

AN ALGORITHM FOR EVALUATING GRAM MATRICES IN VERMA MODULES OF \mathcal{W} -ALGEBRAS

Daniel Whalen

*SITP, Department of Physics and Theory Group, SLAC,
Stanford University, Stanford, CA 94305, USA*

ABSTRACT. I present a simple dynamic programming algorithm for the evaluation of operations in a wide range of superconformal algebras. Special care is taken to describe the computation of the Gram matrix. A *Mathematica* package, *Weaver*, is provided that implements the algorithm.

1. INTRODUCTION

It is a generally unresolved question of how to determine the structure of highest-weight modules of \mathcal{W} -algebras. I propose an algorithm that will efficiently perform evaluations on a Verma module at low level, enabling a number of explicit calculations. These calculations include evaluating characters of irreducible representations, evaluating Gram matrices, constructing ladder diagrams, and finding primitive and p-singular vectors.

Related algorithms have appeared in previous work, but only for the Virasoro algebra [2]. The algorithms described in this paper are applicable to general \mathcal{W} -algebras.

I provide an implementation of algorithms 1, 2, and 3 as a *Mathematica* package in the attached files. Examples are provided for use of the package with the Virasoro algebra and the $SW(3/2, 2)$ algebra [3, 6].

2. DEFINITIONS

A \mathcal{W} -algebra is an operator algebra that is generated by a finite number of fields and superfields, with explicit commutation relations and anti-commutation relations that are linear combination of single fields, normal ordered products of derivatives of two fields, and the operator describing the central extension of a Virasoro subalgebra¹. We will construct an action of the \mathcal{W} -algebra on Verma modules of a finite-dimensional highest-weight space.

For a given \mathcal{W} -algebra, write the simple fields X^i , for $i = 1, \dots, d$ along with their conformal weights Δ_i . Write the parity F^i of the field X^i , which is 0 if X^i is bosonic, and is 1 if X^i is fermionic. If X^i is bosonic, we write the corresponding set of operators X_n^i for $n \in \mathbf{Z}$. If X^i is fermionic, we choose for each i , $n \in \mathbf{Z}$ (R sector) or $n \in \mathbf{Z} + \frac{1}{2}$ (NS sector).

The commutation and anti commutation relations can be written explicitly. Let $[X_n^i, X_m^j] = X_n^i X_m^j - (-1)^{F^i F^j} X_m^j X_n^i$ and express the relations as

$$[X_n^i, X_m^j] = e^{ij}(n)\delta_{n+m} + \sum_{k=1}^d f_k^{ij}(n, m)X_{n+m}^k + \sum_{b,c=1}^d g_{bc;xy}^{ij}(n, m):\partial^x X^b \partial^y X^c:_{n+m}.$$

¹See, for example, [1] for a detailed discussion of \mathcal{W} -algebras.

The normal-ordered product is given in terms of $\partial^x X_n^i = x! \binom{-n-\Delta^i}{x} X_n^i$ by

$$:\partial^x X^i \partial^y X^j:_n = \sum_{k \leq -\Delta_i - x} \partial^x X_k^i \partial^y X_{n-k}^j + (-1)^{F^i F^j} \sum_{k > -\Delta_i - x} \partial^y X_{n-k}^j \partial^x X_n^i.$$

The presence of normal-ordered products can be problematic for computational purposes. Two additional assumptions can be made regarding which g -functions are nontrivial in order for the relations to resolve nicely. Construct the directed graph with a node for every unordered pair of operators $\{X^i, X^j\}$ and an edge from $\{X^i, X^j\}$ to $\{X^b, X^c\}$ if $g_{bc;xy}^{ij}(n, m)$ is not uniformly zero for all n, m, x , and y . We first require that this graph be acyclic. We second require that if there exists a path from $\{X^i, X^j\}$ to $\{X^a, X^b\}$, we have $\max(i, j) > \min(a, b)$. These requirements are sufficient, but not necessary conditions for the algorithm to work. The requirements will, however, apply to most algebras of interest up to a relabeling of the indices.

