Southern Illinois University Carbondale
OpenSIUC

Honors Theses | University Honors Program

5-1997

Kronecker Product Factorization of de Boor's Mixed-Radix FFT

Leonard Hoffnung

KRONECKER PRODUCT FACTORIZATION

OF DE BOOR'S MIXED-RADIX FFT

Leonard Hoffnung

May 1997

1. Introduction.

Carl de Boor's algorithm [1] provides a clever way to compute the mixed-radix FFT in a series of steps. Using this algorithm, the intermediate results produced by the algorithm have a meaningful interpretation. In this paper, we will describe this algorithm as published by de Boor. We will then introduce a compact new form of matrix notation, the Kronecker product, and a special-purpose matrix called the shuffle permutation. Finally, using some elementary properties of the Kronecker product and shuffle permutations, we can demonstrate how the de Boor algorithm can be rewritten as a product of matrices in a variety of powerful and simple ways.

At the heart of the algorithm, de Boor rewrites an N-vector (one-dimensional array) interchangeably as a two- or three-dimensional array whenever it is more convenient; we will use de Boor's left-subscript notation (1) to communicate how this is done:

$$(1) \qquad {}_{A\times B}\mathbf{z} := \begin{vmatrix} \mathbf{z}(1) & \mathbf{z}(A+1) & \cdots & \mathbf{z}(A[B-1]+1) \\ \mathbf{z}(2) & \mathbf{z}(A+2) & & \mathbf{z}(A[B-1]+2) \\ \vdots & & \ddots & \vdots \\ \mathbf{z}(A) & \mathbf{z}(2A) & \cdots & \mathbf{z}(AB) \end{vmatrix}, \text{ or } {}_{A\times B}\mathbf{z}(m,n) := \mathbf{z}(m + A[n-1]).$$

The three-dimensional form will be

$${}_{A\times B\times C}\mathbf{z}(a,b,c) := \mathbf{z}(a + A[b-1] + AB[c-1]).$$

When we actually rewrite the algorithm in terms of matrix multiplication, such matrix expressions of $\mathbf{z}$ won't be very useful. For the most part, we will instead view this description as a system to index the location of each component of the vector.

Since we will use complex exponential terms frequently, for compactness we write:

(2) $\omega_A := e^{-2\pi i/A}$, $A = 1,2,\cdots$, so that $\omega_A^A = 1$.

Furthermore, we will number the components of any arbitrary N-vector from 1 to N, in accordance with de Boor; hence, we use de Boor's equation for the DFT accordingly:

(3) $$\hat{\mathbf{z}}(n) := \sum_{\nu=1}^{N} \mathbf{z}(\nu)\omega_N^{(\nu-1)(n-1)} \quad , \quad 1 \le n, \nu \le N .$$

This equation differs from other analysis equations, such as the one used by Kammler [3], not only in the numbering but also in the lack of a $1/N$ constant on the right side. These differences are only a matter of preference, and will not pose a problem as long as we remember to adjust the results, or use the corresponding synthesis equation.

2. De Boor's algorithm.

The algorithm works for the mixed-radix or general case $\mathrm{N} = \mathrm{P}_1\mathrm{P}_2\mathrm{P}_3\cdots\mathrm{P}_\mathrm{M}$, where $P_1, P_2, \cdots, P_M$ are positive integers. We will construct a series of data arrays, numbered $\mathbf{z}_0$ through $\mathbf{z}_M$. For each integer k such that $0 \le k \le M$, we define

$$A := P_{k+1}P_{k+2}\cdots P_M; \quad A_M := 1$$

(4) $$B := P_1P_2\cdots P_{k-1} \text{ for } \mathrm{k} \ge 2; \quad B := 1 \text{ for } \mathrm{k} = 0, 1$$

$$P := P_k, \quad P_0 := 1.$$

It should be noted that A, B, and P depend on k, and properly should be written as A_k, B_k, and P_k, respectively. For the majority of this paper, we will be treating k as a constant, and will omit the subscripts for notational convenience. The subscripts will be utilized only when it becomes necessary to clarify to which values of A and B we are referring.

Now we write our original N-vector $\mathbf{z}$ as a two-dimensional $BP \times A$ array, and let $\mathbf{z}_k$ be an $A \times BP$ array such that each column of $\mathbf{z}_k$ is the DFT of the corresponding row of $\mathbf{z}$, like so:

$$_{BP\times A}\mathbf{z} = \begin{vmatrix} \mathbf{z}(1) & \mathbf{z}(BP+1) & \mathbf{z}(2BP+1) & \cdots & \mathbf{z}(BP[A-1]+1) \\ \mathbf{z}(2) & \mathbf{z}(BP+2) & \mathbf{z}(2BP+2) & & \mathbf{z}(BP[A-1]+2) \\ \mathbf{z}(3) & \mathbf{z}(BP+3) & \mathbf{z}(2BP+3) & & \mathbf{z}(BP[A-1]+3) \\ \vdots & & & \ddots & \vdots \\ \mathbf{z}(BP) & \mathbf{z}(2BP) & \mathbf{z}(3BP) & \cdots & \mathbf{z}(N) \end{vmatrix} = \begin{vmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \vdots \\ \mathbf{r}_{BP} \end{vmatrix}_{BP\times A},$$

$$_{A\times BP}\mathbf{z}_k = \begin{vmatrix} \hat{\mathbf{r}}_1(1) & \hat{\mathbf{r}}_2(1) & \hat{\mathbf{r}}_3(1) & \cdots & \hat{\mathbf{r}}_{BP}(1) \\ \hat{\mathbf{r}}_1(2) & \hat{\mathbf{r}}_2(2) & \hat{\mathbf{r}}_3(2) & & \hat{\mathbf{r}}_{BP}(2) \\ \hat{\mathbf{r}}_1(3) & \hat{\mathbf{r}}_2(3) & \hat{\mathbf{r}}_3(3) & & \hat{\mathbf{r}}_{BP}(3) \\ \vdots & & & \ddots & \vdots \\ \hat{\mathbf{r}}_1(A) & \hat{\mathbf{r}}_2(A) & \hat{\mathbf{r}}_3(A) & \cdots & \hat{\mathbf{r}}_{BP}(A) \end{vmatrix}_{A\times BP}$$

or compactly

$$_{A\times BP}\mathbf{z}_k(m,n) = \hat{\mathbf{r}}_n(m) = \sum_{i=1}^{A} {}_{BP\times A}\mathbf{z}(n,i)\omega_A^{(i-1)(m-1)},$$

(5) $$_{A\times BP}\mathbf{z}_k(m,n) = \sum_{i=1}^{A} \mathbf{z}[n + BP(i-1)]\omega_A^{(i-1)(m-1)},\ 0 \le k \le M.$$

Since it is readily apparent that $A_{k-1} = A_k P_k$ and $B_{k-1} P_{k-1} = B_k$ we can deduce immediately that

(6) $$_{AP\times B}\mathbf{z}_{k-1}(m,n) = \sum_{v=1}^{AP} {}_{B\times AP}\mathbf{z}(n,v)\omega_{AP}^{(v-1)(m-1)} = \sum_{v=1}^{AP} \mathbf{z}[n + B(v-1)]\omega_{AP}^{(v-1)(m-1)},\ 1 \le k \le M.$$

We note that in equation (5) we set each component of $\mathbf{z}_k$ to a complex polynomial with coefficients taken from $\mathbf{z}$, selected by "steps" of size BP; in (6), we found $\mathbf{z}_{k-1}$ by a similar process, the difference being that steps of size B were used instead. One might guess that by summing the correct P components of $\mathbf{z}_k$ we would collect all of the necessary samples of the original vector $\mathbf{z}$, and possibly derive a component of $\mathbf{z}_{k-1}$. In fact, by adjusting the ω multipliers appropriately, a modification of this summation will determine $\mathbf{z}_{k-1}$.

