

Title	ディスクレパンシーを効率よく求めるアルゴリズムに対する研究
Author(s)	小山, 紘樹
Citation	
Issue Date	2010-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8925
Rights	
Description	Supervisor:浅野 哲夫, 情報科学研究科, 修士

修 士 論 文

ディスクリパンシーを効率よく求める
アルゴリズムに対する研究

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

小山 紘樹

2010 年 3 月

修 士 論 文

ディスクレパンシーを効率よく求める アルゴリズムに対する研究

指導教官 浅野哲夫 教授

審査委員主査 浅野哲夫 教授
審査委員 上原隆平 准教授
審査委員 宮地充子 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

0810025 小山 紘樹

提出年月: 2010 年 2 月

概 要

本研究では短い計算時間でディスクレパンシーを求めることができるアルゴリズムの開発を目的とする．特に，点を内部に含む四角形を考慮する方法について， $O(n^3)$ 時間でディスクレパンシーを求めるアルゴリズムを提案する．また，本論文ではいくつかの点集合生成方法を取り上げ，それぞれの点数とディスクレパンシーの関係も実験的に求める．

目次

第1章	はじめに	1
1.1	研究の背景と目的	1
1.2	研究の目的	2
1.3	本論文の流れ	2
第2章	ディスクレパンシーの計算	3
2.1	ディスクレパンシー問題の例	3
2.2	定義	3
2.3	評価対象の長方形	4
2.4	ディスクレパンシーの計算例	6
第3章	効率のよいアルゴリズム	9
3.1	素朴な方法	9
3.2	効率のよいアルゴリズム	9
3.2.1	上辺・下辺の決定	9
3.2.2	左辺・右辺の決定	10
3.2.3	ディスクレパンシーの計算	10
3.2.4	計算時間	12
3.2.5	タイプ1, タイプ2の長方形による評価	12
第4章	点集合の種類とディスクレパンシー	14
4.1	ランダム点集合	14
4.2	格子状点集合	15
4.3	格子状を斜めにした点集合	15
4.4	Van der Corput 集合	16
4.4.1	ディスクレパンシーの証明	17
4.5	点数とディスクレパンシー	20
第5章	まとめ	24
第6章	謝辞	25

第1章 はじめに

1.1 研究の背景と目的

ディスキレパンシー (discrepancy) とは、日本語では不一致を意味するが、本論文ではどれくらい一様なのかを示す指標を表している [1]。たとえば一次元では、直線状に点集合が配置されているとき、図 1.1 のように、どの点についても隣の点との距離がほぼ同じなら、これらの点は一様に分布している、すなわち、対応する点集合のディスキレパンシーは低いといえる。



図 1.1: 一様な一次元点集合



図 1.2: 一様でない一次元点集合

二次元では、平面上に配置された点集合に対しても分布の一様さを評価することはできるが、一次元のように点同士の距離だけで単純に判断することはできない。一次元の場合は隣の2点との関係だけを考えればよかったが、二次元では隣の点を決めることが難しいからである。そのため、点を含む四角形すべてを考慮し、最も多く点を含んだときと少なかったときの差によってディスキレパンシーを求める方法や、四角形に理想的に含まれるであろう点の数と実際に含まれている点の数の差をとる方法が、さらには、点集合内で最大空円をとる方法など、様々な方法が考えられる。当然、評価方法が変われば、一様とみなせるかの評価、すなわち、ディスキレパンシーも変わってくる。

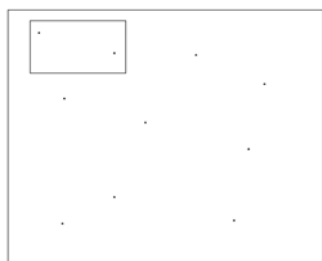


図 1.3: 四角形を考慮する方法

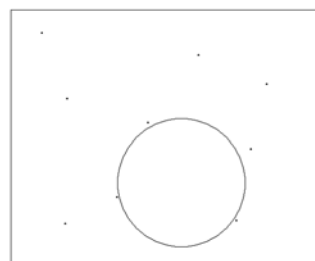


図 1.4: 最大空円をとる方法

ディסקレパンシーを考えるにあたって、点の一様性をどのように評価するかという問題と、評価の高い点集合をどのように生成するかという問題がある。前者においては、点を含む図形を変えることで評価がどのように変化するか、また、すべての図形を調べるとなるときりがないので、効率的に図形を選ぶにはどうするか、といったことが考えられる。たとえば、点を含む四角形について考えると、四角形のそれぞれの辺に与えられた点のうちの一つが含まれるようなものだけでも $O(n^4)$ 通りの四角形を考えることとなる。そのような四角形それぞれについて、与えられた点がどれだけ含まれるのかを判定をしなければならないため、単純な方法では $O(n^5)$ 時間がかかることとなる。そのため、この問題は点数の多い集合のディスクレパンシーを求める際には、いかに効率よく計算できるかが焦点となる。

一様な点集合を生成する方法としては、Van der Corput 集合や、格子状に配置された点集合を少し傾けたものなどが知られている。その応用として、デジタルハーフトーニングがある [2], [3]。限られた色数で元の画像を表現したり、白黒画像で濃淡を表現するときに、この一様性が重要となってくる。また、ディスクレパンシーが小さくなるよう定義された「超一様分布列」というものがあり、定積分の数値を高速に求めるために利用されている [4]。

1.2 研究の目的

本研究では、より短い時間でディスクレパンシーを計算できるアルゴリズムの開発を目的とする。特に、二次元において、点集合領域内で考えうるすべての四角形の中に含まれる点数と、理想的な一様分布の場合に含まれるであろう点数との差からディスクレパンシーを計算する方法について求める。

また、その応用例として、一般にディスクレパンシーの値が優れているとされる Van der Corput 集合や、格子状点集合を斜めに傾けたもの、その比較としてランダム点集合と格子状点集合について、点数とディスクレパンシーの関係も調査する。

1.3 本論文の流れ

本稿では、まず 2 章でディスクレパンシー計算を定義し、3 章で効率のよいアルゴリズムを提案する。ついで、4 章より今回ディスクレパンシーを計算する点集合の定義と実験結果を示す。

第2章 ディスクレパンシーの計算

2.1 ディスクレパンシー問題の例

2次元における一様な点集合とはどのようなものかと聞かれたとき、たいていの人は図 2.1 のような点集合を思い浮かべるであろう。しかし、1次元では等間隔に並んだ点集合が一様だったと考えると、図 2.2 のような点集合の方が一様に見える。



図 2.1: ランダムに 100 点配置した点集合

図 2.2: 格子状に 100 点配置した点集合

では、実際どちらが一様なのか。それを定量的に示すものがディスクレパンシーである。

2.2 定義

点の一様性を評価する単純な方法として、点を含む軸平行長方形を考えるものがある。この方法では、長方形 X 内に配置された n 点の一様さ、すなわちディスクレパンシーを評価する際に、長方形 X に含まれる任意の長方形 R を X の中で移動させたとき、 R に含まれる点の個数がどのように変化するか注目する。もし点が一様に配置されていれば、 R を動かしても R 内の点の個数に余り大きな変化はないはずである。このような観点から、長方形 R に関するディスクレパンシーを次のように定める。

