2004－AL－94　(4)
2004／3／19

ガウスの掘割り問題に対する数値計算とその考察

土村 展之*

「ガウス素数をある有限の歩幅で飛び石のように飛び, 無限遠までたどれるか.」この未解決問題に対して, 本研究では計算機を用いて様子を探るアプローチを採った. 原点から到達できる最遠点を見つけるための, 並列計算も可能な効率のよい手法を提案し, 幅 $k = \sqrt{36}$ の堀割りの存在を示した. これは Gethner らの記録 $k = \sqrt{26}$ を塗り替えるものである. さらに, 到着できる最遠点までの距離の見積りについて, パーコレーション理論に基づく Vardi のモデルを, ガウス素数の分布に関する性質を考慮して修正し, より精密に見積もることに成功した.

Computational Results for Gaussian Moat Problem

Nobuyuki Tsuchimura

"Can one walk to infinity on Gaussian primes taking steps of bounded length?" We adopted computational techniques to probe into this open problem. We propose an efficient method to search for the farthest point reachable from the origin, which can be parallelized easily, and have confirmed the existence of a moat of width $k = \sqrt{36}$, whereas the best previous result was $k = \sqrt{26}$ due to Gethner et al. A refinement of Vardi's estimate for the farthest distance reachable from the origin is proposed. The proposed estimate incorporates discreteness into Vardi's that is based on percolation theory.

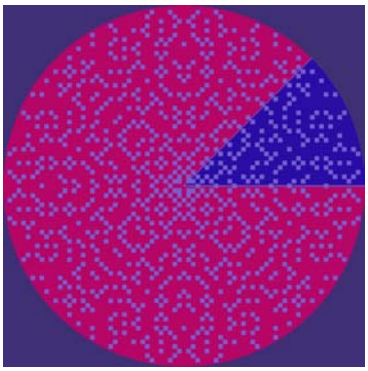
1　はじめに

ガウス素数を飛び石と見なして, 有限の足の長さの人が無限遠方まで歩けるか. 正確に定義すると, 次のように表せる. ガウス整数とは, 整数 a, b を用いて $a + bi$（i は虚数単位）と書ける複素数である. ガウス素数とは, ガウス整数のうちで自明でないガウス整数の積（つまり $\pm 1, \pm i$ を使わない）に分解できないものである. そして複素平面上で次のようなグラフ G を考える. グラフの点集合はガウス素数であり, 2 点間の距離が歩幅 k 以下であれば連結であるとして枝を付け加える. 図 1(b) は歩幅 $k = 16$ の場合の原点からの連結成分である. 問題は, ある固定された歩幅 k に対して, グラフが原点から無限遠までつながっているかどうか, ということである. もし無限遠までのパスが存在しないなら, 原点のまわりには幅 k の堀割り存在することになる.

この問題は 1962 年 Basil Gordon によって提起されたが, まだ未解決として残っている [1][2][4][5]. 文献 [1][2][4] では否定的な予想が立てられている.

計算機による先行研究はいくつかあるが, まとめると次の表のようになる [1][5]. $\xi(k)$ は原点から 歩幅 k で到達できる一番遠い点を表すものとする.

* 東京大学 大学院情報理工学系研究科 数理情報学専攻
Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo
http://www.misojiro.t.u-tokyo.ac.jp/~tutimura/
tutimura@mist.i.u-tokyo.ac.jp



(a) ガウス素数 $|a+bi| \leq 39$ と
最初の八分円 $0 \leq b \leq a$

(b) 歩幅 $k = \sqrt{16}$ の場合の
原点から連結なガウス素数

図 1: ガウス素数

歩幅 k	最遠点 $\xi(k)$	最遠点までの 距離 $\lvert\xi(k)\rvert$	原点から連結する ガウス素数の個数
$\sqrt{1}$	$2+i$	2.23	2
$\sqrt{2}$	$11+4i$	11.70	14
$\sqrt{4}$	$42+17i$	45.31	92
$\sqrt{8}$	$84+41i$	93.47	380
$\sqrt{10}$	$976+311i$	1024.35	31221
$\sqrt{16}$	$3297+2780i$	4312.61	347638
$\sqrt{18}$	$8174+6981i$	10749.4	2386129
$\sqrt{20}$	$109677+64268i$	127120	Finite
$\sqrt{26}$	??	$\leq$ 5586757	Finite

