Numerical procedures for computing constrained coordinate transforms

Kris Innanen

ABSTRACT

In other papers published in this report, general numerical procedures for determining $N\times N$ transformation matrices $\mathbf{T}$, which satisfy constraints involving the action of $\mathbf{T}$ on a given symmetric $N\times N$ matrix Φ, are assumed available. Here procedures of this kind are presented. Remarks on their development, and versions of the procedure that are efficient in that they do not duplicate any large scale calculation, are included. Although we use them in this report to adapt the Hessian matrix in inverse problems, we remark that, if used in statistical applications, the procedures in this paper can be regarded as general *whitening transform* algorithms.

PROBLEM STATEMENTS

We treat two related problems. First: given the symmetric $N\times N$ matrix

$$\Phi = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \cdots & \phi_{1,N} \\ \phi_{1,2} & \phi_{2,2} & \phi_{2,3} & \cdots & \phi_{2,N} \\ \phi_{1,3} & \phi_{2,3} & \phi_{3,3} & \cdots & \phi_{3,N} \\ & & \vdots & \ddots & \\ \phi_{1,N} & \phi_{2,N} & \phi_{3,N} & \cdots & \phi_{N,N} \end{bmatrix}, \tag{1}$$

find the $N\times N$ transformation matrix

$$\mathbf{T} = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} & \cdots & t_{1,N} \\ t^*_{2,1} & t_{2,2} & t_{2,3} & \cdots & t_{2,N} \\ t^*_{3,1} & t^*_{3,2} & t_{3,3} & \cdots & t_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \\ t^*_{N,1} & t^*_{N,2} & t^*_{N,3} & \cdots & t_{N,N} \end{bmatrix}, \tag{2}$$

such that

$$\mathbf{T}^T \Phi \mathbf{T} = \mathbf{I}, \tag{3}$$

where $\mathbf{I}$ is the $N\times N$ identity matrix, and where the $N(N-1)/2$ elements of $\mathbf{T}$ lying below the diagonal (marked *) have been pre-selected. The second, related, problem, is to repeat the original problem, but (1) add to the list of preselected elements of $\mathbf{T}$ the diagonals $t_{1,1}$, $t_{2,2}$, etc., and (2) remove from the list of constraints the requirement that the diagonals of Φ be equal. That is, solve for the $N(N-1)/2$ elements of $\mathbf{T}$ above the diagonal:

$$\mathbf{T} = \begin{bmatrix} t^*_{1,1} & t_{1,2} & t_{1,3} & \cdots & t_{1,N} \\ t^*_{2,1} & t^*_{2,2} & t_{2,3} & \cdots & t_{2,N} \\ t^*_{3,1} & t^*_{3,2} & t^*_{3,3} & \cdots & t_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \\ t^*_{N,1} & t^*_{N,2} & t^*_{N,3} & \cdots & t^*_{N,N} \end{bmatrix}, \tag{4}$$

subject to

$$\sum_{n,m} t_{m,i}\,\phi_{m,n}\,t_{n,j} = 0, \;\; i \neq j. \tag{5}$$

Transforms satisfying equation (3) when applied to covariance matrices in statistical applications are referred to as *whitening transforms*. As with the Hessian transformations we consider in this report, all whitening transform approaches used in the literature appear to be based on eigen-decompositions of Φ; more general transformation matrices of the type in equation (4) are not estimated, as far as the author is aware. So, what we present here can also be regarded as a new class of statistical whitening transforms.

Note on indexing

The vector / matrix indexing in this paper is different from that used in the companion papers. The convention we use here is designed to support algorithm/computer program development, and so it involves *[row,column]* indexing; elsewhere, where the development is theoretical, the algebra is more abstract, and essentially the opposite indexing style appears. The matrix $\mathbf{T}$ produced by the procedure in this paper maps element by element to the components of the tensor t^{μ}_{ν} used in the other papers as follows:

$$\begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} & \dots & t_{1,N} \\ t_{2,1} & t_{2,2} & t_{2,3} & \dots & t_{2,N} \\ t_{3,1} & t_{3,2} & t_{3,3} & \dots & t_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \\ t_{N,1} & t_{N,2} & t_{N,3} & \dots & t_{N,N} \end{bmatrix} \rightarrow \begin{bmatrix} t^1_1 & t^1_2 & t^1_3 & \dots & t^1_N \\ t^2_1 & t^2_2 & t^2_3 & \dots & t^3_N \\ t^3_1 & t^3_2 & t^3_3 & \dots & t^3_N \\ \vdots & \vdots & \vdots & \ddots & \\ t^N_1 & t^N_2 & t^N_3 & \dots & t^N_N \end{bmatrix}. \tag{6}$$

PROBLEM 1 PROCEDURE

The procedure involves filling in missing elements of $\mathbf{T}$ column-by-column. We start at the left, with the column vector

$$\begin{bmatrix} t_{1,1} \\ t^*_{2,1} \\ t^*_{3,1} \\ \vdots \\ t^*_{N,1} \end{bmatrix}. \tag{7}$$

