Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 5, N. 1, 2017.

Trabalho apresentado no CNMAC, Gramado - RS, 2016.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Classificação dos Grupos de Ordem Pequena com o Auxílio do GAP

Rosemary Miguel Pires[1]
Departamento de Matemática, Instituto de Ciências Exatas, UFF, Volta Redonda, RJ
Túlio Joaquim Altoé[2]
Graduando em Matemática com ênfase em Matemática Computacional, Instituto de Ciências Exatas, UFF, Volta Redonda, RJ

Resumo.
Neste trabalho são apresentados alguns algoritmos para serem implementados com o sistema computacional GAP (Groups, Algorithms, Programming) que auxiliam na construção da Tabela de Classificação dos Grupos de ordem pequena. Diversas metodologias diferentes podem ser elaboradas para o ensino da Álgebra Abstrata no Ensino Superior usando os programas computacionais existentes. Por exemplo, com o sistema GAP, vários temas da área podem ser melhor estudados e aprofundados, tendo em vista que vários dos resultados algébricos nem sempre são fáceis de serem ilustrados e visualizados manualmente. Mostramos como atividades simples com o uso do GAP ajudam no estudo da Teoria de Grupos. Por fim, como aplicação da tabela construída, são utilizadas algumas funções do GAP para acessar seus elementos de modo que seja possível determinar, por exemplo, subgrupos, subgrupos normais, subgrupos de Sylow e tabela de multiplicação de grupos.

Palavras-chave. Grupos, Classificação de Grupos, GAP

1 Introdução

Nos cursos de Álgebra voltados para a graduação são abordados vários conceitos e resultados básicos da Teoria de Grupos, tais como subgrupos, grupos quocientes, homomorfismos de grupos, Teorema de Lagrange etc. Um dos resultados mais importantes desta teoria é o Teorema de Sylow. Ao longo dos anos, através da experiência em sala de aula, percebemos que os alunos encontram uma certa dificuldade para entender a aplicação deste teorema no estudo da classificação dos grupos, que é um dos problemas mais importantes da Teoria de Grupos. Existem muitos livros que apresentam um estudo completo e detalhado sobre este problema [1–3]. Com relação aos grupos abelianos finitos, o Teorema Fundamental dos Grupos Abelianos Finitos nos permite realizar a classificação destes grupos, a menos de isomorfismo [2, 4]. Por outro lado, os grupos não abelianos são casos mais complexos de se determinar algebricamente a classificação dos grupos de toda ordem.

[1]rosemarypires@id.uff.br
[2]tulioaltoe@id.uff.br

A finalidade deste trabalho é apresentar algoritmos computacionais para determinação de uma tabela com a classificação dos grupos de ordem pequena. Para realizar este trabalho, utilizamos a ferramenta computacional GAP(Groups, Algorithms, Programming), um sistema para Álgebra Discreta Computacional, disponível em http://www.gap-system.org/, distribuído gratuitamente e com uma linguaguem de programação e uma biblioteca com diversas funções que implementam algoritmos algébricos.

2 Implementando Algoritmos com o sistema GAP

O GAP possui uma biblioteca com vários grupos, entretanto, o usuário precisa ter um conhecimento prévio de programação para acessá-los. Sendo assim, desenvolvemos uma aplicação para apresentar os grupos, a menos de isomorfismo, e exibir uma série de recursos para que o usuário trabalhe com qualquer grupo apresentado na tabela. A diferença da aplicação desenvolvida em relação ao que o próprio GAP oferece é que o usuário precisa de ter apenas um mínimo de conhecimento em programação para executar as mesmas operações introduzidas no algoritmo.

2.1 Classificação de Grupos Finitos

Sabemos que para a classificação de um grupo abeliano G é necessário, primeiramente, fazer a decomposição de G em soma direta de p-subgrupos de Sylow de G, e depois fazer a decomposição de cada p-grupo como soma direta de subgrupos cíclicos de G, de forma que se obtenha a decomposição de G em soma direta de grupos cíclicos. Estas duas decomposições determinam o Teorema Fundamental dos Grupos Abelianos Finitos, conforme [3, 4]. Por outro lado, não existe um modo semelhante para a classificação dos grupos não abelianos. Em [3], podemos encontrar os principais Teoremas de classificação. Neste trabalho, vamos assumir algumas definições básicas da Teoria de Grupos.

