

集中講義「電子情報数理インテンシブ」

電子工学と情報数理工学の融合研究事例

電子回路での新アルゴリズムをいかにして考案したか

群馬大学 大学院理工学府 電子情報部門

小林 春夫

k_haruo@el.gunma-u.ac.jp

- アルゴリズムを発見する
「気持ち」を伝える
- フィボナッチ数列、黄金比
- 逐次比較近似AD変換器冗長アルゴリズム
- 最後に

電子情報部門(電気電子工学専攻)
小林研究室での研究

東大名誉教授 北森俊行先生



北森先生

思考力・創造力の向上のために

- 数学の定理を教え、証明してみせるよりも、
定理を発見する気持ちを教える。
- 物理法則を教えるよりも、
物理法則を見つけ出そうという気持ちを教える。
- 出来上がった理論を教えるよりも、
理論を創る気持ちを教える。

フンボルトの大学の理念

「知識は 発展している、
作り出されている、進歩している。

大学は 学問を未だに
完全には解決されていない
問題として、
たえず研究されるものとして
扱うことに特色がある。」



Friedrich Wilhelm
Christian Karl Ferdinand
Freiherr von Humboldt
1767年 - 1835年.

ドイツの言語学者・政治家・貴族。
フンボルト大学（ベルリン大学）
の創設者

学生時代の「数学」の講義にて印象に残る言葉
「数学の定理の発見は、論理的に
ひとつひとつ積み上げてなされるのではない。
このような定理が成立するのではないかと
直感で予測して、論理的に証明していく。」

電子回路分野:

「回路解析だけでなく 設計を教えよ」

学生時代の線形代数の講義

「なぜこんなに定理が次から次へとでてくる！」



線形を扱っているから

少しでも非線形が含まれると
ほとんど定理はでてこない。

フィボナッチ数列とは

フィボナッチ数列

$$F_0 = 0$$

$$F_1 = 1$$

$$F_{n+2} = F_n + F_{n+1}$$

初めの項を計算

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

⇒ フィボナッチ数と呼ばれる



Leonardo Fibonacci
(伊: 1170-1250年頃)

隣り合う項の比率は以下の値 φ に収束

$$\lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = 1.618033988749895 = \varphi$$

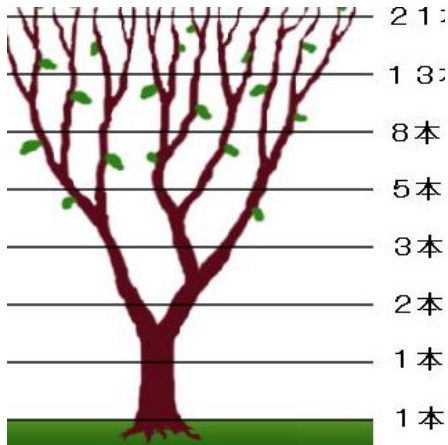
収束比率 φ : 黄金比 (Golden ratio)

不思議な数“フィボナッチ数”

フィボナッチ数 : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

植物と関係が深い⇒最も日光を浴びれる、最も多く配置できる

木の枝分かれ



花びら枚数

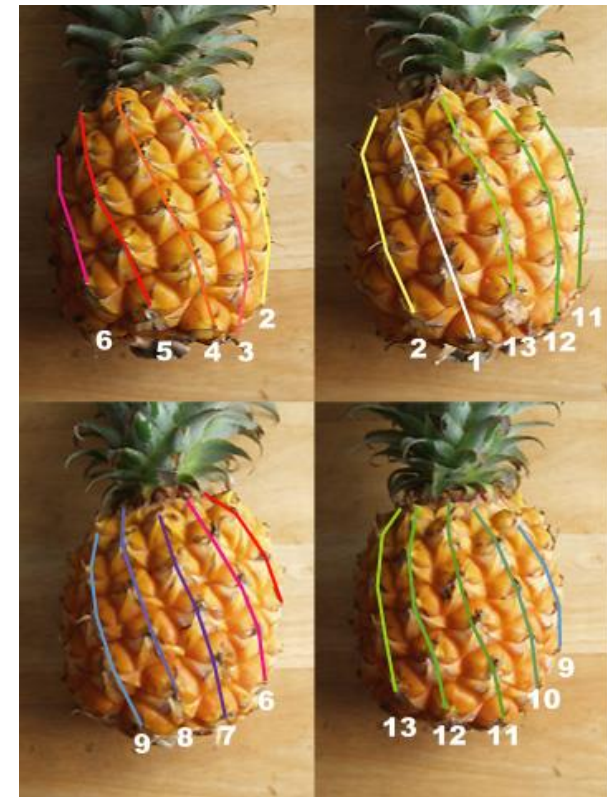


実の付け方

松ぼっくり



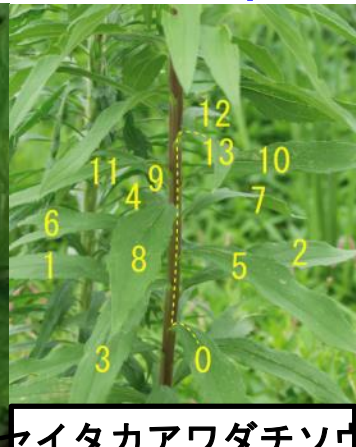
パイナップル



葉序(植物の葉の付け方)



カキ



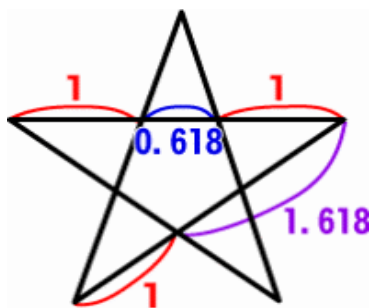
セイタカアワダチソウ

最も美しい比率“黄金比”

黄金比: $\lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = 1.618033988749895 = \varphi$

フィボナッチ・黄金比⇒“最も安定し効率の良い配置” 自然の知恵

正五芒星



クフ王のピラミッド(埃)



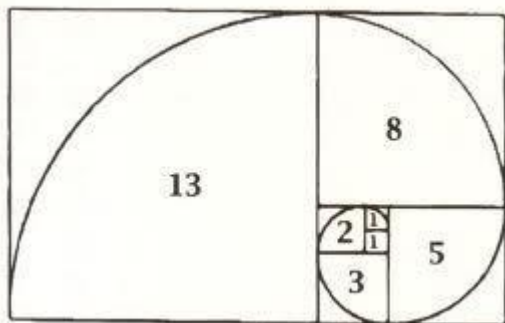
パルテノン神殿(希)



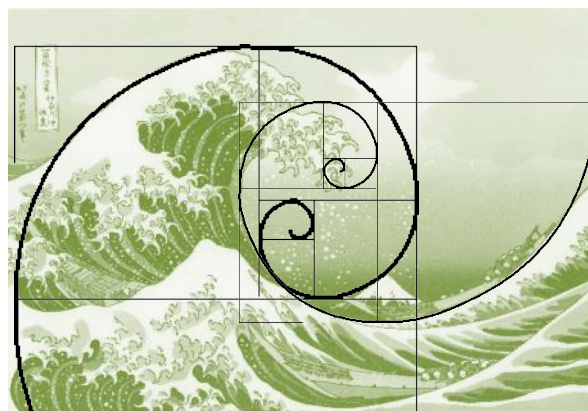
ヴィーナス



黄金螺旋



神奈川冲浪裏(葛飾北斎)



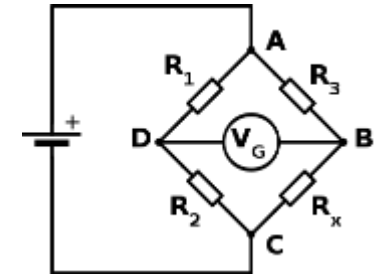
モナ・リザ(ダヴィンチ)



● 零位法

測定量が基準値と等しいかを調べる

天秤、ブリッジ回路



● 偏位法

測定量の結果として生じる

計器の指示値を読む

体重計、電圧計

零位法

(ゼロ位法、Zero Method, Null Method)

- 利点:

平衡の検知は高精度可能

測定対象からエネルギーをとることがない。

基準量の精度で測定可能

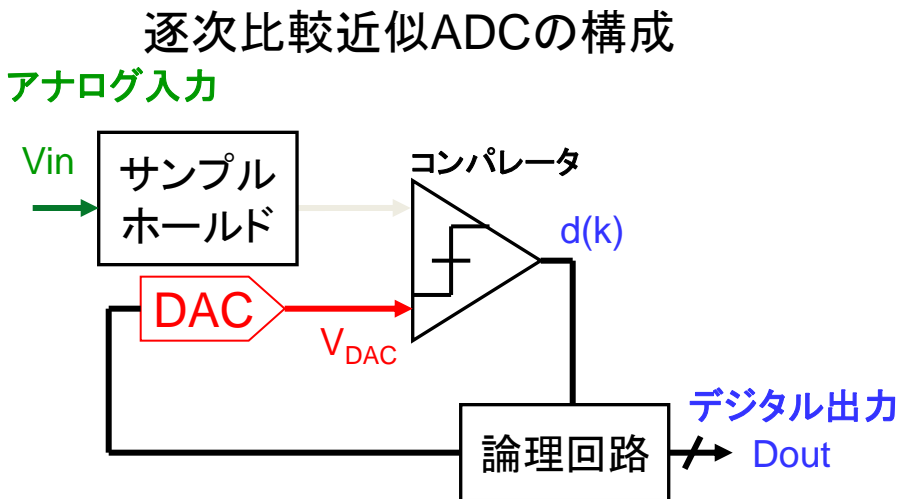
高精度測定では零位法を使用

- 欠点:

測定量と基準量が等しくなるまで調整要

逐次比較近似ADC

逐次比較近似ADC



$$\begin{aligned} \text{Dout} &= 4d(1) + 2d(2) + 1d(3) \\ &= 4 + 1 \\ &= 5 \end{aligned}$$

