International Journal of Soft Computing 2 (3): 417-421, 2007

Plotting Algorithm and Computing Area for Any Plane Polygon and Realization by Computer Algebra

[1]Huang Dong-wei and [2]Wang Hong-li
[1]School of Sciences, Tianjin Polytechnic University, Tianjin 300160, China
[2]College of Mechanical Engineering, Tianjin University, Tianjin 30072, China

Abstract: In this study, we presented a method for plotting arbitrary n-gon with known coordinates of vertexes. In the meanwhile, we established a formula for calculating the area of the n-gon plotted. We compiled a Mathematica package to finish all the reseach.

Key words: Arbitrary n-gon, positive boundary, Mathematica, computer algebra, fractal

INTRODUCTION

As we know, it is not difficult to computing the area of an equilateral triangle or a regular n-gon (n>3). But it is not easy to computing the area of an arbitrary n-gon (n>3). We have denoted the Cartesian coordinates of the vertex of an arbitrary n-gon, but it is rather difficult for us to calculate the area of the polygon. Thus there are two problems we should cope with, one is how to draw lines between the vertexes without I nterleaving, the other one is how to compute the area of the polygon that has been plotted already. With the increasing of the vertexes number, it is necessary to establish a more efficient method for plotting the n-gon and an more efficient method for calculating the area of n-gon. The computer algebra system such as Mathematica has been used widely in many engineering fields (Bowring, 1990; Bian, 1997; Bian and Wu, 1995).

METHOD FOR PLOTTING AN ARBITRARY N-GON

The rules we must obey in plotting n-gon are that the coordinates of the vertexes we obtained (marked) is just a set of points with random position; the boundary lines we drawn from left to right and from down to up or on the contrary; all the lines of the boundary do not cross at other points; a vertex can be used as extremal point of only two lines of boundary.

We have got the coordinates of the vertexes of a polygon (Fig. 1).

If we draw lines with order of originset, the Fig. 2 is the result. It is obvious that it is not what we need.

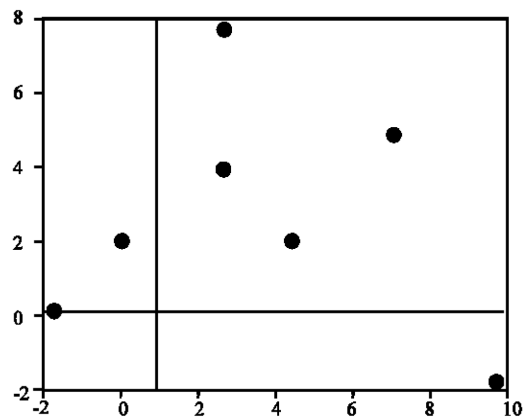

Fig. 1: Graph of the points originset

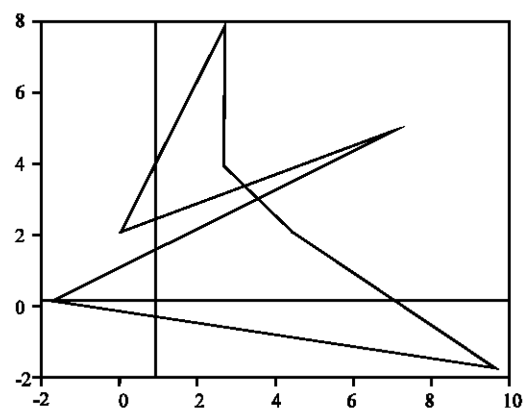

Fig. 2: Draw lines with order of originset

Corresponding Author: Huang Dongwei, School of Science, Tianjin Polytechnic University, No. 63 ChengLinZhuang Road HeDong District, (P.C.) 300160, Tianjin, People RepublicChina

