At the Heart of the Automation of Linear Algebra Algorithms

Paolo Bientinesi

AICES, RWTH Aachen
pauldj@aices.rwth-aachen.de

Workshop on Program Composition and Optimization
May 9-12, 2010
Dagstuhl, Germany

Objective: generation of algorithms

- Operations: $B \leftarrow L^{-1}BL^{-T}$, $y \leftarrow G^{-1}Ax$, $Ax = b$, ...

Objective: generation of algorithms

- Operations: $B \leftarrow L^{-1}BL^{-T}$, $y \leftarrow G^{-1}Ax$, $Ax = b$, ...
- Is it possible to automatically transform

$LU = A$ into

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 $n(A_{TL}) < n(A)$ **do**

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

$$A_{01} := L_{00}^{-1} A_{01}$$
$$A_{10} := A_{10} U_{00}^{-1}$$
$$A_{11} := \mathsf{LU}(A_{11} - A_{10}A_{01})$$

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

endwhile

?

Objective: generation of algorithms

- Operations: $B \leftarrow L^{-1}BL^{-T}$, $y \leftarrow G^{-1}Ax$, $Ax = b$, ...
- Is it possible to automatically transform

$LU = A$ into

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 $n(A_{TL}) < n(A)$ **do**

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

$$A_{01} := L_{00}^{-1} A_{01}$$
$$A_{10} := A_{10} U_{00}^{-1}$$
$$A_{11} := \mathsf{LU}(A_{11} - A_{10}A_{01})$$

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

endwhile

?

- High-level of abstraction
- Algorithms are then transformed into code: multithreaded, DAG, Matlab, symbolic...

Motivation: Performance

Algorithm of choice influenced by (too) MANY factors:

- cache sizes, #registers, memory bandwidth
- #cores, #sockets
- storage by rows, columns; sparse formats
- problem size
- GotoBLAS vs. MKL
- variants (transpose, upper/lower storage, side)
- parallelism: SMP, MPI, hybrid
- mesh size, nodes vs. threads, interconnect
- . . . (not even to mention Cloud)

Motivation: Performance

Algorithm of choice influenced by (too) MANY factors:

- cache sizes, #registers, memory bandwidth
- #cores, #sockets
- storage by rows, columns; sparse formats
- problem size
- GotoBLAS vs. MKL
- variants (transpose, upper/lower storage, side)
- parallelism: SMP, MPI, hybrid
- mesh size, nodes vs. threads, interconnect
- . . . (not even to mention Cloud)

Performance modeling? Tough one!

Performance $\Longleftrightarrow$ Multiple algorithms

- No one “best” algorithm for all the scenarios
- Optimization $\Rightarrow$ local minimum

Performance $\Longleftrightarrow$ Multiple algorithms

- No one “best” algorithm for all the scenarios
- Optimization $\Rightarrow$ local minimum

- Generation of many algorithms
- LA algorithms: nicely layered and modular
- Performance: nope

Performance $\Longleftrightarrow$ Multiple algorithms

- No one “best” algorithm for all the scenarios
- Optimization $\Rightarrow$ local minimum

- Generation of many algorithms
- LA algorithms: nicely layered and modular
- Performance: nope

- For now: automatic **generation** of algorithms
- Future: automatic **selection** of “best” algorithm

More motivation: unknown target operation

$$f(\alpha, x, y) = \alpha x + y \quad (\text{axpy})$$

$$\frac{df}{dv} = ?$$

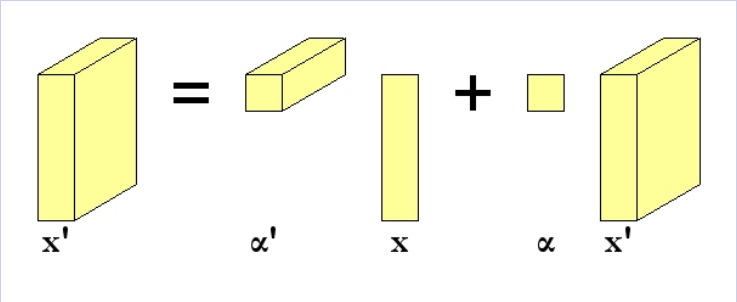
- $\alpha' x$
- $\alpha x'$
- $\alpha' x + \alpha x'$
- $\alpha' x + y'$
- $\alpha x' + y'$
- $\alpha' x + \alpha x' + y'$