We begin by using a double summation for the right-hand side:

$$_{AP\times B}\mathbf{z}_{k-1}(m,b) = \sum_{v=1}^{AP} \mathbf{z}[b + B(v-1)]\omega_{AP}^{(v-1)(m-1)}$$

$$= \sum_{\eta=1}^{P}\sum_{i=1}^{A} \mathbf{z}[b + B(\eta + P[i-1] - 1)]\omega_{AP}^{[\eta+P(i-1)-1](m-1)}, \quad \mathrm{v} = \eta + \mathrm{P}(\mathrm{i}-1).$$

In order to simplify the exponential term, we convert $\mathbf{z}_{k-1}$ to a three-dimensional array with the substitution $\mathrm{m} = \mathrm{a} + \mathrm{A}(\mathrm{p}-1), 1 \le \mathrm{a} \le \mathrm{A}, 1 \le \mathrm{p} \le \mathrm{P}$:

$$_{A\times P\times B}\mathbf{z}_{k-1}(a,p,b) = \sum_{\eta=1}^{P}\sum_{i=1}^{A} \mathbf{z}[b + B(\eta-1) + BP(i-1)]\omega_{AP}^{[\eta-1+P(i-1)](a-1+A[p-1])}$$

$$= \sum_{\eta=1}^{P}\sum_{i=1}^{A} \mathbf{z}[b + B(\eta-1) + BP(i-1)]\omega_{AP}^{(\eta-1)[a-1+A(p-1)]}\omega_{AP}^{P(i-1)(a-1)}\omega_{AP}^{AP(i-1)(p-1)}.$$

The last ω_{AP} factor reduces to 1, and the next to last reduces to a power of ω_A:

$$_{A\times P\times B}\mathbf{z}_{k-1}(a,p,b) = \sum_{\eta=1}^{P}\sum_{i=1}^{A} \mathbf{z}[b + B(\eta-1) + BP(i-1)]\omega_{A}^{(i-1)(a-1)}\omega_{AP}^{(\eta-1)[a-1+A(p-1)]}$$

$$= \sum_{\eta=1}^{P}\left\{\sum_{i=1}^{A} \mathbf{z}[b + B(\eta-1) + BP(i-1)]\omega_{A}^{(i-1)(a-1)}\right\}\omega_{AP}^{(\eta-1)[a-1+A(p-1)]}.$$

Now we see that the summation inside the braces can be written in terms of $\mathbf{z}_k$:

$$\sum_{i=1}^{A} \mathbf{z}[b + B(\eta-1) + BP(i-1)]\omega_{A}^{(i-1)(a-1)} = {}_{A\times BP}\mathbf{z}_k[a, b + B(\eta-1)]$$

$$= {}_{A\times B\times P}\mathbf{z}_k(a,b,\eta),$$

leaving

(7) $${}_{A\times P\times B}\mathbf{z}_{k-1}(a,p,b) = \sum_{\eta=1}^{P} {}_{A\times B\times P}\mathbf{z}_k(a,b,\eta)\omega_{AP}^{(\eta-1)[a-1+A(p-1)]}$$

as de Boor's expression of $\mathbf{z}_{k-1}$ in terms of $\mathbf{z}_k$.

Now that we have a procedure for finding each vector from the one next higher in number, we examine the meaning of the first and last in the series. Following the discussion presented earlier, $\mathbf{z}_M$ is constructed by forming $\mathbf{z}$ into an $N\times 1$ matrix and finding the DFT of each row; since the DFT of a 1-vector is certainly itself, the result is a $1\times N$ matrix storing the original vector $\mathbf{z}$. On the other hand, to find $\mathbf{z}_0$ we write $\mathbf{z}$ as an $N\times 1$ matrix, or a single row. Since only one row is present, $\mathbf{z}_0$ contains the DFT of the entire vector $\mathbf{z}$. We can show this formally using equations (5) and (6); since $A_M = 1$,

(8) $$\mathbf{z}_M[n] = {}_{A\times BP}\mathbf{z}_M[1,n] = \sum_{i=1}^{1} \mathbf{z}[n+N(i-1)]\omega_A^{(i-1)(m-1)} = \mathbf{z}[n],$$

and using (6) and the fact that $B_1 = 1$,

(9) $$\mathbf{z}_0[n] = {}_{AP\times B}\mathbf{z}_o[n,1] = \sum_{v=1}^{N} \mathbf{z}[1+(v-1)]\omega_N^{(v-1)(n-1)} = \sum_{v=1}^{N} \mathbf{z}[v]\omega_N^{(v-1)(n-1)} = \hat{\mathbf{z}}[n].$$

3. Efficiency and implementation of de Boor's algorithm.

The summation performed in equation (7) is simply the evaluation of a complex polynomial of degree $P = P_k - 1$, requiring 1 operation (defined as a complex multiplication followed by an addition) to adjust the ω factor, and $P_k - 1$ operations to compute each component. The procedure can be implemented as follows.

Let $x(a,p) := \omega_{AP}^{a-1+A(p-1)}$; we can compute the summation in equation (7) by the following nested multiplication.

```
        set T := z_k (a, b, P)
        for nu = P-1 to 1 step -1
(10)        set T := T·x(a,p) + z_k (a, b, nu)
        next nu
        set z_k-1 s(a, p, b) := T
```

Ignoring the minimal cost of copying the temporary variable T into and out of the arrays in the first and last lines of the algorithm, the cost is equal to $P_k - 1$ operations, with $x(a,p)$ already computed. Since $x(1, 1) = 1$ and $x(a,p) = x(a-1,p)\cdot\omega_{AP}$, we can perform the whole computation of $\mathbf{z}_{k-1}$ by using the following algorithm:

```
        set x := 1
(11)    for b = 1 to B
            for p = 1 to P
                for a = 1 to A
                    ! Compute the summation with code (10) here, using x (a, p) = x
                    set x := x·ω_AP
                next a
            next p
        next b
```

A savings of B operations would be possible if the outermost loop simply reset $x = 1$, instead of multiplying it by ω_{AP} when $a = A,\ b = P$; it might also improve the precision by eliminating the floating-point error in calculating $\omega_{AP}^{AP} = 1$. As it is written here, the algorithm (11) consumes P_k operations in the innermost loop, for a total of $A_k B_k P_k P_k = NP_k$ operations. To compute the entire DFT, we only need to load the $\mathbf{z}_M$ array, construct a loop which sets A, B, and P and executes the algorithm in (11) for each k from M to 1, and then unload the $\mathbf{z}_0$ array. When this is done, the cost of computing $\mathbf{z}_0$ from $\mathbf{z}_M$ is

$\sum_{k=1}^{M} NP_k$ operations, as expected. With the modification, the cost savings when $N = Q^M$, (i.e., $P_1 = P_2 = \cdots = P_M = Q$) is about

$$\frac{\sum_{k=1}^{M} B_k}{\sum_{k=1}^{M} NP_k} = \frac{\sum_{k=1}^{M} Q^{k-1}}{\sum_{k=1}^{M} NQ} = \frac{Q^M - 1}{NQM} = \frac{Q^M - 1}{Q^{M+1} M} \approx \frac{1}{QM}.$$

An improvement of about 5% results when $N = 1024 = 2^{10}$.

