分解方程式，原始元，ガロア群 の基本
with *Mathematica* 13.3

2025 年2月　by　mixedmoss

§1.　分解方程式の作成

Galois 分解方程式を求めるには　色々方法があると思いますが，ここでは「数学の教科書に載っている方法」と「Grobner基底による方法」の2つを Mathematica でプログラムしたのでご紹介します．例として，４次関数 $f(x) = x^4 - 10x^2 + 1$ を取り上げます．

---

§1 -A. 解と係数の関係＆対称式の利用(参考文献[7][8])

In[•]:=
```
ClearAll["`*"];

f[x_] = x^4 - 10 x^2 + 1;
x /. Solve[f[x] == 0, x] (*f(x)=0の解*)
```

Out[•]=

$$\left\{-\sqrt{5-2\sqrt{6}},\ \sqrt{5-2\sqrt{6}},\ -\sqrt{5+2\sqrt{6}},\ \sqrt{5+2\sqrt{6}}\right\}$$

f[x] = 0 の解は Mathematicaでは

$$\left\{\left\{x\to-\sqrt{5-2\sqrt{6}}\right\},\left\{x\to\sqrt{5-2\sqrt{6}}\right\},\left\{x\to-\sqrt{5+2\sqrt{6}}\right\},\left\{x\to\sqrt{5+2\sqrt{6}}\right\}\right\}$$

のようにrulesで出力されるので 下のコマンドを使って リストに直しています．

> *expr* /. *rules* または ReplaceAll[*expr*, *rules*]
> 式 *expr* の下位区分のそれぞれを変換しようとするとき，規則または規則のリストを適用する．

f[x] = 0 の解を $\{\alpha, \beta, \gamma, \delta\}$ とする．同じ値がないように vs を取る．（下の係数は{1,3,-1,0}ですが，vsの中に同じ「値」がなければ何でも大丈夫です．もし同じ「値」があれば，次のR(x)が重解を持つので分かります．）

In[•]:=
```
vs = Permutations[{α, β, γ, δ}].{1, 3, -1, 0}
```

Out[•]=

$$\begin{aligned}&\{\alpha+3\beta-\gamma,\ \alpha+3\beta-\delta,\ \alpha-\beta+3\gamma,\ \alpha+3\gamma-\delta,\ \alpha-\beta+3\delta,\ \alpha-\gamma+3\delta,\\ &\ 3\alpha+\beta-\gamma,\ 3\alpha+\beta-\delta,\ -\alpha+\beta+3\gamma,\ \beta+3\gamma-\delta,\ -\alpha+\beta+3\delta,\ \beta-\gamma+3\delta,\\ &\ 3\alpha-\beta+\gamma,\ 3\alpha+\gamma-\delta,\ -\alpha+3\beta+\gamma,\ 3\beta+\gamma-\delta,\ -\alpha+\gamma+3\delta,\ -\beta+\gamma+3\delta,\\ &\ 3\alpha-\beta+\delta,\ 3\alpha-\gamma+\delta,\ -\alpha+3\beta+\delta,\ 3\beta-\gamma+\delta,\ -\alpha+3\gamma+\delta,\ -\beta+3\gamma+\delta\}\end{aligned}$$

List a の第*i* 成分は a[[*i*]] を使う．R(x)=(x-vs[[1]])(x-vs[[2]])⋯(x-vs[[24]])を作ると,R(x)は$\alpha,\beta,\gamma,\delta$の対称式．解と係数の関係により $\{\alpha, \beta, \gamma, \delta\}$ の 1~4次対称式の値は{0,-10,-1,0}なので，ProductとSymmetricReductionを使うと「10秒程度」で結果が出る．

Product[f, {i, i_{max}}]
乗積$\prod_{i=1}^{i_{max}} f$ を評価する.

