

**Example:**  $C := AB^T + BA^T + C$  — Team: —

## 1.1 Operation

Consider the operation

$$C := AB^T + BA^T + C$$

where  $C$  is a  $m \times m$  lower triangular matrix and  $A$  and  $B$  is a  $m \times m$  matrix. This is what we call a symmetric rank-2k update with the Lower triangular matrix on the Right. We will refer to this operation as SYR2K\_LN where the LN stands for lower triangular no-transpose.

## 1.2 Precondition and postcondition

In the precondition

$$C = \hat{C}$$

$\hat{C}$  denotes the original contents of  $C$ . This allows us to express the state upon completion, the postcondition, as

$$C := AB^T + BA^T + \hat{C}.$$

It is implicitly assumed that  $C$  is nonunit lower triangular.

## 1.3 Partitioned Matrix Expressions and loop invariants

There are two PME's for this operation.

### 1.3.1 PME 1

To derive the first PME, partition

$$C \rightarrow \left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right), \quad A \rightarrow \left( \begin{array}{c} A_T \\ \hline A_B \end{array} \right), \quad \text{and} \quad B \rightarrow \left( \begin{array}{c|c} B_T^T & B_B^T \end{array} \right)$$

Substituting these into the postcondition yields

$$\left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right) = \left( \begin{array}{c} A_T \\ \hline A_B \end{array} \right) \left( \begin{array}{c|c} B_T^T & B_B^T \end{array} \right) + \left( \begin{array}{c} B_T \\ \hline B_B \end{array} \right) \left( \begin{array}{c|c} A_T^T & A_B^T \end{array} \right) + \left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right)$$

or, equivalently,

$$\left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right) = \left( \begin{array}{c|c} A_TB_T^T + B_TA_T^T + C_{TL} & * \\ \hline A_BB_T^T + B_BA_B^T + C_{BL} & A_BB_B^T + B_BA_B^T + C_{BR} \end{array} \right)$$

which we refer to as the first PME for this operations.

From this, we can choose five loop invariants:

**Invariant 1:**  $\left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right) = \left( \begin{array}{c|c} A_TB_T^T + B_TA_T^T + \hat{C}_{TL} & * \\ \hline \hat{C}_{BL} & \hat{C}_{BR} \end{array} \right).$

(The top left part has been left alone and the bottom parts have been completely computed).

**Invariant 2:**  $\left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right) = \left( \begin{array}{c|c} \hat{C}_{TL} & * \\ \hline \hat{C}_{BL} & A_BB_B^T + B_BA_B^T + \hat{C}_{BR} \end{array} \right).$

(The bottom right part has been left alone and the left parts have been completely computed).

**Invariant 3:**  $\left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right) = \left( \begin{array}{c|c} A_TB_T^T + B_TA_T^T + \hat{C}_{TL} & * \\ \hline A_BB_T^T + \hat{C}_{BL} & \hat{C}_{BR} \end{array} \right).$

(The top left part has been left alone, the bottom left part has been partially computed, and the bottom right part has been completely computed).

**Invariant 4:**  $\left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right) = \left( \begin{array}{c|c} A_TB_T^T + B_TA_T^T + \hat{C}_{TL} & * \\ \hline B_BA_B^T + \hat{C}_{BL} & \hat{C}_{BR} \end{array} \right).$

(The top left part has been left alone, the bottom right part has been partially computed, and the bottom left part has been completely computed).

**Invariant 5:**  $\left( \begin{array}{c|c} C_{TL} & * \\ \hline C_{BL} & C_{BR} \end{array} \right) = \left( \begin{array}{c|c} \hat{C}_{TL} & * \\ \hline A_BB_B^T + \hat{C}_{BL} & A_BB_B^T + B_BA_B^T + \hat{C}_{BR} \end{array} \right).$

(The bottom right part has been left alone, the bottom left part has been partially computed, and the top left part has been completely computed).

### 1.3.2 PME 2

To derive the second PME, partition

$$A \rightarrow \left( \begin{array}{c|c} A_L & A_R \end{array} \right), \quad \text{and} \quad B \rightarrow \left( \begin{array}{c|c} B_L & B_R \end{array} \right)$$

and do not partition  $C$ . Substituting these into the postcondition yields

$$C = \left( A_L \mid A_R \right) \begin{pmatrix} B_L^T \\ B_R^T \end{pmatrix} + \left( B_L \mid B_R \right) \begin{pmatrix} A_L^T \\ A_R^T \end{pmatrix} + C$$

or, equivalently,

$$C = A_L B_L^T + A_R B_R^T + B_L A_L^T + B_R A_R^T + \hat{C}$$

which we refer to as the second PME.

From this, we can choose one more loop invariant:

**Invariant 6:**  $C = A_L B_L^T + B_L A_L^T + \hat{C}$ .

(The right part has been completely finished).

### 1.3.3 Notes

How do I decide to partition the matrices in the postcondition?

