Getting into the Galois Group Program

by *Mathematica 13.3* 2025 年2月 by mixedmoss

GaloisGroupProgram.nb では　かなり基本的なことから解説をしましたが，ここでは数学的な話は余りしないで　「Mathematicaのプログラム」としての側面に焦点を当ててみたいと思います．従って 理解するには ある程度のMathematica の知識が必要です．まずは「§0」で基本的な説明の復習をしていますが「Galois-GroupProgram.nb§5」と全く同じです．不要の方は「§1」まで飛んでください．

§0. StepByStep Program
(重解&Groebner基底のチェックが必要)

残念ながら私の知る限り Mathematica ではMagmaやSageMathのようなコマンドはありません . しかし5次ぐらいまでなら，今までと同様にして求めることができます．
以下は4次方程式に対するプログラムです .「GaloisGroupProgram.nb§1~§4」の内容を完結にまとめたものです．ここでR (x) が重解を持つときはvsのリストの変更が必要です . そのためにR (x) を出力させてます . (しかしそういうケースは非常に稀です .)

Step1 . 分解方程式

出力は 24次の方程式R (x), そして分解方程式V(x)です. 例として (x^4 - 2) のGalois群を求めます.

In[983]:=
```
ClearAll["`*"]
```

In[984]:=
```
f[x_] = x^4 -2;
vars={α,β,γ,δ};
vs=Permutations[vars].{1,3,-1,0};(*R(x)に重解があるときはここを変える*)
coef=CoefficientList[f[x],x]//Reverse//Drop[#,1]&;
{s1,s2,s3,s4}=Table[SymmetricPolynomial[i,vars],{i,1,4}];
contents={s1+coef[[1]],s2-coef[[2]],s3+coef[[3]],s4-coef[[4]],v-vs[[1]]};
rx=GroebnerBasis[contents,{δ,γ,β,α,v}][[1]]//Factor(*重解を持たないことの確認はここで*)
V[v_]=If[IrreduciblePolynomialQ[rx],rx,Last@(List@@Factor[rx])]
```

Out[990]=
$$\left(334\,084 - 644\,v^4 + v^8\right)\left(2500 + 28\,v^4 + v^8\right)\left(114\,244 + 476\,v^4 + v^8\right)$$

Out[991]=
$$114\,244 + 476\,v^4 + v^8$$

Step2 . f (x)の解を原始元vで表す

In[992]:=

```
GroebnerBasis[contents,{α,v},{δ,γ,β}]//PolynomialMod[Last[#],V[v]]&;
α'=α/.Solve[%==0,α][[1]]//Collect[#,v]&;
GroebnerBasis[contents,{β,v},{δ,γ,α}]//PolynomialMod[Last[#],V[v]]&;
β'=β/.Solve[%==0,β][[1]]//Collect[#,v]&;
GroebnerBasis[contents,{γ,v},{δ,β,α}]//PolynomialMod[Last[#],V[v]]&;
γ'=γ/.Solve[%==0,γ][[1]]//Collect[#,v]&;
GroebnerBasis[contents,{δ,v},{γ,β,α}]//PolynomialMod[Last[#],V[v]]&;
δ'=δ/.Solve[%==0,δ][[1]]//Collect[#,v]&;
sols={α',β',γ',δ'}
```

Out[1000]=

$$\left\{\frac{199\,v}{520}+\frac{v^5}{1040},\ \frac{61\,v}{780}-\frac{v^5}{1560},\ -\frac{199\,v}{520}-\frac{v^5}{1040},\ -\frac{61\,v}{780}+\frac{v^5}{1560}\right\}$$

Step 3 . V(x)の解を原始元vで表す

In[1001]:=

```
vs/.{α→α',β→β',γ→γ',δ→δ'}//Simplify;
PolynomialMod[V[#],V[v]]&/@%;
Position[%,0]//Flatten;
vset=vs[[%]]/.{α→α',β→β',γ→γ',δ→δ'}//Collect[#,v]&
```

