計算効率の良いarctan関係式の探索の試み

松元 隆二
九州工業大学 情報工学部 技術部

平成21年3月5日

概 要

円周率の公式の一種に，マチンの公式に代表される三角関数のarctanを用いた式がある．arctanを用いた式は簡単に探索することができる．今回既知のarctan関係式より計算効率が良い式が無いかを，九州大学スーパーコンピュータを用い検索を試みた．結果として既知の公式より効率が良い公式は発見出来なかったが，得られたデータを報告する．

1 はじめに

円周率の公式の一種に，マチンの公式に代表される，三角関数tanの逆関数arctanを用いた式がある．arctanを用いた式 (arctan関係式) は計算が容易で，手計算の時代から円周率の記録更新に用いられている．以下に代表的な式を挙げる．

$$\frac{\pi}{4} = \arctan(1). \quad \text{三角関数の定義より} \tag{1}$$

$$= 4\arctan(\frac{1}{5}) - \arctan(\frac{1}{239}). \quad \text{1706年, マチン, 円周率の「マチンの公式」として有名} \tag{2}$$

$$= 44\arctan(\frac{1}{57}) + 7\arctan(\frac{1}{239}) - 12\arctan(\frac{1}{682}) + 24\arctan(\frac{1}{12943}). \quad \text{1896年, ステルマー} \tag{3}$$

$$= 12\arctan(\frac{1}{49}) + 32\arctan(\frac{1}{57}) - 5\arctan(\frac{1}{239}) + 12\arctan(\frac{1}{110443}). \quad \text{1982年, 高野喜久雄} \tag{4}$$

式 (1) が一番簡単な公式であるが，計算効率が悪いため，改善のためマチンの公式 (2) のように複数の項に分解を行う．分解した式が等式である事の証明はtanの加法定理を用いれば可能 [1] である．arctanの分解の組み合わせは多数存在し，探索も容易であるため，多くの円周率愛好家によって計算効率の良い式の探索が試みられている．

今回スーパーコンピュータを用いて既知のarctan関係式より計算効率が良い式の探索を試みた．

2 arctan関係式の計算効率の良さ

arctan関係式の良さの指標として，$\arctan(1/x)$ のテイラー級数の収束の速さが用いられる．

$$\arctan(\frac{1}{x}) = \sum_{k=0}^{n} \frac{(-1)^k}{(2k+1)(x^{2k+1})}. \quad (1 \leq |x|, \text{奇関数})$$

級数を小数点以下N桁の精度まで計算するには，級数の誤差 ε_n が 10^{-N} 以下になるまで計算すれば良い．

$$\varepsilon_n = \left| \frac{(-1)^n}{(2n+1)(x^{2n+1})} \right| < 10^{-N}. \quad \text{変形すると，} \quad (2n+1)(x^{2n+1}) > 10^N.$$

x の値ごとに ε_n を検討する．式 (1) に相当する $x=1$ の場合は誤差 ε_n として $1/(2n+1)$ のみが残るため収束が遅い．事実上計算不可．$x>1$ の場合は誤差 ε_n は x^{-2n} にほぼ比例する．つまり x が大きいほど収束が速い．以上をまとめると，

$$N\text{桁の精度で計算するのに必要な級数の項数 } n \simeq \begin{cases} (10^N)/2, & \text{for } x=1 \\ N/(2\log_{10}(x)), & \text{for } x>1 \end{cases}$$

計算効率が良いarctan関係式は x が大きい式である．また公式を構成する $\arctan(1/x)$ の項数が少ない式である．

表 1: x^2+1 の素因数に現れる素数の種類 (素因数分解の精度は関数 $factor()$ に依存)

	$x < 1.0e6$	$x < 1.0e7$	$x < 1.0e8$	$x < 1.0e9$	$x < 1.0e10$
素数の種類 (注 1)	704,536	7,026,558	70,122,424	700,184,484	6,693,568,566
複数回出現素数 (注 2)	53,039	450,395	3,916,877	34,634,735	310,529,018
〃　割合	7%	6%	6%	5%	5%

注 1: 素数の累積個数ではなく，種類である．何度同じ素数が出現しても 1 種類と数える．
注 2: 2 回以上出現する素数を数えた．但し「$7^2 = 2 \cdot 5^2$」の場合は 5 が 1 回出現したと数える．

3　arctan 関係式の探索法

arctan 関係式を探索する方法として，実装が容易なのは三角関数の加法定理を使って総あたり方法で調べる方法や，加法定理を変形し方程式を立て，整数解を求める方法である．小規模な検索では本手法で可能であるが，範囲を少し広げると組合せ数が多すぎ効率が悪い．先人によって効率が良い大規模検索の方法が幾つか提案されている [3, 4]．今回はステルマーが発見した「共通の素数」を手掛かりに使って探索する手法を用いた [4].

