Computing the Fitting Subgroup and Solvable Radical of Small-Base Permutation Groups in Nearly Linear Time

EUGENE M. LUKS AND ÁKOS SERESS

Abstract. We present Monte Carlo algorithms for computing the centralizer of a normal subgroup, the maximal normal $p$-group for primes $p$, the Fitting subgroup, and the maximal solvable normal subgroup in a permutation group $G = \langle S \rangle$ acting on a set $\Omega$. The running time of the algorithms can be expressed in the form $O(n|S|\log^e |G|)$, where $n$ is the size of $\Omega$. In particular, in the case of small-base groups, when $\log |G|$ is polylogarithmic as a function of $n$, the algorithms run in nearly linear, $O(n \log^e n)$ time. If the order of $G$ is known in advance, the algorithms can be upgraded to Las Vegas. The algorithms have been implemented in GAP.

1. Introduction

A remarkable recent development in the field of permutation-group computation has been the construction of a library of algorithms that run in time $O(n|S|\log^e |G|)$, given $G = \langle S \rangle \leq S_n$ [1, 2, 4, 5, 13, 15, 17, 18, 19]. We consider such algorithms to be nearly linear, the terminology justified by the following observation: if one is dealing with a family of groups $G$ for which $\log |G| = O(\log^e n)$, the running time is $O(n|S|\log^e n)$, which, up to only logarithmic factors, is a linear function of the input length $n|S|$. Indeed, such families of groups are commonly encountered and are referred to as small-base groups. Important classes of groups in the small-base category include all permutation representations of finite simple groups except the alternating groups.

In addition to their guarantee of a fast asymptotic running time, nearly linear-time algorithms appear to be well suited for implementation; in fact, many of them are already implemented in the GAP system [16]. Hence, they achieve the long sought goal of combining the theoretical and practical sides of permutation group manipulation.

The present paper describes some additions to the nearly linear-time library. For a group $G$ and a prime number $p$, we denote by $O_p(G)$ the largest normal $p$-subgroup of $G$, by $O_{\infty}(G)$ the largest solvable normal subgroup (solvable radical)

1991 Mathematics Subject Classification. 20B40, 68Q40.

The first author acknowledges support by NSF Grant CCR-9013410. The second author was supported in part by NSF Grants CCR-9201303 and CCR-9503430.

This is the final version of this paper.

© 1997 American Mathematical Society
of $G$, and by $F(G)$ the largest nilpotent normal subgroup (Fitting subgroup) of $G$. Our main result is

**Theorem 1.1.** Given generators $S$ for a permutation group $G \leq S_n$ and a prime $p$, the subgroups $O_p(G)$, $O_\infty(G)$, $F(G)$ can be computed by a Monte Carlo algorithm in time $O(n|S| \log^c |G|)$.

A randomized algorithm is called *Monte Carlo* if there is only a small probability (prescribed by the user) that it returns a wrong answer. Doubling the running time squares the error probability. In our case, an error can occur only at the initial strong-generating-set computation (cf. Subsection 2.2). In particular, if $|G|$ is known in advance, in which case the correctness of the strong-generating-set construction can be verified, then the algorithm indicated in Theorem 1.1 never returns an incorrect answer. Hence, the algorithm is upgraded to *Las Vegas*: its output is always correct and it runs in nearly linear expected running time.

Polynomial-time algorithms for constructing $O_p(G)$ were first given by Kantor [9] and Neumann [14]. An alternative approach, and a construction of $O_\infty(G)$, was described by Luks in [11] (see also [12]). A critical point in the algorithm presented here is based on the method of [11]. In particular, we employ a recursive approach having constructed an action $\varphi : G \to S_n$, with $\ker(\varphi)$ an abelian $p$-group, the nature of the kernel guaranteeing that $\varphi(O_p(G)) = O_p(\varphi(G))$. The construction of $\varphi$ can be carried out once we have any normal $p$-subgroup. (A similar procedure for $O_\infty(G)$ makes use of any normal solvable subgroup.) This useful normal subgroup is found as the normal closure of $C_N(M)$ (centralizer of $M$ in $N$) for some successive terms $M \lhd N$ in a composition series of $G$ and so we make use of the nearly linear time construction of composition series by Beals and Seress [4].

Observe that this approach also requires the efficient construction of centralizers of normal subgroups, which should be of independent interest.

**Theorem 1.2.** Given generators $S$ for the permutation groups $N, G$ with $N \vartriangleleft G$, the centralizer $C_G(N)$ can be computed by a Monte Carlo algorithm in time $O(n|S| \log^c |G|)$.

The nearly linear constraint seems to prohibit standard methods for $C_G(N)$, e.g., in [10] this subgroup is realised as the kernel of an induced action on a set of size $O(n^2)$. Instead, we compute the kernel $\tilde{G}$ of an action of $G$ on a set of size $\leq n$ and then the kernel of an action of $\tilde{G}$ on a set of size $\leq n$. The second kernel is $C_G(N)$.

Again, if $|G|$ is known in advance, then this algorithm can be upgraded to Las Vegas.

Implementations of the algorithms underlying Theorems 1.1 and 1.2 are available in the GAP [16] library.

2. Preliminaries

2.1. Group theory. For basic definitions and facts about groups and permutation groups, we refer the reader to [8] and [22]. We summarize the notation and terminology that we use.

