Structure and Efficient Computation of Multiplication Tables and Associated Quadratic Forms

Hoon Hong
Josef Schicho
Wolfgang Stöcher
Research Institute for Symbolic Computation
Johannes Kepler University, 4040 Linz, Austria

December 1, 1995

Abstract

We characterize the multiplication table of an algebra with a multiplicative unity and derive properties of multiplication tables and associated quadratic forms, which allow efficient computation. We compare several ways of computing a multiplication table and associated quadratic forms. By index permutation the complexity of the computation time of associated quadratic forms can be reduced. Exploiting the structure concerning equal entries in the multiplication table gives a further speedup. We apply our results to the case of the factor ring $K[\bar{x}]/I$ and give speedups in the appendix.

1 Domain and Motivation

Let A be a finite dimensional vector space over a field K and let $\times$ be a commutative and associative operation ("*multiplication*") such that $(A, \times)$ becomes a commutative algebra (associativity is silently assumed here). Furthermore let $\times$ have a unity ("1") throughout. Instead of $a \times b$, for $a, b \in A$, we shall simply write ab. Given a basis $V = \{v_1, \dots v_d\}$ of A the operation $\times$ induces a multiplication table $w_{ij}^k, 1 \le i, j, k \le d$, given by

$$v_i v_j = \sum_{k=1}^{d} w_{ij}^k v_k. \tag{1}$$

In the following we will omit the symbol $\sum$ for summation. Instead an index appearing both as super-index and subindex will indicate summation over this index (*Einstein sum convention, tensor notation*). The range of the index will here always be d, the dimension of A. Hence (1) can be written simply as $v_i v_j = w_{ij}^k v_k$.

Any such algebra is isomorphic to a factor ring $A := K[\bar{x}]/I$, where $\bar{x}$ is an abbreviation for $x_1, \dots x_r$ and I a zero-dimensional ideal (see the appendix for a proof). The need to compute the multiplication table and associated quadratic forms for a factor ring comes from an algorithm counting common real zeros of a polynomial system ([Pederson et al. 94]).

2 Multiplication Table

2.1 Characterization

Since $\times$ is commutative and associative and has a multiplicative unity, for the multiplication table several properties hold. Actually a multiplication table can be characterized by the following lemma.

Lemma 2.1 *A three dimensional matrix $w_{ij}^k, i, j, k = 1 \dots d$, is a multiplication table of an algebra $(A, \times)$ with a multiplicate unity, if and only if the following properties hold:*

1. $w_{ij}^k = w_{ji}^k$,

2. $w^k_{j_1 i} w^l_{kj_2} = w^l_{j_1 k} w^k_{ij_2}$ *(or, combining it with the above property, in another version:* $w^k_{ij_1} w^l_{kj_2} = w^k_{ij_2} w^l_{kj_1}$*)*
3. *For some* $u^1, \ldots u^d \in K$ *we have* $u^k w^j_{ki} = \delta_{ij}$.

Proof: Let us first start with a commutative algebra $(A, \times)$ with a multiplicative unity, and show that the corresponding multiplication table fulfills the stated properties. $[v_k]a$ shall denote the coefficient of v_k in the representation of $a \in A$ w.r.t. basis V. Then we have because of commutativity of $\times$:

$$w^k_{ij} = [v_k](v_i v_j) = [v_k](v_j v_i) = w^k_{ji}.$$

From associativity we get:

$$w^k_{j_1 i} w^l_{kj_2} = [v_l](w^k_{j_1 i} v_k v_{j_2}) = [v_l]((v_{j_1} v_i) v_{j_2}) = [v_l](v_{j_1}(v_i v_{j_2})) = [v_l](v_{j_1} w^k_{ij_2} v_k) = w^l_{j_1 k} w^k_{ij_2}.$$

Since $1 \in A$ there exist $u^1, \ldots u^d \in K$ such that $1 = u^k v_k$. Therefor we get:

$$u^k w^j_{ki} = [v_j] u^k v_k v_i = [v_j] 1 v_i = \delta_{ij}.$$

Now let us start from a three dimensional matrix w^k_{ij}, for which the three properties hold. Then we construct an algebra $(A, \times)$ by $A := <v_1, \ldots v_d>$ and $v_i \times v_j := w^k_{ij} v_k$. By the above equations $(A, \times)$ is a commutative and associative algebra with a multiplicative unity, which has the given multiplication table. □

Because of the isomorphism between a finitely generated commutative algebra with a multiplicative unity and a corresponding factor ring this lemma holds in the same way for factor rings.