共通の素数:　arctan 関係式の各項の $\arctan(1/x)$ の x^2+1 を素因数分解する．前出の高野の式 (4) は以下のようになる．

$$\begin{aligned} 49^2+1 &= 2 \cdot 1201 \\ 57^2+1 &= 2 \cdot 5^3 \cdot 13 \\ 239^2+1 &= 2 \cdot 13^4 \\ 110443^2+1 &= 2 \cdot 5^8 \cdot 13 \cdot 1201 \end{aligned}$$

共通の素数 2,5,13,1201 が現れる．arctan 関係式は必ず x^2+1 に共通の素因数を持つという顕著な性質がある．但し素数 2 は現れない場合もある．この性質を利用して検索すれば，闇雲に検索するより非常に効率が良い．

本性質を使った具体的な探索法として，まず素数を幾つか選び，選んだ素数が現れる x^2+1 を探す．次に文献 [4] で述べられている方程式を解く．しかしこの探索を行なうには，x^2+1 の素因数分解の表を作成する必要がある．

なお，x^2+1 の素因数分解には以下のような性質 [2] があり，素因数には $P \equiv 3 \pmod 4$ 型素数は出現しない．

$$x^2+1 = 2^\tau \cdot P_1^{\alpha(1)} \cdot P_2^{\alpha(2)} \cdots P_S^{\alpha(S)} \quad \begin{cases} \tau & 0 \text{ もしくは } 1. \\ \alpha(s) & \text{整数.} \\ P_s & P_s \equiv 1 \pmod 4 \text{ を満たす素数.} \end{cases} \tag{5}$$

4　x^2+1 の素因数分解

第 2 節で述べたように arctan 関係式は x が大きい方ほど収束が速い．そのため可能な限り大きな x を含む式を探索したい．しかし x が大きくなるほど探索のために必要な x^2+1 の素因数分解の表の作成コストが増加する．今回様々な工夫を行なってコスト削減に成功した．(以降数値表現として指数形式を使う．凡例: $3.14 \cdot 10^5 = 3.14e5$)

数学ライブラリによる素因数分解:　はじめに x が最大 10 桁 ($1 < x < 1.0e10$) の範囲で x^2+1 の素因数分解を実施した．x^2 が含まれるため，最大 20 桁の整数の素因数分解である．大きな整数であり，検索範囲が広く，効率の良い素因数分解方法が求められる．素因数分解法は複数知られるが，試し割り法以外は実装が難しいため，数学ライブラリ PARI/GP[7] に含まれる素因数分解関数 $factor()$ を用いた．計算機資源として九州大学情報基盤研究開発センター所有のスーパーコンピュータ PRIMEQUEST580 (2 コア x 32CPU) を用いて約 6.5 日間で完了した．計算結果として約 67 億種類の素数が出現した．今回は共通の素数を探すのが目的のため，複数回出現する素数を数えると約 5%の 3 億種類であった．詳細は表 1 に挙げた．結果として素数の種類が多すぎた．共通の素数を探す方法では組合せ数が多すぎる．

表 2: x が最大 10 桁 ($x < 1.0e10$) までの素因数分解の計算時間

	$x < 1.0e8$	$x < 1.0e9$	$x < 1.0e10$
素数の範囲制限無し	13m	3h 16m	6d 11h 18m
素数の範囲限定有り ($2 \leq P < 65536$)	13m	(推定)1h 50m	(推定) 20h

計算環境: 富士通 PRIMEQUEST580, Intel Itanium2 1.6GHz(2 コア) x 32CPU, C 言語, MPI

表 3: 素数の範囲を限定した場合に残る x の個数と，残る確率 (%)

素数範囲＼x 範囲	$x < 1.0e6$	$x < 1.0e9$	$x < 1.0e14$
$2 \leq P < 100$	149(0.15%)	155(0.00%)	155(0.0000%)
$2 \leq P \leq 761$	4,620(0.46%)	18,505(0.00%)	43,934(0.0000%)
$2 \leq P < 1000$	6,499(0.65%)	33,259(0.00%)	109,192(0.0000%)
$2 \leq P \leq 3733$	26,258(2.63%)	411,654(0.04%)	8,818,304(0.0000%)
$2 \leq P < 10000$	54,611(5.46%)	1,623,721(0.16%)	114,236,766(0.0001%)
$2 \leq P < 65536$	141,885(14.2%)	10,261,986(1.03%)	3,926,532,525(0.0039%)

素数の範囲限定: 組合せ数を減らすために，素数 P の範囲を 65536 以下にした．65536 以下の $P \equiv 1 \pmod 4$ 型の素数は 3257 個になる．素数を限定すると関数 $factor()$ の計算コストが減り時間短縮した．素数の範囲制限の有無で，計算時間の比較を表 2 に挙げる．結果を検討すると，x が小さい範囲では差がでないが，x が大きくなるにつれ計算時間が大幅減となった．

