

# Javaによるポロノイ図の3次元表示

M2004MM010 市橋 翔輔

指導教員 鈴木 敦夫

## 1 はじめに

最適施設配置問題にはさまざまな型の問題がある。これらの解法は、それぞれ独自の解法があり、これらの問題を解く方法には、共通に使われる「ポロノイ図」と呼ばれる幾何図形である。ポロノイ図の効率的作図法が進んだおかげで、平面上の最適配置問題の解法は実用的になった。また平面上だけでなく空間上の問題にも適用するなど、さまざまな形で応用されている。インターネット上でも、ポロノイ図のような幾何図形の知識のない人や、PC上で図形を作成するソフトウェアの知識のない人でもポロノイ図を扱えるように、本研究ではWeb上でもポロノイ図を描写できるよう、「Web上」かつ「リアリティのある」描写に重点をおいてJavaとJava3Dを用いてポロノイ図の描写アプレットを平面・球面上にて作成した。

## 2 Javaによるポロノイ図の描写

### 2.1 Web上での表示

Javaを使うとプラットフォームに依存せず、多種多様なハードウェア/ソフトウェア環境で動作するプログラムを作成することができる。Javaが使われる環境の中で、特に重要なのはインターネットであり、Webページには、アプレットと呼ばれるJavaプログラムへの参照を埋め込むことができ、アプレットはWebブラウザ内で実行される。Javaで作成できるプログラムには、アプリケーションとアプレットの2種類があり、本研究ではアプレットを主に作成し、座標などの計算結果においては、アプリケーションを用いる。アプレットはWebブラウザのみでなく、アプレットビューアというツールを使って実行することができ、アプレットのテスト(コマンドプロンプト)によって使用できる。一般的に、アプレットはWebサーバーからユーザーのマシンにダウンロードされ、Webページにアプレットの参照が含まれている場合に、自動的に実行される。

### 2.2 平面ポロノイ図を表示するにあたって

ポロノイ図とは距離空間に複数の点を与えられたとき、それぞれの点に対する勢力圏でその空間を分割したものである。k次元空間にn個の点(母点) $p_1, \dots, p_n$ が与えられているとすると、このn個の母点の集合を $S = \{p_1, \dots, p_n\}$ とする。このとき、点 $p_i$  ( $i = 1, \dots, n$ )の勢力圏 $V(p_i)$ を、他のどの母点よりも母点 $p_i$ に近い点の集合とする、つまり、 $V(p_i) = \{p | d(p, p_i) \leq d(p, p_j), j = 1, 2, \dots, n, j \neq i\}$  ( $d(p, p_i)$ は $p, p_i$ の距離)と表すことができる。これをポロノイ領域という。点 $p_i$ と $p_j$ の2つのポロノイ領域の境界が共有す

る線分を $B(p_i, p_j) = \{p | d(p, p_i) = d(p, p_j), i \neq j\}$ をポロノイ辺という。3つのポロノイ領域の境界が存在する点 $A(p_i, p_j, p_k) = \{p | d(p, p_i) = d(p, p_j) = d(p, p_k), i \neq j \neq k\}$ が存在すればポロノイ点という。複数の母点を与えられたとき、ポロノイ点、またポロノイ辺の交点を計算し、平面、また空間や球面上にて、Javaアプレットを用いて出力することが本研究の主な課題である。

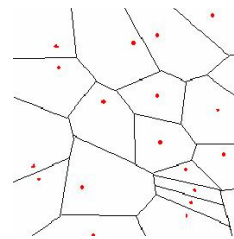


図1 ポロノイ図

### 2.3 ポロノイ図作成の算法

一般的に、ポロノイ図の構成方法には、大別して逐次添加法と再帰二分法の2種類があり逐次添加法を用いた。逐次添加法は、n個の母点 $p_1, p_2, \dots, p_n$ に対するポロノイ図 $V_n$ を作成する場合、次のような手順で行う。まず、4つの母点 $p_1, p_2, p_3, p_4$ よりなるポロノイ図 $V_5$ を作る。このようにしてm個の母点よりなるポロノイ図 $V_m$ に、母点 $p_{m+1}$ を添加してポロノイ図 $V_{m+1}$ を作成する。この作業を、 $m = 4, 5, \dots, n-1$ に対して $n-4$ 回行って、 $V_n$ を作成する。母点数が2つの時は、この2点を結ぶ垂直二等分線で直ちに求められる。3つのときはドロネー三角形の外心からそれぞれの辺に下ろした垂直二等分線として直ちに求められる。母点が4つ以上の時は、この3つの母点よりなるポロノイ図から作図を始めることになる。

