An implementation of the generalized Lusztig–Shoji algorithm*

Pramod N. Achar

Version 0.98f, December 2019

1 Introduction

Let V be a finite-dimensional complex vector space, and let $W \subset GL(V)$ be a complex reflection group. Let q be an indeterminate. For an irreducible character $\chi \in \mathrm{Irr}(W)$, let $R(\chi)$ denote its fake degree, a polynomial in q. We define $R(\chi)$ for reducible characters as well, by extending linearly. Let N^* be the number of reflections in W, and let $\Omega = (\omega_{\chi,\chi'})_{\chi,\chi' \in \mathrm{Irr}(W)}$ be the square matrix with entries in $\mathbb{Z}[q]$ given by

$$\omega_{\chi,\chi'}(q) = q^{N^*} R(\chi \otimes \chi' \otimes \overline{\det}_V).$$

For $\chi \in \mathrm{Irr}(W)$, let $b(\chi)$ be the lowest power of q occurring in $R(\chi)$.

A **Lusztig–Shoji datum** for W is an ordered collection X of disjoint subsets of $\mathrm{Irr}(W)$ such that for each $\chi \in \mathrm{Irr}(W)$, χ and its complex conjugate $\bar\chi$ belong to the same member of X. (The last condition is vacuous for Coxeter groups, of course.) The members of X are called **phyla**. For any phylum $\mathcal{C}$, we define

$$a(\mathcal{C}) = \min\{b(\chi) \mid \chi \in \mathcal{C}\}.$$

Given a Lusztig–Shoji datum, if $\chi \in \mathrm{Irr}(W)$ occurs in some phylum, we denote that class by $\mathrm{phy}\,\chi$. By an abuse of notation, we sometimes write $a(\chi)$ for $a(\mathrm{phy}\,\chi)$. Typically, we also assume that each phylum $\mathcal{C}$ contains a *unique* member χ such that $a(\mathcal{C}) = b(\chi)$. This representation is called the **leading term** of $\mathcal{C}$, and is denoted $\chi_{\mathcal{C}}$.

Let $I(X) \subset \mathrm{Irr}(W)$ denote the union of all phyla. (In other sources, it has been assumed that $I(X) = \mathrm{Irr}(W)$, but we allow $I(X)$ to be smaller here.) Let Ω_X be the submatrix $(\omega_{\chi,\chi'})_{\chi,\chi' \in I(X)}$. Consider the matrix equation

$$P \Lambda P^{\mathrm{t}} = \Omega_X, \tag{1}$$

where $P = (p_{\chi,\chi'})_{\chi,\chi' \in I(X)}$ and $\Lambda = (\lambda_{\chi,\chi'})_{\chi,\chi' \in I(X)}$ are unknown square matrices over $\mathbb{Q}(q)$ satisfying

$$\begin{aligned} p_{\chi,\chi'} &= 0 && \text{if } \mathrm{phy}\,\chi < \mathrm{phy}\,\chi', \\ p_{\chi,\chi'} &= \delta_{\chi,\chi'} q^{a(\chi)} && \text{if } \mathrm{phy}\,\chi = \mathrm{phy}\,\chi', \\ \lambda_{\chi,\chi'} &= 0 && \text{if } \mathrm{phy}\,\chi \neq \mathrm{phy}\,\chi'. \end{aligned} \tag{2}$$

Theorem 1.1 (Lusztig, Shoji)**.** *There are unique matrices P, Λ over $\mathbb{Q}(q)$ satisfying* (1) *and* (2)*.*

The proof of this theorem consists primarily of an algorithm, the **generalized Lusztig–Shoji algorithm**, for computing P and Λ. The software described here is an implementation of this algorithm.

Various kinds of information can be extracted from the output of the Lusztig–Shoji algorithm. First, there is a natural partial order on X, denoted $\preceq$, obtained by taking the transitive closure of the preorder defined by

$$\mathcal{C}' \preceq \mathcal{C} \quad \text{if } p_{\chi,\chi'} \neq 0 \text{ for some } \chi \in \mathcal{C},\ \chi' \in \mathcal{C}'.$$

(The given total order $\leq$ on a Lusztig–Shoji datum is always compatible with $\preceq$, but $\preceq$ is in general weaker.)

