AT&T Bell Laboratories

subject: **Rank of N-Way Arrays and the Geometry of 2x2x2 Arrays**

date: **February 1985**

from: **Joseph B. Kruskal**
MH 11215
2C-281 x3853

TM

MEMORANDUM FOR FILE

INTRODUCTION

The concept of rank, which is fundamental to the theory of matrices, can be extended to many-way arrays of all orders. This extension is so useful and natural, particularly for 3-way arrays, that it has been introduced on several separate occasions for different purposes.

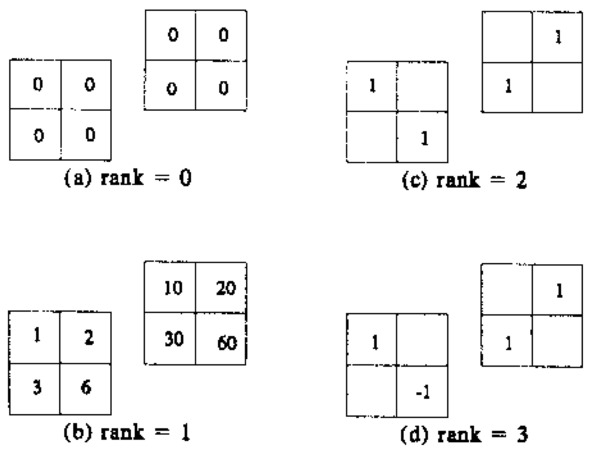


FIGURE 1

Furthermore, if we embed the arrays of order N in those of order $N + 1$ in the obvious way, then rank is preserved by the embedding. For example, the rank of an $I \times J$ matrix $x_{..}$ is the same as the rank of the $I \times J \times 1$ array $y_{...}$ defined by $y_{ij1} = x_{ij}$. Thus there is only a single unified concept of rank for all arrays, not a separate concept for the arrays of each order.

The smallest 3-way arrays which are not effectively the same as matrices are $2 \times 2 \times 2$. The maximum rank of such arrays is three. Examples of each possible rank are shown in Figure 1. In the 8-dimensional space of $2 \times 2 \times 2$ arrays, almost all arrays have rank 2 or rank 3 (i.e., the arrays of other ranks have Lebesgue measure 0, and both rank 2 and rank 3 arrays occur with nonzero measure). The rank of $2 \times 2 \times 2$ arrays is explored in considerable detail later on. The maximum rank of a $2 \times J \times J$ array is $\lfloor 3J/2 \rfloor$. More generally, if $J \leq K$, the maximum rank of a $2 \times J \times K$ array is $J + \min(J, \lfloor K/2 \rfloor)$. This bound is sharp.

Currently, the most common use of rank in mathematics is in connection with computational complexity theory. Suppose we have a collection of bilinear forms. One matrix describes each form, so a 3-way array describes the entire collection. A fundamental theorem states, approximately, that the minimum number of multiplications needed to calculate all the forms is the rank of the 3-way array. For example, the product of two complex numbers is described by the array of Figure 1(d), so the minimum number of real multiplications required to form the product is 3. (This and the following example are explained in more detail in a subsequent section.) The product of two 2×2 matrices is described by a $4 \times 4 \times 4$ array containing eight 1's and all other entries 0, and this array turns out to have rank 7, so the minimum number of multiplications needed to form this product is 7. The product of two 3×3 matrices is described by a $9 \times 9 \times 9$ array containing twenty-seven 1's and all other entries 0, and this array turns out to have rank between 17 and 23, so the minimum number of multiplications needed to form this product is between 17 and 23.

My own interest in rank of 3-way arrays arises from another source. There are two useful statistical models important in psychometrics, PARAFAC and INDSCAL, in which 3-way arrays of data are approximated by arrays of low rank. The use of these models leads to a number of interesting questions, such as what is the maximum rank of an $I \times J \times K$ array?

