MATEMÁTICA UNIVERSITÁRIA
nº. 22/23 – junho/dezembro 1997 – pp.42-52

Computando extensões abelianas dos racionais utilizando o sistema GAP

Noraí R. Rocco[1]

1 Introdução

Os programas de álgebra para os cursos de bacharelado em matemática nas universidades brasileiras, geralmente incluem a teoria de Galois (em característica zero) como um dos últimos tópicos a serem abordados. Na maioria das vezes, devido às limitações impostas, entre outros, pelo fator tempo, essa abordagem contempla, basicamente, a determinação de grupos de Galois de certos polinômios, o controle de subcorpos do corpo de decomposição de um polinômio via a correspondência de Galois e, finalmente, a exibição de um ou dois exemplos de polinômios (de grau 5) com grupo de Galois não-solúvel.

Entretanto, entendemos ser oportuno e instrutivo explorar-se ainda nesse contexto a realização dos grupos abelianos finitos como grupos de Galois sobre os racionais.

Tal realização desses grupos, a propósito, constitui uma das situações de mais simples solução do (caso clássico do) "problema inverso da teoria de Galois", o qual refere-se à questão *se todo grupo finito pode ser realizado como grupo de Galois de um polinômio sobre os racionais.* Este problema foi proposto por Hilbert em 1892 e está ainda em aberto, não obstante todo o progresso feito no assunto durante os últimos quinze ou vinte anos. O próprio Hilbert provou que os grupos simétricos são grupos de Galois de polinômios racionais, se bem que uma maneira mais simples e construtiva de ver isto encontra-se em van der Waerden [9],

[1]Com apoio do projeto Álgebra junto à FAP-DF

§61 (veja também [5] para maiores detalhes). Um resultado importante, tido como pedra angular no assunto, é um profundo teorema devido a Shafarevich [10]: *todo grupo solúvel finito é grupo de Galois sobre os racionais* (veja também Išhanov [11]). No início dos anos 80 uma nova investida de ataque ao problema foi lançada por Fried e outros (cf. [6], [7]) que levou a uma resposta também positiva para a maioria dos casos de grupos simples finitos esporádicos, entre outros grupos (o leitor interessado pode consultar também [2] para um apanhado dessas técnicas). Mas esta já é outra história.

O nosso objetivo aqui é tratar justamente o caso abeliano. É um exercício relativamente fácil de aplicação do teorema de decomposição dos grupos abelianos finitos, que não envolve mais do que grupos de Galois de polinômios ciclotômicos e o teorema de Dirichlet sobre primos em progressão aritmética, provar que todo grupo abeliano finito é um grupo de Galois de um polinômio racional (veja a próxima seção).

Com o crescente desenvolvimento e disponibilidade de eficientes *softwares* algébricos, torna-se cada vez mais na "ordem-do-dia" a utilização de apoio computacional no ensino da Matemática universitária, tanto na elaboração de exemplos e aplicações mais significativos como na exploração dos aspectos construtivos inerentes à nossa disciplina, onde o labor de cálculos tediosos pode ficar a cargo dos computadores. No ensino da Álgebra, em particular, não é diferente.

Nas seções seguintes desta nota procuramos dar uma idéia de utilização do sistema **GAP - Groups, Algorithms and Programming** - (Cf. [13]) na construção de exemplos concretos de corpos numéricos abelianos com um dado grupo de Galois.

2 Grupos abelianos como grupos de Galois

Com o intuito de facilitar a leitura e fixar a notação utilizada na seqüência desta nota, inserimos nesta seção uma demonstração do

Teorema 1 *Todo grupo abeliano finito é grupo de Galois sobre os racionais.*

Este fato, contudo, é bem conhecido e pode ser encontrado nos textos básicos de Álgebra Abstrata (veja p.ex. [1]). O ponto principal reside

no fato que *todo grupo abeliano finito é (isomorfo a) um quociente do grupo dos automorfismos de um grupo cíclico.*

