IOPscience iopscience.iop.org

Home Search Collections Journals About Contact us My IOPscience

Algorithmic simplification of tensor expressions

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

1999 J. Phys. A: Math. Gen. 32 7779

(http://iopscience.iop.org/0305-4470/32/44/313)

View the table of contents for this issue, or go to the journal homepage for more

Download details:
IP Address: 171.66.16.111
The article was downloaded on 02/06/2010 at 07:48

Please note that terms and conditions apply.

J. Phys. A: Math. Gen. **32** (1999) 7779–7789. Printed in the UK PII: S0305-4470(99)04458-3

Algorithmic simplification of tensor expressions

R Portugal

Centro Brasileiro de Pesquisas Físicas, Rua Dr Xavier Sigaud 150, Urca, CEP 22290-180 Rio de Janeiro, Brazil

E-mail: `portugal@cbpf.br`

Received 19 May 1999

Abstract. We present efficient algorithms for simplifying tensor expressions that obey generic symmetries. We define the canonical form of a single tensor and we show that the problem of finding the canonical form of a generic tensor expression reduces to finding the canonical form of single tensors. Special symmetries are considered in order to push the efficiency further. We also present algorithms to address the cyclic symmetry of the Riemann tensor. With these algorithms it is possible to simplify generic Riemann tensor polynomials.

1. Introduction

The simplification of tensor expressions is tricky and cumbersome to perform by manual calculations. Expressions with the Riemann tensor provide good examples. It is difficult to show by hand that

$$R^{abcd}R^{efkh}R^{i}{}_{a}{}^{j}{}_{e}R_{bcdi}R_{fkhj} = 0.$$

This example does not require the use of the cyclic identity of the Riemann tensor. The next example is more difficult to prove:

$$\begin{aligned}2R^{abcd}R^{ef}{}_{ak}R^{h}{}_{c}{}^{k}{}_{b}R_{dhef} + 4R^{abcd}R^{efk}{}_{a}R^{h}{}_{bce}R_{dkfh}\\ -R^{abcd}R^{e}{}_{f}{}^{k}{}_{a}R^{h}{}_{b}{}^{f}{}_{e}R_{cdkh} + 4R^{abcd}R^{hk}{}_{df}R^{ef}{}_{ka}R_{hbce} = 0.\end{aligned}$$

These examples, among others, show the necessity of constructing algorithms for tensor simplifications that can be implemented in a computer language. Such algorithms are extensions of the simplification problem of algebraic expressions and have a natural place in computer algebra systems.

The main references that address the tensor simplification problem from an algorithmic point of view are Portugal [1], Dresse [2], Ilyin and Kryukov [3] and Jaén and Balfagón [4]. The simplification of Riemann tensor polynomials is addressed by Parker and Christensen [5] and Fulling *et al* [6].

In this paper we use the abstract-index notation for tensor expressions (see a careful description of the abstract-index notation in Penrose and Rindler [7], and take Lovelock and Rund [8] as a general reference for the tensor calculus). We assume that the metric is symmetric and the tensor product is commutative as usual. This work can be extended easily for antisymmetric metrics (such as the two-spinor calculus metric [7]) and for anti-commutative products (such as in Grassmann algebras). We take the definition of canonical form from Geddes *et al* [9].

0305-4470/99/447779+11$30.00 © 1999 IOP Publishing Ltd

The strategy of this paper is as follows. The first part addresses the simplification problem considering only the monoterm symmetries (symmetries that involve only index permutations). The second part addresses the multiterm symmetries (cyclic symmetries). In the first part we define the canonical form of a single tensor with a generic monoterm symmetry (section 2). By using the full reduction method, we define the canonical form of a generic tensor expression (section 3). Special methods for two kinds of symmetry are developed in order to increase the efficiency (section 4). The cyclic symmetry of the Riemann tensor is addressed in section 5. The algorithms presented here allow one to simplify Riemann tensor polynomials efficiently.