SymmetricReduction [f, {x_1, …, x_n}, {s_1, …, s_n}]
p における初等対称式を s_1, …, s_n で置き換えたペア {p, q} を与える.

```
In[•]:= Product[(x - vs[[k]]), {k, 1, Length[vs]}] //
         SymmetricReduction[#, {α, β, γ, δ}, {0, -10, 0, 1}] &
Out[•]=
{2 845 177 430 298 890 625 - 2 005 037 381 967 738 300 x^2 +
  540 254 637 222 727 266 x^4 - 74 964 181 748 810 700 x^6 + 6 115 603 032 316 015 x^8 -
  314 736 576 091 000 x^10 + 10 611 439 620 700 x^12 - 238 205 543 800 x^14 +
  3 554 349 295 x^16 - 34 525 900 x^18 + 207 970 x^20 - 700 x^22 + x^24, 0}
```

第2成分は対称式に直しきれなかった余りとなる. ここでは当然0なので, 第1成分をR(x)と定めて, 因数分解する.「%」は直前の出力を, Factor は因数分解を表す.

```
In[•]:= Factor[%[[1]]]
Out[•]=
(3249 - 186 x^2 + x^4) (529 - 154 x^2 + x^4) (5329 - 154 x^2 + x^4)
 (2209 - 106 x^2 + x^4) (625 - 58 x^2 + x^4) (225 - 42 x^2 + x^4)
```

vsの第 1 項をvとおく. 即ちv=α + 3 β - γ. これが原始元となる. また α, β, γ, δの取り方の順序は任意なので, vの満たす方程式V[x] (Galois 分解方程式) をR(x)の 6 番目のカッコ内の式に決めて良い. (6 番目のカッコの中の式が一番簡単そうなので. なお @@の解説は長くなるので省略する.)

```
In[•]:= V[x_] = (List @@ %)[[6]]
Out[•]=
225 - 42 x^2 + x^4
```

以上が解と係数の関係を利用した原始元と分解方程式の求め方となります .理論的には分かりやすいですが,次数が増えると計算時間が指数関数的に伸びていきます.

§1 - B. Groebner基底の利用

これは試行錯誤で見つけました． fの基本対称式: s1,s2,s3,s4の値と v=α + 3 β - γを素にしてGroebner基底を求めます．その第1項が R(v)となります． なおMathematicaでは Groebner基底の使い方は次のようになります．

GroebnerBasis [{$poly_1$, $poly_2$, ...}, {x_1, x_2, ...}]

多項式の集合 $poly_i$についてグレブナー(Gröbner)基底を形成する多項式をリスト形式で返す．

GroebnerBasis [{$poly_1$, $poly_2$, ...}, {x_1, x_2, ...}, {y_1, y_2, ...}]

変数 y_iを消去したグレブナー基底を求める．

In[ ]:=
```
s1 = SymmetricPolynomial[1, {α, β, γ, δ}];(*α+β+γ+δ*)
s2 = SymmetricPolynomial[2, {α, β, γ, δ}];(*α β+α γ+β γ+α δ+β δ+γ δ*)
s3 = SymmetricPolynomial[3, {α, β, γ, δ}];(*α β γ+α β δ+α γ δ+β γ δ*)
s4 = SymmetricPolynomial[4, {α, β, γ, δ}];(*α β γ δ*)

GroebnerBasis[{s1, s2 + 10, s3, s4 - 1, v - (α + 3 β - γ)}, {δ, γ, β, α, v}][[1]]
Factor[%]
V[x_] = (List @@ %)[[6]] /. {v → x}
```

Out[ ]=
$$2\,845\,177\,430\,298\,890\,625 - 2\,005\,037\,381\,967\,738\,300\,v^2 + 540\,254\,637\,222\,727\,266\,v^4 - 74\,964\,181\,748\,810\,700\,v^6 + 6\,115\,603\,032\,316\,015\,v^8 - 314\,736\,576\,091\,000\,v^{10} + 10\,611\,439\,620\,700\,v^{12} - 238\,205\,543\,800\,v^{14} + 3\,554\,349\,295\,v^{16} - 34\,525\,900\,v^{18} + 207\,970\,v^{20} - 700\,v^{22} + v^{24}$$