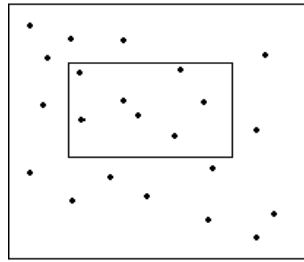


図 2.3: 点集合内で点を含む長方形の例

$$D = \max \left| \#R - n \frac{|R|}{|X|} \right|$$

上記の式において， $\#R$ は長方形 R に含まれる点の数， $|R|$ は長方形 R の面積， $|X|$ は点を配置した長方形 X の面積を示す．

ディスクレパンシーは点差なので絶対値で定義されるが，実際に計算するときは正の値の最大値と，負の値の最小値を求めることになる．

2.3 評価対象の長方形

このように考えると，あらゆる長方形を考慮するため，無限に時間がかかってしまうように思われる．しかし，実際は境界に点を含むような長方形だけを考えればよい．

正の値のディスクレパンシーを考えると，これを最大にしようとすると，長方形 R に含まれる点の数を変えないまま面積を小さくすることになる．図 2.4 の点線のような長方形 R を考えたとき，少し小さくしても含まれる点の数は変わらない．そこで，点の数が変わるぎりぎりまで長方形 R を小さくすると，実線の長方形のように境界に点が来る形になる．これが最大となるため，点線のような長方形は考えなくてもよくなる．

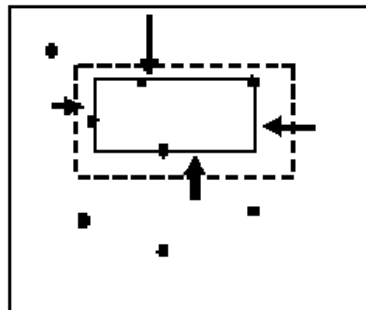


図 2.4: 長方形を小さくする

逆に負の値のディスクリパンシーを考えると、これを最小にしようとすると、長方形 R に含まれる点の数を変えないまま面積を大きくすることになる。図 2.5 の点線のような長方形 R を考えたとき、少し大きくしても含まれる点の数は変わらない。そこで、点の数が変わるぎりぎりまで長方形 R を大きくすると、実線の長方形のように境界に点をぎりぎり含まない形になる。これが最小となるため、点線のような長方形は考えなくてもよくなる。

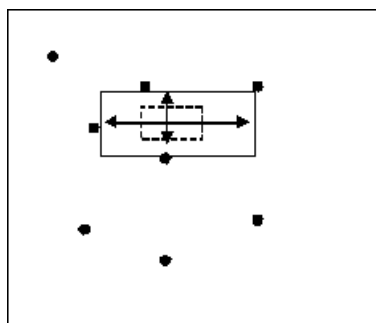


図 2.5: 長方形を大きくする

どちらの場合も、境界に点がある長方形を考えることになり、境界の点を含むか含まないかが違うだけである。

境界に点を含む長方形は、次の 3 タイプに分けられる。

- タイプ 1: 2 つの対角頂点が与えられた点で決まる場合
- タイプ 2: 1 つの対角頂点が与えられた点で決まり、その頂点に接続しない 2 辺が別の 2 点で決まる場合
- タイプ 3: 4 つの辺が与えられた点で決まる場合

これらはそれぞれ図 2.6, 図 2.7, 図 2.8 のようになる。

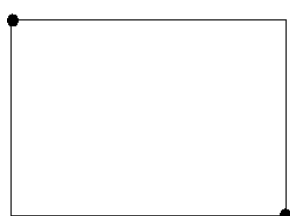


図 2.6: タイプ 1



図 2.7: タイプ 2

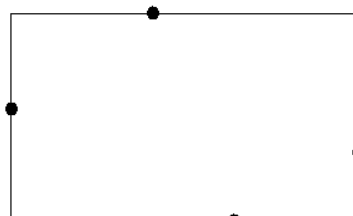


図 2.8: タイプ 3

タイプ 1 の長方形は，与えられた点集合に対して全 2 点对を考えることになるため， $O(n^2)$ 通り存在する．

タイプ 2 の長方形は，角におく点に注目すると，図 2.9 のように 4 つの領域でそれぞれ 2 点对を考えることになる．角におく点は n 点存在するため， $O(n^3)$ 通り存在することになる．



図 2.9: 角におく点に注目したとき

タイプ 3 の長方形は，図 2.10 のように，上辺と下辺のパターンとそのときの左辺・右辺のパターンを考えるとわかりやすい．上辺・下辺のパターンは 2 点对を考えるに等しいので $O(n^2)$ 通り，左辺・右辺のパターンも上辺・下辺の間の領域で 2 点对を考えることになるので $O(n^2)$ 通り存在する．そのため，タイプ 3 の長方形は $O(n^4)$ 通り存在することになる．

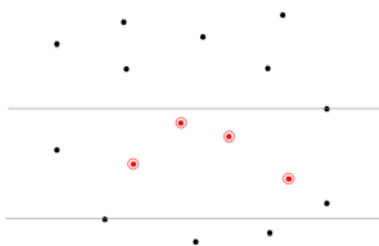


図 2.10: 上辺・下辺を固定した状態

実際に左辺・右辺を考えるとき，上辺・下辺の間に存在する点 (図 2.10 で円で囲った点) は，長方形にしたとき，その内部に入ってしまうので考える必要はなく，調べる長方形は $O(n^4)$ 通りよりも少なくなる．

2.4 ディスクレパンシーの計算例

2.1 節の問題でこの方法をとったとき，結果は次のようになる．

- 図 2.1(ランダムに配置した点集合):13.024
- 図 2.2(格子状に配置した点集合):19.0

以上より，ランダムに配置した点集合の方が一様であるという結果になる．

格子状に点を配置するとディスクリパンシーが大きくなる理由として，格子状の点集合では，同じ大きさの長方形でも少し位置をずらすだけで，図 2.12 のように点一行が全部入ったり，図 2.11 のように全く入らなかったりと，最大・最小の差が大きくなるからである．

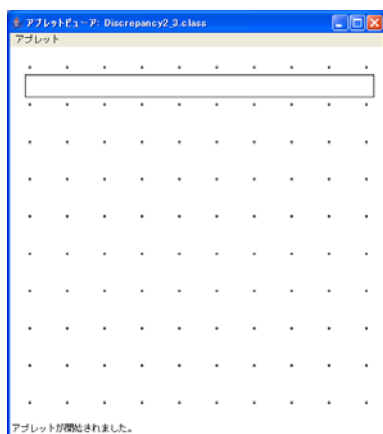


図 2.11: 点を含まない長方形



図 2.12: 少しずらすことで点を含む長方形

今回の 100 点での例では，図 2.13 のように考える．

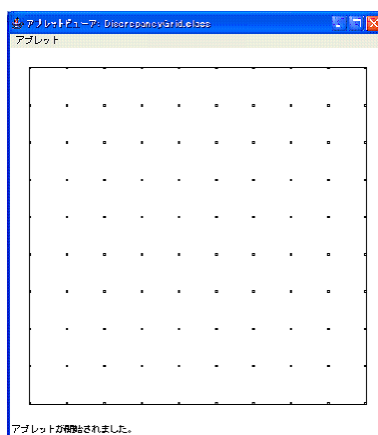


図 2.13: 点をすべて含む最も小さい長方形

点集合領域は正方形なので，一辺を 1 とすると，点と点の間の感覚は $\frac{1}{10}$ ，点と端の間は $\frac{1}{10} \times \frac{1}{2}$ となる．そのため，図 2.13 の長方形の大きさは $\frac{9}{10} \times \frac{9}{10}$ となる．