Given a set of phyla $Y \subset X$, let us define its **closure** $\bar{Y}$ to be the set $\{\mathcal{C}' \mid \mathcal{C}' \preceq \mathcal{C} \text{ for some } \mathcal{C} \in Y\}$. Y is **closed** if $Y = \bar{Y}$, **locally closed** if $\bar{Y} \smallsetminus Y$ is closed, and **irreducible** if it contains a unique maximal member with respect to $\preceq$. An irreducible locally

*This software was developed with the support of NSF Grant DMS-0500873. **Warning:** The present version has undergone nothing resembling rigorous testing.

closed set Y, with maximal member $\mathcal{C}$, is said to be **rationally smooth** if

$$p_{\chi_\mathcal{C},\chi'} = \begin{cases} q^{a(\mathcal{C})} & \text{if } \chi' = \chi_{\mathcal{C}'} \text{ for some } \mathcal{C}' \in Y, \\ 0 & \text{if phy}\,\chi' \in Y. \end{cases}$$

(No condition is imposed if phy $\chi' \notin Y$.)

Finally, a phylum $\mathcal{C} \in X$ is **special** if its leading term $\chi_\mathcal{C}$ is a special representation of W in the sense of [10]. Given a special phylum $\mathcal{C}$, let $P_\mathcal{C} \subset X$ be the irreducible locally closed subset obtained by deleting from $\bar{\mathcal{C}}$ the closures of any other special phyla it contains. A set obtained in this way is a **special piece**. Every phylum belongs to a special piece; often (see below), the special pieces are disjoint.

The most important (and, historically, the first) instance of the Lusztig–Shoji algorithm is that in which W is the Weyl group of a simple algebraic group G, and X is the partitioning of $\mathrm{Irr}(W)$ according to the Springer correspondence, with phyla in bijection with unipotent classes. In this case, all the concepts discussed above have well-known interpretations in geometric representation theory. The entries of P describe the stalks of intersection cohomology complexes on the unipotent variety, and the entries of Λ describe a certain "inner product" of local systems on unipotent classes. In particular, the polynomial $\lambda_{\chi_\mathcal{C},\chi_\mathcal{C}}$ gives the number of $\mathbb{F}_q$-points of the unipotent class corresponding to $\mathcal{C}$. The order $\preceq$ is simply the usual closure partial order on unipotent classes, and the topological terms "closed," "locally closed," and "irreducible" have their usual meanings. A variety Y is rationally smooth if $\mathrm{IC}(Y, \bar{\mathbb{Q}}_\ell) \simeq \bar{\mathbb{Q}}_\ell$; this coincides with the condition on P given above. The special pieces are disjoint (see [16]), and the full unipotent variety (*i.e.*, all of X) is rationally smooth [4], as is each special piece [11, 14, 8].

2 Basic Functions

This software requires GAP version 3.4.4 (*not* the newer GAP 4) and the "development version" of the CHEVIE package, available from the homepage of Jean Michel:

```
http://www.institut.math.jussieu.fr/
  ~jmichel/chevie/chevie.html
```

Once `lsalg.g` has been downloaded, it can be loaded into GAP with the command

```
> Read("lsalg.g");
```

This command will load the CHEVIE package by itself; it is not necessary to enter the command `RequirePackage("chevie")`.

Henceforth, the word **group** will always mean an irreducible complex reflection group, created by one of the CHEVIE commands `ComplexReflectionGroup` or `CoxeterGroup`. Each group `W` comes equipped with a fixed ordered list of irreducible characters, accessible via commands such as `CharTable(W)` and `ChevieCharInfo(W)`. Assigning each $\chi \in \mathrm{Irr}(\mathtt{W})$ to its position in this list gives us a fixed bijection

$$\mathrm{Irr}(\mathtt{W}) \longleftrightarrow \{1, \ldots, |\mathrm{Irr}(\mathtt{W})|\}.$$

If $\chi \in \mathrm{Irr}(W)$ corresponds to the number i under this bijection, let $\sigma(i)$ denote the number corresponding to $\bar{\chi}$.