Here there is only one unknown. It is the solution of the quadratic equation

$$\alpha\, t^2_{1,1} + \beta\, t_{1,1} + \gamma = 0, \tag{8}$$

where

$$\alpha = \mathbf{u}^T \Phi \mathbf{u}, \;\beta = 2\mathbf{u}^T \Phi \mathbf{v}, \text{ and } \gamma = \mathbf{v}^T \Phi \mathbf{v} - 1, \tag{9}$$

and where

$$\mathbf{u} = [1, 0, 0, ..., 0]^T, \text{ and } \mathbf{v} = [0, t^*_{2,1}, t^*_{3,1}, ..., t^*_{N,1}]^T. \tag{10}$$

Solving equation (8), we obtain the first column of $\mathbf{T}$ in full. The solution will then be complete if we can find a procedure for filling in the jth column, assuming columns 1 through $j-1$ are determined. We seek, in other words, to find the upper j elements of

$$\begin{bmatrix} t_{1,j} \\ t_{2,j} \\ \vdots \\ t_{j,j} \\ t^*_{j+1,j} \\ \vdots \\ t^*_{N,j} \end{bmatrix}. \tag{11}$$

Let the $j-1$ columns of $\mathbf{T}$ to the left of this vector, which are known, be collected up in $\mathbf{T}'$:

$$\mathbf{T}' = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} & \dots & t_{1,j-1} \\ t^*_{2,1} & t_{2,2} & t_{2,3} & \dots & t_{2,j-1} \\ t^*_{3,1} & t^*_{3,2} & t_{3,3} & \dots & t_{3,j-1} \\ \vdots & \vdots & \vdots & & \vdots \\ t^*_{N,1} & t^*_{N,2} & t^*_{N,3} & \dots & t^*_{N,j-1} \end{bmatrix}. \tag{12}$$

We then construct several useful intermediate vectors and matrices. The first of these is the $(j-1)\times N$ matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,j} & a_{1,j+1} & \dots & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,j} & a_{2,j+1} & \dots & a_{2,N} \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ a_{j-1,1} & a_{j-1,2} & \dots & a_{j-1,j} & a_{j-1,j+1} & \dots & a_{j-1,N} \end{bmatrix} = \mathbf{T}'^T\Phi. \tag{13}$$

The second is an alteration of $\mathbf{A}$. The rightmost $N-j$ columns of $\mathbf{A}$ are included, as a block, in a product with a vector containing the $\mathbf{T}$ entries below the diagonal at the current j. For all but the $j=N$ case, this produces $\mathbf{B}$:

$$\mathbf{B} = \begin{bmatrix} b_1 \\ \vdots \\ b_{j-1} \end{bmatrix} = \begin{bmatrix} a_{1,j+1} & \dots & a_{1,N} \\ a_{2,j+1} & \dots & a_{2,N} \\ \vdots & \dots & \vdots \\ a_{j-1,j+1} & \dots & a_{j-1,N} \end{bmatrix} \begin{bmatrix} t^*_{j+1,j} \\ \vdots \\ t^*_{N,j} \end{bmatrix}, \tag{14}$$

which then takes the place of that right-hand block in $\mathbf{A}$:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,j} & b_1 \\ a_{2,1} & a_{2,2} & \dots & a_{2,j} & b_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ a_{j-1,1} & a_{j-1,2} & \dots & a_{j-1,j} & b_{j-1} \end{bmatrix}. \tag{15}$$

For the $j=N$ case, this rightmost column is a zero-vector. The altered $\mathbf{A}$ is then subjected to a partial elimination/backsubstitution process, which stops when each row contains three nonzero elements. This produces a system of the form

$$\begin{bmatrix} c_{1,1} & 0 & \dots & 0 & 0 & c_{1,2} & c_{1,3} \\ 0 & c_{2,1} & \dots & 0 & 0 & c_{2,2} & c_{2,3} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & c_{j-2,1} & 0 & c_{j-2,2} & c_{j-2,3} \\ 0 & 0 & \dots & 0 & c_{j-1,1} & c_{j-1,2} & c_{j-1,3} \end{bmatrix}, \tag{16}$$

whose elements have been re-named to fit in a new $(j-1)\times 3$ matrix

$$\mathbf{C} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ \vdots & \vdots & \vdots \\ c_{j-1,1} & c_{j-1,2} & c_{j-1,3} \end{bmatrix}. \tag{17}$$

One simple (though numerically inefficient) way of accomplishing this is through a row-by-row procedure on $\mathbf{A}$. At the kth row, exchange the kth and $j-1$th columns, subject the new matrix to row operations until the left $(j-1)\times(j-1)$ square block is upper triangular, and store the bottom rightmost 3 elements as the kth row of $\mathbf{C}$.

