Fast Change of Basis in Algebras

Irvin Roy Hentzel
Department of Mathematics
Iowa State University
Ames, Iowa 50011 USA

D.P. Jacobs *
Department of Computer Science
Clemson University
Clemson, S.C. 29634-1906 USA

Abstract

Given an n-dimensional algebra $\mathcal{A}$ represented by a basis B and structure constants, and given a transformation matrix for a new basis C, we wish to compute the structure constants for $\mathcal{A}$ relative to C. There is a straightforward way to solve this problem in $O(n^5)$ arithmetic operations. However given an $O(n^\omega)$ matrix multiplication algorithm, we show how to solve the problem in time $O(n^{\omega+1})$. Using the method of Coppersmith and Winograd, this yields an algorithm of $O(n^{3.376})$.

Key words: algebra, vector space, transformation matrix.

0 Introduction

Consider the following problem. We are given the structure constants, relative to a certain basis, for an n-dimensional nonassociative algebra. We are also given a transformation matrix for changing to a new basis. Our problem is then to compute the structure constants relative to the new basis. In Section 1 we will formulate this problem precisely, and in Section 2 we will outline a straightforward $O(n^5)$ solution. In the remainder of the paper we explain how the problem can be solved in $O(n^{3.376})$ scalar operations.

1 Notation and Terminology

A (nonassociative) *algebra* $\mathcal{A}$ over a field $\mathbf{F}$ is a vector space over $\mathbf{F}$ along with a multiplication operator for which

1. $\lambda(ab) = (\lambda a)b = a(\lambda b)$

*This research was partially supported by NSF Grant #CCR8905534.

2. $a(b+c) = ab + ac$

3. $(a+b)c = ac + bc$

for all $a, b, c \in \mathcal{A}$ and $\lambda \in \mathbf{F}$. Throughout this paper we assume $\mathcal{A}$ is an algebra of finite dimension n over $\mathbf{F}$. Finite dimensional nonassociative algebras play an important role in particle physics (cf. [3]). Assume $\mathcal{A}$ has been represented by a basis $B = \{b_1, \ldots, b_n\}$, along with n^3 structure constants $\delta_{ijk} \in \mathbf{F}$ such that

$$b_i b_j = \sum_{k=1}^{n} \delta_{ijk} b_k \tag{1}$$

for all $1 \leq i, j \leq n$. Products of arbitrary linear combinations of basis elements can then be computed using these rules along with properties 1–3.

Let $C = \{c_1, \ldots, c_n\}$ be another basis of $\mathcal{A}$. One might want to change bases, for example, when elements of $\mathcal{A}$ are represented as nonassociative polynomials and a more "simplified" basis is desirable. Now suppose we are given a matrix $Q = [q_{ij}]$ such that

$$c_j = \sum_{i=1}^{n} q_{ij} b_i$$

for $j = 1 \ldots n$. Expressed another way,

$$[b_1, b_2, \ldots, b_n]Q = [c_1, c_2, \ldots, c_n].$$

We will express a linear combination v of the C by writing $v = [c_1, c_2, \ldots, c_n]v^C$, where v^C is a column of scalars. Q is called the C-to-B transformation matrix because

$$[c_1, c_2, \ldots, c_n]v^C = ([b_1, b_2, \ldots, b_n]Q)v^C = [b_1, b_2, \ldots, b_n](Qv^C)$$

shows that $v^B = Qv^C$ are the coefficients of v relative to B. *The problem that we wish to solve is finding the structure constants relative to the new basis C.* That is, we wish to find the n^3 scalars γ_{ijk} such that

$$c_i c_j = \sum_{k=1}^{n} \gamma_{ijk} c_k$$

for all $1 \leq i, j \leq n$.

2 Straightforward Approach

A straightforward algorithm for computing the γ_{ijk} is to first compute Q^{-1}, the transformation matrix for B-to-C. Now write

$$\begin{aligned} c_i c_j &= (\sum_k q_{ki} b_k)(\sum_t q_{tj} b_t) = \sum_{k,t} q_{ki} q_{tj} (b_k b_t) \\ &= \sum_{k,t} q_{ki} q_{tj} (\sum_m \delta_{ktm} b_m) = \sum_m (\sum_{k,t} q_{ki} q_{tj} \delta_{ktm}) b_m, \end{aligned}$$

obtaining a linear combination $[b_1, b_2, \ldots, b_n] v^B$ over B. Thus

$$c_i c_j = [b_1, b_2, \ldots, b_n] v^B = [c_1, c_2, \ldots, c_n](Q^{-1} v^B),$$

and so $Q^{-1} v^B$ gives us the n structure constants for the product $c_i c_j$. It is easy to see that each such computation requires at least n^3 arithmetic operations, since each δ_{ktm} is multiplied. Because there are n^2 products $c_i c_j$, this method takes $\Omega(n^5)$ arithmetic operations. We now improve this.