A **Lusztig–Shoji datum** for a group `W` is a list of lists of integers in the range $\{1, \ldots, |\mathrm{Irr}(\mathtt{W})|\}$, such that no integer occurs twice, and such if i occurs in some sublist, then $\sigma(i)$ occurs in the same sublist. To identify this kind of Lusztig–Shoji datum with that of the previous section, the list should be regarded as a list of phyla in **increasing** order.

The Lusztig–Shoji algorithm is carried out by the following commands:

`LSAlg(` *group*`,` *LSdatum* `)`
`LSAlgRatl(` *group*`,` *LSdatum* `)`

These commands differ in their behavior when there is some entry of P or Λ that is not at least a Laurent polynomial. `LSAlg` simply stops when it finds such an entry, whereas `LSAlgRatl` is able to carry out arithmetic with rational functions, and always completes the algorithm. `LSAlgRatl` is rather slow when rational-function arithmetic is required; usually, it is preferable to use `LSAlg`.

The output of these commands is a record containing the following fields:

- `P`, `Pdenom`, `Lam`, `Ldenom`: `P` and `Lam` are matrices of Laurent polynomials, and `Pdenom` and `Ldenom` are nonzero polynomials with nonzero constant term. Together, these fields give the solution to (1), by the formulas

$$P = \frac{\texttt{P}}{\texttt{Pdenom}}, \qquad \Lambda = \frac{\texttt{Lam}}{\texttt{Ldenom}}.$$

The rows and columns of `P` and `Lam` are ordered in a way compatible with the Lusztig–Shoji datum, so that `P` is upper-triangular.

When `LSAlg` is used, `Pdenom` and `Ldenom` will always be equal to `1`. If `LSAlg` fails to complete the algorithm, the entries of `P` and `Lam` whose values were not determined will contain the boolean value `false` rather than a Laurent polynomial.

- `valid`: this has the value `true` if `LSAlg` completes the algorithm, and `false` if it does not. It is always `true` when `LSAlgRatl` is used.
- `label`: a list of names for characters, in the order in which they should be used to label rows and columns of `P` and `Lam`.
- `phy`: a copy of the input *LSdatum*.

Example 2.1. Let us compute the Green functions for an algebraic group of type G_2. We first create the group:

```
> W := CoxeterGroup("G",2);
```

The command

```
> ChevieCharInfo(W).charnames;
```

shows the ordered list of characters using Carter's notation:

$$[\ \phi_{1,0},\ \phi_{1,6},\ \phi'_{1,3},\ \phi''_{1,3},\ \phi_{2,1},\ \phi_{2,2}\ ]$$

The relevant Lusztig–Shoji datum has one phylum for each unipotent class in G_2, and each phylum consists of those characters attached to the corresponding unipotent class by the Springer correspondence. The Springer correspondence is given explicitly in Carter's book; we find that the desired Lusztig–Shoji datum is

```
> LS := [ [2], [4], [6], [5,3], [1] ];
```

We run the algorithm with

```
> out := LSAlg( W, LS );;
```

The matrices P and Λ may be examined interactively using the commands

```
> PrintArray(out.P);
> PrintArray(out.Lam);
```

We can also use the LaTeX output command

```
> TeXLSP( "output.tex", W, out );
```

to generate the table

	$\phi_{1,0}$	$\phi_{2,1}$	$\phi'_{1,3}$	$\phi_{2,2}$	$\phi''_{1,3}$	$\phi_{1,6}$
$\phi_{1,0}$	1	1		1	1	1
$\phi_{2,1}$		q		q	q	q^5+q
$\phi''_{1,3}$			q	q		q^3
$\phi_{2,2}$				q^2	q^2	q^4+q^2
$\phi'_{1,3}$					q^3	q^3
$\phi_{1,6}$						q^6

3 Lusztig–Shoji Data

Entering Lusztig–Shoji data by hand, as was done in Example 2.1, is an arduous and error-prone task. This section describes a number of functions that automatically generate Lusztig–Shoji data of various kinds. These functions come in pairs of the form

name`LS(` *group*`,` *parameters* `)`
name`LSdatum(` *group*`,` *parameters* `)`

The first version invokes `LSAlg` with the automatically generated Lusztig–Shoji datum, and returns its output. The second version simply returns the Lusztig–Shoji datum without invoking `LSAlg`.

