Exercise 5.2. Derive an algorithm that, given a symmetric indefinite tridiagonal matrix A, computes $A = UDU^T$. Overwrite only the upper triangular part of A.

Solution :-

Partition $A \rightarrow \left[\begin{array}{c|c|c} A_{FF} & \alpha_{mF} e_L & 0 \\ \hline * & \alpha_{mm} & \alpha_{Lm} e_F^T \\ \hline * & * & A_{LL} \end{array}\right]$

where A_{LL} is 0×0

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

Repartition

$$\left[\begin{array}{c|c|c} A_{FF} & \alpha_{mF} e_L & 0 \\ \hline * & \alpha_{mm} & \alpha_{Lm} e_F^T \\ \hline * & * & A_{LL} \end{array}\right] \rightarrow \left[\begin{array}{c|c|c|c} A_{00} & \alpha_{01} e_L & 0 & 0 \\ \hline * & \alpha_{11} & \alpha_{12} & 0 \\ \hline * & * & \alpha_{22} & \alpha_{23} e_F^T \\ \hline * & * & * & A_{33} \end{array}\right]$$

where α_{11} is scalar

$\alpha_{01} := \alpha_{01} / \alpha_{11}$

$\alpha_{11} := \alpha_{11} - \alpha_{11} \alpha_{01}^2$

Continue with

$$\left[\begin{array}{c|c|c} A_{FF} & \alpha_{mF} e_L & 0 \\ \hline * & \alpha_{mm} & \alpha_{Lm} e_F^T \\ \hline * & * & A_{LL} \end{array}\right] \leftarrow \left[\begin{array}{cc|c|c} A_{00} & \alpha_{01} e_L & 0 & 0 \\ * & \alpha_{11} & \alpha_{12} & 0 \\ \hline * & * & \alpha_{22} & \alpha_{23} e_F^T \\ * & * & * & A_{33} \end{array}\right]$$

endwhile

To come up with algorithm for $A = UDU^T$ for a symmetric indefinite tridiagonal matrix, we need to explain the algorithmic steps needed to compute $A = UDU^T$ for a symmetric indefinite matrix to derive key insights.

Computing $A = UDU^T$ for symmetric indefinite matrix A

Partition

$$A \rightarrow \left[\begin{array}{c|c} A_{00} & a_{01} \\ \hline a_{01}^T & \alpha_{11} \end{array}\right] \quad U \rightarrow \left[\begin{array}{c|c} U_{00} & u_{01} \\ \hline 0 & 1 \end{array}\right] \quad D \rightarrow \left[\begin{array}{c|c} D_{00} & 0 \\ \hline 0 & \delta_1 \end{array}\right]$$

$$\begin{bmatrix} A_{00} & a_{01} \\ a_{01}^T & \alpha_{11} \end{bmatrix} = \begin{bmatrix} U_{00} & u_{01} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} D_{00} & 0 \\ 0 & \delta_1 \end{bmatrix} \begin{bmatrix} U_{00} & u_{01} \\ 0 & 1 \end{bmatrix}^T$$

$$= \left[\begin{array}{c|c} U_{00}D_{00} & u_{01}\delta_1 \\ \hline 0 & \delta_1 \end{array}\right] \begin{bmatrix} U_{00}^T & 0 \\ u_{01}^T & 1 \end{bmatrix}$$

$$= \left[\begin{array}{c|c} U_{00}D_{00}U_{00}^T + \delta_1 u_{01}u_{01}^T & u_{01}\delta_1 \\ \hline * & \delta_1 \end{array}\right]$$

This suggests the following algorithm for overwriting the strictly upper triangular part of A with strictly upper triangular part of U and the diagonal of A with D.

- Partition $A \to \begin{bmatrix} A_{00} & a_{01} \\ a_{01}^T & \alpha_{11} \end{bmatrix}$

- $\alpha_{11} := \delta_1 = \alpha_{11}$ (no-op)

- Compute $u_{01} = a_{01} / \alpha_{11}$

- Update $U_{00} D_{00} U_{00}^T + \delta_1 u_{01} u_{01}^T := A_{00}$

$$U_{00} D_{00} U_{00}^T := A_{00} - \delta_1 u_{01} u_{01}^T$$

new A_{00} is old A_{00} minus the vector mult because new term $U_{00} D_{00} U_{00}^T$ is updated A_{00}

$$A_{00} := A_{00} - \delta_1 u_{01} \frac{a_{01}^T}{\alpha_{11}}$$

(Since we want to update only upper triangular part)

$$A_{00} := A_{00} - \alpha_{11} u_{01} \frac{a_{01}^T}{\alpha_{11}}$$

$$A_{00} := A_{00} - u_{01} a_{01}^T$$

- Update $a_{01} := u_{01}$

- Continue with computing $A_{00} \to U_{00} D_{00} U_{00}^T$
  Algorithm will complete as long as $\delta_{11} \neq 0$.

The key insight is recognizing that $a_{01} = \begin{bmatrix} 0 \\ \hline \alpha_{01} \end{bmatrix}$

So update $a_{01} := a_{01} / \alpha_{11}$ becomes

$$\begin{bmatrix} 0 \\ \hline \alpha_{01} \end{bmatrix} := \begin{bmatrix} 0 \\ \hline \alpha_{01} \end{bmatrix} / \alpha_{11} = \begin{bmatrix} 0 \\ \hline \alpha_{01} / \alpha_{11} \end{bmatrix}$$

Then update to $A_{00} := A_{00} - \alpha_{11} a_{01} a_{01}^T$ becomes

$$\left[\begin{array}{c|c} A_{00} & \alpha_{01} e_L \\ \hline * & \alpha_{11} \end{array}\right] := \left(\begin{array}{c|c} A_{00} & \alpha_{01} e_L \\ \hline * & \alpha_{11} \end{array}\right) - \alpha_{11} \left[\begin{array}{c} 0 \\ \hline \alpha_{01} \end{array}\right] \left[\begin{array}{c} 0 \\ \hline \alpha_{01} \end{array}\right]^T$$

$$= \left[\begin{array}{c|c} A_{00} & \alpha_{01} e_L \\ \hline * & \alpha_{11} \end{array}\right] - \alpha_{11} \left[\begin{array}{c} 0 \\ \alpha_{01}/\alpha_{11} \end{array}\right] \left[\begin{array}{cc} 0 & \alpha_{01} \end{array}\right]$$

$$= \left[\begin{array}{c|c} A_{00} & \alpha_{01} e_L \\ \hline * & \alpha_{11} \end{array}\right] - \alpha_{11} \left[\begin{array}{cc} 0 & 0 \\ 0 & \alpha_{01}^2 \end{array}\right]$$

$$\left[\begin{array}{c|c} A_{00} & \alpha_{01} e_L \\ \hline * & \alpha_{11} \end{array}\right] := \left[\begin{array}{c|c} A_{00} & \alpha_{01} e_L \\ \hline * & \alpha_{11} - \alpha_{11} \alpha_{01}^2 \end{array}\right]$$

These steps show the updates needed to proceed to the next steps of the algorithm.

$$\alpha_{01} := \alpha_{01}/\alpha_{11}$$

$$\alpha_{11} := \alpha_{11} - \alpha_{11} \alpha_{01}^2$$