例えば 3 行めは, 歩幅 $k = \sqrt{4}$ で原点から 45.31 離れた $42+17i$ まで到達可能であり, 最初の八分円 $\left\{z \in \mathbf{C} \;\middle|\; 0 \leq \arg z \leq \pi/4\right\}$ の中に 92 個の到達可能なガウス素数があることを示す. (ガウス素数は対称に存在するので, 1/8 の領域を考えるだけで十分である.) 歩幅 $k = \sqrt{26}$ については, 最遠点までの距離の上界 5586757 が知られているが, 具体的な最遠点と連結するガウス素数の個数は不明である. 歩幅 $k = \sqrt{20}$ では, 最遠点がわかっているので, 連結するガウス素数の個数は不明ながらも有限と考えられる.

本研究では, 上の表の最後の 2 行を完成させ, さらに 3 行を付け加えた.

歩幅 k	最遠点 $\xi(k)$	最遠点までの 距離 $\lvert\xi(k)\rvert$	原点から連結する ガウス素数の個数
$\sqrt{20}$	$120510+57857i$	133679.065	273791623
$\sqrt{26}$	$943460+376039i$	1015638.765	14542615005
$\sqrt{32}$	$2106442+1879505i$	2823054.542	103711268594
$\sqrt{34}$	??	$<$ 25051948	Finite
$\sqrt{36}$	??	$<$ 90011601	Finite

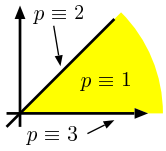
2 ガウス素数の生成法

ガウス整数とは, 実部虚部ともに整数の複素数である. ガウス素数とは, ガウス整数 $a+bi$ のうちで $\pm 1, \pm i, \pm(a+bi), \pm(b-ai)$ 以外のガウス整数の積として表せないものを言う. $a+bi$ がガウス素数であるならば, $\pm a \pm bi$ や $\pm b \pm ai$ もガウス素数である. 従って, 以下では $0 \leq b \leq a$ の領域に限って議論する.

ガウス素数を求める方法はいくつかある. エラトステネスの篩をガウス素数に拡張することもできるが, 必

要な記憶領域が大きいので, 大きなガウス素数を求めるのには向かない. 本研究では 2 つの方法を試みたが, いずれも以下のよく知られた性質 [1][3] を利用する. p を普通の素数であるとする. もし $p \equiv 3 \pmod 4$ ならば $p+0i$ はガウス素数である. もし $p \equiv 1 \pmod 4$ ならば $a^2+b^2=p(0 \le b \le a)$ となる整数 a, b がただ一組存在して, しかも $a+bi$ がガウス素数である. これの逆も成り立つ.

普通の素数 p	$\pmod 4$	ガウス素数 $z=a+bi$		
$p=5,13,\ldots$	$p \equiv 1$	$a^2+b^2=p$	$0<b<a$	$\lvert z\rvert^2=p$
$p=2$	$p \equiv 2$	$a=b=1$	$0<b=a$	$\lvert z\rvert^2=p$
$p=3,7,11,\ldots$	$p \equiv 3$	$a=p, b=0$	$0=b<a$	$\lvert z\rvert=p$



最初の方法は, ガウス整数 $a+bi(0 \le b \le a)$ の素数性をテストするものである.

- もし $b \neq 0$, $a^2+b^2 \equiv 1 \pmod 4$ で a^2+b^2 が普通の素数ならば, $a+bi$ はガウス素数である.
- もし $b=1$, $a=1$ ならば $a+bi$ はガウス素数である.
- もし $b=0$, $a \equiv 3 \pmod 4$ で a が普通の素数なら $a+bi$ はガウス素数である.
- これ以外なら $a+bi$ はガウス素数でない.

この方法は 3.1 節の Gethner の方法とともに用いた. 近年, 整数 n の素数判定が多項式時間 $O(\log^{12+\epsilon} n)$ で行えることが証明された [8] が, 現在のところ, Miller-Rabin テスト [9] のような確率的アルゴリズムのほうが実用上高速である.

もう一つの方法は, 普通の素数 p からガウス素数を生成するものである. これは 3.2 節の提案手法で用いた.

- もし $p \equiv 1 \pmod 4$ なら p を二つの整数の 2 乗和 $p=a^2+b^2$ に分解して $a+bi$ を得る.
- もし $p \equiv 2 \pmod 4$ つまり $p=2$ なら $1+i$ を得る.
- もし $p \equiv 3 \pmod 4$ なら $p+0i$ を得る.

