Computing Irreducible
Representations of Supersolvable
Groups over Small Finite Fields

A. Omrani and A. Shokrollahi

TR-96-005
January 1996

Abstract
We present an algorithm to compute a full set of irreducible representations of a
supersolvable group $G$ over a finite field $K$, $\text{char} K \nmid [G]$, which is not assumed to be
a splitting field of $G$. The main subroutines of our algorithm are a modification of
the algorithm of Baum and Clausen [1] to obtain information on algebraically conjuga-
tive representations, and an effective version of Speiser’s generalization of Hilbert’s
Theorem 90 stating that $H^1(\text{Gal}(L/K), \text{GL}(n, L))$ vanishes for all $n \geq 1$. 
1 Introduction and Main Results

Recently Baum and Clausen [1] published an efficient algorithm for computing the absolutely irreducible representations of a supersolvable group \( G \) given in pc-presentation. The matrix representations their algorithm computes are adapted to a chief series \( T := (G = G_n > G_{n-1} > \cdots > G_0 = \{1\}) \), i.e., any such representation \( D \) satisfies the following conditions: (1) the restriction \( D \upharpoonright G_j \) of \( D \) to \( G_j \) is equal to a direct sum of irreducible matrix representations of \( G_j \), and (2) equivalent irreducible constituents of \( D \upharpoonright G_j \) are equal. The algorithm traverses the chief series \( T \) bottom-up and constructs in each step \( j \) among other data a complete set of nonequivalent absolutely irreducible representations of \( G_j \). These representations are almost unique: if \( L \) is a field containing a primitive \( \ell \)th root of unity, \( e \) being the exponent of \( G \), and \( D \) and \( \Delta \) are two equivalent irreducible \( T \)-adapted representations of \( LG \) of degree \( \ell \), say, then the intertwining space

\[
\text{Int}(D, \Delta) := \{ X \in L^{d \times d} \mid \forall g \in G : \quad XD(g) = \Delta(g)X \}
\]

is generated over \( L \) by a monomial matrix (see [2, Theorem 7.4]).

Now let \( K \) be a finite field, \( G \) be a supersolvable group such that \( \text{char} K \nmid |G| \), \( T \) be a chief series of \( G \), and \( L \) be a finite extension of \( K \) which contains a primitive \( \ell \)th root of unity. The Galois group \( \text{Gal}(L/K) \) acts on the irreducible matrix representations of \( LG \) in a straightforward manner. In Section 2 we shall modify the algorithm of Baum and Clausen by collecting at each step information about the \( \text{Gal}(L/K) \)-orbits of the representations constructed. We then employ the information obtained at level \( n \) to compute realizations of direct sums of these representations over the field \( K \). By a realization of a matrix representation \( D \) of \( LG \) over \( K \) we mean a matrix \( T \in \text{GL}(d, L) \), \( d \) being the degree of \( D \), such that \( T^{-1}DT(g) \) has entries in \( K \) for all \( g \in G \). Not every representation has a realization over \( K \). Even more, if \( K \) is a prime field, \( \chi \) denotes the character of \( D \), and \( K(\chi(g)) \mid g \in G \) its character field, then \( D \) cannot have a realization over a proper subfield of \( K(\chi) \). The question whether an absolutely irreducible representation \( D \) of \( G \) has a realization over \( K(\chi) \) is hard to answer in general, i.e., for arbitrary \( K \) and arbitrary \( G \). (This amounts to the question whether the Schur-index of the character of \( D \) equals 1, see [3, Kapitel V, §14].) It is however well-known that for finite fields and arbitrary finite groups the question has an affirmative answer [3, Kapitel V, Satz 14.10].