In this paper, rank is first extended from matrices to 3-way arrays, and then to arrays of all orders, and some basic general facts are presented. Then a little computational complexity theory and some other applications of array rank are discussed briefly, to provide further motivation for the study of rank. Next, the theory of 3-way arrays is illustrated by selected results and some open questions. Finally, the rank and geometry of $2 \times 2 \times 2$ arrays is analyzed in some detail. Many of the definitions and results apply to arrays over an arbitrary field, but the more detailed results, including the analysis of $2 \times 2 \times 2$ arrays, is restricted to arrays over the real numbers.

RANK OF 3-WAY AND N-WAY ARRAYS

Among the many different but equivalent definitions for the rank of a matrix over an arbitrary field, the definition which most conveniently permits extension is *dyadic rank*. A matrix $d_{..}$ is a *dyad* if it can be written as an outer product of two vectors, $d_{..} = a_{.} \otimes b_{.}$, i.e., it has the form of a multiplication table, $d_{ij} = a_i b_j$. Concisely stated, the dyadic rank of $x_{..}$ is the minimum number of dyads whose sum is $x_{..}$, i.e., the minimum number R of dyads $d_{..}^{(r)}$ for which $x_{..} = \sum_{r=1}^{R} d_{..}^{(r)}$. Since the entries of $d_{..}^{(r)}$ can be written in the form $d_{ij}^{(r)} = a_{ir} b_{jr}$, the dyadic rank of $x_{..}$ is the minimum R for which we can write $x_{ij} = \sum_{r=1}^{R} a_{ir} b_{jr}$, i.e., $x_{..} = a_{..} b'_{..}$ where $b'_{..}$ is the transpose of $b_{..}$ (and the ordinary matrix product is indicated by proximity). We refer to the expression on the right as a *(bilinear) decomposition of* $x_{..}$ *of rank* R. Also, we refer to the $d_{..}^{(r)} = a_{.r} \otimes b_{.r}$ as the dyads of the decomposition, and to the matrices $a_{..}$ and $b_{..}$ as the *factors* or the *loading matrices* of the decomposition. Then *the dyadic rank of* $x_{..}$ *is the minimum rank of any (bilinear) decomposition*. This is the form of the definition that we prefer to use. Since the decomposition above is just a matrix product, we shall feel free to write decompositions in matrix form. Thus the rank of $x_{..}$ is also the minimum value of R such that $x_{..} = a_{..} b'_{..}$ for properly chosen matrices $a_{..}$ and $b_{..}$ having R columns each.

Suppose now that $y_{...}$ is an $I \times J \times K$ array, with entries y_{ijk}. (It can be pictured as a rectangular parallelopiped.) To define the *triadic rank* of $y_{...}$, we first define an array $t_{...}$ to be a *triad* if it can be written as an outer product of three vectors, $t_{...} = a_{.} \otimes b_{.} \otimes c_{.}$, i.e., $t_{ijk} = a_i b_j c_k$. Concisely stated, the triadic rank of $y_{...}$ is the minimum number of triads whose sum is $y_{...}$, i.e., the minimum number R of triads $t^{(r)}_{...}$ for which $y = \sum_{r=1}^{R} t^{(r)}_{...}$. Since the entries of each $t^{(r)}_{...}$ can be written in the form $t^{(r)}_{ijk} = a_{ir} b_{jr} c_{kr}$, the *triadic rank of* $y_{...}$ *is the minimum R for which we can write* $y_{ijk} = \sum_r a_{ir} b_{jr} c_{kr}$. We refer to the expression on the right as a *(trilinear) decomposition of* $y_{...}$ *of rank R*. Also, we refer to the $t^{(r)}_{...} = a_{.r} \otimes b_{.r} \otimes c_{.r}$ as the triads of the decomposition, and to $a_{..}$, $b_{..}$ and $c_{..}$ as the *factors* or the *loading matrices* of the decomposition. Then the *triadic rank of* $y_{...}$ *is the minimum rank of any (trilinear) decomposition.*

