APPENDIX

A 2-Dimensional matrix addition

The addition of two 2-dimensional matrices is achieved using element-by-element addition, as shown in Equation 1 [1]. Algorithm 1 displays the pseudo-code used to calculate the addition of two $N \times N$ matrices.

$$c_{ij} = a_{ij} + b_{ij} \tag{1}$$

Algorithm 1: rank2TensorAdd finds the addition of two $N \times N$ matrices

Input: Three $N \times N$ constant integer arrays: $a[N][N], b[N][N]$ and $c[N][N]$
Output: void

$N \leftarrow \text{size}(a)$
for $i \leftarrow 0$ ***to*** $N - 1$ **do**
    for $j \leftarrow 0$ ***to*** $N - 1$ **do**
        $c[i][j] = a[i][j] + b[i][j]$

B 2-Dimensional matrix multiplication

The multiplication of two 2-dimensional matrices is achieved by performing the dot product on the respective rows and columns, as illustrated in Equation 2 [2]. Algorithm 2 shows the pseudo-code used to multiply two $N \times N$ matrices.

$$c_{ij} = \sum_k a_{ij} \times b_{ij} \tag{2}$$

Algorithm 2: rank2TensorMulti finds the multiplication of two $N \times N$ matrices

Input: Three $N \times N$ constant integer arrays: $a[N][N], b[N][N]$ and $c[N][N]$
Output: void

$N \leftarrow \text{size}(a)$
for $k \leftarrow 0$ ***to*** $N - 1$ **do**
    for $i \leftarrow 0$ ***to*** $N - 1$ **do**
        for $j \leftarrow 0$ ***to*** $N - 1$ **do**
            $c[i][j] = c[i][j] + a[i][j] \times b[i][j]$

C 3-Dimensional array addition

The addition of two 3-dimensional arrays is achieved using element-by-element addition. Algorithm 3 shows the pseudo-code used to sum two $N \times N \times N$ matrices.

Algorithm 3: rank3TensorAdd finds the addition of two $N \times N \times N$ matrices

Input: Three $N \times N \times N$ constant integer arrays: $a[N][N][N], b[N][N][N]$ and $c[N][N][N]$
Output: void

$N \leftarrow \text{size}(a)$
for $k \leftarrow 0$ ***to*** $N - 1$ **do**
    for $i \leftarrow 0$ ***to*** $N - 1$ **do**
        for $j \leftarrow 0$ ***to*** $N - 1$ **do**
            $c[i][j][k] = a[i][j][k] + b[i][j][k]$

D 3-Dimensional array multiplication

The multiplication of two 3-dimensional arrays makes use of 2-dimensional matrix multiplication. The i^{th} row-plane of array A and the j^{th} column-plane of array B are multiplied using traditional 2-dimensional matrix multiplication shown in Algorithm 2. The result is the k^{th} layer-plane of array C. Algorithm 4 shows the pseudo-code used to multiply two $N \times N \times N$ matrices.

Algorithm 4: rank3TensorMulti finds the multiplication of two $N \times N \times N$ matrices

Input: Three $N \times N \times N$ constant integer arrays: $a[N][N][N], b[N][N][N]$ and $c[N][N][N]$

Output: void

$N \leftarrow \text{size}(a)$
$temp\_c = zeros(N, N)$
for $k \leftarrow 0$ ***to*** $N - 1$ **do**
  $temp\_a = a[k][:][:]$
  $temp\_b = b[:][k][:]$
  $rank2TensorMulti(temp\_a[N][N], temp\_b[N][N], temp\_c[N][N])$
  $c[:][:][k] = temp\_c$

References

[1] E. Stapel. "Adding and Subtracting Matrices." Online, 2012. URL https://www.purplemath.com/modules/mtrxadd.htm.

[2] MathsIsFun.com. "How to Multiply Matrices." Online, 2017. URL https://www.mathsisfun.com/algebra/matrix-multiplying.html.