Submitted exclusively to the London Mathematical Society
doi:10.1112/0000/000000

Computing characters of groups with central subgroups

Vahid Dabbaghian and John D. Dixon

Abstract

The so-called Burnside-Dixon-Schneider (BDS) method currently used as the default method of computing character tables in GAP for groups which are not solvable is often inefficient in dealing with groups with large centres. If G is a finite group with centre Z and λ a linear character of Z, then we describe a method of computing the set $\mathrm{Irr}(G, \lambda)$ of irreducible characters χ of G whose restriction χ_Z is a multiple of λ. This modification of the BDS method involves only $|\mathrm{Irr}(G, \lambda)|$ conjugacy classes of G and so is relatively fast. A generalization of the method can be applied to computation of small sets of characters of groups with a solvable normal subgroup.

1. *Introduction*

Let G be a finite group with a central subgroup Z, and let $\mathrm{Irr}(G)$ and $\mathrm{Irr}(Z)$ denote the sets of irreducible (ordinary) characters of G and Z, respectively. For each $\chi \in \mathrm{Irr}(G)$ the restriction χ_Z has the form $\chi(1)\lambda$ for some $\lambda \in \mathrm{Irr}(Z)$. If we define $\mathrm{Irr}(G, \lambda) := \{\chi \in \mathrm{Irr}(G) \mid \chi_Z = \chi(1)\lambda\}$, then the sets $\mathrm{Irr}(G, \lambda)$ $(\lambda \in \mathrm{Irr}(Z))$ form a partition of $\mathrm{Irr}(G)$ (see [**6**, Theorem 6.2]). We consider the problem of computing the characters in $\mathrm{Irr}(G, \lambda)$; it turns out that these can be computed independently for the different λ. This has two advantages over computing the full character table. It allows us to compute selected characters for G without computing the full character table, and even if we do need the full character table, the computation of $\mathrm{Irr}(G, \lambda)$ for all λ can be more efficient than the so-called Burnside-Dixon-Schneider (BDS) method (see [**2**] and [**10**]). The BDS method is the default method used in GAP [**4**] for groups which are not solvable (see Section 71.14 of the GAP manual). The method described in the present paper is a modification of the BDS method which takes advantage of certain structure in the group. In many cases, the new method requires that fewer characters are computed explicitly since others are easily derived from these. It also allows us to compute just a few irreducible characters at a time, and then perhaps use alternative methods to construct further characters. Although GAP already has the capability of doing this (see Section 71.17 of the GAP manual), the proposed method allows us to be selective in the choice of which characters we compute.

The following is an outline of this paper. The next two sections provide details of the computation of characters of central extensions as described above. We then make some brief comments about how this approach can be adapted to compute characters of groups with an abelian or, more generally, solvable normal subgroup using standard GAP functions. In the last section of the paper we give some statistics about the performance of an implementation of the algorithm (for central extensions) in GAP which indicate that the new method usually computes the full character table faster whenever the centre is not very small.

Remark 1. In 1991 Fischer [**3**] introduced what is now called the method of Clifford or Fischer matrices for computing the character tables of groups with normal subgroups (see [**8**,

---

2000 *Mathematics Subject Classification* 20C15 (primary), 20C40 (secondary).

The first author is partially supported by the NSERC Discovery Grant # 371974-2010.

Section 3.8]). A search of the recent mathematical literature shows that this method has been used successfully by a number of authors to compute the characters of some very large groups (typically, the normal subgroup is abelian or at most nilpotent of class 2). For example, [**9**] has calculated the character table of an extension of the second Conway group Co_2 using a hybrid of theoretical investigation and GAP computations. Another method which applies to certain classes of extensions is given in [**1**]. Although these methods can be very powerful, their use requires each case to be dealt with individually and, as yet, there seems no way to "automate" the process. In 2006 Unger [**11**] published a general method for computing character tables by inducing characters from p-elementary subgroups and then using the LLL lattice reduction algorithm to find irreducible constituents of these composite characters. An implementation in Magma [**12**] shows that Unger's method can be very successful, especially for groups which are nearly simple, and there are cases in which it can handle some significantly larger groups than the BDS method.

2. *Central extensions*

We start by fixing the notation. Let G be a finite group with centre Z. Then Z acts on the set $\mathcal{C}$ of G-conjugacy classes via right multiplication: $C \longmapsto Cz$ ($C, Cz \in \mathcal{C}$ and $z \in Z$). Let $C_1 := \{1\}, C_2, \ldots, C_r$ be representatives of the corresponding Z-orbits and for each i define $h_i := |C_i|$ and $Z_i := \{z \in Z \mid C_i z = C_i\}$ (the stabilizer of C_i in Z). Then the number $k(G)$ of G-conjugacy classes is equal to $\sum_{i=1}^{r} |Z : Z_i|$.