2.2 Computation

For fixed i, j the vector w^k_{ij}, $1 \le k \le d$, contains the coefficients of the basis representation of $v_i v_j$. In case of a factor ring algebra the basis representation is computed via the Groebner basis of the ideal I. By building V out of the monomials in the Groebner staircase the normal form of $v_i v_j$ is a linear combination of the basis elements (see [Buchberger 85] and [Cox et al. 92] for more details).

Now let us go into details on how to make the computation efficient. First because of commutativity of multiplication we have $w^k_{ij} = w^k_{ji}$, $1 \le k \le d$ and it is sufficient to do the computations for $1 \le i \le j \le d$. The following two improvements are more sophisticated and dependent on the structure of the algebra.

Equal Products

Among the pairs of basis elements (v_i, v_j) with $1 \le i \le j \le d$ there might still be several for which $v_{i_1} v_{j_1} = v_{i_2} v_{j_2}$. Especially in the case of a factor ring algebra using a monomial basis this is very likely to happen and exploiting this fact can reduce the order of computation time and memory usage. This is not only important for the computation of the multiplication table, but also for computing associated quadratic forms, as lemma 3.1 will show.

To avoid multiple reduction of the same polynomial and to keep the structural information for later use in a *structure matrix Qs* we proceed as follows:

```
DistCtr := 0 // counting distinct products of basis elements
for i := 1 to d do
  for j := i to d do
    if w_ij is not already computed
      DistCtr := DistCtr + 1
      Qs_ij := Qs_ji := DistCtr
      p := v_i v_j
      compute the basis representation of p and extract its coefficients into w^k_ij and w^k_ji.
      for all pairs (l, m) with v_l v_m = p
        Qs_lm := Qs_ml := DistCtr
        w^k_lm := w^k_ml := w^k_ij, 1 <= k <= d.
```

The gain heavily depends on the basis and the structure of the algebra. See [Hong, Stöcher 95] for more details when applied to factor rings.

Recursive Computation

Consider $v_i = v_{i_1} v_{i_2}$ such that $1 < i_1 \leq i_2 < i$. Then $v_i v_j = v_{i_1}(v_{i_2} v_j)$ and hence we have the recurrence formula:

$$w_{ij}^k = w_{i_1 l}^k w_{i_2 j}^l.$$

By the above described order in computing the multiplication table w, the entries $w_{i_1 l}^k, 1 \leq l \leq d$, and $w_{i_2 j}^k$ will already be computed when coming to w_{ij}^k. This gives the following modified algorithm:

$DistCtr := 0$
for $i := 1$ to d do
  $reducible :=$ false
  if there exist $1 < i_1 \leq i_2$, s. t. $v_i = v_{i_1} v_{i_2}$
    $reducible :=$ true
  for $j := i$ to d do
    if w_{ij} is not already computed
      $DistCtr := DistCtr + 1$
      $Qs_{ij} := Qs_{ji} := DistCtr$
      $p := v_i\, v_j$
      if $(reducible)$
        for $k := 1$ to d do
          compute $w_{ij}^k = w_{ji}^k$ by summing up $w_{i_1 l}^k w_{i_2 j}^l$
      else
        compute the basis representation of p and extract its coefficients into w_{ij}^k and w_{ji}^k.
      for all pairs (l, m) with $v_l v_m = p$
        $Qs_{lm} := Qs_{ml} := DistCtr$
        $w_{lm}^k := w_{ml}^k := w_{ij}^k, 1 \leq k \leq d.$

The gain of this way to compute the multiplication table depends on whether and how much time is saved by using the recurrence formula instead of computing the basis representation of p.

