

# 有限群解析プログラム

## ※全体的に加筆中

作製: 立石卓也

2021 年 1 月 27 日

## 目次

1	本プログラムについて	2
1.1	利用規約 . . . . .	2
1.2	利用方法 . . . . .	2
1.3	概要 . . . . .	2
1.4	具体的な使用手順 . . . . .	3
2	資料内での用語	3
3	コマンド一覧	4
3.1	引数に群を一つとるコマンド . . . . .	4
3.2	未実装（実装を検討中） . . . . .	5
4	有限群論	6
4.1	記号など . . . . .	6
4.2	位数 . . . . .	6
4.3	共役類 . . . . .	6
4.4	群の中心 . . . . .	7
4.5	正規部分群 . . . . .	8
4.6	導来部分群 . . . . .	8
5	処理の概要	10
5.1	群の生成 . . . . .	10
5.2	乗積表の作成 . . . . .	10
5.3	共役類の計算 . . . . .	10

# 1 本プログラムについて

## 1.1 利用規約

本プログラムやその計算結果を用いたことによるいかなる不利益に対して、作製者は一切の責任を負いません。

## 1.2 利用方法

本プログラムは、Python 3.x で書かれた一連のソースファイルです。利用には Python 3.x の実行環境、および標準ライブラリが必要となります。Python の実行環境は各自で用意してください。

本プログラムは、使用するにあたり Python のソースコードを（数行程度）ユーザー自身が編集する必要があります。そのため、Python のコードを書くための基本的な知識、および数値計算用ライブラリ NumPy の知識が必要となります。知識の目安としては、3 次程度の複素正方行列を記述できれば十分です。

本プログラムのソースファイルは、以下の URL から入手することができます。ダウンロードした zip ファイルを解凍後、特定のソースファイルを Python で実行することで使用できます。インストールは不要です。

### ■配布場所

- <https://github.com/THTatsuishi/finite-group-analyzer>
- 上記 URL にアクセス後、「Code」 → 「Download Zip」 から zip ファイルをダウンロード

## 1.3 概要

有限群解析プログラム (finite-group-analyzer) は、有限群の生成および生成された群の構造を調べるためのプログラムです。本プログラムは、主に以下の機能を有しています。

- いくつかの複素行列を生成元として、有限群を生成する。  
全ての元のあらゆる組み合わせの積をとり、新しい元が現れなくなるまで生成を続ける。ただし、元の数が予め定めた値を超えた場合には、群が有限では閉じないものと判定する。
- 生成された有限群の性質を調べる。  
共役類、可換性、群の中心、正規部分群、群同型など。
- 生成された群の直積分解や半直積分解を調べる。

本プログラムでは、初めに行列表現で与えられた生成元から群を生成し、群の乗積表を作成します。この計算過程では、行列演算を繰り返すため、行列のサイズや群の位数に応じた計算時間がかかります。乗積表の作成後は、行列表現された元をナンバリングして整数に置き換えて乗積表に従った

演算を行うことで、極めて高速な計算を実現しています。基本的に、解析画面上での行う計算は短時間で完了します。

## 1.4 具体的な使用手順

1. Python 3.x の実行環境を整える。  
一つの選択肢として、Anaconda をインストールして統合開発環境 Spyder を用いる方法があります。
2. 本プログラムのソースファイルを入手する (§??参照)。
3. `finite-group-analyzer/starting-program/template.py` をコピーし、名前を変更して同じディレクトリに保存する。  
ファイル名は半角英数字、ハイフン (-)、アンダースコア (\_) のみの使用を推奨。  
コピー後のファイルを「起点ファイル」と呼ぶ。
4. 起点ファイルのソースコードを編集する。  
ソースコード内の説明に従い、以下の値を設定する。
  - 生成元となる行列を一つ以上。
  - 「有限では閉じない」と判定するための要素数の上限値。
  - 浮動小数点の計算誤差を補正するための許容誤差（基本的に編集不要）。
5. 起点ファイルを実行する。  
→ 有限群が正しく生成されない場合はエラーが発生する。
  - Python のコードの書き方が誤っている（特にインデントに注意せよ）。
  - 群が有限で閉じない。エラーの発生原因を解消して、再度起点ファイルを実行する。
6. 群が正しく生成された場合、解析用画面が開く。  
以降操作は解析画面上で行う。
7. 解析画面のコマンド入力欄に解析用のコマンドを入力する。
8. 解析を終了したら、解析画面を閉じる。

