Appendix A

Recursive Formulas and Algorithms for the Transformation Based Planar Array Synthesis Technique

The notationally complex derivations and algorithmic details of the transformation based synthesis technique, discussed in Chapter 3, are given in this Appendix. As in Chapter 3, the two planar array categories will be treated separately. The odd case (an odd number of elements along each principal plane) will be discussed in Section A.1 and even case (an even number of elements along each principal plane) in Section A.2. The derivation of the formulas, as well as, "ready to implement" algorithms for b_q, c_{mn} and c^{xx}_{mn} for both cases are supplied. Although it may be possible to write these in what may be considered a more mathematically elegant fashion, such recursion relations are ideally suited to computation.

A.1 Formulas and Algorithms : The Odd Case

A.1.1 Computation of b_q

The prototype linear array factor, a summation of cosines weighted by the relative excitations a_q, can be expressed in a polynomial form, with b_q the coefficients and $\cos\psi_p$ the variable of the polynomial.

The prototype linear array is a $2Q+1$ element, uniformly spaced linear array with symmetrical excitation and with inter-element spacing d. The prototype linear array factor can be in the usual form (3.4) or in a polynomial form (3.6)

$$F_p(\psi_p) = \sum_{q=0}^{Q} \zeta_q \, a_q \, \cos(q\psi_p) = \sum_{q=0}^{Q} b_q \cos^q \psi_p \tag{A.1}$$

The recurrence relations (3.5) can be written in one equation

$$\cos^n x = \frac{1}{2^n} \sum_{i=0}^{n}{}^{2} \zeta_i \binom{n}{\frac{1}{2}(n-i)} \cos(ix) \tag{A.2}$$

where $\sum^2$ indicates a step size of two (i.e. $i=i+2$). Using the recurrence relation (A.2) the prototype linear array factor in a polynomial form (A.1) can be expressed as

$$F_p(\psi_p) = \sum_{q=0}^{Q} \sum_{i=0}^{q}{}^{2} \frac{\zeta_q\, b_i}{2^q} \binom{i}{\frac{1}{2}(q-i)} \cos(i\psi_p) \tag{A.3}$$

but for any function of two indices $f(m,n)$,

$$\sum_{m=0}^{M} \sum_{n=0}^{m} f(m,n) = \sum_{n=0}^{M} \sum_{m=n}^{M} f(m,n) \tag{A.4}$$

hence, the summations on the right side of (A.3) can be changed,

$$F_p(\psi_p) = \sum_{q=0}^{Q} \left\{ \sum_{i=q}^{Q}{}^{2} \frac{\zeta_q\, b_i}{2^i} \binom{i}{\frac{1}{2}(i-q)} \right\} \cos(q\psi_p) \tag{A.5}$$

The equation is matched with the prototype linear array factor (A.1) to obtain b_q. The b_q values are computed from the prototype linear array excitations, a_q, with the recursive formula

$$b_q = 2^q \left[a_q - \sum_{i=q+2}^{Q}{}^{2} \frac{b_i}{2^i} \binom{i}{\frac{1}{2}(i-q)} \right] \tag{A.6}$$

in the order $q=Q, Q-1, \ldots, 0$.

A.1.2 Computation of c_{mn} for Quadrantal Symmetric Contours

Derivation of the Formulas

This section deals with the formulas necessary for the computation of c_{mn} of Section 3.2.1. Although these formulas appeared in[95, 94] they are included here for completeness. The only information needed to compute these coefficients are the b_q values forthcoming from the prototype linear array excitations, and the transformation parameters, t_{ij}.

In order to structure the computation let h_{kl} denominate the coefficient of the general term $\cos ku \cos lv$, h_{kl}^q the of $[H(u,v)]^q$. $[H(u,v)]^q$ can then be written as

$$\begin{aligned} [H(u,v)]^q &= [H(u,v)]^{q-1} \times H(u,v) \\ &= \sum_{k=0}^{(q-1)I} \sum_{l=0}^{(q-1)J} h_{kl}^{q-1} \cos(ku)\cos(lv) \times \sum_{i=0}^{I} \sum_{j=0}^{J} t_{ij} \cos(iu)\cos(jv) \end{aligned} \tag{A.7}$$