On the other hand, consider the set $\mathrm{Irr}(G)$ of irreducible characters of G. If e is divisible by the exponent of G and ω is a primitive eth root of 1 in $\mathbb{C}$, then the values of the characters of G all lie in $\mathbb{Q}(\omega)$ and the Galois group $\Gamma := \mathrm{Gal}(\mathbb{Q}(\omega)/\mathbb{Q})$ acts on $\mathrm{Irr}(G)$ via $\chi^\sigma(x) := \chi(x)^\sigma$ ($\sigma \in \Gamma$, $\chi, \chi^\sigma \in \mathrm{Irr}(G)$ and $x \in G$) and similarly for $\mathrm{Irr}(Z)$. As we noted in the introduction, we can partition $\mathrm{Irr}(G) = \bigcup_{\lambda \in \mathrm{Irr}(Z)} \mathrm{Irr}(G, \lambda)$, and it is clear that $\mathrm{Irr}(G, \lambda)^\sigma = \mathrm{Irr}(G, \lambda^\sigma)$ for all $\sigma \in \Gamma$. In particular, if we know $\mathrm{Irr}(G, \lambda)$, then we can find $\mathrm{Irr}(G, \lambda^\sigma)$ immediately for each Galois conjugate λ^σ.

Every finite multiplicative group of a field is cyclic and so $Z/\ker\lambda$ is cyclic for each $\lambda \in \mathrm{Irr}(Z)$, and the values of λ are the mth roots of 1 where $m := |Z/\ker\lambda|$; hence λ has exactly $\varphi(m)$ distinct Galois conjugates where $\varphi(m)$ is the Euler phi-function. Since a cyclic group of order m has exactly $\varphi(m)$ faithful characters, this also shows that two characters $\lambda, \mu \in \mathrm{Irr}(Z)$ have the same kernel if and only if λ and μ are Galois conjugates.

LEMMA 2.1. *Let $\chi \in \mathrm{Irr}(G, \lambda)$ and suppose that χ takes the value χ_i on C_i ($i = 1, \ldots, r$). If $x \in C_i z$ for some $z \in Z$, then $\chi(x) = \chi_i \lambda(z)$. Thus knowing the values χ on the classes $C_1, \ldots, C_r$ determines the value of χ on every class. Furthermore if the stabilizer Z_i is not contained in $\ker\lambda$ then $\chi_i = 0$.*

Proof. Let R be a representation of G which affords χ. Since R is irreducible, $R(z)$ is a scalar for each $z \in Z$, and since the restriction of χ to Z is equal to $\chi(1)\lambda$, we have $R(z) = \lambda(z)R(1)$. Thus if $x = uz$ where $u \in C_i$ and $z \in Z$ then $R(x) = R(u)R(z) = \lambda(z)R(u)$, and hence $\chi(x) = \lambda(z)\chi_i$ as claimed. Moreover, if $Z_i \not\leq \ker\lambda$, then choose $z \in Z_i \setminus \ker\lambda$. Now $C_i z = C_i$ but $\lambda(z) \neq 1$, so $\chi_i \lambda(z) = \chi(x) = \chi_i$ implies $\chi_i = 0$. □

LEMMA 2.2. *Let $\mathcal{K} := \{\ker\lambda \mid \lambda \in \mathrm{Irr}(Z)\}$ and for each $K \in \mathcal{K}$ define m_K to be the number of i for which $Z_i \leq K$. Then the number $k(G)$ of conjugacy classes of G equals*

$$\sum_{K\in\mathcal{K}} \varphi(|Z:K|)m_K.$$

Proof. Put $\mathcal{J} := \{(i,\lambda) \mid Z_i \leq \ker\lambda\}$. Since the number of λ with $Z_i \leq \ker\lambda$ is equal to $|\mathrm{Irr}(Z/Z_i)| = |Z:Z_i|$, we have

$$|\mathcal{J}| = \sum_{(i,\lambda)\in\mathcal{J}} 1 = \sum_{i=1}^{r} |Z:Z_i| = k(G)$$

as noted at the beginning of this section. On the other hand, as we saw above there are $\varphi(|Z:K|)$ characters λ with $\ker\lambda = K$ for each $K \in \mathcal{K}$, and so

$$|\mathcal{J}| = \sum_{\lambda\in\mathrm{Irr}(Z)} m_{\ker\lambda} = \sum_{K\in\mathcal{K}} \varphi(|Z:K|)m_K.$$