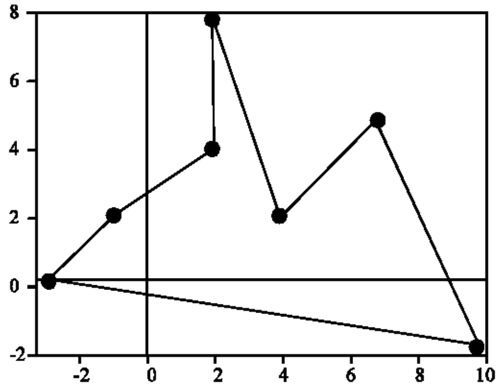

Fig. 3: Draw lines with order of readyset1

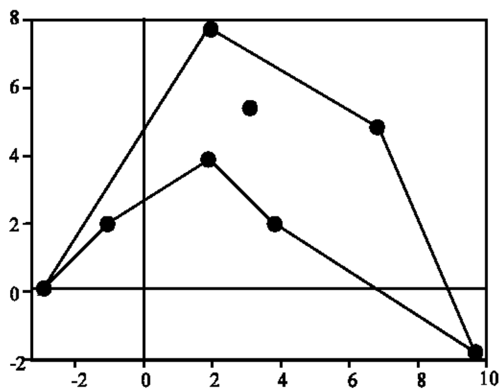

Fig. 4: Draw lines with order of readyset2

originset = {P_1 (2,4), P_2 (2,8), P_3 (-1,2), P_5 (-3,0), P_6 (10-2), P_7 (4,2)}

Rearranging the points according to the rules we listed above, we can obtain readyset1 (From left to right and from down to up).

readyset1 = {P_5 (-3,0), P_6 (10-2), P_4 (7,5), P_7 (4,2), P_2 (2,8), P_1 (2,4), P_3 (-1,2)}

We can draw lines with order of readyset1, a polygon is obtained as in Fig. 3 and the area of the polygon is 49.

If we rearrange the points according to the rules we listed above, we can obtain readyset2 (From left to right and from up to down).

readyset2 = {P_5 (-3,0), P_2 (2,8), P_4 (7,5), P_6 (10,-2), P_7 (4,2), P_1 (2,4), P_3 (-1,2)}

We can draw lines with order of readyset2, a polygon is obtained as in Fig. 4 and the area of the polygon is 40. The algorithm of rearranging points is arranging the points by the x-coordinate in ascending (from left to right). So the point at the left is the first point of readyset1. The second step is to find the next point, the point with the least y-coordinate in the other points is right one we search for. The third step is similar to the second step and so on until we have reached the point with the largest x-coordinate. Then the only step left is reverse the other points, so we obtain the points set with right order. Let's take originset as an example to demonstrate the algorithm.

I Originset = {P_1 (2,4), P_2 (2,8), P_3 (-1,2), P_4 (7,5), P_5 (-3,0), P_6 (10-2), P_7 (4,2)}

$$\xrightarrow{\text{sort as x acsending}} \left\{\begin{array}{l} P_5(-3,0), P_3(-1,2), P_1(2,4), P_2(2,8), \\ P_7(4,2), P_4(7,5), P_6(10,-2) \end{array}\right.$$

$$\rightarrow \left\{\begin{array}{l} P_5(-3,0), \quad \overbrace{P_6(10,-2)}^{\text{least y, and with largest x by chance}}, \\ \overbrace{P_3(-1,2), P_1(2,4), P_2(2,8), P_7(4,2), P_4(7,5)}^{\text{need to reverse}} \end{array}\right\}$$

$$\rightarrow \{P_5(-3,0), P_6(10,-2), P_4(7,5), P_7(4,2), P_2(2,8), P_1(2,4), P_3(-1,2), = \text{readysett1};$$

II Originset = {$P_1(2,4), P_2(2,8), P_3(-1,2), P_4(7,5), P_5(-3,0), P_6(10,-2), P_7(4,2)$}

$$\xrightarrow{\text{sort as x acsending}} \{P_5(-3,0), P_3(-1,2), P_1(2,4), P_2(2,8), P_7(4,2), P_4(7,5), P_6(10,-2)\}$$

$$\rightarrow \left\{\begin{array}{l} P_5(-3,0), \overbrace{P_2(2,8)}^{\text{largest y}}, \\ \overbrace{P_3(-1,2), P_1(2,4), P_7(4,2), P_4(7,5), P_6(10,-2)}^{\text{continue to find next point until reach the point with largest x}} \end{array}\right\}$$

$$\rightarrow \left\{\begin{array}{l} P_5(-3,0), P_2(2,8), P_4(7,5), \overbrace{P_6(10,-2)}^{\text{point with largest x}}, \\ \overbrace{P_3(-1,2), P_1(2,4), P_7(4,2)}^{\text{need to reverse}} \end{array}\right\}$$

$$\rightarrow \left\{\begin{array}{l} P_5(-3,0), P_2(2,8), P_4(7,5), P_6(10,-2), \\ P_7(4,2), P_1(2,4), P_3(-1,2) \end{array}\right\} = \text{readyset2}$$

Remarks: In part (II), the result of arrangement of 4 points in the front of readyset2 can be expressed as follow:

Step1: Beginning at point P_5 (-3,0), searching the point with the largest y, so point P_2 (2,8) is the next one.