In theory we thus know that any irreducible matrix representation \( D \) of \( LG \) has a realization over its character field. How can we compute such a realization? Let \( M \) be a subfield of \( L \) of index \( \ell \), and \( \beta \) be the Frobenius automorphism of \( L/M \). If \( D \) is an irreducible representation of \( LG \) of degree \( \ell \), then so is \( D^\beta \), where \( D^\beta(g) := D(g)^\beta \) for all \( g \in G \). If \( M \) is the character field of \( D \), then \( D \) is equivalent to \( D^\beta \), hence \( \text{Int}(D, D^\beta) \) is generated by an invertible matrix \( S \). A generalization of Hilbert's Theorem 90 due to Speiser [7] states that the first cohomology \( H^1((\beta), GL(d, L)) \) is trivial. (This is a modern interpretation of Speiser's result; see also [6, Chapter X, §1].) Hence, for \( S \in \text{GL}(d, L) \) there exists \( T \in \text{GL}(d, L) \) such that \( T^{-1}T^\beta = S \) if and only if the norm \( S^\beta S^{-1} \cdots S^\beta \) of \( S \) equals the \( n \times n \)-identity matrix \( I_n \). Such a matrix \( T \) will give the desired realization of \( D \) over its character field \( M \). In our applications, \( S \) is a monomial matrix and this allows to compute \( T \) from \( S \) efficiently, see Section 3.

Now we are almost done. Namely, we may suppose that \( D \) is an absolutely irreducible
representation of \( G \) with character \( \chi \) such that \( D(g) \) has entries in \( K(\chi) \). Let \( \sigma \) be the Frobenius automorphism of \( K(\chi)/K \). Then the trace of \( D \) over \( K \) defined as \( \text{Tr}_K (D) := D \oplus D^\sigma \oplus \cdots \oplus D^\sigma^{m-1} \), \( m := [K(\chi):K] \), has character field equal to \( K \) and a realization of \( \text{Tr}_K (D) \) over \( K \) can be computed easily, see Section 3. Furthermore, \( \text{Tr}_K (D) \) is an irreducible \( KG \)-representation (since any of its irreducible constituents over \( K \) has to be invariant under \( \sigma \)); conversely, any irreducible \( KG \)-representation is the trace over \( K \) of some irreducible representation of \( LG \). (For these and related facts see [4, Chapter VII, §1].) To obtain the irreducible representations of \( KG \) we first compute a set \( \mathcal{F}' \) of representatives of \( \text{Gal}(L/K) \)-orbits of irreducible representations of \( LG \), and for each such representation a realization of its trace over \( K \). Starting from a pc-presentation of \( G \) and the chief series \( T \) induced by that, the first two steps of our algorithm are as follows:

**Step 1.** We first modify the algorithm of Baum and Clausen to compute a full set \( \mathcal{F} \) of pairwise nonequivalent irreducible monomial and \( T \)-adapted representations of \( LG \), where \( L \) is a field extension of \( K \) containing a primitive \( \epsilon \)th root of unity, and a permutation \( \gamma \) of \( \mathcal{F} \) such that \( F^\alpha \) is equivalent to \( \gamma F \): \( F^\alpha \sim \gamma F \). Here \( \alpha \) is the Frobenius automorphism of \( L/K \). We then compute a full set \( \mathcal{F}' \) of representatives of \( \text{Gal}(L/K) \)-orbits of \( \mathcal{F} \) and for each \( F \in \mathcal{F} \) the degree of the character field of \( F \) over \( K \).

**Step 2.** For each \( F \in \mathcal{F}' \) we compute a realization \( T_F \) of \( F \) over its character field and then a realization of the trace of \( T_F^{-1} FT_F \) over \( K \).

Similar to the algorithm of Baum and Clausen, the arithmetics we use in these two steps consists just of symbolic computation in \( L^x \), where \( L \) is a field extension of \( K \) containing an \( \epsilon \)th root of unity. More precisely, we represent nonzero elements of \( L \) as integers \( i \) with \( 0 \leq i < |L| - 1 \), where \( i \) corresponds to the element \( \omega^i \) and \( \omega \) is a fixed generator of \( L^x \). This representation of \( L \) allows to solve efficiently equations of the form \( N(x) = \alpha \) or \( x(x^\sigma)^{-1} = \alpha \), where \( \alpha \in L, N \) is the norm of \( L \) relative to a subfield \( M \), and \( \sigma \) is the Frobenius automorphism of \( L/M \). We shall need solutions to these kinds of equations in the second step of our algorithm. Moreover, as we will need primitive elements for subfields of \( L \), this representation of \( L \) allows us to compute in advance these generators and store them in a list \( \Omega \).