3 Fast Solution

The axioms for an algebra imply that for any fixed $x \in \mathcal{A}$ the function

$$L_x : \mathcal{A} \rightarrow \mathcal{A}$$

defined by $L_x(v) = xv$ is a linear transformation on $\mathcal{A}$. In particular, the maps L_{b_i} and L_{c_i} are linear transformations on $\mathcal{A}$. If T is any linear transformation on $\mathcal{A}$ we let $M^B(T)$ and $M^C(T)$ denote the matrix of T relative to the bases B and C, respectively. For example, if $v = [c_1, c_2, \ldots, c_n] v^C$ then we have $T(v) = [c_1, c_2, \ldots, c_n](M^C(T) v^C)$.

Now consider the matrix $M^B(L_{b_i})$. By Equation (1)

$$M^B(L_{b_i}) = \begin{bmatrix} \delta_{i11} & \ldots & \delta_{ij1} & \ldots & \delta_{in1} \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ \delta_{i1n} & \ldots & \delta_{ijn} & \ldots & \delta_{inn} \end{bmatrix}. \tag{2}$$

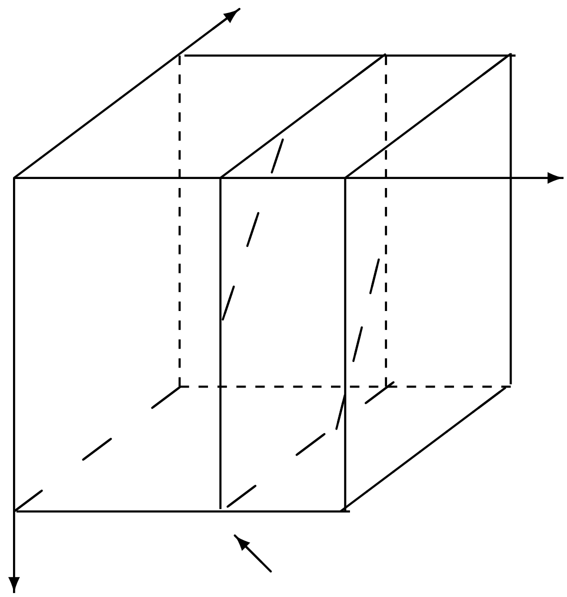

Figure 1: Input of structure constants.

We construct a three-dimensional array of structure constants δ_{ijk} called the *δ-cube*. This is shown in Figure 1. Note that by Equation (2), the slices of the cube parallel to the jk-plane are exactly the matrices $M^B(L_{b_i})$. *The problem of finding the new structure constants γ_{ijk} amounts to constructing the corresponding γ-cube. This in turn, amounts to computing all n of the matrices $M^C(L_{c_i})$.*

Our algorithm now consists of three steps. In Step 0 we invert Q, obtaining Q^{-1}. Recall (cf. [2], p. 286) that for any linear transformation T we have

$$M^C(T) \;=\; Q^{-1}\; M^B(T)\; Q.$$

In particular, for each i, we have

$$M^C(L_{b_i}) \;=\; Q^{-1}\; M^B(L_{b_i})\; Q.$$

Step 1 is shown in Figure 2. We multiply each of the matrices $M^B(L_{b_i})$ on the left by Q^{-1} and then on the right by Q. Step 2 is shown in Figure 3. Since $c_i = \sum_{k=1}^{n} q_{ki} b_k$ it follows

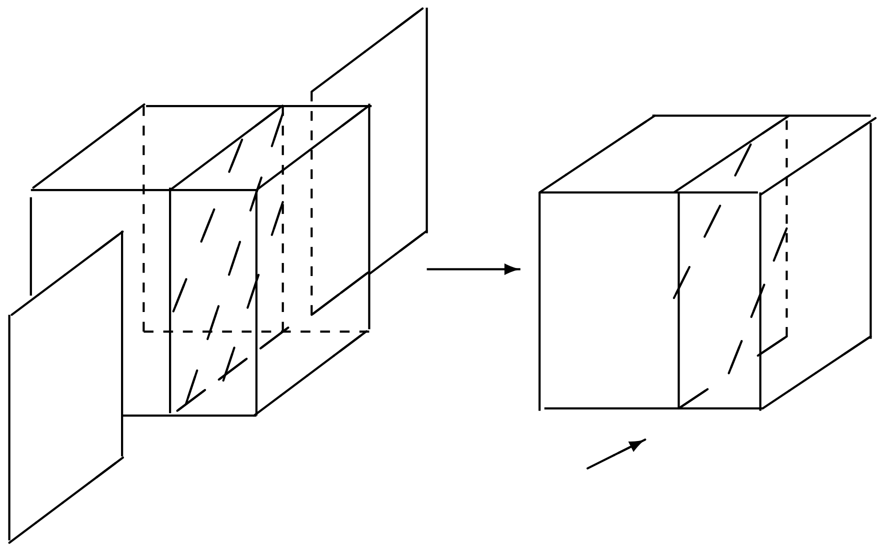

Figure 2: Step 1 of algorithm.