2.2 Tabela de Classificação de Grupos de ordem pequena

Como queremos apenas mostrar algumas funcionalidades do GAP para o estudo de Álgebra, determinaremos a tabela de classificação apenas dos grupos de ordem ≤ 15. Lembremos que todo grupo de ordem prima é cíclico e, assim, existe um único grupo (a menos de isomorfismo) com esta ordem. Além disso, existem, a menos de isomorfismo, exatamente 2 grupos de ordem 4, 6, 9, 10 e 14, e exatamente 5 grupos de ordem 8 e 12, conforme [1].

Com este objetivo, inicialmente, vamos ilustrar alguns comandos básicos disponíveis na biblioteca do GAP que são utilizados para a determinação de grupos de ordem n. Todas estas funções podem ser encontradas e melhor explicadas no manual do GAP. A função **AllSmallGroups** nos apresenta uma lista de todos os grupos de ordem n. A função **SmallGroup** contém uma lista de vários grupos que são organizados numa certa ordem sequencial preestabelecida pelo próprio programa; esta função retorna o grupo com a “ordem” escolhida e o apresenta na posição “pos” correspondente na sequência. Por sua vez, a função **StructureDescription** retorna uma *string* que descreve a representação do grupo G no GAP.

Para exemplificar como estas funções são executadas no GAP, relembremos que existem dois grupos a menos de isomorfismo com ordem igual a 4, são eles: $G_1 = C_4$ e $G_2 = C_2 \times C_2$; existem dois grupos a menos de isomorfismo com ordem igual a 6, são eles: $G_1 = S_3$ e $G_2 = C_6$; e existem cinco grupos a menos de isomorfismo com ordem igual a 8, são eles: $G_1 = C_8$, $G_2 = C_4 \times C_2$, $G_3 = D_8$, $G_4 = Q_8$ e $G_5 = C_2 \times C_2 \times C_2$. Vejamos a representação no GAP de alguns destes grupos usando as funções descritas acima:

gap> StructureDescription(SmallGroup(4,2)); StructureDescription(SmallGroup(8,3));
"$C2 \times C2$" "$D8$"

O comando "StructureDescription(SmallGroup(n,i))" nos dá a representação do grupo de ordem n que está na posição i da lista de grupos de ordem n determinados com o GAP.

A seguir apresentamos um algoritmo para determinar os grupos não isomorfos de ordem n com o auxílio do GAP.

1. Escolha a ordem de um grupo: n.

2. Use o comando **lista:= AllSmallGroups(**n**);** para encontrar uma lista com todos os grupos, a menos de isomorfismos, de ordem n.

3. Use o comando **Size(lista);** para encontrar o tamanho da sua lista, ou seja, o número de grupos não isomorfos de ordem n.

4. Percorra toda a sua lista para encontrar o elemento x que está na posição i ($1 \leq i \leq Size(lista)$) usando o comando **lista[i]**.

5. Encontre a representação do grupo com o comando **StructureDescription(lista[i])** para cada $i = 1, \cdots, Size(lista)$.

Para listar o grupo de ordem n que está na posição i, podemos usar o comando **SmallGroups(n,i);**. Também podemos determinar de modo direto a descrição (ou representação) do grupo de ordem n que está na posição i, para isto basta entrar com o comando **StructureDescription(SmallGroup(n,i));**. Caso não tenha um grupo com uma determinada ordem numa certa posição escolhida, o GAP exibirá uma mensagem de erro.