Next, letting $f_k = -c_{k,2}/c_{k,1}$ and $g_k = -c_{k,3}/c_{k,1}$, we construct two N-length vectors:

$$\mathbf{u} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{j-1} \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{j-1} \\ 0 \\ t^*_{j+1,j} \\ t^*_{j+2,j} \\ \vdots \\ t^*_{N,j} \end{bmatrix}, \tag{18}$$

where in $\mathbf{v}$ we again use the preselected $\mathbf{T}$ elements found in the jth column below the diagonal. The $t_{j,j}$th element of the current column then satisfies a similar quadratic equation to that used for column 1:

$$\alpha\, t_{j,j}^2 + \beta\, t_{j,j} + \gamma = 0, \tag{19}$$

where the α, β and γ are the same bilinear forms used equation (10), but using the updated vectors in (18). Solving equation (19) fills in the centre element $t_{j,j}$ in equation (11). The remaining (upper) $j-1$ elements are related to $t_{j,j}$ linearly and are determined by

$$\begin{bmatrix} t_{1,j} \\ t_{2,j} \\ \vdots \\ t_{j-1,j} \end{bmatrix} = \begin{bmatrix} f_1 & g_1 \\ f_2 & g_2 \\ \vdots & \vdots \\ f_{j-1} & g_{j-1} \end{bmatrix} \begin{bmatrix} t_{j,j} \\ 1 \end{bmatrix}. \tag{20}$$

This completes the solution for the jth column; given the solution for the 1st column, and proceeding iteratively, the entire matrix $\mathbf{T}$ is thus determined.

PROBLEM 2 PROCEDURE

Problem 2, in which we pre-select the diagonals on $\mathbf{T}$ and solve for N fewer unknowns, can be solved with a small change to the above procedure. Column 1 is fully determined at the outset, and so the column 1 stage can be skipped. The procedure for column j is the same up to the construction of $\mathbf{C}$ in equation (17) and the derived quantities $f_k = -c_{k,2}/c_{k,1}$ and $g_k = -c_{k,3}/c_{k,1}$. We then skip directly to their use in determining all of the j elements above the diagonal:

$$\begin{bmatrix} t_{1,j} \\ t_{2,j} \\ \vdots \\ t_{j-1,j} \end{bmatrix} = \begin{bmatrix} f_1 & g_1 \\ f_2 & g_2 \\ \vdots & \vdots \\ f_{j-1} & g_{j-1} \end{bmatrix} \begin{bmatrix} t^*_{j,j} \\ 1 \end{bmatrix}. \tag{21}$$

REMARKS ON THE ORIGINS OF THE ALGORITHMS

The above procedures are deducible by direct inspection of the constraint equations. We will make enough remarks here that the procedure can be re-derived quickly if desired. The constraint equations in (3) are of the form

$$\phi_{1,1}t_{1,i}t_{1,i} + \phi_{1,2}t_{1,i}t_{2,i} + ... + \phi_{N,N}t_{N,i}t_{N,i} - 1 = 0, \tag{22}$$

for the diagonal elements of the transformed Φ, and

$$\phi_{1,1}t_{1,i}t_{1,j} + \phi_{1,2}t_{1,i}t_{2,j} + ... + \phi_{N,N}t_{N,i}t_{N,j} = 0, \tag{23}$$

for its off-diagonal elements. In the column 1 case, equation (22) applies, and only $t_{1,1}$ needs to be determined. Setting i=1, we observe that the sum contains terms quadratic in $t_{1,1}$, terms linear in $t_{1,1}$, and terms without $t_{1,1}$:

$$\left(\phi_{1,1}\right)t_{1,1}^2 + 2\left(\phi_{1,2}t^*_{2,1} + ... + \phi_{1,N}t^*_{N,1}\right)t_{1,1} + \left(\phi_{2,2}t^*_{2,1}t^*_{2,1} + ... + \phi_{N,N}t^*_{N,1}t^*_{N,1} - 1\right) = 0. \tag{24}$$

The terms in brackets $(\cdot)$ are, then, the α, β and γ in equation (8). Each are, by inspection, bilinear forms involving reduced portions of the full matrix Φ, which can be generated by projecting it onto the vectors $\mathbf{u}$ and $\mathbf{v}$ as set up in equations (10).

Consider column 2 next, focusing on two equations in (3) which relate to the elements $[t_{1,2}, t_{2,2}, t^*_{3,2}, ..., t^*_{N,2}]^T$ of this column. These are, first, equation (22) with $i = 2$:

$$\begin{aligned} &\phi_{1,1}t_{1,2}^2 + \phi_{1,2}t_{1,2}t_{2,2} + \phi_{1,3}t_{1,2}t^*_{3,2} + \phi_{1,4}t_{1,2}t^*_{4,2} + ... \\ &+ \phi_{2,1}t_{2,2}t_{1,2} + \phi_{2,2}t_{2,2}^2 + \phi_{2,3}t_{2,2}t^*_{3,2} + \phi_{2,4}t_{2,2}t^*_{4,2} + ... \\ &+ \phi_{3,1}t^*_{3,2}t_{1,2} + \phi_{3,2}t^*_{3,2}t_{2,2} + \phi_{3,3}{t^*_{3,2}}^2 + \phi_{3,4}t^*_{3,2}t^*_{4,2} + ... - 1 = 0, \end{aligned} \tag{25}$$

and second, equation (23) with $i = 1$, $j = 2$:

$$\begin{aligned} &\phi_{1,1}t_{1,1}t_{1,2} + \phi_{1,2}t_{1,1}t_{2,2} + \phi_{1,3}t_{1,1}t^*_{3,2} + \phi_{1,4}t_{1,1}t^*_{4,2} + ... \\ &+ \phi_{2,1}t^*_{2,1}t_{1,2} + \phi_{2,2}t^*_{2,1}t_{2,2} + \phi_{2,3}t^*_{2,1}t^*_{3,2} + \phi_{2,4}t^*_{2,1}t^*_{4,2} + ... \\ &+ \phi_{N,1}t^*_{N,1}t_{1,2} + \phi_{N,2}t^*_{N,1}t_{2,2} + \phi_{N,3}t^*_{N,1}t^*_{3,2} + \phi_{N,4}t^*_{N,1}t^*_{4,2} + ... = 0. \end{aligned} \tag{26}$$