今，面積 1 で 100 点なので，一樣ならば

$$100 \times \frac{9}{10} \times \frac{9}{10} = 81$$

と計算でき，81 点あるはずである．しかし，実際はこの大きさに 100 点全部含んでいるので，その差

$$100 - 81 = 19$$

がディスクレパンシーとなる．また，図 2.14 のように考える必要もある．

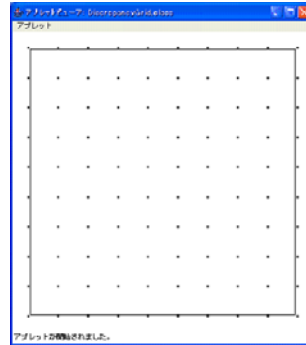


図 2.14: 点をすべて含む最も小さい長方形を少し小さくした場合

図 2.14 は，図 2.13 の長方形を，外側の点をぎりぎり含まない程度に小さくしたものである．

ほんのわずかに小さくしただけで，面積をほとんど変えずに 36 点も外側に出てしまっていることがわかる．このときのディスクレパンシーは，面積が変化してないものとして，

$$(100 - 36) - 81 = -17$$

と計算できる．

100 点では，図 2.13 の考え方のほうがディスクレパンシーは大きくなる．しかし点数が増え，点をすべて含む最も小さい長方形と領域全体の大きさが近づく代わりに，長方形を小さくしたときに外に出てしまう点数が増えるため，図 2.14 の考え方で求まるディスクレパンシーの方が大きくなる．

第3章 効率のよいアルゴリズム

3.1 素朴な方法

ディスクレパンシーを計算する際、単純にやろうとすると、調査する長方形を列挙するだけで $O(n^4)$ 時間かかることになる。さらに、長方形一つ一つに対して、点集合全体のうち、内部または境界に存在する点の数を調べなければならないため、長方形ごとに $O(n)$ 時間かかることになる。

よって、単純な方法では $O(n^5)$ 時間かかってしまい、点数が増えると計算がなかなか終わらなくなってしまう。

3.2 効率のよいアルゴリズム

そこで、より短い時間でディスクレパンシーを計算できるアルゴリズムを提案する。このアルゴリズムを使えば、 $O(n^3)$ 時間でディスクレパンシーを求めることができる。最初に、4点で4辺を決めるタイプ3の長方形を考える場合のアルゴリズムを説明する。

3.2.1 上辺・下辺の決定

まず、点集合を y 軸方向にソートしてから上辺と下辺を定める。このとき、適当に選ぶのではなく、図 3.1、図 3.2 のように y 座標の小さい点から順番に決めていく。最初の下辺を固定して、上辺を一点ずつ上げていく。

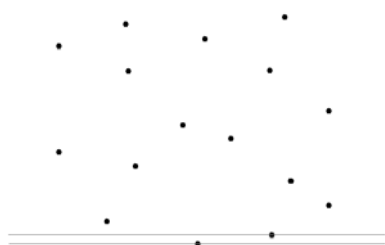


図 3.1: 最初の上辺と下辺

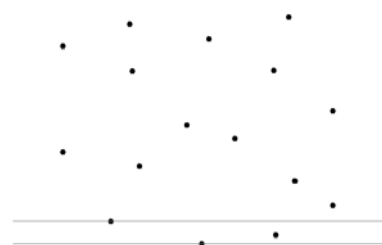


図 3.2: 次の上辺と下辺

上辺が一番上まで着たら，図 3.3 のように，下辺を一点分上に上げて同様に考えていく．この一点ずつ上げていく操作のため，最初にソートしておく必要がある．

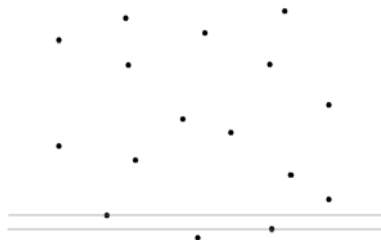


図 3.3: 上辺が一番上までいった後

また，上辺と下辺の間にある点は，別に記憶しておく．上辺をあげる際，前の上辺の点を x 軸方向にソートした状態となるよう追加していく．1 点目からすることで，わざわざソート関数を作らなくても，常にソートされた状態を保つことができる．

3.2.2 左辺・右辺の決定

上辺と下辺が決まったら，その間の点で左辺と右辺を考える．このとき，上辺と下辺を決めている点の x 座標が重要となる． x 座標が上辺と下辺の点の間にあるような点は，境界上の点になり得ないからである．

そのため，図 3.4 の 2 つの点線よりも左側にある点で左辺を，右側にある点で右辺を考えることになる．また，2 つの点線の中の点は，四角形の内部に含まれる点となる．

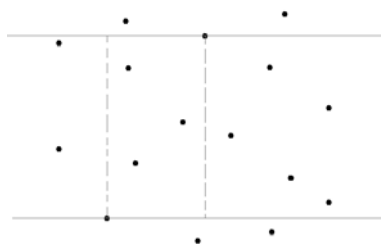


図 3.4: 上辺と下辺の点の x 座標による区分け

3.2.3 ディスクレパンシーの計算

左辺候補と右辺候補がわかったら，いよいよディスクレパンシーを計算する．まず，それぞれの点に対して，図 3.5，図 3.6 のような長方形を考え，それらのディスクレパンシー

を計算する．その後，左辺候補のうち最も小さいディスクレパンシーと，右辺候補のうち最も大きいディスクレパンシーを取り出し，最大ディスクレパンシーから最小ディスクレパンシーを引くことで，現在の上辺・下辺に対するディスクレパンシーが求められる．

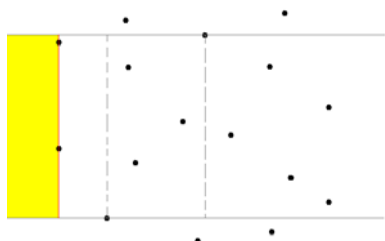


図 3.5: 左辺候補のディスクレパンシー

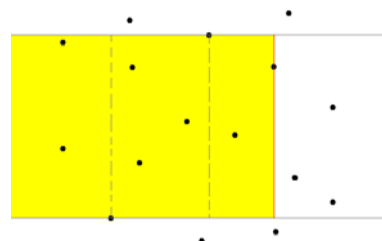


図 3.6: 右辺候補のディスクレパンシー

左辺候補の点において，ディスクレパンシーを Dl ，このときの長方形の面積を $|Rl|$ ，含まれる点の数を $\#Rl$ とする．このとき Dl は次のように表される．

$$Dl = \#Rl - n|Rl|/|R|$$

同様に右辺候補では

$$Dr = \#Rr - n|Rr|/|R|$$

と表される．これらより，図 3.7 のような面積 $|Rt|$ ，点数 $\#Rt$ の長方形のディスクレパンシー Dt は次のように計算できる．

$$\begin{aligned} Dt &= \#Rt - n|Rt|/|R| \\ &= (\#Rr - \#Rl + 1) - n(|Rr| - |Rl|)/|R| \\ &= \#Rr - \#Rl + 1 - n|Rr|/|R| + n|Rl|/|R| \\ &= (\#Rr - n|Rr|/|R|) - (\#Rl - n|Rl|/|R|) + 1 \\ &= Dr - Dl + 1 \end{aligned}$$