3 Associated Quadratic Forms

3.1 Notation

Definition 3.1 *For a commutative and associative algebra $(A, \times)$ defined as above, we introduce the following functions:*

- $m_f : A \longrightarrow A,\ g \mapsto m_f(g) = fg.$
- $B^{<h>} : A \times A \longrightarrow K,\ (f, g) \mapsto B^{<h>}(f, g) = \mathrm{Tr}(m_{fgh})$, *where Tr is the trace function.*

The function $g \mapsto m_f(g)$ is called the multiplication endomorphism induced by f. The function $f \mapsto B^{<h>}(f, f)$ is called the associated quadratic form *for h.*

In a given Basis V for A the matrices for m_f and $B^{<h>}$ shall be denoted by m_f^* and $Q^{<h>}$, respectively. B and Q will be used as abbreviations for $B^{<1>}$ and $Q^{<1>}$, respectively.

3.2 Computation

There are several ways to compute the matrix $Q^{<h>}$. For instance Q can be computed in the following two ways:

1. $Q_{ij} = w_{il}^k w_{kj}^l$
2. $Q_{ij} = w_{ij}^k w_{kl}^l$

That both formulas give the same result stems from the second property of a multiplication table (see lemma 2.1)., which allows that kind of index permutation. A direct derivation of the second formula can be found in the appendix.

Table 1: Comparing several ways of computing assoc. quadr. forms

formula	# multipl.	# extra storage
1 (*simple*)	d^4	
2 (*perm. simple*)	d^2+d^3	d
3 (*trace*)	d^2+d^3	d
4 (*endom.*)	d^3+d^3	d^2

Both formulas have two summation indices, giving d^2 multiplications for each matrix element. However, in the second formula the expression $t_k := w_{kl}^l$ (summed up over l) can be computed in advance. Hence we have

$$Q_{ij} = w_{ij}^k\, t_k, \tag{2}$$

which contains only one summation index and thus involves only d multiplications for each matrix element.

For the general case of computing $Q^{<h>}$ there are even more ways:

1. $Q_{ij}^{<h>} = w_{kl}^k\, w_{mj}^l\, w_{in}^k\, h^n = Q_{kj}\, w_{in}^k\, h^n$ (*simple* formula)
2. $Q_{ij}^{<h>} = w_{ij}^k\, w_{kn}^m\, w_{ml}^l\, h^n = w_{ij}^k\, w_{kn}^m\, t_m h^n = w_{ij}^k\, Q_{kn}\, h^n$ (*permuted simple* formula)
3. $Q_{ij}^{<h>} = w_{ij}^k\, t_k^{<h>}$ (*trace* formula),
   where $t_k^{<h>} := \mathrm{Tr}(m_{hv_k}^*)$ can be either computed directly (see above or using 6) or by using one of the recurrence formulas:
   $$t_k^{<h_1h_2>} = Q_{kn}^{<h_1>}\, h_2^n \tag{3}$$
   $$t_k^{<h_1h_2>} = (m_{h_2}^*)_{lk}\, t_l^{<h_1>} \tag{4}$$
4. $Q^{<h_1h_2>} = (m_{h_2}^*)^T\, Q^{<h_1>}$ (*endomorphism* formula)

Before evaluating the efficiency of the different ways of computing associated quadratic forms let us show their correctness.

Correctness

In the appendix the third formula together with the first recurrence formula (3) for $t^{<h>}$ is derived directly. The second recurrence formula (4) can be shown in an analogous way. From (3) we get by setting $h_1 = 1$ and $h_2 = h$

$$t_k^{<h>} = Q_{kn}h^n,$$

which proofs the second formula for $Q_{ij}^{<h>}$. Now by two index permutations (see the second property of a multiplication table in lemma 2.1) as done above for computing Q_{ij} one gets the first formula[1]. The remaining fourth way of computing $Q_{ij}^{<h>}$ can be shown using the third one, and plugging in (4).