It is helpful to introduce a new term. The *triple product* $[a_{..}, b_{..}, c_{..}]$ of three matrices, $a_{..}$, $b_{..}$, and $c_{..}$, is analogous to the ordinary matrix product $a_{..}b'_{..}$, and exists only if $a_{..}$, $b_{..}$, and $c_{..}$ all have the same number of columns. If $a_{..}$, $b_{..}$, and $c_{..}$ are $I \times R$, $J \times R$, and $K \times R$, then $[a_{..}, b_{..}, c_{..}]$ is defined to be the $I \times J \times K$ array $y_{...}$ given by $y_{ijk} = \sum_r a_{ir} b_{jr} c_{kr}$. Obviously, a triple product is just another way of writing a decomposition and we shall use the two terms interchangeably. Thus the rank of any 3-way array $y_{...}$ is the minimum value of R such that $y = [a_{..}, b_{..}, c_{..}]$ for properly chosen matrices $a_{..}$, $b_{..}$, and $c_{..}$ having R columns each.

To avoid confusion with a very different use of the word "rank" which clashes with our use, we compare our terminology in Figure 2 with another widely used terminology. The other use of rank corresponds to our word "order". Thus rank in our sense is entirely unrelated to rank in the other sense. In this paper we always explicitly indicate the order of an array by the number of letters and/or dots in the subscript.

We introduce some simple concepts and some elementary but important facts about triadic rank. A 3-way array has 6 transposes including itself, corresponding to the 6 permutations of the subscripts. It is elementary that transpositions preserve triadic rank. More generally, it is clear

OUR TERMINOLOGY	ANOTHER TERMINOLOGY
array	tensor
order	rank
scalar = 0-way array = array of order 0	scalar = tensor of rank 0
vector = 1-way array = array of order 1	vector = tensor of rank 1
matrix = 2-way array = array of order 2	tensor of rank 2
3-way array = array of order 3	tensor of rank 3
rank	——

FIGURE 2

that all concepts of interest are invariant or have obvious analogues under permutation of subscripts, so we shall henceforth understand that definitions and results hold also for other permutations than those mentioned.

To justify a claim made above we need to show that the triadic rank of a "skinny" 3-way array is the same as the dyadic rank of the corresponding matrix. (A *skinny* array is an array in which at least one index has only a single value.) Thus let $y_{...}$ be an $I \times J \times 1$ array and $x_{..}$ the corresponding $I \times J$ matrix, which is defined by $x_{ij} = y_{ij1}$ for all i,j. The dyadic rank of $x_{..}$ is the minimum rank of any bilinear decomposition, and the triadic rank of $y_{...}$ is the minimum rank of any trilinear decomposition. Consider any decomposition $x_{..} = a_{..}b'_{..}$ of $x_{..}$. Define $c_{..}$ by $c_{kr} = 1$ for $k = 1$ and $r = 1$ to R. Then $y_{...} = [a_{..}, b_{..}, c_{..}]$ is a decomposition of $y_{...}$ having the same rank as $a_{..}b'_{..}$. Similarly, suppose $y_{...} = [a_{..}, b_{..}, c_{..}]$ is any decomposition of $y_{...}$. Define $\bar{b}_{..}$ by $\bar{b}_{jr} = b_{jr} c_{1r}$ for all j,r. Then $x_{..} = a_{..}\bar{b}'_{..}$ is a decomposition of $x_{..}$ having the same rank as $[a_{..}, b_{..}, c_{..}]$. Therefore rank($x_{..}$) = rank($y_{...}$), which justifies the claim, so it is permissible henceforth to drop the modifiers "dyadic" and "triadic" and refer simply to "rank".