逐次比較近似ADCの動作

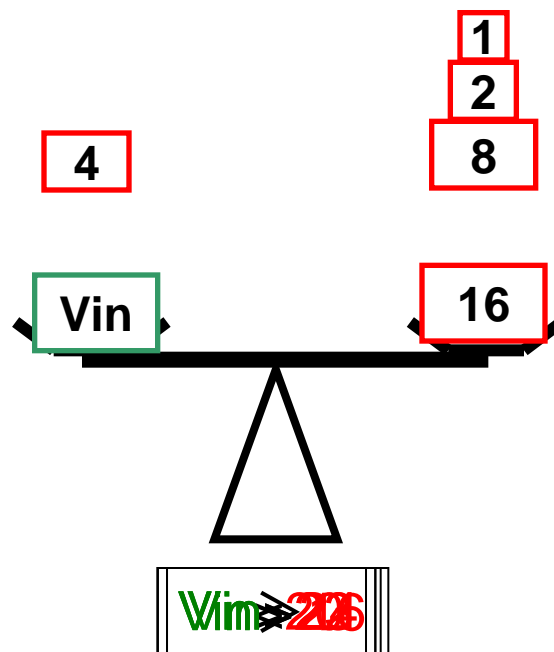
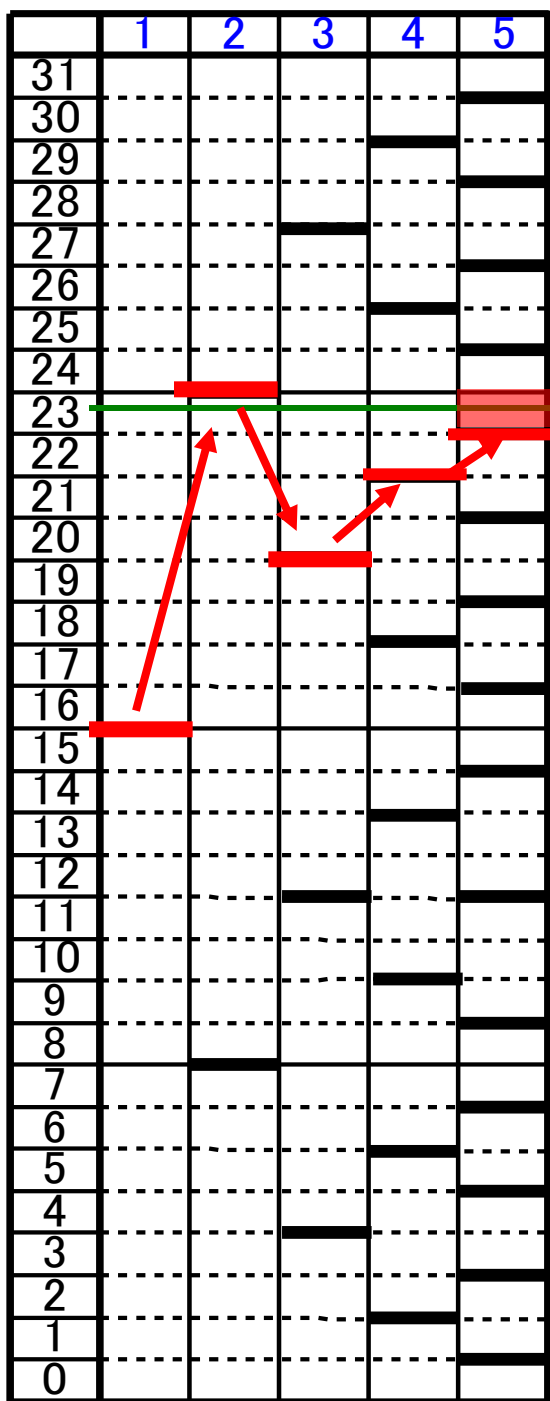
比較サイクル レベル	1	2	3	出力コード Dout
7				111
6				110
5		0	1	101
4	1			100
3				011
2				010
1				001
0				000

入力5.3

2進探索アルゴリズム動作 13

5ビット

動作例: アナログ入力 23.5
のとき



$$\boxed{\text{Vin}} = \begin{array}{c} \boxed{1} \\ \boxed{2} \\ \boxed{8} \\ \boxed{16} \end{array} - \boxed{4} = \boxed{23}$$

- フィボナッチ数列

たくさんの定理が現在も生まれている

- 黄金比

自然界・生物、建築等 あちこちにてでてる

- フィボナッチ数列と黄金比は密接な関係



冗長逐次比較近似AD変換アルゴリズムに
適用すれば、面白い結果が得られることが期待



調査・研究に着手

研究室ゼミにて

フィボナッチ数列、黄金比 たくさんの「定理」が次々に導出される

小林佑太郎、香積正基

趣味・楽しみで数学の啓蒙書を読む

アルフレッド・S・ポザマンティエ
イングマル・レーマン 著

松浦俊輔 訳(2010年8月)

『不思議な数列フィボナッチの秘密』

日経BP出版社



レオナルド・ダ・ピサ(1170年頃 - 1250年頃)

ピサのレオナルドもしくはフィボナッチと呼ばれる

中世で最も才能があったと評価されるイタリアの数学者

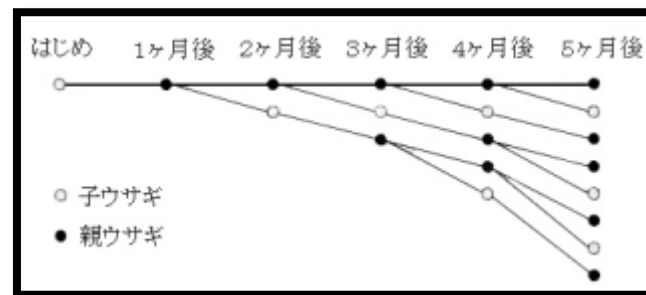
主な業績

1202年に出版した『算盤の書(リベル・アバキ)』において

- ・アラビア数字の採用
- ・**フィボナッチ数列(兎問題)の紹介**

ヨーロッパの数学の発展に大きく貢献した

19世紀
リュカによって
有名になる



フィボナッチ数列とは

フィボナッチ数列は以下のように定義される数列

$$F_0 = 0$$

$$F_1 = 1$$

$$F_{n+2} = F_n + F_{n+1}$$

科学・数学や自然界（特に植物）に
性質がみられることが多い数列

初めの項を計算すると

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377,
610, 987, 1596, 2583, 4180, 6764, 10945...

また隣り合う項の比率は以下に収束する(黄金比)

$$\lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = 1.618033988749895 = \varphi$$

(1:1.618もしくは0.618:1に収束)

自然界のフィボナッチ数

- * 花びらの数はフィボナッチ数である⇒コスモス(8枚)
- * 植物の花や実に現れる螺旋の数もフィボナッチ数である
⇒ヒマワリ、パイナップル、松ぼっくり
- * 葉序(植物の葉の付き方)はフィボナッチ数と関連している
- * 蜜蜂の家系を辿っていくとフィボナッチ数列が現れる



自然界の黄金比

- * オウムガイやカタツムリの殻は黄金らせんで構成される
- * パルテノン神殿・ピラミッドなどの建造物に黄金比を見いだせる
- * 美術作品や音楽に取り入れられている
- * 名刺などの長方形を作るときに使われる



注)ここで使用される n は $n \geq 1$ となる任意の自然数である
取り上げるフィボナッチの基本性質は
13個あるが細かな紹介は一部とする

基本性質

- ①連続する10個のフィボナッチ数の和は11で割り切れる。(A|B \longleftrightarrow AはBで割り切れる)
 $11 \mid (F_n + F_{n+1} + F_{n+2} + F_{n+3} + F_{n+4} + F_{n+5} + F_{n+6} + F_{n+7} + F_{n+8} + F_{n+9})$
- ②連続するフィボナッチ数は互いに素である。
- ③合成数番目のフィボナッチ数(4番を除く)も合成数である。(合成数=素数でない数)
これを別の言い方で表すと n が素数でない場合、 F_n は素数ではない。

フィボナッチ数列の性質の例示②

②連続するフィボナッチ数は互いに素である。

例1) F_{20} と F_{21}

$$F_{20} = 6765 = 3 * 5 * 11 * 41$$

$$F_{21} = 10946 = 2 * 13 * 421$$

例2) F_{175} と F_{176}

$$F_{175} = 177244579041379840132227567949787325$$

$$= 5^2 * 13 * 701 * 3001 * 141961 * 17231203730201189308301$$

$$F_{176} = 2706074082469569338358691163510069157$$

$$= 3 * 7 * 43 * 47 * 89 * 199 * 263 * 307 * 881 * 967 * 93058241 * 562418561$$

どんなに値が大きくなったとしても
隣り合うフィボナッチ数ならば
公約数は1以外存在しない

基本性質

④フィボナッチ数の最初の n 個の和は2つ後の項から1引いたものに等しい。

$$\sum_{i=1}^n F_i = F_1 + F_2 + F_3 + \cdots + F_n = F_{n+2} - 1$$

⑤連続する偶数番目のフィボナッチの数の和は、
和の最後の偶数番目のフィボナッチ数の次のフィボナッチ数より1小さい。

$$\sum_{i=1}^n F_{2i} = F_2 + F_4 + F_6 + \cdots + F_{2n-2} + F_{2n} = F_{n+2} - 1$$

⑥連続する奇数番目のフィボナッチ数の和は、
和の最後の奇数番目のフィボナッチ数の次のフィボナッチ数に等しい。

$$\sum_{i=1}^n F_{2i-1} = F_1 + F_3 + F_5 + \cdots + F_{2n-1} = F_{2n}$$

フィボナッチ数列の性質の例示⑥

⑥連続する奇数番目のフィボナッチ数の和は、
和の最後の奇数番目のフィボナッチ数の次のフィボナッチ数に等しい。

$$\sum_{i=1}^n F_{2i-1} = F_1 + F_3 + F_5 + \cdots + F_{2n-1} = F_{2n}$$

例2) n=5のとき

$$F_1 + F_3 + F_5 + F_7 + F_9 = 1 + 2 + 5 + 13 + 34 = 55 = F_{10}$$

例1) n=7のとき

$$\begin{aligned} F_1 + F_3 + F_5 + F_7 + F_9 + F_{11} + F_{13} \\ = 1 + 2 + 5 + 13 + 34 + 89 + 133 = 377 = F_{14} \end{aligned}$$