For $H, G \leq \text{Sym}(\Omega)$, $\langle H^G \rangle$ denotes the $G$-closure of $H$ and $C_G(H)$ is the centralizer of $H$ in $G$. For $H \leq G$, $G/H$ denotes the set of right cosets of $H$ in $G$; a transversal for $H$ in $G$ is a complete set of right coset representatives.
The full symmetric group acting on the domain $\Omega$ is denoted by $\text{Sym}(\Omega)$, while $S_n$ denotes the symmetric group of degree $n$. We say that $G$ acts on a set $\Omega$ if a homomorphism $\varphi : G \to \text{Sym}(\Omega)$ is given, usually $\varphi$ is clear in context; the action is faithful if $\ker(\varphi) = 1$. If $G$ acts on $\Omega$ and $\omega \in \Omega$, we denote by $\omega^G$ the orbit of $\omega$ under $G$, namely $\{\omega^g \mid g \in G\}$; we say $G$ acts transitively on $\Omega$ if $\Omega$ consists of one orbit. If $G$ acts transitively on $\Omega$, a nonempty subset $\Delta \subseteq \Omega$ is called a block of imprimitivity for $G$ if, for all $g \in G$, $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$. If $B$ is a sequence of points from $\Omega$, $G_B$ denotes the pointwise stabilizer of $B$ in $G$. A group $G \leq \text{Sym}(\Omega)$ is called semiregular if $G_\omega = 1$ for all $\omega \in \Omega$; if $G$ is semiregular and transitive, it is called regular. For $G \leq \text{Sym}(\Omega)$, we denote by $\text{fix}(G)$ the set of fixed points of $G$, namely, $\{\omega \in \Omega \mid \forall g \in G : \omega^g = \omega\}$. If $\Delta \subseteq \Omega$ is the union of orbits of some $G \leq \text{Sym}(\Omega)$, then we denote by $G|\Delta \leq \text{Sym}(\Delta)$ the restriction of $G$ to $\Delta$. The wreath product of $G \leq \text{Sym}(\Omega)$ and $H \leq \text{Sym}(\Delta)$ is denoted by $G \wr H$; we consider its action on the disjoint union of $|\Delta|$ copies of $\Omega$ (see, e.g., [8, p. 81] for details).

2.2. Algorithmic concepts. It is assumed that permutation groups are input and output via a list of generators.

In his pioneering work [20, 21], Sims introduced the notions of a base and strong generating set as the fundamental data structures for computing with permutation groups. A base for a permutation group $G \leq \text{Sym}(\Omega)$ of degree $n$ is a sequence $B = (\beta_1, \beta_2, \ldots, \beta_M)$ of points from $\Omega$ such that $G_B = 1$. The point stabilizer chain of $G$ relative to $B$ is the chain of subgroups

$$G = G^{(1)} \geq G^{(2)} \geq \cdots \geq G^{(M+1)} = 1,$$

where $G^{(i)} = G_{(\beta_1,\ldots,\beta_{i-1})}$. The base $B$ is called nonredundant if there is strict inclusion $G^{(i)} > G^{(i+1)}$ for all $1 \leq i \leq M$. A strong generating set (SGS) for $G$ relative to $B$ is a set $S$ of generators of $G$ with the property that

$$(S \cap G^{(i)}) = G^{(i)}, \text{ for } 1 \leq i \leq M + 1.$$

With reference to some constant $c$, an (infinite) family $G$ of permutation groups is called a family of small-base groups if all $G \in G$ of degree $n$ admit bases of size $O(\log^c n)$. Equivalently, $\log |G| = O(\log^c n)$ for a constant $c'$ and each $G \in G$ of degree $n$.

Let $B = (\beta_1, \ldots, \beta_M)$ be a base of the group $G$ and let $G = G^{(1)} \geq G^{(2)} \geq \cdots \geq G^{(M+1)} = 1$ be the corresponding point stabilizer chain. Moreover, let $R_i$ denote a transversal for $G^{(i+1)}$ in $G^{(i)}$, $1 \leq i \leq M$. Each $g \in G$ can be written uniquely in the form $g = r_Mr_{M-1}\cdots r_2r_1$, $r_i \in R_i$. The process of factoring $g$ in this form is called sifting or stripping.

In practical computation, the transversals $R_i$ are not usually computed and stored explicitly; rather, they are encoded in a Schreier-tree data structure. Suppose that a base $B$ and an SGS $S$ for $G$ relative to $B$ are given. A Schreier-tree data structure for $G$ is a sequence of pairs $(S_i, T_i)$ called Schreier trees, one for each base point $\beta_i$, $1 \leq i \leq M$, where $T_i$ is a directed labelled tree with all edges directed toward the root $\beta_i$, and with edge labels selected from the set $S_i := S \cap G^{(i)} \subseteq G^{(i)}$. The nodes of $T_i$ are the points of the orbit $\beta_i^{G^{(i)}}$. The labels satisfy the condition that, for each directed edge from $\gamma$ to $\delta$ with label $h$, $\gamma^h = \delta$. If $\gamma$ is a node of $T_i$, then the sequence of the edge labels along the path from $\gamma$ to $\beta_i$ in $T_i$ is a word in the elements of $S_i$ such that the product of these permutations moves $\gamma$ to $\beta_i$. 
Thus each Schreier tree \( (S_t, T_t) \) defines inverses of the elements of a transversal for \( G^{(t+1)} \) in \( G^G \).

The reason for storing the inverses of transversal elements is that the sifting procedure uses these inverses. In some applications in the present paper, we need to recover the transversal elements themselves. This can be done, for example, by storing the inverses of the strong generators. Following the path from a node \( \gamma \) to the root \( \beta \) in \( T_t \), and writing the inverses of the labels we encountered in the opposite order, we get a word representing a permutation which moves \( \beta \) to \( \gamma \).

2.3. Some nearly linear-time algorithms. In this subsection, we describe some nearly linear-time algorithms used in our procedures. The fundamental result is that bases and strong generating sets can be computed.

**Theorem 2.1.** (Babai, Cooperman, Finkelstein, Seress [1]) Given a permutation group \( G \leq S_n \), a nonredundant base \( B \) and an SGS for \( G \) relative to \( B \) can be computed by a Monte Carlo algorithm in nearly linear time. The Schreier-tree data structure computed by the algorithm supports membership testing in \( G \) in \( O(n \log |G|) \) time.