Suppose $z_{\cdot\ldots\cdot}$ is an N-way $I_1 \times \ldots \times I_N$ array, with entries $z_{i_1 \ldots i_N}$. To define *N-adic rank*, we first define an N-way array $m_{\cdot\ldots\cdot}$ to be an *N-ad* if it can be written as an outer product, $m_{\cdot\ldots\cdot} = a_{\cdot}^{(1)} \otimes \ldots \otimes a_{\cdot}^{(N)}$. Concisely stated, the N-adic rank of $z_{\cdot\ldots\cdot}$ is the minimum number of N-ads whose sum is $z_{\cdot\ldots\cdot}$, i.e., the minimum number of N-ads $m_{\cdot\ldots\cdot}^{(r)}$ for which $z_{\cdot\ldots\cdot} = \sum_{r=1}^{R} m_{\cdot\ldots\cdot}^{(r)}$. Since the entries of each $m_{\cdot\ldots\cdot}^{(r)}$ can be written in the form $m_{i_1 \ldots i_N}^{(r)} = a_{i_1 r}^{(1)} \ldots a_{i_N r}^{(N)}$, the N-adic rank of $z_{\cdot\ldots\cdot}$ is the minimum R for which we can write $z_{i_1 \ldots i_N} = \sum_{r=1}^{R} a_{i_1 r}^{(1)} \ldots a_{i_N r}^{(N)}$. We refer to the expression on the right as an *(N-linear) decomposition of* $z_{\cdot\ldots\cdot}$ *of rank R*. Also, we refer to the $m_{\cdot\ldots\cdot}^{(r)} = a_{\cdot r}^{(1)} \otimes \ldots \otimes a_{\cdot r}^{(N)}$ as the N-ads of the decomposition, and to the matrices $a_{\cdot\cdot}^{(r)}$ as the *factors* or the *loading matrices* of the decomposition. Then the *N-adic rank of* $z_{\cdot\ldots\cdot}$ *is the minimum rank of any (N-linear) decomposition*. If $a_{\cdot\cdot}^{(n)}$ is $I_n \times R$ for $n = 1$ to N, then we define the N-tuple product $[a_{\cdot\cdot}^{(1)}, \ldots, a_{\cdot\cdot}^{(N)}]$ to be the $I_1 \times \ldots \times I_N$ array $z_{\cdot\ldots\cdot}$ given by $z_{i_1 \ldots i_N} = \sum_{r=1}^{R} a_{i_1 r}^{(1)} \ldots a_{i_N r}^{(N)}$. Obviously, an N-tuple product is just another way of writing a decomposition, and the two terms will be used interchangeably. Thus the rank of any N-way array is the minimum value of R such that $z_{\cdot\ldots\cdot} = [a_{\cdot\cdot}^{(1)}, \ldots, a_{\cdot\cdot}^{(N)}]$ for properly chosen matrices $a_{\cdot\cdot}^{(n)}$ having R columns each.

An N-way array has $N!$ transposes including itself, and they all have the same rank. It is not hard to extend the proof given in the 3-way case to show that the N-adic rank of a "skinny" N-way array is the same as the $(N-1)$-adic rank of the corresponding $(N-1)$-way array. Thus it is permissible to omit the modifier "N-adic" and refer simply to "rank".

ROW RANK AND COLUMN RANK

The concepts of row rank and column rank of a matrix can also be extended to 3-way and N-way arrays. Since an N-way array has N such values, we shall use the notation rank_n ($n = 1$ to N) for these values. Thus ordinary row rank and column rank of matrices can be referred to as rank_1 and rank_2.

It will turn out, as one would wish, that $\text{rank}_n(z_{\cdot\ldots\cdot}) = \text{rank}(z_{\cdot\ldots\cdot})$ over any field.

Nevertheless, the concept of rank_n plays an important role. When trying to find the rank of a specific 3-way array, it is almost always easier to work with one of the rank_n rather than directly with rank. Furthermore, it is frequently important to choose wisely among the rank_n, since one of these concepts may be easier to apply to this array than another.

It is also easy to extend the concepts of row rank and column rank in the wrong way. In Kruskal (1978) I introduced the concepts $\dim_n$ and described them as extensions of row and column rank. Though the $\dim_n$, which are defined below, are useful and important concepts, I now feel that they are not the correct extensions of row rank and column rank. One indication of this is that $\dim_n(z_{\cdot\ldots\cdot})$ is frequently smaller than $\text{rank}(z_{\cdot\ldots\cdot})$.