4. Kronecker products and shuffle matrices: definitions & properties.

We would like to rewrite de Boor's algorithm in matrix form, using only $N \times N$ matrices. The matrices that express each step of the algorithm are patterned, but confusing and bulky to write. We will introduce a new notational tool called the Kronecker product, used to write large matrices made up of repeated submatrix "blocks" that differ from one another by a scalar multiplier. The notation looks like this:

$$\text{(12)} \qquad \mathbf{A}_{K\times L} \otimes \mathbf{B}_{M\times N} := \left|\begin{array}{c:c:c:c} \mathbf{A}[1,1]\cdot\mathbf{B} & \mathbf{A}[1,2]\cdot\mathbf{B} & \cdots & \mathbf{A}[1,L]\cdot\mathbf{B} \\ \hdashline \mathbf{A}[2,1]\cdot\mathbf{B} & \mathbf{A}[2,2]\cdot\mathbf{B} & & \mathbf{A}[2,L]\cdot\mathbf{B} \\ \hdashline \vdots & & \ddots & \vdots \\ \hdashline \mathbf{A}[K,1]\cdot\mathbf{B} & \mathbf{A}[K,2]\cdot\mathbf{B} & \cdots & \mathbf{A}[K,L]\cdot\mathbf{B} \end{array}\right|_{KM\times LN}.$$

In terms of individual components, we have

(13) $$\mathbf{A}_{K\times L} \otimes \mathbf{B}_{M\times N}[m + M(k-1), n + N(l-1)] := \mathbf{A}[k,l] \cdot \mathbf{B}[m,n].$$

In this paper, we will be working exclusively with square matrices; thus we will always set $M = N$. Also, we use an even further abbreviated notation for the special case when the left matrix is an identity:

$$\mathbf{I}_{P\times P} \otimes \mathbf{B}_{N\times N} := \left|\begin{array}{c:c:c:c:c} \mathbf{B} & & & & \\ \hdashline & \mathbf{B} & & & \\ \hdashline & & \mathbf{B} & & \\ \hdashline & & & \ddots & \\ \hdashline & & & & \mathbf{B} \end{array}\right|_{NP\times NP},$$

$$\mathbf{I}_{P\times P} \otimes \mathbf{B}_{N\times N}[m + N(k-1), n + N(l-1)] = \mathbf{I}[k,l] \cdot \mathbf{B}[m,n].$$

For compactness, we will sometimes write this as $\mathbf{B}^{(P)}_{N\times N}$.

Two simple properties of Kronecker products will be useful. The first one, associativity, is tedious, but not difficult to show. For $\mathbf{A}_{K\times L}$, $\mathbf{B}_{M\times N}$, and $\mathbf{C}_{O\times P}$, we have

$$\begin{aligned}
&\{\mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C})\}[o + O(m-1) + MO(k-1), p + P(n-1) + NP(l-1)] \\
&\quad = \mathbf{A}[k,l] \cdot (\mathbf{B} \otimes \mathbf{C})[o + O(m-1), p + P(n-1)] \\
&\quad = \mathbf{A}[k,l] \cdot (\mathbf{B}[m,n] \cdot \mathbf{C}[o,p]) = (\mathbf{A}[k,l] \cdot \mathbf{B}[m,n]) \cdot \mathbf{C}[o,p] \\
&\quad = (\mathbf{A} \otimes \mathbf{B})[m + M(k-1), n + N(l-1)] \cdot \mathbf{C}[o,p] \\
&\quad = \{(\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C}\}[o + O(m-1) + MO(k-1), p + P(n-1) + NP(l-1)]
\end{aligned}$$

so that

(14) $$\mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C}.$$

The second one is slightly longer. For $\mathbf{A}_{K\times K}$, $\mathbf{B}_{M\times M}$, $\mathbf{C}_{K\times K}$, $\mathbf{D}_{M\times M}$, we have

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D})[b + M(a-1), c + M(d-1)]$$

$$= \sum_{j=1}^{N}\sum_{i=1}^{M} (\mathbf{A}\otimes\mathbf{B})[b+M(a-1), i+M(j-1)]\cdot(\mathbf{C}\otimes\mathbf{D})[i+M(j-1), c+M(d-1)]$$

$$= \sum_{j=1}^{N}\sum_{i=1}^{M} \mathbf{A}[a,j]\cdot\mathbf{B}[b,i]\cdot\mathbf{C}[j,d]\cdot\mathbf{D}[i,c]$$

$$= \sum_{j=1}^{N} \mathbf{A}[a,j]\cdot\mathbf{C}[j,d]\sum_{i=1}^{M}\mathbf{B}[b,i]\cdot\mathbf{D}[i,c] = (\mathbf{AC})[a,d]\cdot(\mathbf{BD})[b,c]$$

$$= \{\mathbf{AC}\otimes\mathbf{BD}\}[b+M(a-1), c+M(d-1)]$$

so that

$$(15) \qquad (\mathbf{A}\otimes\mathbf{B})(\mathbf{C}\otimes\mathbf{D}) = \mathbf{AC}\otimes\mathbf{BD}.$$

(This result also holds for matrices that are not square, as long as **AC** and **BD** are defined.)

In addition to the Kronecker product, we will also need special notation to describe the operation of rearranging the components of an arbitrary vector by "shuffling" its components. We write $\mathbf{S}_{P,Q}$ to denote a permuted $PQ\times PQ$ identity matrix with the following action: first, the arbitrary PQ column vector **z** is laid down as a $Q\times P$ array by columns, and then the rows are picked up to produce the product vector, $\mathbf{S}_{P,Q}\mathbf{z}$. Thus we lay down **z** by columns

$$\left|\begin{array}{cccc} \mathbf{z}[1] & \mathbf{z}[Q+1] & \cdots & \mathbf{z}[Q(P-1)+1] \\ \mathbf{z}[2] & \mathbf{z}[Q+2] & & \mathbf{z}[Q(P-1)+2] \\ & & \ddots & \\ \vdots & & & \vdots \\ \mathbf{z}[Q] & \mathbf{z}[2Q] & \cdots & \mathbf{z}[PQ] \end{array}\right|_{Q\times P}$$

and pick up the rows to obtain

$$\mathbf{S}_{P,Q}\mathbf{z} := (\mathbf{z}[1], \mathbf{z}[Q+1], \cdots, \mathbf{z}[Q(P-1)+1], \mathbf{z}[2], \mathbf{z}[Q+2], \cdots, \mathbf{z}[Q(P-1)+2], \\ \cdots, \mathbf{z}[Q], \mathbf{z}[2Q], \cdots, \mathbf{z}[PQ])^T.$$

If we write the equation in terms of individual vector components, we have

$$\text{(16)} \qquad \{\mathbf{S}_{P,Q}\mathbf{z}\}[p+(q-1)P] := \mathbf{z}[q+(p-1)Q], \quad 1 \le p \le P,\ 1 \le q \le Q$$

The action is called a P-shuffle of $\mathbf{z}$, and the $\mathbf{S}_{P,Q}$ array can be described as a Q-shuffle of the columns of the PQ by PQ identity matrix. The effect of $\mathbf{S}_{P,Q}$ is to shuffle the adjacent components of $\mathbf{z}$ to positions P apart; conversely, it brings together components that were originally spaced Q positions apart.

When we examine the shuffle permutation's effects on vectors interpreted as 2-dimensional arrays, the following property follows immediately from equation (16):

$$_{P\times Q}\{\mathbf{S}_{P,Q}\mathbf{z}\}(p,q) = {}_{Q\times P}\mathbf{z}(q,p)$$

Reshuffling the shuffled vector, we have:

$$\begin{aligned} _{Q\times P}\{\mathbf{S}_{Q,P}\mathbf{S}_{P,Q}\mathbf{z}\}(q,p) &= {}_{P\times Q}(\mathbf{S}_{P,Q}\mathbf{z})(p,q) \\ &= {}_{Q\times P}\mathbf{z}(q,p), \end{aligned}$$

and in this way we deduce that

$$\mathbf{S}_{Q,P}\mathbf{S}_{P,Q} = \mathbf{I},$$

$$\text{(17)} \qquad \mathbf{S}^{-1}_{P,Q} = \mathbf{S}_{Q,P}.$$

If we shuffle a vector and express it 3-dimensionally, we get:

$$\begin{aligned} _{A\times P\times B}\{\mathbf{S}_{A,BP}\mathbf{z}\}(a,p,b) &= \{\mathbf{S}_{A,BP}\mathbf{z}\}[a+(p-1)A+(b-1)AP] \\ &= \{\mathbf{S}_{A,BP}\mathbf{z}\}[a+[p+(b-1)P-1]A] \\ &= \mathbf{z}[p+(b-1)P+(a-1)BP] \end{aligned}$$

(18) $\qquad = {}_{P\times B\times A}\mathbf{z}(p,b,a)$.

