magic Formula and New Way to Solve Quadratic Equations. viXra.org (General Mathematics) viXra:2007.0244 (2020).
3. Arbai, A. Principes d'Algébre et de Géométrie; Editeur Aziz Arbai; ISBN 978-9954-32-699-2 (2012).
4. Arbai, A. Introduction en Analyse Numérique et Algorithmique; Editeur Aziz Arbai; ISBN 978-9954-34-617-4 (2014).
5. Arbai, A. Principes d'Algébre et de Géométrie; pour SMP et SMC - 2ème édition; ISBN 978-9954-32-699-2 (2016).
6. Chazelle, B. Triangulating a simple polygon in linear time. Discrete Comput Geom 6, 485–524 (1991).
7. El-Salamony, M., Guaily, A. Enhanced Modified-Polygon Method for Point-in-Polygon Problem. In: Farouk, M., Hassanein, M. (eds) Recent Advances in Engineering Mathematics and Physics. Springer, Cham. https://doi.org/10.1007/978-3-030-39847-7_4. (2020)
8. Geopandas: v0.12.2 Kelsey Jordahl, Joris Van den Bossche, Martin Fleischmann, Jacob Wasserman, James McBride, Jeffrey Gerard, Jeff Tratner, Matthew Perry, Adrian Garcia Badaracco, Carson Farmer, Geir Arne Hjelle, Alan D. Snow, Micah Cochran, Sean Gillies, Lucas Culbertson, Matt Bartos, Nick Eubank, maxalbert, Aleksey Bilogur, Sergio Rey, Christopher Ren, Dani Arribas-Bel, Leah Wasser, Levi John Wolf, Martin Journois, Joshua Wilson, Adam Greenhall, Chris Holdgraf, Filipe, and François Leblanc. geopandas/geopandas: v0.12.2, doi: 10.5281/zenodo.3946761 - 12 2022.
9. Nordbeck, S. & Rystedt, B. Computer cartography point in polygon programs. Bit Numerical Mathematics, 7(1), 39–64.(1967).
10. Saalfeld, A. It doesn't make me nearly as CROSS Some advantages of the point-vector representation of line segments in automated cartography. Geographical Information Systems 1(4):379-386 DOI: 10.1080/02693798708927823 (1987).
11. Schwinger, E. Point in polygon calculation using vector geometric methods with application to geospatial data. arXiv:2306.04316v1 [cs.CG] 7 Jun 2023.
12. Taylor, G. Point in polygon test. Survey Review 32(254):479-484 DOI: 10.1179/003962694791964997 (1994).