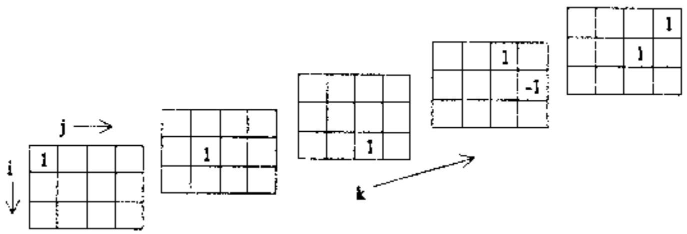
To extend row rank and column rank, some terminology is necessary. Define an *n-slice* of an N-way array to be an $(N-1)$-way subarray formed by setting the n-th index to have some constant value. Thus the 1-slices and 2-slices of a matrix are its rows and columns. The 1-slices of a three-way array $y_{\cdot\cdot\cdot}$ are matrices $y_{i\cdot\cdot}$. Next we introduce the special term *monad*, parallel to dyad and triad, for a 1-ad. When the definition of N-ad is applied to the case $N = 1$, it turns out that a monad is *any* vector (i.e., any 1-way array), so monad is synonymous with vector. Even so, it is a useful word because it indicates that we are going to generalize to dyads, triads, etc., instead of to matrices, 3-way arrays, etc. Finally, we define the n-$(N-1)$-ads of an N-linear decomposition: this includes the 1-monads and 2-monads of a bilinear decomposition, the 1-, 2-, and 3-dyads of a trilinear decomposition, and so forth. If the N-ads of an N-linear decomposition of $z_{\cdot\ldots\cdot}$ are $a_{\cdot r}^{(1)} \otimes \ldots \otimes a_{\cdot r}^{(N)}$, then we define the *n-(N−1)-ads* of this decomposition to be the $(N-1)$-ads

$$a_{\cdot r}^{(1)} \otimes \ldots \otimes a_{\cdot r}^{(n-1)} \otimes a_{\cdot r}^{(n+1)} \otimes \ldots \otimes a_{\cdot r}^{(N)} .$$

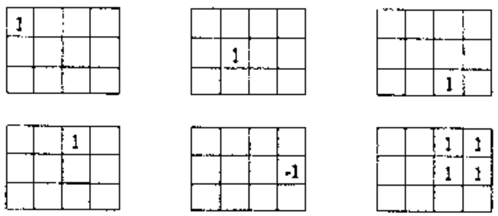
Thus the 1-monads and 2-monads of $x_{\cdot\cdot} = a_{\cdot\cdot} b_{\cdot\cdot}'$ are respectively the vectors $b_{\cdot r}$ and $a_{\cdot r}$, and the 1-dyads, 2-dyads, and 3-dyads of $y_{\cdot\cdot\cdot} = [a_{\cdot\cdot}, b_{\cdot\cdot}, c_{\cdot\cdot}]$ are respectively the dyads $b_{\cdot r} \otimes c_{\cdot r}$, $a_{\cdot r} \otimes c_{\cdot r}$, and $a_{\cdot r} \otimes b_{\cdot r}$. In general, since $z_{\cdot\ldots\cdot}$ is the sum of the N-ads of its decomposition, each n-slice of $z_{\cdot\ldots\cdot}$ is a linear combination of its n-$(N-1)$-ads.

To set the stage for the general definition of rank_n of any array, we first consider arrays that are matrices. Define $\text{rank}_n(x_{..})$ *to be the minimum number of monads that span the n-slices of* $x_{..}$. Thus for $n = 1$, rank_1 means the minimum number of vectors that span the rows of $x_{..}$, which agrees with the conventional definition of row rank, and similarly for $n = 2$. Given any decomposition of $x_{..}$, every n-slice of $x_{..}$ is a linear combination of its n-monads, so $\text{rank}_n(x_{..}) \leq \text{rank}(x_{..})$. Conversely, it is easy to see that any set of monads that spans the n-slices of $x_{..}$ yields a decomposition, which gives the reverse inequality, so $\text{rank}_n = \text{rank}$ for matrices.