More motivation: unknown target operation

$$f(\alpha, x, y) = \alpha x + y \quad (\text{axpy})$$

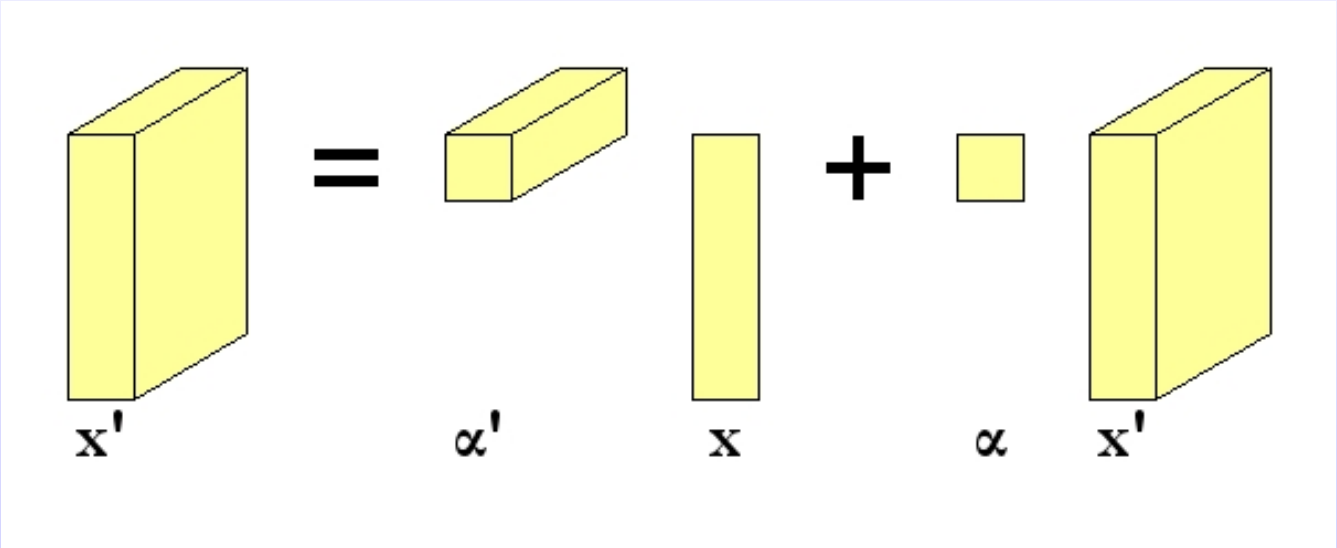
$$\frac{df}{dv} = ?$$

- $\alpha' x$
- $\alpha x'$
- $\alpha' x + \alpha x' \quad \rightarrow$
- $\alpha' x + y'$
- $\alpha x' + y'$
- $\alpha' x + \alpha x' + y'$