Out[1004]=

$$\left\{v,\ \frac{69\,v}{130}+\frac{v^5}{260},\ \frac{407\,v}{312}+\frac{v^5}{624},\ -\frac{119\,v}{120}-\frac{v^5}{240},\ -\frac{69\,v}{130}-\frac{v^5}{260},\ -v,\ \frac{119\,v}{120}+\frac{v^5}{240},\ -\frac{407\,v}{312}-\frac{v^5}{624}\right\}$$

Step4 . Galois群を求める

In[1005]:=

```
Table[
perm=sols/.{vset[[1]]→vset[[i]]}//PolynomialMod[#,V[v]]&;
Table[FirstPosition[Simplify[perm-sols[[k]]{1,1,1,1}],0][[1]],{k,1,4}]//Flatten,{i,1,Length[vset
MatrixForm[Sort[%]]
```

Out[1006]//MatrixForm=

$$\begin{pmatrix}1&2&3&4\\1&4&3&2\\2&1&4&3\\2&3&4&1\\3&2&1&4\\3&4&1&2\\4&1&2&3\\4&3&2&1\end{pmatrix}$$

今度は Galois群の位数は 8 となりました．V(x)も 8 次です．

§1. 分解方程式＆vによるf(x)の解の表現を求める関数　galoisBase

[§0 Step1,Step2]をTableでまとめてModule化しただけです. ただし，vの係数:selected によりR(x)=0が重解を持ったり，Groebner基底が整式にならないことがあるので，While 文を使って，その様な事が起こらない様にしています． ちなみにUntil文は ver13.1 の導入（ちょっと驚きですね?!) なので 避けました.

While [*test*, *body*]

test がTrue を与えなくなるまで，*test* と *body* を繰り返し評価する.

これをこの様に使っています.

```
While[
 ! AllTrue[Append[sols, rx], PolynomialQ[#, v] &] || Discriminant[rx, v] == 0, Body...]
```

testは「Groebner基底の何れかが整式でない」v「R(x)が重解を持つ」です． Bodyの中には「Groebner基底を見つける式」並びに「vの係数を更新する式」が入っています．よってWhileを抜けた時は「Groebner基底が全て整式」∧「R(x)が重解を持たない」を満たす Groebner基底 が保存されているはずです.

In[1007]:=
```
ClearAll["`*"]
```

In[1008]:=
```
galoisBase[f_,selected0_:0]:=Module[{n,rx,allsets,coef,symm,base,contents,oldselected,x1,x2,x
n=Exponent[f,x];
allsets=Subsets[Range[-5,5],{n}];
vars=Take[{x1,x2,x3,x4,x5,x6,x7,x8,x9,x10},n];
coef=CoefficientList[f,x]//Reverse//Drop[#,1]&;
symm=Table[SymmetricPolynomial[i,vars],{i,1,n}];
(*[Step1&2]while文の実行*)
sols=Table[1/v,{i,1,n}];rx=V=v;(*whileの初期値*)
selected=If[SameQ[selected0,0],RandomChoice[allsets],selected0];(*whileの初期値*)
While[
!AllTrue[Append[sols,rx],PolynomialQ[#,v]&]||Discriminant[rx,v]==0,(*「R(x)が重解を持つ」v「V,sols
oldselected=selected;
vs=Permutations[vars].selected;
contents=Append[symm+coef*Table[(-1)^(k-1),{k,1,n}],v-vs[[1]]];
rx=GroebnerBasis[contents,Append[vars,v]][[1]];(*原始元vを解に持つ因数分解前の式*)
V=If[IrreduciblePolynomialQ[rx],rx,Last@(List@@Factor[rx])];(*VのGroebner基底*)
sols=Table[base=GroebnerBasis[contents,{vars[[i]],v},Drop[vars,{i}]]//PolynomialMod[Last[#],V]
          vars[[i]]/.Solve[base==0,vars[[i]]][[1]]//Collect[#,v]&,{i,1,n}];(*解のGroebner基底*)
selected=RandomChoice[allsets]
];
selected=oldselected;
{V,sols}]
```

使い方は，次のように関数fを入力するだけです．出力は 分解方程式と f(x)の解の原始元による表現です． fは既約でなくとも大丈夫ですが，$\left(x^4+2x^2+1\right)$など，重解を持つときは駄目です．また「vars,vs,selected,sols」は Globalです.