Evaluation of Efficiency

Let us first assume that the basis representation of h is known. The first and the second formula for $Q_{ij}^{<h>}$ both contain two summation indices, involving d^2 multiplications each. Hence the total number of multiplications for computing $Q^{<h>}$ is d^4. In the second formula, however, the expressions $t_k^{<h>} = Q_{kn}h^n$ can be computed in advance, involving altogether d^2 multiplications and requiring the storage of additional d values. Then each matrix element can be computed via d multiplications, giving a total number of d^3 multiplications. In the case that the basis representation of h is not known, but computation for factors h_1, h_2 have already been performed, one can use the third formula together with (3), giving the same complexity.

The fourth formula first involves the computation of $m_{h_2}^*$, i. e. d^3 multiplications and extra storage of d^2 elements. The computation of $Q^{<h>}$ then requires another d^3 multiplications. See table 1 for an overview.

[1]The first (*simple*) formula is actually the first formula we had derived. Only after finding out about the "*index permutation* law" we came across the other ways to compute associated quadratic forms.

Table 2: Heuristics on computing assoc. quadr. forms using 'equal products'

formula	# multipl.
1 (*simple*)	$O(d^3)$
2 (*perm. simple*)	$d^2 + O(d^2)$
3 (*trace*)	$d^2 + O(d^2)$
4 (*endom.*)	$d^3 + O(d^2)$

Equal Products

Orthogonal to the choice of an efficient formula for computing $Q^{<h>}$ is the observation of equal products of basis elements. From the third formula or (8) we can derive the following lemma:

Lemma 3.1 *Let w_{ij}^k be the multiplication table of an algebra $(A, \times)$ w.r.t. the basis $V = \{v_1, \dots v_d\}$, and $Q^{<h>}$ as defined above. Then for each quadruple $(i_1, j_1, i_2, j_2), 1 \leq i_1, j_1, i_2, j_2 \leq d$, for which $v_{i_1}v_{j_1} = v_{i_2}v_{j_2}$, we have :*

1. $w_{i_1 j_1}^k = w_{i_2 j_2}^k, \quad 1 \leq k \leq d,$
2. $Q_{i_1 j_1}^{<h>} = Q_{i_2 j_2}^{<h>}, \quad h \in A.$

Given the multiplication table w and the information about equal products in the structure matrix Qs, we can compute the associated quadratic form Q using the 2. formula as follows:

for $k := 1$ to d do
    $t_k := \sum_l w_{kl}^l$
$DistCtr := 0$ // counting the number of computed distinct matrix elements
for $i := 1$ to d do
    for $j := i$ to d do
        $RefNum := Qs_{ij}$ // requested index in the DistinctElements table
        if ($DistCtr < RefNum$)
            $DistinctElements[RefNum] := \sum_k w_{ij}^k t_k$
            $DistCtr := DistCtr + 1$ // by construction of Qs we now have $DistCtr = RefNum$
        $Q_{ij} = DistinctElements[RefNum]$

Because of lemma 3.1 and relation (11) we can compute $Q^{<h>}$ from Q or from $Q^{<h_1>}$ with $h = h_1 h_2$ in an analogous way.

Heuristics ([Hong, Stöcher 95]) and test runs indicate that in the general case the number of distinct elements in $Q^{<h>}$ is of order $O(d)$. Hence the comparison of the different ways of computing evaluates to the picture given in table 2.

4 Conclusion

We derived properties of a multiplication table, which are useful for manipulating and generating different formulas for computing associated quadratic forms. The formulas differ as well in the order of complexity for computation time as in the order of complexity of memory usage. Orthogonal to the different formulas is the observation of a certain structure of associated quadratic forms $Q^{<h>}$ not depending on h, i. e. equal entries stemming from equal products of basis elements. This gives a further speedup and also seems to reduce the order of required multiplications. These observations match the timings given in the appendix for the case of a factor ring algebra.