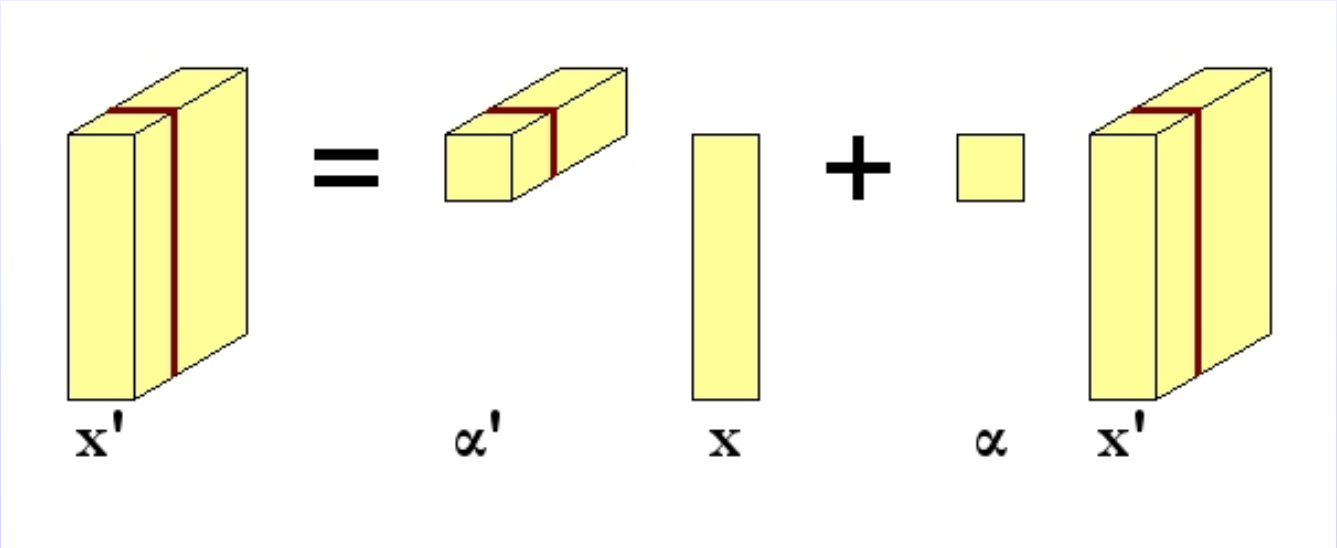


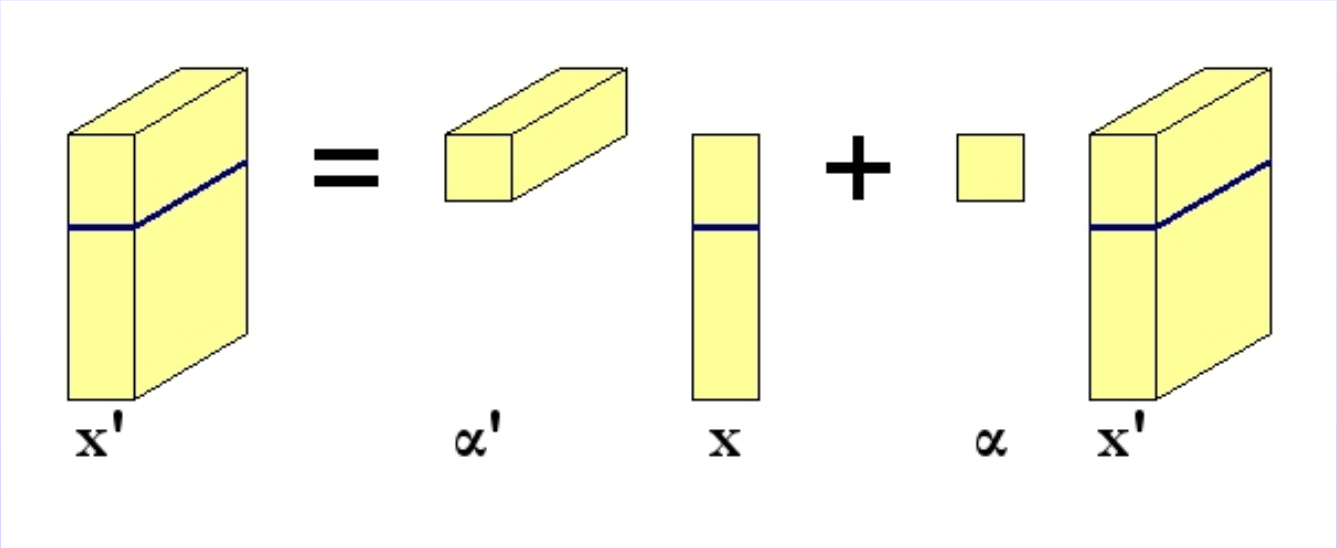
Extra complication: dimensionality

How to map high-dimensional objects onto BLAS?