Antes de tudo uma palavra de motivação. Um famoso resultado da teoria dos corpos numéricos, formulado por Kronecker e provado por Weber, conhecido assim como teorema de Kronecker-Weber, assegura que *toda extensão galoisiana finita* $\mathbb{F}$ *de* $\mathbb{Q}$, *com grupo de Galois abeliano* (extensão abeliana), *é subcorpo de um corpo ciclotômico* $\mathbb{Q}(\zeta_n)$, para alguma raiz n-ésima primitiva da unidade ζ_n (veja p.ex. [12] para uma demonstração "resumida" deste teorema). Se $\mathcal{G}$ denota o grupo de Galois de $\mathbb{Q}(\zeta_n)$ sobre $\mathbb{Q}$ e $\mathcal{H}$ o subgrupo de $\mathcal{G}$ correspondente ao subcorpo $\mathbb{F}$, então o grupo de Galois $\mathcal{A} = Gal(\mathbb{F}/\mathbb{Q})$ da extensão $\mathbb{F}$ é isomorfo ao grupo quociente, $\mathcal{A} \cong \mathcal{G}/\mathcal{H}$. Além disso, denotando por C_n o grupo (multiplicativo) cíclico de ordem n, sabe-se que $\mathcal{G}$ é isomorfo ao grupo dos automorfismos de C_n, $\mathcal{G} \cong Aut(C_n)$ que, a menos de isomorfismo, nada mais é do que o grupo (multiplicativo) dos resíduos relativamente primos com n, módulo n.

Assim, para um dado grupo abeliano $\mathcal{A}$, a questão de encontrar uma extensão dos racionais com grupo de Galois isomorfo a $\mathcal{A}$ pode ser facilmente resolvida se conseguirmos "descolar" um número natural n e um conveniente subgrupo N de $Aut(C_n)$ tal que $Aut(C_n)/N \cong \mathcal{A}$.

Um tal subgrupo N pode sempre ser encontrado a partir da decomposição de $\mathcal{A}$ como produto direto de grupos cíclicos (Teor. Fundamental dos Grupos Abelianos Finitos), acoplada a um adequado conjunto de primos cuja existência é garantida pelo teorema de Dirichlet sobre primos em progressão aritmética.

De fato, suponhamos que o dado grupo abeliano $\mathcal{A}$ seja decomposto num produto direto de k grupos cíclicos C_{n_i}, $i = 1, \ldots, k$:

$$\mathcal{A} \cong C_{n_1} \times \cdots \times C_{n_k}$$

Desde que, para cada $i = 1, \cdots, k$, existem infinitos primos p_i da forma $tn_i + 1$ (teorema de Dirichlet, Cf. [8]), podemos encontrar k primos distintos $p_1, \ldots, p_k$, tais que $p_i - 1 = t_i n_i$, $i = 1, \ldots, k$. Definindo n como sendo o produto desses k primos, $n := p_1 \cdots p_k$, resulta que $Aut(C_n) \cong C_{p_1-1} \times \cdots \times C_{p_k-1}$, uma vez que os primos p_i's são distintos e $Aut(C_p) \cong C_{p-1}$ para todo primo p (veja p.ex. [3]). Portanto

$$Aut(C_n) \cong C_{t_1 n_1} \times \cdots \times C_{t_k n_k}$$

Se a i-ésima componente $C_{t_i n_i}$ acima é gerada, digamos, por x_i, seja $N_i := \langle x_i^{n_i} \rangle$, para cada $i = 1, \ldots, k$. Isto é, N_i é o subgrupo de $C_{t_i n_i}$ de ordem t_i . Definimos então o subgrupo N como sendo o produto direto, $N := N_1 \times \cdots \times N_k$, obtendo finalmente

$$\begin{aligned} Aut(C_n)/N &\cong (C_{t_1 n_1} \times \cdots \times C_{t_k n_k})/(C_{t_1} \times \cdots \times C_{t_k}) \\ &\cong (C_{t_1 n_1}/C_{t_1}) \times \ldots \times (C_{t_k n_k}/C_{t_k}) \\ &\cong C_{n_1} \times \ldots \times C_{n_k} \\ &\cong \mathcal{A} \end{aligned}$$

3 Usando o sistema GAP

GAP (cf. [13]) é um sistema para se programar e computar em Álgebra Discreta. Especializado para apoio computacional em teoria dos grupos, este sistema teve início no RWTH - Aachen, RFA - em 1986, e tem tido um crescimento vigoroso e continuado nos últimos dez anos, com contribuições de diversas pessoas de diferentes instituições e vem proporcionando a cada dia mais e mais funções para se computar com estruturas algébricas em geral. É um sistema de fácil portabilidade que traz embutida uma linguagem de programação (da família Pascal), na qual estão escritas a maioria das funções disponíveis na vasta biblioteca que o acompanha. Embora não seja considerado de domínio público, pode ser obtido a **custo zero** via *ftp*, para diversos ambientes e com toda a documentação, num dos *servidores* (entre outros):