This proves the assertion. □

Now let $c_i := \sum_{u\in C_i} u$ $(i = 1,2,\ldots,r)$ be the class sums in $\mathbb{C}[G]$ of our representative conjugacy classes. The other class sums in $\mathbb{C}[G]$ are of the form $c_i z$ where $z \in Z$ (we have $c_i z = c_i z' \iff z, z'$ lie in the same Z_i-coset). Define $\nu_{ijk,z}$ to be the number of pairs $(x,y) \in C_i \times C_j$ such that $xy = zw$ for some specified $z \in Z$ and specified $w \in C_k$ (this is independent of the choice of w and depends only on the Z_k-coset to which z belongs). Then for any choice of transversals T_i of Z_i in Z $(i = 1,\ldots,r)$ the familiar formula for multiplication of class sums [**6**, Theorem (2.4)] takes the form

$$c_i c_j = \sum_{k=1}^{r} \left(\sum_{z\in T_k} \nu_{ijk,z} z \right) c_k \quad \text{for all } i,j. \tag{2.1}$$

The number of triples $(x,y,w) \in C_i \times C_j \times C_k$ such that $xy = zw$ for specified $z \in Z$ is $\nu_{ijk,z}h_k$. Since $xy = zw \iff x^{-1}w = z^{-1}y$, this shows that $\nu_{ijk,z}h_k = \nu_{i'kj,z^{-1}}h_j$ where $C_{i'}$ is the class consisting of the inverses of the elements in C_i. For $i,k,j \in \{1,2,\ldots,r\}$ we define μ_{ikj} by the condition $h_j\mu_{ikj} = h_k \sum_{z\in T_k} \nu_{ijk,z}\lambda(z)$ and so $\mu_{ijk} = \sum_{z\in T_j} \nu_{i'jk,z^{-1}}\lambda(z)$.

PROPOSITION 2.3. *Fix $\lambda \in \mathrm{Irr}(Z)$ and suppose that the classes $C_1,\ldots,C_r$ have been ordered so that $Z_i \leq \ker\lambda$ for $1 \leq i \leq m$ and $Z_i \not\leq \ker\lambda$ for $m+1 \leq i \leq r$ (in terms of Lemma 2.2, $m = m_{\ker\lambda}$). Now for $i = 1,\ldots,m$ define M_i as the $m \times m$ matrix $[\mu_{ijk}]_{j,k=1}^{m}$. Then*

(a) For each $\chi \in \mathrm{Irr}(G,\lambda)$ the row vector

$$v_\chi := (\chi_1, \chi_2, \ldots, \chi_m)$$

is a left eigenvector for M_i with eigenvalue $h_i\chi_i/\chi_1$ $(i = 1,\ldots,m)$;

(b) The eigenvectors v_χ $(\chi \in \mathrm{Irr}(G,\lambda))$ are linearly independent and, up to scalar multiples, are the only common eigenvectors of $M_1,\ldots,M_m$;

(c) $\mathrm{Irr}(G,\lambda)$ consists of exactly m characters, and Lemma 2.1 shows that the values of these characters are completely determined by the vectors v_χ.

Proof. Suppose that $\chi \in \mathrm{Irr}(G,\lambda)$ is afforded by the representation R. Then $R(c_i)$ is the scalar $(h_i\chi_i/\chi_1)R(1)$, so applying R to both sides of the equation (2.1) and using Lemma 2.1

we obtain

$$h_i\chi_i h_j\chi_j = \chi_1\left(\sum_{z\in T_k}\nu_{ijk,z}\lambda(z)\right)h_k\chi_k = \chi_1\sum_{k=1}^{m}\mu_{ikj}h_j\chi_k$$

which immediately gives (a).

Lemma 2.1 also shows that χ is completely determined by the values in the vector v_χ and shows how to write down all the values of χ once we know v_χ. Any linear dependence between the vectors v_χ ($\chi \in \mathrm{Irr}(G,\lambda)$) implies the corresponding linear dependence between the characters. Since the characters in $\mathrm{Irr}(G,\lambda)$ are linearly independent, the v_χ must also be linearly independent. In particular $|\mathrm{Irr}(G,\lambda)| \le m$ $(= m_{\ker\lambda})$.

As we noted at the beginning of this section, if $K := \ker\lambda$ then λ has exactly $\varphi(|Z:K|)$ Galois conjugates and these are the only characters in $\mathrm{Irr}(Z)$ which have K as their kernel. Thus with the notation of Lemma 2.2 the inequality $|\mathrm{Irr}(G,\lambda)| \le m_{\ker\lambda}$ shows that

$$|\mathrm{Irr}(G)| = \sum_{\lambda\in\mathrm{Irr}(Z)}|\mathrm{Irr}(G,\lambda)| \le \sum_{\lambda\in\mathrm{Irr}(Z)} m_{\ker\lambda} = \sum_{K\in\mathcal{K}}\varphi(|Z:K|)m_K. \tag{2.2}$$

By Lemma 2.2 the right hand side of (2.2) is equal to $k(G)$ $(= |\mathrm{Irr}(G)|)$ so the inequality in (2.2) is an equality. Thus we must have $|\mathrm{Irr}(G,\lambda)| = m_{\ker\lambda}$ for each λ. This proves (c).