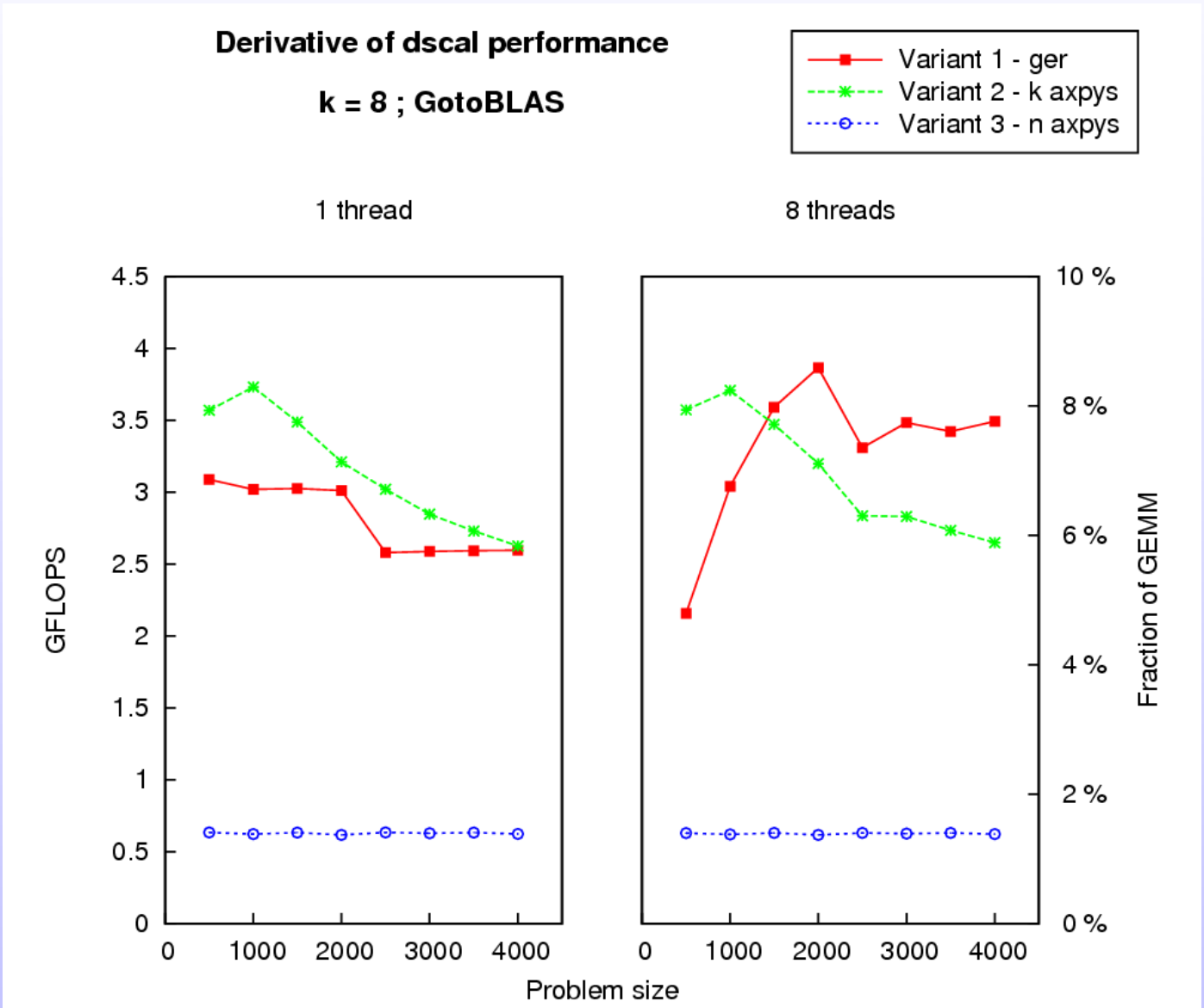

x'
=
α'
x
+
α
x'