In[1037]:=
```
galoisBase[x^4 - 4 x^2 + 1]
```

Out[1037]=
$$\left\{1 - 52\, v^2 + v^4,\ \left\{\frac{209\, v}{15} - \frac{4\, v^3}{15},\ -\frac{209\, v}{15} + \frac{4\, v^3}{15},\ -\frac{56\, v}{15} + \frac{v^3}{15},\ \frac{56\, v}{15} - \frac{v^3}{15}\right\}\right\}$$

Randomで 係数:selected を選んでいるので，評価すると,ほぼ毎回分解方程式や fの解の表現が変わります．また第 2 引数でselectedの初期値(絶対値は5以下がお勧め）を与えることも出来ます．係数が不適切な場合は自動的に変更されます．

In[1038]:=
```
galoisBase[x^4 - 4 x^2 + 1, {1, -1, 2, 3}](*v=x1-x2+2x3+3x4としてTry⇒成功*)
selected
```

Out[1038]=
$$\left\{9 - 12\, v^2 + v^4,\ \left\{-\frac{2\, v}{3} + \frac{v^3}{9},\ \frac{2\, v}{3} - \frac{v^3}{9},\ -\frac{7\, v}{3} + \frac{2\, v^3}{9},\ \frac{7\, v}{3} - \frac{2\, v^3}{9}\right\}\right\}$$

Out[1039]=
```
{1, -1, 2, 3}
```

In[1040]:=
```
galoisBase[x^4 - 4 x^2 + 1, {1, 2, 3, 4}](*v=x1+2x2+3x3+4x4としてTry⇒重解が有り失敗*)
selected
```

Out[1040]=
$$\left\{4 - 28\, v^2 + v^4,\ \left\{-\frac{41\, v}{8} + \frac{3\, v^3}{16},\ \frac{41\, v}{8} - \frac{3\, v^3}{16},\ \frac{11\, v}{8} - \frac{v^3}{16},\ -\frac{11\, v}{8} + \frac{v^3}{16}\right\}\right\}$$

Out[1041]=
```
{-5, -4, 2, 5}
```

§2. galoisGroup[ ] のパーツ作成

2-1. V(x)の解の発見と表示のパーツ

§0の[Step3]に当たります．方程式は 先ほど評価した $f(x) = x^4 - 4x^2 + 1$ を使います．まず R(x)の解集合 vsは，もともと f=0 の解x_iで表されていましたが，x_i を原始元vで表した式を代入して vs_0とおきます．またVや解の原始元 vによる表現も直前のものを使います．実装には AssociationThreadと replaceAll(/.)を使います．

```
AssociationThread[{key1, key2, ...} → {val1, val2, ...}]
連想 <|key1 → val1, key2 → val2, ...|> を与える.
```

In[1014]:=

```
vs0 = vs /. AssociationThread[vars → sols] // Collect[#, v] &
V
```