Finally the m linearly independent vectors v_χ ($\chi \in \mathrm{Irr}(G,\lambda)$) form a basis of $\mathbb{C}^m$ and consist of common left eigenvectors for $M_1,\dots,M_m$. For any two different vectors, say v_χ and v_θ, there exists at least one i for which the eigenvalues $h_i\chi_i/\chi(1)$ and $h_i\theta_i/\theta(1)$ are different. Hence, up to scalar multiples, v_χ ($\chi \in \mathrm{Irr}(G,\lambda)$) are the only common eigenvectors for the matrices $M_1,\dots,M_m$. This proves (b) and completes the proof of the proposition. □

3. *Implementation*

(A) We implemented the process described in Proposition 2.3 using the methods of [**10**]. With the notation of Proposition 2.3 fix λ and put $m = m_{\ker\lambda}$. Let $V = \mathbb{C}^m$ be the vector space spanned by the vectors v_χ ($\chi \in \mathrm{Irr}(G,\lambda)$). Each subspace of V which is spanned by a subset of the v_χ is called a character subspace. By Proposition 2.3(a) each character subspace U is invariant under M_i for $i = 1,\dots,m$ and the eigenspaces of M_i in its action on U are also character subspaces. If $V = \bigoplus_{j=1}^{s} U_j$ is a direct sum of nonzero character subspaces and at least one U_j has dimension > 1, then we can find an index i such that this U_j decomposes properly into eigenspaces under M_i. In this way we can successively refine the direct decomposition of V until we obtain a sum of character spaces of dimension 1. The bases of these subspaces are then just scalar multiples of the v_χ. All these computations will be done, not over $\mathbb{C}$, but over a finite prime field as described in [**2**] and [**10**].

(B) The most expensive part of the process is the computation of the entries of M_i since this involves identifying the conjugacy classes of G to which products of pairs of elements of G belong. Minor modifications of the arguments used in [**10**] prove the following facts which enable us to reduce the number of these computations.

(1) The first column of M_i (corresponding to the class $\{1\}$) has only one nonzero entry, namely, the ith entry is equal to h_i. {Proof: $\mu_{ij1} = \sum_{z\in T_j}\nu_{i'j1,z^{-1}}\lambda(z)$ where we can assume $T_j \cap Z_j = \{1\}$. Now $\nu_{i'j1,z^{-1}}$ is the number of pairs $(x,y) \in C_i \times C_j$ such that $x^{-1}y = z^{-1}$. This number is h_i if $i = j$ and $z \in Z_i$ and is 0 otherwise. Thus $\mu_{ij1} = 0$ if $i \ne j$ and $\mu_{ii1} = \sum_{z\in Z_i\cap T_i} h_i\lambda(z) = h_i$.}

(2) Suppose that U is a nonzero character subspace with a basis $u_1, \ldots, u_s$ in echelon form (the leading 1 for u_1 occurs in the first place since U contains some v_χ). Then U splits into two or more eigenspaces under M_i if and only if at least one of the vectors $u_2M_i, \ldots, u_sM_i$ has its first entry nonzero. By property 1 this is equivalent to saying that U splits under M_i if and only if at least one of the vectors $u_2, \ldots, u_s$ has its ith entry nonzero.
(3) If U does split under M_i then we can compute basis vectors for the distinct eigenspaces of M_i in its action on U from a knowledge of the columns $k_1, \ldots, k_s$ of M_i where $k_1, \ldots, k_s$ are the positions of the leading 1 for the basis vectors $u_1, \ldots, u_s$.

Property 2 can be used to choose a value of i which maximizes the number of U_t which decompose under M_i and property 3 means that we only have to compute part of M_i in order to carry out the decompositions. However, as [**10**] shows, when U_t has dimension 2 we can usually avoid computations of the entries of M_i by using a combinatorial splitting of U_t. We used a simplified version of Schneider's method which is described in (E) below.

(C) This reduction of V to a direct sum of 1-dimensional character subspaces only gives scalar multiples of the vectors v_χ. To recover v_χ from a nonzero scalar multiple, say $y = \theta v_\chi$, we proceed as follows. Let $y = (\eta_1, \ldots, \eta_m)$. Since $C_1 = \{1\}$ by hypothesis, we know that $\eta_1 = \theta\chi_1 \neq 0$ and can assume that y has been normalized so that $\eta_1 = 1$, so $\theta = 1/\chi_1$. Now, using the fact that $|\chi(xz)| = |\chi_i\lambda(z)| = |\chi_i|$ for all $x \in C_i$ and $z \in Z$, and that $\chi_i = 0$ for all $i > m$, we obtain the value of θ from the calculation

$$\sum_{i=1}^{m} |Z : Z_i|\, h_i\eta_i\eta_{i'} = \theta^2 \sum_{i=1}^{m} |Z : Z_i|\, h_i\chi_i\chi_{i'} = \theta^2 \sum_{x\in G} \chi(x)\chi(x^{-1}) = \theta^2\, |G|$$

All these computations are done over a finite field $GF(p)$ (with p a suitable prime) in the way described in [**2**] and [**10**]. The fact that the coefficients μ_{ijk} in the present case are cyclotomic integers rather than ordinary integers does not change the computation significantly, assuming that we have chosen p such that the exponent e of G divides $p-1$ (this ensures that $GF(p)$ contains a primitive eth root of 1).

