2019年度卒業研究レポート

多角形の凸性の判定

明治大学　総合数理学部　現象数理学科
4年2組28番　高崎　寛大

平成32年3月31日

目 次

1 はじめに **2**

2 準備 **2**

2.1 点・直線・平面 2

2.2 線分 2

2.3 座標平面 2

3 三角形の符号付き面積 **2**

4 線分の交差判定 **3**

5 多角形 **4**

5.1 N 多角形 4

5.2 自己交差 4

5.3 自己接触 4

6 多角形の凸性 **5**

6.1 凸領域 5

6.2 定理 5

7 N 多角形の凸性を判定するアルゴリズム **5**

8 N 多角形の凸性を判定するプログラム **6**

9 最後に **8**

1　はじめに

私は卒業研究をする以前に、計算幾何学というものに興味をもっていた。しかし、なかなか学ぶ機会というものを得ることができなかった。実際、明治大学で計算幾何学の授業を受講するためには、他学科の授業を受講するしかなかった。そのため、本研究で計算幾何学を取り扱うことにした。

2　準備

2.1　点・直線・平面

・点とは部分をもたないものである。
・直線とはその上にある点について一様に横たわる線である。
・平面とはその上にある直線について一様に横たわる面である。
ただし、$\mathbb{R}$ は直線、$\mathbb{R}^2$ は平面とみなせる。

2.2　線分

線分 AB とは、異なる 2 点 A, B が存在し、$\{(1-t)A + tB \mid t \in [0,1]\}$ である。

2.3　座標平面

平面上に、互いに直交する 2 つの座標軸 x 軸,y 軸を用意する。交差する点 O を原点と呼び、$\mathbb{R}^2 = \{(x,y) \,|\, x \in \mathbb{R}\,,\, y \in \mathbb{R}\}$ の各要素 (a,b) に対して、x 座標が a,y 座標が b である点を対応させることで、平面上の点と $\mathbb{R}^2$ の要素と一対一の対応が得られる。この時、$\mathbb{R}^2$ を座標平面と呼ぶ。

3　三角形の符号付き面積

三角形の頂点の座標が、$P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3)$ と与えられている時、三角形の符号付き面積 S は、次の式で求められる。

$$S = \frac{1}{2}((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))$$

また、三角形の頂点の座標の並び方は、面積 S の符号によって判断することができる。

$$S\begin{cases} >0 & \leftrightarrows\ P_1,P_2,P_3\text{が反時計周りに並んでいる。}\\ =0 & \leftrightarrows\ P_1,P_2,P_3\text{が一直線上に並んでいる。}\\ <0 & \leftrightarrows\ P_1,P_2,P_3\text{が時計周りに並んでいる。}\end{cases}$$

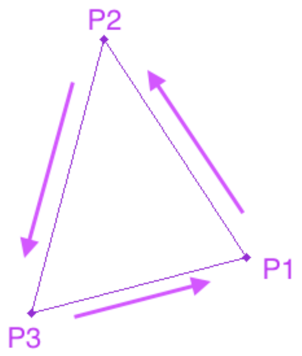

図 1: S > 0

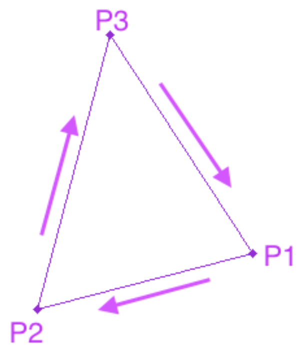

図 2: S < 0

以下、$P_1(x_1,y_1),P_2(x_2,y_2),P_3(x_3,y_3)$ からなる三角形の符号付き面積を $\Delta(P_1,P_2,P_3)$ で表す。

4 線分の交差判定

線分 P_1P_2 と P_3P_4 が互いに交差するときの必要十分条件は、

$$\Delta(P_1,P_2,P_3)\times\Delta(P_1,P_2,P_4)<0,\text{かつ}\Delta(P_3,P_4,P_1)\times\Delta(P_3,P_4,P_2)<0$$