The use of the recurrence relations (3.5) enable $[H(u,v)]^q$ to be written in the subsequent form

$$[H(u,v)]^q = \sum_{k=0}^{Iq} \sum_{l=0}^{Jq} h_{kl}^q \cos(ku)\cos(lv) \tag{A.8}$$

Algorithm for the computation of c_{mn}

- Step #1: Initiate

$$c_{mn} = 0 \ \text{ for } \begin{cases} m = 0,1,2,\dots,QI \\ n = 0,1,2,\dots,QJ \end{cases}$$

$$h_{kl}^q = 0 \ \text{ for } \begin{cases} k = 0,1,2,\dots,QI \\ l = 0,1,2,\dots,QJ \\ q = 0,1,2,\dots,Q \end{cases}$$

$$c_{00} = b_0$$

- Step #2: $q = 1$

$$h_{ij}^q = t_{ij} \ \text{ for } \begin{cases} i = 0,1,2,\dots,I \\ j = 0,1,2,\dots,J \end{cases}$$

$$c_{mn} = c_{mn} + b_q h_{mn}^q \ \text{ for } \begin{cases} m = 0,1,2,\dots,qI \\ n = 0,1,2,\dots,qJ \end{cases}$$

- Step #3: $q = q + 1$

$$h_{mn}^q = h_{mn}^q + \frac{1}{\zeta_{|i|}\zeta_{|j|}} t_{|i||j|} h_{kl}^{q-1} \ \text{ for } \begin{cases} i = -I,-(I-1),\dots,-1,0,1,\dots,I \\ j = -I,-(I-1),\dots,-1,0,1,\dots,J \\ k = 0,1,2,\dots,(q-1)I \\ l = 0,1,2,\dots,(q-1)J \\ m = |k+i| \\ n = |l+j| \end{cases}$$

$$c_{mn} = c_{mn} + b_q h_{mn}^q \ \text{ for } \begin{cases} m = 0,1,2,\dots,qI \\ n = 0,1,2,\dots,qJ \end{cases}$$

Repeat Step #3 up to, and including, the case $q = Q$. As only the information of the previous iteration is needed to compute the values of the current iteration, only two matrices for h_{kl} are needed. The first matrix contains the values of the previous iteration and the second matrix the updated values of the current iteration.

A.1.3 Computation of c_{mn}^{xx} for Arbitrary Contours

Derivation of the Formulas

The algorithms and formulas necessary to compute c_{mn}^{cc}, c_{mn}^{ss}, c_{mn}^{cs} and c_{mn}^{sc} of Section 3.2.2 are provided in this section. The only data used in these formulas are are the b_q values (forthcoming from the prototype linear array excitations) and the transformation parameters, t_{ij}^{cc}, t_{ij}^{ss}, t_{ij}^{cs} and t_{ij}^{sc}.

To structure the computation of these coefficients let g_{kl}^q denominate the coefficient of the general term $\cos ku \cos lv$, h_{kl}^q the coefficient of the the general term $\sin ku \sin lv$, r_{kl}^q the coefficient of the the general term $\cos ku \sin lv$ and s_{kl}^q the coefficient of the the general term $\sin ku \cos lv$ in $[H(u,v)]^q$. $[H(u,v)]^q$ is then formulated as

$$
\begin{aligned}
[H(u,v)]^q &= [H(u,v)]^{q-1} \times H(u,v) \\
&= \sum_{k=0}^{(q-1)I} \sum_{l=0}^{(q-1)J} [g_{kl}^{q-1} \cos(ku)\cos(lv) + h_{kl}^{q-1} \sin(ku)\sin(lv) + \\
&\qquad r_{kl}^{q-1} \cos(ku)\sin(lv) + s_{kl}^{q-1} \sin(ku)\cos(lv)] \\
&\quad \times \sum_{i=0}^{I} \sum_{j=0}^{J} [t_{ij}^{cc} \cos(iu)\cos(jv) + t_{ij}^{ss} \sin(iu)\sin(jv) + \\
&\qquad t_{ij}^{cs} \cos(iu)\sin(jv) + t_{ij}^{sc} \sin(iu)\cos(jv)]
\end{aligned}
\tag{A.9}
$$

With the use of the following relations

$$
\begin{aligned}
\cos A \cos B &= \tfrac{1}{2}[\cos(A-B) + \cos(A+B)] \\
\sin A \cos B &= \tfrac{1}{2}[\sin(A-B) + \sin(A+B)] \\
\sin A \sin B &= \tfrac{1}{2}[\sin(A-B) + \sin(A+B)]
\end{aligned}
\tag{A.10}
$$

