Author(s) : H. Lipps **Library:** KERNLIB
Submitter : **Submitted:** 18.12.1979
Language : Fortran or Assembler or COMPASS **Revised:** 27.05.1987

Matrix Multiplication

These subprograms calculate the matrix product

$$\mathrm{Z} = \mathrm{XY} \text{ or } \mathrm{Z} = \mathrm{X}\overline{\mathrm{Y}},$$

where $\overline{Y}$ denotes the conjugate of the complex matrix **Y**, or one of the matrix expressions

$$\mathrm{Z} = \mathrm{XY} + \mathrm{Z}, \quad \mathrm{Z} = \mathrm{XY} - \mathrm{Z}, \quad \mathrm{Z} = -\mathrm{XY} + \mathrm{Z}, \quad \mathrm{Z} = -\mathrm{XY} - \mathrm{Z}.$$

Structure:

`SUBROUTINE` subprograms
User Entry Names: `RMMLA`, `RMMLS`, `RMMLT`, `RMNMA`, `RMNMS`,
`DMMLA`, `DMMLS`, `DMMLT`, `DMNMA`, `DMNMS`,
`CMMLA`, `CMMLS`, `CMMLT`, `CMNMA`, `CMNMS`, `CMMLTC`
External References: `LOCF` (N100) (some Fortran versions only).

Usage:

For `t` = `R` (type `REAL`), `t` = `D` (type `DOUBLE PRECISION`), `t` = `C` (type `COMPLEX`):

`CALL tMMLT (M,N,K,X11,X12,X21,Y11,Y12,Y21,Z11,Z12,Z21,W)`	$\mathrm{Z} = \mathrm{XY}$
`CALL tMMLA (M,N,K,X11,X12,X21,Y11,Y12,Y21,Z11,Z12,Z21)`	$\mathrm{Z} = \mathrm{XY} + \mathrm{Z}$
`CALL tMMLS (M,N,K,X11,X12,X21,Y11,Y12,Y21,Z11,Z12,Z21)`	$\mathrm{Z} = \mathrm{XY} - \mathrm{Z}$
`CALL tMNMA (M,N,K,X11,X12,X21,Y11,Y12,Y21,Z11,Z12,Z21)`	$\mathrm{Z} = -\mathrm{XY} + \mathrm{Z}$
`CALL tMNMS (M,N,K,X11,X12,X21,Y11,Y12,Y21,Z11,Z12,Z21)`	$\mathrm{Z} = -\mathrm{XY} - \mathrm{Z}$
`CALL CMMLTC(M,N,K,X11,X12,X21,Y11,Y12,Y21,Z11,Z12,Z21,W)`	$\mathrm{Z} = \mathrm{X}\overline{\mathrm{Y}}$

`M,N,K` (`INTEGER`) The mathematical dimensions of the matrices: **X** has `M` rows and `N` columns, **Y** has `N` rows and `K` columns, **Z** has `M` rows and `K` columns.

`X11,X12,X21` (Type according to `t`) Array elements. They must contain the elements x_{11}, x_{12}, x_{21} of the matrix **X**.

`Y11,Y12,Y21` (Type according to `t`) Array elements. They must contain the elements y_{11}, y_{12}, y_{21} of the matrix **Y**.

`Z11,Z12,Z21` (Type according to `t`) Array elements. On exit, they will contain the elements z_{11}, z_{12}, z_{21} of the matrix **Z**.

`W` (Type according to `t`) Working space array as specified below, required only if **Z** overlaps **X** or **Y**. Otherwise a dummy variable.

For `M` $<$ `1` or `N` $<$ `1` or `K` $<$ `1`, all subroutines return control without action.

The matrices **X**, **Y** and **Z** need not to be stored according to the Fortran conventions: any equidistant spacing of their rows and columns is permitted. In particular, matrices may be stored row-wise. Each subroutine can work with the transpose of a matrix. To make this possible, each matrix is specified in the calling sequence by three arguments. For example, the called subroutine will operate on the matrix $\mathbf{A} = (a_{ij})$ if the actual arguments which replace `X11`, `X12`, `X21` in the calling sequence are a_{11}, a_{12}, a_{21}, and will operate on the transpose $\mathbf{A}'$ of **A** if the actual arguments are a_{11}, a_{21}, a_{12}.

