

Libflame – no more "0 users, 0 complaints"

Robert van de Geijn

The Science of High-Performance Computing Group



It takes a village

- Many have contributed to the FLAME libflame projects. My apologies for not listing all
- Early support for FLAME came from NSF and a number of corporate gifts (Microsoft, Intel, and others)
- The purpose of this talk is to quickly get to talking about the future



FLAME is ...

- A notation
- A methodology for deriving families of algorithms
- A family of APIs
- A library (libflame)
- A productivity multiplier
- A future



FLAME is a notation

Algorithm: $A := \text{CHOL_BLK_VAR3}(A)$

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

where A_{TL} is 0×0

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

Repertition

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

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

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

$A_{21} := L_{21} = A_{21}L_{11}^{-T}$ (TRSM)

$A_{22} := A_{22} - L_{21}L_{21}^T$ (SYRK)

Continue with

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

endwhile

FLAME is a methodology

Step	Annotated Algorithm: $A := \text{CHOL}(A)$
1a	$\{A = \hat{A}\}$
4	Partition $A \rightarrow \left(\begin{array}{c c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right)$, $L \rightarrow \left(\begin{array}{c c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right)$, $U \rightarrow \left(\begin{array}{c c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array} \right)$ where A_{TL} , L_{TL} , and U_{TL} are 0×0
2	$\left\{ \left(\begin{array}{c c} A_{TL} & * \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c c} L_{TL} & * \\ \hline L_{BL} & \hat{A}_{BR} - L_{BL}L_{TL}^T \end{array} \right) \wedge \begin{array}{l} L_{TL}L_{TL}^T = \hat{A}_{TL} \\ L_{BL}L_{TL}^T = \hat{A}_{BL} \end{array} \right\}$
3	while $m(A_{TL}) < m(A)$ do
2,3	$\left\{ \left(\left(\begin{array}{c c} A_{TL} & * \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c c} L_{TL} & * \\ \hline L_{BL} & \hat{A}_{BR} - L_{BL}L_{TL}^T \end{array} \right) \wedge \begin{array}{l} L_{TL}L_{TL}^T = \hat{A}_{TL} \\ L_{BL}L_{TL}^T = \hat{A}_{BL} \end{array} \right) \wedge m(A_{TL}) < m(A) \right\}$
5a	Repartition $\left(\begin{array}{c c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c c c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c c c} L_{00} & 0 & 0 \\ \hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array} \right)$ where A_{11} and L_{11} are $b \times b$
6	$\left\{ \left(\begin{array}{c c c} A_{00} & * & * \\ \hline A_{10} & A_{11} & * \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right) = \left(\begin{array}{c c c} L_{00} & * & * \\ \hline L_{10} & \hat{A}_{11} - L_{10}L_{10}^T & * \\ \hline L_{20} & \hat{A}_{21} - L_{20}L_{10}^T & \hat{A}_{22} - L_{20}L_{20}^T \end{array} \right) \wedge \begin{array}{l} L_{00}L_{00}^T = \hat{A}_{00} \\ L_{10}L_{00}^T = \hat{A}_{10} \\ L_{20}L_{00}^T = \hat{A}_{20} \end{array} \right\}$
8	$A_{11} := L_{11} = \text{Chol}(A_{11})$ $A_{21} := L_{21} = A_{21}L_{11}^{-T}$ (TRSM) $A_{22} := A_{22} - L_{21}L_{11}^T$ (SYRK)
5b	Continue with $\left(\begin{array}{c c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c c c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c c c} L_{00} & 0 & 0 \\ \hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array} \right)$
7	$\left\{ \begin{array}{l} \left(\begin{array}{c c c} A_{00} & * & * \\ \hline A_{10} & A_{11} & * \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right) = \left(\begin{array}{c c c} L_{00} & * & * \\ \hline L_{10} & L_{11} & * \\ \hline L_{20} & L_{21} & \hat{A}_{22} - L_{20}L_{20}^T - L_{21}L_{11}^T \end{array} \right) \wedge \begin{array}{l} L_{00}L_{00}^T = \hat{A}_{00} \\ L_{10}L_{00}^T = \hat{A}_{10} \\ L_{20}L_{00}^T = \hat{A}_{20} \end{array} \\ L_{10}L_{11}^T + L_{11}L_{11}^T = \hat{A}_{11} \\ L_{20}L_{11}^T + L_{21}L_{11}^T = \hat{A}_{21} \end{array} \right\}$
2	$\left\{ \left(\begin{array}{c c} A_{TL} & * \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c c} L_{TL} & * \\ \hline L_{BL} & \hat{A}_{BR} - L_{BL}L_{TL}^T \end{array} \right) \wedge \begin{array}{l} L_{TL}L_{TL}^T = \hat{A}_{TL} \\ L_{BL}L_{TL}^T = \hat{A}_{BL} \end{array} \right\}$
	endwhile
2,3	$\left\{ \left(\left(\begin{array}{c c} A_{TL} & * \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c c} L_{TL} & * \\ \hline L_{BL} & \hat{A}_{BR} - L_{BL}L_{TL}^T \end{array} \right) \wedge \begin{array}{l} L_{TL}L_{TL}^T = \hat{A}_{TL} \\ L_{BL}L_{TL}^T = \hat{A}_{BL} \end{array} \right) \wedge \neg (m(A_{TL}) < m(A)) \right\}$
1b	$\{A = L \wedge LL^T = \hat{A}\}$