For each of the functions below, we describe the assumptions imposed on the *group*, and we describe how the phyla are determined. These functions fall into two categories: they are **full** if $I(\texttt{W}) = \mathrm{Irr}(\texttt{W})$, and **partial** otherwise.

3.1 Full Lusztig–Shoji data

`SymbolLS(` *group*`,` *r*`,` *s* `)`
`SymbolLSdatum(` *group*`,` *r*`,` *s* `)`

The *group* must be imprimitive, and r and s should be integers with $0 \leq s \leq r$. Phyla are similarity classes of generalized symbols of type (r, s), as defined in [15, §1.2].

```
RouquierLS( group )
RouquierLSdatum( group )
```

The *group* must be imprimitive or spetsial primitive. Phyla are Rouquier blocks [13] associated to spetsial cyclotomic Hecke algebras. For Coxeter groups, these coincide with two-sided cells. For complex reflection groups, Rouquier blocks have been determined in [5, 7, 12].

```
SpringerLS( group )
SpringerLSdatum( group )
```

```
Springer2LS( group )
Springer2LSdatum( group )
```

The *group* must be spetsial. For Weyl groups, phyla are determined by the Springer correspondence: each phylum is the set of characters attached to some unipotent class in the corresponding algebraic group. For imprimitive complex reflection groups, the phyla are similarity classes of symbols of the type associated to Springer representations in [2]. For primitive groups, the phyla are in bijection with the set of Springer representations in the sense of [2]. Each phylum contains a Springer representation and its complex conjugate. Each phylum containing a special Springer representation also contains all other representations in the same Rouquier block that are not Springer or conjugate-to-Springer.

For groups with two distinct root lattices—*viz.*, $G(d, 1, n)$ with d a prime power (including $G(2, 1, n)$, the Weyl group of type B_n), G_6, and G_{26}—there are two distinct notions of "Springer representation." For these groups, the functions `Springer2LS` and `Springer2LSdatum` behave just as described above, but with respect to the "second" root lattice.

3.2 Partial Lusztig–Shoji data

Most of the functions in this section do not, in general, lead to valid output from `LSAlg` even for Weyl groups. Rather, they are useful (in the *name*`LSdatum` form) for obtaining lists of representations, and they are used in this way internally by the software for such tasks as determining the special pieces.

```
LeadingTermsLS( group, LSdatum )
LeadingTermsLSdatum( group, LSdatum )
```

This function takes an existing Lusztig–Shoji datum and produces a new one from it by discarding from each phylum all but the member with minimal b-value and its complex conjugate. If there is more than one such member, the behavior is unpredictable.

```
DistSymbolsLS( group, r, s )
DistSymbolsLSdatum( group, r, s )
```

The *group* must be imprimitive. Each phylum consists of a character whose symbol of type (r, s) is distinguished, together with its complex conjugate. Equivalent to applying `LeadingTermLS` to the output of `SymbolLS`.

```
SpecialLS( group )
SpecialLSdatum( group )
```

The *group* must be spetsial. Each phylum consists of a special character and its complex conjugate. Equivalent to applying `LeadingTermsLS` to the output of `RouquierLSdata`.

```
UnipotentLS( group )
Unipotent2LS( group )
```

Equivalent to applying `LeadingTermLS` to the output of `SpringerLS` or `Springer2LS`. Each phylum consists of a Springer representation and its complex conjugate. For spetsial complex reflection groups,

```
TrivialLS( group )
```

Each phylum consists of a character and its complex conjugate.

4 Using the Output of `LSAlg`

This section describes a number of functions that are useful for studying the output of `LSAlg`. The *algrec* argument to each of these functions should be a record returned by `LSAlg` or `LSAlgRatl`.

4.1 Boolean functions

The functions in this section test `P` and `Lam` for various properties, and return `true` or `false`.

`IsLSIntegral(` *group*`,` *algrec* `)`

This function returns `true` if P and Λ have polynomial entries; *i.e.*, if `Pdenom` and `Ldenom` are both `1`, and if all entries of `P` and `Lam` are polynomials.

`IsLSPerverse(` *group*`,` *algrec* `)`

This function is very similar to `IsLSIntegral`, but it imposes on `P` the stronger condition that all nonzero terms in the row corresponding to character χ have degree at least $a(\chi)$. This condition holds in the original version of the Lusztig–Shoji algorithm, where the entries of P describe stalks of simple perverse sheaves on the unipotent variety.

