CERN-TH.6032/91
ITFA - 91 - 05

Fusion rule engineering

Jürgen Fuchs

Theory Division
CERN
CH - 1211 Genève 23

and

Peter Van Driel

Instituut voor Theoretische Fysica
Valckenierstraat 65
NL - 1018 XE Amsterdam

Abstract

The determination of the fusion rules of WZW theories constitutes an important step in the program of classifying and solving conformal field theories. A drawback of the existing algorithms for such calculations is that already at a modestly large number of primary fields they require an enormous amount of computer time and memory. However, by breeding together two of these algorithms in an appropriate manner it is possible to speed up the calculations and therefore extend the accessible range of theories very much.

CERN-TH.6032/91
March 1991

/

1 Introduction

Wess-Zumino-Witten (WZW) theories [1, 2] seem to be the building blocks of all (rational) two-dimensional conformal field theories. An important step in the program of classifying and solving conformal field theories is therefore the determination of the fusion rule algebras of WZW theories, i.e. the structure constants $\mathcal{N}_{\Lambda\Lambda'}^{\Lambda''}$ in the fusion rules

$$\phi_{\Lambda} \star \phi_{\Lambda'} = \sum_{\Lambda''} \mathcal{N}_{\Lambda\Lambda'}^{\Lambda''} \phi_{\Lambda''}. \quad (1)$$

Here $\{\phi_{\Lambda}\}$ denotes the set of chiral primary fields of the WZW theory based on some given untwisted affine Lie algebra g . The WZW primaries carry integrable irreducible highest weight modules of g , with a fixed value of the level k of g , and hence can be labelled by the highest weights $\bar{\Lambda}$ of the finite-dimensional modules $L_{\bar{\Lambda}}$ of the horizontal subalgebra $\bar{g}$ of g .

The problem of the actual construction of the fusion rules of the WZW model is by now a well-settled issue. The first algorithm for the computation of fusion rules was the so-called depth rule, introduced in [2]. Unfortunately, with this method the calculation of the fusion rule (1) requires information about the weight systems of both $L_{\bar{\Lambda}}$ and $L_{\bar{\Lambda}'}$, and also about the Clebsch-Gordan coefficients of the tensor product $L_{\bar{\Lambda}} \times L_{\bar{\Lambda}'}$; as a consequence, the depth rule does not provide a satisfactory tool to do actual computations in theories with more than a few primaries. A further algorithm became available when the relation between the fusion rules of a conformal field theory and its behaviour under modular transformations was uncovered [3, 4]: given the matrix S that implements the modular transformation $\tau \mapsto -1/\tau$ on the characters of the theory, the fusion rules can be constructed from the Verlinde formula [3]

$$\mathcal{N}_{\Lambda\Lambda'}^{\Lambda''} = \sum_{\mu \in P_k} \frac{S_{\Lambda\mu} S_{\Lambda'\mu} S_{\mu\Lambda''}^{-1}}{S_{k\Lambda(0),\mu}}. \quad (2)$$

Here the sum is over all integer weights in the closure of the fundamental affine Weyl chamber of g at level k . General expressions for the modular transformation matrix S are known [5]. These expressions still include however a sum over the Weyl group of $\bar{g}$ (which e.g. for E_8 has order 696 729 600) so that for large $\bar{g}$ explicit calculations become problematic.

More recently, an algorithm was found [6–8] which essentially reduces the calculation of the fusion rule (1) to a sum over the distinct weights of either $L_{\bar{\Lambda}}$ or $L_{\bar{\Lambda}'}$. With this algorithm, the fusion rules of fields carrying low-dimensional modules $L_{\bar{\Lambda}}$ are very easily calculable. However, just like the methods mentioned before, this algorithm becomes cumbersome when modules $L_{\bar{\Lambda}}$ with large dimension are involved.