Out[ ]=
$$\left(3249 - 186\,v^2 + v^4\right)\left(529 - 154\,v^2 + v^4\right)\left(5329 - 154\,v^2 + v^4\right)\left(2209 - 106\,v^2 + v^4\right)\left(625 - 58\,v^2 + v^4\right)\left(225 - 42\,v^2 + v^4\right)$$

Out[ ]=
$$225 - 42\,x^2 + x^4$$

先ほど求めた R(x)の式と同じ式が　遥かに短時間で求められます．Groebner基底は，以前「幾何の定理の自動証明」のときに Risa/Asirで触っていました．使うことは出来ますが，詳しい理論は知りません．なお，MathematicaのGroebner基底は非常に使いやすいです．

§2. α,β,γ,δを, 原始元 v=α + 3 β - γ で表す

v = α + 3 β - γを原始元としてf (x) の解を全て v の多項式で表します .

2 - A. 解と係数の関係とユークリッドの互除法を使う(参考文献[7][8])

2- A-1. β,γ,δの対称式をαの式で表す.(f[x]を(x-α)で割った商は (x-β)(x-γ)(x-δ)であることを使う)

以下のMathematicaのコマンドで整式の割り算とModを考える.

PolynomialQuotient[*p*, *q*, *x*]
x の多項式として *p* を *q* で割った商を, 剰余は除去して与える.
PolynomialRemainder[*p*, *q*, *x*]
x の多項式として *p* を *q* で割った余りを与える.
PolynomialQuotientRemainder[*p*, *q*, *x*]
x の多項式として扱われる *p* を *q* で割った商と剰余のリストを与える.

PolynomialMod[*poly*, *m*]
m を法として多項式 *poly* を与える.
PolynomialMod[*poly*, {m_1, m_2, ...}]
すべての m_iを法として簡約する.

```
In[ ]:= PolynomialQuotient[f[x], x - α, x]
        CoefficientList[%, x] // Reverse
```

Out[ ]=

$$x^3 - 10\,\alpha + x^2\,\alpha + \alpha^3 + x\left(-10 + \alpha^2\right)$$

Out[ ]=

$$\left\{1,\ \alpha,\ -10 + \alpha^2,\ -10\,\alpha + \alpha^3\right\}$$

f(x) = (x − α)(x − β)(x − γ)(x − δ) だから
$\beta + \gamma + \delta = -\alpha,\ \beta\gamma + \beta\delta + \gamma\delta = -10 + \alpha^2,\ \beta\,\gamma\,\delta = -\left(-10\,\alpha + \alpha^3\right)$ -- (#1)
(解と係数の関係からも求まる)

2- A-2. vを原始元としてαをvで表す

In[•]:=
```
vs
```
Out[•]=
$$\{\alpha+3\beta-\gamma,\ \alpha+3\beta-\delta,\ \alpha-\beta+3\gamma,\ \alpha+3\gamma-\delta,\ \alpha-\beta+3\delta,\ \alpha-\gamma+3\delta,$$
$$3\alpha+\beta-\gamma,\ 3\alpha+\beta-\delta,\ -\alpha+\beta+3\gamma,\ \beta+3\gamma-\delta,\ -\alpha+\beta+3\delta,\ \beta-\gamma+3\delta,$$
$$3\alpha-\beta+\gamma,\ 3\alpha+\gamma-\delta,\ -\alpha+3\beta+\gamma,\ 3\beta+\gamma-\delta,\ -\alpha+\gamma+3\delta,\ -\beta+\gamma+3\delta,$$
$$3\alpha-\beta+\delta,\ 3\alpha-\gamma+\delta,\ -\alpha+3\beta+\delta,\ 3\beta-\gamma+\delta,\ -\alpha+3\gamma+\delta,\ -\beta+3\gamma+\delta\}$$

vsのうちαの係数がvと同じとなるものを見つけ(この場合は最初の6個)，その全ての積を取る．するとβ, γ, δの対称式ができる．これを(#1)を使って変形する．