We are interested in constructing highest-weight representations of the \mathcal{W} -algebra. Start with vectors $|v\rangle$ with $X_n^i |v\rangle = 0$ whenever $i > 0$. The Verma module M generated by the $|v\rangle$ s is the span of vectors of the form $X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle$. Define a grading on the space such that one of the basis elements above has level $l = -\sum_{k=1}^r n_k$ and write the subspace of M at level l as M_l . Any operator X_n^i acts as a linear operator on each level: $X_n^i |M_p\rangle = X_n^i |p\rangle : M_p \rightarrow M_{p-n}$.

M_p has a basis that consists of states of the form $X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle$ with the operators in the following order: the n_i s are in ascending order, with $i_{j+1} \geq i_j$ if $n_{i+1} = n_i$, and such that two identical fermionic operators never appear. We call such a state properly ordered.

The \mathcal{W} -algebra has a 0 subalgebra that consists of operators of the form X_0^i . Since $X_0^i |M_0\rangle \rightarrow M_0$, M_0 forms a representation of the 0-algebra. Denote this algebra by ρ and write $X_0^i |v\rangle = |\rho(X^i) \cdot v\rangle$.

We can construct a Hermitian conjugate on M . On the highest weight space, we write $|v\rangle^\dagger = \langle v|$. In general, we define $X_n^i{}^\dagger = X_{-n}^i$, although this may vary based on the construction of the \mathcal{W} -algebra. Unitarity permits us to define an inner product on M . Construct an inner product on M_0 that is unitary under the action of the 0-algebra. The inner product can be extended to higher levels through the definition $\langle \vec{v}, \vec{w} \rangle = \vec{v}^\dagger \vec{w}$.

Observe that two vectors at different levels are necessarily orthogonal. Let $(\vec{v}_1, \dots, \vec{v}_r)$ be a basis for M_l . We can therefore describe the inner-product completely by constructing the Gram matrix at level l , $G(l)$ where $G(l)_{ij} = \langle \vec{v}_i, \vec{v}_j \rangle$.

3. MATRIX CONSTRUCTIONS

For computational purposes, we will need to construct matrix representations of the above operators. For a given level l , enumerate a properly ordered basis of M_l as described above. Write the dimensions as $\dim M_l = P(l)$. Given a properly ordered state, $\vec{w} = X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle$ at level l , write the $P(l)$ -dimensional unit vector corresponding to \vec{w} as $\vec{w} = \overline{X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle}$.

The operators $X_n^i |l\rangle$ can each be expressed as a $P(l-n) \times P(l)$ matrix acting on the left of M_l . Denote this matrix by $\overline{X_n^i |l}$. Given a normal ordered product, $:X_n^i X_m^j:$, we express its action on M_l as $:X_n^i X_m^j : |l\rangle$. Note that when applied to a state at finite level, only a finite number of terms in the infinite sum defining the normal-ordered product survive, so this matrix is well-defined. We also use the following shorthand:

$$\overline{[X_n^i, X_m^j] |l} = e(n) \delta_{n+m} I + \sum_{k=1}^d f_k^{ij}(n, m) \overline{X_{n+m}^k |l} + \sum_{b,c=1}^d g_{bc;xy}^{ij}(n, m) : \partial^x X^b \partial^y X^c :_{n+m} |l,$$

where it should be assumed that the expression on the left is replaced by the expression on the right whenever it appears in this paper. Similarly, define as shorthand,

$$\overline{((X_n^i X_m^j))|l} = \begin{cases} \overline{X_n^i|l-m} \cdot \overline{X_m^j|l} & \text{if } n < m \text{ or } n = m \text{ and } i < j \\ \overline{[X_n^i X_m^j]|l} + (-1)^{F_i F_j} \overline{X_m^j|l-n} \cdot \overline{X_n^i|n} & \text{otherwise.} \end{cases}$$

The expressions $\overline{((X_n^i X_m^j))|l}$ and $\overline{X_n^i|l-m} \cdot \overline{X_m^j|l}$ are the same as operators, but may differ in the order of evaluation.