In short, all existing algorithms for the calculation of WZW fusion rules require a substantial amount of computer time and memory. In practice, this means that they cannot be used for the actual computation of fusion rules for WZW theories with more than a few primary fields unless the algebra $\bar{g}$ is small; for instance, for $\bar{g} = E_8$ already the level 4 case (10 fields) is challenging. In this note we point out that it is possible to combine the best aspects of the two last-mentioned algorithms to get a new algorithm by which one computes both the S matrix and the fusion rules. This algorithm turns out to be much faster than the other ones (and needs only a small amount of memory). As a consequence, one can analyse theories with considerably larger number n_{pf} of primary fields, certainly up to n_{pf} of the order of a few hundred. Of course for still larger n_{pf} there arises again a computational bottle-neck, and it seems unlikely that it will ever be possible to determine the fusion rules of such theories explicitly.

2 The algorithm

The spectrum of the WZW model in the classical limit (i.e. for $k \rightarrow \infty$) contains the fields ϕ_Λ corresponding to all finite-dimensional modules $L_{\bar{\Lambda}}$ of g . In this limit, the fusion rules are just the ordinary tensor product decomposition rules and hence can be calculated using the formula (see e.g. [9])

$$(\mathcal{N}_{\Lambda\Lambda'}^{\Lambda''})_{k \rightarrow \infty} = \sum_{\bar{w} \in \bar{W}} \bar{\sigma}(\bar{w}) \text{mult}_{\bar{\Lambda}}(\bar{w}(\bar{\Lambda}'' + \bar{\rho}) - \bar{\rho} - \bar{\Lambda}'). \quad (3)$$

Here $\text{mult}_{\bar{\Lambda}}(\bar{\lambda})$ denotes the multiplicity of the weight $\bar{\lambda}$ in the module $L_{\bar{\Lambda}}$, $\bar{\rho}$ is the Weyl vector (i.e., half the sum of positive roots) of $\bar{g}$, the sum is over the Weyl group $\bar{W}$ of $\bar{g}$, and $\bar{\sigma}(\bar{w})$ is the sign of the Weyl reflection $\bar{w}$. For finite level, the allowed range of $\bar{\Lambda}$ is truncated to (the horizontal projection of) the fundamental affine Weyl chamber, and accordingly the fusion rule coefficients are given by

$$\mathcal{N}_{\Lambda\Lambda'}^{\Lambda''} = \sum_{w \in W_{\text{hor}}} \sigma(w) \text{mult}_{\bar{\Lambda}}(w(\bar{\Lambda}'' + \bar{\rho}) - \bar{\rho} - \bar{\Lambda}'), \quad (4)$$

where the sum now extends over the horizontal projection W_{hor} of the affine Weyl group W of g . Note that while W_{hor} is an infinite group, it is finitely generated, and only a finite number of terms contributes in (4), thus leading to a finite algorithm. (For further information about the formula (4), see [6–8]; detailed proofs are given in [7].)

Being interested in all fusion rules at a given level, we would have to calculate the weight systems of $L_{\bar{\Lambda}}$ for all primary fields ϕ_Λ . This becomes quite cumbersome if large-dimensional modules $L_{\bar{\Lambda}}$ are involved. Therefore our idea is to do the following. First we apply the prescription (4) to just one field

ϕ_Λ whose fusion rules are non-trivial.¹ In practice, one of course just uses the lowest-dimensional non-trivial module, e.g. the defining module of A_r , the adjoint of E_8 , etc. Next one reads off from the Verlinde formula (2) that the fusion matrix $\mathcal{N}_\Lambda$ (i.e. the matrix with entries $(\mathcal{N}_\Lambda)_{\Lambda'}^{\Lambda''} = \mathcal{N}_{\Lambda\Lambda'}^{\Lambda''}$) is diagonalized by the matrix S . This property of S can thus be employed to find S using $\mathcal{N}_\Lambda$ as an input. Having determined S we can then simply compute the fusion rules from the Verlinde formula.