2. Definition of the canonical form of a single tensor

We address monoterm symmetries separately from multiterm symmetries. A monoterm symmetry of a tensor T is a finite equivalence class of the form

$$\{\epsilon_{\sigma_1} T^{\sigma_1}, \epsilon_{\sigma_2} T^{\sigma_2}, \ldots\} \tag{1}$$

where ϵ_σ is either 1 or -1, and σ is a permutation of the indices of T. For now, suppose that T obeys a generic monoterm symmetry. Multiterm symmetries will be considered in section 5.

2.1. Configurations with no dummy indices

Suppose that all indices of T are contravariant free indices. The description of the symmetry is *complete* if it is not possible to generate a new member in the equivalence class (1), that is, all combinations $\sigma_k = \sigma_i \circ \sigma_j$ are present. Clearly, the set $\{\sigma_1, \sigma_2, \ldots\}$ of a complete symmetry is a subgroup of the permutation group S_n where n is the rank of T. The symmetry cancels the tensor if and only if there are two equal permutations $\sigma_i = \sigma_j$ such that $\epsilon_{\sigma_i} = -\epsilon_{\sigma_j}$. For example, the symmetry described incompletely by

$$T^{ijk} = -T^{kij} \tag{2}$$

cancels the tensor, since the equivalence class is

$$\{T^{ijk}, -T^{kij}, T^{jki}, -T^{ijk}, T^{kij}, -T^{jki}\}. \tag{3}$$

For each element of the equivalence class we can associate a numerical list. Now we establish an ordering for the numerical lists. Consider two ordered lists (L_1 and L_2) of non-repeated positive integers. The lists have the same number of elements n. We define recursively the relation '$<$' for lists: $L_1 < L_2$ if

$$L_1[1] < L_2[1] \qquad \text{or} \qquad (L_1[1] = L_2[1] \quad \text{and} \quad L_1[2..n] < L_2[2..n]) \tag{4}$$

where $L[i]$ means the ith element of L and $L[2..n]$ means $\mathrm{list}(L[2], L[3], \ldots, L[n])$.

We define the *canonical form* of T using the following procedure. Let F be the index list sorted into alphabetical order. Consider the list of equations $\mathrm{list}(F[i] = i, i = 1..|F|)$, where $|F|$ is the number of elements of list F. The indices of each element of the equivalence class (1) are replaced by numbers according to this list of equations. So, to each element of the equivalence class, we associate a numerical list. The canonical form is the configuration $\epsilon_i T^{\sigma_i}$ associated with the smallest numerical list. If there are free covariant indices, a similar procedure is applied. Let F^{up} be the sorted list of contravariant indices and F^{dn} be the sorted list of covariant indices. The substitution equations are

$$\mathrm{list}(F^{\mathrm{up}}[i] = i, i = 1..|F^{\mathrm{up}}|) \qquad \mathrm{list}(F^{\mathrm{dn}}[i] = i + |F^{\mathrm{up}}|, i = 1..|F^{\mathrm{dn}}|). \tag{5}$$

For example, let T be a totally symmetric tensor. The configuration $T_j{}^b{}_i{}^a$ is associated with [4, 2, 3, 1], since the substitution equations are $a = 1, b = 2, i = 3, j = 4$. The canonical configuration is $T^{ab}{}_{ij}$, which is associated with [1, 2, 3, 4].

2.2. Configurations with dummy indices

Now suppose that there are dummy indices in the original configuration of T. The equivalence class (1) can be generated by the permutations σ_i pretending that all indices are free. There are two operations that generate new elements in the equivalence class when there are dummy indices. The character of a pair of dummy indices can be inverted (e.g. $T^k{}_k \to T_k{}^k$) and the names of the dummy indices can be replaced by other ones (e.g. $T^k{}_k \to T^j{}_j$). One of the goals of this work is to show that only the symmetry class (1) need be considered in order to find the canonical form. We follow two conventions throughout this work.

Convention 1. The character of the first index of a pair of dummy indices is contravariant, and the character of the second is covariant.

Convention 2. The name of the dummy indices is given by $\_i\_j$, where i is the position of the contravariant index, j is the position of the covariant index and $\_$ is some separator. e.g. for the configuration $T^{kl}{}_{lk}$, we have $k = \_1\_4$ and $l = \_2\_3$.