Given a vector $\vec{v} \in M_l$, we can express \vec{v}^\dagger as a $P(l)$ -dimensional row-vector \vec{v}^\dagger such that for $\vec{w} \in M_l$, $\vec{v}^\dagger \vec{w} = \vec{v}^\dagger \cdot \vec{w}$. The unitary constraint then states that that

$$\left(\overline{X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v} \right)^\dagger = \left(\overline{X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v} \right)^\dagger \cdot \overline{X_{n_1}^{i_1} | l}$$

The Gram matrix has a simple expression in this formalism. The row corresponding to \vec{v} is given in entirety by \vec{v}^\dagger .

Algorithms for evaluating the $\overline{X_n^i|l}$, the $\overline{X_n^i X_m^j|l}$ and the Gram matrices are given in Algorithms 1, 2, and 3 respectively.

Algorithm 1 Evaluation of $\overline{X_n^i|l}$

```

function OPERATOR( $i, n, l$ ) =  $\overline{X_n^i|l}$ 
  if OPERATOR( $i, n, l$ ) has previously been evaluated then
    return OPERATOR( $i, n, l$ )
  end if
   $N \leftarrow \{\}$ 
  for  $X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v$  in the basis of  $M_l$  do
     $row \leftarrow \text{OPATSTATE}(i, n, l, X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v)$ 
     $N \leftarrow \text{APPEND}(N, row)$ 
  end for
  return  $N$ 
end function

function OPATSTATE( $i, n, l, X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v$ ) =  $\overline{X_n^i|l} \cdot \overline{X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v}$ 
  if  $l = 0$  and  $n = 0$  then
    return  $|\rho(X_0^i) \cdot v\rangle$ 
  end if
  if  $X_n^i X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v$  is properly ordered then
    return  $\overline{X_n^i X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v}$ 
  end if
  if  $i = i_1$  and  $n = n_1$  and  $X^i$  is fermionic then
    return  $\frac{1}{2} [\overline{X_n^i}, \overline{X_{n_1}^{i_1}}]_{l+n} \cdot \overline{X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v}$ 
  end if
  if  $n > l + i_1$  then
     $a \leftarrow \overline{[X_n^i, X_{n_1}^{i_1}]_{l+n_1} \cdot X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v}$ 
    return  $a$ 
  else
     $a \leftarrow \overline{X_n^i X_{n_1}^{i_1}}_{l+n_1} \cdot \overline{X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v}$ 
     $b \leftarrow (-1)^{F_i F_{i_1}} \overline{X_{n_1}^{i_1}}_{l+n_1-n} \cdot \overline{X_n^i}_{l+n_1} \cdot \overline{X_{n_2}^{i_2} \dots X_{n_r}^{i_r} | v}$ 
    return  $a + b$ 
  end if
end function

```

Algorithm 2 Evaluation of $\overline{\partial^x X^i \partial^y X^j} \cdot_n |l\rangle$

```

function NOP( $i, j, x, y, n, l$ ) =  $\overline{\partial^x X^i \partial^y X^j} \cdot_n |l\rangle$ 
   $a \leftarrow \sum_{n-l \leq p \leq -\Delta_i} x!y! \binom{-p-\Delta_i}{x} \binom{-n+p-\Delta_j}{y} ((X_p^i X_{n-p}^j))|l\rangle$ 
   $b \leftarrow \sum_{-\Delta_i < p \leq l} x!y! \binom{-p-\Delta_i}{x} \binom{-n+p-\Delta_j}{y} ((X_{n-p}^j X_p^i))|l\rangle$ 
  return  $a + b$ 
end function