`ftp-gap.dcs.st-and.ac.uk`:
School of Mathematical and Computational Sciences
University of St Andrews, Escócia
diretório: `/pub/gap/gap/`

`ftp.math.rwth-aachen.de`:
Lehrstuhl D für Mathematik, RWTH Aachen, RFA
diretório: `pub/gap/`

`math.ucla.edu`:
Math. Dept., Univ. of California at Los Angeles, USA,
diretório: `/pub/gap/`

O procedimento descrito na seção anterior pode ser efetivado com relativa rapidez com o uso do GAP para se produzir inúmeros exemplos. No que segue procuramos manter a notação introduzida na seção 2.

Para exemplificar, vamos supor que o dado grupo abeliano $\mathcal{A}$ seja um produto direto de 3 grupos cíclicos, isto é, de acordo com a notação da seção anterior, $k = 3$ e $\mathcal{A} \cong C_{n_1} \times C_{n_2} \times C_{n_3}$.

Passo 1. Neste primeiro passo encontramos um inteiro n com um procedimento bem simples, escrevendo uma adequada função de três variáveis na linguagem GAP.

Observação. O símbolo gap> que aparece no início de um procedimento ou de uma linha de comandos indica o *prompt* do sistema durante uma sessão interativa. Cada linha de comando(s) no GAP sempre termina com um ponto-e-vírgula; um duplo ponto-e-vírgula evita o eco na tela.

```
gap> pp:=function(n1, n2, n3)
local t1, p1, t2, p2, t3, p3, n;
t1:=0; p1:=1; t2:=0; p2:=1; t3:=0; p3:=1;
repeat
p1:=p1+n1; t1:=(p1-1)/n1;
until IsPrime(p1);
repeat
p2:=p2+n2; t2:=(p2-1)/n2;
until IsPrime(p2) and p2 <> p1;
repeat
p3:=p3+n3; t3:=(p3-1)/n3;
until IsPrime(p3) and p3 <> p1 and p3 <> p2;
n:=p1*p2*p3;
Print(" ", "n = ", n,",", " ", "t1 = ", t1,",", " ",
"t2 = ", t2,","," ", "t3 = ", t3,".", "\n");
return;
end;
function ( n1, n2, n3 ) ... end
```

Por exemplo, se $\mathcal{A} \cong C_3 \times C_6 \times C_{12}$, temos

```
gap> n1:=3;; n2:=6;; n3:=12;;
gap> pp(n1,n2,n3);
 n = 3367, t1 = 2, t2 = 2, t3 = 3.
```

Para registrar os valores acima atribuimos os mesmos às respectivas variáveis:

```
gap> n:=3367;; t1:=2;; t2:=2;; t3:= 3;;
```

Certamente podemos checar no GAP a fatoração de n em primos (no caso, $n = p1 * p2 * p3$):

```
gap> Collected(Factors(n));
[ [ 7, 1 ], [ 13, 1 ], [ 37, 1 ] ]
```

Passo 2. Uma vez encontrado o número n, gera-se o corpo ciclotômico $\mathbb{Q}(\zeta_n)$ utilizando-se a função *CyclotomicField*, `CF(n)`:

```
gap> K:=CF(n);
CF(3367)
```

No GAP uma estrutura algébrica é um **domínio** (conjunto estruturado) e é assim armazenada na forma de um *record* (*registro*). Constituido por *campos*, um *record* é um dos principais tipos de estrutura de dados no sistema. Os campos armazenam muitas informações sobre a estrutura computada. Para verificar quais campos já estão computados e registrados, usa-se o comando `RecFields`:

```
gap> RecFields(K);
[ "isDomain", "isField", "isCyclotomicField", "char",
"degree", "generators",   "zero", "one", "size",
"isFinite", "field", "dimension", "base", "isIntegralBase",
"zumbroichbase", "stabilizer", "operations" ]
gap> K.degree;
2592
gap> l:=K.base;;
gap> Length(K.base)=K.degree;
true
```