Since the application of $\mathbf{S}_{A,BP}$ cycles the subscripts to the left, we can apply three successive shuffle matrices to bring back the identity:

$$_{A\times P\times B}\{\mathbf{S}_{A,PB}\mathbf{S}_{P,BA}\mathbf{S}_{B,AP}\mathbf{z}\}(a,p,b) = {}_{P\times B\times A}\{\mathbf{S}_{P,BA}\mathbf{S}_{B,AP}\mathbf{z}\}(p,b,a)$$

$$= {}_{B\times A\times P}\{\mathbf{S}_{B,AP}\mathbf{z}\}(b,a,p)$$

$$= {}_{A\times P\times B}\mathbf{z}(a,p,b),$$

and, in this way, we see that

$$\mathbf{S}_{A,PB}\mathbf{S}_{P,BA}\mathbf{S}_{B,AP} = \mathbf{I}.$$

It then follows that

(19) $\qquad \mathbf{S}_{B,AP} = (\mathbf{S}_{A,PB}\mathbf{S}_{P,BA})^{-1} = \mathbf{S}_{BA,P}\mathbf{S}_{PB,A}$.

Setting $B = 1$ in equation (19) shows that a 1-shuffle is simply the identity:

(20) $\qquad \mathbf{S}_{1,AP} = \mathbf{S}_{A,P}\mathbf{S}_{P,A} = \mathbf{I}; \quad \mathbf{S}_{AP,1} = (\mathbf{S}_{1,AP})^{-1} = \mathbf{I}$

Let's consider the action of a Kronecker product of an identity and a shuffle permutation, say $\mathbf{S}^{(P)}_{A,B}$. When this matrix is applied to an N-vector $\mathbf{z}$, it partitions $\mathbf{z}$ into AB-subvectors and performs an A-shuffle on each piece. Since the A-shuffle reverses the 2-dimensional interpretation of each AB-subvector from $A\times B$ to $B\times A$, and the total matrix leaves the order of the subvectors unchanged, the result is to convert ${}_{A\times B\times P}\mathbf{z}(a,b,p)$ to ${}_{\mathrm{B\times A\times P}}\mathbf{z}(b,a,p)$. We can demonstrate this formally:

$$_{A\times B\times P}\{\mathbf{S}^{(P)}_{A,B}\mathbf{z}\}(a,b,p) = \{(\mathbf{I}_{P\times P}\otimes\mathbf{S}_{A,B})\mathbf{z}\}[a+(b-1)A+(p-1)AB]$$

$$= \sum_{n=1}^{N}(\mathbf{I}_{P\times P}\otimes\mathbf{S}_{A,B})[a+(b-1)A+(p-1)AB,n]\cdot\mathbf{z}(n)$$

$$= \sum_{i=1}^{AB} \sum_{j=1}^{P} (\mathbf{I}_{P\times P} \otimes \mathbf{S}_{A,B})[a+(b-1)A+(p-1)AB, i+AB(j-1)] \cdot \mathbf{z}[i+AB(j-1)]$$

$$= \sum_{i=1}^{AB} \sum_{j=1}^{P} \mathbf{I}_{P\times P}[p,j] \cdot \mathbf{S}_{A,B}[a+(b-1)A,i] \cdot \mathbf{z}[i+AB(j-1)]$$

$$= \sum_{i=1}^{AB} \mathbf{S}_{A,B}[a+(b-1)A,i] \cdot \mathbf{z}[i+(p-1)AB]$$

$$= \sum_{i=1}^{AB} \mathbf{S}_{A,B}[a+(b-1)A,i] \cdot {}_{AB\times P}\, \mathbf{z}(i,p) .$$

Letting $\mathbf{y}[i] = {}_{AB\times P}\, \mathbf{z}(i,p)$, $1 \le i \le P$, we have

$$_{A\times B\times P}\{\mathbf{S}_{A,B}^{(P)}\mathbf{z}\}(a,b,p) = \sum_{i=1}^{P} \mathbf{S}_{A,B}[a+(b-1)A,i]\mathbf{y}[i] = \{\mathbf{S}_{A,B}\mathbf{y}\}[a+(b-1)A]$$

$$= {}_{A\times B}\{\mathbf{S}_{A,B}\mathbf{y}\}(a,b) = {}_{B\times A}\mathbf{y}(b,a) = \mathbf{y}[b+(a-1)B] = {}_{AB\times P}\mathbf{z}[b+(a-1)B,p]$$

so that

$$(21) \qquad {}_{A\times B\times P}\{\mathbf{S}_{A,B}^{(P)}\mathbf{z}\}(a,b,p) \;=\; {}_{B\times A\times P}\mathbf{z}(b,a,p)$$

Thus, with combinations of shuffle permutations like those in equations (18) and (21), we can construct matrix sequences that permute the indices in a 3-dimensionally interpreted vector in any desired way.

5. Single-step matrix expression of de Boor's algorithm.

Now let's try to express equation (7) in terms of $N \times N$ matrices. We will find an expression for the matrix $\mathbf{T}_k$ such that.

$$\mathbf{z}_{k-1} = \mathbf{T}_k \mathbf{z}_k, \quad 1 \le k \le M .$$

In other words, $\mathbf{T}_k$ will perform a single "step" of de Boor's algorithm; because it doesn't produce a complete FFT, but only an intermediate result, we will call it an intermatrix. In this section we will write out the matrix formula using a "brute force" approach, and then find a compact expression for $\mathbf{T}_k$ using Kronecker products. For comparison we will tackle de Boor's equation (7) directly, using shuffle products to find another expression for $\mathbf{T}_k$. After we find $\mathbf{T}_k$, we will arrive at a complete Fourier transform just by multiplying all of the $\mathbf{T}_k$ together:

$$\hat{\mathbf{z}} = \mathbf{z}_0 = \mathbf{T}_1\mathbf{T}_2\mathbf{T}_3\cdots\mathbf{T}_M\mathbf{z}; \quad F_N = \mathbf{T}_1\mathbf{T}_2\mathbf{T}_3\cdots\mathbf{T}_M .$$

Let's proceed to write out the entire matrix on a line-by-line basis. Referring to equation (7), we get the following unwieldy result:

$$(22) \qquad \mathbf{T}_k = \left|\begin{array}{ccc|ccc|c|ccc}
\begin{matrix} w_{0,0} \\ w_{1,0} \\ w_{2,0} \\ \vdots \\ w_{P-1,0} \end{matrix} & & & \begin{matrix} w_{0,1} \\ w_{1,1} \\ w_{2,1} \\ \vdots \\ w_{P-1,1} \end{matrix} & & & \cdots\cdots & \begin{matrix} w_{0,P-1} \\ w_{1,P-1} \\ w_{2,P-1} \\ \vdots \\ w_{P-1,P-1} \end{matrix} & & \\
\hline
 & \begin{matrix} w_{0,0} \\ w_{1,0} \\ w_{2,0} \\ \vdots \\ w_{P-1,0} \end{matrix} & & & \begin{matrix} w_{0,1} \\ w_{1,1} \\ w_{2,1} \\ \vdots \\ w_{P-1,1} \end{matrix} & & \cdots\cdots & & \begin{matrix} w_{0,P-1} \\ w_{1,P-1} \\ w_{2,P-1} \\ \vdots \\ w_{P-1,P-1} \end{matrix} & \\
\hline
 & \vdots & & & \vdots & & \vdots & & \vdots & \\
\hline
 & & \begin{matrix} w_{0,0} \\ w_{1,0} \\ w_{2,0} \\ \vdots \\ w_{P-1,0} \end{matrix} & & & \begin{matrix} w_{0,1} \\ w_{1,1} \\ w_{2,1} \\ \vdots \\ w_{P-1,1} \end{matrix} & & & & \begin{matrix} w_{0,P-1} \\ w_{1,P-1} \\ w_{1,P-1} \\ \vdots \\ w_{1,P-1} \end{matrix}
\end{array}\right|$$