A Proof of Isomorphism Lemma

Lemma A.1 *Let A be a finite dimensional vector space over the field K and let $(A, \times)$ be a commutative algebra with a multiplicative unity. Then there exist $\bar{x} = x_1, \dots x_r$ and a zero-dimensional ideal I, such that $A = K[\bar{x}]/I$.*

Proof: Let d, V and w_{ij}^k be defined as above. We assume w.l.o.g. that v_d is 1, the one-element of $\times$. Now we will show that $\bar{x} = x_1, \ldots x_{d-1}$ and I being the ideal generated by $<x_i x_j - w_{ij}^k x_k, x_d - 1>$ in $K[x_1, \ldots x_d]$ and projected onto $K[x_1, \ldots x_{d-1}] = K[\bar{x}]$ are suitable settings.

Let us consider the following K-Homomorphism:

$$\phi : K[\bar{x}] \to A$$

such that

$$x_i \mapsto v_i, \text{for} 1 \leq i \leq d-1, \quad 1 \mapsto v_d.$$

Obviously ϕ is surjective. We will show now that the kernel of ϕ is I as defined above. Then we know by the first theorem of isomorphism that $A \cong K[\bar{x}]/I$, what we want to proof.

Since

$$\ker(\phi) = \phi^{-1}(0) = \{p \in K[\bar{x}] | p(v_1, \ldots v_{d-1}) = 0\},$$

we immediately have by the definition of w_{ij}^k that $I \subset \phi^{-1}(0)$. It is left to show that $I \supset \phi^{-1}(0)$. Consider $a \in \phi^{-1}(0)$, i. e. $a \in K[\bar{x}]$ and $\phi(a) = a(v_1, \ldots v_{d-1}) = 0$. Since I contains by definition all quadratic terms, there exists $\tilde{a} \in K[\bar{x}]$ of the form $\tilde{a} = a_1 x_1 + \cdots a_{d-1} x_{d-1} + a_d$ such that $a - \tilde{a} \in I$. Since $\phi(a) = \phi(I) = 0$ we also have $\phi(\tilde{a}) = a_1 v_1 + \cdots a_{d-1} v_{d-1} + a_d = 0$, which can only hold for $a_1 = \cdots a_d = 0$. Hence $\tilde{a} = 0$ and therefor $a \in I$. $\square$

B Formulas for Computing Associated Quadratic Forms

In order to compute the matrix elements of $Q^{<h>}$ we will have to derive some relations between the multiplication table and the just defined functions. For $f, g \in A$ we have

$$m_f(g) = fg = (f^i v_i)(g^j v_j) = f^i \, (v_i v_j) \, g^j = f^i \, w_{ij}^k \, g^j \, v_k,$$

from which we can extract

$$(fg)^k = f^i \, w_{ij}^k \, g^j, \tag{5}$$

and the matrix m_f^* for the endomorphism m_f given by

$$\left(m_f^*\right)_{kj} = f^i w_{ij}^k. \tag{6}$$

In the special case of a basis element we have

$$\left(m_{v_k^*}\right)_{ij} = w_{kj}^i. \tag{7}$$

Using (6) and (7) we get

$$B^{<h>}(f,g) = \mathrm{Tr}\left(m_{fgh}^*\right) = \mathrm{Tr}\left(m_{fg}^* m_h^*\right) = \mathrm{Tr}\left((fg)^k m_{v_k}^* m_h^*\right) = (fg)^k \, \mathrm{Tr}(m_{hv_k}^*).$$