Notice that, because column 1 is now determined, the two unknowns $t_{1,2}$ and $t_{2,2}$ appear linearly in equation (26), rather than quadratically, as they do in equation (25). In fact, equation (26) can be re-written

$$c_{1,1}\, t_{1,2} + c_{1,2}\, t_{2,2} + c_{1,3} = 0, \tag{27}$$

where

$$c_{1,1} = \phi_{1,1} t_{1,1}, \;\; c_{1,2} = \phi_{1,2} t_{1,1}, \;\; c_{1,3} = \phi_{1,3} t_{1,1} t^*_{3,2} + ..., \tag{28}$$

or, solving for $t_{1,2}$,

$$t_{1,2} = f_1\, t_{2,2} + g_1, \tag{29}$$

where

$$f_1 = -c_{1,2}/c_{1,1}, \;\; g_1 = -c_{1,3}/c_{1,1}. \tag{30}$$

Equation (29) allows us to return to equation (25), eliminate $t_{1,2}$, and leave behind a quadratic equation in only one unknown, $t_{2,2}$. By inspection, we can carry out the elimination and generate the quadratic equation by forming the vectors

$$\mathbf{u} = \begin{bmatrix} f_1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \text{ and } \mathbf{v} = \begin{bmatrix} g_1 \\ 0 \\ t^*_{3,2} \\ t^*_{4,2} \\ \vdots \\ t^*_{N,2} \end{bmatrix}, \tag{31}$$

whereupon equation (25) becomes

$$\left(\mathbf{u}^T \Phi \mathbf{u}\right) t^2_{2,2} + \left(2\mathbf{u}^T \Phi \mathbf{v}\right) t_{2,2} + \left(\mathbf{v}^T \Phi \mathbf{v} - 1\right) = 0. \tag{32}$$

Upon solving for $t_{2,2}$, we then recover $t_{1,2}$ via:

$$t_{1,2} = [f_1, g_1] \begin{bmatrix} t_{2,2} \\ 1 \end{bmatrix}, \tag{33}$$

completing the solution for the 2 elements of column 2. Subsequent columns are solved for similarly.

AN EFFICIENT PROCEDURE FOR PROBLEM 1

The procedures above are in a logically useful form, because their origins in terms of the constraint equations are relatively clear. However, actually computing transformation matrices with these procedures involves a large number of duplicate calculations as we go from column 1 to column N, and this is inefficient, especially for large N. Here we present alternative, though slightly more opaque, procedures for problems 1 and 2 with duplicate calculations removed. The descriptions are slightly repetitive, but they are self-contained. In this section we treat problem 1; in the next problem 2.

The targets are the diagonal and upper triangular components of the matrix

$$\mathbf{T} = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} & \dots & t_{1,N} \\ t^*_{2,1} & t_{2,2} & t_{2,3} & \dots & t_{2,N} \\ t^*_{3,1} & t^*_{3,2} & t_{3,3} & \dots & t_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \\ t^*_{N,1} & t^*_{N,2} & t^*_{N,3} & \dots & t_{N,N} \end{bmatrix}. \tag{34}$$

We will carry out the calculations for columns 1, 2 and 3 explicitly, then move to the general case $j \geq 4$ to complete the solution.

Column 1

The column 1 procedure is the same as in the original procedure. We determine $t_{1,1}$ by solving the quadratic equation

$$\alpha\, t_{1,1}^2 + \beta\, t_{1,1} + \gamma = 0, \tag{35}$$

where $\alpha = \mathbf{u}^T \Phi \mathbf{u}$, $\beta = 2\mathbf{u}^T \Phi \mathbf{v}$ and $\gamma = \mathbf{v}^T \Phi \mathbf{v} - 1$, and where $\mathbf{u} = [1, 0, 0, ..., 0]^T$ and $\mathbf{v} = [0, t^*_{2,1}, t^*_{3,1}, ..., t^*_{N,1}]^T$. Column 1 is then completely determined.