Agora vamos apresentar uma função criada a partir das funções introduzidas acima com o objetivo de determinar a classificação dos grupos de ordem n. Com esta função mais simples (Grupo(n)), para conhecermos todos os grupos não isomorfos de uma determinada ordem, basta executar dois comandos no GAP, que são: **Read("caminho/nome");** e **Grupo(n);**. As etapas do algoritmo são apresentadas a seguir:

```
Grupo:=function(ordem)
local lista,apresent,i;
lista:=AllSmallGroups(ordem);
apresent:=List(lista,x->StructureDescription(x));
Print(``\t'',ordem,``\t'');
```

```
for i in [1..Size(apresent)] do
Print(apresent[i],``;'');
od;
Print(``\n'');
end;
```

Usando a função criada acima, a tabela de classificação pode ser encontrada usando o seguinte algoritmo:

```
TabelaClassGrupos:= function(ordemmax)
local i,ord;
ord:=ordemmax;
Print(``\n\t\t Ordem \t Grupo \n'');
for i in [1..ord] do
Print(``\t''); Grupo(i);
od;
end;
```

Para gerar a tabela, basta executar o algoritmo dado, escolher o valor de n e depois entrar com o comando para executar a função definida no algoritmo (**TabelaClassGrupos(n);**). Como ilustração, para $n = 15$, obtemos a seguinte tabela com a classificação de todos os grupos não isomorfos com ordens de 1 até 15:

```
gap> TabelaClassGrupos(15);
```

ORDEM	GRUPO
1	1;
2	C2;
3	C3;
4	C4; C2 x C2;
5	C5;
6	S3; C6;
7	C7;
8	C8; C4 x C2; D8; Q8; C2 x C2 x C2;
9	C9; C3 x C3;
10	D10; C10;
11	C11;
12	C3 : C4; C12; A4; D12; C6 x C2;
13	C13;
14	D14; C14;
15	C15;

3 Aplicação da Tabela de Classificação de Grupos

Nesta etapa, mostramos como acessar qualquer grupo exibido na tabela de classificação e ilustramos como certos conteúdos de Álgebra podem ser melhor compreendidos através dos recursos que o GAP oferece. Primeiramente, apresentamos uma lista com todos os algoritmos elaborados para facilitar o cálculo de subgrupos, subgrupos normais, subgrupos de Sylow, Tabela de Multiplicação de um grupo. Também mostramos como verificar se um dado grupo é abeliano.

- Cálculo de Subgrupos:

  ```
  Subgrupos:= function(ordem, pos)
  local g,h,lista;
  g:= SmallGroup(ordem,pos);
  h:= Subgroups(g);
  Print(``\n\t Subgrupos \n\n");
  Print(``\t  —> ",h);
  lista:= List(h, x-> StructureDescription(x));
  Print(``\n\n\t Subgrupos \n\n");
  Print(``\t  —> ",lista,``\n\n");
  end;
  ```

  Para executar algumas funções como esta, é necessário, primeiramente, "carregar" o pacote chamado **Sonata** através do comando: **LoadPackage("sonata")**.

- Cálculo dos Subgrupos Normais:

  ```
  SubgruposNormais:= function (ordem,pos)
  local g,h,i,listanormais,normais;
  listanormais:=[];
  g:= SmallGroup(ordem,pos);
  h:= Subgroups(g);
  for i in [1..Size(h)] do
  if IsNormal( g, h[i] )= true then
          Add(listanormais,h[i]);
  fi;
  od;
  normais:= List(listanormais, x -> StructureDescription(x));
  Print(``\n\t", normais, ``\n\n");
  end;
  ```

- Cálculo dos Subgrupos de Sylow:

  ```
  SubgruposDeSylow:= function (ordem,pos,primo)
  local g,h,lista;
  g:= SmallGroup(ordem,pos);
  h:= SylowSubgroup(g,primo);
  Print(``\n \t Grupo: ", StructureDescription(g),``\n");
  Print(``\n\n\t ",primo,``-Subgrupo de Sylow \n\n");
  Print(``\t  —> ",StructureDescription(h),``\n\n");
  end;
  ```


- Cálculo da Tabela de Multiplicação:

```
TabelaMultiplicacao:= function(ordem,pos)
local g;
g:= SmallGroup(ordem,pos);
Print(``\n \t Grupo: ", StructureDescription(g),``\n\n");
PrintTable(g);
Print(``\n\n");
end;
```