FLAME is a family of APIs

Algorithm: $A := \text{CHOL_BLK_VAR3}(A)$

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

where A_{TL} is 0×0

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

Repartition

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

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

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

$$A_{21} := L_{21} = A_{21}L_{11}^{-T} \quad (\text{TRSM})$$

$$A_{22} := A_{22} - L_{21}L_{21}^T \quad (\text{SYRK})$$

Continue with

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

endwhile

```

FLA_Part_2x2( A,      &ATL, &ATR,
              &ABL, &ABR,      0, 0, FLA_TL );

while ( FLA_Obj_length( ATL ) < FLA_Obj_length( A ) ){

    b = FLA_Determine_blocksize( ABR, FLA_BR, FLA_Cntl_blocksize( cntl ) );

    FLA_Repart_2x2_to_3x3( ATL, /**/ ATR,      &A00, /**/ &A01, &A02,
                          /* ***** */ /* ***** */
                          &A10, /**/ &A11, &A12,
                          ABL, /**/ ABR,      &A20, /**/ &A21, &A22,
                          b, b, FLA_BR );

    /*-----*/

    // A11 = chol( A11 )
    r_val = FLA_Chol_internal( FLA_LOWER_TRIANGULAR, A11,
                              FLA_Cntl_sub_chol( cntl ) );

    if ( r_val != FLA_SUCCESS )
        return ( FLA_Obj_length( A00 ) + r_val );

    // A21 = A21 * inv( tril( A11 )' )
    FLA_Trsm_internal( FLA_RIGHT, FLA_LOWER_TRIANGULAR,
                      FLA_CONJ_TRANSPOSE, FLA_NONUNIT_DIAG,
                      FLA_ONE, A11, A21,
                      FLA_Cntl_sub_trsm( cntl ) );

    // A22 = A22 - A21 * A21'
    FLA_Herk_internal( FLA_LOWER_TRIANGULAR, FLA_NO_TRANSPOSE,
                      FLA_MINUS_ONE, A21, FLA_ONE, A22,
                      FLA_Cntl_sub_herk( cntl ) );

    /*-----*/

    FLA_Cont_with_3x3_to_2x2( &ATL, /**/ &ATR,      A00, A01, /**/ A02,
                              A10, A11, /**/ A12,
                              /* ***** */ /* ***** */
                              &ABL, /**/ &ABR,      A20, A21, /**/ A22,
                              FLA_TL );

}

```

Algorithm: $A := \text{CHOL_BLK_VAR3}(A)$

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

where A_{TL} is 0×0

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

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \end{array} \right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right)$$

```
FLA_Trsm_internal( FLA_RIGHT, FLA_LOWER_TRIANGULAR,
                   FLA_CONJ_TRANSPOSE, FLA_NONUNIT_DIAG,
                   FLA_ONE, A11, A21,
                   FLA_Cntl_sub_trsm( cntl ) );
```