```

Algorithm 3 Evaluation of Gram Matrices

```

function GRAMMATRIX( $l$ )
  if  $l=0$  then
    for  $i, j \in \{1, \dots, P(0)\}$  do
       $G_{ij} \leftarrow \langle v_i | v_j \rangle$ 
    end for
    return  $G$ 
  end if
  for  $X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle$  in the basis of  $M_l$  do
     $Row \leftarrow \overline{X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle} \cdot G(l + n_1) \cdot \overline{X_{n_1}^{i_1} |l\rangle}$ 
     $G \leftarrow \text{APPEND}(N, Row)$ 
  end for
  return  $G$ 
end function

```

4. CONSISTENCY OF THE ALGORITHM

The approach of the algorithm is straightforward. OPERATOR computes $\overline{X_n^i} |l\rangle$ by using OPATSTATE to act on a basis for M_l . If $X_n^i X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle$ is properly ordered, then we can look up its value. Otherwise, we need to commute the X_j^i through $X_{n_1}^{i_1}$ and compute in terms of the other operators recursively.

We demonstrate through induction over l and i that the recursion terminates. For the moment, assume that normal ordered operators do not appear: that all the $g_{bc;xy}^{ij}$ s are zero. I delay extending the proof to normal ordered operators until the next section.

For the base case at $l = 0$, OPATSTATE($i, n, 0, |v\rangle$) returns an explicit expression line by line for each generator $|v\rangle$ of M_0 . In particular,

$$\text{OPATSTATE}(i, n, 0, |v\rangle) = \begin{cases} \overline{|\rho(X_0^i) \cdot v\rangle} & \text{if } n = 0 \\ \overline{X_j^i |v\rangle} & \text{if } n < 0 \end{cases}$$

Now proceed with the induction. Assume inductively that $\overline{X_n^j} |p\rangle$ has been evaluated for all $p < l$ and for all $p = l, j < i$. We wish to show that $\overline{X_n^i} |l\rangle$ can then be evaluated. Consider the evaluation of $\text{OPATSTATE}(i, n, l, X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle)$. There are only two cases that occur. If $X_n^i X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle$ is properly ordered, the function immediately evaluates. In the other case, we have $n_1 < n$ or $n_1 = n$ and $i_1 < i$ or if X^i is fermionic, we may have $n_1 = n$ and $i_1 = i$. In these cases, OPATSTATE calls three operators, $\overline{[X_n^i, X_{n_1}^{i_1}]_{l+n_1}}$, $\overline{X_{n_1}^{i_1} |_{l+n_1-n}}$, and $\overline{X_n^i |_{l+n_1}}$, all of which have already been evaluated by the assumption.

5. CONSISTENCY OF THE ALGORITHM WITH NORMAL ORDERED OPERATORS

Recall that a normal ordered product is defined in terms of the infinite sum

$$:\partial^x X^i \partial^y X^j:{}_n = \sum_{k \leq -\Delta_i - x} \partial^x X_k^i \partial^y X_{n-k}^j + (-1)^{F^i F^j} \sum_{k > -\Delta_i - x} \partial^y X_{n-k}^j \partial^x X_n^i,$$

of which only a finite number of terms survive when acting on a given level. This expression is evaluated in NOP as

$$\begin{aligned} \overline{:\partial^x X^i \partial^y X^j:{}_n|_l} &= \sum_{n-l \leq p \leq -\Delta_i} x!y! \binom{-p - \Delta_i}{x} \binom{-n + p - \Delta_j}{y} \overline{((X_p^i X_{n-p}^j))|_l} \\ &\quad + \sum_{-\Delta_i < p \leq l} x!y! \binom{-p - \Delta_i}{x} \binom{-n + p - \Delta_j}{y} \overline{((X_{n-p}^j X_p^i))|_l} \end{aligned}$$

Return to the claim in the last chapter that the recursion in OPERATOR terminates. I wish to extend the result to instances where the normal ordered products appear. Assume, as before, that we have calculated all the operators $X_n^j|_p$ for $p < l$ and for $p = l, j < i$. We wish to then evaluate OPATSTATE for $\overline{X_n^i|_l}$ acting on $\overline{X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r}|_v}$.