$\#Rt$ を展開したときにでる $+1$ は，左辺候補上の点のことである． x 座標が同じ点がないとわかっているときは $+1$ のままであるが，図 3.7 のように左辺候補と x 座標が同じ点がある場合は，その分も加える必要がある．

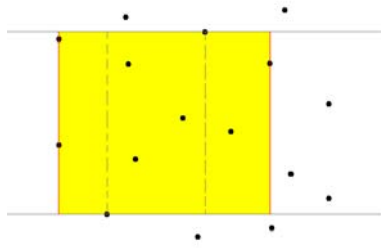


図 3.7: 左辺候補・右辺候補から計算できる四角形

以上より，上辺と下辺を決めたときのディスクレパンシーは，左辺候補の最小ディスクレパンシーと右辺候補の最大ディスクレパンシーとの差に1(もしくは左辺候補と x 座標が同じ点の数)を加えることで計算できる．

この計算を各上辺，下辺のパターンごとに行い，最大となるものが最終的なディスクレパンシーとなる．

3.2.4 計算時間

このアルゴリズムの計算時間は次のように求める．

まず，上辺・下辺の組み合わせを考える．これは2点対を考えるようなものなので $O(n^2)$ 通りである．続いて，それぞれの上辺，下辺ごとに，前回の上辺を決めていた点を順番になるよう入れていく．上辺と下辺の間の点を調べることになるので，各上辺，下辺ごとに $O(n)$ 時間かかることになる．この時点で $O(n^3)$ 時間かかっている．

実際にディスクレパンシーを計算するときだが，上辺と下辺の間にある点は x 軸方向にソートされているため，内部に含まれる点をいちいち調べなくとも計算で求めることができる．あとは面積を調べて計算するだけなので，全点のディスクレパンシー計算にかかる時間は $O(n)$ ですむ．