D(axpy): k axpys

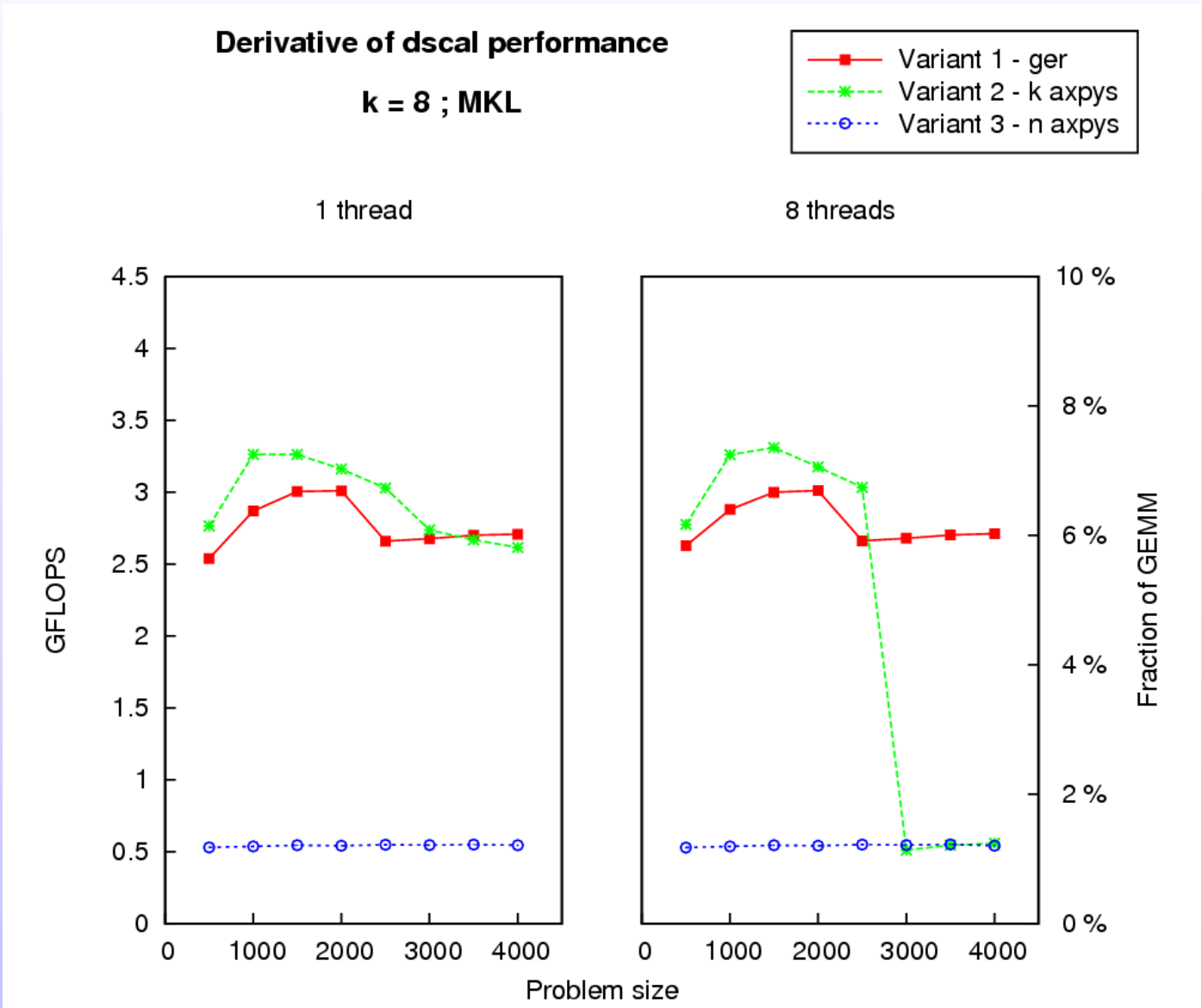



x'
=
α'
x
+
α
x'

D(axpy) – Performance



D(axpy) – Performance



$$f: \mathcal{X} \rightarrow \mathcal{Y} \quad \Rightarrow \quad \mathcal{P}$$

$$f\colon \mathcal{X} \to \mathcal{Y} \quad \Rightarrow \quad \mathcal{P}$$

Is $\mathcal{P}$ correct?

- $\forall x \in \mathcal{X}, \quad \mathcal{P}(x) \stackrel{?}{=} f(x)$

$$f\colon \mathcal{X} \to \mathcal{Y} \quad \Rightarrow \quad \mathcal{P}$$

Is $\mathcal{P}$ correct?