基本性質

⑦フィボナッチ数の平方の和は、最後の数とその次のフィボナッチ数との積に等しい。

$$\sum_{i=1}^n F_i^2 = F_n F_{n+1}$$

⑧2つの交互的フィボナッチ数の平方の差は、
両者の番号の和を番号とするフィボナッチ数に等しい。

$$F_n^2 - F_{n-2}^2 = F_{2n-2}$$

⑨2つの連続するフィボナッチ数の平方の和は、
その番号の和を番号とするフィボナッチ数に等しい。

$$F_n^2 + F_{n+1}^2 = F_{2n+1}$$

⑩4つの連続するフィボナッチ数については、中2項の平方の差が両端の
2項の積に等しい。

$$F_{n+1}^2 - F_n^2 = F_{n-1} * F_{n+2}$$

フィボナッチ数列の性質の例示⑦

⑦ フィボナッチ数の平方の和は、最後の数とその次のフィボナッチ数との積に等しい。

$$\sum_{i=1}^n F_i^2 = F_n F_{n+1}$$

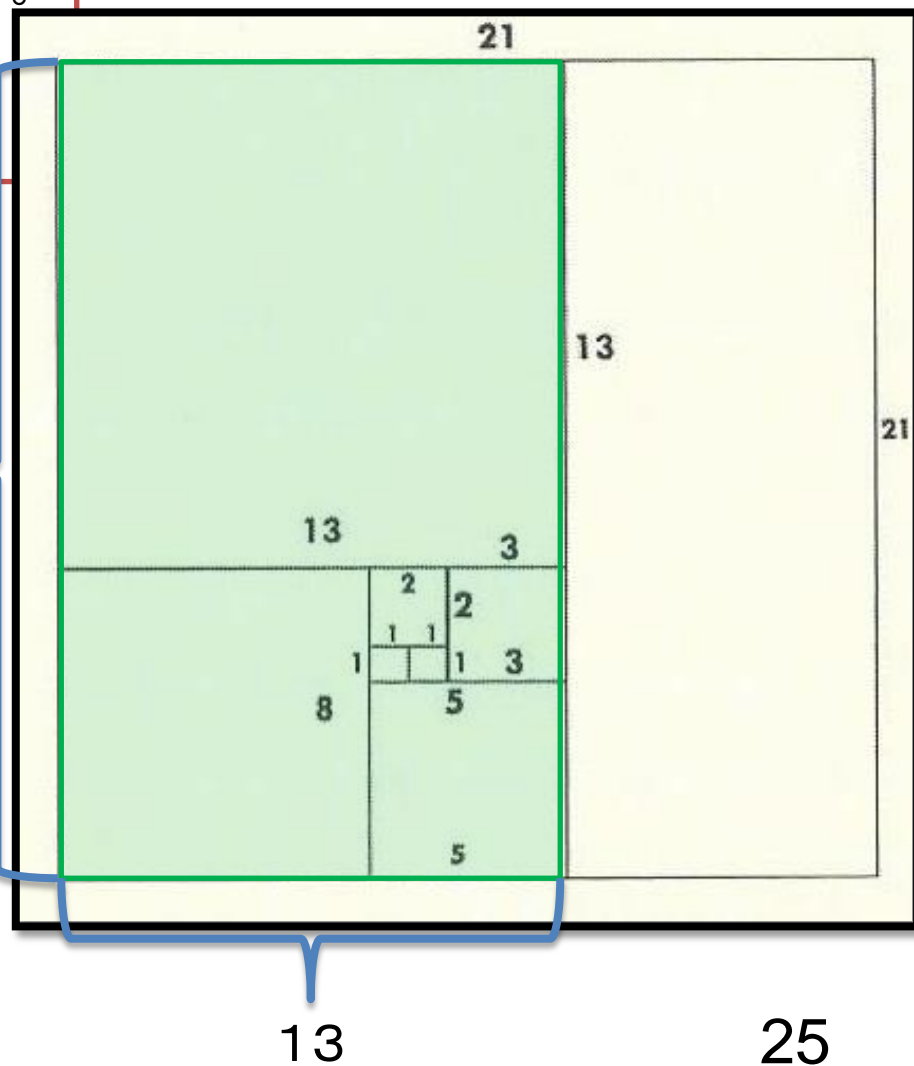
外側の長方形の面積

正方形の面積の和

21

この正方形の対角線を結ぶ弧を結んでいくと延々と螺旋を描くことができる

黄金らせん



13

25

フィボナッチ数列の性質の例示⑨

⑨ 2つの連続するフィボナッチ数の平方の和は、
その番号の和を番号とするフィボナッチ数に等しい。

$$F_n^2 + F_{n+1}^2 = F_{2n+1}$$

例1) $n=5$

$$\begin{aligned} F_5^2 + F_6^2 &= 5^2 + 8^2 \\ &= 25 + 64 = 89 = F_{11} \end{aligned}$$

例2) $n=8$

$$\begin{aligned} F_8^2 + F_9^2 &= 21^2 + 34^2 \\ &= 441 + 1156 = 1597 = F_{17} \end{aligned}$$

基本性質

- ⑪ 交互的フィボナッチ数の2つの積は、
両者の間にあるフィボナッチ数の平方より1多いか少ないか、いずれかである。

$$F_{n-1}F_{n+1} = F_n^2 + (-1)^n$$

選んだフィボナッチ数の平方とそのフィボナッチ数から等距離にある
フィボナッチ数の積の差は、別のフィボナッチ数の平方である。(ただし $k \geq 1$)

$$F_{n-k}F_{n+k} - F_n^2 = \pm F_k^2$$

- ⑫ $m \times n$ 番目のフィボナッチ数 F_{mn} は、 m 番目のフィボナッチ数 F_m で割り切れる。

- ⑬ 連続するフィボナッチ数の積の和は、フィボナッチ数の平方に等しいか、
フィボナッチ数の平方より1小さい。

$$n \text{ が奇数のとき } \sum_{i=2}^{n+1} F_i F_{i-1} = F_{n+1}^2$$

$$n \text{ が偶数のとき } \sum_{i=2}^{n+1} F_i F_{i-1} = F_{n+1}^2 - 1$$

どの性質もすべてのフィボナッチ数で成り立つ

フィボナッチ差

フィボナッチ数列には公差は存在しないが
差を取ると(さらにその差を取ると)面白いことがわかる

元の数列	1	1	2	3	5	8	13	21	34	55	89
第1階差	0	1	1	2	3	5	8	13	21	34	
第2階差		0	1	1	2	3	5	8	13	21	
第3階差			0	1	1	2	3	5	8		
第4階差					0	1	1	2	3		

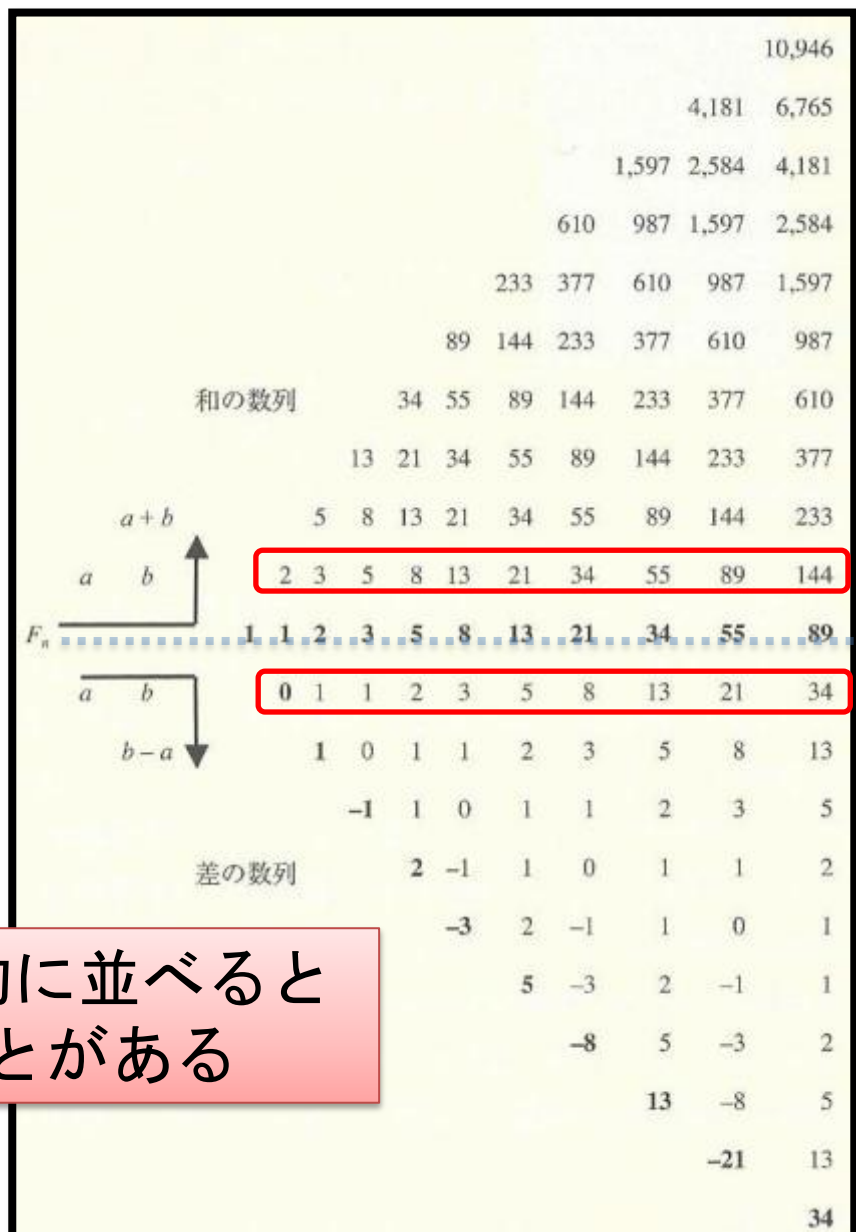
2項の差をとる

フィボナッチ数列
横と斜めに出現する

(筆者曰く)
このような性質がフィボナッチ数に
不思議な特性を与えている

フィボナッチ差と和

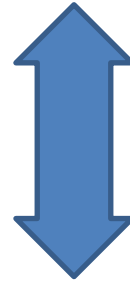
フィボナッチ差と同様に
 フィボナッチ和を計算してみると
 同様にフィボナッチ数列が出現する



フィボナッチ数を幾何学的に並べると
 特殊な性質が見えることがある

フィボナッチ数列の隣り合う項の比率は以下に収束する(黄金比)

$$\lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = 1.618033988749895 = \varphi$$



差がぴったり1

$$\lim_{n \rightarrow \infty} \frac{F_{n-1}}{F_n} = 0.618033988749895 = \frac{1}{\varphi}$$

すなわち

$$\frac{1}{\varphi} = \varphi - 1$$



この関係が成り立つ
唯一の正の値が
黄金比である

黄金比 $\varphi(=\frac{\sqrt{5}+1}{2})$ をべき乗してみると...

$$\varphi^2 = \left(\frac{\sqrt{5}+1}{2}\right)^2 = \frac{2\sqrt{5}+6}{4} = \frac{\sqrt{5}+3}{2} = \frac{\sqrt{5}+1}{2} + 1 = \varphi + 1$$

この関係を利用して次数を下げる



$$\varphi^3 = \varphi \times \varphi^2 = \varphi(\varphi + 1) = \varphi^2 + \varphi = (\varphi + 1) + \varphi = 2\varphi + 1$$

$$\begin{aligned}\varphi^4 &= \varphi^2 \times \varphi^2 = (\varphi + 1)(\varphi + 1) = \varphi^2 + 2\varphi + 1 \\ &= (\varphi + 1) + 2\varphi + 1 = 3\varphi + 2\end{aligned}$$

⋮

前スライドの計算から以下の結果を得る

$$\varphi = 1\varphi + 0$$

$$\varphi^2 = 1\varphi + 1$$

$$\varphi^3 = 2\varphi + 1$$

$$\varphi^4 = 3\varphi + 2$$

$$\varphi^5 = 5\varphi + 3$$

$$\varphi^6 = 8\varphi + 5$$

$$\varphi^7 = 13\varphi + 8$$

$$\varphi^8 = 21\varphi + 13$$

↓

$$\varphi^X = a\varphi + b$$

黄金比のべき乗は

黄金比のa倍と整数bの和で表現できる

そしてaとbはフィボナッチ数である

フィボナッチ数と黄金比には
密接な関係がある

フィボナッチ数から黄金比
黄金比からフィボナッチ数

- **連分数**: 分母が帯分数(整数と真分数の和) になっているもの

Ex.