Column 2

We construct the $1 \times N$ matrix $\mathbf{A}$ with Φ and the most recently-determined column of $\mathbf{T}$:

$$\mathbf{A} = \begin{bmatrix} t_{1,1} & t^*_{2,1} & \dots & t^*_{N,1} \end{bmatrix} \Phi = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N} \end{bmatrix}. \tag{36}$$

Then we construct the 1×1 matrix $\mathbf{B}$ using the rightmost $N-2$ elements of $\mathbf{A}$ and the $(N-2)$-length segment of the current column of $\mathbf{T}$ containing pre-selected entries:

$$\mathbf{B} = \begin{bmatrix} a_{1,3} & a_{1,4} & \dots & a_{1,N} \end{bmatrix} \begin{bmatrix} t^*_{3,2} \\ \vdots \\ t^*_{N,2} \end{bmatrix} = \begin{bmatrix} b_1 \end{bmatrix}. \tag{37}$$

The matrix $\mathbf{C}$ is then constructed with the left side of $\mathbf{A}$ and with $\mathbf{B}$:

$$\mathbf{C} = \begin{bmatrix} a_{1,1} & a_{1,2} & b_1 \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \end{bmatrix}. \tag{38}$$

We then form two 1-length vectors from the three entries in $\mathbf{C}$:

$$\mathbf{F} = \begin{bmatrix} -c_{1,2}/c_{1,1} \end{bmatrix} = [\, f_1 \,], \;\; \mathbf{G} = \begin{bmatrix} -c_{1,3}/c_{1,1} \end{bmatrix} = [\, g_1 \,]. \tag{39}$$

These are then concatenated with 1s, 0s, and the pre-selected entries from the current column below the diagonal, to create new length N vectors $\mathbf{u}$ and $\mathbf{v}$:

$$\mathbf{u} = \begin{bmatrix} f_1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \;\; \mathbf{v} = \begin{bmatrix} g_1 \\ 0 \\ t^*_{3,2} \\ \vdots \\ t^*_{N,2} \end{bmatrix}. \tag{40}$$

This allows a new quadratic equation to be formed for the second diagonal entry:

$$\alpha\, t_{2,2}^2 + \beta\, t_{2,2} + \gamma = 0, \tag{41}$$

where as before $\alpha = \mathbf{u}^T\Phi\mathbf{u}$, $\beta = 2\mathbf{u}^T\Phi\mathbf{v}$ and $\gamma = \mathbf{v}^T\Phi\mathbf{v} - 1$. Finally, the column element above the diagonal is determined by

$$\left[\, t_{1,2} \,\right] = \left[\, f_1 \; g_1 \,\right] \left[\begin{array}{c} t_{2,2} \\ 1 \end{array} \right]. \tag{42}$$

Column 2 is then completely determined.

Column 3

We compute the N-length row vector

$$\left[\begin{array}{cccc} t_{1,2} & t_{2,2} & \dots & t^*_{N,2} \end{array} \right] \Phi = \left[\begin{array}{cccc} a_{2,1} & a_{2,2} & \dots & a_{2,N} \end{array} \right], \tag{43}$$

and append it to the bottom of the matrix $\mathbf{A}$ computed at the beginning of the previous step:

$$\mathbf{A} = \left[\begin{array}{cccc} a_{1,1} & a_{1,2} & \dots & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,N} \end{array} \right]. \tag{44}$$

We then compute the 2×1 matrix $\mathbf{B}$ using the rightmost $N-3$ columns of $\mathbf{A}$ and the pre-selected elements of the current column:

$$\mathbf{B} = \left[\begin{array}{cccc} a_{1,4} & a_{1,5} & \dots & a_{1,N} \\ a_{2,4} & a_{2,5} & \dots & a_{2,N} \end{array} \right] \left[\begin{array}{c} t^*_{4,3} \\ \vdots \\ t^*_{N,3} \end{array} \right] = \left[\begin{array}{c} b_1 \\ b_2 \end{array} \right]. \tag{45}$$

We will again construct a new matrix $\mathbf{C}$ using these new $\mathbf{A}$ and $\mathbf{B}$ quantities, but this time, we will do so not by fully building $\mathbf{C}$ but by adding to previously-determined entries. Specifically, we strip the rightmost column from the $\mathbf{C}$ determined in the column 2 step:

$$\left[\begin{array}{ccc} c_{1,1} & c_{1,2} & c_{1,3} \end{array} \right] \rightarrow \left[\begin{array}{cc} c_{1,1} & c_{1,2} \end{array} \right], \tag{46}$$

then add two new columns to the right and one new row to the bottom:

$$\left[\begin{array}{cc} c_{1,1} & c_{1,2} \end{array} \right] \rightarrow \left[\begin{array}{cccc} c_{1,1} & c_{1,2} & a_{1,3} & b_1 \\ a_{2,1} & a_{2,2} & a_{2,3} & b_2 \end{array} \right]. \tag{47}$$

This is our new $\mathbf{C}$ matrix, but we will label it with a prime for the moment:

$$\mathbf{C}' = \left[\begin{array}{cccc} c_{1,1} & c_{1,2} & c_{1,3} & b_1 \\ a_{2,1} & a_{2,2} & a_{2,3} & b_2 \end{array} \right] = \left[\begin{array}{cccc} c'_{1,1} & c'_{1,2} & c'_{1,3} & c'_{1,4} \\ c'_{2,1} & c'_{2,2} & c'_{2,3} & c'_{2,4} \end{array} \right]. \tag{48}$$