For fast sifting, it is crucial that the depths of trees in the Schreier-tree data structure are small. In fact, the algorithm in [1] constructs Schreier trees such that the sum of depths is \( O(\log |G|) \). We call a Schreier-tree data structure shallow if the depth of each tree is at most \( 2 \log |G| \). A shallow Schreier-tree data structure supports membership testing in nearly linear time.

**Lemma 2.2** in [1] states that, given \( G = \langle S \rangle \leq \text{Sym}(\Omega) \) and \( \alpha \in \Omega \), group elements \( g_1, \ldots, g_k \) can be computed in deterministic time \( O(|S| \log^2 |G|) \) for some \( k \leq \log |G| \) such that the orbits \( \alpha^G \) and \( \alpha^{g_1 \cdots g_k} \) are equal, and the breadth-first-search tree for \( \alpha^G \), using the group elements \( g_1, \ldots, g_k, g_1^{-1}, \ldots, g_k^{-1} \), has depth at most \( 2k \leq 2 \log |G| \). This implies that given a nonredundant base \( B = (\beta_1, \ldots, \beta_M) \) and an SGS \( S \) relative to \( B \) for a group \( G \), we can replace \( S \) in nearly linear deterministic time by another SGS \( T \) relative to \( B \) with \( |T| \leq 2M \log |G| \) such that the Schreier-tree data structure obtained from \( T \) is shallow. Therefore, we can assume that all Schreier-tree data structures computed in our algorithms are shallow.

The method for constructing an SGS can be applied to a wide variety of tasks. In particular the following are straightforward consequences of Theorem 2.1.

**Lemma 2.2.** The following problems can be solved in nearly linear Monte Carlo time.

(i) Given \( G \leq \text{Sym}(\Omega) \) and an action \( \varphi : G \to \text{Sym}(\Delta) \), find the kernel of \( \varphi \). In particular, the pointwise stabilizer of a block system on \( \Omega \) can be computed.

(ii) Given \( G \leq \text{Sym}(\Omega) \) and \( A \subset \Omega \), find the pointwise stabilizer \( G_A \).

(iii) Given \( H \leq G \leq \text{Sym}(\Omega) \), find the normal closure \( \langle H^G \rangle \).

(iv) Given \( G \leq \text{Sym}(\Omega) \), find the derived series of \( G \).

For \( g \in G \), the \( g \)-image of a base of \( G \) uniquely determines \( g \). It is also possible to reconstruct \( g \) from the image of the base.

**Lemma 2.3.** Let \( S \) be an SGS for \( G \leq \text{Sym}(\Omega) \) relative to \( B \) and let \( t \) denote the sum of depths of Schreier trees coding the coset representative sets along the point stabilizer chain of \( G \). Then, given an injection \( f : B \to \Omega \), it is possible to
find a permutation \( g \in G \) with \( B^g = f \), or determine that no such element of \( G \) exists, in \( O(t|\Omega|) \) deterministic time.

There is no polynomial time algorithm known for computing the intersection of two permutation groups (see, e.g., [12]). However, generalizing a method from linear algebra for computing the intersection of subspaces, Cooperman, Finkelstein, and Luks [6] describe an algorithm which reduces the computation of \( (H^G) \cap G \) for \( H, G \leq \text{Sym}(\Omega) \) to the computation of a pointwise stabilizer in a permutation action of an appropriate group containing \( H \) and \( G \). Combining with Theorem 2.1 and Lemma 2.2(ii), we obtain the following.

**Theorem 2.4.** Given \( G, H \leq \text{Sym}(\Omega) \), \(|\Omega| = n\), the group \( (H^G) \cap G \) can be computed in \( O(n \log^2(||(H^G)||G||)) \) Monte Carlo time. In particular, if \( H \) is normalized by \( G \), then \( H \cap G \) can be computed.

In [10], Luks described a polynomial-time method for computing composition series in permutation groups. There is a nearly linear version of this result.

**Theorem 2.5.** (Beals, Seress [4]) Given \( G = \langle S \rangle \leq S_n \), a composition series of \( G \) can be computed in Monte Carlo time \( O(n|S|\log^2|G|) \). As in [10], the procedure also yields permutation representations of degree at most \( n \) for the composition factors of \( G \).

**Remark 2.6.** The composition series constructions of [10] and [4] are dependent upon the classification of finite simple groups (specifically, they cite "Schreier's Conjecture"). However, while the use of a composition series simplifies both our exposition and the present implementation in GAP, the main results of this paper can be achieved without this device (see Remark 5.4). Alternately, Beals [3] has shown that one can avoid classification dependence in the construction of a composition series.

3. Centralizer of a normal subgroup

The purpose of this section is to prove Theorem 1.2. The problem is reduced to the transitive case. For \( N \) normal in a transitive \( G \), \( C_G(N) \) is constructed in two main stages. An action of \( G \) is constructed with a kernel \( \tilde{G} \) (Lemma 3.4) and then an action of \( \tilde{G} \) is constructed with kernel \( C_G(N) \) (Lemma 3.6).

We start by recalling some well-known facts about centralizers of subgroups in the full symmetric group.

First we consider the case of transitive \( G \). Thus, let \( G \leq \text{Sym}(\Omega) \) be transitive and fix \( \alpha \in \Omega \). Observe that \( G_\beta = G_\alpha \) for all \( \beta \in \text{fix}(G_\alpha) \), since \( G_\alpha \leq G_\beta \) by the definition of \( \text{fix}(G_\alpha) \) and the transitivity of \( G \) implies that \(|G_\alpha| = |G_\beta| \). In general, the index of \( G_\alpha \) in \( G \) is the size of the orbit of \( \alpha \). Then the following holds.

**Lemma 3.1.** Suppose \( G \leq \text{Sym}(\Omega) \) is transitive and \( \alpha \in \Omega \). Then \( C_{\text{Sym}(\Omega)}(G) \) is semiregular and \( \text{fix}(G_\alpha) \) is an orbit of \( C_{\text{Sym}(\Omega)}(G) \).

