

```

c***** Matrix times matrix subroutine
c..... stored in sub_directory *duc*todd*qin*dense*/qin.f (see subroutine matmat
SUBROUTINE matmat(M,N,L,A,B,C)

```

```

c
c
c Runesha Dec 1, 99
c
c subroutine for matrix matrix multiplication
c
c      C = A * B
c Dimensions:
c      C(M,N)
c      A(M,L)
c      B(L,N)
c
c
c The blocking technique is used: IB for I, JB for J and KB for K
c A temporary array TEMP is used to reduce the stride on A
c different loop unrolling level are considered
c
c NOTE:
c IBLOCK, JBLOCK, KBLOCK : pour considerer soit block ou le leftover
c ILOOP, JLOOP, KLOOP : pour considerer le unrolling du block considere
c Temp est une petite matrix de service qui contenir dans le cache
c
c

```

```

INTEGER      M,N, L
DOUBLE PRECISION  A( M, * ), B( L, * ), C( M, * )
INTEGER      I, II, IBLOCK, ILOOP, J, JJ, JBLOCK, JLOOP,
$            K, KK, KBLOCK, KLOOP
DOUBLE PRECISION  C11, C12, C21, C22, C31, C32, C41, C42
DOUBLE PRECISION  C13, C14, C23, C24, C33, C34, C43, C44
INTRINSIC      MIN, MOD
INTEGER      IB, JB, KB
PARAMETER      ( IB = 32, JB = 32, KB = 32 )
DOUBLE PRECISION  TEMP( KB, IB )

```

```

*
```

```

DO 20 JJ = 1, N, JB
JBLOCK = MIN( JB, N-JJ+1 )
JLOOP = JBLOCK-MOD( JBLOCK, 4 )
DO 30 KK = 1, L, KB
KBLOCK = MIN( KB, L-KK+1 )
KLOOP = KBLOCK-MOD( KBLOCK, 2 )

```

```

DO 10 II = 1, M, IB
IBLOCK = MIN( IB, M-II+1 )
ILOOP = IBLOCK-MOD( IBLOCK, 2 )

```

c

c mettre une portion de A dans tcache éviter le grand stride

```
DO 13, K = KK, KK+KLOOP-1, 2
  DO 11, I = II, II+ILOOP-1, 2
    TEMP( K-KK+1, I-II+1 ) = A( I, K )
    TEMP( K-KK+2, I-II+1 ) = A( I, K+1 )
    TEMP( K-KK+1, I-II+2 ) = A( I+1, K )
    TEMP( K-KK+2, I-II+2 ) = A( I+1, K+1 )
11  CONTINUE
    IF( ILOOP.LT.IBLOCK )THEN
      TEMP( K-KK+1, IBLOCK ) = A( II+IBLOCK-1, K )
      TEMP( K-KK+2, IBLOCK ) = A( II+IBLOCK-1, K+1 )
    END IF
13  CONTINUE
    IF( KLOOP.LT.KBLOCK )THEN
      DO 12, I = II, II+IBLOCK-1
        TEMP( KBLOCK, I-II+1 ) = A( I, KK+KBLOCK-1 )
12  CONTINUE
      END IF
```