The general validity of this prescription depends on the uniqueness of the matrix that diagonalizes the fusion rules. Now of course, without further specifications the matrix $\tilde{S}$ that diagonalizes a given matrix $\mathcal{N}_\Lambda$ is by no means unique. However, the matrix we are looking for must also possess the following properties [5, 3, 4]: it has to be symmetric, $S = S^t$, must square to the charge conjugation matrix, $S^2 = C$, and there must exist a column with all entries real and positive, corresponding to the identity primary field that carries the highest weight $k\Lambda_{(0)}$ of g , i.e. $S_{\Lambda, k\Lambda_{(0)}} \geq S_{k\Lambda_{(0)}, k\Lambda_{(0)}} > 0$. It has been proven in [12] that for unitary theories the matrix with all the aforementioned properties is unique up to possibly a permutation of its columns that commutes with the matrix, i.e. if S is one such matrix, then any other matrix with the required properties is given by $\tilde{S} = S\Pi = \Pi S$ with Π a permutation.

This implies in particular that there is no essential ambiguity in the identification of the columns of S with the primary fields of the theory. Namely, any permutation $\tilde{\Pi}$ of the primaries is equivalent to a simultaneous permutation $S \mapsto \tilde{S} = \tilde{\Pi}^{-1}S\tilde{\Pi}$ of the rows and columns of S and to an analogous change in the diagonal matrix T ($T_{\Lambda\Lambda'} = \delta_{\Lambda\Lambda'} \exp(2\pi i[(\bar{\Lambda}, \bar{\Lambda} + 2\bar{\rho}) - k \dim(\bar{g})]/(k+h))$ with h the dual Coxeter number of $\bar{g}$) that implements the modular transformation $\tau \mapsto \tau + 1$ on the characters. But the ordering of the diagonal entries of T can be fixed by imposing the defining relation $(ST)^3 = C$ of the modular group, up to permutations that commute with T , and this in turn fixes the permutation Π defined by $\tilde{S} = S\Pi$ up to the latter ambiguity. In other words, the matrix S that obeys the aforementioned restrictions as well as $(ST)^3 = C$ is unique up to permutations Π of its columns that commute both with S and with T . But the latter non-uniqueness is quite irrelevant from the conformal field theory point of view, since $[S, \Pi] = 0 = [T, \Pi]$ implies that Π is an automorphism of the fusion rules (which induces a non-diagonal modular invariant partition function) so that both S and $\tilde{S} = S\Pi$ are sensible S matrices.

3 The computer program

In practice it can be quite involved to implement the idea presented in the previous section on the computer. Therefore a few technicalities will be men-

¹This excludes, besides the identity field, also the simple currents [10, 11].

tioned. After the construction of the fusion matrix $\mathcal{N}_\Lambda$ corresponding to the lowest-dimensional non-trivial module $L_{\bar{\Lambda}}$ of $\bar{g}$ one can use standard routines [13] for the diagonalization of this matrix. These routines yield a set of orthonormal eigenvectors. Of course the order of these eigenvectors is random, so they need to be ordered. Furthermore, the matrix $\mathcal{N}_\Lambda$ can and typically does have degenerate eigenvalues, which means that we must rotate the corresponding eigenvectors correctly.

To implement this, the first thing one can do is to identify the column corresponding to the identity primary field $\phi_{k\Lambda_{(0)}}$ which has all entries real and positive. Even this can be problematic if this column corresponds to a degenerate eigenvalue, but fortunately such a degeneracy can be resolved by using the fact that $S_{\Lambda, k\Lambda_{(0)}} = D_\Lambda S_{\Lambda_{(0)}, \Lambda_{(0)}}$ and $S_{\Lambda_{(0)}, \Lambda_{(0)}} = (\sum_{\Lambda \in P_k} (D_\Lambda)^2)^{-1/2}$, where D_Λ are the so-called quantum dimensions of the fields ϕ_Λ , and that the quantum dimensions can be determined by an independent argument, namely by the quantum Weyl formula [5, 14]

$$D_\Lambda = \prod_{\bar{\alpha} > 0} \left[\sin\left(\frac{\pi(\bar{\Lambda} + \bar{\rho}, \bar{\alpha})}{k+g}\right) / \sin\left(\frac{\pi(\bar{\rho}, \bar{\alpha})}{k+g}\right) \right], \quad (5)$$

where the sum is over all positive roots of $\bar{g}$.