We consider this to be a 2×2 matrix with 2 additional columns appended to the right. We then complete the construction of the unprimed $\mathbf{C}$ by applying row operations to $\mathbf{C}'$ such that only the diagonals of the matrix and the two columns are non-zero. These can be contained in the 2×2 matrix

$$\mathbf{R} = \begin{bmatrix} 1 & w_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ w_1 & 1 \end{bmatrix}, \tag{49}$$

where the w_i are weights (e.g., $w_1 = -c'_{2,1}/c'_{1,1}$) chosen such that

$$\mathbf{C} = \mathbf{R}\mathbf{C}' = \begin{bmatrix} c_{1,1} & 0 & c_{1,3} & c_{1,4} \\ 0 & c_{2,2} & c_{2,3} & c_{2,4} \end{bmatrix}. \tag{50}$$

The elements of $\mathbf{C}$ are then used to create new $\mathbf{F}$ and $\mathbf{G}$ vectors

$$\mathbf{F} = \begin{bmatrix} -c_{1,3}/c_{1,1} \\ -c_{2,3}/c_{2,2} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \ \mathbf{G} = \begin{bmatrix} -c_{1,4}/c_{1,1} \\ -c_{2,4}/c_{2,2} \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \tag{51}$$

these in turn are included in new $\mathbf{u}$ and $\mathbf{v}$ vectors

$$\mathbf{u} = \begin{bmatrix} f_1 \\ f_2 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ \mathbf{v} = \begin{bmatrix} g_1 \\ g_2 \\ 0 \\ t^*_{4,3} \\ \vdots \\ t^*_{N,3} \end{bmatrix}, \tag{52}$$

the diagonal element $t_{3,3}$ is found as the solution of

$$\alpha\, t^2_{3,3} + \beta\, t_{3,3} + \gamma = 0, \tag{53}$$

where as before $\alpha = \mathbf{u}^T \Phi \mathbf{u}$, $\beta = 2\mathbf{u}^T \Phi \mathbf{v}$ and $\gamma = \mathbf{v}^T \Phi \mathbf{v} - 1$, and the column elements above the diagonal are determined by

$$\begin{bmatrix} t_{1,3} \\ t_{2,3} \end{bmatrix} = \begin{bmatrix} f_1 & g_1 \\ f_2 & g_2 \end{bmatrix} \begin{bmatrix} t_{3,3} \\ 1 \end{bmatrix}. \tag{54}$$

Column 3 is then completely determined.

Columns $j \geq 4$

The first three columns contain the full pattern, so we can now write down a general procedure for the jth column, assuming all columns to the left of j in $\mathbf{T}$ have been determined. An N-length vector is computed from the $j-1$ column of $\mathbf{T}$ and Φ, and this is appended to the bottom of the matrix $\mathbf{A}$ from the $j-1$ calculation. The new $\mathbf{A}$ thus has an extra row:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,j} & a_{1,j+1} & \dots & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,j} & a_{2,j+1} & \dots & a_{2,N} \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ a_{j-1,1} & a_{j-1,2} & \dots & a_{j-1,j} & a_{j-1,j+1} & \dots & a_{j-1,N} \end{bmatrix}, \tag{55}$$

where

$$\begin{bmatrix} a_{j-1,1} & \dots & a_{j-1,N} \end{bmatrix} = \begin{bmatrix} t_{j-1,1} & \dots & t_{j-1,N} \end{bmatrix} \Phi. \tag{56}$$

The rightmost columns ($j+1$ to N) of $\mathbf{A}$ and the pre-selected elements of the jth column of $\mathbf{T}$ are then used to build $\mathbf{B}$:

$$\mathbf{B} = \begin{bmatrix} a_{1,j+1} & \dots & a_{1,N} \\ a_{2,j+1} & \dots & a_{2,N} \\ \vdots & \dots & \vdots \\ a_{j-1,j+1} & \dots & a_{j-1,N} \end{bmatrix} \begin{bmatrix} t^*_{j+1,j} \\ \vdots \\ t^*_{N,j} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_{j-1} \end{bmatrix}. \tag{57}$$

We take this $\mathbf{B}$, and the jth column of $\mathbf{A}$, arrange them into a matrix, cut off the bottom row, and pre-multiply them by the $(j-2)\times(j-2)$ matrix $\mathbf{R}$ that contains all previous row and column operations in it:

$$\begin{bmatrix} a'_{1,j} & b'_1 \\ a'_{2,j} & b'_2 \\ \vdots & \vdots \\ a'_{j-2,j} & b'_{j-2} \end{bmatrix} = \mathbf{R} \begin{bmatrix} a_{1,j} & b_1 \\ a_{2,j} & b_2 \\ \vdots & \vdots \\ a_{j-2,j} & b_{j-2} \end{bmatrix}. \tag{58}$$