Recall the assumptions that we made regarding the $g_{bc;xy}^{ij}(n, m)$. We constructed the directed graph with an edge from $\{X^i, X^j\}$ to $\{X^b, X^c\}$ if $g_{bc;xy}^{ij}(n, m)$ is not uniformly zero for all n, m, x , and y . We assumed that this graph is acyclic and that paths may only exist from $\{X^i, X^j\}$ to $\{X^a, X^b\}$ if $\max(i, j) > \min(a, b)$.

OPATSTATE only introduces a normal-ordered product if X_n^i and $X_{n_1}^{i_1}$ need to be commuted, when $n > n_1$ or $n = n_1$ and $i > i_1$. In this case, a normal ordered product will be introduced that depends on terms of the form $\overline{((X_n^j X_{n_1}^j))|_l}$. We can expand out the commutator in that term which may give a normal-ordered product. Iteratively repeat this process. The iteration terminates because of the acyclic condition we imposed on the normal-ordered dependency graph.

When we are done, we are left with terms that involve operators of the form $\overline{X_{n_1+n}^k|_{l+n_1}}$ and possibly of the form $\overline{X_{n+n_1-p}^a|_{l+n_1-p} \cdot X_p^b|_{l+n_1}}$ if there is a path from (n, n_1) to (a, b) in the normal-ordered dependency graph and if $p > n + n_1 - p$ or $p = n + n_1 - p$ and $b > a$. Operators of the first form have all been evaluated by the induction hypothesis. Operators of the second form can only fail to have been evaluated if $p = n_1 = n = n + n_1 - p$, $a < b$ and if $a \geq n$. Since $a = \min(a, b)$ and $n = \max(n, n_1)$, this is impossible by our second condition on the normal-ordered dependency graph. Thus, the recursion in the algorithm terminates.

6. COMPUTATIONAL COMPLEXITY

Asymptotics of $P(l)$. The number of states at a given level l has an asymptotic expression, which is a slight generalization of the asymptotics of the partition function. Let N_B be the number of bosonic operators, N_F be the number of fermionic operators, $\alpha = \pi\sqrt{N_F + 2N_B}/\sqrt{3}$, and $\beta = -(N_B + 3)/4$. Then the number of states at level l is [4, 5]

$$P(l) = \dim M_l = O\left(l^\beta e^{\alpha\sqrt{l}}\right).$$

We will make frequent use of the asymptotic form of sums over the partition functions, $\sum_{p=0}^l p^\beta e^{\alpha\sqrt{p}} = \frac{2}{\alpha} l^{\beta+\frac{1}{2}} e^{\alpha\sqrt{l}} + O(l^\beta e^{\alpha\sqrt{l}})$.

Matrix multiplication. Multiplication of two $n \times n$ square matrices can be performed in $O(n^\theta)$ time for $2 \leq \theta < 3$. The best known algorithms give $\theta \approx 2.373$, although in practice, textbook matrix multiplication with $\theta = 3$ or the Strassen algorithm with $\theta \approx 2.81$ will be employed.

All of these algorithms can be applied to rectangular matrices by subdividing the matrices into blocks and performing square matrix multiplication on the subblocks. Doing so allows for the multiplication of an $l \times m$ and an $m \times n$ matrix to be performed in $O(lmn \cdot \min(l, m, n)^{\theta-3})$ time.

Operators. Consider the problem of evaluating all of the operators $\overline{X_n^i|_p}$ such that $0 \leq p \leq l$, and $-l \leq n \leq l$ subject to the constraint that $0 \leq p - n \leq l$. For calculating a single operator $\overline{X_n^i|_p}$, assume that the matrix expressions for lower level operators and normal ordered operators have been evaluated. The evaluation of the normal-ordered products will be treated in the next subsection.