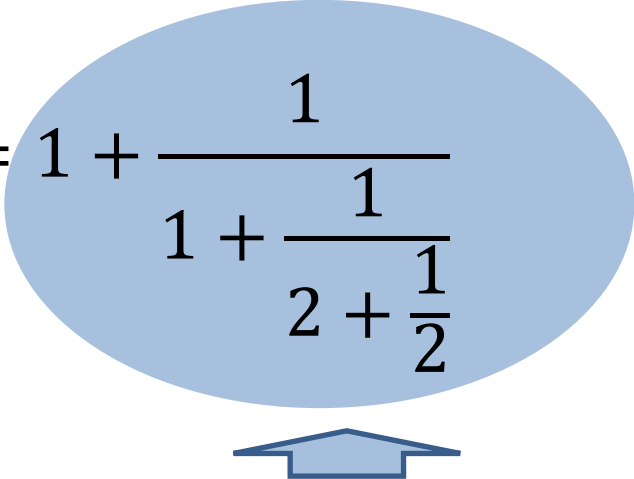
$$1\frac{6}{7} = 1 + \frac{6}{7} = 1 + \frac{1}{\frac{7}{6}} = 1 + \frac{1}{1 + \frac{1}{6}}$$

単位分数に達すると、基本的に終了

連分数とは(続き)

- 有限連分数: 有理数(単分数で表せる数)

Ex.

$$\frac{12}{7} = 1 + \frac{5}{7} = 1 + \frac{1}{\frac{7}{5}} = 1 + \frac{1}{1 + \frac{1}{\frac{5}{2}}} = 1 + \frac{1}{1 + \frac{1}{2 + \frac{1}{2}}}$$


- 無限連分数: 無理数

Ex.

$$\sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}} = [1; 2, 2, 2, 2, \dots] = [1; \bar{2}]$$

和を校正する成分(近似分数)に分けると、元の分数の値に近づく

- $1 = 1$
- $2 = 1 + \frac{1}{1} = 2$
- $\frac{3}{2} = 1 + \frac{1}{1 + \frac{1}{1}} = 1.5$
- $\frac{5}{3} = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}} = 1.\bar{6}$
- $\frac{8}{5} = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}}} = 1.6$

フィボナッチ数列の比は
連分数で表すと、
すべて1で表現可能

$$x = \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \dots}}}}$$

両辺を2乗すると、

$$x^2 = 1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \dots}}}$$

このことから、

$$x^2 - x - 1 = 0$$

この x の値のうち正になるのは、

$$\frac{1 + \sqrt{5}}{2} = \varphi$$

フィボナッチ数列



ビネーの公式

$$F_n = \frac{1}{\sqrt{5}} \left[\varphi^n - \left(-\frac{1}{\varphi} \right)^n \right]$$

Ex.

$$\begin{aligned} F_{128} &= \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{128} - \left(\frac{1-\sqrt{5}}{2} \right)^{128} \right] \\ &= 251,728,825,683,549,488,150,424,261 \end{aligned}$$

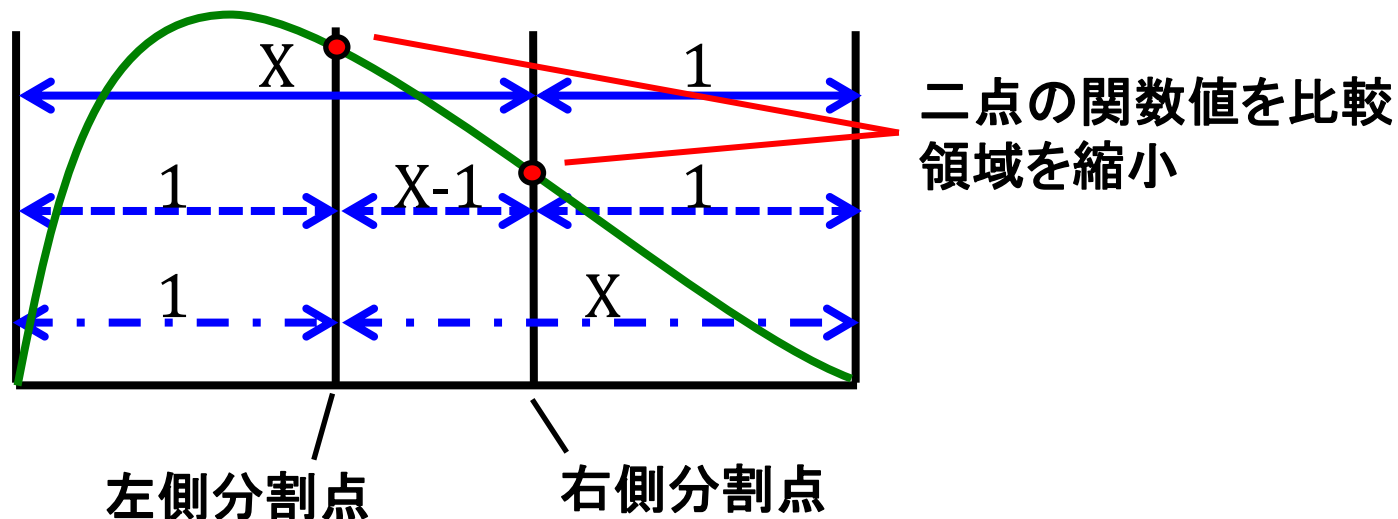
単峰関数の極値を求めるアルゴリズム

ある決まった区間の二点の関数値を比較
解の存在領域を縮小させ極値を見つける

一定の割合で無限に分割する条件 $\Rightarrow x : 1 = 1 : x - 1$

すなわち $x^2 - x - 1 = 0$ を満たす x の値は？

この x の値は **黄金比 (=1.618...)** である！



黄金分割法の動作

黄金分割法

単峰関数

X:黄金比

極値

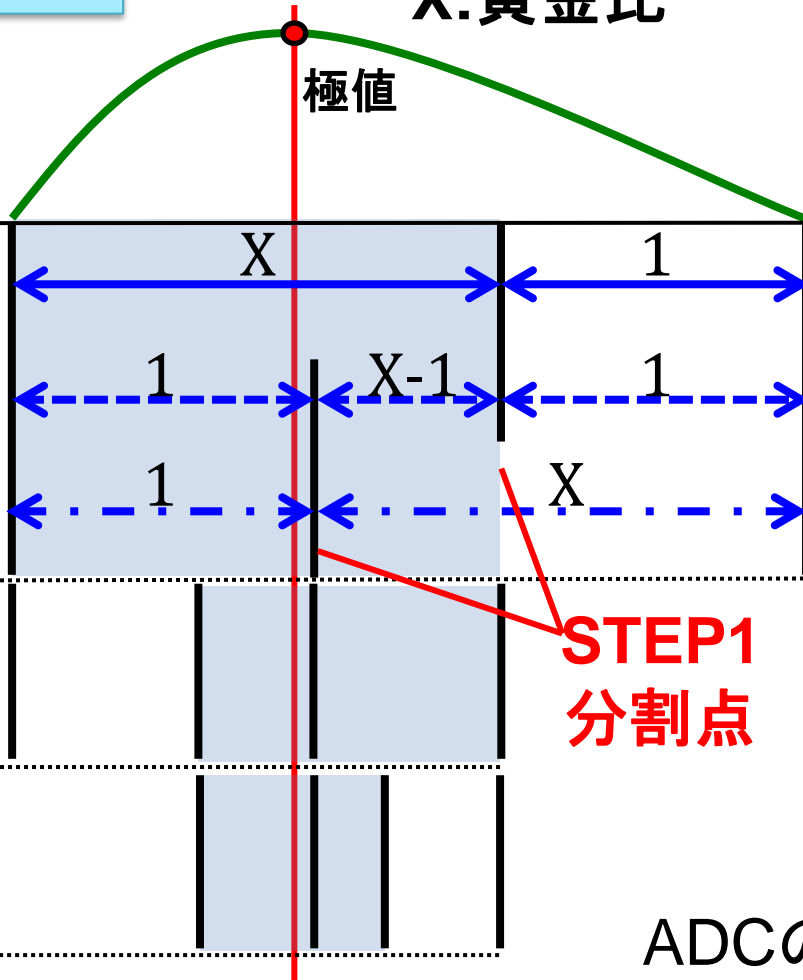
STEP1
判定:左

STEP2
判定:右

STEP3
判定:左

⋮

解



分割を繰り返すことで
極値存在可能範囲縮小



AD変換器へ応用

極値 = 入力アナログ値
比較分割を繰り返せば
デジタル値を得る

STEP1
分割点

ADCの構成は整数を扱う

⇒黄金比率は実現が難しい

フィボナッチ探索法

黄金分割法を整数のみで行う唯一の方法：フィボナッチ探索法

フィボナッチ数の性質

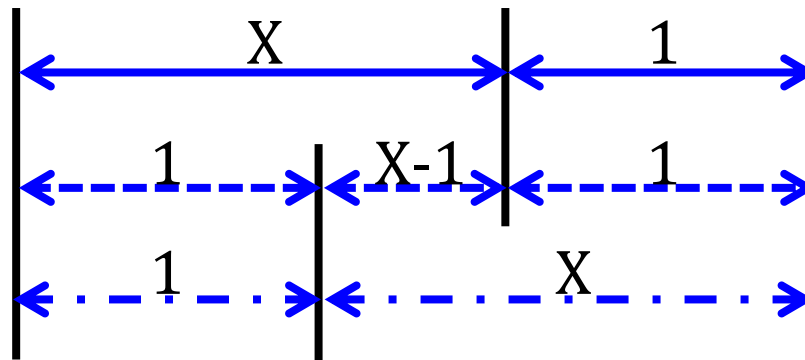
- ・隣り合う二項の和が次の項になる
- ・隣り合う項の比率が黄金比となる



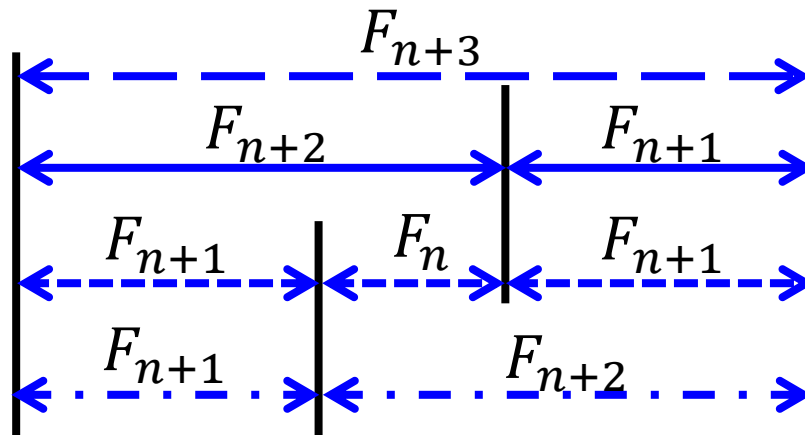
フィボナッチ探索法

整数により収束を約束
黄金探索法ADCを実現

黄金分割法
(X :黄金比)



フィボナッチ探索法
(F_x :フィボナッチ数)



リュカ数列の定義

$$L_0 = 2$$

$$L_1 = 1$$

$$L_{n+2} = L_n + L_{n+1}$$



Édouard Lucas
(1842-1891)

初めの項を計算すると(リュカ数)

(2,) 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843,
1364, 2207, 3571, 5778, 9349, 15127...

※はじめの2を数列の値として入れないこともある

隣り合う項の比率は以下の値 φ に収束する(黄金比)

$$\lim_{n \rightarrow \infty} \frac{L_n}{L_{n-1}} = 1.618033988749895 = \varphi$$

基本性質

⑭最初のn個のリュカ数の和は、二つ後のリュカ数から3を引いた数になる。

$$\sum_{i=1}^n L_i = L_1 + L_2 + L_3 + L_4 + \cdots + L_n = L_{n+2} - 3$$

例) n=4のとき

$$1+3+4+7=15=18-3$$

⑮リュカ数の平方の和は、最後の数と次の数の積から2引いた数となる。

$$\sum_{i=1}^n L_i^2 = L_n L_{n+1} - 2$$

例) n=5のとき

$$1+9+16+49+121=196$$

$$11*18-2=196$$

トリボナッチ数列は以下のように定義される数列

$$T_0 = T_1 = 0$$

$$T_2 = 1$$

$$T_{n+3} = T_n + T_{n+1} + T_{n+2}$$

初めの項を計算すると

0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274,
504, 927, 1705, 3136, 5768, 10609...