Given \( \beta \in \text{fix}(G_\alpha) \), we can construct the (unique) \( c = c(\alpha, \beta) \in C_{\text{Sym}(\Omega)}(G) \) such that \( \alpha^c = \beta \). For this, given arbitrary \( \gamma \in \Omega \), \( \gamma^c \) is determined as follows. Take \( g \in G \) such that \( \alpha^g = \gamma \). Then

\[
\gamma^c = \alpha^{gc} = \alpha^{cg} = \beta^g.
\]

This construction is independent of the choice \( (g) \) of representative of the coset of \( G_\alpha \) carrying \( \alpha \) to \( \gamma \), since if \( \alpha^{g_1} = \alpha^{g_2} \) then \( g_1 g_2^{-1} \in G_\alpha = G_\beta \) and so \( \beta^{g_1} = \beta^{g_2} \).
Hence, given a transversal of \( G/G_\alpha \), encoded in a shallow Schreier tree, \( c \) can be computed in nearly linear time.

By semiregularity, \( C_{\text{Sym}(\Omega)}(G) = \langle U \rangle \) for any \( U \subseteq C_{\text{Sym}(\Omega)}(G) \) such that \( \alpha^{(U)} = \text{fix}(G_\alpha) \). Thus, to construct such \( U \), start with \( U = \emptyset \) and while there is some \( \beta \in \text{fix}(G_\alpha) \setminus \alpha^{(U)} \), add the corresponding \( c(\alpha, \beta) \) to \( U \). As each such augmentation of \( U \) at least doubles \( |\alpha^{(U)}| \), we have to apply the construction above for at most \( \log \log |C_{\text{Sym}(\Omega)}(G)| \leq \log n \) points \( \beta \in \text{fix}(G_\alpha) \). Therefore, \( C_{\text{Sym}(\Omega)}(G) \) can be computed in nearly linear time as well.

Next, we treat the case of intransitive \( G \leq \text{Sym}(\Omega) \). Since \( C_{\text{Sym}(\Omega)}(G) \) normalizes \( G \), it permutes the orbits of \( G \). We say that two orbits \( \Delta_1, \Delta_2 \) of \( G \) are equivalent, denoted \( \Delta_1 \equiv \Delta_2 \), if there is a bijection \( \varphi : \Delta_1 \to \Delta_2 \) such that for all \( g \in G \) and \( \delta \in \Delta_1 \),

\[
\delta^{\varphi g} = \delta^{g \varphi g}.
\]

(3.2)

Clearly, \( \equiv \) is an equivalence relation. Also, it is clear that if \( c \in C_{\text{Sym}(\Omega)}(G) \) maps \( \Delta_1 \) to \( \Delta_2 \), then the restriction \( c|_{\Delta_1} \) is a bijection satisfying (3.2). Conversely, if \( \varphi \) is a bijection satisfying (3.2) then \( \varphi \) induces an involution \( \varphi \) of \( C_{\text{Sym}(\Omega)}(G) \) such that \( \varphi|_{\Delta_1} = \varphi|_{\Delta_2} = \varphi^{-1}|_{\Delta_1} \), and \( \varphi \) fixes pointwise \( \Omega \setminus (\Delta_1 \cup \Delta_2) \). Moreover, if \( \Delta_1 \equiv \Delta_2 \) then \( C_{\text{Sym}(\Delta_1)}(G|_{\Delta_1}) \cong C_{\text{Sym}(\Delta_2)}(G|_{\Delta_2}) \). From these observations, we obtain the following.

**Lemma 3.2.** Let \( C_1, \ldots, C_t \) be the \( \equiv \)-equivalence classes of orbits of \( G \), \( |C_i| = k_i \), and, for each \( i \), choose a representative \( \Gamma_i \in C_i \). Then

\[
C_{\text{Sym}(\Omega)}(G) \cong \prod_{i=1}^{t} (C_{\text{Sym}(\Gamma_i)}(G|_{\Gamma_i}) \wr S_{k_i}).
\]

\[\square\]

If \( \alpha, \beta \in \Omega \) with \( G_\alpha = G_\beta \), then the orbits of \( \alpha \) and \( \beta \) are equivalent (a well-defined bijection \( \varphi : \alpha^G \to \beta^G \) satisfying (3.2) is given by \( (\alpha^g)^{\varphi} = \beta^g \), for all \( g \in G \)). By the above,

**Lemma 3.3.** Let \( \Delta_1, \Delta_2 \) be orbits of \( G \) of the same size, \( \alpha \in \Delta_1 \). Then \( \Delta_1 \equiv \Delta_2 \) if and only if \( \Delta_2 \cap \text{fix}(G_\alpha) \neq \emptyset \). \[\square\]

If \( \alpha \) is the first base point of \( G \) then, using the set of fixed points of \( G_\alpha \) and a shallow Schreier tree coding the transversal for \( G \mod G_\alpha \), we can compute the orbits equivalent to the orbit of \( \alpha \) and bijections between them in nearly linear time. Equivalence among other orbits can be determined by applying a base change algorithm to obtain a base starting with a point in the targeted orbit. However, if \( G \) has a lot of inequivalent orbits (but of the same size, so equivalence must be tested), then such computation of \( C_{\text{Sym}(\Omega)}(G) \) may require \( \Theta(n^2) \) time even for small-base inputs.

We turn now to the computation of centralizers of normal subgroups. If \( N \triangleleft G \leq \text{Sym}(\Omega) \) then \( C_{\text{Sym}(\Omega)}(N) \) is normalized by \( G \) so \( G(N) = C_{\text{Sym}(\Omega)}(N) \cap G \) is computable by Lemmas 3.1–3.3 and Theorem 2.4. However, as mentioned in the previous paragraph, \( C_{\text{Sym}(\Omega)}(N) \) may not be computable in nearly linear time and/or it may not be a small-base group even if \( G \) is. Hence we have to follow a different tack.