## 2 資料内での用語

■**master** ユーザーが指定した生成子から生成された群。本プログラムでは、この群とその部分群について調べることができる。

■**コマンド** 解析画面上で使用する、解析を実行するためのプログラムに対する命令文。全てのコマンドは、半角英数字で「コマンド名 [引数]」という形式をとる。大文字と小文字を区別する。

## 3 コマンド一覧

### 3.1 引数に群を一つとるコマンド

Command[group] の形式をとり, “group” で指定された群に関する処理を行う.

■?[group] 群の概要を表示する. 群の同定を行うだけであれば, このコマンドだけを使用すれば十分である.

■Element[group] 群の要素を表示する. 行列表現ではなく, 元ナンバリングした整数の一覧が表示される.

■Table[group] 群の乗積表を表示する. 単位元は必ず “0” となる. 位数が大きい場合には一部のみが表示されるため, 現状ではあまり役に立たない.

■ConjClass[group] 群の共役類 (位数, 要素数, 要素) の一覧を表示する.

■ConjCount[group] 群の共役類のカウント (位数, 要素数, 重複度) の一覧を表示する.

■Isomorphic[group] 群が名前の付いた群と同型の場合に, 群同型を表示する.

■IsAbelian[group] 群が可換群であるかを表示する.

■IsPerfect[group] 群が完全群であるかを表示する.

■IsSimple[group] 群が単純群であるかを表示する.

■IsSolvable[group] 群が可解群であるかを表示する.

■Center[group] 群の中心を表示する.

■Centralizer[group] master における群の中心化群を表示する.

■Normalizer[group] master における群の正規化群を表示する.

■Derived[group] 群の導来部分群を表示する.

■DerivedSeries[group] 群の導来列を表示する.

■Normal[group] 群の全ての正規部分群を表示する.

■DirectDecompose[group] 群' の全ての可能な直積分解を表示する.

$$\text{group} = \text{left} \times \text{right} \tag{1}$$

として, 二つの群に分解する. 位数の大きい方の群を (left) とする. 分解後の群は, さらに分解可能である場合がある.

■SemidirectDecompose[group] 群を半直積に分解する. 直積は含まず,

$$\text{group} = \text{left} \rtimes \text{right} \quad (2)$$

として, 右半直積として表現する. ただし, 商群 (right) は master の元による共役変換の範囲で一意である.

■Decompose[group] 群の直積分解および半直積分解を表示する.

### 3.2 未実装 (実装を検討中)

■Subgroup[group] 群の部分群を表示する. ただし, 部分群は master の元による共役変換の範囲で一意である.

## 4 有限群論

ほとんど部分で証明を書かずに命題だけを記している. 各自で裏付けをとっていただきたい. 特に, プログラム内で前提として使用している重要な命題を下線で示す.

### 4.1 記号など

以下の記号は, 文中で説明なく使用する.

- $\mathbb{N}_+$  : 0 を含まない自然数
- $e$  : 群の単位元
- $g^{-1}$  : 群の元  $g$  の逆元
- $=, \simeq$  : 集合の等価, 群の等価, 群同型など, あまり区別せずに使用する
- $Z_n$  :  $n$  次の巡回群
- $\{ \text{要素} \}$  : 集合
- $\langle \text{集合} \rangle$  : (集合) を生成系として生成される群

### 4.2 位数

#### 4.2.1 群の位数 (order)

群の要素数を指す. 群  $G$  の部分群を  $H \subset G$  とすると,  $H$  の位数は  $G$  の位数の約数である. よって, 位数が素数である群には非自明な部分群が存在しない.

#### 4.2.2 群の元の位数 (order)

群  $G$  の元  $g \in G$  に対して,

$$g^n = e, \quad n \in \mathbb{N}_+ \quad (3)$$

を満たす最小の  $n$  を指す. 元の位数は群の位数の約数である.

