Q1 We can propose a bordered algorithm for computing the ~~factorization~~ Cholesky factorization of a SPD matrix A inspired by the bordered LU factorization algorithm as follows:-

A blocked algorithm can be derived as follows:-

We can Partition

$$A = \left(\begin{array}{c|c} A_{00} & * \\ \hline A_{10} & A_{11} \end{array}\right) \text{ and } L = \left(\begin{array}{c|c} L_{00} & 0 \\ \hline L_{10} & L_{11} \end{array}\right)$$

We can subsitute these partitioned matrices into $A = LL^T$ we find that

$$\left(\begin{array}{c|c} A_{00} & * \\ \hline A_{10} & A_{11} \end{array}\right) = \left(\begin{array}{c|c} L_{00} & 0 \\ \hline L_{10} & L_{11} \end{array}\right) \left(\begin{array}{c|c} L_{00} & 0 \\ \hline L_{10} & L_{11} \end{array}\right)^T$$

$$= \left(\begin{array}{c|c} L_{00} L_{00}^T & * \\ \hline L_{10} L_{00}^T & L_{10} L_{10}^T + L_{11} L_{11}^T \end{array}\right)$$

from which we can conclude that

$$\begin{array}{c|c} L_{00} = \text{CHOL}(A)_{00} & \star \\ \hline L_{10} = A_{10} L_{00}^{-T} & L_{11} = \text{CHOL}(A_{11} - L_{10} L_{10})^T \end{array}$$

The above equalities motivate the following algorithm

1. Partition $A \rightarrow \left(\begin{array}{c|c} A_{00} & \star \\ \hline A_{10} & A_{11} \end{array} \right)$

2. We can assume that $A_{00} := L_{00} = \text{CHOL}(A_{00})$ has been computed by previous iterations of the loop-based algorithm.

3. We can overwrite $A_{10} := L_{10} = A_{10} L_{00}^{-T}$

4. Overwrite $A_{11} := A_{11} - L_{10} L_{10}^T$

5. Also, we can overwrite ~~A_{11}~~

$A_{11} := L_{11} = \text{CHOL}(A_{11})$

Hence, the above step justifies the blocked variant 1 which is as follows:-

Algorithm: $A := \text{CHOL\_BLK}(A)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & * \\ \hline A_{BL} & A_{BR} \end{array}\right)$

where A_{TL} is 0×0

while $m(A_{TL}) < m(A)$ do

Determine block size b

Repartition

$$\left(\begin{array}{c|c} A_{TL} & * \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & * & * \\ \hline A_{10} & A_{11} & * \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

where A_{11} is $b \times b$

Variant 1:- (Bordered Algorithm)

$A_{10} := A_{10}\,\text{TRIL}(A_{00})^{-T}$

$A_{11} := A_{11} - A_{10}A_{10}^{T}$

$A_{11} := \text{CHOL}(A_{11})$

b) We can prove the theorem by using bordered cholesky factorization algorithm for a matrix A that is SPD.

We can first write the bordered algorithm

Consider the loop invariant:

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right)$$

$$\wedge \; L_{TL} U_{TL} = \hat{A}_{TL}$$

The above equation shows that the leading principal submatrix A_{TL} has been overwritten with its LU factorization and the remainder of the matrix has not yet been touched.

At the top of the loop, after repartitioning, A then contains

$$\left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22}\end{array}\right) =$$

$$\left(\begin{array}{c|cc} L\backslash U_{00} & \hat{a}_{01} & \hat{A}_{02} \\ \hline \hat{a}_{10}^T & \hat{\alpha}_{11} & \hat{a}_{12}^T \\ \hat{A}_{20} & \hat{a}_{21} & \hat{A}_{22}\end{array}\right) \wedge L_{00}U_{00} = \hat{A}_{00}$$

while after updating A, it must contain ...

$$\left(\begin{array}{cc|c} A_{00} & a_{01} & A_{02} \\ a_{10}^T & \alpha_{11} & a_{12}^T \\ A_{20} & a_{21} & A_{22}\end{array}\right) = \left(\begin{array}{cc|c} L\backslash U_{00} & u_{01} & \hat{A}_{02} \\ l_{10}^T & u_{11} & \hat{a}_{12}^T \\ \hat{A}_{20} & \hat{a}_{21} & \hat{A}_{22}\end{array}\right)$$

$$\wedge \underbrace{\begin{pmatrix} L_{00} & 0 \\ l_{10}^T & 1 \end{pmatrix}\begin{pmatrix} U_{00} & u_{01} \\ 0 & u_{11} \end{pmatrix} = \begin{pmatrix} \hat{A}_{00} & \hat{a}_{01} \\ \hat{a}_{10}^T & \hat{\alpha}_{11} \end{pmatrix}}$$

$$L_{00}U_{00} = \hat{A}_{00} \qquad L_{00}u_{01} = \hat{a}_{01}$$

$$l_{10}^T U_{00} = \hat{a}_{10}^T \qquad l_{10}^T u_{01} + u_{11} = \hat{\alpha}_{11}$$