あとは定数時間でできるため，全体として計算時間は $O(n^3)$ 時間となる．

3.2.5 タイプ1，タイプ2の長方形による評価

2点，3点で考えるタイプ1，タイプ2の長方形も同様に計算することができる．

タイプ1の時は，上辺と下辺を決めたとき，その点でそのまま右辺と左辺も決めてしまえば，4点と同じ方法を使うことができる．

図3.8のように，上辺を決める点の下辺を決める点より右にある場合，上辺を決める点のまま右辺を決める点となる．この場合左辺は，下辺を決める点によって定まる．

図3.9のように，上辺を決める点の下辺を決める点より左にある場合，上辺を決める点のまま左辺を決める点となる．この場合右辺は，下辺を決める点によって定まる．

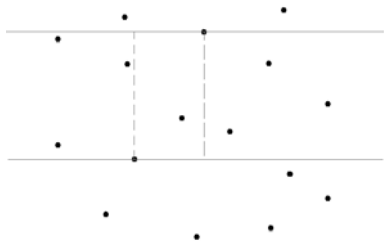


図 3.8: 上辺が右边を決める場合

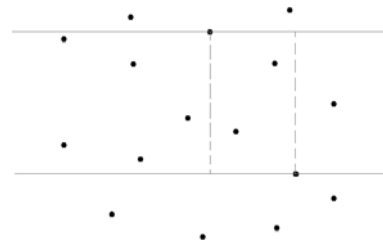


図 3.9: 上辺が左边を決める場合

タイプ 2 の時は，上辺と下辺を決めたとき，まず図 3.10，図 3.11 のように，もう一点で右边を決めて，左边を上辺と下辺を決めた点で定める．

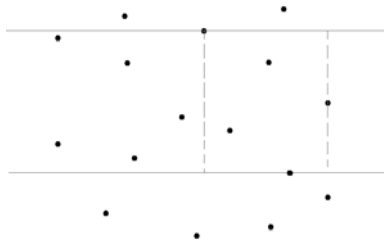


図 3.10: 上辺が左边を決める場合

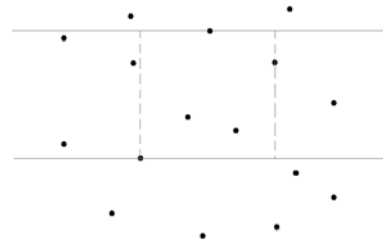


図 3.11: 下辺が左边を決める場合

その後，また最初から上辺と下辺を決めなおし，今度は図 3.12，図 3.13 のように，右边を上辺と下辺を決めた点で定め，もう一点で左边を決める．

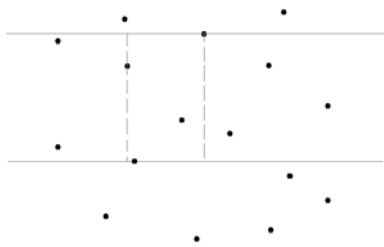


図 3.12: 上辺が右边を決める場合

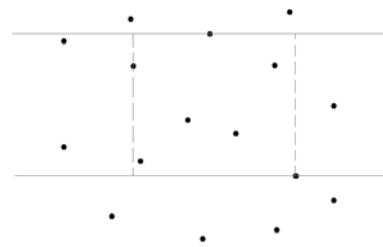


図 3.13: 下辺が右边を決める場合

つまり，タイプ 2 の長方形を考えるときは，3 点目で右边を決める周と，3 点目で左边を決める周の 2 周する必要がある．

第4章 点集合の種類とディスクレパンシー

本研究では，アルゴリズムを提案するだけでなく，それを使ってさまざまな点集合のディスクレパンシーの計算を行った．この章では，今回用いた点集合と計算結果を示す．

4.1 ランダム点集合

ランダム点集合は，その名の通り，各点の x 座標， y 座標をランダムで設定した点集合である．

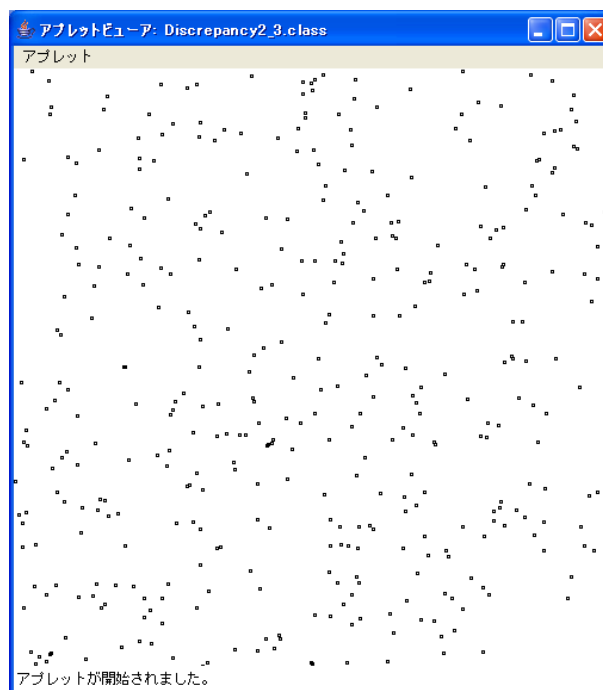


図 4.1: 400 点でのランダム点集合

図 4.1 は，Java のランダム関数を使って 400 点で生成したランダム点集合である．

4.2 格子状点集合

格子状点集合は、結ぶと格子状になるよう配置された点集合である。

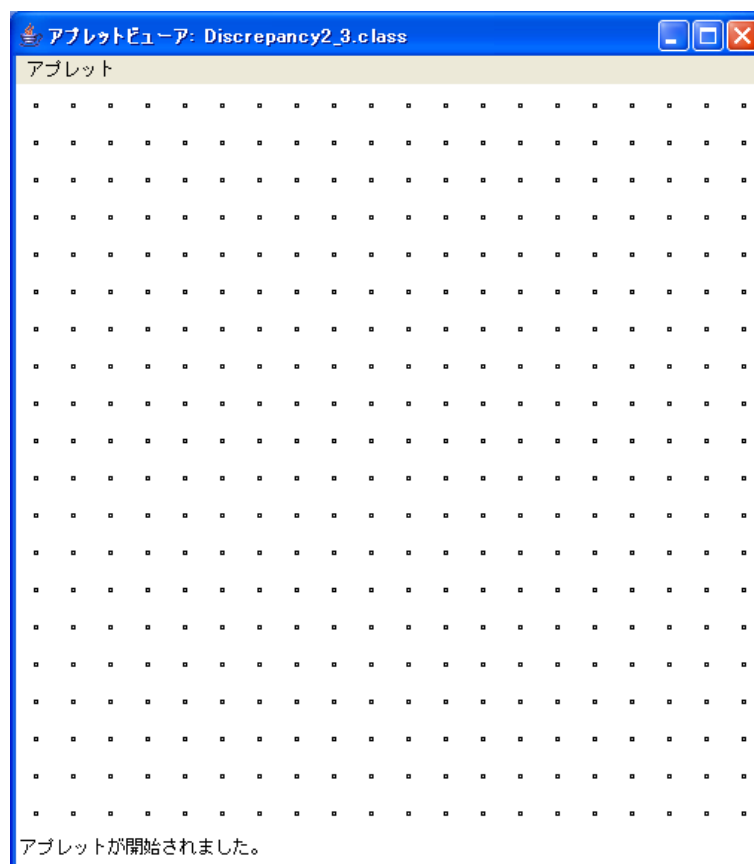


図 4.2: 400 点での格子状点集合

図 4.2 は、400 点で生成した格子状点集合である。

一般に格子状点集合のディスキレパンシーはよくないとされる。

4.3 格子状を斜めにした点集合

格子状を斜めにした点集合は、各点が $(i/n, [(i \times \text{無理数}) \text{ の小数部分}])$ で定義される点集合である。

n, i は「全 n 点のうちの i 番目の点」という意味である。また実際に生成するときは、 x 座標、 y 座標に領域における x 軸、 y 軸方向の大きさをかける。

図 4.3 は、無理数を $\sqrt{5}$ として 400 点で生成した点集合である。

見た目は格子状点集合を傾けただけであるが、ディスキレパンシーはかなり優れているとされる。

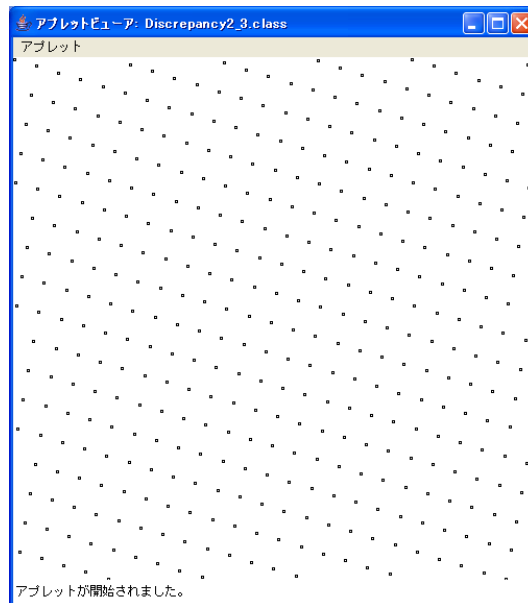


図 4.3: 400 点での格子状を斜めにした点集合

4.4 Van der Corput 集合

Van der Corput 集合は、各点が $(i/n, r(i))$ で定義される点集合である。

n, i は「全 n 点のうちの i 番目の点」という意味である。また実際に生成するときは、 x 座標、 y 座標に領域における x 軸、 y 軸方向の大きさをかける。

$r(i)$ とは、 i の値を 2 進数表現し、逆から表記したものを小数点以下に置くという動作をする。例として、 $i=13$ のとき $(13)_{10}=(1101)_2$ となり、逆から表記して小数点以下に置くと $(0.1011)_2=(0.6875)_{10}$ となる。つまり、 $r(13)=0.6875$ となる。