Out[1014]=

$$\left\{v, -\frac{1931\,v}{693}+\frac{16\,v^3}{2079}, \frac{34\,v}{99}+\frac{v^3}{297}, -\frac{6026\,v}{693}+\frac{79\,v^3}{2079}, -\frac{2714\,v}{693}+\frac{25\,v^3}{2079}, -\frac{706\,v}{77}+\frac{3\,v^3}{77},\right.$$
$$\frac{1931\,v}{693}-\frac{16\,v^3}{2079}, -v, \frac{2714\,v}{693}-\frac{25\,v^3}{2079}, \frac{706\,v}{77}-\frac{3\,v^3}{77}, -\frac{34\,v}{99}-\frac{v^3}{297}, \frac{6026\,v}{693}-\frac{79\,v^3}{2079},$$
$$\frac{1021\,v}{693}-\frac{2\,v^3}{2079}, -\frac{749\,v}{99}+\frac{10\,v^3}{297}, \frac{251\,v}{77}-\frac{2\,v^3}{231}, \frac{5899\,v}{693}-\frac{74\,v^3}{2079}, -\frac{76\,v}{11}+\frac{v^3}{33}, \frac{5116\,v}{693}-\frac{65\,v^3}{2079},$$
$$\left.-\frac{251\,v}{77}+\frac{2\,v^3}{231}, -\frac{5899\,v}{693}+\frac{74\,v^3}{2079}, -\frac{1021\,v}{693}+\frac{2\,v^3}{2079}, \frac{749\,v}{99}-\frac{10\,v^3}{297}, -\frac{5116\,v}{693}+\frac{65\,v^3}{2079}, \frac{76\,v}{11}-\frac{v^3}{33}\right\}$$

Out[1015]=

$$1089 - 228\,v^2 + v^4$$

上のvsの24個の要素の内，V[x] = 0 の解となるものを選ばないといけない．§0では代入して V[x]=0(vの恒等式として) となるものを探したが，これは計算量がかなり多い． そこですこしずるいが V=0 の数値解 vsols を利用する.

In[1016]:=

```
vsols = v /. NSolve[V == 0, v]
```

Out[1016]=

```
{-14.9372, -2.20925, 2.20925, 14.9372}
```

ここでvs0の vに vsols[[1]] を代入してみる

In[1017]:=

```
vs1 = vs0 /. {v → vsols[[1]]}
```

Out[1017]=

```
{-14.9372, 15.9725, -16.3514, 3.24453, 18.4219, 7.10823, -15.9725, 14.9372,
 -18.4219, -7.10823, 16.3514, -3.24453, -18.8009, 0.79504, -19.8362, -8.52245,
 2.20925, -6.07296, 19.8362, 8.52245, 18.8009, -0.79504, 6.07296, -2.20925}
```

このうち，vsols[[2]]と 一致するvsの要素を探す．有効数字は，目では等しくとも異なることが普通にある．実際，差をとっても0にはなっていない(下の式).

In[1018]:=
```
Abs[vs1 - vsols[[2]]]
```
Out[1018]=
$$\{12.7279, 18.1817, 14.1421, 5.45378, 20.6312, 9.31749, 13.7632, 17.1464, 16.2127, 4.89898, 18.5606, 1.03528, 16.5916, 3.00429, 17.6269, 6.31319, 4.41851, 3.8637, 22.0454, 10.7317, 21.0101, 1.41421, 8.28221, 1.02141 \times 10^{-14}\}$$

故に，差の絶対値が最小となっている要素のPositionを求める.

In[1019]:=
```
PositionSmallest[Abs[vs1 - vsols[[2]]]]
```
Out[1019]=
$$\{24\}$$

vsols[[3]], vsols[[4]] についても同様に行う.

In[1020]:=
```
Table[PositionSmallest[Abs[vs1 - vsols[[i]]]], {i, 1, Length[vsols]}] // Flatten
vset = vs0[[%]]
```
Out[1020]=
$$\{1, 24, 17, 8\}$$

Out[1021]=
$$\left\{v, \frac{76\,v}{11} - \frac{v^3}{33}, -\frac{76\,v}{11} + \frac{v^3}{33}, -v\right\}$$

上のカッコ内が V=0の解となる. R (x) の中に重解はないのでこの中に間違った要素が入り込んでいることはない. ずるいやり方だが，実行速度は全く違う.

2-2. Galois群を求めるパーツ

「§0 Part4」に当たる. f = 0 の解の原始元vによる表現は sols に入っている. 一方，vset には V=0の解のvによる表現が入っている.