c

c commencer la multiplication ici

```
ILOOP = IBLOCK-MOD( IBLOCK, 4 )
DO 22 J = JJ, JJ+JLOOP-1, 4
  DO 21 I = II, II+ILOOP-1, 4
    C11 = C( I,J )
    C21 = C( I+1,J )
    C12 = C( I,J+1 )
    C22 = C( I+1,J+1 )
    C13 = C( I,J+2 )
    C23 = C( I+1,J+2 )
    C14 = C( I,J+3 )
    C24 = C( I+1,J+3 )
    C31 = C( I+2,J )
    C41 = C( I+3,J )
    C32 = C( I+2,J+1 )
    C42 = C( I+3,J+1 )
    C33 = C( I+2,J+2 )
    C43 = C( I+3,J+2 )
    C34 = C( I+2,J+3 )
    C44 = C( I+3,J+3 )
  DO 23 K = KK, KK+KBLOCK-1
    C11 = C11 + TEMP( K-KK+1, I-II+1 )*B( K, J )
    C21 = C21 + TEMP( K-KK+1, I-II+2 )*B( K, J )
    C12 = C12 + TEMP( K-KK+1, I-II+1 )*B( K, J+1 )
    C22 = C22 + TEMP( K-KK+1, I-II+2 )*B( K, J+1 )
    C13 = C13 + TEMP( K-KK+1, I-II+1 )*B( K, J+2 )
```

```

C23 = C23 + TEMP( K-KK+1, I-II+2 ) * B( K, J+2 )
C14 = C14 + TEMP( K-KK+1, I-II+1 ) * B( K, J+3 )
C24 = C24 + TEMP( K-KK+1, I-II+2 ) * B( K, J+3 )
C31 = C31 + TEMP( K-KK+1, I-II+3 ) * B( K, J )
C41 = C41 + TEMP( K-KK+1, I-II+4 ) * B( K, J )
C32 = C32 + TEMP( K-KK+1, I-II+3 ) * B( K, J+1 )
C42 = C42 + TEMP( K-KK+1, I-II+4 ) * B( K, J+1 )
C33 = C33 + TEMP( K-KK+1, I-II+3 ) * B( K, J+2 )
C43 = C43 + TEMP( K-KK+1, I-II+4 ) * B( K, J+2 )
C34 = C34 + TEMP( K-KK+1, I-II+3 ) * B( K, J+3 )
C44 = C44 + TEMP( K-KK+1, I-II+4 ) * B( K, J+3 )
23  CONTINUE
C( I, J ) = C11
C( I+1, J ) = C21
C( I, J+1 ) = C12
C( I+1, J+1 ) = C22
C( I, J+2 ) = C13
C( I+1, J+2 ) = C23
C( I, J+3 ) = C14
C( I+1, J+3 ) = C24
C( I+2, J ) = C31
C( I+3, J ) = C41
C( I+2, J+1 ) = C32
C( I+3, J+1 ) = C42
C( I+2, J+2 ) = C33
C( I+3, J+2 ) = C43
C( I+2, J+3 ) = C34
C( I+3, J+3 ) = C44
21  CONTINUE
IF( ILOOP.LT.IBLOCK ) THEN
DO 221 I = II+ILOOP, II+IBLOCK-1
C11 = C( I, J )
C12 = C( I, J+1 )
C13 = C( I, J+2 )
C14 = C( I, J+3 )
DO 223 K = KK, KK+KBLOCK-1
C11 = C11 + TEMP( K-KK+1, I-II+1 ) * B( K, J )
C12 = C12 + TEMP( K-KK+1, I-II+1 ) * B( K, J+1 )
C13 = C13 + TEMP( K-KK+1, I-II+1 ) * B( K, J+2 )
C14 = C14 + TEMP( K-KK+1, I-II+1 ) * B( K, J+3 )
223  CONTINUE
C( I, J ) = C11
C( I, J+1 ) = C12
C( I, J+2 ) = C13
C( I, J+3 ) = C14
221  CONTINUE
END IF
22  CONTINUE

```

```

IF( JLOOP.LT.JBLOCK )THEN
DO 32 J = JJ+JLOOP, JJ+JBLOCK-1
  DO 31 I = II, II+ILOOP-1, 4
    C11 = C( I,J )
    C21 = C( I+1, J )
    C31 = C( I+2, J )
    C41 = C( I+3, J )
    DO 33 K = KK, KK+KBLOCK-1
      C11 = C11 + TEMP( K-KK+1, I-II+1 )*B( K, J )
      C21 = C21 + TEMP( K-KK+1, I-II+2 )*B( K, J )
      C31 = C31 + TEMP( K-KK+1, I-II+3 )*B( K, J )
      C41 = C41 + TEMP( K-KK+1, I-II+4 )*B( K, J )
33    CONTINUE
      C( I,J ) = C11
      C( I+1, J ) = C21
      C( I+2, J ) = C31
      C( I+3, J ) = C41
31    CONTINUE
    DO 41 I = II+ILOOP, II+IBLOCK-1
      C11 = C( I, J )
      DO 43 K = KK, KK+KBLOCK-1
        C11 = C11 + TEMP( K-KK+1, I-II+1 )*B( K, J )
43      CONTINUE
        C( I, J ) = C11
41      CONTINUE
32    CONTINUE
  END IF

10  CONTINUE
30  CONTINUE
20  CONTINUE
*
*

RETURN
*

END

```