さらに，対象とする素数を限定したことにより，arctan 関係式探索時に調査する x の数量も減らす事が出来た．例えば素数の範囲を 1000 未満に限定すると，x^2+1 に 1000 以上の素因数が出現する x を捨てる事が出来る．例えば $x = 49$ の場合の $49^2 + 1 = 2 \cdot 1201$ は素数が 1000 以上なので捨てる．素数の範囲を限定した場合に有効な x がどのように変化するかを表 3 に挙げる．

素数の範囲限定により，素因数分解のコストと有効な x の個数が大幅に減少した．しか文献 [6] によると素因数分解のアルゴリズムは異なるが x が 14 桁 ($x < 1.0e14$) までの範囲は素数が 761 以下という条件で探索済みである．新規の公式を見つけるにはさらに広範囲の x および素数の探索が必要である．

4.1 フルイを使った高速化

素数表を作るアルゴリズムに「エラトステネスの篩 (フルイ)」がある．本手法は素数表を作るアルゴリズムであり，素因数分解とは基本的に関係ないが，剰余の周期性とフルイを応用すると，x^2+1 の素因数分解の高速化が可能である．

x^2+1 の剰余の周期性: x^2+1 を素数 P で割った剰余には周期性がある．

P:素数, M:正の整数, t:整数 $(0 \leq t < P)$ と置く． x を次のように定義: $x = P \cdot M + t$.

$$\begin{aligned} x^2+1 &= (PM+t)^2+1. \\ &= (PM)^2 + 2PMt + t^2 + 1. \\ &= P(PM^2+2Mt) + t^2 + 1. \\ x^2+1 &\equiv t^2+1 \pmod P. \end{aligned}$$

従って，x^2+1 の剰余には P を周期とする規則性がある．

この結果を使うと x^2+1 の素因数分解の試し割りの大幅削減ができる．エラトステネスのフルイの応用である．

フルイを使った素因数分解: これから説明する手法は文献 [6] に掲載されている手法の改良版である．

以降簡単のために，特記した場合を除き，素数 P の範囲は 2 および 65536 以下の $P \equiv 1 \pmod 4$ 型素数に限定する．

- 範囲内の素数 P 全てに対して，$t^2 + 1 \equiv 0 \pmod P$ を満たす t を調べる．$P = 5$ の場合 t は 2 と 3 になる．
- t を使って数列 $P \cdot M + t$ を生成する．$P = 5$ の場合は以下の数列になる．

$$x = \{5 \cdot 0 + 2, \quad 5 \cdot 0 + 3, \quad 5 \cdot 1 + 2, \quad 5 \cdot 1 + 3, \quad 5 \cdot 2 + 2, \quad 5 \cdot 2 + 3, \quad \cdots\}$$

- x の検索範囲を定め，探索範囲のフルイを作成する．フルイの構造はエラトステネスのフルイとは異なり 2 次元の表である．次に数列を使ってフルイに印 c を付ける．$P = 2$ は頻度が 50%なので例外扱いにした方が良い．

	t	$x=2$	$x=3$	$x=4$	$x=5$	$x=6$	$x=7$	$x=8$	$x=9$	x=...
$P=5$	2,3	c	c				c	c		...
$P=13$	5,8				c			c		...
...	...									...
$P=65521$	24297,41224									...

- 各 x ごとに印の付いている素数で試し割りする．$x = 8$ は素数 5 と 13 の二箇所に印が付いている．$8^2 + 1 = 5 \cdot 13$ だから割り切れる．注意点として，フルイで素数の範囲を限定しているため，65536 以上の素因数が含まる可能性がある．

以上の手法で大幅に試し割りのコストを省ける[1]．

フルイを使った手法により，大幅な時間削減に成功した．しかし本手法をもっても目標の 14 桁までの素因数分解には計算時間がかかり過ぎる．時間測定を行なうと，フルイの作成より試し割りが大幅に時間を消費している．

試し割の削減: フルイに印が付いている全ての x に対して試し割を行うのでなく以下のような基準を設けた[2]．

- フルイの印が四箇所以上の x のみ試し割りを行なう．印が三箇所以下の x は，出現頻度が高いのだが，素因数分解に失敗する可能性が高い．(失敗するのは 65536 以上の素因数が含まれる場合．)

  以下の表は，$x^2 + 1$ の素因数が何種類の素因数を含むかの一覧である．但し素数 2 は調査対象外．素因数の種類が 3 個以下の x は 1.0e11 で頭打ちになっている． ($2^{64} = 1.8e19$)