The slowest step in OPATSTATE is the evaluation of b , since all of the other operators are multiplication of a matrix by a unit vector or the lookup of the unit vector corresponding to a state, both of which can be performed in slightly over $O(P(l))$ time. Both cases have an upper bound of time $P(p)P(p - m)$, so the net complexity of the operation is $P(p)^2P(p - n)$. Summing over p and n , the total time is $lP(l)^3$.

There is a modification to the algorithm that decreases the complexity. Make the following modification to Algorithm 1. Instead of evaluating the $P(p)$ rows separately as in OPERATOR, we evaluate all of the rows $X_{n_1}^{i_1} X_{n_2}^{i_2} \dots X_{n_r}^{i_r} |v\rangle$ that share the same $X_{n_1}^{i_1}$ together. For a fixed $X_{n_1}^{i_1}$, write $N(X_{n_1}^{i_1})$ for the submatrix of N from OPERATOR whose rows correspond with the states starting with $X_{n_1}^{i_1}$. When calculating b in OPATSTATE, multiple vector multiplications may then be turned into a single matrix multiplication. Estimating the runtime of this modified algorithm is beyond the scope of this paper, but I believe that the limit is asymptotically in $O(\text{polynomial}(l)P(l)^\theta)$.

Normal Ordered Operators. In order to evaluate the normal ordered operators $\overline{X^i X^j :_n|_p}$, the normal ordered product needs to be expanded into a sum of products of operators from levels p to k and from k to $p - n$ as k varies from 0 to $p - \frac{n}{2}$. The products can be calculated in $O(P(p)P(k)P(p - n)P(\min(p, k, p - n))^{\theta-3})$ time. Summing over k, p and n , the total time is $l^{\frac{3}{2}}P(l)^\theta$.

Gram Matrices. Having evaluated all of the operators and normal ordered products of operators, we can evaluate the Gram matrix. The Gram matrix has $P(l)$ rows, each of which can be evaluated recursively from a previous Gram matrix and a operator of size less than $P(l) \times P(l)$. Evaluating the Gram matrices for level 0 through l can therefore be performed in $O(\sqrt{l}P(l)^3)$ once the operators have been computed.

The vector multiplications can be merged in the evaluation of $G(l)$ in Algorithm 3 similarly to the evaluation of OPERATOR. Doing so will reduce the complexity of the Gram matrix evaluations to be less than that for evaluating OPERATOR.

Total Complexity. For very large n , the above calculations provide an upper bound for the run time that is dominated by the evaluation of operators, $lP(l)^3$. In practice, however, $\theta \approx 3$ and the total run time for the algorithm is dominated by the evaluation of the normal-ordered products, which takes $O(l^{\frac{3}{2}}P(l)^\theta)$ time.

REFERENCES

- [1] P Bouwknegt, Jim McCarthy, and Krzysztof Pilch. *The W3 Algebra. modules, semi-infinite cohomology and BV algebras*. Springer Science & Business Media, January 1996.

- [2] G Feverati and P A Pearce. Critical RSOS and minimal models I: Paths, fermionic algebras and Virasoro modules. *arXiv.org*, 2002.
- [3] Doron Gepner and Boris Noyvert. Unitary representations of $SW(3/2,2)$ superconformal algebra. *arXiv.org*, January 2001.
- [4] J L Nicolas and A Sárközy. On the asymptotic behaviour of general partition functions. *Ramanujan Journal. An International Journal Devoted to the Areas of Mathematics Influenced by Ramanujan*, 4(1):29–39, 2000.
- [5] Jean-Louis Nicolas and András Sárközy. On the asymptotic behaviour of general partition functions. II. *Ramanujan Journal. An International Journal Devoted to the Areas of Mathematics Influenced by Ramanujan*, 7(1-3):279–298, 2003.
- [6] S L Shatashvili and C Vafa. Superstrings and Manifolds of Exceptional Holonomy. *arXiv.org*, July 1994.