In[1022]:=
```
sols
```
Out[1022]=
$$\left\{\frac{619\,v}{693} - \frac{8\,v^3}{2079}, -\frac{619\,v}{693} + \frac{8\,v^3}{2079}, -\frac{164\,v}{693} + \frac{v^3}{2079}, \frac{164\,v}{693} - \frac{v^3}{2079}\right\}$$

In[1023]:=
```
vset
```
Out[1023]=
$$\left\{v, \frac{76\,v}{11} - \frac{v^3}{33}, -\frac{76\,v}{11} + \frac{v^3}{33}, -v\right\}$$

例えば，vset[[1]] ⇒ vset[[4]] に変えると，

$$\texttt{sols} = \{\texttt{x1}, \texttt{x2}, \texttt{x3}, \texttt{x4}\} \Rightarrow \left\{-\frac{2\,v}{3} + \frac{v^3}{9}, \frac{2\,v}{3} - \frac{v^3}{9}, \frac{7\,v}{3} - \frac{2\,v^3}{9}, -\frac{7\,v}{3} + \frac{2\,v^3}{9}\right\} =$$

$\{x2, x1, x4, x3\}$ と変わる.

同様に vset[[1]] を vset[[2]], vset[[3]] へ代入し，(Mod V) で簡略化すれば良いが，この計算も数が多いと大変になる.「毒を食らわば皿まで」でこれも数値計算でやって見る. .

In[1024]:=
```
perm1 = sols /. {v → vsols[[1]]}
```
Out[1024]=
```
{-0.517638, 0.517638, 1.93185, -1.93185}
```

vsolsはvsetを数値化した集合だから上は (x1, x2, x3, x4)の数値化に当たる．v ⇒ vset[[4]]=vsols[[4]] とすると

In[1025]:=
```
perm4 = sols /. {v → vsols[[4]]}
```
Out[1025]=
```
{0.517638, -0.517638, -1.93185, 1.93185}
```

この様に (x1, x2, x3, x4) ⇒ (x2, x1, x4, x3) となる．この方が計算速度はずーと速いだろう．ただし，有効数字が完全に一致する事は余りないので，先ほどと同じようにプログラムする．

In[1026]:=
```
Table[PositionSmallest[Abs[perm4 - perm1[[k]]], {k, 1, Length[vset]}] // Flatten
```
Out[1026]=
```
{2, 1, 4, 3}
```

これを vsols の全ての要素についてやればよいので次のようにプログラムすればよい．

In[1027]:=
```
perm1 = sols /. {v → vsols[[1]]};
Table[
  permi = sols /. {v → vsols[[i]]};
  Table[PositionSmallest[Abs[permi - perm1[[k]]]][[1]], {k, 1, Length[vset]}] // Flatten,
  {i, 1, Length[vset]}];
MatrixForm[Sort[%]]
```
Out[1029]//MatrixForm=
$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

§3.ガロア群を求める関数 galoisGroup

§1で作ったgaloisBase と§2のプログラムを合わせて，Galois群を求めるプログラムを作りました．このプログラムの global変数は，galoisBase のGlobal変数も含むので沢山あります．特に VとsolsはGlobalです．従ってこのプログラムを実行すれば Vと「fの解のvによる表現=sols」にアクセスできるので，改めて galoisBaseを実行する必要はありません．また galoisgroupでは，Galois群をリストで得ることができます．さらに galoisBaseと同様に 係数:selected の初期値を設定することもできます．

In[1030]:=

```
galoisGroup[f_,selected0_:0]:=Module[
{n,m,vsols,vs0,vs1,pos,vset,perm1,permi,irQ},
{V,sols}=galoisBase[f,selected0];
n=Exponent[f,x];
m=Exponent[V,v];
irQ=SameQ[m,Factorial[n]];(*=irreducibleQ*)
(*[Step3]V(x)の解の集合vsを,その第１要素v(=v1)の式で表す．Galois群がSnの時はバイパスしています*)
If[!irQ,vsols=v/.NSolve[V==0,v];
vs0=vs/.AssociationThread[vars→sols]//Collect[#,v]&;
vs1=vs0/.{v→vsols[[1]]};
pos=Table[PositionSmallest[Abs[vs1-vsols[[i]]],{i,1,m}]//Flatten;
vset=vs0[[pos]];
(*[Step4]Galois群を求める．Galois群がSnの時はバイパスしています*)
galoisgroup=If[irQ,Permutations[Range[n]],
   perm1=sols/.{v→vsols[[1]]};
   Table[
    permi=sols/.{v→vsols[[i]]};
    Table[PositionSmallest[Abs[permi-perm1[[k]]]][[1]],{k,1,n}]//Flatten,{i,1,m}]];
MatrixForm[Sort[galoisgroup]]]
```