The final step of the algorithm computes the \( KG \)-representations from the already computed realizations. This step requires matrix multiplication over \( L \), and symbolic computation in \( L^x \) does not suffice for this purpose. Strategies to solve this problem are discussed in the last section.

Many thanks go to an anonymous referee for important comments and to Michael Clausen for communicating to us the problem discussed in this paper and for many stimulating discussions.

## 2 Irreducibles of \( LG \) and \( \text{Gal}(L/K) \)-Orbits

The first step of our algorithm takes as input a supersolvable group \( G \) in pc-presentation and a finite extension \( L \) of \( K \) containing a primitive \( \epsilon \)th root of unity; it outputs a list \( \mathcal{F} \) of pairwise inequivalent irreducible representations of \( LG \) and a permutation \( \gamma \) of \( \mathcal{F} \) such that \( F^\alpha \sim \gamma F \), \( \alpha \) being the Frobenius automorphism of \( L/K \).

For the rest of this section we set \( T_i := (G_i > G_{i-1} > \cdots > G_0 = \{1\}) \) for \( 1 \leq i \leq n \). In particular, \( T = T_n \).
The algorithm of Baum and Clausen in [1] computes the list \( \mathcal{F} \); we modify this algorithm to obtain additional information on the orbits of \( \mathcal{F} \) under the action of the Galois group of \( L/K \); this information is encoded as the permutation \( \gamma \).

Our algorithm works bottom up along \( T \). At level \( i \), \( 1 \leq i \leq n \), it takes the following input:

1. \( \mathcal{F} \), a full set of nonequivalent irreducible \( e \)-monomial representations of \( LG_{i-1} \) such that \( \oplus_{F \in \mathcal{F}} F \) is \( T_{i-1} \)-adapted;

2. For every \( i < j \leq n \) a permutation \( \pi_j \) of \( \mathcal{F} \) such that \( F_{\pi_j} \sim \pi_j F \) for all \( F \in \mathcal{F} \) as well as \( e \)-monomial matrices \( X_{j,F} \in \text{Int}(F_{\pi_j}, \pi_j F), \ F \in \mathcal{F} \);

3. A permutation \( \gamma \) of \( \mathcal{F} \) such that \( F^\gamma \sim \gamma F \), as well as \( e \)-monomial matrices \( M_F \in \text{Int}(F^\alpha, \gamma F), F \in \mathcal{F} \);

and computes the following output:

1. \( \mathcal{D} \), a full set of nonequivalent irreducible \( e \)-monomial representations of \( LG_i \) such that \( \oplus_{D \in \mathcal{D}} D \) is \( T_i \)-adapted;

2. For every \( i < j \leq n \) a permutation \( \tau_j \) of \( \mathcal{D} \) such that \( D_{\tau_j} \sim \tau_j D \) for all \( D \in \mathcal{D} \) as well as \( e \)-monomial matrices \( Y_{j,D} \in \text{Int}(D_{\tau_j}, \tau_j D), D \in \mathcal{D} \).

3. A permutation \( \delta \) of \( \mathcal{D} \) such that \( D^\alpha \sim \delta D \), as well as \( e \)-monomial matrices \( N_D \in \text{Int}(D^\alpha, \delta D), D \in \mathcal{D} \);

Outputs (1) and (2) are computed in exactly the same way as in the algorithm of Baum and Clausen [1]. Therefore, we only discuss the computation of Output (3) and assume that we have already performed the two phases of the algorithm in [1]. Note that during the construction at level \( i \) in Phase 1 there is built a bipartite graph in which \( F \in \mathcal{F} \) and \( D \in \mathcal{D} \) are linked iff \( F \) is a constituent of \( D \upharpoonright G_{i-1} \). We will need this information to compute \( \delta \) and \( N_D \). For this we proceed in a similar way as does Phase 2 of the Baum-Clausen algorithm. Let \( F \in \mathcal{F} \) and \( p := [G_i:G_{i-1}] \). We distinguish two cases.