Note que o campo `"base"` acima é uma lista (no caso, um conjunto) de cardinalidade igual a $\Phi(n)$, onde Φ é a função de Euler. A raiz n-ésima primitiva $\zeta_n = e^{2\pi i/n}$, gerador do corpo ciclotômico, é calculada pela função `E`; no caso temos $E(3367) = \zeta_{3367}$. As 2592 raízes primitivas que constituem a base do corpo são as potências de $E(3367)$ cujos expoentes (resíduos relativamente primos mod n) estão armazenados no campo `"zumbroichbase"` do registro `K`. A i-ésima entrada de uma lista `lis` é obtida por `lis[i]`.

```
gap> Phi(n);
2592
gap> Length(K.zumbroichbase);
2592
gap> K.zumbroichbase[1];
1
gap> K.zumbroichbase[2592];
3366
gap> Length(K.generators);
1
gap> K.generators[1];
E(3367)
gap> K.isIntegralBase;
true
```

Uma vez computado o corpo ciclotômico, computa-se o seu grupo de Galois com o comando `GaloisGroup`. Tal grupo é apresentado pelos seus geradores, que são *Number Field Automorphisms*. Seguindo o exemplo, obtemos

```
gap> G:=GaloisGroup(K);
Group( NFAutomorphism( CF(3367) , 2887 ),
NFAutomorphism( CF(3367) , 3109 ),
NFAutomorphism( CF(3367) , 2185 ))
```

Para simplicar a leitura, vamos atribuir um nome ao grupo. Isto certamente acrescentará ao registro `G` o campo `"name"`; calculamos também a ordem de `G` com a função `Size`:

```
gap> G.name:="G";
"G"
gap> RecFields(G);
[ "isDomain", "isGroup", "identity", "generators",
"operations", "1", "2",  "3", "name" ]
gap> Size(G);
2592
gap> IsAbelian(G);
true
```

Passo 3. Para encontrar agora o subgrupo N $\leq$ G segundo a seção 2, vamos calcular, para $i = 1, 2, 3$, a potência n_i do i-ésimo gerador de G. Os geradores estão armazenados no campo "generators", que é uma lista, no registro G (ou então, neste caso, nos campos "1", "2" , "3"). Seguindo a notação anterior, denominamos esses geradores por x_i, $i = 1, \ldots, 3$, calculamos suas ordens em G e obtemos os geradores y_i de N, $y_i = x_i^{n_i}$, $i = 1, \ldots, 3$.

```
gap> x1:=G.generators[1];
NFAutomorphism( CF(3367) , 2887 )
gap> x2:=G.generators[2];
NFAutomorphism( CF(3367) , 3109 )
gap> x3:=G.generators[3];
NFAutomorphism( CF(3367) , 2185 )
gap> Order(G, x1);
6
gap> Order(G, x2);
12
gap> Order(G, x3);
36
gap> y1:=x1^n1;;
gap> y2:=x2^n2;;
gap> y3:=x3^n3;;
```

Para obter o subgrupo N basta usar a função Subgroup:

```
gap> N:=Subgroup( G,  [y1,y2,y3] );;
gap> Size(N);
12
```

O grau da extensão procurada F é igual à ordem do grupo quociente G/N:

```
gap> Size(G/N);
216
```

Passo 4. Pelo que sabemos da teoria de Galois, o corpo procurado F é o subcorpo de K fixado por N (cf. [4]), cujo grupo de Galois é então

isomorfo ao dado grupo $\mathcal{A}$. Para encontrarmos esse subcorpo F $\subset$ K, primeiro observamos que se ℓ é uma $\mathbb{Q}$-base de K, então N permuta os elementos de ℓ, de modo que a soma dos elementos de cada N-órbita é fixado por N; reciprocamente, se um elemento $\alpha \in$ K é fixado por N então, pela unicidade das coordenadas de α na base ℓ, conclui-se que essas coordenadas são constantes para os elementos básicos que compoem uma mesma N-órbita. Consequentemente, o conjunto constituido pelas somas dos elementos de cada órbita forma uma $\mathbb{Q}$-base de F.
No GAP, o cálculo de órbitas é feito com o comando `Orbit`, enquanto a soma de todos os elementos de uma lista é dada pelo comando `Sum`. Na linha de comandos abaixo, calculamos uma lista que tem uma órbita (sob a ação de N) em cada posição, aproveitando para isto a base `l` de K já fornecida anteriormente. O comando `Set` transforma uma lista num conjunto, evitando-se redundâncias indesejáveis.

```
gap> s:=List([1..K.degree], i -> Set(Orbit(N, l[i],
OnPoints)));;
gap> s:=Set(s);;
gap> Length(last);
216
gap> ls:=List([1..216], i -> Sum(s[i]));;
```