(D) Calculating the entries of M_i is significantly speeded up using the following lemma which refines an idea used in the current implementation of the BDS method in GAP which is based on the thesis of Hulpke [**5**]. A referee has pointed out that Hulpke states that he found that the use of double cosets did not seem to be advantageous except for larger classes (see [**5**, Sec. 2.5.4]), but that does not seem to be true in our situation. This may be a result of the increased memory now available.

Define $K_i := N_G(x_iZ)$ for $i = 1, \ldots, r$ (so $K_i/Z = C_{G/Z}(x_iZ)$). We have $|K_i : C_G(x_i)| = |Z_i|$ since x_i has $|Z_i|$ G-conjugates in x_iZ.

LEMMA 3.1. *Suppose that $i, k \leq m$ and let T be a transversal for the set of (K_i, K_k)-double cosets in G. For each $t \in T$ we have $t^{-1}x_itx_k \in C_{j_t}z_t$ for some j_t and some $z_t \in Z$ (z_t is only determined up to a factor from Z_{j_t}, but this will not matter since we shall only be interested in the value of $\lambda(z_t)$ which is uniquely determined when $j_t \leq m$). Then*
(i) $u^{-1}x_iux_k \in C_{j_t}Z_iZ_kz_t$ for all $u \in K_itK_k$;
(ii) for each $j \leq m$ we have

$$|C_G(x_i)|\mu_{ijk} = \sum\nolimits'_t |K_itK_k|\, \lambda(z_t)$$

where the sum is over all $t \in T$ with $j_t = j$.

Proof. (i) First note that if $a \in K_i$ then $a^{-1}x_ia = x_iz$ for some $z \in Z$ by the definition of K_i; it follows that z stabilizes C_i and so $z \in Z_i$. Write $u = atb$ with $a \in K_i$ and $b \in K_k$. Then

$$u^{-1}x_iux_k = b^{-1}(t^{-1}a^{-1}x_iat)(bx_kb^{-1})b \in b^{-1}(t^{-1}x_iZ_it)(x_kZ_k)b \subseteq C_{j_t}Z_iZ_kz_t.$$

(ii) We have

$$|C_G(x_i)|\,|C_G(x_k)|\,c_ic_k = \sum_{v\in G} v^{-1}\left(\sum_{u\in G} u^{-1}x_iux_k\right)v$$
$$= \sum_{t\in T}\sum_{v\in G} v^{-1}\left(\sum_{u\in K_itK_k} u^{-1}x_iux_k\right)v.$$

Now $\sum_{v\in G} v^{-1}\left(\sum_{u\in K_itK_k} u^{-1}x_iux_k\right)v$ lies in the centre of $\mathbb{C}[G]$ and (i) shows that it has the form $b_tc_{j_t}z_t$, say, where $b_t \in \mathbb{C}[Z_iZ_k]$ and where the sum of the coefficients of b_t is $|G|\,|K_itK_k|\,/h_{j_t}$. Since the class sums are linearly independent we conclude from equation (2.1) that

$$|C_G(x_i)|\,|C_G(x_k)| \sum_{z\in T_j} \nu_{ikj,z}\lambda(z) = \sum\nolimits'_t \lambda(b_tz_t)$$

where the right hand sum is over all $t \in T$ with $j_t = j$. Since $Z_iZ_k \le \ker\lambda$ by the definition of m, $\lambda(b_t) = |G|\,|K_itK_k|\,/h_j$ and so

$$|C_G(x_i)|\,|C_G(x_k)|\,(h_k/h_j)\mu_{ijk} = (|G|\,/h_j)\sum\nolimits'_t |K_itK_k|\lambda(z_t).$$

The result now follows. □

(E) [**10**] describes a fast combinatorial method of splitting 2-dimensional character spaces. We use a modified version. On the space of class functions over a field F whose characteristic does not divide $|G|$ we define the inner product $[\lambda,\mu] := (1/\,|G|)\sum_{x\in G}\lambda(x)\mu(x^{-1})$. Note that this is not quite the same as the usual inner product on the class functions over $\mathbb{C}$ since it is symmetric and bilinear, not skew symmetric and sesquilinear. Recall that in considering class functions we assume that the first class is $\{1\}$.