Now for 3-way arrays, define $\text{rank}_n(y_{...})$ to be the minimum number of dyads that span the n-slices of $y_{...}$, and for N-way arrays define $\text{rank}_n(z_{....})$ to be the minimum number of $(N-1)$-ads that span the n-slices of $z_{....}$. The same argument used above shows that $\text{rank}_n = \text{rank}$ for arrays of all orders.



(a) Array which has $\dim_1 = 3$, $\dim_2 = 4$, $\dim_3 = 5$, $\text{rank} = 6$



(b) Six dyads which span the five 3-slices of the array

FIGURE 3

The older concept $\dim_n$ can now be easily defined: For any N-way array $z_{....}$, $\dim_n(z_{....})$ *is the minimum number of* $(N-1)$*-way arrays which span the n-slices of* $z_{....}$. Comparing this definition with the definition of rank_n, the only difference is the change from "$(N-1)$-ads" to "$(N-1)$-way arrays", so it is clear that

$$\dim_n(z_{....}) \leq \text{rank}_n(z_{....}) = \text{rank}(z_{....}).$$

(It is also easy to see how I was misled into presenting the $\dim_n$ as the extensions of row and column rank.) By contrast with the rank_n which are all equal and equal to rank, the $\dim_n$ can all be different and different from rank. As an illustration, the $3 \times 4 \times 5$ array in Figure 3a has $\dim_1 = 3$, $\dim_2 = 4$, $\dim_3 = 5$ and $\text{rank} = 6$. (Due to certain inequalities, no smaller 3-way array exists with all four values different.) The $\dim_n$ values are easy to verify, since for each n the n-slices are obviously independent. To show that $\text{rank} \leq 6$ is not hard: The six 3×4 dyads shown in Figure 3b span the five 3-slices in Figure 3a. To show $\text{rank} \geq 6$ is a good deal more difficult. One method is a fairly straightforward application of Corollary 2 from Kruskal (1978, page 109), using $M = 2$, but the proof is not presented here.

PRACTICAL CALCULATION OF TRIPLE PRODUCTS

In following sections, the reader may sometimes wish to algebraically calculate a triple product $y_{...} = \{a_{..}, b_{..}, c_{..}\}$. The naive method, direct evaluation of the formulas $\Sigma_r\, a_{ir}\, b_{jr}\, c_{kr}$, is not at all a desirable method, either in theory or in practice. Far better is to select two factors, taken as $a_{..}$ and $b_{..}$ below and use this formula:

$$y_{..k} = \Sigma_r\, c_{kr}\,(a_{.r} \otimes b_{.r}).$$

In other words, first form the R 3-dyads of the decomposition. Then for each $k = 1$ to K, form a linear combination of the the dyads using the coefficients from the k-th row of $c_{..}$. For each k, this yields the k-th 3-slice of the triple product. The amount of calculation required by the two methods is as follows:

naive method: $IJK(2R)$ multiplications + $IJK(R-1)$ additions;

recommended method: $(IJ+K)R$ multiplications + $K(R-1)$ additions.

For efficient calculation, the latter formula suggests that the two factors selected should correspond to the two smallest values from among I, J, and K.

APPLICATION TO COMPUTATIONAL COMPLEXITY

The most frequent current use of rank for 3-way arrays occurs in computational complexity theory for the problem of calculating sets of bilinear forms. A simple example is provided by the product of complex numbers, considered as an operation on pairs of real numbers. The product of (u_1, u_2) by (v_1, v_2) is

$$(w_1, w_2) = (u_1 v_1 - u_2 v_2, u_1 v_2 + u_2 v_1).$$

Each w_k is a bilinear form in the u_i and v_j, and can be conveniently represented by a matrix, where it is easy from the marginal elements to see which variables are involved:

$$w_1 \text{ by } \begin{array}{c|cc} & v_1 & v_2 \\ \hline u_1 & 1 & \\ u_2 & & -1 \end{array}, \qquad w_2 \text{ by } \begin{array}{c|cc} & v_1 & v_2 \\ \hline u_1 & & 1 \\ u_2 & 1 & \end{array}$$