- $\forall x \in \mathcal{X}, \quad \mathcal{P}(x) \stackrel{?}{=} f(x)$
- Function definition

$$f \equiv \{f_{\mathrm{Pre}}\} \wedge \{f_{\mathrm{Post}}\}$$

- Hoare triples

$\{\mathrm{Pre}\}$	$\{f_{\mathrm{Pre}}\}$
Prog	$\mathcal{P}$
$\{\mathrm{Post}(?)\}$	$\{\mathrm{Post}(?)\}$ $\stackrel{?}{\Rightarrow} \{f_{\mathrm{Post}}\}$

How to generate $\mathcal{P}$ correct?

- $f \equiv \{f_{\text{Pre}}\} \wedge \{f_{\text{Post}}\}$

$$\{f_{\text{Pre}}\}$$
$$\mathcal{P}\ (?)$$
$$\{f_{\text{Post}}\}$$

How to generate $\mathcal{P}$ correct?

- $f \equiv \{f_{\text{Pre}}\} \wedge \{f_{\text{Post}}\}$

$\{f_{\text{Pre}}\}$

Init (?); *Loop* (?)

$\{f_{\text{Post}}\}$

How to generate $\mathcal{P}$ correct?

- $f \equiv \{f_{\text{Pre}}\} \wedge \{f_{\text{Post}}\}$

$\{f_{\text{Pre}}\}$

Init (?)

$\{Q\}$

Loop (?)

$\{f_{\text{Post}}\}$

How to generate $\mathcal{P}$ correct?

- Fundamental Invariance Theorem; Loop Invariant?

$\{f_{\text{Pre}}\}$

Init (?)

$\{Q\}$

Loop (?)

$\{f_{\text{Post}}\}$

How to generate $\mathcal{P}$ correct?

- Fundamental Invariance Theorem; Loop Invariant?

$$\{f_{\mathrm{Pre}}\}$$
Init (?)
$$\{\mathrm{L.Inv}\}$$
Loop (?)
$$\{\mathrm{L.Inv} \wedge \neg G\}$$
$$\{f_{\mathrm{Post}}\}$$

How to generate $\mathcal{P}$ correct?

- *Init* $\equiv$ Partitioning

$\{f_{\text{Pre}}\}$

Init (?)

$\{\text{L.Inv}\}$

Loop (?)

$\{\text{L.Inv} \wedge \neg G\}$

$\{f_{\text{Post}}\}$

How to generate $\mathcal{P}$ correct?

- *Init* $\equiv$ Partitioning

$$\{f_{\text{Pre}}\}$$
`Partition` (?)
$$\{\text{L.Inv}\}$$
Loop (?)
$$\{\text{L.Inv} \wedge \neg G\}$$
$$\{f_{\text{Post}}\}$$

How to generate $\mathcal{P}$ correct?

- *Loop* $\equiv$ Operand traversal + Computation

$\{f_{\text{Pre}}\}$

`Partition` (?)

$\{\text{L.Inv}\}$

Loop (?)

$\{\text{L.Inv} \wedge \neg G\}$

$\{f_{\text{Post}}\}$

Loop Invariant $\longrightarrow \cdots \longrightarrow$ Algorithm

Step	Annotated Algorithm: $\mathcal{P}$
1a	$\{\ f_{\text{Pre}}\ \}$
4	**Partition ... where ...**
2	$\{\ \text{L.Inv}\ \}$
3	**While** G **do**
2,3	$\{(\ \text{L.Inv}\ ) \wedge (\ G\ )\}$
5a	**Repartition ... where ...**
6	$\{\ P_{\text{before}}\ \}$
8	S_U
7	$\{\ P_{\text{after}}\ \}$
5b	**Continue with ...**
2	$\{\ \text{L.Inv}\ \}$
	endwhile
2,3	$\{(\ \text{L.Inv}\ ) \wedge \neg (\ G\ )\}$
1b	$\{\ f_{\text{Post}}\ \}$

Loop Invariant $\longrightarrow \cdots \longrightarrow$ Algorithm

Step	Annotated Algorithm: $\mathcal{P}$
1a	$\{\ f_{\text{Pre}}\ \}$
4	**Partition ... where ...**
2	$\{\ \text{L.Inv}\ \}$
3	**While** G **do**
2,3	$\{(\ \text{L.Inv}\ ) \wedge (\ G\ )\}$
5a	**Repartition ... where ...**
6	$\{\ P_{\text{before}}\ \}$
8	S_U
7	$\{\ P_{\text{after}}\ \}$
5b	**Continue with ...**
2	$\{\ \text{L.Inv}\ \}$
	endwhile
2,3	$\{(\ \text{L.Inv}\ ) \wedge \neg (\ G\ )\}$
1b	$\{\ f_{\text{Post}}\ \}$