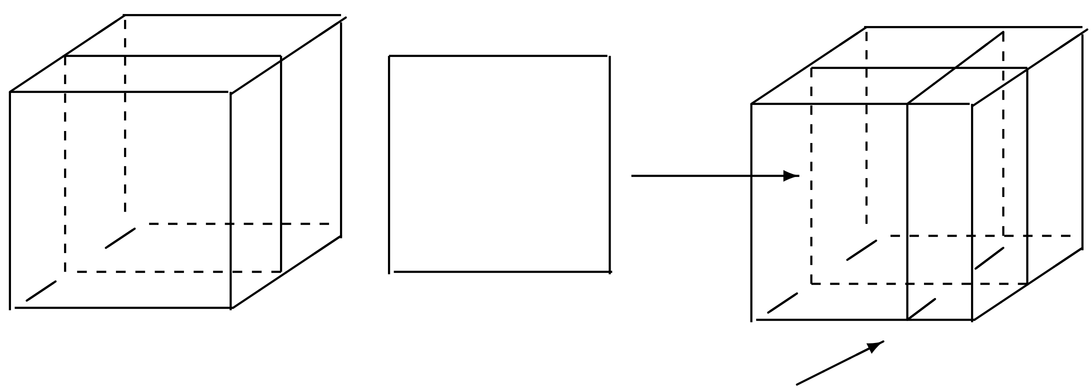

Figure 3: Step 2 of algorithm.

that

$$M^C(L_{c_i}) \ = \ \sum_{k=1}^{n} q_{ki} M^C(L_{b_k}).$$

This says that we can compute the $M^C(L_{c_i})$ by forming linear combinations of the slices formed in Step 1. But note that the coefficients of these linear combinations are exactly columns of Q. Therefore these linear combinations can be computed by repeatedly multiplying the cube obtained in Step 1 on the right by Q, but this time parallel to the ik-plane. Our algorithm follows.

procedure fastchange(**var** Cube; Q);
inputs Cube : cube of structure constants as in Figure 1
Q : transformation matrix for a new basis
output Cube : cube of structure constants relative to new basis
begin
0. Compute Q^{-1}.
1. **for** $i := 1$ to n
Multiply each $M^B(L_{b_i})$ (in Cube) by Q^{-1} and Q as in Figure 2.
2. **for** $j := 1$ to n
Multiply j^{th} slice (of Cube) parallel to ik-plane, by Q as in Figure 3.
end;

4 Analysis

For simplicity, our analysis assumes all scalar operations have unit cost. This assumption is valid, for example, when $\mathbf{F}$ is finite. However, it is clear that the key idea and benefits of our algorithm apply to any field.

Theorem: *Assume that two $n \times n$ matrices can be multiplied in time $O(n^\omega)$. Then given the structure constants for an n-dimensional algebra $\mathcal{A}$, and given a transformation matrix Q, we can compute the structure constants relative to the new basis in $O(n^{\omega+1})$ scalar operations.*

Proof: Consider the algorithm is depicted above. In Step 0, the matrix Q^{-1} must first be computed. This can be done using a straightforward $O(n^3)$ method. In Step 1, $2n$ matrix multiplications are performed, and so this takes time $O(n^{\omega+1})$. Step 2 also takes time $O(n^{\omega+1})$, since it involves n matrix multiplications. □

By merely using the traditional $O(n^3)$ method of matrix multiplication we obtain an $O(n^4)$ algorithm, an improvement over the straightforward method described earlier. However by using the $O(n^{2.376})$ method of Coppersmith and Winograd [1] we have

Corollary: *Structure constants can be found in time $O(n^{3.376})$.*

It is easy to see that the order of Steps 1 and 2 can be reversed. Also, by the associativity of matrix multiplication, the multiplications in Step 1 can be performed as $(Q^{-1}M^B(L_{b_i}))Q$ or $Q^{-1}(M^B(L_{b_i})Q)$. Note that in Step 2 the n matrix multiplications by Q are independent and can be performed in parallel. Step 1 can also be parallelized by performing the multiplications by Q^{-1} in parallel followed by the multiplications by Q in parallel.

References

[1] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions", Proceedings of 19th Annual ACM STOC (1987), pp. 1-6.

[2] S. MacLane and G. Birkhoff, *Algebra*, MacMillan Company, 1967, New York.

[3] H.C. Myung and M. Ruggero, "Foundations of the hadronic generalization of the atomic mechanics III. Bimodular-genotopic Hilbert space formulation of the interior strong problem", Proceedings of the First International Conference on Nonpotential Interactions and their Lie-Admissible Treatment, Part C, Orléans, 1982, Hadronic Journal, 5 (1981/82), no. 4, 1367-1403.

[4] R.D. Schafer, *An Introduction to Nonassociative Algebras*, Academic Press, 1966, New York.