$[H(u,v)]^q$ can then be written in the subsequent form

$$
\begin{aligned}
[H(u,v)]^q = \sum_{k=0}^{Iq} \sum_{l=0}^{Jq} & [g_{kl}^q \cos(ku)\cos(lv) + h_{kl}^q \sin(ku)\sin(lv) + \\
& r_{kl}^q \cos(ku)\sin(lv) + s_{kl}^q \sin(ku)\cos(lv)]
\end{aligned}
\tag{A.11}
$$

Algorithm for the computation of c_{mn}^{xx}

- Step #1: Initiate

$$\left.\begin{aligned} c_{mn}^{cc} &= 0 \\ c_{mn}^{ss} &= 0 \\ c_{mn}^{cs} &= 0 \\ c_{mn}^{sc} &= 0 \end{aligned}\right\} \text{for} \left\{\begin{aligned} m &= 0,1,2,\dots,QI \\ n &= 0,1,2,\dots,QJ \end{aligned}\right.$$

$$\left.\begin{aligned} g_{kl}^{q} &= 0 \\ h_{kl}^{q} &= 0 \\ r_{kl}^{q} &= 0 \\ s_{kl}^{q} &= 0 \end{aligned}\right\} \text{for} \left\{\begin{aligned} k &= 0,1,2,\dots,QI \\ l &= 0,1,2,\dots,QJ \\ q &= 0,1,2,\dots,Q \end{aligned}\right.$$

$$c_{00} = b_0$$

- Step #2: $q = 1$

$$\left.\begin{aligned} g_{ij}^{q} &= t_{ij}^{cc} \\ h_{ij}^{q} &= t_{ij}^{ss} \\ r_{ij}^{q} &= t_{ij}^{cs} \\ s_{ij}^{q} &= t_{ij}^{sc} \end{aligned}\right\} \text{for} \left\{\begin{aligned} i &= 0,1,2,\dots,I \\ j &= 0,1,2,\dots,J \end{aligned}\right.$$

$$\left.\begin{aligned} c_{mn}^{cc} &= c_{mn}^{cc} + b_q g_{mn}^{q} \\ c_{mn}^{ss} &= c_{mn}^{ss} + b_q h_{mn}^{q} \\ c_{mn}^{cs} &= c_{mn}^{cs} + b_q r_{mn}^{q} \\ c_{mn}^{sc} &= c_{mn}^{sc} + b_q s_{mn}^{q} \end{aligned}\right\} \text{for} \left\{\begin{aligned} m &= 0,1,2,\dots,qI \\ n &= 0,1,2,\dots,qJ \end{aligned}\right.$$

- Step #3: $q = q + 1$

$$g_{mn}^{q} = g_{mn}^{q} + \frac{1}{\varsigma_{|i|}\varsigma_{|j|}}\left[t_{|i||j|}^{cc}\, g_{kl}^{q-1} + \Upsilon_i \Upsilon_j\, t_{|i||j|}^{ss}\, h_{kl}^{q-1} - \Upsilon_j\, t_{|i||j|}^{cs}\, r_{kl}^{q-1} - \Upsilon_i\, t_{|i||j|}^{sc}\, s_{kl}^{q-1}\right]$$

$$h_{mn}^{q} = h_{mn}^{q} + \frac{\Upsilon_{k+i}\Upsilon_{l+j}}{\varsigma_{|i|}\varsigma_{|j|}}\left[t_{|i||j|}^{cc}\, h_{kl}^{q-1} + \Upsilon_i \Upsilon_j\, t_{|i||j|}^{ss}\, g_{kl}^{q-1} + \Upsilon_j\, t_{|i||j|}^{cs}\, s_{kl}^{q-1} + \Upsilon_i\, t_{|i||j|}^{sc}\, r_{kl}^{q-1}\right]$$

$$r_{mn}^{q} = r_{mn}^{q} + \frac{\Upsilon_{l+j}}{\varsigma_{|i|}\varsigma_{|j|}}\left[t_{|i||j|}^{cc}\, r_{kl}^{q-1} - \Upsilon_i \Upsilon_j\, t_{|i||j|}^{ss}\, s_{kl}^{q-1} + \Upsilon_j\, t_{|i||j|}^{cs}\, g_{kl}^{q-1} - \Upsilon_i\, t_{|i||j|}^{sc}\, h_{kl}^{q-1}\right]$$