Case 1. \( \pi_i F = F \), i.e., \( F^{\pi_i} \sim F \). Since \( (F^{\pi_i})^\alpha = (F^\alpha)^{\pi_i} \), we obtain

\[
(\gamma F)^{\beta_i} \sim (F^\alpha)^{\beta_i} \sim (F^{\pi_i})^\alpha \sim F^\alpha \sim \gamma F.
\]

We already know \( p \) extensions \( D_0, \ldots, D_{p-1} \) of \( F \) and \( p \) extensions \( \Delta_0, \ldots, \Delta_{p-1} \) of \( \gamma F \). For \( 0 \leq k < p \) we have

\[
D_k^\alpha \downarrow G_{i-1} = (D_k \downarrow G_{i-1})^\alpha = F^\alpha \sim \gamma F,
\]

hence \( D_k^\alpha \) is equivalent to one of the representations \( \Delta_0, \ldots, \Delta_{p-1} \). Thus there exists a permutation \( \rho \) of \( \{0, \ldots, p-1\} \) such that \( D_k^\alpha \sim \Delta_{\rho k} \) for \( 0 \leq k < p \). Since \( \text{Int}(D_k^\alpha, \Delta_{\rho k}) = \text{Int}(F^\alpha, \gamma F) \), we may set \( N_{D_k} := M_F \). To determine \( \delta D_k \), note that

\[
M_F D_k^\alpha(g_i) M_F^{-1} = \Delta_{\rho}(g_i) = \chi^\ell(g_i G_{i-1}) \Delta_0(g_i)
\]

for a unique integer \( \ell \) with \( 0 \leq \ell < p \). To compute \( \ell \), we just need to compare a nonzero entry of both sides of the above \( e \)-monomial matrix equation. We then set \( \delta D_0 := \Delta_\ell \). For other values of \( k \) we can determine \( \delta D_k \) by cyclic shifts: \( D_k^\alpha = (\chi^k \otimes D_0)^\alpha = (\chi^k)^\alpha \otimes D_0^\alpha \sim
\]

3
\((x^a)^k \otimes (x^\ell \otimes \Delta_0)\). Hence \(\delta D_k = \Delta_{(k+\ell) \mod p}\), since \(\alpha\) is the Frobenius automorphism over 
\(\bar{K} = \mathbb{F}_q\).

Case 2. \(\pi; F \neq F\), i.e., \(F^{\gamma^\pi} \neq F\). In Phase 1 we have already constructed \(D \in \mathcal{D}\) such that 
\(D \downarrow G_{i-1} = \bigoplus_{k=0}^{p-1} F_k\) and \(F_k = \pi^k F\) is of degree, say, \(f\). Since \((F \uparrow G)^\alpha = F^\alpha \uparrow G\) and 
\(F^\alpha \sim \gamma F\), \(\delta D\) is the unique representation \(\Delta \in \mathcal{D}\) such that \(\gamma F\) is an irreducible constituent of 
\(D \downarrow G_{i-1}\). According to our construction, \(\Delta \downarrow G_{i-1} = \bigoplus_{k=0}^{p-1} \Phi_k\) with \(\Phi_k = \pi^k \Phi\) for some 
\(\Phi \in \mathcal{F}\). There is a permutation \(\rho\) of \(\{0, \ldots, p-1\}\) such that \(\gamma F_k = \Phi_{\rho k}\) as well as 
e-monomial matrices 
\(M_k := M_{F_k} \in \text{Int}(F^\alpha, \Phi_{\rho k})\). To compute \(N_D \in \text{Int}(D^\alpha, \delta D)\), we consider 
\(\text{Int}(D^{\alpha} \downarrow G_{i-1}, \delta D \downarrow G_{i-1})\). By Schur’s Lemma there exist constants \(d_0, \ldots, d_{p-1} \in \mathbb{L}^x\) such that
\[N_D = \left( \frac{P_{\rho} \otimes I_f}{P_{\rho} \otimes I_f} \cdot \left( \bigoplus_{k=0}^{p-1} d_k M_k \right) \right).\]

We may assume that \(d_0 = 1\). To determine the other \(d_k\), we use the equation
\[N_D D^\alpha(g_i) N_D^{-1} = (\delta D)(g_i). \tag{1}\]