In[•]:=
```
g1[x_] = Product[(x - vs[[i]]), {i, 1, 6}]
g1[x_] = SymmetricReduction[%, {β, γ, δ}, {-α, -10 + α^2, -(-10 α + α^3)}][[1]] //
  Collect[#, {x, α}] &
```
Out[•]=
$$(x-\alpha+\beta-3\gamma)(x-\alpha-3\beta+\gamma)(x-\alpha+\beta-3\delta)(x-\alpha+\gamma-3\delta)(x-\alpha-3\beta+\delta)(x-\alpha-3\gamma+\delta)$$

Out[•]=
$$-144\,000+x^6-2x^5\alpha+108\,100\alpha^2-20\,260\alpha^4+1105\alpha^6+x^4(-260+19\alpha^2)+$$
$$x^3(-120\alpha+28\alpha^3)+x^2(16\,900-1960\alpha^2+35\alpha^4)+x(49\,400\alpha-9400\alpha^3+414\alpha^5)$$

「v=vs[[1]]=$\alpha+3\beta-\gamma$」なのでg1[v] = 0 が成り立つ．　g1[x]のαをyに変えg1[x,y]を作り，さらにxをvと名前を変えg1v[y]を作る．このときvの式を V(x)で割った余りを考え，簡単にしておく．

In[•]:=
```
g1[x_, y_] = g1[x] /. {α → y}
g1v[y_] = g1[v, y] // PolynomialMod[#, V[v]] & // Collect[#, y] &
```
Out[•]=
$$-144\,000+x^6-2x^5y+108\,100y^2-20\,260y^4+1105y^6+x^4(-260+19y^2)+$$
$$x^3(-120y+28y^3)+x^2(16\,900-1960y^2+35y^4)+x(49\,400y-9400y^3+414y^5)$$

Out[•]=
$$-94\,950+7519v^2+(49\,850v-204v^3)y+(103\,825-1162v^2)y^2+$$
$$(-9400v+28v^3)y^3+(-20\,260+35v^2)y^4+414vy^5+1105y^6$$

g1v(α) = g1(v, α) = g1(v) = 0 だから，f(y)とg1v(y)はy = αを共通解に持つ．
さらにg1v(β) ≠ 0, g1v(γ) ≠ 0, g1v(δ) ≠ 0 から　共通解はy＝αのみである．
故に，ユークリッドの互除法を使いf(y)とg1v(y)の最大公約式を求めると
それは(y − α)となる．これでαはvの式で表される．

ユークリッドの互除法を実現するために，まずはg1v[y] をf[y] で割った余りr1[y] を求める．

In[•]:=
```
r1[y_] =
 PolynomialRemainder[g1v[y], f[y], y] // PolynomialMod[#, V[v]] & // Collect[#, y] &
```
Out[•]=
$$-85\,740+7484v^2+(49\,436v-204v^3)y+(10\,620-812v^2)y^2+(-5260v+28v^3)y^3$$

次にf[y]をr1[y] で割った余りr2[y] を求める．

```
In[ ]:= r2[y_] =
         PolynomialRemainder[f[y], r1[y], y] // PolynomialMod[#, V[v]] & // Collect[#, y] &
```

Out[ ]=

$$\frac{32\,227\,200 - 5\,601\,920\,v^2}{3\,679\,200 + 1\,031\,416\,v^2} + \frac{\left(5\,243\,520\,v - 149\,120\,v^3\right) y}{3\,679\,200 + 1\,031\,416\,v^2} + \frac{\left(-7\,833\,600 + 689\,920\,v^2\right) y^2}{3\,679\,200 + 1\,031\,416\,v^2}$$