Now let us define the canonical form for a configuration with dummy indices. We have already established the substitution equations for the free indices†. Now we define the substitution equations for the dummy indices. There are two kinds of pairs of dummy indices depending on how they are affected by the symmetry. The first kind consists of the pairs for which both indices change position in the equivalence class (1). The second kind consists of the pairs for which one index changes position while the other maintains its position for all elements of the equivalence class. The pairs that have both indices fixed need not be considered, they simply obey convention 2.

The contravariant indices of the first kind form class A^{up} and the covariant indices form class A^{dn}. The non-fixed indices of the second kind form class B and the fixed indices form class B^{fixed}. These classes are ordered lists. The order of indices of classes A^{up} follows the occurrence order of the indices in each configuration $\epsilon_i T^{\sigma_i}$. The substitution equations are

$$\mathrm{list}(A^{\mathrm{up}}[i] = i + |F^{\mathrm{up}}| + |F^{\mathrm{dn}}|, i = 1..|A^{\mathrm{up}}|). \tag{6}$$

The substitution equations for class A^{dn} are the same for class A^{up} except for a shift of $|A^{\mathrm{up}}|$. For example, if the index $k \in A^{\mathrm{up}}$ is replaced by 5, the corresponding index of class A^{dn} is replaced by $5 + |A^{\mathrm{up}}|$.

The order of class B^{fixed} follows the occurrence order of the index list of T. Notice that all elements of (1) have the same class B^{fixed}. The substitution equations are

$$\mathrm{list}(B^{\mathrm{fixed}}[i] = i + |F^{\mathrm{up}}| + |F^{\mathrm{dn}}| + 2|A^{\mathrm{up}}|, i = 1..|B^{\mathrm{fixed}}|). \tag{7}$$

The indices of class B^{fixed} are not replaced by numbers. The substitution equations (7) are used to replace the indices of class B.

For example, let $T_j{}^i{}_a{}^{abc}{}_{cb}$ be the actual configuration of the indices of a tensor T totally symmetric in the first six indices. The classes are $F^{\mathrm{up}} = i$, $F^{\mathrm{dn}} = j$, $A^{\mathrm{up}} = a^{\mathrm{up}}$, $A^{\mathrm{dn}} = a^{\mathrm{dn}}$, $B^{\mathrm{fixed}} = c, b$. Only the indices affected by the symmetry need to be replaced by numbers. We obtain the following list: [2, 1, 4, 3, 6, 5]. The equivalence class of T has all permutations

† The method presented here is a simplified version of the method of [1]. Balfagón kindly informed us that he used a similar method in [4].

of [2, 1, 4, 3, 6, 5]. The smallest list is [1, 2, 3, 4, 5, 6], and the canonical form is $T^{i}{}_{j}{}^{a}{}_{a}{}^{cb}{}_{cb}$ where $a = \_3\_4$, $c = \_5\_7$, $b = \_6\_8$.

We have defined the canonical form of a single tensor (with or without contractions) with a generic monoterm symmetry. Using the *full reduction method* described below, we show that the problem of finding the canonical form of a generic tensor expression reduces to finding the canonical form of single tensors (which will be called *merged tensors*).

3. Canonical form of a tensor expression: the full reduction method

Now we describe the reduction process that a generic tensor expression passes through in order that the canonical form be obtained. Part of this process is described in [1]. We consider here only the main steps.

A generic tensor expression is expanded and expressed as a linear combination of tensor monomials. A tensor monomial is split into two groups. Group I consists of tensors with symmetries. Group II consists of tensors with no symmetries. Tensors of group I are placed in the left-hand positions and tensors of group II in the right-hand positions using the commutativity property of the product.

Tensors of group II with the same name and number of indices and with no free indices merge into a new tensor. Suppose that each original tensor has n indices and that there are N equal tensors, then the new tensor has nN indices and is totally symmetric under interchange of the groups of the n indices. After the merging, the resulting tensors are incorporated into group I. The information about the relation with original tensors is stored, since it will be used at the end of the algorithm to replace the new tensors by the original tensors.

Tensors of group I with the same names, same number of indices and same number of free indices merge to form a new tensor. The new tensor has the same symmetry under the interchange of group of indices as described for group II, and each group of indices inherits the symmetries of the original tensors.