After the identification of the column corresponding to the identity, we impose $S = S^t$ and $S^2 = C$, thereby successively reordering columns as well as rotating columns corresponding to degenerate eigenvalues. Finally, any remaining ordering problems are resolved, up to fusion rule automorphisms, by imposing $(ST)^3 = C$.

We have implemented all these manipulations into a Fortran code. The computational limitations turn out to be quite mild: asymptotically, the time needed for the computation of $\mathcal{N}_\Lambda$ grows as n_{pf} , the time for the computation of S grows as $(n_{\text{pf}})^3$, and the time needed for all fusion rules as $(n_{\text{pf}})^4$; for n_{pf} at the order of 100 the total time consumption is of the order of 1 minute.² Thus we can conclude that with our program one can reach theories with n_{pf} of the order of at least several hundred.

It must also be remarked that the Verlinde formula (2) expresses the integers $\mathcal{N}_{\Lambda\Lambda'}^{\Lambda''}$ as sums over complex numbers, which means that the algorithm gives us the fusion rule coefficients as integers plus some non-vanishing rest R . Again this places only very mild restrictions on the applicability of the method: inspection shows that, using double precision variables, the rest R roughly behaves as $R \simeq (n_{\text{pf}})^2 \cdot 10^{-14}$.

²The computations have been performed on CERNVM. In the range cited, one is not yet in the asymptotic region: the calculation of S still requires about 10% of the time at $n_{\text{pf}} \simeq 100$.

4 Examples

As examples, we list some results for the fusion rules of G_2 at levels 1 to 20, and of E_8 at levels 1 to 10, and present for one special case the explicit form of the S matrix. The latter example is given by E_7 at level 3. In this case the S matrix has only a few distinct entries, and it is essential to impose $(ST)^3 = \mathbf{1}$ in order to fix S uniquely; the result reads

$$\left(\begin{array}{cccccccccccc|c} a_1 & a_1 & b & b & b & b & b & b & a_2 & a_2 & c & c & (0000000) \\ a_1 & -a_1 & b & b & b & -b & -b & -b & -a_2 & a_2 & c & -c & (0000030) \\ b & b & d_1 & d_2 & d_3 & d_3 & d_2 & d_1 & -b & -b & 0 & 0 & (0000020) \\ b & b & d_2 & d_3 & d_1 & d_1 & d_3 & d_2 & -b & -b & 0 & 0 & (1000000) \\ b & b & d_3 & d_1 & d_2 & d_2 & d_1 & d_3 & -b & -b & 0 & 0 & (0100000) \\ b & -b & d_3 & d_1 & d_2 & -d_2 & -d_1 & -d_3 & b & -b & 0 & 0 & (0001000) \\ b & -b & d_2 & d_3 & d_1 & -d_1 & -d_3 & -d_2 & b & -b & 0 & 0 & (0000110) \\ b & -b & d_1 & d_2 & d_3 & -d_3 & -d_2 & -d_1 & b & -b & 0 & 0 & (0000010) \\ a_2 & -a_2 & -b & -b & -b & b & b & b & -a_1 & a_1 & c & -c & (0000001) \\ a_2 & a_2 & -b & -b & -b & -b & -b & -b & a_1 & a_1 & c & c & (0000011) \\ c & c & 0 & 0 & 0 & 0 & 0 & 0 & c & c & -c & -c & (0000100) \\ c & -c & 0 & 0 & 0 & 0 & 0 & 0 & -c & c & -c & c & (1000010) \end{array} \right) \quad (6)$$