最後にr1[y] をr2[y] で割った余り r3[y] を求める.

```
In[ ]:= r3[y_] =
         PolynomialRemainder[r1[y], r2[y], y] // PolynomialMod[#, V[v]] & // Collect[#, y] &
```

Out[ ]=

$$\frac{264\,554\,019\,000 - 51\,898\,817\,880\,v^2}{-27\,912\,825 + 5\,604\,522\,v^2} + \frac{\left(113\,338\,683\,000\,v - 5\,541\,649\,560\,v^3\right) y}{-27\,912\,825 + 5\,604\,522\,v^2}$$

これはyの一次式なので これを yについて解くと, αはvの式で表される (これを α0とおく)

```
In[ ]:= α0 = y /. Solve[r3[y] == 0, y][[1]]
```

Out[ ]=

$$\frac{244\,957\,425 - 48\,054\,461\,v^2}{v\left(-104\,943\,225 + 5\,131\,157\,v^2\right)}$$

これがαをvで表した有理式. このままでは以後の計算が大変なので,

V（v）を分母の因子（ここでは2個） で割る事により整式に直し,

改めて$αと名付ける. （DenominatorとNumeratorは, それぞれ分母と分子を取り出すコマンド）

```
In[ ]:= bunbo1 = (List @@ Denominator[α0])[[1]]
        bunbo2 = Plus @@ (List @@ Denominator[α0])[[2]]
```

Out[ ]=

v

Out[ ]=

$$-104\,943\,225 + 5\,131\,157\,v^2$$

```
In[ ]:= (*V(v)をbunbo1/2で割った商がq1/2,余りがr1/2*)
```

```
In[ ]:= {q1, r1} = PolynomialQuotientRemainder[V[v], bunbo1, v];
        {q2, r2} = PolynomialQuotientRemainder[V[v], bunbo2, v];
        $α = (-q1 / r1) * (-q2 / r2) * (Numerator[α0]) // PolynomialMod[#, V[v]] & // Expand
```

Out[ ]=

$$\frac{4\,v}{15} - \frac{v^3}{45}$$

これが α を vの整式で表した式となる. 同様にして, β, γ,

δもvの式で表すことができるが, 次の Groebner基底を使う方が速いのでここでは省略する.

（なお, この例では r2が0次となったのでこれで終了だが,

もしr2が1次の時は, これを繰り返せば整式に直る.）

2 - B. Groebner基底の利用

これも試行錯誤で見つけました．したがって理論的に正しいか否かは不明です．先ずは αをvで表します．V[x] をGroebner基底を使って求めた時と殆ど同じですが，αとvを残すように指定します．99%うまく行きますが，ごくごく稀に vの整式でなく $\sqrt{\text{vの整式}}$ の形になることがあります．原因は不明です．

```
In[•]:= GroebnerBasis[{s1, s2 + 10, s3, s4 - 1, v - (α + 3 β - γ)}, {α, v}, {δ, γ, β}] //
          PolynomialMod[Last[#], V[v]] &;
        α' = α /. Solve[% == 0, α][[1]] // Collect[#, v] &
```

Out[•]=

$$\frac{4\,v}{15} - \frac{v^3}{45}$$

先の結果と一致しました．同様に，β, γ, δをvの式で表すことができます．最後に $\{\alpha,\beta,\gamma,\delta\}$をまとめて sols に入れて，表示します．

```
In[•]:= GroebnerBasis[{s1, s2 + 10, s3, s4 - 1, v - (α + 3 β - γ)}, {β, v}, {δ, γ, α}] //
          PolynomialMod[Last[#], V[v]] &;
        β' = β /. Solve[% == 0, β][[1]] // Collect[#, v] &;
        GroebnerBasis[{s1, s2 + 10, s3, s4 - 1, v - (α + 3 β - γ)}, {γ, v}, {δ, β, α}] //
          PolynomialMod[Last[#], V[v]] &;
        γ' = γ /. Solve[% == 0, γ][[1]] // Collect[#, v] &;
        GroebnerBasis[{s1, s2 + 10, s3, s4 - 1, v - (α + 3 β - γ)}, {δ, v}, {γ, β, α}] //
          PolynomialMod[Last[#], V[v]] &;
        δ' = δ /. Solve[% == 0, δ][[1]] // Collect[#, v] &;
        sols = {α', β', γ', δ'}
```