The tensors are lexicographically sorted inside each group. Tensors with the same name but a different number of indices are sorted according to the number of indices. Tensors with the same name and the same number of indices are sorted according to the number of free indices. Tensors of group II with the same name, the same number of indices and the same number of free indices are sorted according to the name of the first free index. Notice that the free indices have fixed positions for tensors of group II.

The whole monomial merges into a single tensor (from now on called the *merged tensor*). The number of indices of the merged tensor is equal to the total number of indices of the monomial. The ordering of the indices obeys the ordering of the indices of each term of the monomial. The merged tensor inherits all the symmetries of the tensors of the monomial.

Let us show an example. Suppose that the monomial to be simplified is

$$S_{ij} T_{mp} A^{i} B_{k} A^{j} S^{km} \tag{8}$$

where S^{km} is totally symmetric and T_{mp} has no symmetry. The repeated tensors merge and the ordering of the terms is

$$A\_A^{ij} S\_S_{ij}{}^{km} B_{k} T_{mp} \tag{9}$$

where A_A is the name for the new tensor generated after merging A^{i} and A^{j}. The tensor A_A^{ij} is totally symmetric and the symmetries of S_S^{ijkm} are

$$S\_S^{ijkm} = S\_S^{jikm}$$
$$S\_S^{ijkm} = S\_S^{ijmk}$$
$$S\_S^{ijkm} = S\_S^{kmij}.$$

The merged tensor is

$$M^{ij}{}_{ij}{}^{km}{}_{kmp} \tag{10}$$

and the symmetries are

$$\begin{aligned}
M^{ijklmnopq} &= M^{jiklmnopq}\\
M^{ijklmnopq} &= M^{ijlkmnopq}\\
M^{ijklmnopq} &= M^{ijklnmopq}\\
M^{ijklmnopq} &= M^{ijmnklopq}.
\end{aligned}$$

The canonical form of (10) is

$$M^{ijkm}{}_{ijkmp} \tag{11}$$

and therefore the canonical form of (8) is

$$A^i A^j S^{km} S_{ij} B_k T_{mp} \tag{12}$$

where $i = \_1\_5$, $j = \_2\_6$, $k = \_3\_7$ and $m = \_4\_8$.

To summarize, we have described an algorithm that converts a tensor expression into a sum of merged tensors. Each merged tensor can be put into canonical form. The elements of the equivalence class (1) must obey convention 1. The indices affected by the symmetry are replaced by numbers according to the substitution equations for classes $F^{\rm up}$, $F^{\rm dn}$, $A^{\rm up}$, $A^{\rm dn}$ and B. Note that the substitution equations for classes $F^{\rm up}$, $F^{\rm dn}$ and B are the same for all elements of the equivalence class (1), while the substitution equations for $A^{\rm up}$ and $A^{\rm dn}$ depend on each element. The configuration with the smallest numerical list corresponds to the canonical form. By the inverse process, each merged tensor can be converted back to a tensor monomial with the indices in the new order and obeying convention 2. The resulting expression is the canonical form of the original expression with respect to the monoterm symmetries†.

3.1. *Variations of the full reduction*

There are cases that do not need the full monomial reduction.

(a) Tensors with no symmetries and no dummy indices need not be merged.

(b) If there are two or more groups that have no dummy indices in common, they can be split and each group can be put into the canonical form independently of each other. For example, the monomial (8) can be split into two groups

$$S_{ij} A^i A^j \qquad\qquad S^{km} B_k T_{mp} \tag{13}$$

and each one can be considered separately. A special case occurs when there are two or more groups with the same canonical form. These groups cannot split. They merge like the other tensors in order to generate (unique) canonical names for the dummy indices.

† *Note about the original algorithm of* [1]. Classes S_I and S_{II} disappear with the full reduction process. Both subclass $S_I^{0^+}$ and class S_{II} reduce to class B, subclass $S_I^{0^-}$ to $B^{\rm fixed}$ and subclasses S_I^i to class A. Although the implementation of the monomial reduction to a single tensor is cumbersome, it simplifies the general structure of the algorithm by the complete elimination of classes S_I and S_{II}. The result of the new algorithm is not the same compared to the original algorithm, since the positions of the classes S_I and S_{II} have changed.