Blank entries indicate zero values. The set of bilinear forms can be represented by the 3-way array formed from the the matrices:

$$y_{\dots} = \underset{w_1}{\begin{array}{c|cc} & v_1 & v_2 \\ \hline u_1 & 1 & \\ u_2 & & -1 \end{array}} \quad \underset{w_2}{\begin{array}{|cc|} & 1 \\ 1 & \end{array}}$$

The marginal variables are not part of the array $y_{\dots}$, and are just shown for convenience. More

generally, suppose $w_1, \dots, w_K$ are bilinear forms, using coefficients from some field $\mathbf{F}$, in $u_1, \dots, u_I$ and $v_1, \dots, v_J$. Then each w_k can be described by an $I \times J$ matrix over $\mathbf{F}$, and the set of bilinear forms can be described by a three-way $I \times J \times K$ array.

Now something surprising happens. The formulas for w_1 and w_2 in the product of two complex numbers contain four multiplications (plus some additions and subtractions), so it would appear that four multiplications are necessary to obtain the desired results. It turns out, however, that three multiplications are enough (together with several additions and subtractions), as shown by the following computation:

$$\begin{aligned} L_1 &= u_1 - u_2, & L_1' &= v_1 + v_2, \\ L_2 &= u_2, & L_2' &= v_1, \\ L_3 &= u_1, & L_3' &= v_2, \end{aligned}$$

$$\left.\begin{aligned} P_1 &= L_1 L_1' \\ P_2 &= L_2 L_2' \\ P_3 &= L_3 L_3' \end{aligned}\right\} \text{ the three multiplications,}$$

$w_1 = P_1 + P_2 - P_3$

$w_2 = P_2 + P_3$.

More generally, consider *canonical* calculations, similar to the calculation above, that consist of finding R linear combinations of the u_i and R linear combinations of the v_j (with coefficients in $\mathbf{F}$), multiplying corresponding combinations, and then forming linear combinations of the products. It is frequently possible to use less multiplications in a canonical calculation than the number which appear in the original bilinear forms (i.e., the number of nonzero entries in the three-way array). For the purpose of this subject, the only operations which are counted are (i) multiplications where both factors contain variables and (ii) divisions where the denominator contains a variable. Addition, subtraction, multiplication by an element in $\mathbf{F}$, and division by an element in $\mathbf{F}$ are considered free. It has been proved in general, using quite a broad model of calculation, that any method for calculating a set of bilinear forms uses at least as many multiplications as some canonical calculation. This implies not only that divisions are unnecessary for calculating sets of bilinear forms, but that using them does not decrease the number of multiplications needed.

Since three multiplications are enough in the example, we inevitably wonder whether it is possible to get away with less. What is the minimum number? This is where rank enters. As noted above, it is only necessary to consider canonical calculations, and any canonical calculation corresponds directly to a decomposition (i.e., triple product), e.g., the decomposition above yields that

$$y_{\dots} = \left[\left[\begin{matrix} 1 & 0 & 1 \\ -1 & 1 & 0 \end{matrix}\right], \left[\begin{matrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}\right], \left[\begin{matrix} 1 & 1 & -1 \\ 0 & 1 & 1 \end{matrix}\right]\right],$$

where the first factor consists of the coefficients of the u_i, the second factor of the coefficients of the v_j, and the third factor the coefficients in the expressions for the w_k. (If you wish to verify that the triple product above actually is equal to the array $y_{\dots}$ given at the beginning of this section, see the section on calculating triple products.) Therefore the minimum number of multiplications needed to calculate a set of bilinear forms is the rank of the array.

It turns out that the minimum number of multiplications needed may depend on what field the coefficients are embedded in, i.e., if the coefficients of the bilinear forms belong to two different fields $\mathbf{F}_1$ and $\mathbf{F}_2$, then the minimum number of multiplications over $\mathbf{F}_1$ may differ from the minimum number of multiplications over $\mathbf{F}_2$. This is not surprising, since the free operations that are available depend on the field. However, because the minimum number of multiplications = the rank of the array, this dependence implies that the rank of an array may depend on what field its elements are embedded in. In strong contrast, the rank of a *matrix* does *not* depend on what field its elements are embedded in.