.. for the loop invariant to again hold after the iteration

With this, we can compute the desired parts of L and U:

From above, we can solve $L_{00}U_{01} = a_{01}$, overwriting a_{01} with the result. Also, we can notice that $a_{01} = \hat{a}_{01}$ before this update.

We can also solve $l_{10}^T U_{00} = a_{10}^T$ (or $U_{00}^T (l_{10}^T)^T = (a_{10}^T)^T$ for l_{10}^T) where, we can overwrite a_{10}^T with the result. We can also notice that $a_{10}^T = \hat{a}_{10}^T$ before this update.

We can then make a update $\alpha_{11} := \upsilon_{11} = \alpha_{11} - l_{10}^T \upsilon_{01}$. We can also notice that by this computation, $a_{10}^T = l_{10}^T$ and $a_{01} = \upsilon_{01}$

→ Hence the resulting algorithm is given below :-

$A = LU\text{-}var1(A)$

$$A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

A_{TL} is 0×0

while $n(A_{TL}) < n(A)$

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|cc} A_{00} & a_{01} & A_{02} \\ a^T_{10} & \alpha_{11} & a^T_{12} \\ A_{20} & a_{21} & A_{22} \end{array}\right)$$

Solve $L_{00} U_{01} = a_{01}$ overwriting a_{01} with the result

Solve $l^T_{10} U_{00} = a^T_{10}$ overwriting a^T_{10} with the result

$\alpha_{11} := \upsilon_{11} = \alpha_{11} - a^T_{10} a_{01}$

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{cc|c} A_{00} & a_{01} & A_{02} \\ a^T_{10} & \alpha_{11} & a^T_{12} \\ A_{20} & a_{21} & A_{22} \end{array}\right)$$

end while

OR

We can also prove the theorem $A = LL^T$ by bordered cholesky factorization algorithm in following ways:-

We can introduce the bordering method for the cholesky decomposition with the following definitions

$$A_{k+1} = \begin{bmatrix} A_k & a_{k+1} \\ a_{k+1}^T & \alpha_{k+1} \end{bmatrix} = \begin{bmatrix} L_k & \\ v_{k+1}^T & d_{k+1} \end{bmatrix}$$

$$\begin{bmatrix} L_k^T & v_{k+1} \\ & d_{k+1} \end{bmatrix} = L_{k+1} L_{k+1}$$

where :

$$A_k = L_k L_k^T$$
$$v_{k+1} = L_k^{-1} a_{k+1}$$
$$d_{k+1} = \sqrt{\alpha_{k+1} - v_{k+1}^T v_{k+1}}$$

From the above, we can see

that the lower triangular matrix L_K is the cholesky decomposition of A_K, recursively for $K = 1, 2, \ldots N$, where A_1 and L_1 are the scalars α_1 and $d_1 = \alpha_1^{1/2}$. It can also be seen that the lower triangular matrix L_{K+1} is cobbled together from L_K, V_{K+1} and d_{K+1}. In additions the series of matrices $A_1, A_2, \ldots A_N$ are the leading sub-matrices of $A_N = M$, an $N \times N$ symmetric and positive definite matrix with the cholesky decomposition $L_N^T L_N = M$.

Hence, the bordered algorithm method to compute the cholesky decomposition of the matrix M is given below:-

1. We can set $A_N = M$, where we can define all the arrays, A_K, a_K, α_K, $N \geq K \geq 1$, and $A_1 = \alpha_1$ as data entries.

2. We can then evaluate $d_1 = \alpha_1^{1/2}$.

3. For $k = 1, 2, \ldots N-1$, perform the following calculations

3a) ~~Solve~~ we can solve v_{k+1} in the lower triangular system, where $L_k v_{k+1} = a_{k+1}$ by forward substitution.

3b) We can also evaluate the vector product, $\varepsilon_1 = v_{k+1}^T v_{k+1}$, and then we can evaluate $d_{k+1} = (\alpha_{k+1} - \varepsilon_1)^{1/2}$.

Hence, by the above algorithm the matrix M can be half stored, and because its entries are only used once in above method, we can overwrite M with L_N with the bordered algorithm.