(1) 母点 $p_4$ に一番近い母点を、もともとあった母点 $p_1, p_2, p_3, p_4$ の中から探し、その母点を $p_{(1)}$ とする。

(2) 母点 $p_4$ と母点 $p_{(1)}$ を結ぶ線分に垂直で二等分する線を引き、この線とポロノイ多角形 $V_{(p_{(1)})}$ のポロノイ辺の交点を求める。交点は2つあるので、どちらか一方の交点を $Q_1$ とする。

(3) この点 $Q_1$ ののっているポロノイ辺で、ポロノイ多角形 $V_{(p_{(1)})}$ と隣接しているポロノイ辺は、ポロノイ多角形を $V_{(p_{(2)})}$ とする。

(4) 母点  $p_{(2)}$  が  $p_{(1)}$  と同じかどうか確かめる.  $p_{(2)} = p_{(1)}$  で  $p_{(1)} = p_{(3)}$  であるから, 同じではない. 同じでない場合は, 母点  $p_4$  と母点  $p_{(2)}$  を結ぶ線分の垂直二等分線を引き, この線とボロノイ多角形  $V_{(p_{(2)})}$  のボロノイ辺との交点で  $Q_1$  以外の交点を  $Q_2$  とする.

(5)  $Q_2$  ののっているボロノイ辺で, ボロノイ多角形  $V_{(p_{(2)})}$  と隣接しているボロノイ多角形を  $V_{(p_{(3)})}$  とする.

(6)  $p_{(3)}$  が  $p_{(1)}$  と同じかどうかを確かめる.  $p_{(3)} = p_{(2)}$  で  $p_{(1)} = p_{(3)}$  であるから同じでない. 同じでない場合は, 母点  $p_4$  と母点  $p_{(3)}$  を結ぶ線分の垂直二等分線を引き, この線とボロノイ多角形  $V_{(p_{(3)})}$  のボロノイ辺との交点で  $Q_2$  以外の交点を  $Q_3$  とする.

(7)  $Q_3$  ののっているボロノイ辺で, ボロノイ多角形  $V_{(p_{(3)})}$  と隣接しているボロノイ多角形を  $V_{(p_{(4)})}$  とする. ここで  $p_{(4)}$  は  $p_{(1)}$  だから,  $p_{(4)}$  は最初の母点  $p_{(1)}$  と同じになる. 同じ場合は,  $Q_1, Q_2, Q_3$  よりなる多角形の内側にあるボロノイ辺を消す. 以上より得られた図形が  $p_1, p_2, p_3, p_4$  よりなるボロノイ図  $V_4$  である.

## 2.4 プログラムの手順

Java アプレットを用いて, 平面上のボロノイ図を作成した. また作成したプログラムはコマンドプロンプト上に, ボロノイ点の座標を出力する.

- (1) アプレットと枠 (縦・横) を作成する.
- (2) ランダム, もしくは個数を自分で指定してアプレット上に母点を生成する.
- (3) すべての 2 点において, すべての垂直二等分線 (中点と傾き, 始点と終点) を求める.
- (4) 全ての垂直二等分線の交点を配列に入れる.
- (5) 交点をヒープソートなどで, 交点の x 座標の値の小さいもの順に並べ替える.
- (6) 交点と交点に線を引く.
- (7) すべての線分について, 線分内の点と垂直二等分線の元になった点と 1 番近い線分を残すようにする.
- (8) 残った全ての線分を表示させる.

が大まかなプログラムの流れである.

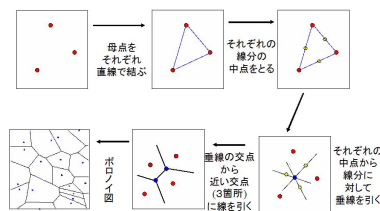


図2 ボロノイ図の簡単な作成手順

## 2.5 ボロノイ点とボロノイ辺の座標出力結果と描写結果

アプレットの枠の大きさに合わせて, その範囲内でランダムに母点をいくらか宣言し, ボロノイ点を計算して, ボロノイ線を引く始めの点からボロノイ線の終点まで線を引きアプレットを作成し, またコマンドプロンプト上に宣言した母点の座標を出力する. 母点の個数やアプレットの範囲は自分でも設定できるように簡単に変数で与えてあるので, その部分のプログラムを少し変えれば自分で設定もできる.