It turns out that the array $y_{\dots}$ has rank 3 over the reals but rank 2 over the complex numbers. We have already shown that over the reals $\text{rank}(y_{\dots}) \leqq 3$. The fact that $\text{rank}(y_{\dots}) \geqq 3$ can be shown fairly quickly, but we defer the demonstration until a later section. Now consider $\text{rank}(y_{\dots})$ over the complex numbers. In view of the source of this problem, which is the multiplication of two complex numbers, it might at first seen as if only *one* complex multiplication

should be sufficient, so it might seem as if the rank should be 1. The reason why this fails is that we want to find both w_1 and w_2 separately, and one complex multiplication gives $w_1 + i\,w_2$ but not the two separate components. In this context, complex numbers are considered as indivisible entities, and such functions as real part, imaginary part, and complex conjugate are not available as primitive operations. (Whether a definition of computational complexity which excludes these primitives is appropriate to use in connection with computing in the real world is another question, but one that leads away from the topic of this paper.)

Working over the complex numbers, to see that $\text{rank}(y_{\dots}) \geqq 2$ it is sufficient to note that any array of rank $\leqq 1$ is a triad, i.e., a multiplication table, which $y_{\dots}$ obviously is not. To see that $\text{rank}(y_{\dots}) \leqq 2$, we present the following decomposition of rank 2,

$$y_{\dots} = \left[\left[\begin{matrix} 1 & 1 \\ i & -i \end{matrix}\right], \left[\begin{matrix} 1 & 1 \\ i & -i \end{matrix}\right], \left[\begin{matrix} k_1 & k_2 \\ k_2 & -k_2 \end{matrix}\right]\right],$$

where $k_1 = 1/2$ and $k_2 = 1/(2i)$, which corresponds to the following canonical calculation:

$$L_1 = u_1 + iu_2, \quad L_1' = v_1 - iv_2,$$
$$L_2 = u_1 - iu_2, \quad L_2' = v_1 - iv_2,$$

$$\left.\begin{matrix} P_1 = L_1 L_1' \\ P_2 = L_2 L_2' \end{matrix}\right\} \text{ the two multiplications,}$$

$$w_1 = k_1(P_1 + P_2),$$
$$w_2 = k_2(P_1 - P_2).$$

Of course, this calculation would not be possible if the input did not contain all four values u_1, u_2, v_1, v_2, as opposed to merely the two complex numbers $u_1 + iu_2$, $v_1 + iv_2$.

The study of this kind of computational complexity was strongly stimulated by the discovery in Strassen (1968) that multiplication of two 2×2 matrices, which appears to require 8 scalar multiplications, can in fact be accomplished using 7. Since 1978, the study of algebraic complexity of matrix multiplication has been undergoing very rapid development, particularly the asymptotic complexity for large matrices, which is reviewed in Pan (1984). The algebraic complexity of other bilinear operations has also received some attention.

To illustrate how difficult the question of rank is for 3-way arrays, we consider the array which describes multiplication of 2×2 matrices. Suppose we number the elements of a matrix as shown here,

$$\begin{bmatrix} u_1 & u_2 \\ u_3 & u_4 \end{bmatrix},$$

and let a matrix of w's be the product of a matrix of u's by a matrix of v's. Then the product consists of 4 bilinear forms, and is described by the $4 \times 4 \times 4$ array in Figure 4a. It has eight 1's and all other entries 0. Strassen's breakthrough is based on the rank 7 decomposition shown in Figure 4b. This shows that over the reals rank $\leqq 7$. The fact that the rank $\geqq 7$ is the subject of at least two independent papers, ...

The product of two 3×3 matrices is described by a $9 \times 9 \times 9$ array that has 27 1's and all other entries 0. Its exact rank is still unknown, but has been proved to lie in the interval from 17 to 23.