De acordo com a observação anterior, a lista `ls` acima é então a base do nosso futuro corpo F. Note que `ls` é um conjunto com 216 elementos, de modo que usamos os dois pontos-e-vírgulas para evitar o eco na tela. Para satisfazer a curiosidade, vejamos como são o primeiro e o último elementos de `ls`:

```
gap> ls[1];
E(3367)^27+E(3367)^64+E(3367)^545+E(3367)^1639+
E(3367)^2120+E(3367)^2157+E(3367)^2367+E(3367)^2638+
E(3367)^2848+E(3367)^2885+E(3367)^2913+E(3367)^3366
gap> ls[216];
E(3367)^449+E(3367)^708+E(3367)^722+E(3367)^930+
E(3367)^981+E(3367)^1086+E(3367)^1189+E(3367)^1203+
E(3367)^1345+E(3367)^1462+E(3367)^1567+E(3367)^1826
```

Isto é, denotando simplesmente por ζ a raíz ζ_{3367}, vemos que o primeiro elemento da base de F é

$$\zeta^{27} + \zeta^{64} + \zeta^{545} + \zeta^{1639} + \zeta^{2120} + \zeta^{2157} + \zeta^{2367} + \zeta^{2638} + \zeta^{2848} + \zeta^{2885} + \zeta^{2913} + \zeta^{3366}$$

Para finalizar, usamos o comando `NumberField` para construir o corpo procurado F e computar o seu grupo de Galois:

```
gap> F:=NumberField(ls);
NF(3367,[ 1, 454, 482, 519, 729, 1000, 1210, 1247, 1728,
2822, 3303, 3340 ])
gap> F.degree;
216
gap> A:=GaloisGroup(F);;
gap> IsAbelian(A);
true
gap> Order(A, A.1);
3
gap> Order(A, A.2);
6
gap> Order(A, A.3);
12
gap>3*6*12;
216
```

Assim, o corpo encontrado F é de fato uma extensão abeliana dos racionais, com grupo de Galois isomorfo ao dado grupo $\mathcal{A}$.

Referências

[1] L.J.Goldstein, *Abstract Algebra - a first course*, Prentice Hall, 1973.

[2] H.Pahlings and N.Rocco, *On the inverse problem of Galois theory*, Trabalho de Matemática nº 286, MAT/UnB, 1995, 41 pp.

[3] J.J.Rotman, *An Introduction to the Theory of Groups*, third ed., Allyn and Bacon, 1984.

[4] A. Gonçalves, *Introdução à Álgebra*, Projeto Euclides, IMPA, 1979

[5] S. B. de Freitas, *Algumas Técnicas computacionais para determinação de grupos de galois sobre os racionais*, Dissertação de Mestrado, MAT/UnB, 1991.

[6] B.H.Matzat, *Konstruktive Galoistheorie*, Lecture Notes in Mathematics $n^{\underline{o}}$ 1284, Springer, 1987.

[7] J.-P. Serre, *Topics in Galois Theory*, A.K.Peters, Wellesley, 1992.

[8] J.-P. Serre, *A Course in Arithmetic*, Springer, 1979.

[9] B.L. van der Waerden, *Modern Algebra*, vol. I, Frederick Ungar Publishing Co., 1953;

[10] I.R. Shafarevich, *Construction of fields of algebraic numbers with given solvable Galois group*, Izv. Akad. Nauk SSSR Ser. Mat. **18** (1954), 525-578; Amer. Math. Soc. Transl. **4** (1956), 185-237;

[11] V.V.Išhanov, *On the semidirect imbedding problem with nilpotent kernel*, Izv. Akad. Nauk SSSR Ser. Mat. **40** (1976), Math USSR Izvestija **10** (1976), 1-23;

[12] M.J.Greenberg, *An elementary proof of the Kronecker-Weber theorem*, Amer. Math. Monthly **81** n.6 (1974), 601-607, **82** n.8 (1975), 803;

[13] M. Schönert, et. al., *GAP – Groups, Algorithms and Programming*, Lehrstuhl D für Mathematik, RWTH - Aachen, Germany, third ed., 1993.

Universidade de Brasília
Departamento de Matemática
70.910-900 Brasília - DF.
e-mail: *norai@mat.unb.br*