素因数の種類	$x < 1.0e8$	$x < 1.0e9$	$x < 1.0e10$	$x < 1.0e11$	$x < 1.0e14$	$x < 1.8e19$
1	102	102	102	102	102	102
2	3,036	3,036	3,036	3,036	3,036	3,036
3	60,670	61,120	61,191	61,196	61,196	61,196
4	576,733	859,076	909,684	918,024	920,195	–
5	1,192,291	3,609,889	7,258,592	10,175,914	11,058,363	–
6	687,885	4,083,634	15,740,876	41,074,699	108,852,587	–
7	123,326	1,465,340	10,636,001	50,801,408	660,544,941	–
8	6,541	173,492	2,473,317	21,696,482	1,477,440,333	–
9	82	6,243	200,042	3,390,141	1,216,104,465	–
10 以上	0	54	4,978	192,162	451,547,307	–

- フルイの印が三箇所以下の場合は別手段で求める．式 (5) において，素数 P_s を三種類以下に限定し，式を満たす x を総あたり方法で調べる[3]．

[1] エラトステネスのフルイの最適化手法の幾つかは，本手法に応用可能である．広範囲探索にはフルイのメモリ削減対策は必須である．

[2] その他，64bit を超える整数演算に GMP などのライブラリを使用せず，マシン語を使用した．PARI/GP のソースコードが参考になる．

[3] 本手法は計算量が $nCr = n!/(n-r)!$ に比例するため，素数の範囲 (n) を広げる/素数の組合せ数 (r) を増やすと，nCr が増大するため，大規模な検索には向かない．

表 4: x^2+1 の素因数分解の改善状況 (イタリック体は推定時間, $2^{64}=1.8e19$)

	$x<1.0e8$	$x<1.0e9$	$x<1.0e10$	$x<1.0e11$	$x<1.0e14$	$x<1.8e19$
素数の範囲制限無し	13m	3h 16m	6d 11h 18m	–	–	–
素数範囲限定 ($2 \le P < 65536$)	13m	*1h 50m*	*20h*	*8d 8h*	*23y*	–
フルイ (印が 4 個以上)	*1s*	*3s*	31s	4m 9s	4d 18h 2m	–
組み合せ (素数 3 個以下)(注 3)	–	–	–	–	–	2d 12h 24m

計算環境: 富士通 PRIMEQUEST580. 但し注 3 はシングルコアで実行.

表 5: スパコン vs PC の計算時間. x が 10 桁 ($x<1.0e10$) までの x^2+1 の素因数分解 (イタリック体は推定)

	スパコン (2 コア x 32CPU)	AMD Opteron1352(4 コア)	Intel Core2DuoE8500(2 コア)
素数の範囲制限無し	6d 11h 18m	23d 21h 36m	*27d 21h*

PC の計算環境: Linux CentOS5.2/x86_64, GCC, OpenMPI

時間削減: 以上の工夫を行なった結果の計算時間を表 4 に挙げる. 最初の手法では x が 10 桁の素因数分解で 6.5 日間必要であったが, 最終的には 1 万倍の 14 桁の素因数分解が, フルイ (4d 18h) と組み合せ (2d 12h) の合計で 7.3 日間で可能になった.

5 arctan 関係式の探索 (まとめ)

本来の目的である x^2+1 の素因数分解の結果を使って arctan 関係式を探索する作業は着手出来なかった. 今後の課題である. 参考データとして, 古いデータだが 2000 年に探索した結果が Web に掲載してある [8].

最後に, arctan 関係式を探すのは難しくないので, 興味のある方は挑戦して欲しい. また, 今回はスーパーコンピュータを用いたが, 最新の PC 用の CPU を用いれば計算時間差はわずか 4〜5 倍程度である. 表 5 参照.

6 謝辞

本研究は, 九州工業大学情報科学センターが主催している「九大研究用計算機システム利用支援」に補助をいただいた.

参考文献

[1] オンライン辞典 Wikipedia の「マチンの公式」の欄, http://ja.wikipedia.org/wiki/マチンの公式

[2] 高木 貞治, 初等整数論講義 第 2 版, p.55, 共立出版, 1971 年, ISBN-13: 978-4320010017.

[3] 金田康正, 「bit ナノピコ教室・π/4 と arctan との関係式」, bit, 出題 1982 年 6 月, 解答 1982 年 10 月 pp.102-109.

[4] 高野喜久雄,「π の arctangent relations を求めて」, bit, 1983 年 4 月, pp.83-91.

[5] 猪口和則,「πの公式をデザインする」, 新風舎, 1998 年, 1400 円, ISBN: 4-7974-0493-0.

[6] Jörg Arndt, Arctan relations for Pi, http://www.jjj.de/arctan/arctanpage.html

[7] PARI/GP Development Headquarters, http://pari.math.u-bordeaux.fr/

[8] 松元隆二, arctan 関係式一覧, 2000 年 3 月,
http://www.pluto.ai.kyutech.ac.jp/plt/matumoto/atan_table.txt