$$s_{mn}^{q} = s_{mn}^{q} + \frac{\Upsilon_{k+i}}{\varsigma_{|i|}\varsigma_{|j|}}\left[t_{|i||j|}^{cc}\, s_{kl}^{q-1} - \Upsilon_i \Upsilon_j\, t_{|i||j|}^{ss}\, r_{kl}^{q-1} - \Upsilon_j\, t_{|i||j|}^{cs}\, h_{kl}^{q-1} + \Upsilon_i\, t_{|i||j|}^{sc}\, g_{kl}^{q-1}\right]$$

$$\text{for}\begin{cases} i=-I,-(I-1),\dots,-1,0,1,\dots,I \\ j=-J,-(J-1),\dots,-1,0,1,\dots,J \\ k=0,1,2,\dots,(q-1)I \\ l=0,1,2,\dots,(q-1)J \end{cases} \quad \text{with}\begin{cases} m=|k+i| \\ n=|l+j| \\ \Upsilon_i=\operatorname{sign}(i) \end{cases}$$

$$\left.\begin{aligned} c^{cc}_{mn} &= c^{cc}_{mn}+b_q g^q_{mn} \\ c^{ss}_{mn} &= c^{ss}_{mn}+b_q h^q_{mn} \\ c^{cs}_{mn} &= c^{cs}_{mn}+b_q r^q_{mn} \\ c^{sc}_{mn} &= c^{sc}_{mn}+b_q s^q_{mn} \end{aligned}\right\} \text{for}\begin{cases} m=0,1,2,\dots,qI \\ n=0,1,2,\dots,qJ \end{cases}$$

Repeat Step #3 up to, and including, $q=Q$. Note that only two matrices for each of h_{kl}, g_{kl} rkl and s_{kl} are needed. The first matrix of each holds the values of the previous iteration and the second matrix holds the revised values of the current iteration. Although this algorithm seems formidable, it is easy to program and execution is extremely rapid.

A.2 Formulas and Algorithms : The Even Case

A.2.1 Computation of b_q

In this case, the prototype linear array is a $2Q$ element, uniformly spaced, symmetrical excited linear array. The prototype linear array factor can be expressed in two forms, the usual form (3.17) or the polynomial form (3.18)

$$F_p(\psi_p)=2\sum_{q=1}^{Q}a_q\cos\left[\tfrac{1}{2}(2q-1)\psi_p\right]=\sum_{q=1}^{Q}b_q\cos^{2q-1}\left(\tfrac{1}{2}\psi_p\right) \tag{A.12}$$

Using the same procedure (only latter of the recurrence relations (3.5b) is needed) the recursive formula for computing the b_q values from the prototype linear array excitations is,

$$b_q=2^{2q-1}\left[a_q-\sum_{i=q+1}^{Q}\frac{b_i}{2^{2i-1}}\binom{2i-1}{i-q}\right] \tag{A.13}$$

in the order $q=Q,Q-1,\dots,0$.

A.2.2 Computation of c_{mn} for Quadrantal Symmetric Contours

Derivation of the Formulas

The formulas required for the computation of c_{mn} coefficients of Section 3.2.1 are presented this section. The information forthcoming from the solution of the two subproblems, are all that are needed to calculate these coefficients.

To structure the computation, let h_{kl} denominate the coefficient of the general term $\cos[\frac{1}{2}(2k-1)u]\cos[\frac{1}{2}(2l-1)v]$ of $[H(u,v)]^{2q-1}$. $[H(u,v)]^{2q-1}$ is then expressed as

$$\begin{aligned}
[H(u,v)]^{2q-1} &= [H(u,v)]^{2q-3} \times [H(u,v)]^2 \\
&= \sum_{k=1}^{(2q-3)I} \sum_{l=1}^{(2q-3)J} h_{kl}^{2q-3} \cos\left[\tfrac{1}{2}(2k-1)u\right] \cos\left[\tfrac{1}{2}(2l-1)v\right] \\
&\quad \times \left\{ \sum_{i=1}^{I} \sum_{j=1}^{J} t_{ij} \cos\left[\tfrac{1}{2}(2i-1)u\right] \cos\left[\tfrac{1}{2}(2j-1)v\right] \right\}^2
\end{aligned} \tag{A.14}$$