$p \equiv 1$ の場合の 2 乗和への分解は, 次のような効率的な方法がある [3]. x を $x^2 \equiv -1 \pmod p$ を満たす整数とする. このような x は, p を法とする最小の平方非剰余を r として $x \equiv r^{(p-1)/4} \pmod p$ を計算することで求められる. そして p と x のペアに対してユークリッドの互除法を適用したときの, $\sqrt{p}$ よりも小さい最初の余りを a とする. このとき $b=\sqrt{p-a^2}$ とすると b は整数となり, $p=a^2+b^2$ の分解が得られたことになる.

3 計算手法

原点からの連結成分のうち最も遠い点を計算するために, 本研究では二つの手法を試みた. 一つは Gethner の手法 [1] であり, もう一つが本研究で提案する方法である. その結果, 提案手法の方が 10 倍〜20 倍高速であることがわかった.

3.1 Gethner の手法

Gethner の手法は, 1 節で導入したグラフ上での幅優先探索に相当する. ガウス素数を原点から到達できる最小の歩数で分類する. つまり, レベル n のガウス素数とは, 原点から n 歩で到達できるものであるとする. レベル n のガウス素数から歩幅 k で到達できるガウス素数は, レベル $n-1$, n, $n+1$ のいずれかに属する. 従って, レベル $n-1$, n のガウス素数だけを記憶しておけば, $n+1$ のものを得ることができる.

レベル n 以下のガウス素数がすべて求められているとする. レベル n のガウス素数それぞれについて, 距離 k 以下のところにあるガウス整数を見つける. 見つけたガウス整数からガウス素数を選び出し, 更にレベル

表 1: 二つの手法の比較

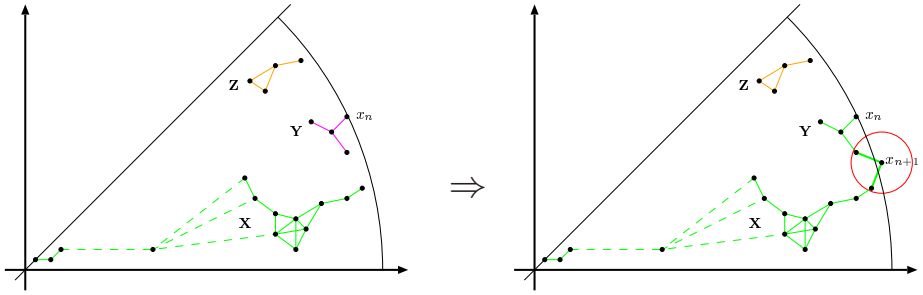
	Gethner の手法	提案手法
基本アイデア	グラフ G 上の幅優先探索	サブグラフの逐次構成
ガウス素数の生成順序	ランダム	絶対値順
適する ガウス素数の生成法	素数テスト	エラトステネスの篩 ＋ 2 乗和分解
生成する ガウス素数の数	原点からの連結成分のみ (右の欄の 50 ～ 75%)	絶対値が $\lvert\xi(k)\rvert + k$ 以下のものすべて

$n-1$ と n に属するものを除くと, レベル $n+1$ のものが残る.

この方法ではガウス整数に対する素数判定を行う必要があるが, 2 節で述べたように普通の整数に対する素数判定ができればよい. エラトステネスの篩は高速ではあるが, ランダムな順序で現れる整数に対して判定を行うには多くの素数を保持する記憶領域が必要となるので, Miller-Rabin テストと Lucas-Lehmer テストを組み合わせた実装 `java.math.BigInteger.isProbablePrime()` を用いた.

3.2 提案手法

本研究では, 2 節で述べた方法により, 絶対値順に生成したガウス素数を利用する. 1 節で導入したグラフ G の点を, 対応するガウス素数の絶対値の順に番号づけしておく. そしてガウス素数を順に生成しながら, それまでに生成した点に制限した G の部分グラフを作って行く. これは次のようにすると効率的である.



G_n
連結成分 X, Y, Z はつながっていない.

G_{n+1}
連結成分 X と Y が x_{n+1} でつながった.