### 4.3 共役類

#### 4.3.1 共役変換

群  $G$  の元を  $g, h \in G$  とする.  $h$  による  $g$  の共役変換を

$$hgh^{-1} \quad (4)$$

とする. また,  $g, g' \in G$  に対して

$$g = hg'h^{-1} \quad (5)$$

となる  $h \in G$  が存在するとき,  $g$  と  $g'$  は共役であるという. 単位元は任意の共役変換で単位元へと写されるため, 単位元と共役な元は単位元のみである.

### 4.3.2 共役類 (conjugacy class)

互いに共役である元からなる集合を指す.

■共役類の元は位数が等しい 群  $G$  の元を  $a, b, g \in G$  とし,  $b = gag^{-1}$  とする.  $a$  の位数を  $n \in \mathbb{N}_+$  とする.

$$b^n = (gag^{-1})^n = ga^n g^{-1} = geg^{-1} = e \quad (6)$$

である

- 共役類の要素数は群の位数の約数である.
- 異なる共役類は共通部分を持たない.

群  $G$  の任意の共役類  $C_1, C_2$  に対して  $C_1 \cap C_2 = \emptyset$ .

- 群は, 共役類の共通部分のない和集合として分解される.

群  $G$  の共役類を  $C_1, \dots, C_n, n \in \mathbb{N}_+$  とすると,  $G = C_1 \cup \dots \cup C_n$ .

- 単位元は自身のみで共役類を構成する.
- 可換群であることと, 全ての元が自身のみで共役類を構成することは, 等価である.

これらより, 位数が素数の群は可換群である. さらに, その群は巡回群である.

## 4.4 群の中心

### 4.4.1 中心 (center)

群  $G$  の中心  $Z(G)$  とは,  $G$  の任意の元と可換である元の集合

$$Z(G) \equiv \{z \in G | zg = gz \text{ for all } g \in G\} \quad (7)$$

である.  $Z(G)$  は群をなす.

### 4.4.2 中心化群 (centralizer)

群  $G$  の部分群を  $S \subset G$  とする.  $G$  における  $S$  の中心化群  $C_G(S)$  とは,  $S$  の全ての元と可換  $G$  の元全体

$$C_G(S) \equiv \{g \in G | gs = sg \text{ for all } s \in S\} \quad (8)$$

である.  $C_G(S)$  は群をなす.

$$C_G(G) = Z(G), \quad C_G(Z_1) = G \quad (9)$$

である.

## 4.5 正規部分群

### 4.5.1 正規部分群 (normal subgroup)

群  $G$  の部分群を  $N \subset G$  とする.  $N$  が  $G$  の正規部分群であるとは, 任意の  $n \in N, g \in G$  に対して

$$gng^{-1} \in N \quad (10)$$

を満たすことをいう.  $N$  が  $G$  の正規部分であることを,  $N \triangleleft G$ , あるいは  $G \triangleright N$  と表す. これは,  $N$  が  $G$  において集合として共役不変であることを表す. つまり,

$$gN = Ng \text{ for all } g \in G \quad (11)$$

である.

群  $G$  に対して,  $G$  自身, および自明群は  $G$  の正規部分群である. これら, 群自身と自明群を自明な正規部分群と呼ぶ. 正規部分群として自明な正規部分群のみを持つ群を単純群 (simple group) と呼ぶ.

$G$  の正規部分群は  $G$  の共役類の和集合である.

### 4.5.2 正規化群 (normalizer)

群  $G$  の部分群を  $S \subset G$  とする.  $G$  における  $S$  の正規化群  $N_G(S)$  とは,  $S$  を共役変換で不変に保つ  $G$  の元の集合

$$N_G(S) \equiv \{g \in G | gS = Sg\} \quad (12)$$

である.  $N_G(S)$  は群をなす.  $S \triangleleft N_G(S)$  である.

$$N_G(G) = N_G(Z_1) = G \quad (13)$$

である.

## 4.6 導来部分群

### 4.6.1 交換子 (commutator)

群  $G$  の元  $g, h \in G$  の交換子  $[g, h]$  とは

$$[g, h] \equiv ghg^{-1}h^{-1} \quad (14)$$

である.  $g, h$  が可換であるとき,

$$[g, h] = e \quad (15)$$

となる.