5 次までは大丈夫ですが，6 次以上はうまく行きませんでした．

Galois群が F_{20} の5次方程式で試してみましょう．（最大30 秒かかります）

In[1036]:=
```
Timing[galoisGroup[x^5 + 15 x + 12]]
```
Out[1036]=

$$\left\{21.6406,\ \begin{pmatrix} 1&2&3&4&5\\ 1&3&5&2&4\\ 1&4&2&5&3\\ 1&5&4&3&2\\ 2&1&5&4&3\\ 2&3&4&5&1\\ 2&4&1&3&5\\ 2&5&3&1&4\\ 3&1&4&2&5\\ 3&2&1&5&4\\ 3&4&5&1&2\\ 3&5&2&4&1\\ 4&1&3&5&2\\ 4&2&5&3&1\\ 4&3&2&1&5\\ 4&5&1&2&3\\ 5&1&2&3&4\\ 5&2&4&1&3\\ 5&3&1&4&2\\ 5&4&3&2&1 \end{pmatrix}\right\}$$

Part3とPart4を有効数字を使わずに厳密にやるプログラムでは2分ぐらい掛かっていたので，だいぶん短くなりました．しかし，Magmaでは 0.11秒でした．200倍も速いです．(私のPCはもうじき12才になるのでそれも関係しているのは間違いありません．しかし Magmaの サーバーが 200倍も速いとは考えられません．）Magmaの凄さはスピードだけでは有りません．本当の凄さは「方程式の次数」です．10次方程式は 0.12秒，20次方程式は 0.23秒, 50次方程式が2秒，100次方程式でさえ27秒，さすがに 200次方程式は 120秒の制限時間内に終わりませんでしたが，100次方程式のGalois群まで求まってしまいます．その最大位数は order(S_{100})=100! ≒ 10^{158} になるのですが...

話を元に戻しましょう.Galois群の位数と分解方程式Vです．

In[1032]:=
```
galoisgroup // Length
```
Out[1032]=
```
20
```

In[1033]:=
```
V
```
Out[1033]=

$$1\,713\,288\,397\,225\,285\,702\,656 - 364\,116\,798\,176\,818\,821\,120\,v + 177\,521\,224\,349\,906\,064\,000\,v^2 + 295\,241\,043\,619\,929\,600\,v^3 + 1\,167\,623\,177\,045\,677\,200\,v^4 + 249\,623\,338\,181\,554\,944\,v^5 + 3\,072\,143\,039\,156\,640\,v^6 + 1\,824\,264\,155\,376\,000\,v^7 + 181\,863\,110\,698\,200\,v^8 - 4\,190\,877\,921\,600\,v^9 + 889\,252\,399\,296\,v^{10} - 59\,694\,183\,360\,v^{11} - 1\,371\,387\,375\,v^{12} - 584\,366\,400\,v^{13} + 3\,661\,200\,v^{14} - 503\,616\,v^{15} + 42\,630\,v^{16} + v^{20}$$

galoisgroupは数字のリストなので, 位数は Length[galoisgroup]で測れます．さて，このプログラムは非常に遅いので，正直，私以外の方の実用には余りお勧めできませんが「過程を楽しむ」ことが共有できていたら幸いです．なお Mathematicaを使って 5次方程式（将来的にはもっと高次）も解いていますので，良かったらそちらもお読みください．(https://mixedmoss/com/Mathematica/Galois)

In[1034]:=