この操作は、図 4.4、図 4.5 のような操作をしていることになる。

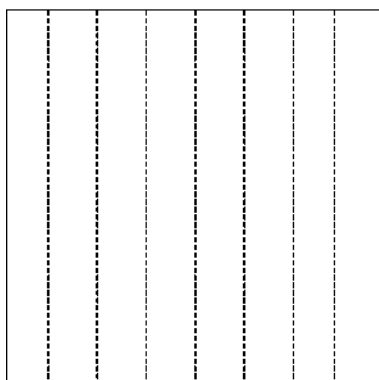


図 4.4: x 軸方向に n 分割

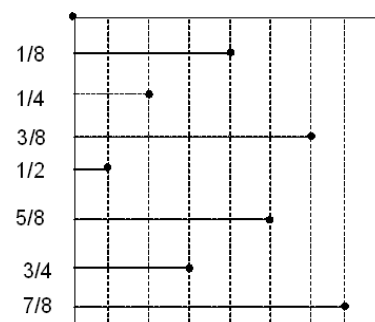


図 4.5: 点線上で点を中央においていく

このような方法で得られる点集合のディスクレパンシーは、点数 n に対して $O(\log n)$ となる。

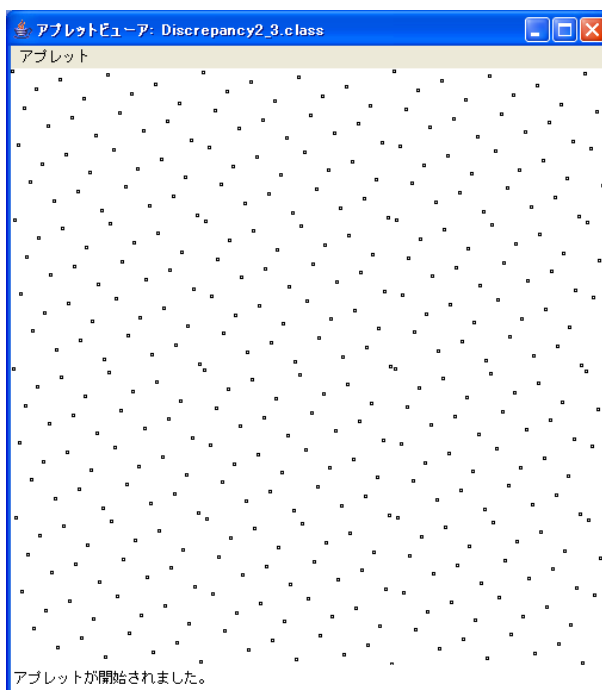


図 4.6: 400 点での Van der Corput 点集合

図 4.6 は、400 点で生成した Van der Corput 集合である。

4.4.1 ディスクレパンシーの証明

まず、図 4.7 のような $[k/2^q, (k+1)/2^q)$, $0 \leq k < 2^q$ を考える。以後、本論文ではこれを正規区間と呼ぶ。

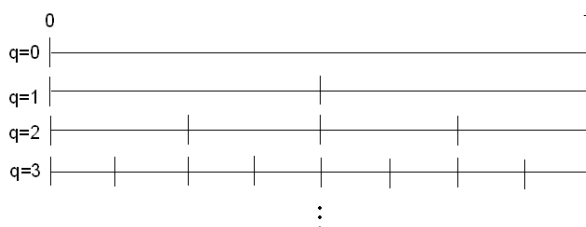


図 4.7: 正規区間

これを使い、区間 $[0, 1)$, $[0, 1/2)$, $[1/2, 1)$, $[0, 1/4)$, $[1/4, 1/2)$, $[1/2, 3/4)$, $[3/4, 1)$, $[0, 1/8)$, $[1/8, 1/4)$... の一つとする。

続いて、2つの主張を証明し、その2つからディスクレパンシーが $O(\log n)$ であることを証明する。

主張 1

正規区間 I と $a \in (0, 1]$ による、 $[0, a) \times I$ のあらゆる長方形 R に対して、 $|D(P, R)| \leq 1$ が成り立つ。

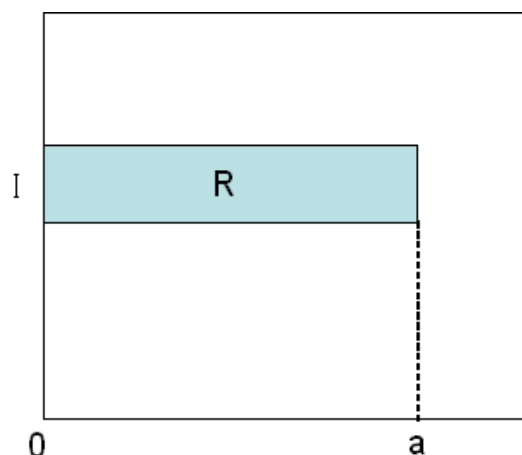


図 4.8: 主張 1

$I = [k/2^q, (k+1)/2^q)$ とする。

このとき、 y 座標に I がある点 $p_i \in P$ は、それぞれ $r(i) \in I$ をもつ (P : 点集合)。また、 I の範囲より $r(i)$ の最初の 2 進数の q 桁が固定されている。このとき、定数 $\bar{k}$ において $i \equiv \bar{k} \pmod{2^q}$ となり、点の x 座標は $2^q/n$ 間隔で規則的に配置される。

そこで、図 4.9 のように $2^q/n$ 間隔で細分すると、それぞれの長方形が領域 $1/n$ をもつことになる。点は全部で n 点あるため、一様ならばそれぞれの長方形に点が 1 つずつあることになる。よって、それぞれの長方形に点が 1 つあればディスクレパンシーは 0、点がなければディスクレパンシーは 1 となる。図 4.9 の色がついた領域において、0 ディスクレパンシーと 1 ディスクレパンシーに分かれるため、領域全体でのディスクレパンシーは高々 1 となる。

以上より、 $|D(P, R)| \leq 1$ が成り立つ。



図 4.9: 主張 1 の証明

主張 1 の例を図 4.10 に示す。

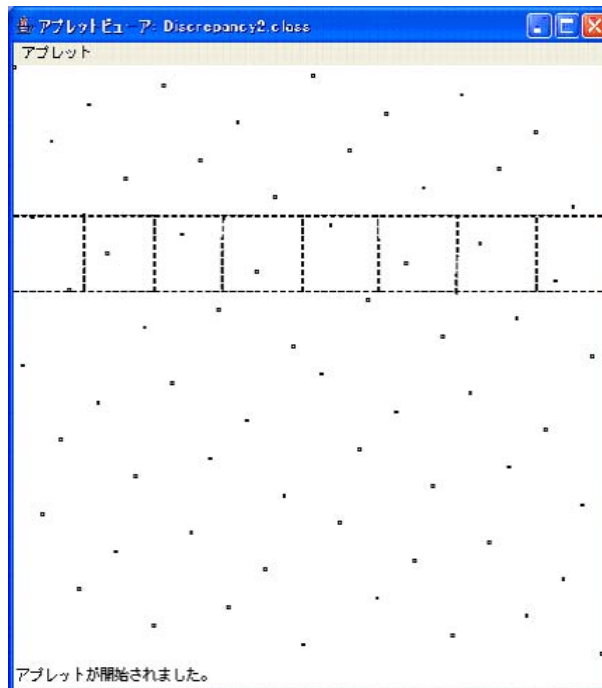


図 4.10: 主張 1 の例 ($q=3$)

主張 2

どんな角 $C(x,y)$ でも, 最大 $\lceil \log_2 n \rceil$ の直和として表現できる .
 (長方形は, 主張 1 に加え $|D(P, R)| \leq 1$ の集合 M となる)

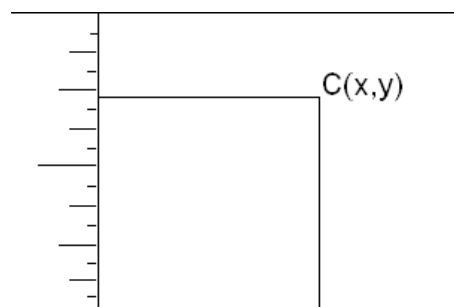


図 4.11: 主張 2

まず, 次のように定義する .