Let \( B \) be a nonredundant base for \( G \), \( \{\Psi_1, \Psi_2, \ldots, \Psi_m\} \) the set of orbits of \( G \) that have non-empty intersections with \( B \), \( \Psi := \bigcup_{i=1}^{m} \Psi_i \), and \( G^*, N^* \) the restrictions of
$G, N$ to $\Psi$, respectively. Since $\Psi$ contains a base for $G$, $G$ acts faithfully on $\Psi$ and $G \cong G^*$. Any element of $G^*$ can be uniquely extended to an element of $G$; moreover, by Lemma 2.3, the extension can be computed in nearly linear time, provided that the coset representatives along the point stabilizer chain corresponding to $B$ are stored in a shallow Schreier-tree data structure. Hence it is enough to compute $C_{G^*}(N^*)$.

Clearly, $C_{G^*}(N^*)$ is contained in

$$K = \prod_{i=1}^m C_{G^*|_{\Psi^1}}(N^*|_{\Psi^1}),$$

$K$ is normalized by $G^*$, and $|K| \leq |G|^m \leq |G|^\log |G|$. Hence, by Theorem 2.4, $C_{G^*}(N^*)$ is computable in nearly linear time as $C_{G^*}(N^*) = K \cap G^*$, and we reduced the computation of $C_G(N)$ to the computation of the groups $C_{G^*|_{\Psi^1}}(N^*|_{\Psi^1})$. So it suffices to solve the following problem.

*Given a transitive $G = \langle S \rangle \leq \text{Sym}(\Omega)$ and $N \triangleleft G$, compute $C_G(N)$ in $O(|\Omega| \log^2 |G|)$ time.* (3.3)

Note that, although $G$ is transitive in (3.3), $N$ may have many orbits. Thus, $|C_{\text{Sym}(\Omega)}(N)|$ may be too large in comparison to $|G|$, and so we still cannot apply Theorem 2.4 for computing $C_{\text{Sym}(\Omega)}(N) \cap G$ in nearly linear time. Instead, we proceed via an algorithm based on the following Lemmas 3.4–3.6.

**Lemma 3.4.** (i) Let $\alpha \in \Omega$ be fixed, and let $A = \text{fix}(\alpha)$. Then $A$ is a block of imprimitivity for $G$.

(ii) Let $B$ be the block system consisting of the $G$-images of $A$ and let $\bar{G}$ be the kernel of the $G$-action on $B$. Then $C_G(N) \leq \bar{G}$.

**Proof.** (i) By Lemmas 3.1, 3.2, and 3.3, $A$ is an orbit of $C_{\text{Sym}(\Omega)}(N)$. Since $G$ normalizes $C_{\text{Sym}(\Omega)}(N)$, the result follows.

(ii) The blocks in $B$ are the orbits of $C_{\text{Sym}(\Omega)}(N)$ so, in particular, are stabilized by $C_G(N)$. □

**Lemma 3.5.** Let $\Delta_1, \Delta_2, \ldots, \Delta_k$ be the orbits of $N$ which have nonempty intersection with $A$ and let $\Delta := \bigcup_{i=1}^k \Delta_i$. Then $\Delta$ is a block of imprimitivity for $G$.

**Proof.** Observe that $\Delta$ is the union of the $N$-images of $A$. Hence, in the $G$-action on $B$, $\{\Delta_1, \Delta_2, \ldots, \Delta_k\}$ is an orbit of $N$, and therefore a block for $G$. Hence, $\bigcup_{i=1}^k \Delta_i$ is a block for the $G$-action on $\Omega$. □

Next, choose $\alpha_i \in A \cap \Delta_i$ for $1 \leq i \leq k$. Since $\Delta$ is a block of imprimitivity for $G$, we can take $G$-images $\Delta_{\alpha_1}, \ldots, \Delta_{\alpha_m}$ of $\Delta$ which partition $\Omega$. Recall that by Lemma 3.3, $\{\Delta_1, \Delta_2, \ldots, \Delta_k\}$ is an equivalence class of $N$-orbits. Hence the set of images $\Gamma := \{\alpha_i^{\beta_j} : 1 \leq i \leq k, 1 \leq j \leq m\}$ define a system of representatives of all $N$-orbits with the property that two $N$-orbits are equivalent if and only if their representatives are in the same orbit of $C_{\text{Sym}(\Omega)}(N)$.
Lemma 3.6. (i) For each $g \in \tilde{G}$, there exists a unique $c_g \in C_{\text{Sym}(\Omega)}(N)$ such that $g c_g^{-1}$ fixes $\Gamma$ pointwise.
(ii) The mapping $\varphi : \tilde{G} \rightarrow \text{Sym}(\Omega)$ defined by $g \mapsto g c_g^{-1}$ is a group homomorphism.
(iii) $C_G(N)$ is the kernel of the homomorphism $\varphi$.

Proof. (i) Recall that by Lemma 3.2,

$$C_{\text{Sym}(\Omega)}(N) \cong \prod_{j=1}^{m} (C_{\text{Sym}(\Delta_{\varphi j}^{\Omega})}((N|_{\Delta_{\varphi j}^{\Omega}}) \wr S_k)).$$

Since $\tilde{G}$ fixes the blocks in $B$, $g$ fixes $\Delta_{\varphi j}$ for $1 \leq j \leq m$. There are $k$ $N$-orbits in $\Delta_{\varphi j}$ and $g$ permutes them; this permutation, regarded as an element of $S_k$, must also be induced by $c_g$ in the wreath product in the $j$th term of (3.4). Since $g$ fixes the $G$-images of $A$ and $C_{\text{Sym}(\Delta_{\varphi j}^{\Omega})}((N|_{\Delta_{\varphi j}^{\Omega}}) \wr S_k)$ acts transitively on $A_{\varphi j} \cap \Delta_{\varphi j}^{\Omega}$, for $1 \leq i \leq k$, $1 \leq j \leq m$, we can also choose $c_g$ to fix $\Gamma$ pointwise; since $C_{\text{Sym}(\Delta_{\varphi j}^{\Omega})}((N|_{\Delta_{\varphi j}^{\Omega}})$ is semiregular, this choice is also uniquely determined.