The only cases in which the result matrix **Z** is permitted to overlap **X** or **Y** are the following:

tMMLT:	$\mathbf{X} = \mathbf{XY}$	or	$\mathbf{Y} = \mathbf{Y}'\mathbf{Y}$,	provided W is an array of at least K elements.
	$\mathbf{Y} = \mathbf{XY}$	or	$\mathbf{X} = \mathbf{XX}'$,	provided W is an array of at least M elements.
CMMLTC:	$\mathbf{X} = \mathbf{X}\overline{\mathbf{Y}}$	or	$\mathbf{Y} = \mathbf{Y}'\overline{\mathbf{Y}}$,	provided W is an array of at least K elements.
	$\mathbf{Y} = \mathbf{X}\overline{\mathbf{Y}}$	or	$\mathbf{X} = \mathbf{X}\overline{\mathbf{X}}'$,	provided W is an array of at least M elements.

Accuracy:

On computers with IBM 370 architecture, all routines that accumulate the inner product of type REAL or COMPLEX use double-precision arithmetic internally; the final result is then rounded to single precision.

Notes:

The product of a matrix and its transpose (or Hermitian conjugate) is recognized by tMMLT (or CMMLTC) and the computation is shortened accordingly.

Examples:

Assume that the two-dimensional arrays A, B, C, D, E, the one-dimensional array W, and the dummy variable V are declared by

```
COMPLEX A(9,9),B(9,9),C(9,9),D(9,9),E(9,9),V,W(99)
```

and that a 4×5 matrix **A**, a 5×7 matrix **B**, and a 7×3 matrix **C** have been stored according to the Fortran conventions in arrays of corresponding name.

1. To compute $\mathbf{D} = \mathbf{AB}$:

   ```
   CALL CMMLT (4,5,7,A,A(1,2),A(2,1),B,B(1,2),B(2,1),D,D(1,2),D(2,1),V).
   ```

   To pack the 4×7 product matrix **AB** row-wise into array W:

   ```
   CALL CMMLT (4,5,7,A,A(1,2),A(2,1),B,B(2,1),B(1,2),W,W(2),W(8),V).
   ```

   (Note that z_{11} goes into W(1), z_{12} into W(2), and z_{21} into W(8)).

   For the purpose of abbreviation we shall denote
   A,A(1,2),A(2,1) by a,   A,A(2,1),A(1,2) by a',
   and similarly for arrays B, C, D, E. The first example above then becomes

   ```
   CALL CMMLT(4,5,7,a,b,d,V).
   ```

2. To compute $\mathbf{D} = \mathbf{B}'\mathbf{A}' = (\mathbf{AB})'$:

   ```
   CALL CMMLT(7,5,4,b',a',d,V)  or  CMMLT(4,5,7,a,b,d',V).
   ```

3. To compute $\mathbf{D} = \mathbf{AA}'$ and $\mathbf{E} = \mathbf{A}'\mathbf{A}$:

   ```
   CALL CMMLT(4,5,4,a,a',d,V)
   CALL CMMLT(5,4,5,a',a,e,V).
   ```

4. To replace **A** by **AB** or by $\mathbf{AA}'$:

   ```
   CALL CMMLT(4,5,7,a,b,a,W)  or  CALL CMMLT(4,5,4,a,a',a,W).
   ```

   These two calls require a working vector W containing 7 or 4 complex elements, respectively.

5. To compute $\mathbf{D} = \mathbf{A}\overline{\mathbf{B}}$ and $\mathbf{E} = \overline{\mathbf{B}}\mathbf{C} = (\mathbf{C}'\overline{\mathbf{B}}')'$:

   ```
   CALL CMMLTC(4,5,7,a,b,d,V)
   CALL CMMLTC(3,7,5,c',b',e',V).
   ```

•