Step2: From point P_2 (2,8), searching the point the largest x, so poin P_6 (10,-2) tseems to be the next one, but point P_4 (7,5) is above the line P_2P_6, The cross of the boundary lines would occur, so

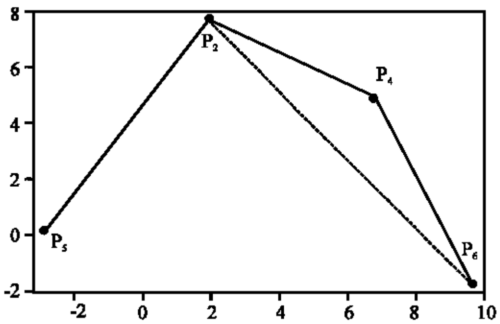

Fig. 5: An example for searching points in originset

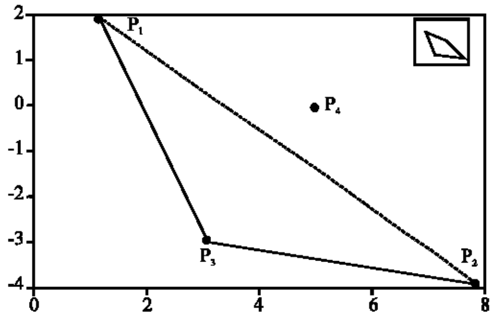

Fig. 6: an example for searching points in examset

according to our method, our program can automatically find point P_4 (7,5) the first point which locates above the line P_2P_6, and then the program continue to check whether there exist any points above the line P_4P_6, and so on (Fig. 5).

While if we have a set of point set examset = {P_1 (1,2), P_2 (8,4), P_3 (3,-3), P_4 (5,0)}, according to our rules it is easy to obtain rearranged point set as follow (Fig. 6):

readyset = {{P_1 (1,2), P_3 (3,-3), P_2 (8,4), P_4 (5,0)}

ESTABLISH FORMULA FOR COMPUTING AREA OF N-GON

As we know, the area of a plane domain with continuous boundary can be obtained by integral. It is well-known that dividing the n-gon into several triangles is a normal method in computing area of n-gon. And it is obvious that the calculation of this method is not rather, difficult but of some prolixity.

Let's assume that we have obtained a polygon B (Fig. 7) with 5 vertexes. We marked the vertexes anticlockwise around ∂B, $P_1 \to P_2 \to P_3 \to P_4 \to P_5$. According to vector operation we can calculate the area of triangle $\Delta P_1P_2P_3$

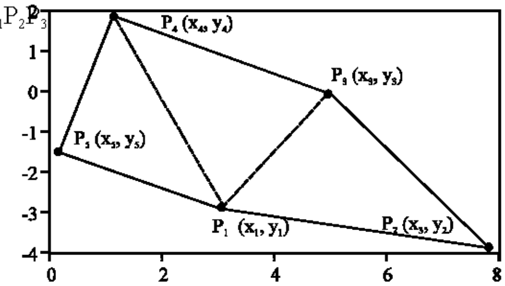

Fig. 7: An example for deriving formula on computing area of n-gon for vector operation

$$S_{\Delta P_1P_2P_3} = \frac{1}{2}\left|\overline{P_1P_2} \times \overline{P_1P_3}\right| = \frac{1}{2}\left\|\begin{matrix} \vec{i} & \vec{j} & \vec{k} \\ x_2 - x_1 & y_2 - y_1 & 0 \\ x_3 - x_1 & y_3 - y_1 & 0 \end{matrix}\right\|$$

$$= \frac{1}{2}[(x_1y_2 - x_2y_1) + (x_3y_2 - x_2y_3) - (x_1y_3 - x_3y_1)] \quad (1)$$

Noticing that coefficient of $\vec{k}$ is positive. In the same way, we have got other results $S\Delta P_1P_3P_4$ and $S\Delta P_1P_4P_5$, so we can obtain

$$S_B = \sum_{i=2}^{4} S_{\Delta P_1P_iP_{i+1}} = \frac{1}{2}\{[\sum_{i=1}^{4}(x_iy_{i+1} - x_{i+1}y_i)] - (x_1y_5 - x_5y_1)\} = \frac{1}{2}\sum_{i=1}^{5}(x_iy_{i+1} - x_{i+1}y_i) \quad (2)$$

Where we remark P_1 as P_6. Furthermore, we can deduce a formula for calculating the area of n-gon.