The vertical lines separate B columns of submatrices, the horizontal lines separate B rows, and $\mathbf{W}_{k,l}$ is an $A \times A$ submatrix given by

$$
\begin{aligned}
\mathbf{W}_{k,l} &:= \left|\begin{matrix} \omega_{AP}^{(kA)l} & & & & \\ & \omega_{AP}^{(kA+1)l} & & & \\ & & \omega_{AP}^{(kA+2)l} & & \\ & & & \ddots & \\ & & & & \omega_{AP}^{(kA+A-1)l} \end{matrix}\right|_{A\times A} \\
&= \omega_P^{kl} \left|\begin{matrix} 1 & & & & \\ & \omega_{AP}^{l} & & & \\ & & \omega_{AP}^{2l} & & \\ & & & \ddots & \\ & & & & \omega_{AP}^{(A-1)l} \end{matrix}\right|_{A\times A} .
\end{aligned}
\tag{23}
$$

The pattern to this expression is apparent if we observe that in each row of $\mathbf{T}_k$, the submatrices are spaced by B; this indicates that dividing the vector following $\mathbf{T}_k$ into A-vectors and P-shuffling the subvectors would simplify $\mathbf{T}_k$. Since the submatrices act on subvectors spaced B apart, we use an adaptation of the $\mathbf{S}_{P,B}$ shuffle to bring the subvectors together, so that the matrix that acts after the shuffle will simplify to the following:

$$
\mathbf{T}_k = \left|\begin{array}{c:c:c:c}
\begin{matrix} \mathbf{w}_{0,0} & \mathbf{w}_{0,1} & \cdots & \mathbf{w}_{0,P-1} \\ \mathbf{w}_{1,0} & \mathbf{w}_{0,1} & & \mathbf{w}_{1,P-1} \\ \vdots & & \ddots & \vdots \\ \mathbf{w}_{P-1,0} & \mathbf{w}_{P-1,1} & \cdots & \mathbf{w}_{P-1,P-1} \end{matrix} & & & \\ \hdashline
& \begin{matrix} \mathbf{w}_{0,0} & \mathbf{w}_{0,1} & \cdots & \mathbf{w}_{0,P-1} \\ \mathbf{w}_{1,0} & \mathbf{w}_{1,1} & & \mathbf{w}_{1,P-1} \\ \vdots & & \ddots & \vdots \\ \mathbf{w}_{P-1,0} & \mathbf{w}_{P-1,1} & \cdots & \mathbf{w}_{P-1,P-1} \end{matrix} & & \\ \hdashline
& & \ddots & \\ \hdashline
& & & \begin{matrix} \mathbf{w}_{0,0} & \mathbf{w}_{0,1} & \cdots & \mathbf{w}_{0,P-1} \\ \mathbf{w}_{1,0} & \mathbf{w}_{1,1} & & \mathbf{w}_{1,P-1} \\ \vdots & & \ddots & \vdots \\ \mathbf{w}_{P-1,0} & \mathbf{w}_{P-1,1} & \cdots & \mathbf{w}_{P-1,P-1} \end{matrix}
\end{array}\right| \cdot (\mathbf{S}_{P,B} \otimes \mathbf{I}_{A\times A}).
\tag{24}
$$

Finally, we can rewrite this using another Kronecker product to get

(25) $\mathbf{T}_k = (\mathbf{I}_{B\times B} \otimes \mathbf{Q}_{A,P})(\mathbf{S}_{P,B} \otimes \mathbf{I}_{A\times A}),$

$$\text{where } \mathbf{Q}_{A,P} := \begin{vmatrix} \mathbf{W}_{0,0} & \mathbf{W}_{0,1} & \cdots & \mathbf{W}_{0,P-1} \\ \mathbf{W}_{1,0} & \mathbf{W}_{1,1} & & \mathbf{W}_{1,P-1} \\ \vdots & & \ddots & \vdots \\ \mathbf{W}_{P-1,0} & \mathbf{W}_{P-1,1} & \cdots & \mathbf{W}_{P-1,P-1} \end{vmatrix}.$$

6. Intermatrix expression: another method.

Although the result in (25) is valid, it has the drawbacks of being messy to write and difficult to verify without a lot of matrix computation. Instead, we can use shuffle permutations and Kronecker products to derive a result directly from de Boor's equation. We proceed by noting that if we break up a vector $\mathbf{z}$ into P-size pieces and Fourier transform each piece, it could be written as:

$${}_{P\times A\times B}\{\mathbf{F}_P^{(AB)}\mathbf{z}\}(p,a,b) = \sum_{\eta=1}^{P} {}_{P\times A\times B}\mathbf{z}(\eta,a,b)\omega_P^{(\eta-1)(p-1)}$$

or equivalently

(26) $${}_{P\times A\times B}\{\mathbf{F}_P^{(AB)}\mathbf{z}\}(p,a,b) = \sum_{\eta=1}^{P} {}_{P\times A\times B}\mathbf{z}(\eta,a,b)\omega_{AP}^{(\eta-1)[A(p-1)]}.$$

In order to bring this closer to de Boor's equation, we define new $N \times N$ diagonal matrices Ω and $\Omega_{P,A}$ as follows:

(27) $\Omega_{P,A} = \mathbf{D}(d_1, d_2, d_3, \cdots, d_{AP}), \quad d_{\eta+P(a-1)} = \omega_{AP}^{(\eta-1)(a-1)}$

$$\begin{aligned} \Omega &= \mathbf{D}(d_1, d_2, d_3, \cdots, d_N), \quad d_{\eta+P(a-1)+AP(b-1)} = \omega_{AP}^{(\eta-1)(a-1)} \\ &= \mathbf{I}_{B\times B} \otimes \Omega_{P,A}. \end{aligned}$$

We can insert the Ω matrix into equation (26).

$$_{P\times A\times B}[\Omega\,\mathbf{z}](\eta,a,b) = \omega_{AP}^{(\eta-1)(a-1)}\;{}_{P\times A\times B}\mathbf{z}(\eta,a,b)$$

$$\begin{aligned}
_{P\times A\times B}\{F_P^{(AB)}\Omega\,\mathbf{z}\}(p,a,b) &= \sum_{\eta=1}^{P} {}_{P\times A\times B}[\Omega\,\mathbf{z}](\eta,a,b)\omega_{AP}^{(\eta-1)[A(p-1)]} \\
&= \sum_{\eta=1}^{P} {}_{P\times A\times B}\mathbf{z}(\eta,a,b)\omega_{AP}^{(\eta-1)(a-1)}\omega_{AP}^{(\eta-1)[A(p-1)]} \\
&= \sum_{\eta=1}^{P} {}_{P\times A\times B}\mathbf{z}(\eta,a,b)\omega_{AP}^{(\eta-1)[a-1+A(p-1)]}.
\end{aligned}$$

The right side almost matches de Boor's equation; we only need to substitute $\mathbf{S}_{P,AB}\mathbf{z}_k$ for $\mathbf{z}$ and simplify:

$$\begin{aligned}
_{P\times A\times B}\{F_P^{(AB)}\Omega\,\mathbf{S}_{P,AB}\mathbf{z}_k\}(p,a,b) &= \sum_{\eta=1}^{P} {}_{P\times A\times B}[\mathbf{S}_{P,AB}\mathbf{z}_k](\eta,a,b)\omega_{AP}^{(\eta-1)[a-1+A(p-1)]} \\
&= \sum_{\eta=1}^{P} {}_{A\times B\times P}\mathbf{z}_k(a,b,\eta)\omega_{AP}^{(\eta-1)[a-1+A(p-1)]}.
\end{aligned}$$

Now we use de Boor's equation (7) to write

$$_{P\times A\times B}\{F_P^{(AB)}\Omega\,\mathbf{S}_{P,AB}\mathbf{z}_k\}(p,a,b) = {}_{A\times P\times B}\mathbf{z}_{k-1}(a,p,b).$$

At this point, we insert a shuffle permutation of the type defined in equation (21), and all of the subscripts match up. Since the indexing is the same on both sides, we can discontinue the indices altogether and write

$$_{A\times P\times B}\mathbf{z}_{k-1}(a,p,b) = {}_{A\times P\times B}\{\mathbf{S}_{A,P}^{(B)}F_P^{(AB)}\Omega\,\mathbf{S}_{P,AB}\mathbf{z}_k\}(a,p,b),$$

or equivalently

$$\mathbf{z}_{k-1} = \mathbf{S}_{A,P}^{(B)}\mathbf{F}_P^{(AB)}\Omega\,\mathbf{S}_{P,AB}\mathbf{z}_k.$$

Since by definition $\mathbf{z}_{k-1} = \mathbf{T}_k\mathbf{z}_k$, we see that

(28) $\mathbf{T}_k = \mathbf{S}_{A,P}^{(B)} F_P^{(AB)} \Omega_{P,A}^{(B)} \mathbf{S}_{P,AB}$.