- Pick a matrix (operand), any matrix.
- If that matrix has
  - a triangular structure (in storage), then you want to either partition is into four quadrants, or not at all. Symmetric matrices and triangular matrices have a triangular structure (in storage).
  - no particular structure, then you partition it vertically (left-right), horizontally (top-bottom), or not at all.
- Next, partition the other matrices similarly, but conformally (meaning the resulting multiplications with the parts are legal).

Take our problem here:  $B := LB$ . Start by partitioning  $L$  in to quadrants:

$$B = \left( \begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \hat{B}.$$

Now, the way partitioned matrix multiplication works, this doesn't make sense:

$$B = \underbrace{\left( \begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \hat{B}}_{\left( \begin{array}{c} L_{TL} \times \text{something} + 0 \times \text{something} \\ \hline L_{BL} \times \text{something} + L_{BR} \times \text{something} \end{array} \right)}.$$

So, we need to also partition  $B$  into a top part and a bottom part:

$$\begin{pmatrix} B_T \\ B_B \end{pmatrix} = \underbrace{\begin{pmatrix} L_{TL} & 0 \\ L_{BL} & L_{BR} \end{pmatrix}}_{\begin{pmatrix} L_{TL}\hat{B}_T + 0 \times \hat{B}_B \\ L_{BL}\hat{B}_T + L_{BR}\hat{B}_B \end{pmatrix}} \begin{pmatrix} \hat{B}_T \\ \hat{B}_B \end{pmatrix}.$$

Alternatively, what if you don't partition  $L$ ? You have to partition *something* so let's try partitioning  $B$ :

$$\begin{pmatrix} B_T \\ B_B \end{pmatrix} = L \begin{pmatrix} \hat{B}_T \\ \hat{B}_B \end{pmatrix}.$$

But that doesn't work... Instead

$$\left( B_L \mid B_R \right) = L \left( \hat{B}_L \mid \hat{B}_R \right) = \left( L\hat{B}_L \mid L\hat{B}_R \right)$$

works just fine.

## 1.4 Deriving all unblocked algorithms


The below table summarizes all loop invariants, with links to all files related to this operation.

The worksheet and code skeletons were generated using the  [Spark webpage](#).

	Invariant	Derivations	Implementations
1	$\begin{pmatrix} B_T \\ B_B \end{pmatrix} = \begin{pmatrix} \hat{B}_T \\ L_{BR}\hat{B}_B \end{pmatrix}$	<a href="#">PDF</a>	<a href="#">trmm_llnn_unb_var1.mlx</a> <a href="#">trmm_llnn_unb_var1.c</a>
2	$\begin{pmatrix} B_T \\ B_B \end{pmatrix} = \begin{pmatrix} \hat{B}_T \\ L_{BL}B_T + L_{BR}\hat{B}_B \end{pmatrix}$	<a href="#">PDF</a>	<a href="#">trmm_llnn_unb_var2.mlx</a> <a href="#">trmm_llnn_unb_var2.c</a>
3	$\left( B_L \mid B_R \right) = \left( L\hat{B}_L \mid \hat{B}_R \right)$	<a href="#">PDF</a>	<a href="#">trmm_llnn_unb_var3.mlx</a> <a href="#">trmm_llnn_unb_var3.c</a>
4	$\left( B_L \mid B_R \right) = \left( \hat{B}_L \mid L\hat{B}_R \right)$	<a href="#">PDF</a>	<a href="#">trmm_llnn_unb_var4.mlx</a> <a href="#">trmm_llnn_unb_var4.c</a>

## 1.5 Deriving all blocked algorithms

The below table summarizes all loop invariants, with links to all files related to this operation.

The worksheet and code skeletons were generated using the  [Spark webpage](#).

	Invariant	Derivations	Implementations
1	$\left( \frac{B_T}{B_B} \right) = \left( \frac{\hat{B}_T}{L_{BR}\hat{B}_B} \right)$	PDF	<a href="#">trmm_llnn_blk_var1.mlx</a> <a href="#">trmm_llnn_blk_var1.c</a>
2	$\left( \frac{B_T}{B_B} \right) = \left( \frac{\hat{B}_T}{L_{BL}B_T + L_{BR}\hat{B}_B} \right)$	PDF	<a href="#">trmm_llnn_blk_var2.mlx</a> <a href="#">trmm_llnn_blk_var2.c</a>
3	$\left( B_L \mid B_R \right) = \left( L\hat{B}_L \mid \hat{B}_R \right)$	PDF	<a href="#">trmm_llnn_blk_var3.mlx</a> <a href="#">trmm_llnn_blk_var3.c</a>
4	$\left( B_L \mid B_R \right) = \left( \hat{B}_L \mid L\hat{B}_R \right)$	PDF	<a href="#">trmm_llnn_blk_var4.mlx</a> <a href="#">trmm_llnn_blk_var4.c</a>