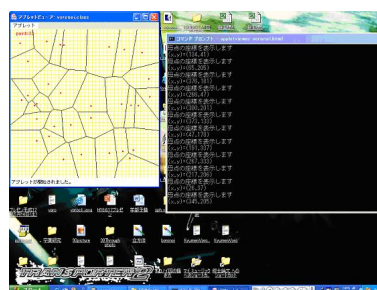


図3 ボロノイ図の Java アプレット

## 3 球面ボロノイ図の表示

### 3.1 球面ボロノイ図を表示するにあたって

ボロノイ図を球面上に拡張したものを球面ボロノイ図と呼ぶ. 球を中心ユークリッド座標の原点とし, 中心  $(0,0,0)$ , 半径を 1 としたものとする. 単位球面上に  $n$  個の点  $p_i (i = 1, \dots, n)$  が与えられたとき, 点  $p_i$  のボロノイ領域  $V_n(p_i)$  を  $V_n(p_i) = \{p | d(p, p_i) < d(p, p_j), j = 1, \dots, n\}$  とする. ここで, 距離は  $d(p, p_i)$  は,  $p$  と  $p_i$  の球面上の大円距離である. 一般的に, ボロノイ図の構成方法には, 大別して, 本研究では構成算法は考えず, 与えられた母点を 1 点ずつ

追加して、その都度ポロノイ辺を修正していく逐次添加法によって、出された極座標により、ポロノイ辺と母点を球面上に描写していく。

### 3.2 Java3D と Web3D

3次元でポロノイ図を表示するために Web3D のソフトウェアである Java3D を用いた Web は HTML だけで表現の豊かさに限界がある。Web 上で動的なコンテンツを実現する手段として、Web 上にもう一つの仮想現実（バーチャルリアリティ）を実現するために生まれた技術が Web3D である。Web3D は特定の企業が開発している特定の技術のことを指しているわけではなく、インターネット上の 3次元技術やコンテンツクリエーション全般を示すもので、Java3D を含め、その他の Web と関わりのある 3次元技術は、すべて Web3D に含まれる。Java で 3次元グラフィックスを実現するには、計算・描写を行うコードや OpenGL など呼び出すネイティブコードを自分で書いたり、利用したりしなければならなかったが、Java3D は手軽に 3次元グラフィックスを扱える。Java3D は、高いパフォーマンスを目的として設計されているため、全部が Java で実装されているわけではなく、描写部分は OpenGL や DirectX などの 3次元グラフィックス API をネイティブコールしていく。つまり、Java3D はこれらの 3次元グラフィックス API の上位 API であり、そのため描写部分に関してはハードウェアの性能を引き出す高速描写を実現している。Java3D は高レベルな API であるため、OpenGL で直接描写するよりプログラムより遅いが、現実的な速度で動作する。

### 3.3 球面ポロノイ図の表示方法

緯度と経度の考え方で  $-1/2$  から  $2$  の範囲で出力された座標データを受け取り、Java の空間座標でも変換できるように変換し、アプレットにて描写させる。変換の式は  $x=r\cos \theta \cos \phi$ ,  $y=r\sin \theta \cos \phi$ ,  $z=r\sin \theta$  となる。受け取るデータは  $(\theta, \phi)$  の型で、母点 P とポロノイ線の始まりの点 M と終わりの点 D である。また、3次元空間で物体を構成するので、全ての方向から物体を観察できるようにマウスでトラックすることにより、回転・移動できるよう実装した。また Java には曲線を描くパッケージは存在しないので、球面上のポロノイ辺をより滑らかに見せるための工夫を加えてみた。作成した球面ポロノイ図のアプレットは曲線を描いているわけではないが、ポロノイ辺を目で見る範囲では十分に曲線を描いているように見える。

- (1) シーングラフの作成
- (2) アプレットと枠(縦・横)を作成する
- (3) 物体を全ての方向から見れるようにする

- (4) データを配列に取り込む
- (5) データを空間座標に変換する
- (6) ポロノイ辺を円滑にする
- (7) 母点とポロノイ辺を描写する

が大まかなプログラムの流れである。

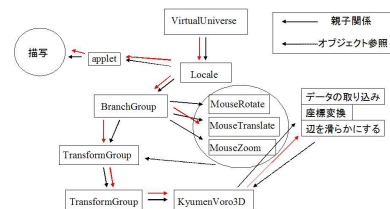


図 4 プログラムのクラス図

### 3.4 球面ポロノイ図の描写結果

球面上に生成された母点と、逐次添加法によって生成されたポロノイ辺の始点と終点の座標を受け取り、Java3D のアプレットにて描写させた。このプログラム内では座標の計算はしておらず、配列に受け取ったデータを組み込み出力している。また、マウスでクリックすることで、回転やトラックすることもできるので、全ての方向から眺めることができる。