This produces a $(j-1)\times 2$ matrix. We then take the matrix $\mathbf{C}$ constructed during the previous column calculation, strip off its rightmost column, replace it with the result above, and append to the bottom part of the new row of $\mathbf{A}$ and the bottom element of $\mathbf{B}$:

$$\mathbf{C}' = \begin{bmatrix} c_{1,1} & 0 & \dots & 0 & c_{1,j-1} & a'_{1,j} & b'_1 \\ 0 & c_{2,2} & \dots & 0 & c_{2,j-1} & a'_{2,j} & b'_2 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & c_{j-2,j-2} & c_{j-2,j-1} & a'_{j-2,j} & b'_{j-2} \\ a_{j-1,1} & a_{j-1,2} & \dots & a_{j-1,j-2} & a_{j-1,j-1} & a_{j-1,j} & b_{j-1} \end{bmatrix}. \tag{59}$$

We then append a zero row and a zero column onto the bottom and right of $\mathbf{R}$, and put a one on the diagonal, such that it grows to size $(j-1)\times(j-1)$. We then carry out as set of row operations on $\mathbf{C}'$, storing them by premultiplying them into the newly enlarged $\mathbf{R}$. First, we use the new bottom row to eliminate all of the other entries in the third column from the right; second, and then we use the upper rows to eliminate all entries in the new bottom row from 1 to $j-2$. The result is the new $\mathbf{C}$ matrix:

$$\mathbf{C} = \begin{bmatrix} c_{1,1} & 0 & \dots & 0 & 0 & c_{1,j} & c_{1,j+1} \\ 0 & c_{2,2} & \dots & 0 & 0 & c_{2,j} & c_{2,j+1} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & c_{j-1,j-1} & c_{j-1,j} & c_{j-1,j+1} \end{bmatrix}. \tag{60}$$

After this the process is identical to the original procedure: letting $f_k = -c_{k,2}/c_{k,1}$ and $g_k = -c_{k,3}/c_{k,1}$, we construct two N-length vectors:

$$\mathbf{u} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{j-1} \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \text{ and } \mathbf{v} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{j-1} \\ 0 \\ t^*_{j+1,j} \\ t^*_{j+2,j} \\ \vdots \\ t^*_{N,j} \end{bmatrix}. \tag{61}$$

The $t_{j,j}$th element of the current column satisfies

$$\alpha\, t_{j,j}^2 + \beta\, t_{j,j} + \gamma = 0, \tag{62}$$

where $\alpha = \mathbf{u}^T \Phi \mathbf{u}$, $\beta = 2\mathbf{u}^T \Phi \mathbf{v}$ and $\gamma = \mathbf{v}^T \Phi \mathbf{v} - 1$. Solving equation (19) fills in the diagonal element $t_{j,j}$, and the upper $j-1$ elements are determined by

$$\begin{bmatrix} t_{1,j} \\ t_{2,j} \\ \vdots \\ t_{j-1,j} \end{bmatrix} = \begin{bmatrix} f_1 & g_1 \\ f_2 & g_2 \\ \vdots & \vdots \\ f_{j-1} & g_{j-1} \end{bmatrix} \begin{bmatrix} t_{j,j} \\ 1 \end{bmatrix}. \tag{63}$$

This completes the solution for the jth column; given the solution for the 1st column, and proceeding iteratively, the entire matrix $\mathbf{T}$ is thus determined.

AN EFFICIENT PROCEDURE FOR PROBLEM 2

We begin again with

$$\mathbf{T} = \begin{bmatrix} t^*_{1,1} & t_{1,2} & t_{1,3} & \dots & t_{1,N} \\ t^*_{2,1} & t^*_{2,2} & t_{2,3} & \dots & t_{2,N} \\ t^*_{3,1} & t^*_{3,2} & t^*_{3,3} & \dots & t_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \\ t^*_{N,1} & t^*_{N,2} & t^*_{N,3} & \dots & t^*_{N,N} \end{bmatrix}, \tag{64}$$

and proceed to fill in the elements above the diagonal. We will carry out a loop over columns, beginning at column 3; column 1 is already full, and we will compute column 2 outside of the loop to establish our basic matrices.

Column 2

We let the 1st column of $\mathbf{T}$ be transposed to a row vector $\mathbf{t}_1 = [t^*_{1,1}, t^*_{2,1}, ..., t^*_{N,1}]$. Let $\mathbf{A}$ be the row vector formed from the product of this row with the $N \times N$ matrix Φ:

$$\mathbf{A} = \begin{bmatrix} t^*_{1,1} & t^*_{2,1} & \dots & t^*_{N,1} \end{bmatrix} \Phi = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N} \end{bmatrix}. \tag{65}$$

Next we let $\mathbf{B}$ be the length-1 column vector formed from the product of the rightmost $N-1$ elements of $\mathbf{A}$ with the $(N-1)$-length segment of the current column of $\mathbf{T}$ containing pre-selected entries:

$$\mathbf{B} = \left[\begin{array}{cccc} a_{1,2} & a_{1,3} & \dots & a_{1,N} \end{array} \right] \left[\begin{array}{c} t^*_{2,2} \\ \vdots \\ t^*_{N,2} \end{array} \right] = \left[\begin{array}{c} b_1 \end{array} \right]. \tag{66}$$

Then $\mathbf{C}$ be formed by appending $\mathbf{B}$ to the right of the first entry of $\mathbf{A}$, and suppressing all other elements of that row:

$$\mathbf{C} = \left[\begin{array}{cc} a_{1,1} & b_1 \end{array} \right] = \left[\begin{array}{cc} c_{1,1} & c_{1,2} \end{array} \right]. \tag{67}$$

The single unknown upper element of the second column of $\mathbf{T}$ is now determined as

$$t_{1,2} = -c_{1,2}/c_{1,1}. \tag{68}$$

This completes the solution for column 2 of $\mathbf{T}$.