The expression (28) is moderately simple, but we would prefer to combine the two shuffle permutations. In order to do this, we observe that $\mathbf{I}_{AB} = \mathbf{I}_B \otimes \mathbf{I}_A$, and use the special properties of the shuffle permutation and Kronecker product:

$$\begin{aligned}
\mathbf{T}_k &= (\mathbf{I}_{B\times B} \otimes \mathbf{S}_{A,P})(\mathbf{I}_{AB\times AB} \otimes F_P)(\mathbf{I}_{B\times B} \otimes \Omega_{P,A})\mathbf{S}_{P,AB} \\
&= (\mathbf{I}_{B\times B} \otimes \mathbf{S}_{A,P})(\mathbf{I}_{B\times B} \otimes \mathbf{I}_{A\times A} \otimes F_P)(\mathbf{I}_{B\times B} \otimes \Omega_{P,A})\mathbf{S}_{P,AB} \\
&= \{\mathbf{I}_{B\times B} \otimes [\mathbf{S}_{A,P}(\mathbf{I}_{A\times A} \otimes F_P)\Omega_{P,A}]\}\,\mathbf{S}_{P,AB} \\
&= \{\mathbf{I}_{B\times B} \otimes [(F_P \otimes \mathbf{I}_{A\times A})\mathbf{S}_{A,P}\Omega_{P,A}]\}\,\mathbf{S}_{P,AB}.
\end{aligned}$$

To go further, we would like to commute the $\mathbf{S}$ and Ω terms:

$$\begin{aligned}
{}_{P\times A}[\Omega_{P,A}\mathbf{z}](p,a) &= \omega_{AP}^{(p-1)(a-1)}\ {}_{P\times A}\mathbf{z}(p,a), \\
{}_{A\times P}[\mathbf{S}_{A,P}\Omega_{P,A}\mathbf{z}](a,p) &= \omega_{AP}^{(p-1)(a-1)}\ {}_{P\times A}\mathbf{z}(p,a) \\
&= \omega_{AP}^{(p-1)(a-1)}\ {}_{A\times P}[\mathbf{S}_{A,P}\mathbf{z}](a,p).
\end{aligned}$$

Comparing the right side with the first equation, we substitute $\mathbf{S}_{A,P}\mathbf{z}$ for $\mathbf{z}$, switch the indices, and combine to get

$${}_{A\times P}[\mathbf{S}_{A,P}\Omega_{P,A}\mathbf{z}](a,p) = {}_{A\times P}[\Omega_{A,P}\mathbf{S}_{A,P}\mathbf{z}](a,p),$$

i.e.

$$\mathbf{S}_{A,P}\Omega_{P,A} = \Omega_{A,P}\mathbf{S}_{A,P}.$$

Returning to the factorization of $\mathbf{T}_k$, we write

$$\begin{aligned}
\mathbf{T}_k &= \{\mathbf{I}_{B\times B} \otimes [(F_P \otimes \mathbf{I}_{A\times A})\Omega_{A,P}\mathbf{S}_{A,P}]\}\,\mathbf{S}_{P,AB} \\
&= \{\mathbf{I}_{B\times B} \otimes [(F_P \otimes \mathbf{I}_{A\times A})\Omega_{A,P}]\}(\mathbf{I}_{B\times B} \otimes \mathbf{S}_{A,P})\,\mathbf{S}_{P,AB}.
\end{aligned}$$

Now we can combine the shuffle permutations. Basically, the left one swaps the first two indices, and the right one cycles all three indices to the left, with the result that only the second and third indices interchange, as follows:

$$
\begin{aligned}
{}_{A\times P\times B}[(\mathbf{I}_{B\times B}\otimes\mathbf{S}_{A,P})\,\mathbf{S}_{P,AB}\mathbf{z}](a,p,b) &= {}_{P\times A\times B}[\mathbf{S}_{P,AB}\mathbf{z}](p,a,b)\\
&= {}_{A\times B\times P}\mathbf{z}(a,b,p)\\
&= {}_{A\times P\times B}[(\mathbf{S}_{P,B}\otimes\mathbf{I}_{A\times A})\mathbf{z}](a,p,b).
\end{aligned}
$$

We substitute the new shuffle in place of the old ones to obtain

$$
\begin{aligned}
\mathbf{T}_k &= \{\mathbf{I}_{B\times B}\otimes[(\boldsymbol{F}_P\otimes\mathbf{I}_{A\times A})\Omega_{A,P}]\}(\mathbf{S}_{P,B}\otimes\mathbf{I}_{A\times A})\\
&= (\mathbf{I}_{B\times B}\otimes\boldsymbol{F}_P\otimes\mathbf{I}_{A\times A})(\mathbf{I}_{B\times B}\otimes\Omega_{A,P})(\mathbf{S}_{P,B}\otimes\mathbf{I}_{A\times A}).
\end{aligned}
\tag{29}
$$

Because of the side proofs, this derivation is a little longer than the proof of our earlier matrix expression (25), but the steps are much easier to verify.

As a check, we can derive (25) from the last result if we notice that

$$
\Omega_{A,P} = \begin{vmatrix} \mathbf{W}_{0,0} & & & \\ & \mathbf{W}_{0,1} & & \\ & & \ddots & \\ & & & \mathbf{W}_{0,P-1} \end{vmatrix}.
$$

We only need to combine the three terms inside the square brackets:

$$
\mathbf{T}_k = \{\mathbf{I}_{B\times B}\otimes[(\boldsymbol{F}_P\otimes\mathbf{I}_{A\times A})\Omega_{A,P}]\}(\mathbf{S}_{P,B}\otimes\mathbf{I}_{A\times A})
$$

$$
(\boldsymbol{F}_P\otimes\mathbf{I}_{A\times A})\Omega_{A,P} = \begin{vmatrix} \omega_P^0\mathbf{I} & \omega_P^0\mathbf{I} & \cdots & \omega_P^0\mathbf{I} \\ \omega_P^0\mathbf{I} & \omega_P^1\mathbf{I} & & \omega_P^{(P-1)}\mathbf{I} \\ \vdots & & \ddots & \vdots \\ \omega_P^0\mathbf{I} & \omega_P^{(P-1)}\mathbf{I} & \cdots & \omega_P^{(P-1)(P-1)}\mathbf{I} \end{vmatrix} \cdot \begin{vmatrix} \mathbf{W}_{0,0} & & & \\ & \mathbf{W}_{0,1} & & \\ & & \ddots & \\ & & & \mathbf{W}_{0,P-1} \end{vmatrix}
$$

$$
(30) \quad = \begin{vmatrix} \omega_P^0 \mathbf{W}_{0,0} & \omega_P^0 \mathbf{W}_{0,1} & \cdots & \omega_P^0 \mathbf{W}_{0,P-1} \\ \omega_P^0 \mathbf{W}_{0,0} & \omega_P^1 \mathbf{W}_{0,1} & & \omega_P^{(P-1)} \mathbf{W}_{0,P-1} \\ \vdots & & \ddots & \vdots \\ \omega_P^0 \mathbf{W}_{0,0} & \omega_P^{(P-1)} \mathbf{W}_{0,1} & \cdots & \omega_P^{(P-1)(P-1)} \mathbf{W}_{0,P-1} \end{vmatrix}
= \begin{vmatrix} \mathbf{W}_{0,0} & \mathbf{W}_{0,1} & \cdots & \mathbf{W}_{0,P-1} \\ \mathbf{W}_{1,0} & \mathbf{W}_{1,1} & & \mathbf{W}_{1,P-1} \\ \vdots & & \ddots & \vdots \\ \mathbf{W}_{P-1,0} & \mathbf{W}_{P-1,1} & \cdots & \mathbf{W}_{P-1,P-1} \end{vmatrix} = \mathbf{Q}_{A,P}
$$

In this way, we can demonstrate again that

$$\mathbf{T}_k = (\mathbf{I}_{B\times B} \otimes \mathbf{Q}_{A,P})(\mathbf{S}_{P,B} \otimes \mathbf{I}_{A\times A}).$$

Thus, the two different forms of $\mathbf{T}_k$ we've found are, in fact, interchangeable.