隣り合う項の比率は以下に収束する(約1.8進)

$$\lim_{n \rightarrow \infty} \frac{T_n}{T_{n-1}} = 1.839286755214162$$

トリボナッチ数列(前3項の和)

$$T_0 = T_1 = 0, T_2 = 1$$

$$T_{n+3} = T_n + T_{n+1} + T_{n+2}$$

$$\lim_{n \rightarrow \infty} \frac{T_n}{T_{n-1}} = 1.839286$$

テトラナッチ数列(前4項の和)

$$T_0 = T_1 = T_2 = 0, T_3 = 1$$

$$T_{n+4} = T_n + T_{n+1} + T_{n+2} + T_{n+3}$$

$$\lim_{n \rightarrow \infty} \frac{T_n}{T_{n-1}} = 1.927562$$

ペンタナッチ数列(前5項の和)

$$H_0 = H_1 = H_2 = H_3 = 0, H_4 = 1$$

$$H_{n+5} = H_n + H_{n+1} + H_{n+2} + H_{n+3} + H_{n+4}$$

$$\lim_{n \rightarrow \infty} \frac{H_n}{H_{n-1}} = 1.965948$$

⋮

項が前n個の項の和の数列をn-ボナッチ数列と呼ぶ
これら隣り合う項の比率 α は黄金比 φ 以上で2より小さい

$$1.61803 \leq \alpha < 2$$

- フィボナッチ数には非常に多くの性質がありそれらは自然界や身の回りで利用されている
- まだ実用化されていない性質がたくさんある
- フィボナッチ数は工学に十分応用できる



Carolus Fridericus Gauss
(独：1777-1855)

「整数論は数学の女王である。」

カール・フリードリヒ・ガウス

過去の整数論

身近にあるが、謎が多く美しい。
他分野へ貢献しない孤高の学問。

現在の整数論

情報通信処理に応用(暗号化・符号論)
⇒ デジタル信号との相性良し

電子回路への整数論応用は未知の世界
今後大きな発見が待っている可能性

フィボナッチ数列と黄金比

逐次比較近似AD変換器 冗長アルゴリズム

小林佑太郎
楊志翔

Shaiful Nizam Mohyar
小林春夫



Kobayashi
Laboratory





自動車エレクトロニクスに注目が集まる



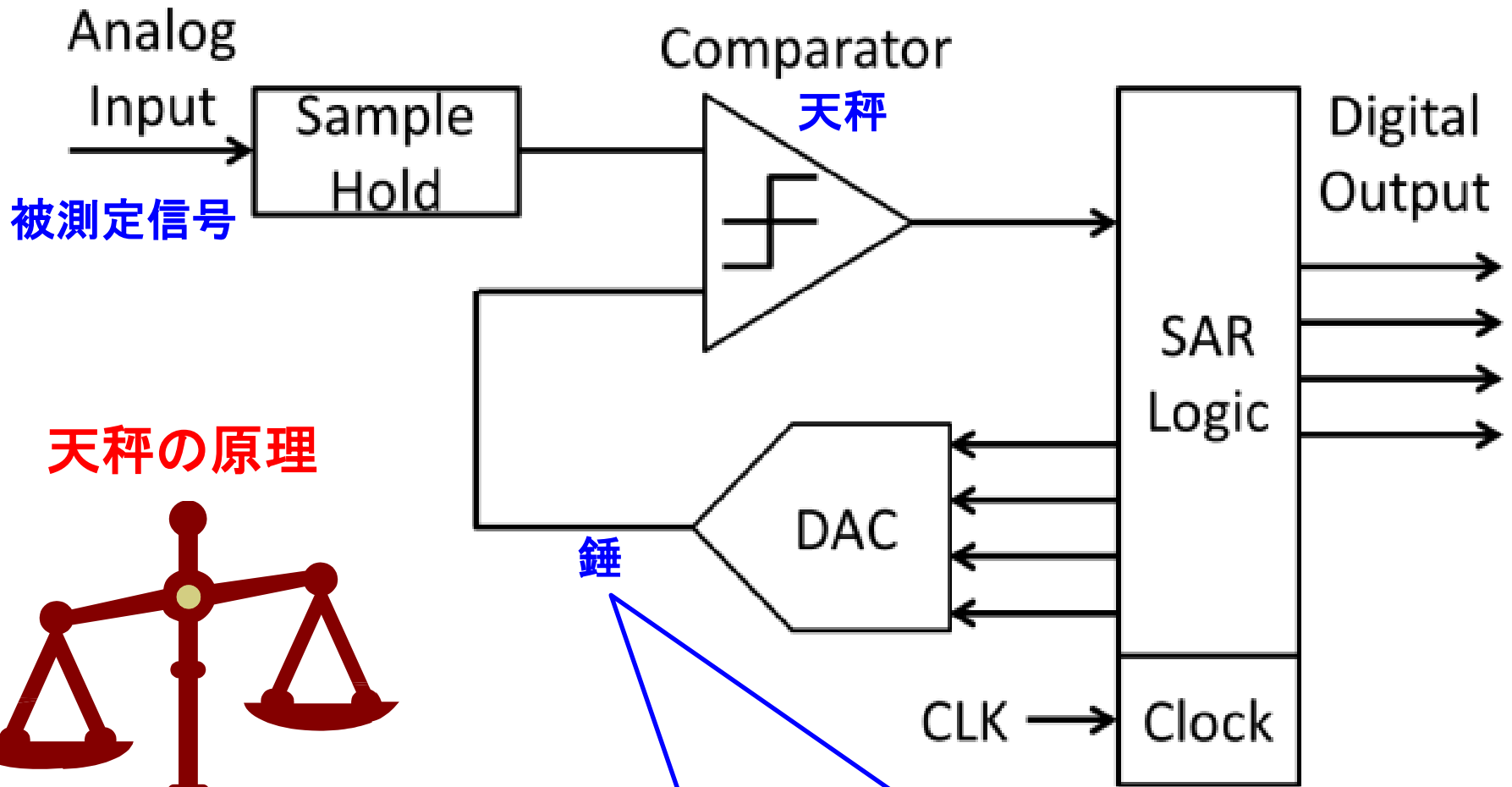
マイクロコントローラーには
従来よりも**高速**かつ**高信頼性**のSAR ADCが必要に



デジタル誤差補正可能な冗長設計で解決！

設計課題の存在 (☹)

逐次比較近似AD変換器



天秤の原理



一般的に 二進重みを利用
(1, 2, 4, 8, 16, 32, 64 ...)



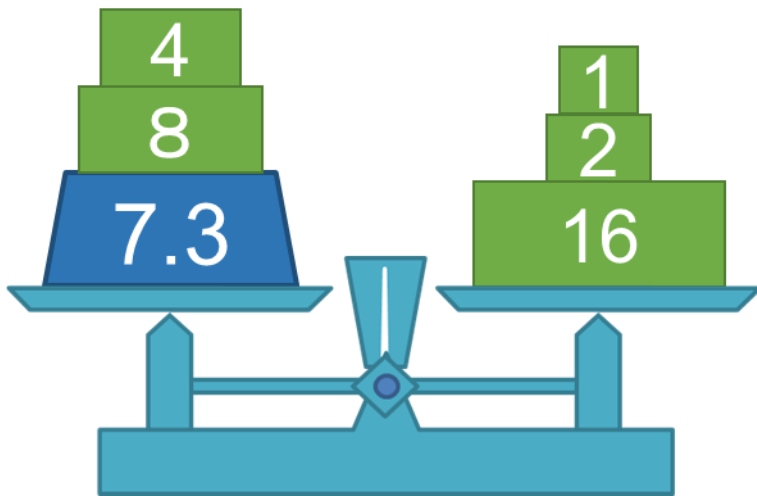
SAR ADCの二進探索動作(1)

5bit5step SAR ADC

- 二進重み

$$p(k) = 16, 8, 4, 2, 1$$

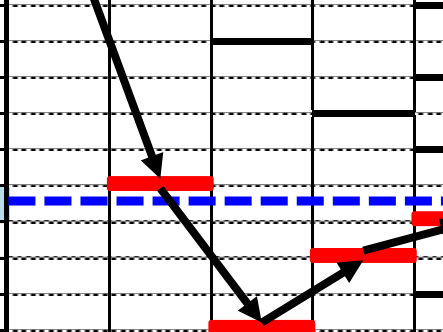
- 入力 7.3 LSB 相当



$$7.3 \Rightarrow 7$$

Step	1st	2nd	3rd	4th	5th	output
Weight p(k)	16	8	4	2	1	
31						31
30						30
29						29
28						28
27						27
26						26
25						25
24						24
23						23
22						22
21						21
20						20
19						19
18						18
17						17
16						16
15						15
14						14
13						13
12						12
11						11
10						10
9						9
8						8
7						7
6						6
5						5
4						4
3						3
2	0	0	1	1	1	2
1						1
0						0

Level



SAR ADCの二進探索動作(2)

5bit 5step SAR ADC

- 二進重み

$$p(k) = 16, 8, 4, 2, 1$$

- 入力 7.3 LSB 相当

二進数と十進数が1対1に対応

$$D_{out} = (00111)_2$$

$$7 = 16 - 8 - 4 + 2 + 1 + 0.5 - 0.5$$

Step	1st	2nd	3rd	4th	5th	output
Weight p(k)	16	8	4	2	1	
31						31
30						30
29						29
28						28
27						27
26						26
25						25
24						24
23						23
22						22
21						21
20						20
19						19
18						18
17						17
16						16
15						15
14						14
13						13
12						12
11						11
10						10
9						9
8						8
7						7
6						6
5						5
4						4
3						3
2	0	0	1	1	1	2
1						1
0						0

Level

SAR ADCの二進探索動作(3)

5bit5step SAR ADC

- 二進重み

$$p(k) = 16, 8, 4, 2, 1$$

- 入力 7.3 LSB 相当

二進数と十進数が1対1に対応

$$D_{out} = (00111)_2$$



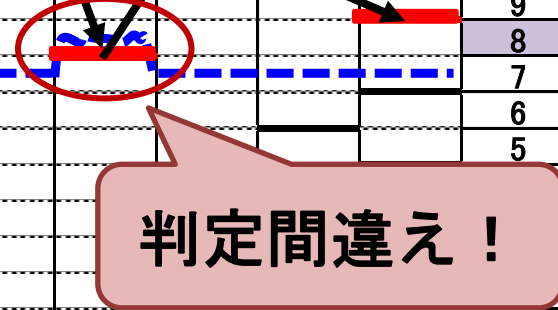
一回の判定誤りが
直接出力間違いとなる

$$D_{out} = (01000)_2 = 8$$

信頼性の劣化

Step	1st	2nd	3rd	4th	5th	output
Weight p(k)	16	8	4	2	1	
31						31
30						30
29						29
28						28
27						27
26						26
25						25
24						24
23						23
22						22
21						21
20						20
19						19
18	0	1	0	0	0	18
17						17
16						16
15						15
14						14
13						13
12						12
11						11
10						10
9						9
8						8
7						7
6						6
5						5
4						4
3						3
2						2
1						1
0						0

Level



判定間違い!