According to our construction in Phase 1 there are \(e\)-monomial matrices \(T_k, S_k \in \mathbb{L} \times \mathbb{L}\) such that
\[D(g_i) = \left( \frac{P_{\pi} \otimes I_f}{P_{\pi} \otimes I_f} \cdot \left( \bigoplus_{k=0}^{p-1} T_k \right) \right)\]
and
\[\left( \frac{\delta D}(g_i) = \left( \frac{P_{\rho} \otimes I_f}{P_{\rho} \otimes I_f} \cdot \left( \bigoplus_{k=0}^{p-1} S_k \right) \right),\right.\]
where \(\pi = (0, \ldots, p-1)\). Hence, (1) is equivalent to
\[\left( \frac{P_{\pi} \otimes I_f}{P_{\pi} \otimes I_f} \cdot \left( \bigoplus_{k=0}^{p-1} d_{\pi k} M_{\pi k} \right) \cdot \left( \bigoplus_{k=0}^{p-1} T_k^\alpha \right) \cdot \left( \bigoplus_{k=0}^{p-1} d_k^{-1} M_k^{-1} \right) \right) = \left( \frac{P_{\rho-1, \pi k} \otimes I_f}{P_{\rho-1, \pi k} \otimes I_f} \cdot \left( \bigoplus_{k=0}^{p-1} S_{\rho k} \right) \right).\]

Since \(d_0 = 1\), we can successively determine \(d_1, \ldots, d_{p-1}\) by comparing for each \(k\) one 
nonzero entry of 
\(M_{\pi k} T_k^{-1} M_k^{-1}\) and \(S_{\rho k}\).

We now compute a set \(\mathcal{F}'\) of representatives of \(\text{Gal}(L/K)\)-orbits of \(\mathcal{F}\) and for each 
\(F \in \mathcal{F}'\) with character \(\chi_F\) the degree of the character field \(d_F := [K(\chi_F) : K]\) of \(F\) as well as 
a matrix \(S_F \in \text{Int}(F^\alpha, F)\). (\(d_F\) generates the Galois group of \(L/K(\chi_F)\).) Notice 
that \(\ell := d_F\) is the smallest integer \(m\) such that \(F^\alpha m \sim F\), i.e., the smallest \(m\) such that 
\(\gamma^m F = F\). Furthermore, it is easily checked that
\[S_F := M_{\gamma^m - 1 F} \cdot M_{\gamma^{m-2} F} \cdot \cdots \cdot M_{\gamma F} \in \text{Int}(F^\alpha, F)\].

The algorithm to compute the required data is now straightforward. We take the first 
representation \(F\) in \(\mathcal{F}\), append it to the list \(\mathcal{F}'\), and set \(M := M_{F'}\). Then we go through all 
\(\gamma^i F\), delete them from the list \(\mathcal{F}\), update \(M := M_{\gamma^i F} M^\alpha\), and stop as soon as \(\gamma^i F\) 
equals \(F\), deleting \(F\) from \(\mathcal{F}\) in this last step. In this way we also obtain \(d_F\). We repeat the whole 
process until the list \(\mathcal{F}\) is empty.
3 Realization over Subfields

In this step of our algorithm we take the output of the last step and compute at first for each \( F \in F' \) a realization \( T_F \) of \( F \) over \( K(\chi_F) \), where \( \chi_F \) is the character of \( F \). We then proceed by computing a realization of the trace of \( T_F T_F^{-1} \) over \( K \).

It is well known that any absolutely irreducible representation of \( LG \) has a realization over its character field [3, Kapitel V, Satz 14.10]. We would like to give here a proof of this fact which builds the basis of our algorithm to find such a realization. We use the following setup: \( F \) is an irreducible representation of \( LG \) of degree \( f \), \( M \) is the character field of \( F \), \([L:M] =: \ell \), and \( \beta \) is a generator of \( \text{Gal}(L/M) \). For a matrix \( A \in L^{m \times m} \), we define the norm of \( A \) by \( N_{L/M}(A) := A^\beta \cdots A \). Note that if \( m \neq 1 \), then the norm of \( A \) does not necessarily belong to \( M^{m \times m} \).

