16 Algebra

16.8 The Character Table of a Finite Group (9 units)

This project is related to material in the Part II course Representation Theory.

1 Introduction

Let G be a finite group. A representation (ρ, V) of G consists of a finite dimensional complex vector space V and a group homomorphism $\rho : G \to \mathrm{GL}(V)$. If V is m-dimensional then we may identify $\mathrm{GL}(V)$ with $\mathrm{GL}(m, \mathbb{C})$; the group of $m \times m$ invertible matrices over $\mathbb{C}$. The character of ρ is the function $\chi : G \to \mathbb{C}$ given by $\chi(g) = \mathrm{tr}\ \rho(g)$. It is known that representations are uniquely determined (up to equivalence) by their characters.

Let G have conjugacy classes $C_1, \ldots, C_r$. The character table of G is the $r \times r$ complex matrix with entries $\chi_i(g_j)$ where $\chi_1, \ldots, \chi_r$ are the irreducible characters of G and $g_1, \ldots, g_r$ are representatives for the conjugacy classes. The character table conveys a great deal of information about the group G. For example, it can be used to decompose any given character as a sum of irreducibles, or to find the normal subgroups of G.

2 Permutation groups

A permutation π of $X = \{1, \ldots, n\}$ is a bijective function from X to X. If x is an element of X then the image of x under π is written πx. If π_1 and π_2 are permutations then their product $\pi_1 \cdot \pi_2$ maps x to $\pi_1(\pi_2 x)$. The set of all permutations of X is the symmetric group S_n. A permutation group is a subgroup of S_n for some n. We specify a permutation group by giving a (usually very small) set of generating permutations $\pi_1, \ldots, \pi_t$.

Question 1 Write a program to compute the conjugacy classes in a permutation group. The program should output the group order, the conjugacy class sizes and a representative for each conjugacy class. Test your program for $G = S_3$.

The permutation groups considered in this project will have order less than 10000. So it should be feasible to store all the elements of G. However you should still try to make your program reasonably efficient. What is the complexity of your method in terms of $|G|$, t and n (where for example storing a permutation is $O(n)$ operations)?

3 Burnside's algorithm

A formal sum $\sum_{g \in G} \lambda_g g$ (where $\lambda_g \in \mathbb{C}$) belongs to the centre $Z(\mathbb{C}[G])$ of the group ring $\mathbb{C}[G]$ if and only if the function $g \mapsto \lambda_g$ is constant on conjugacy classes. Thus $Z(\mathbb{C}[G])$ is a complex vector space with basis $b_1, \ldots, b_r$ where $b_i = \sum_{g \in C_i} g$.

Question 2 Write a program to determine the integers ν_{ijk} such that

$$b_i b_j = \sum_{k=1}^{r} \nu_{ijk} b_k$$

for all $1 \leqslant i, j \leqslant r$. Prove that if N_i is the matrix with (j, k)-entry ν_{ijk} then the matrices $N_1, \ldots, N_r$ pairwise commute. Illustrate for the example in Question 1.

Question 3 Let (ρ, V) be an irreducible representation of G with character χ. An argument using Schur's lemma shows that $\rho(b_i) = \sum_{g \in C_i} \rho(g)$ is a scalar matrix. Determine this scalar by taking the trace, and hence show that the vector

$$\begin{pmatrix} |C_1|\chi(g_1) \\ \vdots \\ |C_r|\chi(g_r) \end{pmatrix}$$

(where $g_i \in C_i$) is an eigenvector for each of the matrices $N_1, \ldots, N_r$ in Question 2. What are the corresponding eigenvalues? Prove that some linear combination of $N_1, \ldots, N_r$ has r distinct eigenvalues.

The inner product of characters χ_1 and χ_2 is $\langle \chi_1, \chi_2 \rangle = \frac{1}{|G|} \sum_{g \in G} \chi_1(g)\overline{\chi_2(g)}$. Computing the simultaneous eigenvectors of $N_1, \ldots, N_r$ determines each row of the character table up to a scalar multiple. The scaling of each row is uniquely determined by the requirement that for each irreducible character χ we have $\langle \chi, \chi \rangle = 1$ and $\chi(1) > 0$.

In computing simultaneous eigenvectors you should be careful to avoid problems due to rounding errors. See also the note on programming at the end of the project.

Question 4 Write a program to compute the character table of a permutation group. Run your program on some alternating and symmetric groups, and on the following groups.

$$\begin{aligned}
G_1 &= \langle (1,2,7,13,8,6,4,12,10,11,5,9,3),\ (1,7,6,12,11,10,5,4,9,13,8,3,2) \rangle, \\
G_2 &= \langle (1,4,7,2)(3,13)(5,12,9,10)(6,11,8,14),\ (2,13,6)(3,7,14)(5,11,9)(8,12,10) \rangle, \\
G_3 &= \langle (1,5,8,6)(2,4),\ (1,4,8,6,7,5)(2,3) \rangle, \\
G_4 &= \langle (1,2,3,9,5,6,11,8,10,4,7),\ (1,3,7,10,4,2,9,5,11,8,6) \rangle.
\end{aligned}$$

In giving your answers you should list the characters in increasing order of dimension and the conjugacy classes in increasing order of size. You should also head each column with the size of the corresponding conjugacy class. It may help to improve the readability of your answers if you record the non-integer entries separately.

Question 5 There is a risk of rounding errors in your answers to Question 4. For each character that appears to take only integer values, how could you modify your program to be sure that the corresponding entries are correct? For one of the groups $G_1, \ldots, G_4$ use results from the course to determine the remaining entries exactly.

4 Some applications of the character table

Let $G \subset S_n$ be a permutation group. There is a natural action of G on $X^{(d)}$, the set of subsets of $X = \{1, \ldots, n\}$ of size d. Let ϕ_d be the character of the corresponding permutation representation.

Question 6 Write a program that uses the character table to decompose a given character as a sum of irreducibles. For each of the groups in Question 4 run your program on the characters ϕ_d.

Question 7 Which, if any, of the groups $G_1, \ldots, G_4$ are simple? In each case what is the smallest possible dimension of a faithful representation, and what is the smallest possible dimension of a representation with abelian kernel? What is the smallest possible index of a proper subgroup?

Programming note

Your programs for Questions 1 and 2 will need to determine whether a given permutation π appears in a list of permutations $\pi_1, \ldots, \pi_k$. You may find that comparing π to each π_i in turn is unreasonably slow. One alternative is to use a hash function, i.e. make some ad hoc choice of function h from permutations to integers that is not too far from being injective. Then use the MATLAB function `find` to find all occurrences of $h(\pi)$ from the list $h(\pi_1), \ldots, h(\pi_k)$. A better method would involve sorting the permutations (e.g. by hash value) but this should not be necessary.

Depending on the method you use for Question 4, it may help to note that the MATLAB function `rref` for Gaussian elimination accepts a tolerance as its second argument.

References

[1] L.C. Grove, *Groups and Characters*, Wiley-Interscience, 1997.