Theorem 1: Consider a n-gon Ω with vertexes P_1 (x_1,y_1), P_2 (x_2,y_2), • • • , P_n (x_n, y_n) and P_{n+1}, y_{n+1}) = P_1 (x_1,y_1), the boundary $\partial\Omega$ is positive as the order of the vertexes is $P_1 \to P_2 \to \cdots \to P_n \to P_{n+1}$., then the area of Ω is

$$S_\Omega = \frac{1}{2}\sum_{i=1}^{n}(x_iy_{i+1} - x_{i+1}y_i) \quad (3)$$

Proof: A method by using exterior product has been given in Bowring (1990). Herein we present an method only by using cross product. If we select point P_1 as starting point of vector P_1P_i ($i = 1,2\cdots,n$), for the boundary $\partial\Omega$ is positive, we can calculate the area of triangle $\Delta P_1P_iP_{i+1}$:

$$S_{\Delta P_1P_iP_{i+1}} = \frac{1}{2}\left|\overline{P_1P_i} \times \overline{P_1P_{i+1}}\right| = \frac{1}{2}[(x_1y_i - x_iy_1) + (x_{i+1}y_i - x_iy_{i+1}) - (x_1y_{i+1} - x_{i+1}y_1)] \quad (4)$$

So as $P_{n+1}(x_{n+1},y_{n+1})=P_1(x_1,y_1)$, we can obtain

$$S_\Omega=\sum_{i=2}^{n-1}S_{\Delta P_1P_iP_{i+1}}=\frac{1}{2}\{[\sum_{i=1}^{n-1}(x_iy_{i+1}-x_{i+1}y_i)]-x_1y_n-x_ny_1)\}=\frac{1}{2}\sum_{i=1}^{n}(x_iy_{i+1}-x_{i+1}y_i) \quad (5)$$

Now we present another method to proof the conclusion by Green Formula also seen in Luo and Zhong (2005), Yu (2001) and Tongji (2003). It is known that the area of $\Omega$with positive boundary $\partial\Omega$ can be calculated as

$$S_\Omega=\iint_\Omega dxdy=\frac{1}{2}\oint_{\partial\Omega}-ydx+xdy=\frac{1}{2}\sum_{i=1}^{n}\int_{P_iP_{i+1}}-ydx+xdy \quad (6)$$

Where the line segment P_iP_{i+1} has parametric equation:

$$\begin{cases}x=x_i+(x_{i+1}-x_i)t,\\ y=y_i+(y_{i+1}-y_i)t,\end{cases}\quad (0\le t\le 1) \quad (7)$$

Substituting (7) into (6), we can obtained

$$S_\Omega=\frac{1}{2}\sum_{i=1}^{n}\int_0^1[-(y_i+t(y_{i+1}-y_i))(x_{i+1}-x_i)+(x_i+t(x_{i+1}-x_i))(y_{i+1}-y_i)]dt$$

$$=\frac{1}{2}\sum_{i=1}^{n}(x_iy_{i+1}-x_{i+1}y_i) \quad (8)$$

Thus we have proved the formula (3) by different ways, cross product and curvilinear integral.

REALIZATION BY MATHEMATICA

In order to realize the method of plotting algorithm and calculating its area by formula (3), we compile a Mathematica package to finish all the work. A user only needs to provide a point set with random order, two polygon and their areas will be obtained, respectively.

Example 1: A random point set we have got in Methematica way, 50 points with x-coordinate within {-16,20} and y-coordinate within {-10,25} in real number. The results can be seen in Fig. 8 and 9.

```
pt1 = Table [{Random [Real, {-16,20}],
    Random[Real, {-10,25}]},{50}];
```

Example 2: Plotting regular 17-gon and calculate its area

```
pt2 = Table [{Cos [2k Pi/17.], Sin [2k Pi/17]},{k,17}];
```