- $m: 2^m \geq n$ となる最も小さい整数
- $y_0: y$ を超えない最も大きい 2^{-m} の整数の倍数
- M : 長方形 $[0, x) \times [y_0, y)$

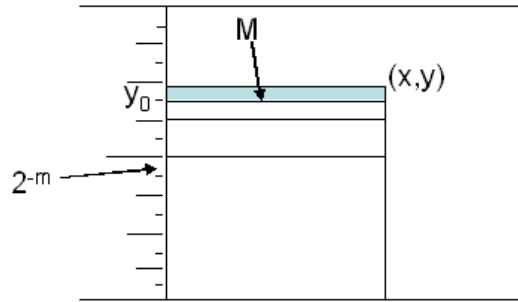


図 4.12: 主張 2 の証明

このとき， M の領域は多くとも $y - y_0 < 2^{-m} \leq 1/n$ (ただし， $2^m \geq n$) となる．また，図 4.12 において， M は多くとも P の点を 1 つだけ含んでいる．点の y 座標は，横のメモリ上に存在しており， y 座標が同じ点は存在しないので， P の二つの点の y 座標は少なくとも 2^{-m} 違うことになる．

すなわち，長方形 M に対して $|D(P, M)| \leq 1$ が成り立つ．

また， $[0, y_0)$ は図 4.13 のように，最大 m の正規区間に分割できる．

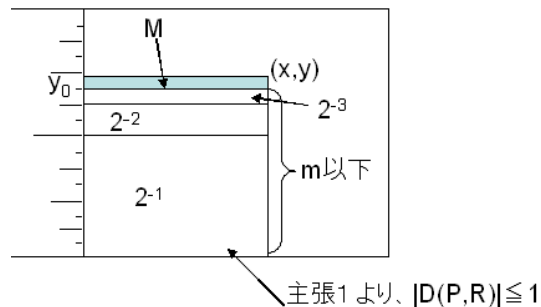


図 4.13: 正規区間に分割

以上より，どんな角 $C(x, y)$ でも，最大 $\lceil \log_2 n \rceil$ の直和として表現することができる．

主張 1, 2 より，角における Van der Corput 集合のディスクレパンシーは最大で $\log_2 n + 2$ となる．

4.5 点数とディスクレパンシー

続いて，それぞれの点集合において，点の数を変えつつディスクレパンシーを計算した結果を示す．

まず，図 4.14 にランダムで配置した点集合での，点数とディスクレパンシーの関係を示す．

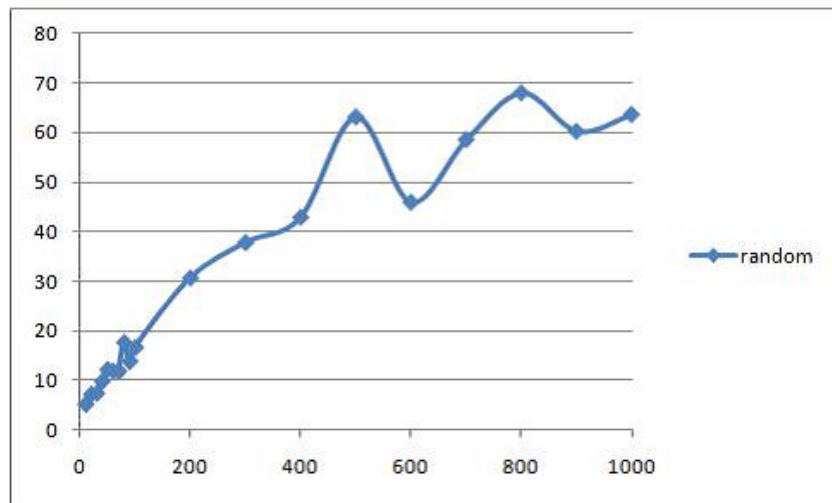


図 4.14: ランダム点集合における点数とディスクリパンシーの関係

図 4.14 のディスクリパンシー計算は、同じ点数で 5 回ランダム配置を行い、それらの平均を取っている。かなりムラがあるものの、点数に比例してディスクリパンシーが大きくなっていることがわかる。