Out[•]=

$$\left\{\frac{4\,v}{15} - \frac{v^3}{45},\ -\frac{4\,v}{15} + \frac{v^3}{45},\ -\frac{23\,v}{15} + \frac{2\,v^3}{45},\ \frac{23\,v}{15} - \frac{2\,v^3}{45}\right\}$$

§3. vs1~vs24からV[x]の解を見つけ, vで表す

vsの要素は24個あるが, このうちV (x) の解を見つけ vsetにまとめて入れる. やり方は
[1] {α,β, γ, δ}をvで表した式を使い vs の要素を全てvで表す. (vs'に入れる)
[2] これをV[x]に代入して0となる vs の要素を見つける
という力技である.

代入には ReplaceAll(/.) を使う. またリストの中の場所を見つけるには Positionや FirstPosition を使う.

> Position [*expr*, *pattern*]
> 式 *expr* に現れるパターン *pattern* にマッチするオブジェクトの位置のリストを与える.

```
In[•]:= vs' = vs /. {α → α', β → β', γ → γ', δ → δ'} // Simplify;(*vsの要素をvで表す*)
        PolynomialMod[V[#], V[v]] & /@ %;
        pos = Position[%, 0] // Flatten
        vs[[pos]]
        vset = vs'[[pos]] // Collect[#, v] &
```

Out[•]=

$$\{1, 8, 17, 24\}$$

Out[•]=

$$\{\alpha + 3\beta - \gamma,\ 3\alpha + \beta - \delta,\ -\alpha + \gamma + 3\delta,\ -\beta + 3\gamma + \delta\}$$

Out[•]=

$$\left\{v,\ -v,\ \frac{14\,v}{5} - \frac{v^3}{15},\ -\frac{14\,v}{5} + \frac{v^3}{15}\right\}$$

上の vset = vs[[{1,8,17,24}]]={α + 3 β - γ, 3 α + β - δ, -α + γ + 3 δ, -β + 3 γ + δ}=$\left\{v,\ -v,\ \frac{14\,v}{5} - \frac{v^3}{15},\ -\frac{14\,v}{5} + \frac{v^3}{15}\right\}$ が V[x] の解となる.

§4. f[x]の解のガロア群を求める

vset={v1,v2,v3,v4}とする． $\alpha,\beta,\gamma,\delta$は原始元v (=v1) の式で表されているので v1をv1,v2, v3,v4にそれぞれ変えると vsetが変化する． 例えば v1→v2とすると
vset→{-v,v,$-\frac{14\,v}{5}+\frac{v^3}{15},\frac{14\,v}{5}-\frac{v^3}{15}$}={v2,v1,v4,v3}と変わる． この例では V(x)の解:vset の置換群は
G_v={{v1,v2,v3,v4},{v2,v1,v4,v3},{v3,v4,v1,v2},{v4,v3,v2,v1}}となる． (v1の場所に従って残りのv2,v3,v4の位置は自動的に決まることに注目)

一方， この置換を f(x)の解の置換と見ることもできる． $\alpha,\beta,\gamma,\delta$は全てvの式で表されているので， v1→v2のとき， $\alpha,\beta,\gamma,\delta$は次のように変化する．
$\{\alpha,\beta,\gamma,\delta\}=\left\{\frac{4\,v}{15}-\frac{v^3}{45},-\frac{4\,v}{15}+\frac{v^3}{45},-\frac{23\,v}{15}+\frac{2\,v^3}{45},\frac{23\,v}{15}-\frac{2\,v^3}{45}\right\}\to$
$\left\{-\frac{4\,v}{15}+\frac{v^3}{45},\frac{4\,v}{15}-\frac{v^3}{45},\frac{23\,v}{15}-\frac{2\,v^3}{45},-\frac{23\,v}{15}+\frac{2\,v^3}{45}\right\}=\{\beta,\alpha,\delta,\gamma\}$
$\{\alpha,\delta,\beta,\gamma\}\Leftrightarrow\{1,2,3,4\}$ と対応させると$(1,2,3,4)\to(2,1,4,3)$ という置換になる.これを「 v1→v1,v1→v2,v1→v3,v1→v4 」の各々について実行すると， f(x)の解についての置換(GaloisによるGalois群)が得られる．