`IsLSPositive(` *group*`,` *algrec* `)`

This function returns `true` if `Pdenom = 1` and all coefficients in `P` are nonnegative.

`IsLSRatlSmooth(` *group*`,` *algrec* `)`

This function returns `true` if the full Lusztig–Shoji datum and all special pieces are all rationally smooth, and `false` otherwise.

4.2 Partial Order on Phyla

Recall that `P` determines a partial order on phyla.

`PartialOrder(` *group*`,` *algrec* `)`

computes this partial order, and returns a matrix with a 1 in the `[i][j]` entry if the `i`-th character is $\preceq$ the `j`-th character, and a 0 otherwise. Here, the numbering of characters is given by *algrec*`.label`, and *not* by the intrinsic numbering of characters used in Section 2 for Lusztig–Shoji data. As a result, the output matrix is always lower-triangular.

`PrintPartialOrder(` *group*`,` *algrec* `)`

This function displays the partial-order matrix on the screen in a compact, aesthetically pleasing manner.

4.3 LaTeX Output

`TeXLSP(` *filename*`,` *group*`,` *algrec* `)`
`TeXLSLam(` *filename*`,` *group*`,` *algrec* `)`

These functions produce LaTeX code for tables showing the P and Λ matrices, respectively. The output is saved in the file *filename*, unless *filename* is the empty string, in which case it is printed to the screen.

`TeXPartialOrder(` *filename*`,` *group*`,` *algrec* `)`

This function creates a Hasse diagram using XY-pic. For this code to work properly in a LaTeX file, the file must contain the command `\usepackage[all]{xy}` in the preamble. The algorithm for creating the diagram is not very sophisticated; manual editing is often necessary to improve the layout and eliminate overlapping features.

5 Global Options

Various aspects of the behavior of the software can be controlled by setting certain global variables.

`LS_VERBOSE = false`

If `true`, `LSAlg` will print information on the screen about its progress.

`LS_SAVE_OMEGA = ""`

If this variable is nonempty, it will be interpreted as the name of a directory in which `LSAlg` can save the matrices Ω as it computes them. It will also look in this directory for the results of past calculations. This feature is useful for large groups, since the calculation of Ω is time-consuming but completely independent of Lusztig–Shoji data.

`LS_MATHEMATICA = 0`

The slowest step in the Lusztig–Shoji algorithm is the calculation of the inverses of certain submatrices of Λ. If Mathematica is installed on the user's system, `LSAlg` can use it to calculate these inverses much faster than can be done interally in GAP. Of course, there is a trade-off: simply launching Mathematica takes time. If `LS_MATHEMATICA` is set to a positive integer n, then `LSAlg` will invoke Mathematica to compute inverses of matrices of size at least $n \times n$, but will compute inverses of smaller matrices internally. On the author's system, `LS_MATHEMATICA = 7` gives good results.

```
LS_TEX_SMALL = false
```

If `true`, the tables produced by the LaTeX output will be formatted with as little whitespace as possible.

```
LS_TEX_POLY_INC = false
```

If `true`, polynomials in LaTeX output will be written with terms in order of increasing degree.

```
LS_TEX_POLY_CYC = false
```

If `true`, polynomials in LaTeX output will be written as a product of cyclotomic factors, with the nth cyclotomic polynomial denoted Φ_n. This is useful in cases where the output includes very long polynomials.

6 Further Examples

Example 6.1. The example of Section 2 can be computed by

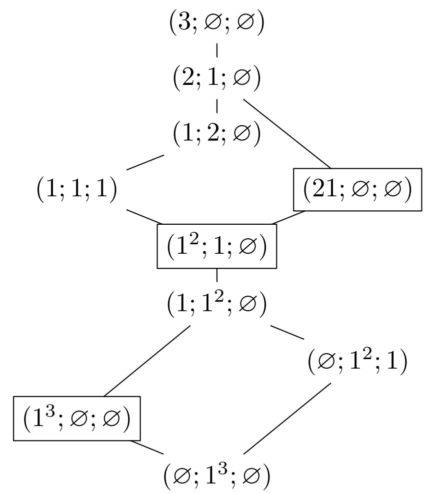
```
W := CoxeterGroup("G",2);
out := SpringerLS(W);;
```

More generally, the local intersection cohomology of the unipotent variety of any algebraic group in good characteristic can be computed by similar means.