#### 4.6.2 導来部分群 (derived subgroup)

交換子部分群 (commutator subgroup) と呼ばれる. 群  $G$  の導来部分群  $[G, G]$  とは,  $G$  の交換子全体を生成系として生成される群

$$[G, G] \equiv \langle \{[g, h] | g, h \in G\} \rangle \quad (16)$$

である. 交換子部分群は正規部分群

$$[G, G] \triangleleft G \quad (17)$$

である. 交換子部分群が自明群である群を可換群 (Abelian group), そうでない群を非可換群 (non-Abelian group) と呼ぶ. 交換子部分群が自身と一致する群を完全群 (perfect group) と呼ぶ.

#### 4.6.3 導来列 (derived series)

群  $G$  自身を 0 次導来部分群,  $[G, G]$  を一次導来部分群とし,  $n$  次導来部分群の導来部分群を  $(n+1)$  次導来部分群と呼ぶ.  $G$  の  $n$  次導来部分群を  $G^{(n)}$  として表す.

$$\begin{aligned} G^{(0)} &\equiv G, \\ G^{(1)} &\equiv [G, G] = [G^{(0)}, G^{(0)}], \\ G^{(n)} &\equiv [G^{(n-1)}, G^{(n-1)}]. \end{aligned} \quad (18)$$

有限群では,  $G^{(n)}$  が完全群となるような有限の  $n \in \mathbb{N}$  が存在する.  $n \in \mathbb{N}$  を  $G^{(n)}$  が完全群となる最小の値とする. このとき,

$$G^{(n)} \triangleleft \dots \triangleleft G^{(1)} \triangleleft G^{(0)} \quad (19)$$

を  $G$  の導来列と呼ぶ. 導来列が自明群で終わる群を可解群 (solvable group) と呼ぶ.

## 5 処理の概要

ここでは、プログラム中で行っている処理の概要を示す。

### 5.1 群の生成

ユーザーが行列表示で指定する生成元を  $n$  個とし、生成系を

$$G = \{g_1, g_2, \dots, g_n\} \quad (20)$$

とする。また、生成された元の第 0 世代の集合として、 $H_0 = G$  とする。第 1 世代の集合  $H_1$  を

$$H_1 = \{hg | h \in H_0, g \in G\} - H_0 \quad (21)$$

とする。 $H_1$  の要素数が 0 でないとき、第 2 世代  $H_2$  を

$$H_2 = \{hg | h \in H_1, g \in G\} - (H_0 \cup H_1) \quad (22)$$

とする。この操作を繰り返し、一般に第  $n$  世代の要素数が 0 でないとき、第  $(n+1)$  世代  $H_{n+1}$  を

$$H_{n+1} = \{hg | h \in H_n, g \in G\} - \bigcup_{i=0}^n H_i \quad (23)$$

とする。この操作を、要素数が 0 となる世代が現れるまで続ける。

### 5.2 乗積表の作成

生成された群を  $G$ 、位数を  $n$  とする。 $g_i, g_j \in G$  に対して

$$g_i g_j = g_k \quad (24)$$

となる  $g_k \in G$  を見つける。続いて、元をナンバリングして  $\{0, 1, \dots, n-1\}$  とし、 $i \cdot j = k$  となる  $i, j, k$  に対して  $T_{ij} = k$  となる行列  $T$  を作成する。この、乗積表に対応する整数値行列  $T$  を計算しておくことで、今後の元の演算において複素行列の計算をする必要がなくなる。また、計算効率化のために、 $g_i^{-1} = g_j$  に対して  $I_i = j$  となる逆元対応ベクトル  $I$  や、 $g_j g_i g_j^{-1} = g_k$  に対して  $C_{ij} = k$  となる共役変換行列  $C$  を作成している。

### 5.3 共役類の計算

群  $G$  に対して  $G_0 = G$  とし、 $a \in G_0$  を任意に一つ選び、一つ目の共役類  $C_1$  を

$$C_1 = \{gag^{-1} | g \in G_0\} \quad (25)$$

とする。続いて、集合  $G_1 = G_0 - C_1$  に対して  $a \in G_1$  を任意に一つ選び、二つ目の共役類  $C_2$  を

$$C_2 = \{gag^{-1} | g \in G_1\} \quad (26)$$

とする。この操作を繰り返す。