4. Special algorithms

In this section we present special algorithms for two kinds of symmetries that occur frequently in practical applications. We define compact forms for them which replace the equivalence class.

The first kind is described by

$$sym(i_1, i_2, \ldots, i_m) \;\&\; sym(j_1, j_2, \ldots, j_n) \;\&\; \cdots \tag{14}$$

such that

$$i_1 < \cdots < i_m < j_1 < \cdots < j_n < \cdots \tag{15}$$

where '*sym*' means symmetric ('*asym*' means antisymmetric) and $i_1, i_2, \ldots$ are numbers describing slots (index positions) in the index configuration. These numbers need not be consecutive.

The second kind is described by

$$sym\_in\_block(i_1, i_2, \ldots, i_m)(j_1, j_2, \ldots, j_m) \ldots . \tag{16}$$

The numbers $i_1, i_2, \ldots$ obey equation (15). This symmetry states that the interchange of blocks $(i_1, \ldots, i_m)$, $(j_1, \ldots, j_m)$ and so on generates equivalent configurations. The symmetry *asym_in_block* means the same, but a negative sign is introduced for odd permutations. For example, the symmetry

$$T^{abcijk} = -T^{ijkabc} \tag{17}$$

is described by $asym(1, 2, 3)(4, 5, 6)$. The symmetries in blocks can be combined among themselves and with symmetries of the first kind using the character &. For example, the compact description of the monoterm symmetries of the Riemann tensor is $asym(1, 2)$ & $sym\_in\_block(1, 2)(3, 4)$. The *complete* description of these compact forms means that whenever there is a symmetry *sym_in_block*, the symmetries of a single block is repeated for all blocks. For example, the complete compact description of the Riemann symmetries is $asym(1, 2)$ & $asym(3, 4)$ & $sym\_in\_block(1, 2)(3, 4)$. In the sequel, we assume that the description of the monoterm symmetries is complete. We will not present a full analysis of the combination of these symmetries, but the interested reader can find some insights in section 3.3 of Penrose and Rindler's book [7].

4.1. Symmetric indices

Suppose that the symmetry of the merged tensor is of the form (14). This symmetry occurs when the merged tensor comes from a monomial built of different tensors, some of them totally symmetric. For this kind of symmetry, it is not necessary to generate the equivalence class in order to find the canonical form. The indices can be put into their canonical positions in a straightforward way. The substitution equations for classes F^{up}, F^{dn} and B are given by (5) and (7). Now we describe how class A is replaced by numbers. Consider group $(i_1, i_2, \ldots, i_m)$ of (14). There are two kinds of indices of class A. The indices that are contracted with indices of the same group (p indices) and indices that are contracted with indices of other groups (q indices such that $p + q \leqslant m$). The first index of the first kind receives the number $f + 1$, the second $f + 2$ and so on until the last that receive the number $f + p$, where $f = |F^{\mathrm{up}}| + |F^{\mathrm{dn}}|$. The corresponding indices receive a shift of $|A^{\mathrm{up}}|$. The indices of the second kind are numbered according to the order of the corresponding indices. The index of the second kind contracted with the first corresponding index receives the numbers $f + p + 1$. The index contracted with the second receives $f + p + 2$, and so on until $f + p + q$. The corresponding indices receive a shift of $|A^{\mathrm{up}}|$. The same procedure is performed for the other groups in sequence. After all indices have been replaced by numbers, they are sorted inside each group. The free indices are replaced by their original names and the dummy indices obey convention 2. The indices are in the canonical positions now.

For example, consider the index configuration $T^{abc}{}_{acibj}$ with the symmetry *sym*$(1, 2, 3, 4)$ & *sym*$(5, 6)$ & *sym*$(7, 8)$. The first group has the following substitution equations: $a^{\mathrm{up}} = 3, b^{\mathrm{up}} = 5, c^{\mathrm{up}} = 4, a^{\mathrm{dn}} = 6$. Eventually we obtain the following numerical configuration: $T^{354}{}_{67182}$. By a straightforward sorting process inside the groups, we obtain the configuration $T^{345}{}_{61728}$. The canonical form is $T^{acb}{}_{aicjb}$, where $a = \_1\_4$, $c = \_2\_6$ and $b = \_3\_8$.