Example 6.2. The study of Green functions for imprimitive complex reflection groups was initiated by Shoji in [15]. Example 7.6 of that paper can be computed by

```
W := ComplexReflectionGroup(3,1,3);
out := SymbolLS(W,2,1);;
```

Note that in this example, `IsLSPerverse(W,out)` returns `false`, because $p_{(1^2;\varnothing;1),(1;\varnothing;1^2)} = q^3$, but $q^{a(1^2;\varnothing;1)} = q^4$.



The figure shows the partial order on phyla. (The output of `TeXPartialOrder` has been slightly modified to eliminate overlapping of edges and entries.)

Example 6.3. The various examples considered in [6] can all be studied with commands of the form

```
out := RouquierLS(W);;
```

The polynomials that are associated to special pieces in *loc. cit.* can then be found by examining `out.Lam`.

Example 6.4. According to the main theorem of [3], the local intersection cohomology of the enhanced nilpotent cone is given by

```
W := CoxeterGroup("B",n);
out := TrivialLS(W);;
```

Example 6.5. Finally, the commands

```
W := CoxeterGroup("H",3);
out := TrivialLS(W);;
```

yield a `P` matrix that is tantalizingly close—it differs in one column—from the matrix of graded W-multiplicities in the irreducible tempered representations of the graded Hecke algebra of type H_3, as computed by Kriloff–Ram [9]. Such a coincidence is surely a hint of a deeper, as-yet unknown phenomenon.

References

[1] P. Achar and A.-M. Aubert, *Springer correspondences for dihedral groups*, Transform. Groups **13** (2008), 1–24.

[2] P. Achar and A.-M. Aubert, *Représentations de Springer pour les groupes de réflexions complexes imprimitifs*, J. Algebra **319** (2008), 4102–4139.

[3] P. Achar and A. Henderson, *Orbit closures in the enhanced nilpotent cone*, Adv. Math. **219** (2008), 27–62.

[4] W. Borho and R. MacPherson, *Partial resolutions of nilpotent varieties*, Analysis and topology on singular spaces, II, III (Luminy, 1981), Astérisque, vol. 101, Soc. Math. France, Paris, 1983, pp. 23–74.

[5] M. Broué and S. Kim, *Familles de caractères des algèbres de Hecke cyclotomiques*, Adv. Math. **172** (2002), 53–136.

[6] M. Geck and G. Malle, *On special pieces in the unipotent variety*, Experiment. Math. **8** (1999), 281–290.

[7] S. Kim, *Families of the characters of the cyclotomic Hecke algebras of $G(de, e, r)$*, J. Algebra **289** (2005), 346–364.

[8] H. Kraft and C. Procesi, *A special decomposition of the nilpotent cone of a classical Lie algebra*, Astérisque (1989), no. 173-174, 10, 271–279, Orbites unipotentes et représentations, III.

[9] C. Kriloff and A. Ram, *Noncrystallographic Springer correspondences*, in progress.

[10] G. Lusztig, *A class of irreducible representations of a Weyl group*, Nederl. Akad. Wetensch. Indag. Math. **41** (1979), 323–335.

[11] G. Lusztig, *Green polynomials and singularities of unipotent classes*, Adv. in Math. **42** (1981), 169–178.

[12] G. Malle and R. Rouquier, *Familles de caractères de groupes de réflexions complexes*, Represent. Theory **7** (2003), 610–640.

[13] R. Rouquier, *Familles et blocs d'algèbres de Hecke*, C. R. Acad. Sci. Paris Sér. I Math. **329** (1999), 1037–1042.

[14] T. Shoji, *On the Green polynomials of a Chevalley group of type F_4*, Comm. Algebra **10** (1982), 505–543.

[15] T. Shoji, *Green functions associated to complex reflection groups*, J. Algebra **245** (2001), 650–694.

[16] N. Spaltenstein, *Classes unipotentes et sous-groupes de Borel*, Lecture Notes in Mathematics, vol. 946, Springer-Verlag, Berlin, 1982.