Using the recurrence relations (3.5b) enable $[H(u,v)]^{2q-1}$ to be written in the subsequent form

$$[H(u,v)]^{2q-1} = \sum_{k=1}^{(2I-1)q} \sum_{l=1}^{(2J-1)q} h_{kl}^{2q-1} \cos\left[\tfrac{1}{2}(2k-1)u\right] \cos\left[\tfrac{1}{2}(2l-1)v\right] \tag{A.15}$$

It will be easier, and computationally faster, to compute $[H(u,v)]^2$ and use its coefficients, t_{ij}, where t_{ij} is the coefficient of the general term $\cos(iu)\cos(jv)$, in the algorithm.

$$[H(u,v)]^2 = \sum_{i=0}^{2I-1} \sum_{j=0}^{2J-1} \mathrm{t}_{ij} \cos(iu)\cos(jv) \tag{A.16}$$

Algorithm for the computation of c_{mn}

- Step #1: Initiate

$$c_{mn} = 0 \ \text{ for } \begin{cases} m = 1,2,3,\dots,QI \\ n = 1,2,3,\dots,QJ \end{cases}$$

$$h_{kl}^{q} = 0 \ \text{ for } \begin{cases} k = 1,2,3,\dots,QI \\ l = 1,2,3,\dots,QJ \\ q = 1,2,3,\dots,Q \end{cases}$$

$$\mathrm{t}_i j =$$

- Step #2: $q = 1$

$$h^q_{ij} = t_{ij} \text{ for } \begin{cases} i = 1, 2, 3, \ldots, I \\ j = 1, 2, 3, \ldots, J \end{cases}$$

$$c_{mn} = c_{mn} + b_q h^q_{mn} \text{ for } \begin{cases} m = 1, 2, 3, \ldots, qI \\ n = 1, 2, 3, \ldots, qJ \end{cases}$$

- Step #3: $q = q + 1$

$$h^q_{mn} = h^q_{mn} + \frac{1}{4} t_{ij} h^{q-1}_{kl} \text{ for } \begin{cases} i = -I, -(I-1), \ldots, -1, 1, \ldots, I \\ j = -I, -(I-1), \ldots, -1, 1, \ldots, J \\ k = 1, 2, 3, \ldots, (q-1)I \\ l = 1, 2, 3, \ldots, (q-1)J \\ m = |k + i| \\ n = |l + j| \end{cases}$$

$$c_{mn} = c_{mn} + b_q h^q_{mn} \text{ for } \begin{cases} m = 1, 2, 3, \ldots, qI \\ n = 1, 2, 3, \ldots, qJ \end{cases}$$

Repeat Step #3 until $q = Q$. Only two matrices for h_{kl} are needed, one with present values and the other with the previous iteration's values.

A.2.3 Computation of c^{xx}_{mn} for Arbitrary Contours

No details of the derivation of the formulas will be presented as they are analogous to that of Section A.1.3. The algorithm for the computation of c^{xx}_{mn} is:

- Step #1: Initiate

$$\left.\begin{array}{l} c^{cc}_{mn} = 0 \\ c^{ss}_{mn} = 0 \\ c^{cs}_{mn} = 0 \\ c^{sc}_{mn} = 0 \end{array}\right\} \text{ for } \begin{cases} m = 1, 2, 3, \ldots, QI \\ n = 1, 2, 3, \ldots, QJ \end{cases}$$

$$\left.\begin{array}{l} g^q_{kl} = 0 \\ h^q_{kl} = 0 \\ r^q_{kl} = 0 \\ s^q_{kl} = 0 \end{array}\right\} \text{ for } \begin{cases} k = 1, 2, 3, \ldots, QI \\ l = 1, 2, 3, \ldots, QJ \\ q = 1, 2, 3, \ldots, Q \end{cases}$$

$$c_{00} = b_0$$

- Step #2: $q = 1$

$$\left.\begin{aligned} g^q_{ij} &= t^{cc}_{ij} \\ h^q_{ij} &= t^{ss}_{ij} \\ r^q_{ij} &= t^{cs}_{ij} \\ s^q_{ij} &= t^{sc}_{ij} \end{aligned}\right\} \text{ for } \begin{cases} i = 0, 1, 2, \dots, I \\ j = 0, 1, 2, \dots, J \end{cases}$$