Continue with

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

endwhile

```
FLA_Part_2x2( A, &ATL, &ATR,
              &ABL, &ABR, 0, 0, FLA_TL );

while ( FLA_Obj_length( ATL ) < FLA_Obj_length( A ) ){

    b = FLA_Determine_blocksize( ABR, FLA_BR, FLA_Cntl_blocksize( cntl ) );

    FLA_Repart_2x2_to_3x3( ATL, /**/ ATR, &A00, /**/ &A01, &A02,
                          /* ***** */ /* ***** */
                          &A10, /**/ &A11, &A12,
                          ABL, /**/ ABR, &A20, /**/ &A21, &A22,
                          b, b, FLA_BR );

    /*-----*/
    // A11 = chol( A11 )

    FLA_One( FLA_ONE, A11, A21,
            FLA_Cntl_sub_trsm( cntl ) );

    // A22 = A22 - A21 * A21'
    FLA_Herk_internal( FLA_LOWER_TRIANGULAR, FLA_NO_TRANSPOSE,
                      FLA_MINUS_ONE, A21, FLA_ONE, A22,
                      FLA_Cntl_sub_herk( cntl ) );

    /*-----*/

    FLA_Cont_with_3x3_to_2x2( &ATL, /**/ &ATR, A00, A01, /**/ A02,
                             A10, A11, /**/ A12,
                             /* ***** */ /* ***** */
                             &ABL, /**/ &ABR, A20, A21, /**/ A22,
                             FLA_TL );

}
```




```

FLA_Part_2x2( A,      &ATL, &ATR,
               &ABL, &ABR,      0, 0, FLA_TL );

while ( FLA_Obj_length( ATL ) < FLA_Obj_length( A ) ){

    FLA_Repart_2x2_to_3x3( ATL, /**/ ATR,      &A00, /**/ &a01,      &A02,
                          /**/ ***** */      /**/ ***** */
                          &a10t, /**/ &alpha11, &a12t,
                          ABL, /**/ ABR,      &A20, /**/ &a21,      &A22,
                          1, 1, FLA_BR );

    /*-----*/

    // alpha11 = sqrt( alpha11 )
    r_val = FLA_Sqrt( alpha11 );

    if ( r_val != FLA_SUCCESS )
        return ( FLA_Obj_length( A00 ) );

    // a21 = a21 / alpha11
    FLA_Inv_scal_external( alpha11, a21 );

    // A22 = A22 - a21 * a21'
    FLA_Her_external( FLA_LOWER_TRIANGULAR, FLA_MINUS_ONE, a21, A22 );

    /*-----*/

    FLA_Cont_with_3x3_to_2x2( &ATL, /**/ &ATR,      A00, a01,      /**/ A02,
                              a10t, alpha11, /**/ a12t,
                              /**/ ***** */      /**/ ***** */
                              &ABL, /**/ &ABR,      A20, a21,      /**/ A22,
                              FLA_TL );
}

```

```

for ( i = 0; i < mn_A; ++i )
{
    float*    alpha11    = buff_A + (i )*cs_A + (i )*rs_A;
    float*    a21        = buff_A + (i )*cs_A + (i+1)*rs_A;
    float*    A22        = buff_A + (i+1)*cs_A + (i+1)*rs_A;

    int        mn_ahead  = mn_A - i - 1;
    int        mn_behind = i;

    /*-----*/

    // r_val = FLA_Sqrt( alpha11 );
    // if ( r_val != FLA_SUCCESS )
    //     return ( FLA_Obj_length( A00 ) + 1 );
    bl1_ssqrte( alpha11, &e_val );
    if ( e_val != FLA_SUCCESS ) return mn_behind;

    // FLA_Inv_scal_external( alpha11, a21 );
    bl1_sinvscalv( BLIS1_NO_CONJUGATE,
                  mn_ahead,
                  alpha11,
                  a21, rs_A );

    // FLA_Her_external( FLA_LOWER_TRIANGULAR, FLA_MINUS_ONE, a21, A22 );
    bl1_ssyr( BLIS1_LOWER_TRIANGULAR,
              mn_ahead,
              buff_m1,
              a21, rs_A,
              A22, rs_A, cs_A );

    /*-----*/

}

```



FLAME is a library (libflame)

The screenshot shows the GitHub repository page for `flame / libflame`. The repository is public and has 78 forks and 212 stars. The main branch is `master`. The repository description is "High-performance object-based library for DLA computations". The repository contains several folders: `build`, `docs/libflame`, `examples`, and `netlib-test`. The repository was last updated on September 19, 2022, by `iotamudelta`. The repository has 113 issues.