If one or more groups are antisymmetric, there are two remarks. First, one must keep track of the sign change for each index permutation during the sorting process at the end. Secondly, one must verify whether there are antisymmetric indices which contract with symmetric indices. This is the only case that can cancel the merged tensor according to lemma 2 of [1].

The solution for this kind of symmetry can be considered optimal in the sense that given an index configuration, the canonical form is obtained in a straightforward manner. Neither the equivalence class nor a subset need be generated.

4.2. Symmetric under the interchange of groups of indices

Suppose that the merged tensor is symmetric under the interchange of N groups of indices (described by (16)). The groups can have two or more totally symmetric indices (combination of (16) and (14)). This kind of symmetry occurs when the merged tensor comes from a monomial that has tensors (some of them totally symmetric) with the same name, same number of indices and same number of free indices. In the general case, it is necessary to generate all permutations of the symmetric groups ($N!$ permutations), and each permutation can be put straightforwardly into the canonical position by the method discussed previously for the symmetric indices (section 4.1).

If the merged tensor is antisymmetric under the interchange of groups of indices, there are two remarks. First, one must keep track of the sign for group permutations. Secondly, one must verify whether there are two canonical forms with opposite signs at the end of the algorithm. Lemma 2 of [1] guarantees that if the merged tensor is zero then the algorithm above will generate two equivalent configurations with opposite signs.

5. The cyclic symmetry

A tensor T obeys the cyclic symmetry in the indices i, j, k if

$$T^{ijk} + T^{kij} + T^{jki} = 0. \tag{18}$$

T can have more indices that are held fixed. If T^{ijk} obeys the cyclic symmetry and is antisymmetric in two indices of the set $\{i, j, k\}$, then

$$T^{[ijk]} = 0 \tag{19}$$

where the square brackets denote antisymmetrization.

5.1. Isolated cyclic symmetry

Suppose that the indices obeying the cyclic symmetry do not obey any other kind of symmetry. In order to address the cyclic symmetry of the merged tensor, we assume that it was put into the canonical form with respect to the monoterm symmetries. Suppose that the indices obeying the cyclic symmetry are i, j, k. Let us establish the substitution equations for the set $\{i, j, k\}$. The contravariant free indices of this set form the class F^{up}. The covariant free indices form the class F^{dn}. The remaining indices form class C. The substitution equations for classes F^{up} and F^{dn} are given by equations (5). The substitution equations for class C are based on the order of the corresponding indices. The numbers reserved for this class are $(|F^{\mathrm{up}}| + |F^{\mathrm{dn}}| + 1)..3$. If there is a pair of dummy indices within class C, the substitution equations are obtained based on the contravariant index, and the number for the covariant index is $|F^{\mathrm{up}}| + |F^{\mathrm{dn}}| + |C| + 1$. For example, if $T^{ab}{}_{ba}$ has cyclic symmetry on the first three indices (and no other monoterm symmetry), classes F^{up} and F^{dn} are empty, and class C is $[a, b^{\mathrm{up}}]$. The substitution equations are $a = 2$, $b^{\mathrm{up}} = 1$ and $b^{\mathrm{dn}} = 3$. For the generic case, the algorithm generates the equation

$$T^{kji} = -T^{ikj} - T^{jik}. \tag{20}$$

Only the indices obeying the cyclic symmetry are represented in equation (20). These indices need not be consecutive. The remaining indices are held fixed. Equation (20) will be used depending on the following criteria. If the numerical list associated with the original expression is smaller† than the lists associated with the numerical configurations of the terms of the right-hand side of equation (20), then the configuration can be improved by (20). Otherwise the original configuration is in the canonical form.

If there are more indices obeying the cyclic symmetry (and obeying no other kind of symmetry), the same procedure is applied to the tensors of the right-hand side of equation (20) in a recursive way, until all independent cyclic symmetries have been covered. The number of terms will be at most 2^n, where n is the number of independent cyclic symmetries of T. In general, the canonical form of a single tensor obeying multiterm symmetries is a sum of tensors, while the canonical form obeying only monoterm symmetries is always a single tensor.