LEMMA 3.2. *Suppose that V is a 2-dimensional F-space with an orthogonal basis $\xi := \chi/\chi(1)$, $\eta := \theta/\theta(1)$ where χ and θ are irreducible characters and define $e := 1/\chi(1)^2$ and $f := 1/\theta(1)^2$. Let v_1, v_2 be a basis of V in echelon form (so v_1 is 1 on $\{1\}$ and v_2 is 0 on $\{1\}$). Define $c_{ij} := [v_i, v_j]$ for $i, j \in \{1, 2\}$ and $\Delta := c_{11}c_{22} - c_{12}^2$. Then*

$$\chi(1)^2 + \theta(1)^2 = c_{22}/\Delta.$$

Proof. Write v_1 and v_2 in terms of the basis ξ, η. Since ξ and η both take the value 1 on $\{1\}$ we see that $v_1 = \alpha\xi + (1-\alpha)\eta$ and $v_2 = \beta(\xi - \eta)$ for some $\alpha, \beta \in F$. Then $c_{11} = \alpha^2e + (1-\alpha)^2f$, $c_{12} = \alpha\beta e - (1-\alpha)\beta f$ and $c_{22} = \beta^2(e+f)$. A short calculation shows that $\Delta = \beta^2ef$ and so $c_{22}/\Delta = 1/e + 1/f$ as claimed. □

We use this lemma to determine candidate values for e and f (F is the prime field of size p). It is known that the degree d of an irreducible character of a group G divides $n := |G : Z(G)|$ and that $d^2 \le n$. The number of divisors of n is very small compared with the size of n once n is not too small and so the chance that there will be more than one pair $\{a, b\}$ of divisors of n such that $a^2 + b^2 = c_{22}/\Delta$ is also small. If there is a unique pair, then we take $e = 1/a^2$ and $f = 1/b^2$. In the case where there is more than one pair we revert to the longer splitting method via the M_i matrices (for each of the groups listed in Tables 1 and 2 fewer than 7% of the integers which can be written as the sum of two squares $a^2 + b^2$ with a, b dividing n and $a^2 \le b^2 \le n$ do not have a unique representation in this form).

Once e and f are determined, we can obtain β from $\beta^2 = c_{22}/(e+f)$ and then $\alpha = (c_{12}/\beta + f)/(e+f)$ (taking $-\beta$ in place of β corresponds to interchanging χ and θ). Now $\chi = v_1 + (1-\alpha)\beta^{-1}v_2$ and $\theta = v_1 - \alpha\beta^{-1}v_2$.

(F) As we noted above, once we have computed the characters in $\mathrm{Irr}(G,\lambda)$ for one character $\lambda \in \mathrm{Irr}(Z)$, the characters in the sets $\mathrm{Irr}(G,\mu)$ where μ is a Galois conjugate of λ are simply Galois conjugates of the characters in $\mathrm{Irr}(G,\lambda)$ and so require very little additional computation.

4. *Solvable normal subgroups*

The technique in the previous sections can be used to compute characters in more general situations. Suppose that A is a normal abelian subgroup of G. Then G acts on the set $\mathrm{Irr}(A)$ of irreducible (ordinary) characters λ of A via $\lambda^x(a) := \lambda(xax^{-1})$ $(x \in G,\ \ a \in A,\ \ \lambda, \lambda^x \in \mathrm{Irr}(A))$. For each G-orbit Λ in $\mathrm{Irr}(A)$ we define $\mathrm{Irr}(G,\Lambda) := \left\{\chi \in \mathrm{Irr}(G) \mid \chi_A \text{ is an integer multiple of } \sum_{\lambda\in\Lambda}\lambda\right\}$; the sets $\mathrm{Irr}(G,\Lambda)$ form a partition of $\mathrm{Irr}(G)$ (see [**6**, Theorem 6.2]). We choose $\lambda \in \Lambda$ and consider the stabilizer

$$H := \{x \in G \mid \lambda^x = \lambda\}$$

(the *inertial group* of λ). Then $|G:H| = |\Lambda|$ and $\{\lambda\}$ is an H-orbit in $\mathrm{Irr}(A)$. Theorem 6.11 of [**6**] shows that the induction mapping $\psi \mapsto \psi^G$ is a bijection of $\mathrm{Irr}(H,\lambda)$ onto $\mathrm{Irr}(G,\Lambda)$. On the other hand, $A/\ker\lambda$ is cyclic of the same order as λ, $\ker\lambda$ is a normal subgroup of H, and $\bar{A} := A/\ker\lambda$ is contained in the centre of $\bar{H} := H/\ker\lambda$. Since the characters $\mathrm{Irr}(H,\lambda)$ can be derived from the characters $\mathrm{Irr}(\bar{H},\bar{\lambda})$ via $\lambda(a) = \bar{\lambda}(a\ker\lambda)$, the characters in $\mathrm{Irr}(H,\lambda)$ can be computed as in the previous section and then induced to give the characters in $\mathrm{Irr}(G,\Lambda)$.