Plugging in (5) gives

$$B^{<h>}(f,g) = f^i \, w_{ij}^k \, \mathrm{Tr}(m_{hv_k}^*) \, g^j,$$

from where we can extract the matrix

$$Q_{ij}^{<h>} = w_{ij}^k \, \mathrm{Tr}(m_{hv_k}^*), \tag{8}$$

such that

$$B^{<h>}(f,g) = (f^*)^T \, Q^{<h>} \, g^*.$$

Let us define

$$t_k^{<h>} := \mathrm{Tr}(m_{hv_k}^*), \tag{9}$$

and use t_k as an abbreviation for $t^{<1>}$. Using (7) we get

$$t_k = \mathrm{Tr}(m_{v_k}^*) = \sum_l w_{kl}^l, \quad 1 \leq k \leq d.$$

Table 3: Timings for computing the multiplication table w

GB data	#(all pairs)	#(dist. pairs)	ratio	comp. all pairs (plain version)	comp. dist. pairs (equal products)	speedup
4 var/deg 2	136	31	4.39	26.8 s	8 s	3.35
3 var/deg 3	105	63	1.67	.86 s	.63 s	1.37

Table 4: Timings for computing Q in ms

Data	simple	trace (without Qs)	trace (using Qs)
4 var/deg 2	430	90	30
3 var/deg 3	780	70	50

Now from (8) and setting $h := 1$ we have

$$Q_{ij} = w_{ij}^{k}\, t_k. \tag{10}$$

For efficiently computing the quadratic form $Q^{<h>}$ we will deduce a more general equation for $Q^{<h_1 h_2>}$. Using the basis representation of h_2 and (1)we have

$$h_1 h_2 v_k = h_1 h_2^n v_n v_k = w_{nk}^{l} h_2^n h_1 v_l.$$

Plugging this into (9) for $h = h_1 h_2$ gives

$$t_k^{<h_1 h_2>} = \mathrm{Tr}(w_{nk}^{l} h_2^n m_{h_1 v_l}) = w_{nk}^{l} \mathrm{Tr}(m_{h_1 v_l}) h_2^n,$$

and using (8) from right to left for $h = h_1$ we get a recursion formula for $Q^{<h>}$:

$$Q_{ij}^{<h_1 h_2>} = w_{ij}^{k} Q_{kn}^{<h_1>} h_2^n. \tag{11}$$

C Timings

We implemented several versions for computing the multiplication table and associated quadratic forms for the factor ring algebra $K[\bar{x}]/I$, where I is a zero-dimensional ideal given by its normed and reduced Groebner basis. The implementation was done in C++ using the STURM library ([Hong et al. 94]).

Multiplication Table

The recursive computation only gave speedups in a more sophisticated and specific modification (in the case that the leading coefficients of the Groebner basis are ± 1), so we did not consider it here. We also give the number of all pairs and the number of pairs of basis elements with distinct products.

Associated Quadratic Forms

We compare the two formulas given for Q. The first one is referred to as *primitive*, the second one, in the improved version using t_k, as *trace* formula. In our experiments we compared the *simple* and the *trace* formula, without and with using the observation of equal products, i. e. the structure information in Qs.

References

[BKR 86] M. Ben-Or, D. Kozen, J. Reif *The complexity of elementary algebra and geometry*, J. of Computation and System Sciences 32 251-264 (1986).

[Buchberger 85] B. Buchberger *Gröbner: An algorithmic method in polynomial ideal theory* in Multidimensional Systems Theory (N. K. Bose Ed.), Chapter 6, D. Reidel (1985).

[Cox et al. 92] D. Cox, J. Little, D. O'Shea *Ideals, Varieties and Algorithms - An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer New York (1992).

[Hong et al. 94] H. Hong, A. Neubacher, V. Stahl. *The STURM Library Manual - A C++ Library for Symbolic Computation* Technical Report 94-30, RISC Linz, J. Kepler University, Linz, Austria, 1994.

[Pederson et al. 94] P. Pederson, M.-F. Roy, A.Szpirglas *Counting Real Zeros in the multivariate case*, Universite de Rennes, France.

[Hong, Stöcher 95] H. Hong, W. Stöcher *Implementing Multivariate Real Root Count with Polynomial Constraints*, RISC-Linz Report Series No. 95-50, RISC Linz, J. Kepler University Linz, Austria.