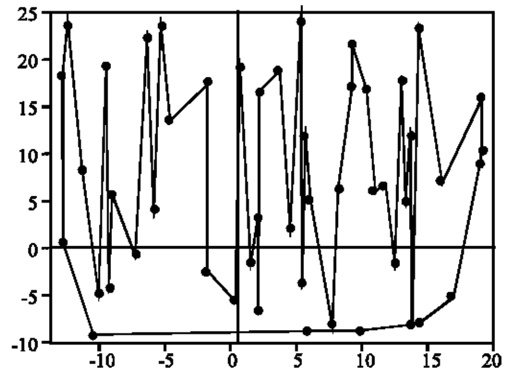

Fig. 8: Rarranging pt L with from up to down and plotting 50-gon and the approximate value of its area is 561.193

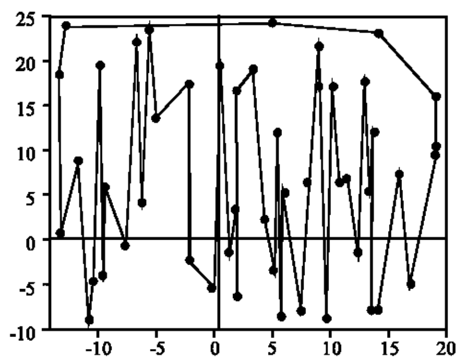

Fig. 9: Rarranging pt L with from up to down and plotting 50-gon and the approximate value of its area is 601.556

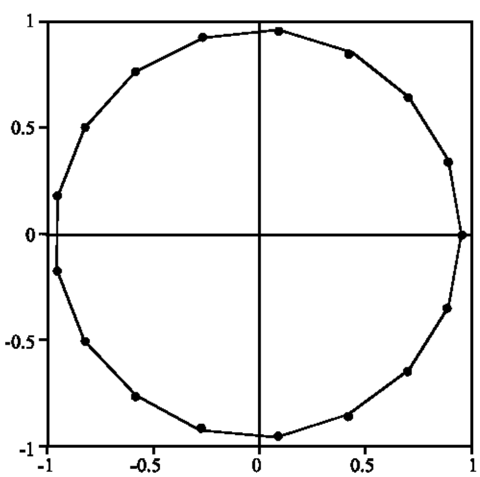

Fig. 10: Regular 17-gon with unit radius

By using our package, we can obtain the graph and its approximate value of area of the polygon is 3.07055 (Fig. 10).

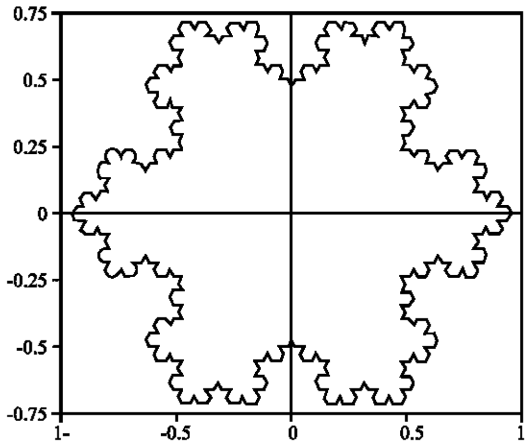

Fig. 11: Koch snowflake with unit radius

Example 3: A fractal graph and its area.

We have got the point set in which the number of the points is 720. By using the package, we obtained the value of area of the polygon is $\frac{298}{243}\sqrt{3}$, its approximate value is 2.12408 (Fig. 11 so called Koch snowflake).

CONCLUSION

In this study, we presented a method for plotting arbitrary n-gon with known coordinates of vertexes. By using cross product, we established a formula for calculating the area of the n-gon plotted. Both of the two jobs can be finished by our Mathematica package. Our Mathematica package can be used in many fields such as in Geodesy and some engineering works.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the National Natural Science Foundation of China (NNSFC) through grants Nos. 10472077 and 50375107.

REFERENCES

Bian, S. and X. Wu, 1995. A new algorithm for terrain corrections, Manuscripta Geodaetica, 20:515-518.

Bian, S., 1997. Some cubature formulas for singular integrals in physical geodesy, J. Geodesy, 71: 443-453.

Bowring, B., 1990. The transverse Mercator projection-a solution by complex numbers, Survey Rev., 30: 325-342.

Luo Zhi-qiang and Zhong Er-jie, 2005. Derivation of Formula for Any Polygonal Area and Its Applications, College Math., 21: 123-125.

Tongji University, 2003. Advanced Mathematics, Publishing House of Higher Education.

Yu De-sheng, 2001. Application of Curvilinear Integral in Double Integral, College Math., 17: 101-102.