図 2: 部分グラフ G_n と G_{n+1}

$x_1, x_2, \ldots, x_n, x_{n+1}, \ldots$ を絶対値順に並べたガウス素数とする. G_n を $\{x_1, x_2, \ldots, x_n\}$ に制限した G の部分グラフとする. そしてグラフの連結成分を (枝の向きが逆の) 根付木で表現する. つまり各成分からは, 連結成分の中で最大の絶対値をもつガウス素数への有向パスを持つようにする (図 3). 従って部分グラフ G_n は根付木の集合として表されることになる. この根付木の集合はポインタの配列で表現する.

新しいガウス素数 x_{n+1} を付け加えるには, まず $|x_{n+1} - x_i| \le k$ となるような $x_i (i \le n)$ を探す. もしそのような x_i がなければ, x_{n+1} は G_{n+1} 上の大きさ 1 の連結成分となる. そのような x_i が複数個見つかれば, $x_{i_1}, x_{i_2}, \ldots, x_{i_m}$ とする. G_n における $x_{i_j} (j = 1, 2, \ldots, m)$ の連結成分の中で絶対値が最大のものを r_j とする. そして r_j から x_{n+1} へのポインタを付け加える $(j = 1, 2, \ldots, m)$. これは G_n 上の $r_1, r_2, \ldots, r_m$ を含む連結成分を, G_{n+1} 上では x_{n+1} を含む一つの連結成分にしていることになる.

実際の計算では, $|x_i| < |x_n| - k$ となる x_i についてのポインタは更新されることはないので, 忘れてしまっ

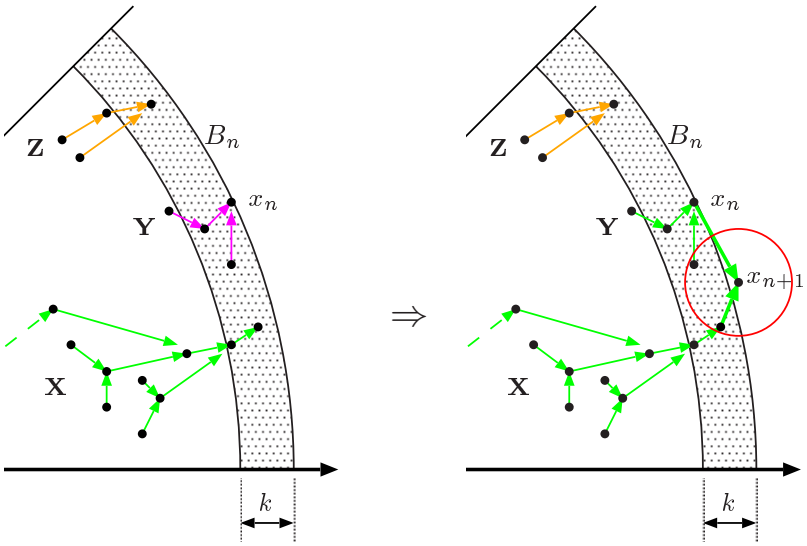


図 3: G_n と G_{n+1} を表す根付木と扇形領域 B_n

てよい. つまり, 図 3 で表される $B_n = \left\{ z \in \mathbf{C} \;\middle|\; |x_n| - k \le |z| \le |x_n|, 0 \le \arg z \le \pi/4 \right\}$ の扇形領域のガウス素数についてのみ記憶すればよい.

もしも原点からの連結成分のうち, 絶対値が最大のもの r_0 が, この扇形領域からなくなってしまえば, つまり $|r_0| < |x_n| - k$ となれば, 手続きは終了である. そして原点から歩幅 k で到達し得る最遠点 $\xi(k)$ が r_0 であるとわかる.

最遠点までの距離 $|\xi(k)|$ の上界は, この手続きを少し変更するだけで求められる. あるガウス素数 x_m を定め, 原点ではなく $x_1, x_2, \ldots, x_m$ のいずれかに連結なガウス素数を求めるのである. それには扇形領域 B_m にあるガウス素数を生成し, これらが原点からの連結成分だと思ってポインタをつなげおいて, 上の手続きを x_{m+1} 以降について実行すればよい. $|x_m|$ を十分大きなものに選べば, 経験的に探索は速く終わり, $|\xi(k)|$ の (緩い) 上界が簡単に求まる.