5.2. The cyclic symmetry of the Riemann tensor

The next important case occurs when the indices i, j, k, l of a tensor T with at least four indices obey the symmetries of the Riemann tensor (and no other symmetries). The symmetries are

$$T^{ijkl} = -T^{jikl} = -T^{ijlk} \tag{21}$$

$$T^{ijkl} = T^{klij} \tag{22}$$

$$T^{ijkl} + T^{iljk} + T^{iklj} = 0. \tag{23}$$

In order to address the Riemann symmetry, we assume that the merged tensor is put into the canonical form with respect to the monoterm symmetries, and its configuration is T^{iljk}. Indices i, j, k, l can be free or dummy indices (they represent generic indices). Using the substitution equations of section 2, we can replace these indices by numbers. Let i_n, j_n, k_n, l_n be the corresponding numbers. The order of the indices of the merged tensor can be improved only if

$$l_n > j_n \quad \text{and} \quad l_n > k_n. \tag{24}$$

In this case, the merged tensor is replaced by

$$T^{iljk} = T^{ikjl} - T^{ijkl}. \tag{25}$$

† See the definition given by expression (4).

If equation (24) is true, the canonical form is the right-hand side of equation (25). If equation (24) is false, the index configuration cannot be improved. If there are other groups of four indices obeying the symmetries of the Riemann tensor, the same procedure is applied recursively to the right-hand side of equation (25).

When the original tensor expression is the product of Riemann tensors with the same number of free indices, the symmetry of the merged tensor is the symmetry of the Riemann tensor in each consecutive group of four indices, and totally symmetric under the interchange of these groups. For example, the Riemann invariants (product of Riemann tensors fully contracted) fall in this case.

In order to find the canonical form, consider the first group of four indices. If equation (24) is satisfied, apply equation (25) and call the whole algorithm recursively. If equation (24) is false, the numerical configuration of T (L_T) is stored in order to be compared with the following result. Equation (25) is used (even with (24) false) generating two terms. The algorithm for monoterm symmetries is applied for each term generating two numerical configurations. Let us call these L_{T_1} and L_{T_2}.

If

$$L_{T_1} < L_T \quad \text{and} \quad L_{T_2} < L_T \tag{26}$$

then the configuration T was improved and the algorithm is called recursively over the sum T_1 and T_2 (with the proper sign).

If

$$(L_{T_1} = L_T \quad \text{and} \quad L_{T_2} < L_T) \qquad \text{or} \qquad (L_{T_1} < L_T \quad \text{and} \quad L_{T_2} = L_T) \tag{27}$$

then the configuration T can be improved by the equation $T = \frac{1}{2}T_2$ if $L_{T_1} = L_T$ (or $T = \frac{1}{2}T_1$ if $L_{T_2} = L_T$). See the forthcoming example in order to verify the sign of T_1 and T_2.

If

$$L_T < L_{T_1} \quad \text{or} \quad L_T < L_{T_2} \tag{28}$$

then the index configuration cannot be improved by the application of the cyclic symmetry over the first four indices. The same procedure is applied to the next group of four indices and, if equation (28) is satisfied again, the same procedure is applied to the next group and so on until we reach the last group†.

For example, consider the following monomial built of Riemann tensors:

$$R^{abcd} R^{e}{}_{a}{}^{f}{}_{c} R_{bfde}. \tag{29}$$

This configuration is in the canonical form with respect to the monoterm symmetries (except for the names of the dummy indices). All indices are members of class A and the numerical configuration of the merged tensor is [1, 2, 3, 4, 5, 7, 6, 9, 8, 12, 10, 11]. Consider group a, b, c, d of expression (29). Even with equation (24) false, we apply equation (25) to obtain

$$(R^{adcb} - R^{acdb}) R^{e}{}_{a}{}^{f}{}_{c} R_{bfde}. \tag{30}$$

The canonical form with respect to the monoterm symmetries is‡

$$R^{abcd} R^{e}{}_{a}{}^{f}{}_{c} R_{bedf} - R^{abcd} R^{ef}{}_{ac} R_{bedf}. \tag{31}$$

† There is an alternate method based on a library of pre-defined rules [5] which has a drawback: the number of rules is huge for Riemann monomials with free indices (see [6]).