7. Zipper identities.

If we take de Boor's algorithm and apply it to the simple case $M = 2$, we can reduce the product of the two intermatrices to simpler, memorable forms called "zipper" identities. For example, the Fourier transform of an RQ-vector using equation (29) reduces like this:

$$F_{RQ} = \mathbf{T}_1\mathbf{T}_2 = \{\mathbf{I}_{1\times1} \otimes [(F_R \otimes \mathbf{I}_{Q\times Q})\Omega_{Q,R}]\}(\mathbf{S}_{R,1} \otimes \mathbf{I}_{Q\times Q}) \cdot$$
$$\{\mathbf{I}_{R\times R} \otimes [(F_Q \otimes \mathbf{I}_{1\times1})\Omega_{1,Q}]\}(\mathbf{S}_{Q,R} \otimes \mathbf{I}_{1\times1}).$$

Remembering that $\Omega_{1,Q} = \mathbf{I}_Q$, we simplify this to

$$F_{RQ} = [(F_R \otimes \mathbf{I}_{Q\times Q})\Omega_{Q,R}](\mathbf{I}_{RQ\times RQ})(\mathbf{I}_{R\times R} \otimes F_Q)\mathbf{S}_{Q,R}.$$

and finally we get Kammler's "twiddle factor" zipper identity [3]

$$(31) \quad F_{RQ} = (F_R \otimes \mathbf{I}_{Q\times Q})\Omega_{Q,R}(\mathbf{I}_{R\times R} \otimes F_Q)\mathbf{S}_{Q,R}.$$

If we work out the simple case of our other intermatrix identity (25), it reduces as follows:

$$\begin{aligned}
\boldsymbol{F}_{PQ} &= [(\mathbf{I}_{1\times1} \otimes \mathbf{Q}_{Q,P})(\mathbf{S}_{P,1} \otimes \mathbf{I}_{Q\times Q})][(\mathbf{I}_{P\times P} \otimes \mathbf{Q}_{1,Q})(\mathbf{S}_{Q,P} \otimes \mathbf{I}_{1\times1})] \\
&= [(\mathbf{Q}_{Q,P})(\mathbf{I}_{P\times P} \otimes \mathbf{I}_{Q\times Q})][(\mathbf{I}_{P\times P} \otimes \mathbf{Q}_{1,Q})(\mathbf{S}_{Q,P})] \\
&= (\mathbf{Q}_{Q,P})(\mathbf{I}_{P\times P} \otimes \mathbf{Q}_{1,Q})(\mathbf{S}_{Q,P}).
\end{aligned}$$

Here we observe using the definitions of $\mathbf{W}$ (23) and $\mathbf{Q}$ (25) that when $A = 1$, $\mathbf{W}_{k,l} = \omega^{kl}$, and thus

$$\mathbf{Q}_{1,Q} = \begin{vmatrix} 1 & \omega_Q^1 & \cdots & \omega_Q^{Q-1} \\ \omega_Q^1 & \omega_Q^2 & & \omega_Q^{2(Q-1)} \\ \vdots & & \ddots & \vdots \\ \omega_Q^{(Q-1)} & \omega_Q^{2(Q-1)} & \cdots & \omega_Q^{(Q-1)(Q-1)} \end{vmatrix} = \boldsymbol{F}_Q.$$

Therefore, this formula yields another zipper identity.

(32) $\quad \boldsymbol{F}_{PQ} = \mathbf{Q}_{Q,P}\boldsymbol{F}_Q^{(P)}\mathbf{S}_{Q,P}$

The reason for the name of this equation and (31) now becomes obvious: if we use a zipper identity to find the Fourier transform of a $P_1P_2P_3\cdots P_M$-vector, the formula will contain a Fourier transform operating on a $P_2P_3\cdots P_M$-vector, which will be replaced by essentially the same formula, containing a Fourier transform matrix of size $P_3\cdots P_M$, and so on. As the formula is simplified to the end, when it contains only $\mathbf{Q}$ and $\mathbf{S}$ terms, each of the original M factors is peeled off through the P variable in the zipper identity (32), and the $\boldsymbol{F}$ term continually "unzips", throwing $\mathbf{Q}$ matrices to the left and $\mathbf{S}$ matrices to the right.

8. The FFT as a product of intermatrices.

The complete FFT, of course, can be written by stringing the intermatrices for each k in order.

$$\begin{aligned}\mathbf{F}_N &= [(\mathbf{I}_{B_1\times B_1}\otimes\mathbf{F}_{P_1}\otimes\mathbf{I}_{A_1\times A_1})(\mathbf{I}_{B_1\times B_1}\otimes\Omega_{A_1,P_1})(\mathbf{S}_{P_1,B_1}\otimes\mathbf{I}_{A_1\times A_1})]\cdot\\ &\quad [(\mathbf{I}_{B_2\times B_2}\otimes\mathbf{F}_{P_2}\otimes\mathbf{I}_{A_2\times A_2})(\mathbf{I}_{B_2\times B_2}\otimes\Omega_{A_2,P_2})(\mathbf{S}_{P_2,B_2}\otimes\mathbf{I}_{A_2\times A_2})]\cdot\ \cdots\ \cdot\\ &\quad [(\mathbf{I}_{B_M\times B_M}\otimes\mathbf{F}_{P_M}\otimes\mathbf{I}_{A_M\times A_M})(\mathbf{I}_{B_M\times B_M}\otimes\Omega_{A_M,P_M})(\mathbf{S}_{P_M,B_M}\otimes\mathbf{I}_{A_M\times A_M})].\end{aligned}\tag{33}$$

In some cases, this might be the most useful way to write the Fourier transform, since we can apply the same formula M times, only adjusting the A, B, and P variables. We could also, however, want to rearrange this formula, and collect all of the shuffles, for example. Therefore, we want to find out how these matrices commute. This is not difficult since each matrix contains an identity term in the Kronecker product, which can be rewritten as the product of two smaller identities. Let's examine how a shuffle commutes with a Fourier term:

$$\begin{aligned}&(\mathbf{S}_{P_i,B_i}\otimes\mathbf{I}_{A_i\times A_i})(\mathbf{I}_{B_j\times B_j}\otimes\mathbf{F}_{P_j}\otimes\mathbf{I}_{A_j\times A_j})\\ &= (\mathbf{S}_{P_i,B_i}\otimes\mathbf{I}_{A_i\times A_i})(\mathbf{I}_{B_{i+1}\times B_{i+1}}\otimes\mathbf{I}_{P_{i+1}\cdots P_{j-1}}\otimes\mathbf{F}_{P_j}\otimes\mathbf{I}_{A_j\times A_j}),\ \text{assuming } i<j\\ &= [(\mathbf{S}_{P_i,B_i})(\mathbf{I}_{B_{i+1}\times B_{i+1}})]\otimes[(\mathbf{I}_{A_i\times A_i})(\mathbf{I}_{P_{i+1}\cdots P_{j-1}}\otimes \boldsymbol{F}_{P_j}\otimes\mathbf{I}_{A_j\times A_j})]\\ &= [(\mathbf{I}_{B_{i+1}\times B_{i+1}})(\mathbf{S}_{P_i,B_i})]\otimes[(\mathbf{I}_{P_{i+1}\cdots P_{j-1}}\otimes \boldsymbol{F}_{P_j}\otimes\mathbf{I}_{A_j\times A_j})(\mathbf{I}_{A_i\times A_i})]\\ &= (\mathbf{I}_{B_{i+1}\times B_{i+1}}\otimes\mathbf{I}_{P_{i+1}\cdots P_{j-1}}\otimes \boldsymbol{F}_{P_j}\otimes\mathbf{I}_{A_j\times A_j})(\mathbf{S}_{P_i,B_i}\otimes\mathbf{I}_{A_i\times A_i})\\ &= (\mathbf{I}_{B_j\times B_j}\otimes \boldsymbol{F}_{P_j}\otimes\mathbf{I}_{A_j\times A_j})(\mathbf{S}_{P_i,B_i}\otimes\mathbf{I}_{A_i\times A_i}).\end{aligned}$$

Thus, a shuffle term commutes freely with any Fourier transform term with a larger k value, basically because the identity portions of the shuffle and the Fourier terms are large enough to overlap. This result would be far from obvious if we examined the matrices involved directly, but it is easy to see using the properties of the Kronecker product.