4 計算結果

3.2 節の提案手法により計算実験を行い, 二つの結果を得た. 第一の結果は, 原点から歩幅 $k = \sqrt{32}$ で $2106442 + 1879505i$ まで到達できることがわかったことである. この点は 2823054.542 の距離にある. この結果を得るために, ガウス素数を 138994584350 個生成した. 原点から連結するガウス素数は 103711268594 個であり, 生成したものの 75%に相当する. 図 3 の扇形領域に含まれるガウス素数の個数は, 最大でおよそ 540000 である. これらを記憶するするのに 16Mbytes の領域が必要であった. 計算には 38CPU の並列計算で 80 時間かかった. 一台の計算機に換算するとおよそ 70 日に相当する.

第二の結果は, 歩幅 $k = \sqrt{36}$ では無限遠方まで到達できないということである. 原点から 90000000 以内の距離にあるガウス素数からの連結成分は, どれも原点から 90011601 の距離まで達することができなかった.

計算を行ったのは, ギガビットイーサでつないだ 19 台のコンピュータで, 次のような仕様である. Intel Pentium III 1.4GHz x 2 (SMP), 1GByte memory, Red Hat Linux 7.1 (kernel 2.4.18), Java J2SE 1.4.2, Java HotSpot Server VM, gcc 2.96.

歩幅 k	最遠点 $\xi(k)$	最遠点までの距離 $\lvert\xi(k)\rvert$	原点から連結するガウス素数の個数	計算時間 (1CPU/38CPU)
$\sqrt{1}$	$2+i$	2.236	2	
$\sqrt{2}$	$11+4i$	11.705	14	
$\sqrt{4}$	$42+17i$	45.310	92	
$\sqrt{8}$	$84+41i$	93.472	380	
$\sqrt{10}$	$976+311i$	1024.352	31221	
$\sqrt{16}$	$3297+2780i$	4312.610	347638	5sec/ —
$\sqrt{18}$	$8174+6981i$	10749.355	2386129	25sec/ —
$\sqrt{20}$	$120510+57857i$	133679.065	273791623	4hour/ 8min
$\sqrt{26}$	$943460+376039i$	1015638.765	14542615005	10day/11hour
$\sqrt{32}$	$2106442+1879505i$	2823054.542	103711268594	— /80hour
$\sqrt{34}$	—	<25051948	Finite	— /24hour
$\sqrt{36}$	—	<90011601	Finite	— /21hour

並列化は次のようにして行った. 計算の作業は主に二つに分けられる. 一つはガウス素数を絶対値順に生成することであり, もう一つが部分グラフ G_n を生成してガウス素数の連結性を調べることである. 計算量は前者が後者よりもはるかに多い.

連結性を調べる過程からは何の影響も受けないため, ガウス素数の生成は容易に並列化できる. 図 4 が役割分担を示している. machine1 が連結性を調べ, machine2 がほかの machine にガウス素数の生成作業を割り振っている. それぞれの machine には, 生成すべきガウス素数の絶対値の区間が与えられる. ガウス素数の生成速度は machine の数だけ速くなる. 全体としては, 連結性のテストがボトルネックになる程度までに効果があり, 10〜20 倍程度の高速化を達成した.

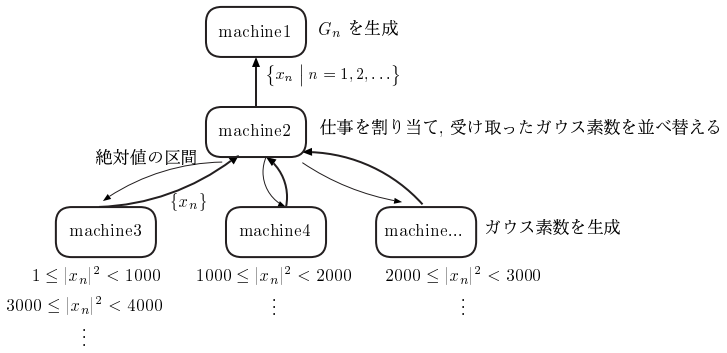


図 4: 分散計算における作業分担

5 到達できる最遠点までの距離の見積り

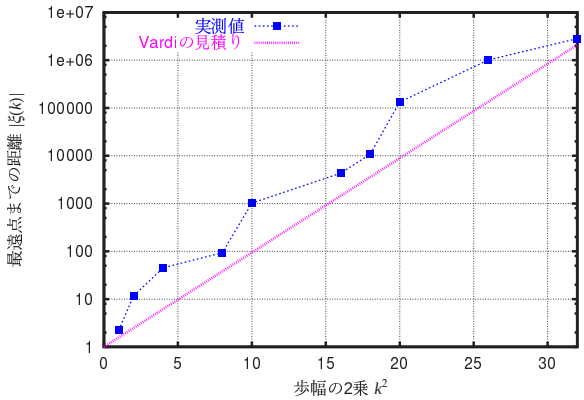
パーコレーション理論に基づき, Vardi はガウス素数のモデルを作り, 次のように予想した [4]:

$$k \sim \sqrt{2\pi\lambda_c \log|\xi(k)|}.$$

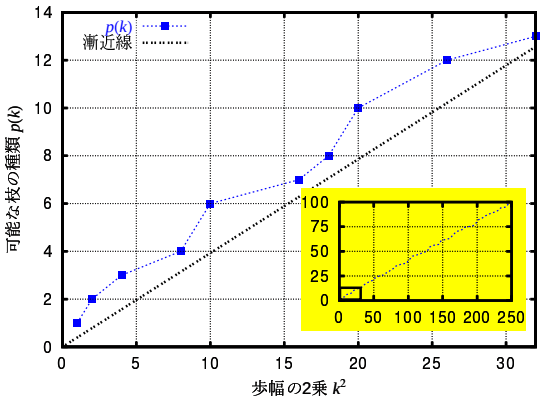
ここで $\lambda_c \approx 0.35$ は連続パーコレーションの定数であり, $\xi(k)$ は原点から歩幅 k で到達できる最遠点である. これは次のように書き直せる (図 5(a)).

$$\log|\xi(k)| \sim \frac{k^2}{2\pi\lambda_c}$$

実測値は確かにこの見積りでよく近似できているが, 見積りよりは大きく, ガタガタしている. 本研究ではこの見積りを改善する.



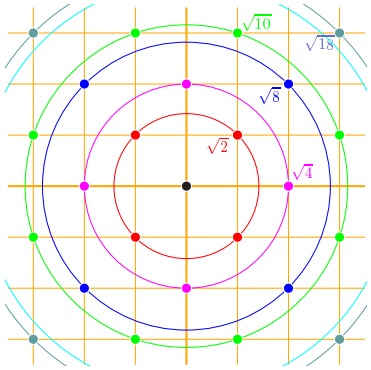
(a)Vardi の見積りと実測値



(b) 可能な枝の種類 $p(k)$ と歩幅 k の 2 乗の関係

図 5: Vardi の見積り

ガウス整数 $a+bi$ が $\pm1\pm i$ 以外のガウス素数であるとき, a と b の片方は偶数, もう片方が奇数になる. 二つのガウス素数の差を $x+yi$ とすると, x と y は共に偶数か, 共に奇数となる. 従ってある歩幅 k のグラフ G に現れる枝の形 (x,y) は, $x^2+y^2\le k^2 (x\equiv y \pmod 2)$ を満たす限られたものとなる. 歩幅 k に対してこのような条件を満たす (x,y) のペアの数を $p(k)$ と定義する. ただし対称性を除くため $x\ge 0$, $y>0$ に限定する.



k	$p(k)$	可能な枝
$\sqrt{1}$	1	(0,1)
$\sqrt{2}$	2	(1,1)
$\sqrt{4}$	3	(0,2)
$\sqrt{8}$	4	(2,2)
$\sqrt{10}$	6	(1,3) (3,1)
$\sqrt{16}$	7	(0,4)
⋮	⋮	⋮
$\sqrt{40}$	18	(2,6) (6,2)
$\sqrt{50}$	21	(1,7) (5,5) (7,1)

90 度ごとの回転対称は同一視

図 6: 歩幅 k の可能な枝の種類

図 5(b) は $p(k)$ と k^2 の関係を表す. 大域的にはもちろん $p(k)\sim \pi k^2/8$ となるが, 細かく見るとずいぶんと折れ曲がっている. ここで注目したいことは, 図 5 の (a) と (b) の折れ曲がり具合があまりにも似ていることである. この二つの図を組み合わせたのが図 7(a) である. これから明らかなように, $|\xi(k)|$ と $p(k)$ は見事に比例している. これが Vardi の見積りに対する改良である.