(ii) Observe that for $g, h \in \tilde{G}$, $g c_g^{-1} h c_h^{-1} = g h (c_g^{-1})^h c_h^{-1}$. Here $(c_g^{-1})^h c_h^{-1} \in C_{\text{Sym}(\Omega)}(N)$ and $g h (c_g^{-1})^h c_h^{-1}$ fixes $\Gamma$ pointwise so, from the unicity of $c_{gh}$, $c_{gh}^{-1} = (c_g^{-1})^h c_h^{-1}$.

(iii) $C_G(N)$ is the kernel of the homomorphism $\varphi$ since $\varphi(g) = 1$ if and only if $g = c_g$, i.e. $g \in \tilde{G} \cap C_{\text{Sym}(\Omega)}(N) = C_G(N)$. \qed

Summarizing, the algorithm for computing $C_G(N)$ is the following.

Procedure CNSinT($G, N$) [[Centralizer of Normal Subgroup in Transitive Group]]

Input: $N, G \leq \text{Sym}(\Omega)$, $N \triangleleft G$, $G$ transitive on $\Omega$.

Output: $C_G(N)$.

Step 1. Compute a base $B$ and an SGS relative to $B$ for $G$. Compute an SGS relative to $B$ for $N$.

Step 2. Let $\alpha$ be the first point of $B$. Compute $A := \text{fix}(N_{\alpha})$.

Step 3. Compute the block system $B$ consisting of the $G$-images of $A$, and $\tilde{G}$, the kernel of the $G$-action on $B$.

Step 4. Compute those $N$-orbits $\Delta_1, \ldots, \Delta_k$ which have nonempty intersection with $A$, and compute $\Delta := \bigcup_{i=1}^{k} \Delta_i$. For $1 \leq i \leq k$, pick $\alpha_i \in A \cap \Delta_i$.

Step 5. Compute $G$-images $\Delta_{\varphi 1}, \ldots, \Delta_{\varphi m}$ which partition $\Omega$. Compute $\Gamma := \{\alpha_{i_j}^{\varphi_j} : 1 \leq i \leq k, 1 \leq j \leq m\}$.

Step 6. For each generator $g$ of $\tilde{G}$, compute $c_g \in C_{\text{Sym}(\Omega)}(N)$ such that $g c_g^{-1}$ fixes $\Gamma$ pointwise.

Step 7. Compute and output the kernel of the homomorphism $\varphi : \tilde{G} \rightarrow \text{Sym}(\Omega)$ defined by $\varphi : g \mapsto g c_g^{-1}$.
procedure CNS($G, N$) \footnote{Centralizer of Normal Subgroup}

INPUT: $N, G \leq \text{Sym}(\Omega)$, $N \triangleleft G$.
OUTPUT: $C_G(N)$.

Step 1. Construct a nonredundant base $B$ for $G$.

Step 2. Let $\{\Psi_1, \ldots, \Psi_k\}$ be the $G$-orbits having a nonempty intersection with $B$. Let $\Psi := \bigcup_{i=1}^{k} \Psi_i$. For $1 \leq i \leq k$ do compute $K_i := \text{CNSInT}(G|\Psi_i, N|\Psi_i)$.

Step 3. Compute $C^* := \prod_{i=1}^{k} K_i \cap G|\Psi$.

Step 4. Extend the generators of $C^*$, considered as restrictions of elements of $G$, to act on $\Omega$. Output the extensions.

4. Quotients with small permutation degree

For iterative processes such as computing the upper central series, it would be useful to construct permutation representations of quotient groups. However, as shown by Neumann [14], there are examples of $G \leq S_n$, $N \triangleleft G$ such that $G/N$ has no faithful permutation representation on less than $2^{n/4}$ points. On the positive side, it is proven by Easdown, Praeger [7], and Luks [11, 12] that if $N$ is abelian then there exists a homomorphism $\varphi : G \rightarrow S_n$ such that $N \leq \ker(\varphi)$ and, in a sense, $\ker(\varphi)$ is not too big. Also, as observed in [11], $\varphi$ can be constructed efficiently. We describe that construction in this section. In the next step, we shall apply it in the proof of Theorem 1.1.

**Theorem 4.1.** Let $G \leq \text{Sym}(\Omega)$, $|\Omega| = n$ and let $N \triangleleft G$ be abelian. Then there exists a homomorphism $\varphi : G \rightarrow S_n$ such that $N \leq \ker(\varphi)$ and the prime divisors of $|N|$ and $|\ker(\varphi)|$ are the same. In particular, if $N$ is a $p$-group then $\ker(\varphi)$ is an abelian $p$-group; moreover, if $N$ is elementary abelian then $\ker(\varphi)$ is elementary abelian. Given generators for $G$ and $N$, $\varphi$ can be constructed in deterministic linear time.

**Proof.** Let $\Delta_1, \ldots, \Delta_k$ be the orbits of $N$, $\Omega = \bigcup_{i=1}^{k} \Delta_i$. Lemma 3.1 implies that a transitive abelian group is regular, so $N$ acts regularly on each set $\Delta_i$. In particular, $N|\Delta_i$ has $|\Delta_i|$ elements. Let $\Sigma_i$ denote the set of permutations in $N|\Delta_i$, and let $\Sigma := \bigcup_{i=1}^{k} \Sigma_i$. Clearly, $|\Sigma| = n$. We shall define an action of $G$ on $\Sigma$, satisfying the conclusion of the theorem.