Similarly, we can break up the **I** factor of the Ω term in the same way, and demonstrate that the shuffle will commute with all of the higher-numbered Ω terms. Since the shuffle commutes freely with all of the non-shuffle terms to the right in the equation, we can collect the shuffle permutations at the end of the equation:

$$(34) \qquad \boldsymbol{F}_N = \prod_{k=1}^{M} (\mathbf{I}_{B_k \times B_k} \otimes \boldsymbol{F}_{P_k} \otimes \mathbf{I}_{A_k \times A_k})(\mathbf{I}_{B_k \times B_k} \otimes \Omega_{A_k, P_k}) \cdot \prod_{k=1}^{M} (\mathbf{S}_{P_k, B_k} \otimes \mathbf{I}_{A_k \times A_k}).$$

The Ω term would commute with lower-numbered $\boldsymbol{F}$ terms; unfortunately, the $\boldsymbol{F}$ term with the same value of K stands in the way, preventing the Ω terms from being collected on the left. We can still use equation (30), which we derived when we showed that the two intermatrix expressions we found were interchangeable, to obtain:

$$(35) \qquad \boldsymbol{F}_N = \prod_{k=1}^{M} (\mathbf{I}_{B_k \times B_k} \otimes \mathbf{Q}_{A_k, P_k}) \cdot \prod_{k=1}^{M} (\mathbf{S}_{P_k, B_k} \otimes \mathbf{I}_{A_k \times A_k}).$$

This form of the DFT can also be found by expanding the zipper identity (31).

Since we cannot commute the matrices any further, let's reexamine the collected shuffle permutations. Let's extend the indexing system we used earlier to express vectors as 2- or 3-dimensional matrices to the M-dimensional case; then we could write

$$_{P_1 \times P_2 \times \cdots \times P_M}\mathbf{z}(p_1, p_2, \cdots, p_M) = \mathbf{z}[p_1 + P_1(p_2 - 1) + \cdots + P_1 P_2 \cdot \cdots \cdot P_M(p_M - 1)].$$

We analyze the effect of the shuffles on an arbitrary vector **z**, as usual, but express the result M-dimensionally:

$$\underset{P_M\times P_{M-1}\times\cdots\times P_1}{}\left\{\left[\prod_{k=1}^{M}(\mathbf{S}_{P_k,B_k}\otimes\mathbf{I}_{A_k\times A_k})\right]\mathbf{z}\right\}(p_M,p_{M-1},\cdots,p_2,p_1)$$
$$=\underset{P_M\times P_{M-1}\times\cdots\times P_1}{}\left\{\left[\prod_{k=2}^{M}(\mathbf{S}_{P_k,B_k}\otimes\mathbf{I}_{A_k\times A_k})\right]\mathbf{z}\right\}(p_M,p_{M-1},\cdots,p_2,p_1),$$

since the first shuffle reduces to an identity. The second shuffle is not an identity; rather, it swaps the last two indices:

$$=\underset{P_M\times P_{M-1}\times\cdots\times P_1\times P_2}{}\left\{\left[\prod_{k=3}^{M}(\mathbf{S}_{P_k,B_k}\otimes\mathbf{I}_{A_k\times A_k})\right]\mathbf{z}\right\}(p_M,p_{M-1},\cdots,p_1,p_2).$$

Furthermore, the third shuffle cycles the last three indices:

$$=\underset{P_M\times P_{M-1}\times\cdots\times P_1\times P_2\times P_3}{}\left\{\left[\prod_{k=4}^{M}(\mathbf{S}_{P_k,B_k}\otimes\mathbf{I}_{A_k\times A_k})\right]\mathbf{z}\right\}(p_M,p_{M-1},\cdots,p_1,p_2,p_3).$$

The pattern continues; after the application of the shuffles for k = 1, 2, up to l are eliminated, the final l indices turn up in reverse order. The process continues to the logical end:

$$\underset{P_M\times P_{M-1}\times\cdots\times P_1}{}\left\{\left[\prod_{k=1}^{M}(\mathbf{S}_{P_k,B_k}\otimes\mathbf{I}_{A_k\times A_k})\right]\mathbf{z}\right\}(p_M,p_{M-1},\cdots,p_2,p_1)$$
$$(36)\quad =_{P_M\times P_1\times\cdots\times P_{M-1}}\left\{\left[\mathbf{S}_{P_M,B_M}\otimes\mathbf{I}_{A_M\times A_M}\right]\mathbf{z}\right\}(p_M,p_1,\cdots,p_{M-2},p_{M-1})$$
$$=_{P_1\times\cdots\times P_M}\{\mathbf{z}\}(p_1,\cdots,p_{M-1},p_M).$$

Therefore, the set of shuffle permutations simply reverses the order of the indexing that we used. In fact, this is a generalization of the idea behind the familiar "bit shuffle" permutation used when $P_1 = P_2 = \cdots = P_M = 2$, which Kammler [3] represents by **B**. If we replace the shuffles by the shorthand representation **B**, the entire Fourier transform is written

(37) $$F_N = \left\{ \prod_{k=1}^{M} (\mathbf{I}_{B_k \times B_k} \otimes \mathbf{Q}_{A_k, P_k}) \right\} \cdot \mathbf{B},$$

which we note is the version of the transform used extensively by Eubanks [2].

9. Summary.

Carl de Boor's algorithm is tremendously useful because it provides a simple way to factor a mixed-radix FFT into sparse matrices that can be applied to a vector one at a time in an orderly fashion, while still leaving the meaning of the intermediate results clear after each step. We can multiply the factors in an iterative procedure, if so desired, or we can collect the components which "shuffle" the intermediate results at the end, making the total FFT simpler in appearance. In the latter case, the reshuffling at the end takes the form of a "bit shuffle."

The elegance behind de Boor's algorithm is revealed when we apply powerful matrix notation, specifically Kronecker products, to simplify the intricate matrices involved. Using simple properties of this notation, we can manipulate the results and express the final form in a logical, readily understandable way, which would be much more difficult and far from obvious without Kronecker products and shuffle permutations. These matrix tools are applied, and simple properties demonstrated, using de Boor's vector indexing system, which is used to create a 2- or 3-dimensional array isomorphic to the N-vector. When we extend his vector-as-matrix concept beyond the 3-dimensional case to M dimensions, the exact operation of the bit shuffle is made obvious.

Improvements in the notation could make de Boor's algorithm clearer. A readable way to index the components of a matrix, in the same way de Boor indexes vectors, would make the proofs of elementary properties of Kronecker products much more intuitive. Also, it might be possible to devise a shuffle permutation notation that would make the interactions of various shuffle permutations easier to visualize.

REFERENCES

[1] CARL DE BOOR, *FFT as nested multiplication, with a twist,* SIAM J. Sci. Stat. Comput., 1 (1980), pp. 173-178.

[2] D. A. EUBANKS, *An analysis of de Boor's algorithm for computing the mixed radix FFT*, M. S. Thesis, Southern Illinois University at Carbondale, Carbondale, 1986.

[3] DAVID KAMMLER, *A First Course in Fourier Analysis,* Prentice-Hall, to appear.