冗長: 余分や余裕のこと

↓ SAR ADCへ適用

時間冗長性を利用

- ◆ 比較判定回数を増加
- ◆ 比較する電圧を変更

↓

デジタルコードによる
表現方法増加

↓

デジタル誤差補正の実現

二進重み

$p(k): 1, 2, 4, 8, 16$



↓

非二進重み

$p(k): 1, 2, 3, 6, 10, 16$



SAR ADCの冗長探索動作(1)

5bit6step SAR ADC

➤ 冗長重み

$$p(k) = 16, 10, 6, 3, 2, 1$$

➤ 入力 6.3 LSB 相当

判定ステップの増加

$$6 \Rightarrow 010001 \Rightarrow 6$$

$$16 - 10 + 6 - 3 - 2 - 1 + 0.5 - 0.5 = 6$$

Step	1st	2nd	3rd	4th	5th	6th	output
Weight p(k)	16	10	6	3	2	1	
31							31
30							30
29							29
28							28
27							27
26							26
25							25
24							24
23							23
22							22
21							21
20							20
19							19
18							18
17							17
16							16
15							15
14							14
13							13
12							12
11							11
10							10
9							9
8							8
7							7
6							6
5							5
4							4
3	0	1	0	0	0	1	3
2							2
1							1
0							0

Level

0 1 0 0 0 1

SAR ADCの冗長探索動作(2)

5bit6step SAR ADC

➤ 冗長重み

$$p(k) = 16, 10, 6, 3, 2, 1$$

➤ 入力 6.3 [LSB]

判定ステップの増加

$$6 \Rightarrow 010001 \Rightarrow 6$$



デジタルコード表現が複数

$$6 \Rightarrow 001111 \Rightarrow 6$$

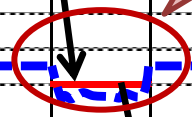
デジタル誤差補正

高信頼性の実現

Step	1st	2nd	3rd	4th	5th	6th	output
Weight p(k)	16	10	6	3	2	1	
31							31
30							30
29							29
28							28
27							27
26							26
25							25
24							24
23							23
22							22
21							21
20							20
19							19
18							18
17							17
16							16
15							15
14							14
13							13
12							12
11							11
10							10
9							9
8							8
7							7
6							6
5							5
4							4
3	0	0	1	1	1	1	3
2							2
1							1
0							0

Level

判定間違い!



補正力を比較するための
評価方法が必要

補正能力 $q(k)$ を定義

一回誤っても後段で補正できる
入力電圧と比較電圧の絶対差

1st step 補正能力
 $q(1)=3$

Step	1st	2nd	3rd	4th	5th	6th	output
Weight $p(k)$	16	10	6	3	2	1	
31							31
30							30
29							29
28							28
27							27
26							26
25							25
24							24
23							23
22							22
21							21
20							20
19							19
18							18
17							17
16							16
15							15
14							14
13							13
12							12
11							11
10							10
9							9
8							8
7							7
6							6
5							5
4							4
3							3
2							2
1							1
0							0

出力は18まで戻せる

判定間違え

Level

$q(1)$

補正力を比較するための
評価方法が必要

補正能力q(k)を定義

一回誤っても後段で補正できる
入力電圧と比較電圧の絶対差

補正可能である条件

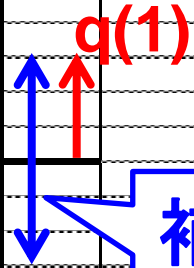
$$|V_{in} - V_{ref}(k)| \leq q(k)$$



q(k)が大きい=補正力高い

Step	1st	2nd	3rd	4th	5th	6th	output
Weight p(k)	16	10	6	3	2	1	
31							31
30							30
29							29
28							28
27							27
26							26
25							25
24							24
23							23
22							22
21							21
20							20
19							19
18							18
17							17
16							16
15							15
14							14
13							13
12							12
11							11
10							10
9							9
8							8
7							7
6							6
5							5
4							4
3							3
2							2
1							1
0							0

Level



補正可能な
入力範囲

補正力を比較するための
評価方法が必要

k step目で判定誤りがあっても
後段で補正が可能な入力範囲差q(k)

$$q(k) = -p(k + 1) + 1 + \sum_{i=k+2}^M p(i)$$

p(k): k step目の比較重み
M: 総ステップ数

補正力は比較重みp(k)によって決まる

どのような比較重みを利用するか？



Step	1st	2nd	3rd	4th	5th	6th	output
Weight p(k)	16	10	6	3	2	1	
31			↓				31
30							30
29							29
28							28
27							27
26		↕	▲ q(2)				26
25		↕					25
24							24
23							23
22							22
21							21
20			↕	▲ q(3)			20
19			↕				19
18	↕	▲ q(1)					18
17	↕						17
16							16
15							15
14							14
13	↕						13
12			↕				12
11			↕				11
10							10
9							9
8							8
7							7
6		↕					6
5		↕					5
4							4
3							3
2							2
1							1
0			↑				0

N bit 全 M step 中 k step 目の比較重み $p(k)$ を決定

従来手法

① 基数radixから決定する $\Rightarrow p(k) = r^{M-k}$ $(1 <$

$r < 2)$

- 適切な基数の決定が難しい
 - 補正力(信頼性)と変換ステップ数(時間)はトレードオフ
- $p(k)$ は必ず小数になる(単位セルによる実現困難)
 - 面積比を基数比で利用 \Rightarrow 精度が悪い
 - 四捨五入して整数を使う \Rightarrow $q(k)$ のばらつき

② 最も適当な $p(k)$ を任意に決定する

- 適切な効果を得づらい
- 決定が難しく設計時間を増加

5bit 6step SAR ADC 冗長設計手法① radix=1.80

比較電圧重み $p(k)$

$$p(1) = 2^{5-1} = 16$$

$$p(2) = 1.8^4 \cong 10$$

$$p(3) = 1.8^3 \cong 6$$

$$p(4) = 1.8^2 \cong 3$$

$$p(5) = 1.8^1 \cong 2$$

$$p(6) = 1.8^0 = 1$$

絶対に補正できない入力範囲

冗長設計効果の減少



適切な $p(k)$ 選択手法が重要

Step	1st	2nd	3rd	4th	5th	6th	output
Weight $p(k)$	16	10	6	3	2	1	
31			↓				31
30							30
29							29
28							28
27							27
26		↕	▲ $q(2)$				26
25							25
24							24
23							23
22							22
21							21
20			↕	▲ $q(3)$			20
19							19
18	↕	▲ $q(1)$					18
17							17
16							16
15							15
14	↕						14
13							13
12			↕				12
11							11
10							10
9							9
8							8
7							7
6		↕					6
5							5
4							4
3							3
2							2
1							1
0			↑				0

Level

フィボナッチ数列

$$F_0 = 0$$

$$F_1 = 1$$

$$F_{n+2} = F_n + F_{n+1}$$

初めの項を計算

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

⇒ フィボナッチ数と呼ばれる



Leonardo Fibonacci
(伊: 1170-1250年頃)

また隣り合う項の比率は以下の値 φ に収束

$$\lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = 1.618033988749895 = \varphi$$

収束比率 φ : 黄金比 (Golden ratio)

比較電圧重み $p(k)$ の決定(提案手法)

N bit 全 M step 中 k step 目の比較重み $p(k)$ を決定

提案手法

フィボナッチ数を $p(k)$ として利用する $\Rightarrow p(k) = F_{M-k+1}$

Binary Weight
二進数

64	32	16	8	4	2	1
----	----	----	---	---	---	---

Radix 1.8 Weight
1.8進数

34.0	18.9	10.5	5.8	3.2	1.8	1
------	------	------	-----	-----	-----	---

Fibonacci Weight
約1.62進数

13	8	5	3	2	1	1
----	---	---	---	---	---	---

隣り合う項の比率が黄金比 φ に収束

\Rightarrow 整数のみで約1.62進数($radix = 1.62$)を実現できる!

フィボナッチ数列SAR ADC

2点の性質を新発見！

- ① 許容値 $q(k)$ は必ずフィボナッチ数
- ② 許容できる範囲が必ず接する

Step	1st	2nd	3rd	4th	5th	6th	7th
Weight p(k)	16	8	5	3	2	1	1
33					↓		
32				↕			
31				↕			
30			↕		↕		
29			↕		↕		
28			↕		↕		
27			↕		↕		
26		↕		↕			
25		↕		↕			
24		↕		↕			
23		↕		↕			
22		↕		↕			
21		↕		↕			
20	↕		↕		↕		
19	↕		↕		↕		
18	↕		↕		↕		
17	↕		↕		↕		
16	↕		↕		↕		
15	↕		↕		↕		
14	↕		↕		↕		
13	↕		↕		↕		
12	↕		↕		↕		
11	↕		↕		↕		
10	↕	↕		↕			
9	↕	↕		↕			
8	↕	↕		↕		↕	
7	↕	↕		↕		↕	
6	↕	↕		↕		↕	
5	↕	↕		↕		↕	
4	↕	↕	↕		↕		
3	↕	↕	↕		↕		
2	↕	↕	↕		↕		
1	↕	↕	↕		↕		
0	↕	↕	↕	↕			
-1	↕	↕	↕	↕			
-2	↕	↕	↕	↕	↕		

Level

$q(1)$

$q(2)$

$q(3)$

$q(4)$

$q(5)$

フィボナッチ数列SAR ADC

2点の性質を新発見！

- ① 許容値 $q(k)$ は必ずフィボナッチ数
- ② 許容できる範囲が必ず接する



フィボナッチ数重み($r = 1.618$)より...

- ・基数 r が大きい \Rightarrow $q(k)$ は離れる
- ・基数 r が小さい \Rightarrow $q(k)$ は重なる



黄金比 ϕ は
冗長さの境界条件であり
設計指針となる

Step	1st	2nd	3rd	4th	5th	6th	7th
Weight $p(k)$	16	8	5	3	2	1	1
33					↓		
32				↕			
31				↕			
30			↕		↕		
29			↕		↕		
28			↕		↕		
27			↕		↕		
26		↕		↕			
25		↕		↕			
24		↕		↕			
23		↕		↕			
22		↕		↕			
21		↕		↕			
20	↕	↕		↕			
19	↕	↕		↕			
18	↕	↕		↕			
17	↕	↕		↕			
16	↕	↕		↕			
15	↕	↕		↕			
14	↕	↕		↕			
13	↕	↕		↕			
12	↕	↕		↕			
11	↕	↕		↕			
10	↕	↕		↕			
9	↕	↕		↕			
8	↕	↕		↕			
7	↕	↕		↕			
6	↕	↕		↕			
5	↕	↕		↕			
4	↕	↕		↕			
3	↕	↕		↕			
2	↕	↕		↕			
1	↕	↕		↕			
0	↕	↕		↕			
-1	↕	↕		↕			
-2	↕	↕		↕			