次に、図 4.15 に格子状に配置した点集合での、点数とディスクリパンシーの関係を示す。

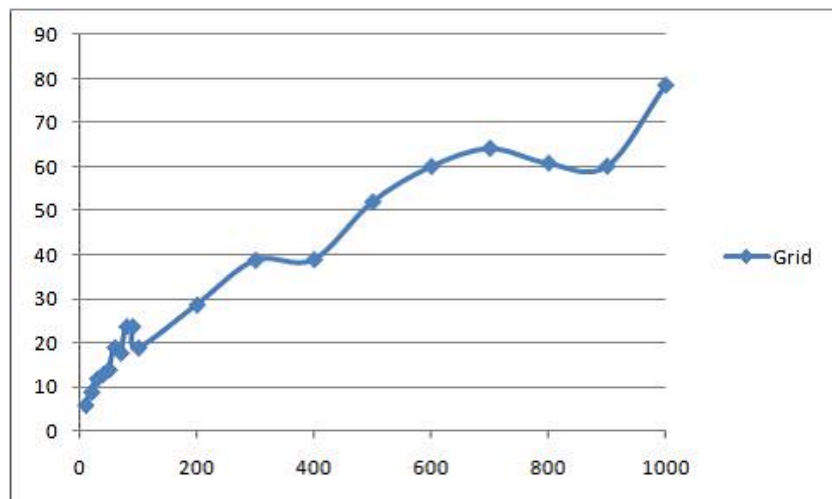


図 4.15: 格子状点集合における点数とディスクリパンシーの関係

点数の 2 乗根が整数となる部分は若干小さくなるものの、やはり点数とともにディスクリパンシーは大きくなっている。

続いて、図 4.16 に格子状に配置した点を斜めに傾けた点集合での、点数とディスクリパンシーの関係を示す。

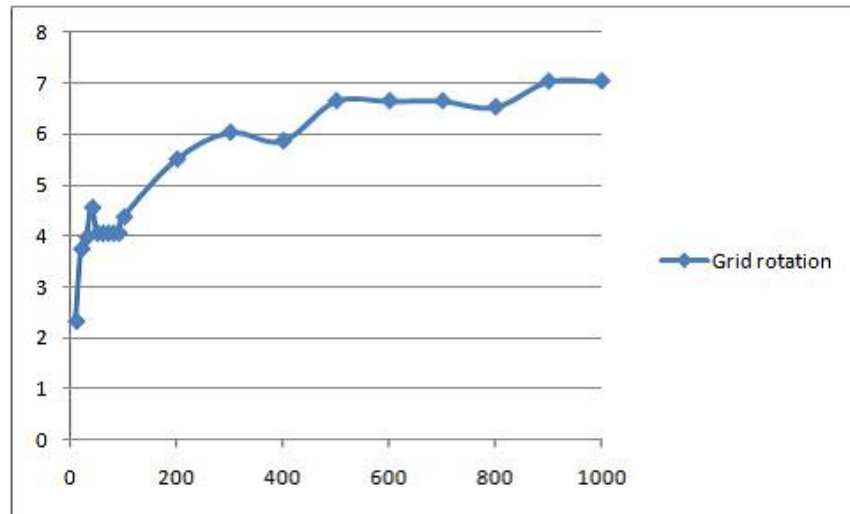


図 4.16: 格子状点集合を傾けた点集合における点数とディスクレパンシーの関係

最初はディスクレパンシーが一気に増えるが，点数が増えるとディスクレパンシーの変化は小さくなり，500 を超えるとほとんど変化しなくなる．

続いて，図 4.17 に Van der Corput 集合での，点数とディスクレパンシーの関係を示す．

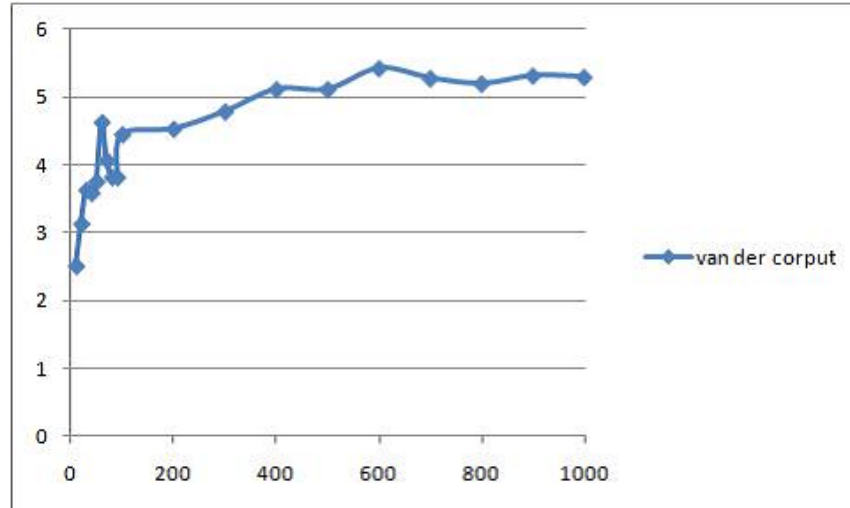


図 4.17: Van der Corput 集合における点数とディスクレパンシーの関係

格子状点集合を傾けた点集合と同様，最初は一気に増えるが，点数がある程度増えるとほとんど変化しなくなる．

また，図 4.18 から，Van der Corput 集合のディスクレパンシーは $O(\log n)$ で抑えられていることがわかる．

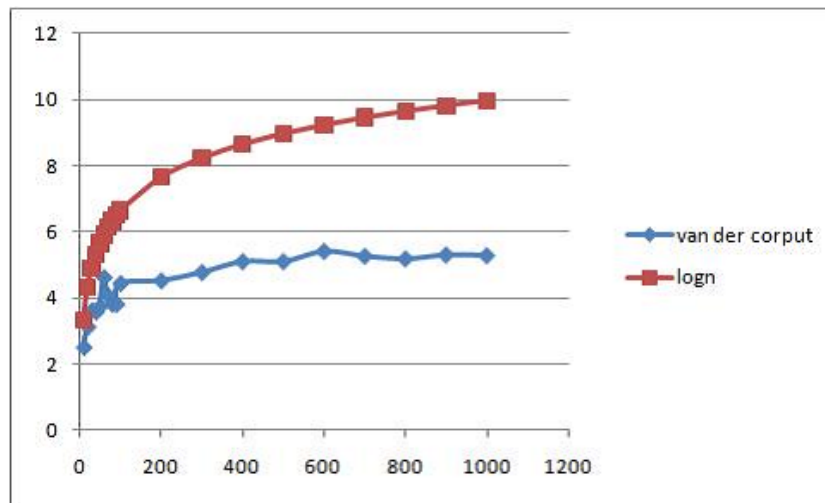


図 4.18: $\log_2 n$ と Van der Corput 集合のディスクリパンシー

最後に，4 つの点集合のディスクリパンシーを一つにまとめたものを図 4.19 に示す．

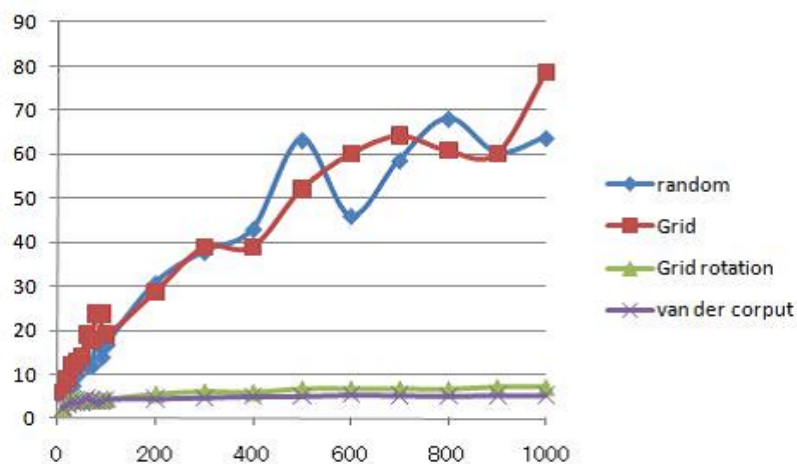


図 4.19: 全部

この図より，ランダム点集合・格子状点集合に比べ，格子状を斜めにした点集合・Van der Corput 集合のほうがディスクリパンシーが小さいことがわかる．また，ランダム点集合と格子状点集合では，ディスクリパンシーの値があまり変わらず，格子状を斜めにした点集合と Van der Corput 集合では，Van der Corput 集合のほうが若干ディスクリパンシーが小さいことがわかる．

第5章 まとめ

今回、2次元上の点集合におけるディスクレパンシーを、四角形を考える方法で、 $O(n^3)$ 時間で計算できるアルゴリズムを開発した。

また、このアルゴリズムを使い、ランダム点集合・格子状点集合・格子状を斜めにした点集合・Van der Corput 集合の4種類の点集合に対して、点数とディスクレパンシーの関係を実験的に求めた。

その結果、ランダム点集合と格子状点集合はディスクレパンシーがあまり変わらないこと、それらに比べ、格子状を斜めにした点集合と Van der Corput 集合のディスクレパンシーが非常に小さいこと、格子状を斜めにした点集合と Van der Corput 集合では、Van der Corput 集合の方がわずかにディスクレパンシーが小さいが、ほとんど変わらないことがわかった。

以上より、ランダム点集合は、ディスクレパンシーがあまりよくないとされる格子状点集合と比べても、一様性はあまり変わらないことがわかった。また、Van der Corput 集合や格子状を斜めにした点集合は、ランダム点集合などと比べ、非常に一様性が高いこともわかった。さらに点数が多いと、各点集合におけるディスクレパンシーの差が大きくなるため、点数が多ければ多いほど、ディスクレパンシーの小さい点集合を作るアルゴリズムの重要性が高まってくることがわかった。

第6章 謝辞

本研究を行うにあたり，終始丁寧な指導と適切な助言を下された浅野哲夫教授に心より感謝いたします．また，ゼミなどでお世話になりました上原隆平准教授，清見礼助教に厚く御礼申し上げます．浅野研究室，上原研究室の皆様には公私ともに大変お世話になりました．ここに感謝の意を表します．

参考文献

- [1] *Jiří Matoušek* , Geometric Discrepancy , Springer , Algorithms and Combinatorics 18 , 1999 .
- [2] D . L . Lau , G . R . Arce , and N . C . Gallaghe , Digital halftoning via green noise masks Proc . IEEE 86 (1998) 2424 .
- [3] T . Mitsa and K . J . Parker , Digital halftoning technique using blue-noise mask , J . Opt . Soc . Am . A , 9 (1992) 1920 .
- [4] 手塚集 , 「超一様分布列の数理」 , 岩波書店「統計科学のフロンティア」シリーズの第 11 巻「計算統計 I」 . 2003 .