Given $g \in G$ and $\sigma_i \in \Sigma_i$, we define $\sigma_i^g \in \Sigma$ as follows. Since $N \triangleleft G$, $G$ permutes the orbits of $N$; suppose that $\Delta_i = \Delta_j$. Take any $x \in N$ with $x|\Delta_i = \sigma_i$, and let $\sigma_i^g := x^g|\Delta_j$. This mapping is well-defined since, if $x|\Delta_i = y|\Delta_i$ for some $x, y \in N$, then $x^g|\Delta_j = y^g|\Delta_j$. It is also clear that this construction defines a homomorphism $\varphi : G \rightarrow \text{Sym}(\Sigma)$.

Now $N \leq \ker(\varphi)$, since $N$ fixes its own orbits and commutes with $N|\Delta_i$. Conversely, if $g \in \ker(\varphi)$ then $g$ fixes the orbits of $N$ and $g|\Delta_i$ centralizes $N|\Delta_i$. Since the regular subgroup $N|\Delta_i$ is self-centralizing in $\text{Sym}(\Delta_i)$, we conclude that $g|\Delta_i \in N|\Delta_i$. Hence the transitive constituents of $N$ and $\ker(\varphi)$ are the same, implying the required relations between the prime factors and exponents.
To see the algorithmic feasibility of the construction, let us fix a representative \( \delta_i \in \Delta_i \) in each orbit of \( N \), and build a (breadth-first-search) Schreier tree \( T_i \) with root \( \delta_i \) for \( N \mod N_i \). In this tree, contrary to the usual Schreier-tree constructions, we direct the edges away from the root. Identifying the trivial permutation in \( \Sigma_i \) with \( \delta_i \), the elements of \( \Sigma_i \) naturally correspond to the elements of \( \Delta_i \). If \( \sigma_i \in \Sigma_i \) corresponds to \( \gamma \in \Delta_i \), then, in order to compute \( \sigma_i^\sigma \), it is sufficient to determine the image of \( \delta_i \) under the permutation \( \sigma_i \), where \( \sigma_i \) is the product of edge labels in \( T_i \) along the path from \( \delta_i \) to \( \gamma \). The path from \( \delta_i \) to \( \gamma \) may have length \( \Theta(|\Delta_i|) \), so computing one image \( \delta_i^\sigma \) may require \( \Theta(|\Delta_i|) \) time. However, if \( \delta \) is the parent of \( \gamma \) in \( T_i \) and \( \delta_i^\sigma \) has already been computed, then \( \delta_i^\sigma \) can be obtained in constant time. Thus, the entire image \( \Sigma_i^\sigma \) can be constructed in \( O(|\Delta_i|) \) time. \( \square \)

5. Computing \( O_p(G) \), \( O_{\infty}(G) \), and \( F(G) \)

In this section, we prove Theorem 1.1. We shall refer to \( O_{\infty}(G) \) as the radical of \( G \) and \( O_p(G) \) as the \( p \)-core of \( G \). Note that, since two solvable normal subgroups generate a solvable normal subgroup, there is a unique maximal solvable normal subgroup in \( G \). Similarly, \( G \) has a unique maximal normal \( p \)-subgroup. By the same argument, we see that \( O_{\infty}(G) \) and \( O_p(G) \) are characteristic in \( G \). Since the Fitting subgroup is the direct product of the \( p \)-cores, where \( p \) runs through the \( O(\log |G|) \) prime divisors of \( |G| \), it is enough to construct \( O_p(G) \) and \( O_{\infty}(G) \).

Theorem 4.1 was the key to the polynomial-time algorithms for computing \( O_{\infty}(G) \) and \( O_p(G) \) in [11]. We use the basic idea from [11] with some changes to achieve the nearly linear running time. The subgroups \( O_p(G) \) and \( O_{\infty}(G) \) are computed similarly, so we handle both of them simultaneously. We shall describe the construction of the radical, indicating the necessary changes for the construction of \( p \)-cores in double brackets [\[\cdots\]].

First, let us consider the special case when \( G \) has a maximal normal subgroup \( N \) with \( O_{\infty}(N) = 1 \) [\[O_p(N) = 1\]]. This condition restricts severely the radical and \( p \)-core of \( G \).

**Lemma 5.1.** Suppose that \( N \triangleleft G \) is a maximal normal subgroup and \( O_{\infty}(N) = 1 \) [\[O_p(N) = 1\]]. If \( G/N \) is not cyclic [\[not cyclic of order \( p \)\]], then \( O_{\infty}(G) = 1 \) [\[O_p(G) = 1\]].

**Proof.** Since the radical [\[p\text{-core}\]] of \( G \) intersects \( N \) trivially, it must centralize \( N \). In particular, if the radical [\[p\text{-core}\]] of \( G \) is non-trivial then \( C_G(N) \) has non-trivial radical [\[p\text{-core}\]]. If \( C_G(N) \leq N \) then its radical [\[p\text{-core}\]] is trivial. If \( C_G(N) \not\leq N \) then \( G = NC_G(N) \) and \( C_G(N)/Z(N) \cong G/N \), so \( C_G(N) \) could have non-trivial radical [\[p\text{-core}\]] only if \( G/N \) is cyclic [\[is cyclic of order \( p \)\]]. \( \square \)

**Lemma 5.2.**

(i) Suppose that \( N \triangleleft G \) is a maximal normal subgroup, \( O_{\infty}(N) = 1 \), and \( G/N \) is cyclic. Then \( O_{\infty}(G) = C_G(G) \).

(ii) Suppose that \( N \triangleleft G \) is a maximal normal subgroup, \( O_p(N) = 1 \), and \( |G/N| = p \). Then \( O_p(G) \neq 1 \) if and only if \( C_G(N) \) is abelian and \( p|C_G(G)| \).