$q(1)$

$q(2)$

$q(3)$

$\Delta q(4)$

$\Delta q(5)$

Level

フィボナッチ数列SAR ADC

2点の性質を新発見！

- ① 許容値 $q(k)$ は必ずフィボナッチ数
- ② 許容できる範囲が必ず接する



接する境界で
すべての入力範囲をもれなく
カバーすることになる



黄金比 ϕ を使うことで

- ・無駄なステップ
- ・補正できない入力範囲

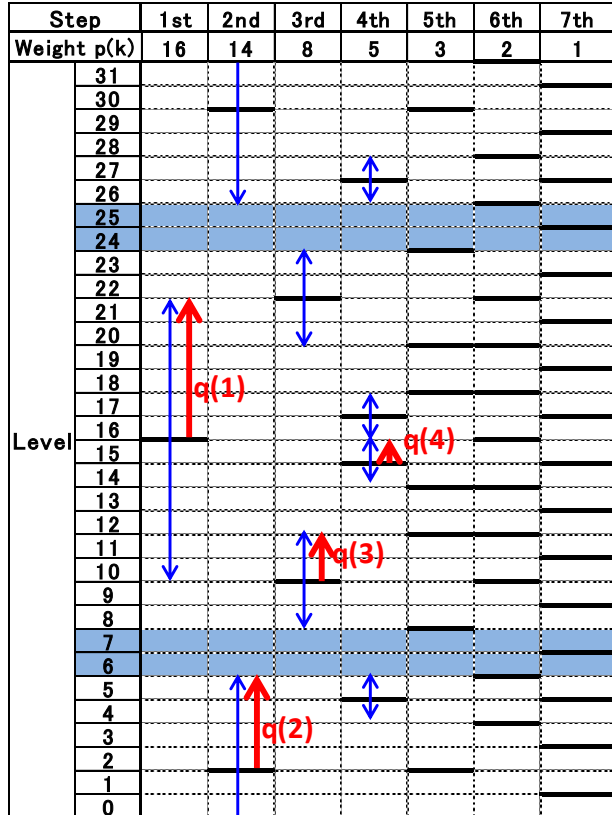
がない最も効率のよい設計が可能

Step	1st	2nd	3rd	4th	5th	6th	7th
Weight p(k)	16	8	5	3	2	1	1
33					↓		
32				↕			
31				↕			
30			↕		↕		
29			↕		↕		
28			↕		↕		
27			↕		↕		
26		↕		↕			
25		↕		↕			
24		↕		↕			
23		↕		↕			
22		↕		↕			
21		↕		↕			
20	↕	↕		↕			
19	↕	↕		↕			
18	↕	↕		↕			
17	↕	↕		↕			
16	↕	↕		↕			
15	↕	↕		↕			
14	↕	↕		↕			
13	↕	↕		↕			
12	↕	↕		↕			
11	↕	↕		↕			
10	↕	↕		↕			
9	↕	↕		↕			
8	↕	↕		↕			
7	↕	↕		↕			
6	↕	↕		↕			
5	↕	↕		↕			
4	↕	↕		↕			
3	↕	↕		↕			
2	↕	↕		↕			
1	↕	↕		↕			
0	↕	↕		↕			
-1	↕	↕		↕			
-2	↕	↕		↕			

従来手法との比較(5bit ADC)