Repository Information:

- Repository: `flame / libflame` (Public)
- Notifications
- Fork: 78
- Star: 212

Repository Navigation:

- Code
- Issues: 22
- Pull requests: 3
- Actions
- Projects
- Wiki
- Security
- Insights

Repository Branches:

- master

Repository Code:

- Code

About:

High-performance object-based library for DLA computations

Tags:

- high-performance
- linear-algebra
- matrix-functions
- high-performance-computing
- lapack
- linear-algebra-library
- flame
- matrix-library
- matrix-computations

Repository Commits:

Commit	Message	Date
<code>iotamudelta</code>	Update rocblas/rocsolver ...	Sep 19, 2022
<code>build</code>	Multithreading must be on for HIP...	Jul 11, 2022
<code>docs/libflame</code>	Improve mGPU handling for HIP. (...)	Jul 23, 2022
<code>examples</code>	Updates to hevd_test, gsvd_test.	Jan 30, 2018
<code>netlib-test</code>	Small tweak to netlib LAPACK tes...	Jan 24, 2018

FLAME is a library (libflame)

- Core libflame
 - Writing with the FLAMEC API
 - Overlaps with much of LAPACK functionality
- flapack
 - Adds functionality of LAPACK not in core libflame
 - LAPACK run through f2c
- libflame = core libflame + flapack
 - Does not require a fortran compiler
 - All of lapack functionality (circa 2014?)



FLAME is a productivity multiplier

- ...
- So many algorithms! When to pick which?
 - Control trees (that also appear in BLIS)
- SuperMatrix (algorithms by blocks)
 - Unit of data: submatrix
 - Unit of computation: operation with submatrices
 - Execute the algorithm to build DAG of tasks
 - Runtime schedules tasks to resources (multiple CPU and/or GPUs)
 - Runtime can, for example, incorporate a software cache



SuperMatrix code

```

FLA_Part_2x2( A,      &ATL, &ATR,
               &ABL, &ABR,      0, 0, FLA_TL );

while ( FLA_Obj_length( ATL ) < FLA_Obj_length( A ) ){

    b = FLA_Determine_blocksize( ABR, FLA_BR, FLA_Cntl_blocksize( cntl ) );

    FLA_Repart_2x2_to_3x3( ATL, /**/ ATR,      &A00, /**/ &A01, &A02,
                           /**/ ***** */    /**/ ***** */
                           &A10, /**/ &A11, &A12,
                           ABL, /**/ ABR,      &A20, /**/ &A21, &A22,
                           b, b, FLA_BR );

    /*-----*/

    // A11 = chol( A11 )
    r_val = FLA_Chol_internal( FLA_LOWER_TRIANGULAR, A11,
                              FLA_Cntl_sub_chol( cntl ) );

    if ( r_val != FLA_SUCCESS )
        return ( FLA_Obj_length( A00 ) + r_val );

    // A21 = A21 * inv( tril( A11 )' )
    FLA_Trsm_internal( FLA_RIGHT, FLA_LOWER_TRIANGULAR,
                      FLA_CONJ_TRANSPOSE, FLA_NONUNIT_DIAG,
                      FLA_ONE, A11, A21,
                      FLA_Cntl_sub_trsm( cntl ) );

    // A22 = A22 - A21 * A21'
    FLA_Herk_internal( FLA_LOWER_TRIANGULAR, FLA_NO_TRANSPOSE,
                      FLA_MINUS_ONE, A21, FLA_ONE, A22,
                      FLA_Cntl_sub_herk( cntl ) );

    /*-----*/

    FLA_Cont_with_3x3_to_2x2( &ATL, /**/ &ATR,      A00, A01, /**/ A02,
                              A10, A11, /**/ A12,
                              /**/ ***** */    /**/ ***** */
                              &ABL, /**/ &ABR,      A20, A21, /**/ A22,
                              FLA_TL );

}

```



FLAME is a future

There are a number of efforts to leverage libflame

- Oracle:
 - Vertical integration of libflame and BLIS
 - Incorporation of key functionality of libflame into BLIS
- AMD:
 -
 -
- ...

How do we coordinate efforts?