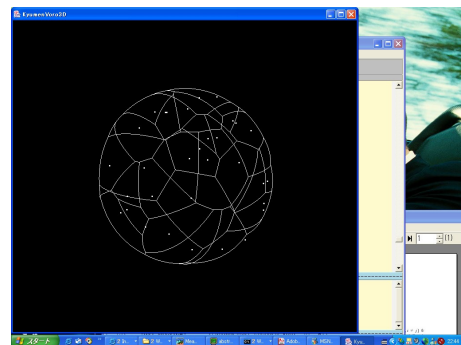


図 5 球面ポロノイ図の Java アプレット母点 32

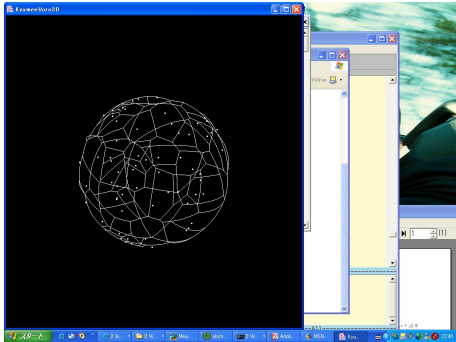


図 6 球面ポロノイ図の Java アプレット母点 64

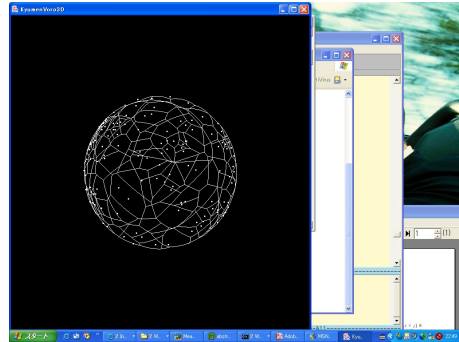


図 8 球面ポロノイ図の Java アプレット母点 128

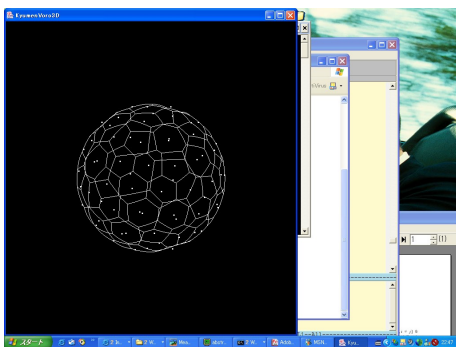


図 7 球面ポロノイ図の Java アプレット母点 64-2

#### 4 考察

本研究では、平面上のポロノイ図と球面ポロノイ図のアプレットを作成した。どのプログラムもまだまだ改良することで、ポロノイ図の正確性、また見た目的にもリアリティを増すことができる。また、作成したポロノイ図の構成算法のアルゴリズムが必ず正確であるかは保証できなく、何度か実行してみると、少し正確な値とは異なる座標が出力されたり、実行中にエラーがでてしまう。球面ポロノイ図のデータは、逐次添加法により計算するプログラムがあったのだが、自分で動かしてみようとしたところ動かなかったため、データは 4 つのものでしか試せなかった。また、Java3D には曲線を描写するパッケージは含まれていないため、曲線部分に工夫が必要であった。Java と Java3D を同時に扱うことでプログラム言語は両方の知識について、知っていなければならない。Java と Java3D、そして幾何学について知識を深めていけば、まだまだ応用させていける可能性は十分にある。

#### 参考文献

- [1] 球面上のポロノイ図とその応用：南山大学 大学院 経営学研究科，近藤明日香 (1997).
- [2] 多次元 Voronoi ダイアグラムの構成算法の研究：玄英澤 (1983).
- [3] プログラミングバイブル：えんどうやすゆきほか (1999).
- [4] 独習 Java：ジョゼフ・オニール.
- [5] やさしくわかる Java3D(2001)，太田篤行.
- [6] 大山崇，<http://www.nirarebakun.com/>
- [7] 上坂陽平，<http://www.th.cs.meiji.ac.jp/home/kamisaka/>