である。

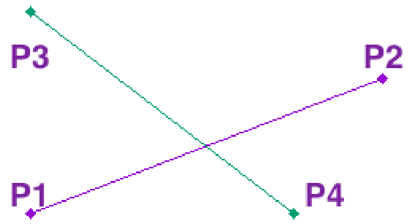

図 3: 線分どうしが交差する場合

5　多角形

多角形とは、以下の条件の線分によって描かれた平面図形である。
(1) 線分のどの端点もちょうど２つの線分に共有されている。
(2) 自己交差や自己接触がない。

また、多角形は平面を２つの互いに素な領域に分割する。すなわち、線分によって囲まれている領域（内部領域）と、それ以外の領域（外部領域）に分割する。多角形という用語で、多角形の境界とその内部領域を表すこともある。

5.1　N 多角形

N 多角形とは、２つの線分に共有されている端点を基準にして、その前後にある端点で三角形の符号付き面積を計算した時、その面積が 0 でない端点が N 個ある図形のことを指す。

5.2　自己交差

自己交差とは、図形を構成している線分 P_sP_{s+1}, P_tP_{t+1} が互いに交差していることを指す。（図４）

5.3　自己接触

自己接触とは、図形を構成している線分 P_sP_{s+1} の内部に P_t $(t \neq s, s+1)$ が存在していることを指す。（図５）

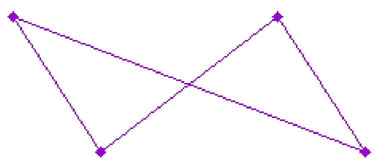

図 4: 自己交差のある図形

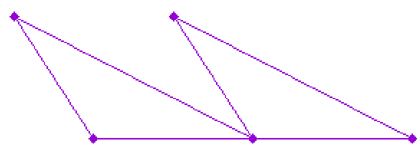

図 5: 自己接触のある図形

6　多角形の凸性

凸多角形（凸性のある多角形）であるとは、内部領域が凸（凸領域）である多角形である。

6.1　凸領域

領域 D における任意の 2 つの点 P_1 と P_2 に対し、線分 P_1P_2 が内部に完全に含まれるとき、領域 D は凸 (convex) であるという。

6.2　定理

「N 多角形 $P_1, P_2, \cdots, P_N$ が凸性を持つ」とは、「N 多角形の全ての点 P_k $(k = 1, 2, \cdots, N)$ に対して、P_{k-1}, P_k, P_{k+1} での符号付き面積の符号が一致する」。

7　N 多角形の凸性を判定するアルゴリズム

今回、以下のように図形を構成するデータを用意し、多角形であるかどうか判定した。そのあと、多角形であった場合、凸性を持つか判定した。

1. 2 次元の座標平面を用意し、
   N 個の点　$\{(x, y) \mid x \in \mathbb{N},\ y \in \mathbb{N},\ 0 \leqq x \leqq 10,\ 0 \leqq y \leqq 10\}$
   を 1 から順に番号付きでランダムに作成する。
2. ランダムに作成した点を 1 から順に線分で結び、図形を作成する。
3. 2 で作成した図形が、多角形であるか判定する。
4. 多角形である場合、その多角形が N 多角形であるか判定する。
5. N 多角形である場合、その N 多角形が凸性を持つか判定する。
6. 図形をデータから描写する。

8 N 多角形の凸性を判定するプログラム

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define N (4) /*N角形*/

/* 三角形の符合付き面積 */
double SA(int x1,int y1,int x2,int y2,int x3,int y3){
  double S=(double)((x2 - x1)*(y3 - y1) - (x3 - x1)*(y2 - y1
     ))/2;
  return S;
}

int main(int argc, char **argv){
  int h, j, Polygon, Convexity, NP;
  int x[N+1],y[N+1];
  double T,A,B,C,D;
  double Co[N];
  char fname[20];

  srand(time(0));
  /* ランダムに座標を用意する。*/
  for( h=0 ; h<=N ; h++){
    if(h<N){
      x[h] = rand()%11;
      y[h] = rand()%11;
      printf("x%d=%d,y%d=%d\n",h,x[h],h,y[h]);
    }else{
      x[h] = x[0];
      y[h] = y[0];
    }
  }

