

# Lab 1

Brendon Swanepoel - 601949, Anita de Mello Koch - 1371116,  
Nicholas Kastanos - 1440624

## 1 3-D Array Multiplication

To obtain each element in matrix  $C$ , matrix  $A$  and  $B$  are divided into two-dimensional matrices,  $A'$  and  $B'$  as can be seen in figure 1. For each two-dimensional matrix, a single row and column is multiplied to get the corresponding element in the  $C$  matrix. This leads to a single value which is the element in the  $C$  matrix. This is done for all elements in the  $C$  matrix.

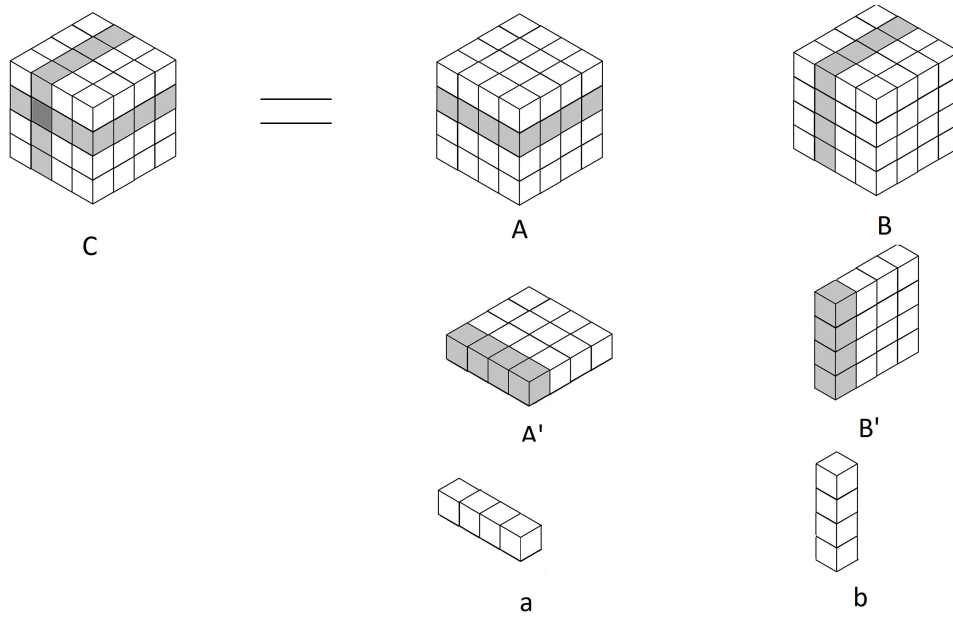


Figure 1: How each element in  $C$  was obtained from matrices  $A$  and  $B$ .

Two `for` loops were used to maintain the row and column of the current element in matrix  $C$ . Another `for` loop was used to traverse the depth of the matrix  $C$ . The row  $a$  and column  $b$  were then obtained from matrices  $A$  and  $B$  as seen in figure 1. Vector multiplication was then used on vectors  $a$  and  $b$ . The resulting value is the corresponding element of  $C$ .

This was repeated for all elements in matrix  $C$ . It was assumed matrix  $A$  and  $B$  are cubes.

## 2 Generalizing for $N$ dimensions

To multiply a multi-dimensional array of  $N$  dimensions, a recursive function can be used.

This can be done by inputting the two  $N$  dimensional arrays and the value  $N$ . For the two, multi-dimensional arrays, extract a single plane and call the

function on these  $N - 1$  multi-dimensional arrays. This process is continued until a single vector remains from each matrix. The vector product of these two vectors is the result of the corresponding element in the resultant  $N$  multi-dimensional array.

### 3 Pseudocode

**input** : Two 2D square matrices

**output:** Calculating the addition of two 2D square matrices

initialise results matrix;

**for** *Each row of the matrix* **do**

**for** *Each column element in the current row* **do**

        Sum the corresponding row and column elements of  
        both matrices;

        Store sum value in result matrix;

**end**

**end**

Return results matrix;

**Algorithm 1:** 2D Addition Algorithm

**input** : Two 3D cubic matrices

**output:** Calculating the addition of two 3D cubic matrices

initialise results matrix;

**for** *Each depth row of the cube* **do**

**for** *Each row of the cube at the current depth* **do**

**for** *Each column element at the current depth and row* **do**

            Sum the corresponding row and column elements of  
            both matrices at the current depth;

            Store sum value in result matrix;

**end**

**end**

**end**

Return results matrix;

**Algorithm 2:** 3D Addition Algorithm

**input** : Two 2D square matrices B  
**output**: Calculating the multiplication of two 2D square matrices  
 initialise results matrix;  
**for** *Each row of the square* **do**  
 | **for** *Each column of the square at the current row* **do**  
 | | **for** *Each element in the row and column of the corresponding*  
 | | *matrices* **do**  
 | | | Multiply and sum the corresponding row and column elements;  
 | | **end**  
 | | Store sum value in result matrix;  
 | **end**  
**end**  
 Return results matrix;

**Algorithm 3:** 2D Multiplication Algorithm

**input** : Matrix A and B, two 3D cubic matrices  
**output**: Matrix C, the multiple of two 3D matrices  
**for** *each row in matrix C* **do**  
 | **for** *each column in matrix C* **do**  
 | | **for** *each depth in matrix C* **do**  
 | | | Get corresponding row at depth from matrix A;  
 | | | Get corresponding column at depth from matrix B;  
 | | | Multiply the obtained row and column to get the value of matrix  
 | | | C at the current row, column and depth.  
 | | **end**  
 | **end**  
**end**  
**end**

**Algorithm 4:** 3D Multiplication Algorithm