Columns $j \geq 3$

With column 2 in hand, we can now treat the problem of computing the $j-1$ unknown elements of column j of $\mathbf{T}$. Append a new row to the growing matrix $\mathbf{A}$:

$$\mathbf{A} = \left[\begin{array}{c} \mathbf{A} \\ \mathbf{t}_{j-1}\Phi \end{array} \right] = \left[\begin{array}{cccc} a_{1,1} & a_{1,2} & \dots & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,N} \\ & \vdots & \vdots & \\ a_{j-1,1} & a_{j-1,2} & \dots & a_{j-1,N} \end{array} \right]. \tag{69}$$

Next, compute the product of the rightmost elements of $\mathbf{A}$, starting at j, with the known elements of the current column of $\mathbf{T}$:

$$\mathbf{B} = \left[\begin{array}{cccc} a_{1,j} & a_{1,j+1} & \dots & a_{1,N} \\ a_{2,j} & a_{2,j+1} & \dots & a_{2,N} \\ & \vdots & \vdots & \\ a_{j-1,j} & a_{j-1,j+1} & \dots & a_{j-1,N} \end{array} \right] \left[\begin{array}{c} t^*_{j,j} \\ \vdots \\ t^*_{N,j} \end{array} \right] = \left[\begin{array}{c} b_1 \\ \vdots \\ b_{j-1} \end{array} \right]. \tag{70}$$

We make use of the matrix $\mathbf{C}$ that was produced during the calculations of the previous column, labelling them with a double prime to keep them distinct.

$$\mathbf{C}'' = \left[\begin{array}{cccccc} c''_{1,1} & 0 & 0\dots & 0 & 0 & c''_{1,j-1} \\ 0 & c''_{2,2} & 0\dots & 0 & 0 & c''_{2,j-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0\dots & 0 & c''_{j-2,j-2} & c''_{j-2,j-1} \end{array} \right], \tag{71}$$

i.e., with two nonzero elements per row. Following the problem 1 approach, to create the matrix $\mathbf{C}$ for the current column calculation, we drop the right column, and augment the result with information newly calculated in $\mathbf{A}$ and $\mathbf{B}$. The new matrix has two new columns on the right, and one new row on the bottom:

$$\mathbf{C}'' = \begin{bmatrix} c''_{1,1} & 0 & 0\dots & 0 & a_{1,j-1} & b_1 \\ 0 & c''_{2,2} & 0\dots & 0 & a_{2,j-1} & b_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0\dots & c''_{j-2,j-2} & a_{j-2,j-1} & b_{j-2} \\ a_{j-1,1} & a_{j-1,2} & a_{j-1,3} & \dots & a_{j-1,j-1} & b_{j-1} \end{bmatrix}. \tag{72}$$

However, the rows of the old $\mathbf{C}'''$ have previously undergone a set of $M = (J-2)\times(J-3)$ elementary row operations, details of which we assume have been stored as in problem 1 in the $(j-2)\times(j-2)$ matrix $\mathbf{R}$. We operate on the top $j-2$ rows of the right two columns with $\mathbf{R}$, indicating this with primes:

$$\begin{bmatrix} c''_{1,1} & 0 & 0\dots & 0 & a'_{1,j-1} & b'_1 \\ 0 & c''_{2,2} & 0\dots & 0 & a'_{2,j-1} & b'_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0\dots & c''_{j-2,j-2} & a'_{j-2,j-1} & b'_{j-2} \\ a_{j-1,1} & a_{j-1,2} & a_{j-1,3} & \dots & a_{j-1,j-1} & b_{j-1} \end{bmatrix}. \tag{73}$$

Following problem 1, we grow $\mathbf{R}$ to size $(j-1)\times(j-1)$, eliminate the left $j-2$ entries in the bottom row, and the top $j-2$ entries in the column second from the right, and store all of these row operations in $\mathbf{R}$. This leaves

$$\mathbf{C} = \begin{bmatrix} c_{1,1} & 0 & 0\dots & 0 & 0 & c_{1,j} \\ 0 & c_{2,2} & 0\dots & 0 & 0 & c_{2,j} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & c_{j-1,j-1} & c_{j-1,j} \end{bmatrix}. \tag{74}$$

Each row contains 2 nonzero elements, and these are used in combination to compute the $j-1$ unknown elements above the diagonal in $\mathbf{T}$:

$$\begin{bmatrix} t_{1,j} \\ t_{2,j} \\ \vdots \\ t_{j-1,j} \end{bmatrix} = \begin{bmatrix} -c_{1,j}/c_{1,1} \\ -c_{2,j}/c_{2,2} \\ \vdots \\ -c_{j-1,j}/c_{j-1,j-1} \end{bmatrix}. \tag{75}$$

This completes the solution for the jth column.