The representations \( F \) and \( F^\beta \) are equivalent since they have the same character. Hence there exists an invertible matrix \( S \in \text{Int}(F^\beta, F) \). Suppose that there exists \( T \in \text{GL}(f, L) \) such that \( T^{-1} T^\beta = S \). Then, \( SFS^{-1} = F^\beta \) implies that \( TFT^{-1} \) is invariant under \( \beta \), hence \( T \) is a realization of \( F \) over \( M \). By Speiser’s Theorem [7] mentioned in the introduction such a matrix \( T \) exists if \( N_{L/M}(S) = I_f \). (Speiser’s original proof works only over infinite fields; for a general proof, see [6, page 15]). A straightforward calculation shows that \( N_{L/M}(S) \in \text{Int}(F, F) \). Hence, Schur’s Lemma implies that \( N_{L/M}(S) = c I_f \) for some \( c \in L \). But \( N_{L/M}(S)^\beta = SN_{L/M}(S)^{-1} = c I_f \), hence \( c \in M \). Since \( L \) is finite, any element in \( M \) is the norm of an element in \( L \), hence there exists \( d \in L \) such that \( N_{L/M}(dS) = I_f \). Replacing \( S \) by \( dS \) if necessary, we obtain the existence of \( T \), a realization of \( F \) over its character field. (See also [3, Kapitel V, Bemerung 14.14].)

From the second step we know \( \ell := [L:K]/d_F \) and an \( \epsilon \)-monomial matrix \( S = S_F \in \text{Int}(F^\beta, F) \), \( \beta = a^d_F \). Let \( S := P_e \text{diag}(S(1), \ldots, S(f)) \). We first compute some auxiliary data. Suppose that \( \pi \) can be written as the product of \( \nu \) disjoint cycles of lengths \( \ell_1, \ldots, \ell_\nu \) and let \( \rho_1, \ldots, \rho_\nu \) be a complete set of disjoint representatives of each cycle. We compute \( \nu, \ell_1, \ldots, \ell_\nu \) and \( \rho_1, \ldots, \rho_\nu \), then a nonzero entry \( \gamma := \prod_{j=1}^{\ell_\nu} S(\pi^j) \beta^{(j(\ell_j-1)j)} \) of \( N_{L/M}(S) \), and some element \( c \) of \( L \) satisfying \( N_{L/M}(c) = \gamma^{-1} \); we then replace \( S \) by \( cS \). Now we have \( N_{L/M}(S) = I_f \). As \( \ell_i \) divides the order of \( \pi \) and the latter divides \( \ell \), we have \( \ell_i | \ell \). Hence, we can extract from the precomputed list \( \Omega \) of primitive elements of subfields of \( L \) elements \( y_1, \ldots, y_\nu \in L \) such that \( y_i \) has degree \( \ell_i \) over \( K \). The rest of the algorithm, written in pseudo code, is now as follows (0\(^j\) means 0, \ldots, 0 \( j \)-times):

```plaintext
for \( i = 1 \) to \( \nu \) do  
    \gamma_i := \prod_{j=1}^{\ell_i} S(\pi^j \rho_i^j) \beta^{(j(\ell_j-1)j)};  
    Compute \( x_i \in L \) such that \( \gamma_i = x_i^{-1} x_i^\beta \);  
    \( T[0] := (0^\ell, x_1, x_2 y_1, \ldots, x_\nu y_\nu^{\ell-1}, 0^{m-\ell})^T; \)
end for  
for \( j = 1 \) to \( \ell - 1 \) do  
    \( T[\pi^j \rho_i] := S(\pi^{j-1} \rho_i)^{-1} T[\pi^{j-1} \rho_i] \beta^j; \)
end for  
for \( j = 1 \) to \( \nu \) do  
    \( t := t + \ell_i; \)
end for  
end for  
for \( j = 1 \) to \( \ell \) do  
    \( T_F := (T[1] | T[2] | \cdots | T[f]). \)
end for
```
It is not clear in advance that the above algorithm is executable since there might be no element $x_i$ satisfying the equation in line 4. In the following we show that such an $x_i$ always exists and prove that the matrix $T$ obtained by our algorithm is in fact a realization of $F$ over $M$.