**Proof.** (i) Since \( O_{\infty}(G) \cap N \leq O_{\infty}(N) = 1 \), we know \( O_{\infty}(G) \leq C_G(N) \). Suppose \( C_G(N) \neq 1 \). Since \( C_G(N) \cap N = Z(N) \leq O_{\infty}(N) = 1 \), it follows from the maximality of \( N \) that \( G = C_G(N)N \). Thus, \( G/N = C_G(N)N/N \cong C_G(N)/(C_G(N) \cap N) \cong C_G(N) \), so that \( C_G(N) \) is cyclic. Hence, \( C_G(N) \leq O_{\infty}(G) \).
(ii) If $C_G(N)$ is abelian and $p | [C_G(N)]$ then the Sylow $p$-subgroup of $C_G(N)$ is a non-trivial normal $p$-subgroup of $G$, so that $O_p(G) \neq 1$. Suppose, conversely, that $O_p(G) \neq 1$. Since $O_p(G) \cap N \leq C_G(N) = 1$, we know $O_p(G) \leq C_G(N)$; in particular, $C_G(N) \leq N$, so that $G = C_G(N) \cdot N$. In this case, $C_G(N)/Z(N) \cong G/N \cong C_p$, so $p | [C_G(N)]$; furthermore, $C_G(N) = \langle Z(N), g \rangle$ for any $g \in C_G(N) \setminus Z(N)$. But then $C_G(N)$ is abelian since $g$ commutes with $N$ and therefore with $Z(N)$.

We shall also need the following simple observation.

**Lemma 5.3.** Suppose that $L < H$, with $H$ subnormal in $G$ and with $L$ solvable [[L a p-group]]. Then $\langle L_G \rangle$ is solvable [[\langle L_G \rangle is a p-group]].

**Proof.** Let $H < G_1 < G_2 < \cdots < G_m = G$. Prove by induction on $i$ that $\langle L_{G_i} \rangle$ is solvable [[p-group]].

Now we are ready to describe the construction of $O_\infty(G) [[O_p(G)]]$.

First, we produce some non-trivial solvable [[p-group]] $H < G$ provided $G$ has a non-trivial radical [[p-core]] (so that failure guarantees that the radical [[p-core]] is trivial). For this, we compute a composition series $1 = N_1 < N_2 < \cdots < N_m = G$. Then we find the smallest index $i$ such that $N_i$ has a non-trivial radical [[p-core]] as follows. Suppose that $O_\infty(N_i) = 1$ [[$O_p(N_i) = 1$]]. If $N_{i+1}/N_i$ is not cyclic [[not cyclic of order $p$]] then, by Lemma 5.1, we conclude that $O_\infty(N_{i+1}) = 1$ [[$O_p(N_{i+1}) = 1$]]. Otherwise, we compute $C_{N_{i+1}}(N_i)$, using the method of Section 3. If $C_{N_{i+1}}(N_i) = 1$ [[$C_{N_{i+1}}(N_i)$ not abelian with $p | C_{N_{i+1}}(N_i)$]] then, by Lemma 5.2, $O_\infty(N_{i+1}) = 1$ [[$O_p(N_{i+1}) = 1$]]. Otherwise, set $L := C_{N_{i+1}}(N_i) \neq 1$ [[$L :=$ any non-trivial p-subgroup of $C_{N_{i+1}}(N_i)$, e.g., take any generator $g$ with $p | \text{order}(g)$ and set $L := \langle g^{\text{order}(g)/p} \rangle$]]. By Lemma 5.3, $H = \langle L_G \rangle$ is a solvable normal subgroup [[normal p-group]] of $G$.

By taking the last non-trivial term in the derived series of $H$, we produce a non-trivial abelian normal subgroup [[abelian normal p-subgroup]] $N$ of $G$. With respect to this $N$, we compute the homomorphism $\varphi : G \to S_n$ described in Theorem 4.1.

Finally, we recursively compute $M = O_\infty(\varphi(G)) [[M = O_p(\varphi(G))]]$. Since $\ker(\varphi)$ is abelian [[an abelian p-group]], $\varphi^{-1}(M)$ is $O_\infty(G) [[O_p(G)]]$. The inverse image, $\varphi^{-1}(M)$, is generated by $\ker(\varphi)$ together with liftings of the generators of $M$, the liftings obtained, say, via Lemma 2.3.

(Note that the composition-series computation need not be repeated for $\varphi(G)$. We have $1 = \varphi(N_1) \trianglelefteq \varphi(N_2) \trianglelefteq \cdots \trianglelefteq \varphi(N_m) = \varphi(G)$, with $\varphi(N_{i+1})/\varphi(N_i) \cong N_{i+1}/N_i$ or $\varphi(N_{i+1})/\varphi(N_i) \cong 1$.)

This completes the description of the algorithm, proving Theorem 1.1.

**Remark 5.4.** As mentioned in Remark 2.6, the use of composition factors can be avoided. In [11, 12], it is shown that one can make use of any proper normal subgroup $N$ in a recursive construction of a suitable $H$ as above. Namely, recursively seek a solvable [[a p-group]] $L$ such that $1 < L \trianglelefteq N$. If this succeeds, set $H := \langle L_G \rangle$. Otherwise similarly seek a solvable [[b p-group]] $L$ such that $1 < L \leq C_G(N)$. A second failure implies $O_\infty(G) = 1 [[O_p(G) = 1]]$. Critical now to the nearly-linear timing is the observation that the second recursive call needs to be made only when $O_\infty(N) = 1 [[O_p(G) = 1]]$ in which case $|N|/|C_G(N)| \leq |G| [[|N_p|/|C_G(N)|_p \leq |G|_p]$, where $|X|_p$ denotes the $p$-part of $|X|$]. Finally, one shows,
under the assumption that $O_{\infty}(G) \neq 1$, one can get by with relatively elementary (in particular, not dependent on the simple groups classification) portions of the methods of [10].

References

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE, UNIVERSITY OF OREGON, EUGENE, OR 97403
E-mail address: luks@cs.uoregon.edu

DEPARTMENT OF MATHEMATICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OH 43210
E-mail address: akos@math.ohio-state.edu