More generally if G has a solvable normal subgroup S, then a similar process allows us to compute various subsets of irreducible characters of G. Let $S = S_0 > S_1 > \cdots > S_l = 1$ be the derived series for S. For each $\chi \in \mathrm{Irr}(G)$ consider the least t such that $S_t \le \ker\chi$. If $t > 0$ then $\bar{A} := S_{t-1}/S_t$ is a normal abelian subgroup of $\bar{G} := G/S_t$ and χ is essentially a character of $\bar{G}$ whose restriction to $\bar{A}$ is nontrivial. If $\chi_{\bar{A}}$ has an irreducible constituent λ and Λ is the $\bar{G}$-orbit in $\mathrm{Irr}(\bar{A})$ containing λ, then $\chi \in \mathrm{Irr}(\bar{G},\Lambda)$ and the problem has been reduced to the case of a normal abelian subgroup.

We can define a partition of the set $\mathrm{Irr}(G)$ into parts which are indexed by $t \in \{0,1,\ldots,l\}$ where χ is in the tth part if t is the least integer such that $S_t \le \ker\chi$. If $t > 0$ then the tth part can be further partitioned into subsets of the form $\mathrm{Irr}(G,\Lambda)$ where Λ is a G-orbit of nontrivial characters in $\mathrm{Irr}(S_{t-1}/S_t)$. The characters in each of the parts of this refined partition can be computed independently of the characters in the other parts.

5. *Runtimes*

We have implemented the algorithm described in Section 2 in GAP. Tables 1 and 2 give the execution times for GAP to compute the character tables of groups G with centre Z. The computations recorded in these tables were carried out using a permutation representation of the group. Each row of the tables lists the structure of G followed by the orders of G and Z, the degree of the permutation representation and the number of classes of G. The column headed T gives the time (in seconds) taken by the new program to compute all irreducible characters of G whilst the last column gives the time T^* taken by the current GAP implementation of the BDS method to calculate these characters. The notation "-" in the T^* column means that after a long enough time the system failed to compute the character table. The times are the cpu times (processor times) in seconds for an Apple G5 with dual 2.8 GHz Quad-Core processors and 4 GB ram. Computer code of the current implementation in GAP of the BDS method is

TABLE 1. *Examples of runtimes*

G	$\lvert G\rvert$	$\lvert Z\rvert$	Degree	Classes	T	T^*
$\mathrm{GL}(2,7)$	2016	6	48	48	1	1
$\mathrm{GU}(2,7)$	2688	8	128	64	1	1
$\mathrm{GL}(2,11)$	13200	10	120	120	1	5
$\mathrm{GU}(2,11)$	15840	12	288	144	2	9
$\mathrm{GL}(2,17)$	78336	16	288	288	10	106
$\mathrm{GU}(2,23)$	291456	24	1152	576	46	10654
$12.M_{22}$	5322240	12	31680	109	562	-
$(((C_5 \times C_5 \times C_5 \times C_5) : A_5) : C_5) : C_2$	375000	5	25	506	10	105
$C_6 \times (((C_6 \times C_6 \times C_6 \times C_6) : A_5) : C_2)$	933120	6	30	918	42	193

listed in the file `ctblgrp.gi` in the GAP subdirectory `lib`. For an overview of the functions available in GAP for computing characters, see Chapter 71 of the GAP manual.

In Table 1 the last two groups are transitive permutation groups of degrees 25 and 30, respectively, taken from the library of GAP. In Table 2 the groups are taken from the library of perfect groups in GAP. The perfect groups in this library are parametrized by pairs $[size, index]$ where $index$ runs over $1 \dots num(size)$ and $num(size)$ is the number of perfect groups of that size. Our sample of perfect groups was obtained by choosing all perfect groups $[size, index]$ with $index = 1$ for which $|Z| > 5$. The notation in the first column of Table 2 is provided by GAP and describes the perfect group in accordance with [**7**]. In almost all cases for the groups considered, the new method is faster (sometimes much faster) than the method currently used. There are two anomalies which stand out in Table 2 (the groups of orders 552960 and 933120) where the old method is 3 to 5 times faster; we have not been able to explain why these two cases occur.