  /* 多角形かどうか判定する。*/
  Polygon = 1;
  NP = 1;
  for( h=0 ; h<=N-2 ; h++){
    for( j=h+1 ; j<=N-1 ; j++){
      A = SA(x[h],y[h],x[h+1],y[h+1],x[j],y[j]);
      B = SA(x[h],y[h],x[h+1],y[h+1],x[j+1],y[j+1]);
      C = SA(x[j],y[j],x[j+1],y[j+1],x[h],y[h]);
      D = SA(x[j],y[j],x[j+1],y[j+1],x[h+1],y[h+1]);
      printf("A=%lf ,B=%lf ,C=%lf ,D=%lf\n",A,B,C,D);

      /* 隣り合う線分についての条件付け */
      if( j == h+1 ){
   if(A == B){
```

```
    Polygon = 0;
    if((x[h]<=x[h+1] && x[h+1]<=x[h+2]) || (x[h]>=x[h+1] &&
       x[h+1]>=x[h+2])){
      Polygon = 1;
      NP = 0;
    }
  }
     }else if( h==0 && j==N-1){
  if(A == B){
    Polygon = 0;
    if((x[j-1]<=x[h] && x[h]<=x[h+1]) || (x[j-1]>=x[h] && x[
       h]>=x[h+1])){
      Polygon = 1;
      NP = 0;
    }
  }
     }else{
  /*  線分の交差判定  */
  if(A*B<0 && C*D<0){
    Polygon = 0;
  }
     }
   }
 }

 if(Polygon){
   printf("多角形である。\n");

   if(NP){
     printf("%d多角形である。\n",N);
   }else{
     printf("%d多角形でない。\n",N);
     Polygon = 0;
   }

 }else{
   printf("多角形でない。\n");
 }

 /*  凸多角形かどうか判定する。*/
 if(Polygon){

   for( h=0 ; h<=N-1 ; h++){
     Co[h] = SA(x[h],y[h],x[(h+1)%N],y[(h+1)%N],x[(h+2)%N],y
        [(h+2)%N]);
     printf("%lf\n", Co[h] );
   }
   Convexity = 1;
   for( h=0 ; h<=N-1 ; h++){
     if(Co[h] * Co[(h+1)%N] < 0){
  Convexity = 0;
     }
```

```
    }
    if(Convexity){
      printf("凸多角形である。\n");
    }else{
      printf("凸多角形でない。\n");
    }

  }
  /*  座標を映し出す。*/
  FILE *f;
  FILE *gid;

  sprintf(fname ,"data/data.txt");
  f = fopen(fname, "w");

  for( h=0 ; h<=N ; h++){
    fprintf( f ,"%d\t%d\n", x[h] , y[h]);
  }

  fclose(f);

  gid = popen("gnuplot" , "w");
  fprintf(gid, "set terminal png\n");
  fprintf(gid, "set xrange [0:10.0] \n");
  fprintf(gid, "set yrange [0:10.0] \n");
  fprintf(gid, "set output 'anim.png' \n");
  fprintf(gid, "plot 'data/data.txt' w lp pt 13\n");

  return 0;
}
```

9 最後に

最後になりましたが、今回の卒業研究にあたり、最後まで面倒を見てくださった明治大学総合数理学部現象 数理学科・桂田祐史先生、本研究を行うにあたり参考にさせていただいた浅野哲夫先生、譚学厚先生、平田富夫先生、砂田利一先生には厚くお礼申し上げま す。

参考文献

[1] 浅野哲夫（2007）『計算幾何ー理論の基礎から実装までー』共立出版.

[2] 譚学厚・平田富夫（2001）『計算幾何学入門』

[3] 砂田利一（2004）『幾何入門』岩波書店.

[4] 中村幸四郎・寺阪英孝・伊東俊太郎・池田美恵（1971）『ユークリッド原論』共立出版.