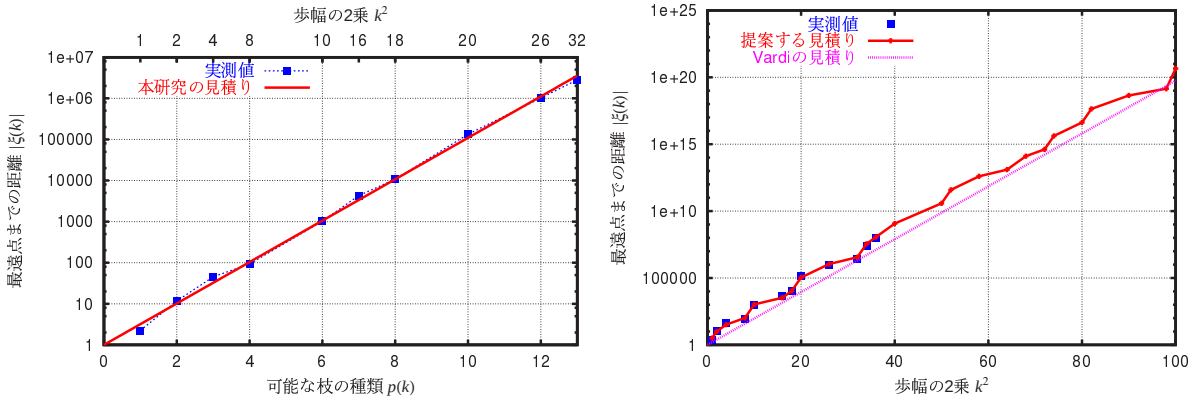
最小二乗法により $1\le k\le 32$ の 10 点の実測値をあてはめると

$$\log|\xi(k)|\sim 1.160p(k)$$

となった. 一方, Vardi の見積りは $p(k)\sim \pi k^2/8$ であることを用いると

$$\log|\xi(k)|\sim \frac{k^2}{2\pi\lambda_c}\sim \frac{1}{2\pi\lambda_c}\frac{8p(k)}{\pi}=\frac{4p(k)}{\pi^2\lambda_c}=1.158p(k)$$

となり, 二つの見積りは大域的にほとんど一致する (図 7(b)).



(a) 本研究の見積りと実測値　　(b) 二つの見積りの比較

図 7: 提案する見積り

パーコレーション理論に基づいた Vardi の見積りは, ガウスの掘割り問題の本質を把えているが, ガウス素数が連続空間にランダムに存在すると仮定している. 本研究ではガウス素数の離散性に着目して修正を行った. その結果, 実測値と非常によく適合し, また Vardi との結果とも大域的に一致する見積りが得られた.

謝辞

東京大学数理第二研究室の室田一雄教授, 松浦史郎助手には有益なアドバイスを頂いた. 同じく矢吹光祐氏にはこの面白い問題を教えて頂いた. 数理第五研究室の来嶋秀治氏には有益なアイデアを頂いた. 科学技術振興調整費先導的研究基盤整備の計算機環境を利用させて頂いた. ここに記して感謝する.

参考文献

[1] E. Gethner, S. Wagon, and B. Wick, "A Stroll Through the Gaussian Primes," *American Mathematical Monthly* **105**:4 (1998), 327–337.

[2] E. Gethner and H. M. Stark, "Periodic Gaussian Moats," *Experimental Mathematics* **6**:4 (1997), 289–292.

[3] S. Wagon, *Mathematica in Action*, Springer-Verlag New York, 1999.

[4] I. Vardi, "Prime Percolation," *Experimental Mathematics* **7**:3 (1998), 275–288.

[5] J. H. Jordan and J. R. Rabung, "A Conjecture of Paul Erdös Concerning Gaussian Primes," *Mathematics of Computation* **24** (1970), 221–223.

[6] P. Ribenboim, *The Little Book of Big Primes*, Springer-Verlag, New York, 1991.

[7] G.H. Hardy, E.M. Wright, *An Introduction to the Theory of Numbers 5th Edition*, Oxford University Press, 1979.

[8] M. Agrawal, N. Kayal and N. Saxena, "PRIMES is in P," Preprint (2002), `http://www.cse.iitk.ac.in/news/primality.html`

[9] M. O. Rabin, "Probabilistic algorithm for testing primality," *Journal of Number Theory* **12** (1980), 128–138.

[10] 奥村晴彦ほか『Java によるアルゴリズム事典』(技術評論社, 2003 年)