- Para determinar se o grupo é abeliano:

```
Abeliano:= function (ordem)
local lista,i;
lista:= AllSmallGroups(ordem);
Print(``\n\t\t Grupo \t\t Abeliano\n");
for i in [1..Size(lista)] do
Print(``\n\t\t", StructureDescription(lista[i]));
Print(``\t\t", IsAbelian(lista[i]),``\n");
od;
end;
```

Observemos que, sempre que for necessário, precisamos repetir a execução dos mesmos algoritmos. Por isto, para facilitar o trabalho, podemos escrever em algum editor de texto (por exemplo, o 'Gedit') todos os algoritmos apresentados acima e depois salvar com os seguintes nomes: 'subgrupos.g', 'subgruposnormais.g', 'subgruposdesylow.g', 'tabelamultiplicacao.g' e 'abeliano.g'. Os algoritmos apresentados anteriormente para se determinar os grupos de ordem n e a respectiva tabela de classificação salvamos como 'grupo(n).g' e 'tabclassificacao.g'. A seguir, apresentamos os passos para acessar os elementos da tabela através do GAP e com uso dos documentos salvos com a extensão do 'Gedit':

1. Leia o documento 'grupo(n).g' e, a seguir,'tabclassificacao.g'.

2. Leia os documentos: 'tabelamultiplicacao.g', 'subgrupos.g', 'subgruposnormais.g', 'subgruposdesylow.g' e 'abeliano.g'.

3. Escolha n e execute a função: **TabelaClassGrupos(n);**. Serão exibidos todos os grupos, a menos de isomorfismos, para cada ordem i, com $1 \leq i \leq n$.

4. Escolha um grupo da tabela; pegue a sua posição ('grupo1'; 'grupo2'; ...) e observe a ordem do grupo escolhido.

5. Para determinar a tabela de multiplicação do grupo escolhido, utilize a função: **TabelaMultiplicacao(ordem,posição);**.

6. Para determinar os subgrupos (subgrupos de sylow) do grupo escolhido, utilize a função: **Subgrupos(ordem,posição);** (**SubgruposDeSylow(ordem,posição,primo);**).

7. Para saber se o grupo é abeliano e encontrar os subgrupos normais, execute a função: **SubgruposNormais(ordem,posição);**.

Como exemplo, vejamos a tabela de multiplicação do grupo cíclico de ordem 4:

```
gap> TabelaMultiplicacao(4,1);
Let:
g0 := <identity> of ...
g1 := f1
g2 := f2
g3 := f1*f2

   *  | g0  g1  g2  g3
 ---------------------
  g0  | g0  g1  g2  g3
  g1  | g1  g2  g3  g0
  g2  | g2  g3  g0  g1
  g3  | g3  g0  g1  g2
```

4 Conclusões

Neste trabalho mostramos como é possível usar de outros recursos metodológicos que possibilitem o ensino de Álgebra Abstrata no Ensino Superior. Usando o sistema computacional GAP criamos algoritmos mais simples usando as próprias funções do sistema que podem ser aplicados para ajudar na exemplificação e visualização de vários dos teoremas clássicos relacionados a Classificação de grupos. Determinamos uma tabela de Classificação de Grupos de ordem pequena, a menos de isomorfismo, e apresentamos aplicações para serem executadas no GAP que facilitam o acesso aos grupos de determinada ordem e que auxiliam no estudo de alguns conceitos básicos da Teoria de Grupos. Por fim, além de promover o estudo da Álgebra, mostramos como é possível complementar o estudo teórico apresentado nos livros recorrendo as simulações computacionais para auxiliar na abstração dos conteúdos algébricos.

Referências

[1] A. Garcia e Y. Lequain. *Elementos da Álgebra.* IMPA, Rio de Janeiro, 2007.

[2] I. N. Herstein. *Topics in Algebra.* John Wiley and Sons, 2nd Edition, 1975.

[3] T. W. Hungeford. *Algebra.* Springer, 8th Edition 1974.

[4] J. J. Rotman. *An Introduction to the Theory of Groups.* Springer, 4th Edition, 1995.

[5] The GAP Group, GAP - Groups, Algorithms, and Programming, Version 4.4 ; 2004, http://www.gap-system.org.

DOI: 10.5540/03.2017.005.01.0241