Let $T \in L^I \times I$ have columns $T[1], \ldots, T[I]$. Then $TS = T^\beta$ iff for all $1 \leq i \leq \nu$ and all $1 \leq j \leq \ell_i$ we have

$$T[\pi^j \rho_i] = S(\pi^{i-1} \rho_i)^{-1} T[\pi^{j-1} \rho_i]^\beta.$$  

This implies that $T[\rho_i] = \gamma_i^{-1} T[\pi^{\ell_i} \rho_i]^{\beta^{\ell_i}}$, hence $T[\rho_i] = N_{L/M_i}(\gamma_i)^{-1} T[\rho_i]^{\beta^{\ell_i}}$ which gives $N_{L/M_i}(\gamma_i) = 1$, where $M_i$ is the fixed field of $\beta^{\ell_i}$. Hence, by Hilbert's Theorem 90 there exists $x_i$ satisfying the condition in line 4 and our algorithm is executable. Line 7 guarantees that (2) is satisfied for all $1 \leq i \leq \nu$, $1 \leq j < \ell_i$. To see that it is also satisfied for $j = \ell_i$, we only need to check that $T[\rho_i] = \gamma_i^{-1} T[\rho_i]^{\beta^{\ell_i}}$. But this follows from the choice of $x_i$ and the fact that $y_i$ is fixed under $\beta^{\ell_i}$. It remains to show that $T$ is invertible. This is true because the Vandermonde matrix $\left((y_i^{\beta^{\ell_i}})^k\right)_{0 \leq j, k \leq \ell_i - 1}$ is invertible (since $y_i$ has degree $\ell_i$ over $K$).

At this stage of our algorithm we have a list $F'$ of representatives of Gal($L/K$)-orbits of the irreducible representations of $LG$, for each $F \in F'$ the degree $d_F$ of the character field of $F$ over $K$, and a realization $T_F$ of $F$ over its character field. We know that $D_F := \bigoplus_{i=0}^{d_F - 1} F^{i'}$ is equivalent to an irreducible representation of $KG$ and that all irreducible representations of $KG$ are obtained this way.

Let $F \in F'$ be of degree $f$ and $\bar{F} := T_F F T_F^{-1}$. We extract from $\Omega$ a primitive element $\gamma$ of the character field of $F$ over $K$, i.e., an element having degree $d = d_F$ over $K$. Let $U := V \otimes I_f$, where $V$ is the Vandermonde matrix $V := \left((\gamma^i)^{a_j}\right)_{0 \leq i, j < d}$. It is easily verified that

$$R := U \cdot \begin{pmatrix} T_F & T_F^\alpha & \cdots & T_F^{\alpha^{d-1}} \end{pmatrix}$$

is a realization of $D = \bigoplus_{i=0}^{d-1} F^{i'}$ over $K$.

### 4 The Final Step and Concluding Remarks

Given a finite field $K$, a supersolvable group $G$ of exponent $e$ in pc-presentation, and a field extension $L$ of $K$, containing a primitive $e$th root of unity, the first two steps of our algorithm have computed a set $F'$ of representatives of Gal($L/K$)-orbits of the irreducibles of $LG$, and for each such representation a realization of its trace over $K$. One possible strategy to compute the $KG$ representations out of these data would be to represent $L$ as the residue class ring modulo an irreducible polynomial, compute a primitive element $\omega$ of $L^\times$, replace each entry of the matrices involved by their corresponding polynomial representations, and proceed with matrix multiplication (and inversion) over $L$. Another strategy is to start with a representation of $L$ as a polynomial residue class ring, and to go through all the steps of the algorithm using field arithmetic in $L$. Here we face the difficulty of solving equations of the type $x^d = \alpha$, where $d$ is a divisor of $|L| - 1$. Both these strategies consume exponential
time, and it seems that in practice a correct implementation of any of these strategies is rather complicated.

Nevertheless, we have implemented our algorithm in the computer algebra system GAP [5]. In this implementation the final step is performed by using a table of Jacobi logarithms for $L$, which needs exponential space (and time). Although it is impractical for large $|L|$, this strategy performs well for small sizes of $L$.

References