TABLE 2. *Examples of runtimes for perfect groups*

G	$\|G\|$	$\|Z\|$	Degree	Classes	T	T^*
$A_6\ 3^1 \times 2^1$	2160	6	98	31	1	1
$A_5(2^4E\ (2^1\ A \times 2^1))\ C\ 2^1$	7680	8	76	48	1	1
$A_7\ 3^1 \times 2^1$	15120	6	285	40	1	2
$A_5(2^4E\ (2^1 A \times 2^1\ A))\ C\ (2^1 \times 2^1)$	15360	16	128	84	1	5
$A_5\ 2^1 \times 3^{4'}\ E\ 3^1$	29160	6	42	87	3	3
$A_5\ 2^1 \times (2^4E\ (2^1 A \times 2^1\ A))\ C\ (2^1 \times 2^1)$	30720	32	152	164	3	15
$A_6\ 3^1 \times 2^4E\ 2^1$	34560	6	30	61	1	1
$A_6\ 2^1 \times (2^4E\ 2^1 A\ )\ C\ 2^1$	46080	8	144	68	2	5
$A_6\ 3^1 \times (2^4E\ 2^1 A\ )\ C\ 2^1$	69120	12	82	89	3	4
$A_5\ 2^1 \times 5^3E\ 5^1$	75000	10	49	149	5	9
$(A_5 \times A_6\ 3^1)\ 2^1$	129600	6	103	155	3	10
$A_6\ 3^1 \times 2^1 \times (2^4E\ 2^1 A)\ C\ 2^1$	138240	24	162	172	5	29
$A_7\ 3^1 \times 2^1 \times 2^4$	241920	6	301	73	8	17
$L_2(8)\ 2^6E\ (2^1 \times 2^1 \times 2^1)$	258048	8	336	94	51	146
$(A_5 \times A_6\ 3^1)\ 2^2$	259200	12	122	279	6	26
$A_5\#\ 2^8\ 5^2$	384000	8	125	144	37	66
$(A_5 \times A_5)\#\ 2^7$	460800	8	88	216	10	16
$L_2(8)\ N\ (2^6E\ (2^1 \times 2^1 \times 2^1\ A))\ C\ 2^1$	516096	16	400	166	95	176
$A_6\ 3^1 \times (2^4 \times 2^4)\ 2^1$	552960	6	46	154	56	18
$A_5\#\ 2^7\ 3^4$	622080	8	81	348	53	82
$(L_3(2) \times A_6\ 3^1)\ 2^2$	725760	12	114	341	18	84
$L_3(2)\ 2^1 \times 3^6\ C\ 3^1$	734832	6	2203	93	102	974
$A_5\#\ 2^9\ 5^2$	768000	16	177	256	40	163
$(A_6 \times A_6)\ 3^1\ 2^1$	777600	6	104	217	9	20
$(A_5 \times A_7\ 3^1)\ 2^1$	907200	6	290	200	13	30
$A_5\ 2^1 \times (2^{4'}\ C\ 2^1)\ 3^4\ C\ 3^1$	933120	6	267	107	27	6
$A_5\#\ 2^7\ 5^3$	960000	8	106	368	93	340
$A_7\ 3^1 \times 2^1 \times 2^6$	967680	6	511	136	29	51

Acknowledgement. We wish to thank two anonymous referees who made detailed comments and suggestions on earlier versions of this paper. These have resulted in significant improvements in both the program and the paper.

References

1. T. Breuer, Computing character tables of groups of type *M.G.A*, *LMS J. Comput. Math.* 14 (2011) 173–178.
2. J.D. Dixon, High speed computation of group characters, *Numer. Math.* 10 (1967) 446–450.
3. B. Fischer, Clifford-matrices *in* "Representation Theory of Finite Groups and Finite-dimensional Algebras (Bielefeld 1991)", Birkhäuser, Basel, 1991 (pp. 1–16).
4. The GAP Group, GAP—Groups, Algorithms, and Programming, Version 4.5.5 (2012). (`http://www.gap-system.org`).
5. J. A. Hulpke, "Zur Berechnung von Charaktertafeln", Diplomarbeit im Fach Mathematik an der Rheinisch-Westfälischen Technischen Hochschule, Aachen, 1993.
6. I.M. Isaacs, "Character Theory of Finite Groups", Academic Press, New York, 1976.
7. D.F. Holt and W. Plesken, *Perfect Groups*, Clarendon Press. Oxford, 1989.
8. K. Lux and H. Pahlings, "Representations of Groups: a computational approach", Cambridge Univ. Press, Cambridge, 2010.
9. H. Pahlings, The character table of $2_+^{1+22}.Co_2$, *J. Algebra* 315 (2007) 301–323.
10. G.J.A. Schneider, Dixon's character table algorithm revisited, *J. Symbolic Comput.* 9 (1990) 601–606.
11. W.R. Unger, Computing the character table of a finite group, *J. Symbolic Comput.* 41 (2006) 847–862.
12. W. Bosma, J. Cannon, and C. Playoust, The Magma algebra system I: the user language, *J. Symbolic Comput.*, 24 (1997) 235–265.

Vahid Dabbaghian
MoCSSy Program
The IRMACS Centre
Simon Fraser University
Burnaby, BC V5A 1S6
Canada

vdabbagh@sfu.ca

John D. Dixon
School of Mathematics and Statistics
Carleton Unversity
Ottawa, ON K1S 5B6
Canada

jdixon@math.carleton.ca