where $a_{1,2} = (7\sqrt{6} \mp 3\sqrt{14})/84$, $b = 1/\sqrt{14}$, $c = 1/\sqrt{6}$, and where $d_{1,2,3} = (\tilde{d}_{1,2,3} - 1)/3\sqrt{14}$ with $\tilde{d}_{1,2,3}$ ($|\tilde{d}_1| < |\tilde{d}_2| < |\tilde{d}_3|$) the three solutions of the cubic equation $\tilde{d}^3 - 21\tilde{d} + 7 = 0$; the ordering of primary fields ϕ_Λ is indicated in the last column which supplies the highest weights $\bar{\Lambda}$ in the Dynkin basis.

Next let us present some fusion rules. Clearly, involving a rapidly increasing number of primary fields we cannot display the fusion rules for large levels in much detail. Instead, just to give some flavour of the results, we give a list of the numbers ν_m of fusion rule coefficients equal to m for $m = 0, 1, 2, 3$, as well as the maximal coefficient $\mathcal{N}_{\max}$ occuring at a given level. These data are provided in tables 1 and 2. As a further information, we note that the largest fusion rule coefficient $\mathcal{N}_{\max} = 8628$ of E_8 at level 10 is $\mathcal{N}_{\Lambda\Lambda}^{\bar{\Lambda}}$ for $\bar{\Lambda} = (10010010)$ (corresponding to $\dim(L_{\bar{\Lambda}}) = 26\,125\,438\,976\,000$).

5 Applications

The most obvious application of our algorithm is the output of the computer program, insofar as one is interested in the exact expressions for the S matrix or the fusion rules. Also, due to the isomorphism between WZW fusion rules and the truncated tensor products of quantum groups [14, 15], the program provides

us with these tensor products, and likewise with similar fusion algebras that appear in other contexts [15]. Note that for a specific problem of interest, one will usually need only a few fusion coefficients explicitly, so that the computing time will rise only as $(n_{\text{pf}})^3$ rather than $(n_{\text{pf}})^4$.

In addition, using the matrix S one can construct new non-diagonal modular invariant partition functions: one simply has to look for those matrices M with non-negative integer entries that commute with S (and T). The simplest examples for such matrices are the automorphism invariants; these are an immediate by-product of the program, as they correspond precisely to the possible ambiguities in the ordering of the columns of S . In this way we have found, for instance, an automorphism invariant for E_8 at level 4; it corresponds to the exchange³

$$\phi_{\Lambda_{(1)}} \longleftrightarrow \phi_{\Lambda_{(6)}}. \quad (7)$$

Finally there exists a rather different application of the algorithm. Namely, as can be read off the formulæ (3) and (4), for $(\bar{\Lambda} + \bar{\Lambda}', \bar{\theta}) \leq k$ with $\bar{\theta}$ the highest root of $\bar{g}$, the fusion rule coefficients coincide with the tensor product multiplicities of $\bar{g}$ tensor products. Thus one may compute the latter multiplicities by determining the corresponding fusion rule at sufficiently high level. This becomes interesting if modules with very large dimensions are involved, for which existing Lie algebraic algorithms for the calculation of tensor products become slow. For example, we have checked our program for some tensor products of E_8 modules of dimensions of the order of 10^8 against the smart software package liE [17] and found that our program indeed tends to be faster.

³The existence of this automorphism invariant had been anticipated by D. Versteegen [16].