$$\left.\begin{aligned} c^{cc}_{mn} &= c^{cc}_{mn} + b_q g^q_{mn} \\ c^{ss}_{mn} &= c^{ss}_{mn} + b_q h^q_{mn} \\ c^{cs}_{mn} &= c^{cs}_{mn} + b_q r^q_{mn} \\ c^{sc}_{mn} &= c^{sc}_{mn} + b_q s^q_{mn} \end{aligned}\right\} \text{ for } \begin{cases} m = 1, 2, 3, \dots, qI \\ n = 1, 2, 3, \dots, qJ \end{cases}$$

- Step #3: $q = q + 1$

$$g^q_{mn} = g^q_{mn} + \tfrac{1}{\varsigma_{|i|}\varsigma_{|j|}} \left[\ t^{cc}_{|i||j|}\, g^{q-1}_{kl} + \Upsilon_i \Upsilon_j\, t^{ss}_{|i||j|}\, h^{q-1}_{kl} - \Upsilon_j\, t^{cs}_{|i||j|}\, r^{q-1}_{kl} - \Upsilon_i\, t^{sc}_{|i||j|}\, s^{q-1}_{kl}\right]$$

$$h^q_{mn} = h^q_{mn} + \tfrac{\Upsilon_{k+i}\Upsilon_{l+j}}{\varsigma_{|i|}\varsigma_{|j|}} \left[t^{cc}_{|i||j|}\, h^{q-1}_{kl} + \Upsilon_i \Upsilon_j\, t^{ss}_{|i||j|}\, g^{q-1}_{kl} + \Upsilon_j\, t^{cs}_{|i||j|}\, s^{q-1}_{kl} + \Upsilon_i\, t^{sc}_{|i||j|}\, r^{q-1}_{kl}\right]$$

$$r^q_{mn} = r^q_{mn} + \tfrac{\Upsilon_{l+j}}{\varsigma_{|i|}\varsigma_{|j|}} \left[\ t^{cc}_{|i||j|}\, r^{q-1}_{kl} - \Upsilon_i \Upsilon_j\, t^{ss}_{|i||j|}\, s^{q-1}_{kl} + \Upsilon_j\, t^{cs}_{|i||j|}\, g^{q-1}_{kl} - \Upsilon_i\, t^{sc}_{|i||j|}\, h^{q-1}_{kl}\right]$$

$$s^q_{mn} = s^q_{mn} + \tfrac{\Upsilon_{k+i}}{\varsigma_{|i|}\varsigma_{|j|}} \left[\ t^{cc}_{|i||j|}\, s^{q-1}_{kl} - \Upsilon_i \Upsilon_j\, t^{ss}_{|i||j|}\, r^{q-1}_{kl} - \Upsilon_j\, t^{cs}_{|i||j|}\, h^{q-1}_{kl} + \Upsilon_i\, t^{sc}_{|i||j|}\, g^{q-1}_{kl}\right]$$

$$\text{for } \begin{cases} i = -I, -(I-1), \dots, -1, 1, \dots, I \\ j = -I, -(I-1), \dots, -1, 1, \dots, J \\ k = 1, 2, 3, \dots, (q-1)I \\ l = 1, 2, 3, \dots, (q-1)J \end{cases} \quad \text{with } \begin{cases} m = |k+i| \\ n = |l+j| \\ \Upsilon_i = \text{sign}(i) \end{cases}$$

$$\left.\begin{aligned} c^{cc}_{mn} &= c^{cc}_{mn} + b_q g^q_{mn} \\ c^{ss}_{mn} &= c^{ss}_{mn} + b_q h^q_{mn} \\ c^{cs}_{mn} &= c^{cs}_{mn} + b_q r^q_{mn} \\ c^{sc}_{mn} &= c^{sc}_{mn} + b_q s^q_{mn} \end{aligned}\right\} \text{ for } \begin{cases} m = 1, 2, 3, \dots, qI \\ n = 1, 2, 3, \dots, qJ \end{cases}$$

Repeat Step #3 up to, and including, the case $q = Q$. Again, just two matrices for each of h_{kl}, $g_{kl}\, rkl$ and s_{kl} are needed. The algorithm is easy to program, though it may looks imposing.