以上から分かるように Galois群の位数は V(x)の次数と一致する． $f(x)=x^4-10x^2+1$ のときは「たまたま」V(x)の次数が f(x)の次数と一致しているが， 4個の解の置換は最大4!=24通りあり， Galois群の位数=V(x)の次数=24 となり得る． 今回は簡単のためにできるだけ位数の小さくなるケースを選んだので， 一致してしまったが， 通常は一致しない． 例えば§5の例「$f(x)=x^4-2$」を参照して頂きたい． 以上を, Tableを使って実装する．

Table[*expr*, {*i*, i_{max}}]
i が1から i_{max}までの場合の *expr* の値のリストを作成する．

```
In[ ]:= sols = {α', β', γ', δ'};
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[%]
```

Out[ ]//MatrixForm=

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

これがGalois群のガロアによる表現． f(x)の解の入れ替えは一行目からスタートすると
{1,2,3,4}→{1,2,3,4},{1,2,3,4}→{4,3,2,1},{1,2,3,4}→{3,4,1,2},{1,2,3,4}→{2,1,4,3}
を表すが， これを2行目からスタートして
{4,3,2,1}→{1,2,3,4},{4,3,2,1}→{4,3,2,1},{4,3,2,1}→{3,4,1,2},{4,3,2,1}→{2,1,4,3}
と読んでも良い. 3行目/4行目からスタートしても同じになる(参考文献[5])
現代の表現では次の様になり， Galois群は「クラインの4元群」となることが分かる(Idは単位元)
{ Id, (1,2)(3,4), (1,3)(2,4), (1,4)(2,3)}

§5. SageMath & Magma &まとめ

5-1. SageMath&Magma

フリーの SageMath や Magma calculator を使うと，ガロア群はすぐ求まる．例えば SageMath では次の様になる．

```
_. < x > = PolynomialRing (QQ)
f = x^4-10*x^2 + 1
G = f . galois_group ()
G . list ()
```

Magma calculator では次の様になる. 結果は生成元で表される.

```
_<x>:=PolynomialAlgebra(Rationals());
//PolynomialRing も大丈夫
f:=x^4-10*x^2 + 1;
G:=GaloisGroup(f); G;
```

SageMathもMagmaも Web Calculator があるので，容易に試すことができる．また SageMath の方は Windows/Mac/Linux にインストールもできる．MagmaやSageMathを使って Galois群をご覧になりたい方は，**`https://mixedmoss.com/mathematica/Galois/`** により詳しい説明が置いてあります．

5-2. StepByStep Program (R(x)のチェックが必要. また極めて稀にGroebner基底がうまく作れない.)

残念ながら私の知る限り Mathematica ではMagmaやSageMathのようなGalois群を求める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 . §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の係数の変更が必要です . そのためにR (x) を出力させてます . また極めて稀に(1回/100回ぐらい)Groebner基底がうまく作れない事があります.