‡ The canonical form uses convention 2 for naming the dummy indices. In the next expressions we have replaced the canonical names by a, b, c, d, e, f.

Let us call the first term T_1, and the second term T_2. The numerical configurations are [1, 2, 3, 4, 5, 7, 6, 9, 8, 11, 10, 12] and [1, 2, 3, 4, 5, 6, 7, 9, 8, 11, 10, 12], respectively. Notice that both configurations are smaller than the original one. The application of the algorithm on T_1 does not generate any improvement, since equations (26) and (27) are not satisfied for any of the 4-index groups. So, T_1 is in the canonical form with respect to the full symmetry (monoterm plus cyclic symmetries). The algorithm is applied recursively to T_2. The application of equation (25) on group a, b, c, d generates the following terms already in the canonical form with respect to the monoterm symmetries:

$$-R^{abcd} R^{ef}{}_{ac} R_{bedf} + R^{abcd} R^{ef}{}_{ab} R_{cedf}. \tag{32}$$

The first term is equal to $-T_2$ while the second term (let us call it T_3) has the numerical configuration [1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 10, 12]. T_2 is replaced by $\frac{1}{2}T_3$. T_3 can be improved further by the application of equation (25) on the last group of indices: c, e, d, f. The result is

$$-R^{abcd} R^{ef}{}_{ab} R_{cedf} + R^{abcd} R^{ef}{}_{ab} R_{cdef}. \tag{33}$$

The first term is equal to $-T_3$ while the second term has the numerical configuration [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] which cannot be improved further. The canonical form of the original expression (29) (except for the names of the dummy indices) is

$$R^{abcd} R^{e}{}_{a}{}^{f}{}_{c} R_{bedf} - \tfrac{1}{4} R^{abcd} R^{ef}{}_{ab} R_{cedf}. \tag{34}$$

6. Conclusion

Reference [1] suggests the use of the Gröbner basis method to address the simplification problem of tensor expressions obeying multiterm symmetries. The Gröbner method is suitable if the number of side relations is not large, since it has very narrow limitations concerning efficiency. For Riemann tensor polynomials the number of side relations is huge. The method presented here solves this drawback.

The full reduction method plays a central role, since it simplifies the general structure of the algorithm. In other words, we have proved the following proposition.

Proposition. *The problem of finding the canonical form of a generic tensor expression reduces to finding the canonical form of single tensors.*

This proposition tells us that it is simpler to address the tensor simplification problem by finding the canonical form of a single tensor and using the full reduction method for monomials than to address it by implementing algorithms for tensor monomials that will be unavoidably repeated for single tensors.

Acknowledgment

I thank M A H MacCallum for drawing my attention to [2].

References

[1] Portugal R 1998 An algorithm to simplify tensor expressions *Comput. Phys. Commun.* **115** 215–30
[2] Dresse A 1993 *PhD Thesis* Université Libre de Bruxelles
[3] Ilyin V A and Kryukov A P 1996 ATENSOR—REDUCE program for tensor simplification *Comput. Phys. Commun.* **96** 36–52

[4] Jaén X and Balfagón A 1998 TTC: symbolic tensor calculus with indices *Comput. Phys.* **12** 286–9
[5] Parker L and Christensen S M 1994 *MathTensor, A System for Doing Tensor Analysis by Computer* (Reading, MA: Addison-Wesley)
[6] Fulling S A, King R C, Wybourne B G and Cummins C J 1992 Normal forms for tensor polynomials: I. The Riemann tensor *Class. Quantum Grav.* **9** 1151–97
[7] Penrose R and Rindler W 1992 Two-spinor calculus and relativistic fields *Spinors and Space-Time* vol 1 (Cambridge: Cambridge University Press)
[8] Lovelock D and Rund H 1975 *Tensors, Differential Forms and Variational Principles* (New York: Wiley)
[9] Geddes K O, Czapor S R and Labahn G 1992 *Algorithms for Computer Algebra* (Dordrecht: Kluwer) p 82