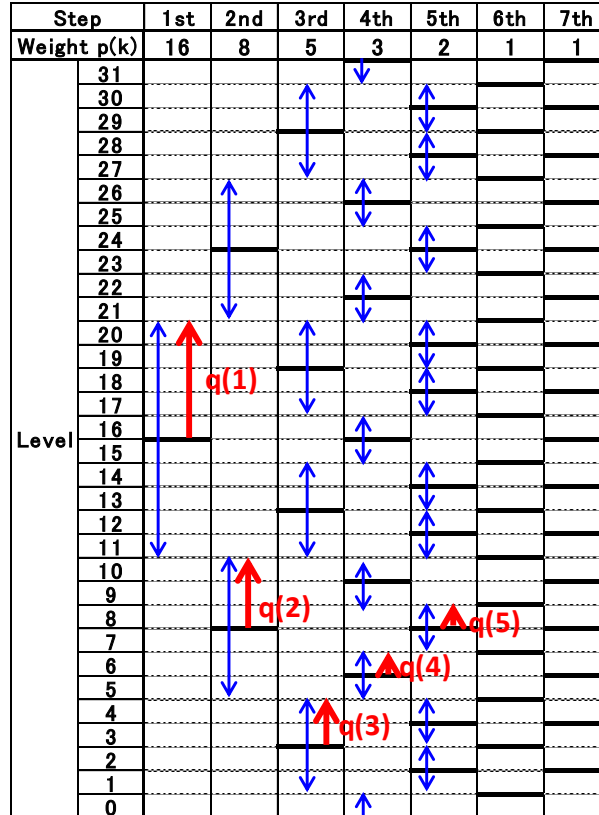
従来手法

1.70進数



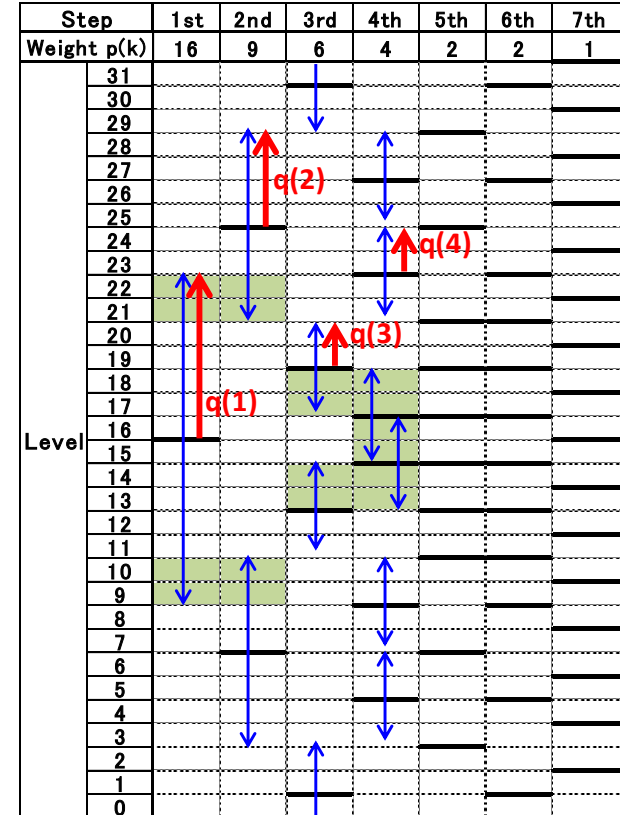
提案手法

1.62進数



従来手法

1.55進数

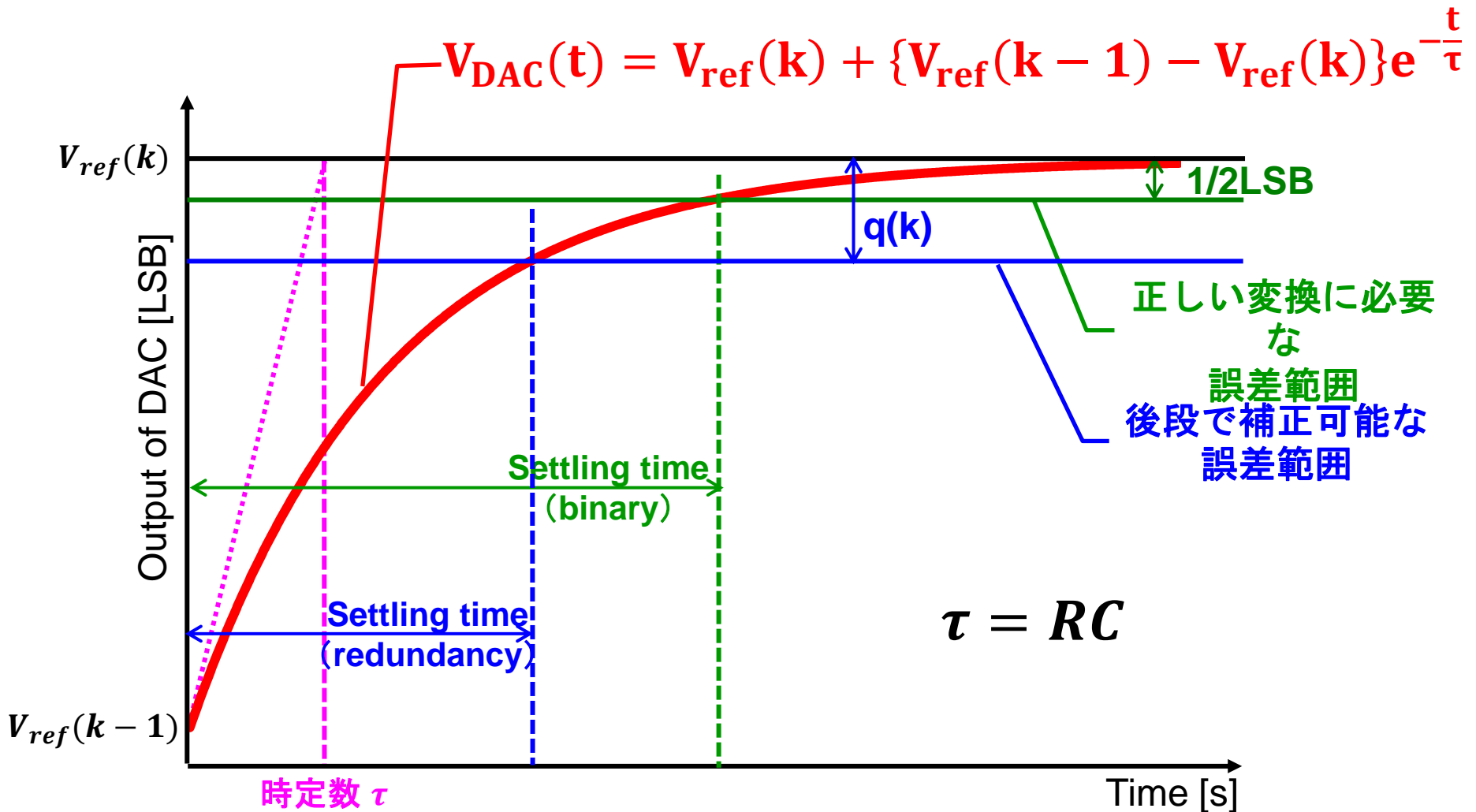


フィボナッチ数列冗長手法

冗長基数の境界条件
効率の良い基準重み

DAC出力の整定時間

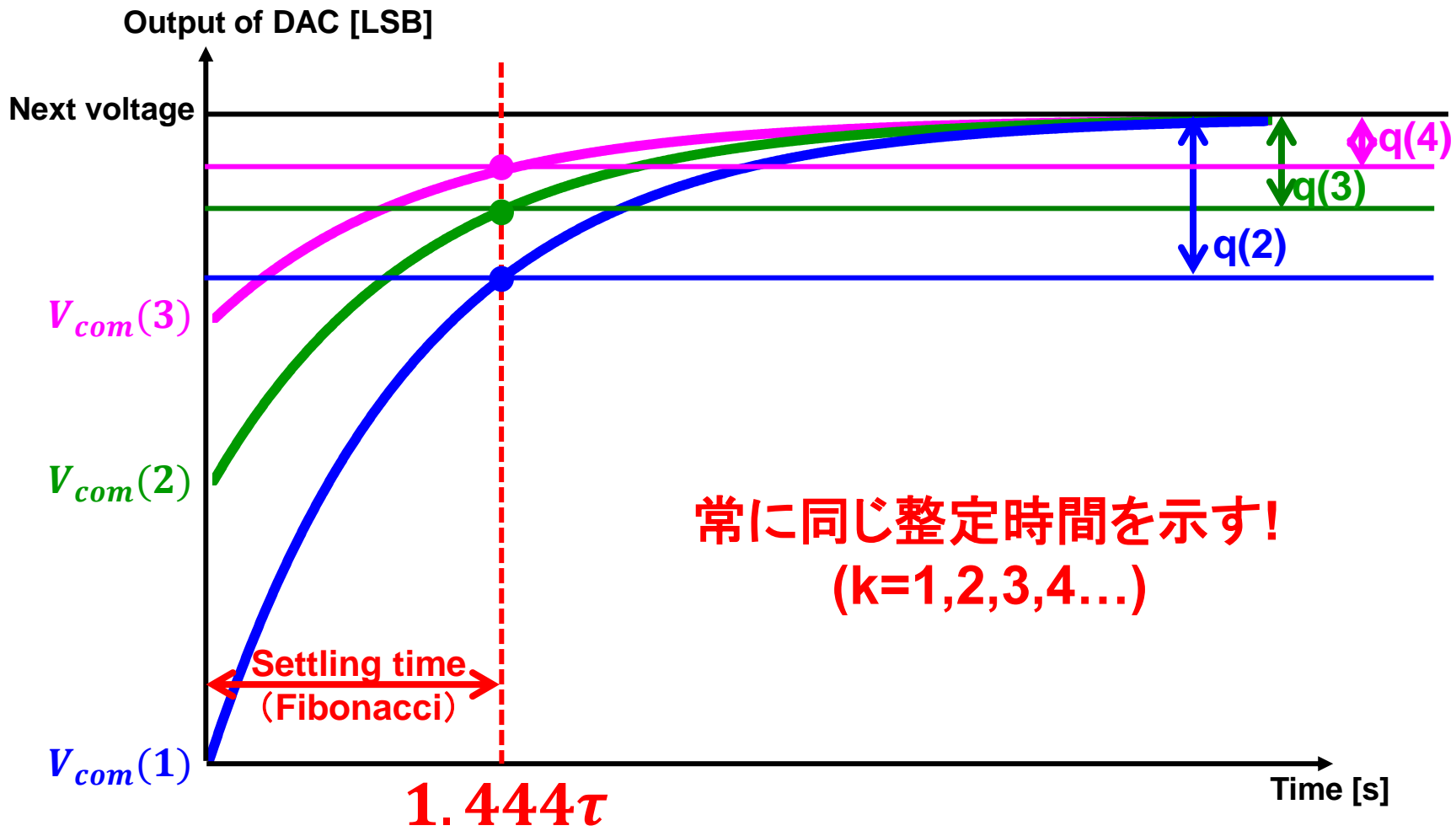
RCモデルでDAC出力の整定を考える



フィボナッチ手法の整定時間の性質

新たな性質の発見！

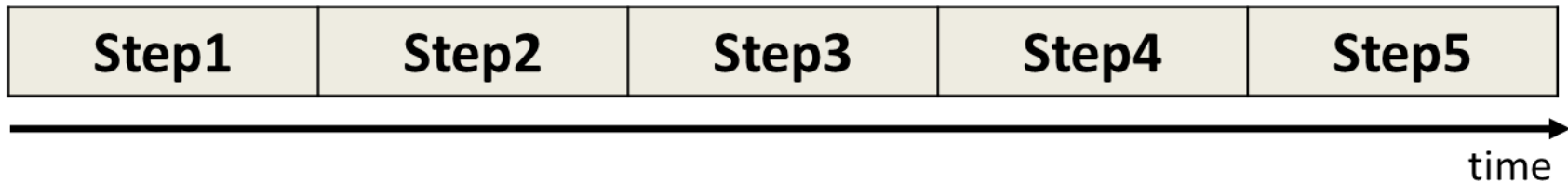
$$T_{settle}(k) = \tau \ln(2\varphi + 1) \\ = 1.444\tau \quad \text{for all } k$$



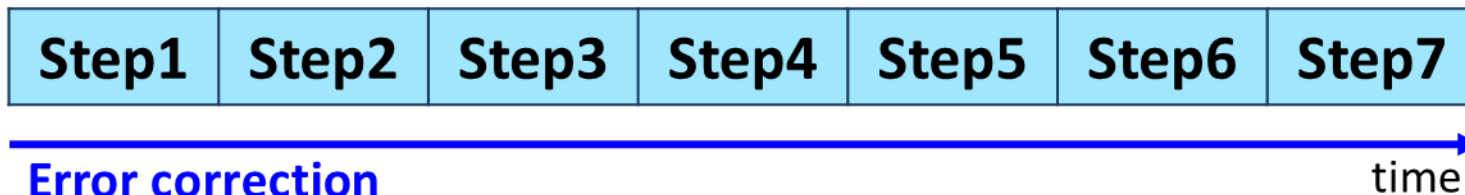
デジタル誤差補正 → DAC 不完全整定に対処

5bit SAR ADC

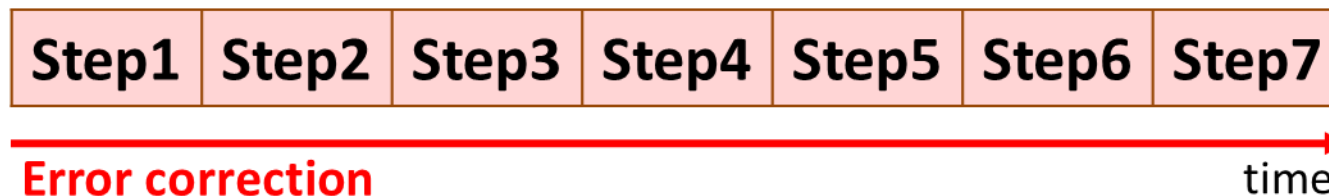
Binary search (complete settling)



Redundant search (incomplete settling)



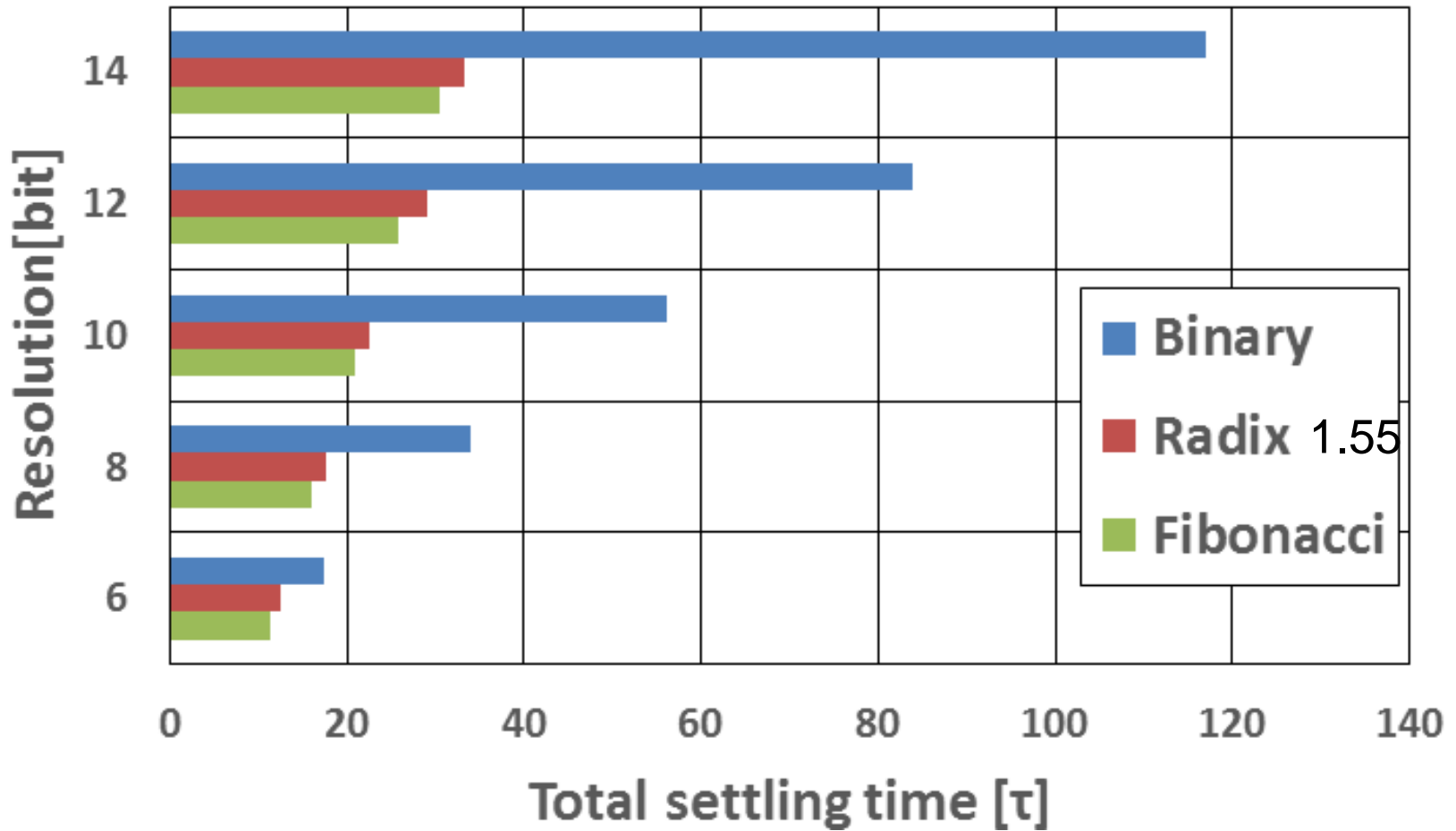
Fibonacci search (incomplete settling)



最短
AD変換時間
の実現！

AD変換時間 結果のまとめ

固定クロックにおける変換時間の比較(分解能別)



固定クロックでは....

どの分解能でもフィボナッチ手法SAR ADCが最も早い

- 冗長SAR ADCの設計手法を提案
- フィボナッチ数列を利用することで重要な性質を得た
 - **高信頼性の実現**
広い入力範囲を補正できる
 - **最速のSAR AD変換**
固定クロック、DAC不完全整定の条件下
 - **基数の基準**
黄金比 $\varphi (= 1.618)$ は基数の基準
 - **一定整定時間**
DAC出力の要求整定時間はすべてのステップで一定

フィボナッチ数列、黄金比と
冗長逐次比較近似AD変換アルゴリズム
を結びつける

Creativity is just connecting things.
「創造力とは、いろいろなものをつなぐ力だ」
(Steve Jobs, Apple社)



フィボナッチ数列でソーラーパネル生成エネルギーを向上

中学生 Aidan Dwyer 君。

冬に葉を落とした木の枝を見てその螺旋状のパターンに気づく。

調べていくうちにフィボナッチ数列に行き着く。

樅の木のフィボナッチパターンを模倣した実験モデルを作り調べる。

Dwyer 君はこのモデルをもとに塩化ビニル管から枝を作り

葉の部分に PV ソーラーパネルを取り付けたソーラー発電装置を作成。

既存の平面パネルのソーラー発電装置と比較。

提案発電装置の電力発生量はフラットパネルと比較して 20 % 多く、

太陽光を集める時間も 2 時間半長い。

冬期には差は開き、提案装置は 50 % 多くの電力を作り、

太陽光を集める時間も 50 % 増加。

アメリカ自然史博物館 「Young Naturalist Award」を受賞