Table 1. G_2 fusion rules.

k	n_{pf}	ν_0	ν_1	ν_2	ν_3	$\mathcal{N}_{\text{max}}$
1	2	3	5	–	–	1
2	4	34	30	–	–	1
3	6	99	113	4	–	2
4	9	348	321	57	3	3
5	12	726	712	232	48	5
6	16	1 738	1 409	601	229	7
7	20	3 108	2 377	1 347	588	10
8	25	6 069	4 013	2 368	1 261	13
9	30	9 864	5 917	3 995	2 495	17
10	36	16 978	9 140	5 987	4 177	24
11	42	25 809	12 454	8 788	6 854	31
12	49	40 779	18 086	12 228	9 993	41
13	56	58 932	23 479	16 492	14 710	48
14	64	87 505	32 612	21 880	19 935	60
15	72	121 557	40 915	27 985	27 351	75
16	81	172 191	54 890	35 916	35 416	96
17	90	231 672	67 216	44 212	46 361	115
18	100	316 270	87 656	55 441	58 070	133
19	110	414 459	105 379	66 316	73 484	159
20	121	549 207	134 228	81 669	89 784	191

Table 2. E_8 fusion rules.

k	n_{pf}	ν_0	ν_1	ν_2	ν_3	$\mathcal{N}_{\text{max}}$
1	1	0	1	–	–	1
2	3	17	10	–	–	1
3	5	70	55	–	–	1
4	10	522	422	56	–	2
5	15	982	1 157	742	312	8
6	27	6 234	4 681	2 574	1 567	29
7	39	12 246	7 234	5 056	3 938	76
8	63	48 865	24 003	14 997	11 059	445
9	90	110 547	42 407	26 005	21 522	2 132
10	135	330 321	108 822	65 386	52 663	8 628

References

- [1] V.G. Knizhnik and A.B. Zamolodchikov, Nucl. Phys. B 247 (1984) 83
- [2] D. Gepner and E. Witten, Nucl. Phys. B 278 (1986) 493
- [3] E. Verlinde, Nucl.Phys. B 300 (1988) 389
- [4] G. Moore and N. Seiberg, Phys.Lett. B 212 (1988) 451
- [5] V.G. Kac and D.H. Peterson, Adv.Math. 53 (1984) 125;
V.G. Kac and M. Wakimoto, Adv.Math. 70 (1988) 156
- [6] V.G. Kac, talk at the Canadian Mathematical Society meeting on *Lie Algebras and Lie Groups* (Montréal, August 1989);
M. Walton, Nucl.Phys. B 340 (1990) 777
- [7] M. Walton, Phys.Lett. B 241 (1990) 365; *ibid.* B 244 (1990) 580 (E);
P. Furlan, A.Ch. Ganchev, and V.B. Petkova, Nucl.Phys. B 343 (1990) 205;
J. Fuchs and P. van Driel, Nucl.Phys. B 346 (1990) 632
- [8] M. Spiegelglas, Phys.Lett. B 247 (1990) 36
- [9] D. Speiser, Helv.Phys.Acta 38 (1965) 73, and in: Istanbul Summer School on *Group Theoretical Concepts and Methods in Elementary Particle Physics*, F. Gursey ed. (Gordon and Breach, New York 1964), p. 237;
G. Racah, *ibid.*, p. 1
- [10] A.N. Schellekens and S. Yankielowicz, Nucl.Phys. B 327 (1989) 673;
Int.J.Mod.Phys. A 5 (1990) 2903
- [11] J. Fuchs, to appear in Commun.Math.Phys.
- [12] P. Degiovanni, Commun.Math.Phys. 127 (1990) 71
- [13] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge 1986);
B.T. Smith et al., *Matrix Eigensystem Routines – EISPACK Guide* (Springer Lecture Notes in Computer Science vol. 6, Springer Verlag, New York 1976)
- [14] V. Pasquier and H. Saleur, Nucl. Phys. B 330 (1990) 523;
L. Alvarez-Gaumé, C. Gómez, and G. Sierra, Nucl.Phys. B 330 (1990) 347;
J. Fuchs and P. van Driel, J.Math.Phys. 31 (1990) 1770
- [15] F.M. Goodman and T. Nakanishi, Nagoya preprint NU-MATH-006 (1991)
- [16] D. Verstegen, private communication
- [17] A.M. Cohen et al., *lie manual*, CWI (Amsterdam 1989)