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[ ]:=
```
ClearAll["`*"]
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[ ]=
$$\left(334\,084 - 644\,v^4 + v^8\right)\left(2500 + 28\,v^4 + v^8\right)\left(114\,244 + 476\,v^4 + v^8\right)$$

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

Step2 . f (x)の解を原始元vで表す

In[ ]:=
```
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[ ]=
$$\left\{\frac{199\,v}{520} + \frac{v^5}{1040}, \frac{61\,v}{780} - \frac{v^5}{1560}, -\frac{199\,v}{520} - \frac{v^5}{1040}, -\frac{61\,v}{780} + \frac{v^5}{1560}\right\}$$

Step 3 . V(x)の解を原始元vで表す

In[ ]:=
```
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[ ]=

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

Step4 . Galois群を求める

In[ ]:=
```
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[ ]//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．

§6. Automated Program

「R(x)の重解問題」&「Groebner基底が時々整式にならない問題」を解消し，かつ計算を速くする為，有効数字計算も Part 3 , Part 4 で入れ，使いやすくするためにModule化しました．次数は理論上は何次でも大丈夫ですが，実際は 5 次ぐらいが限界です．詳細は gettingIntoTheProgram.nb をご覧ください．

In[ ]:=
```
galoisBase[f_,selected0_:0]:=Module[
{n,rx,allsets,coef,symm,base,contents,oldselected,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10},
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,V],PolynomialQ[#,v]&]||Discriminant[rx,v]==0,(*「R(x)が重解を持つ」∨「V,solsの
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R(x)*)
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}]
```

In[ ]:=
```
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]]]
```

galoisBaseは 分解方程式と f の解の原始元vによる表現を出力します． galoisGroup は Galois群を出力します． 5 次まではそこそこの時間(20秒ぐらい）で計算します．(4次なら0.1秒です. 6次なら少なくとも半日です．）まずは，有名な3次方程式でやって見ます．v = α + 2 β + 3 γ としてます．

In[ ]:=
```
galoisBase[x^3 - 2, {1, 2, 3}]
```

Out[ ]=

$$\left\{108 + v^6, \left\{-\frac{v}{2} - \frac{v^4}{36}, \frac{v^4}{18}, \frac{v}{2} - \frac{v^4}{36}\right\}\right\}$$

単に galoisBase[x^3 - 2] や galoisGroup[x^3 - 2] とすると， random に vの式が選ばれます . vの式は selected に入っています．

In[ ]:=
```
galoisBase[x^3 - 2]
selected
```

Out[ ]=

$$\left\{1372 + 40\, v^3 + v^6, \left\{-\frac{55\, v}{126} - \frac{v^4}{252}, \frac{13\, v}{42} + \frac{v^4}{84}, \frac{8\, v}{63} - \frac{v^4}{126}\right\}\right\}$$

Out[ ]=

$$\{-1, 1, 2\}$$

In[ ]:=
```
galoisGroup[x^3 - 2]
```

Out[ ]//MatrixForm=

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

5 次までは大丈夫です． 最大30 秒かかります．

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

Out[ ]=

$$\left\{19.0938, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 5 & 2 & 4 \\ 1 & 4 & 2 & 5 & 3 \\ 1 & 5 & 4 & 3 & 2 \\ 2 & 1 & 3 & 5 & 4 \\ 2 & 3 & 4 & 1 & 5 \\ 2 & 4 & 5 & 3 & 1 \\ 2 & 5 & 1 & 4 & 3 \\ 3 & 1 & 5 & 4 & 2 \\ 3 & 2 & 4 & 5 & 1 \\ 3 & 4 & 1 & 2 & 5 \\ 3 & 5 & 2 & 1 & 4 \\ 4 & 1 & 2 & 3 & 5 \\ 4 & 2 & 5 & 1 & 3 \\ 4 & 3 & 1 & 5 & 2 \\ 4 & 5 & 3 & 2 & 1 \\ 5 & 1 & 4 & 2 & 3 \\ 5 & 2 & 1 & 3 & 4 \\ 5 & 3 & 2 & 4 & 1 \\ 5 & 4 & 3 & 1 & 2 \end{pmatrix}\right\}$$

galoisgroup　には 数字のリストとして入っています． 従って 位数は Length で求まります．

In[ ]:=
```
Length[galoisgroup]
```

Out[ ]=

```
20
```