- Worksheet: modular, general
- **Key point: loop Invariant!**

Roadmap to Loop Invariants

A simple(?) example: Cholesky Factorization

$$
\begin{aligned}
f: &\quad L := \Gamma(A) \\
f_{\text{Pre}}: &\quad \{SPD(A)\} \\
f_{\text{Post}}: &\quad \{LL^T = A \wedge \\
&\qquad \text{LowTri}(L)\}
\end{aligned}
$$

Roadmap to Loop Invariants

A simple(?) example: Cholesky Factorization

$$
\begin{aligned}
f: &\quad L := \Gamma(A) \\
f_{\text{Pre}}: &\quad \{SPD(A)\} \\
f_{\text{Post}}: &\quad \{LL^T = A \wedge \\
&\qquad \text{LowTri}(L)\}
\end{aligned}
$$

- Key question:
  Is it possible to express quadrants of L in terms of quadrants of A?

$$
\begin{aligned}
f: &\quad L := \Gamma(A) \\
f_{\text{Pre}}: &\quad \{SPD(A)\} \\
f_{\text{Post}}: &\quad \{LL^T = A \wedge \\
&\qquad \text{LowTri}(L)\}
\end{aligned}
$$

- Key question:
  Is it possible to express quadrants of L in terms of quadrants of A?

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

$$
\begin{aligned}
f: &\quad L := \Gamma(A) \\
f_{\text{Pre}}: &\quad \{SPD(A)\} \\
f_{\text{Post}}: &\quad \{LL^T = A \wedge \\
&\quad \text{LowTri}(L)\}
\end{aligned}
$$

- Key question:
  Is it possible to express quadrants of L in terms of quadrants of A?

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

Postcondition, partitioning:

$$
\left(\begin{array}{c|c} L_{TL} & \\ \hline L_{BL} & L_{BR} \end{array}\right)
\left(\begin{array}{c|c} L_{TL}^T & L_{BL}^T \\ \hline & L_{BR}^T \end{array}\right) =
\left(\begin{array}{c|c} A_{TL} & A_{BL}^T \\ \hline A_{BL} & A_{BR} \end{array}\right)
$$

A simple(?) example: Cholesky Factorization

$$
\begin{aligned}
f: &\quad L := \Gamma(A) \\
f_{\text{Pre}}: &\quad \{SPD(A)\} \\
f_{\text{Post}}: &\quad \{LL^T = A \wedge \\
&\quad \text{LowTri}(L)\}
\end{aligned}
$$

- Key question:
  Is it possible to express quadrants of L in terms of quadrants of A?

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

Simplification, algebraic manipulation:

$$\left(\begin{array}{c|c} L_{TL}L_{TL}^T = A_{TL} & \\ \hline L_{BL}L_{TL}^T = A_{BL} & L_{BL}L_{BL}^T + L_{BR}L_{BR}^T = A_{BR} \end{array}\right)$$

$$
\begin{aligned}
f: &\quad L := \Gamma(A) \\
f_{\text{Pre}}: &\quad \{SPD(A)\} \\
f_{\text{Post}}: &\quad \{LL^T = A \wedge \\
&\quad \text{LowTri}(L)\}
\end{aligned}
$$

- Key question:
  Is it possible to express quadrants of L in terms of quadrants of A?

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

Pattern matching:

$$
\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL} L_{TL}^{-T} & L_{BR} = \Gamma(A_{BR} - L_{BL} L_{BL}^T) \end{array}\right)
$$

Partitioned Matrix Expression

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL} L_{TL}^{-T} & L_{BR} = \Gamma\left(A_{BR} - L_{BL} L_{BL}^{T}\right) \end{array}\right)$$

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL} L_{TL}^{-T} & L_{BR} = \Gamma\left(A_{BR} - L_{BL} L_{BL}^{T}\right) \end{array}\right)$$

Operations:

$$\left(\begin{array}{c|c} 1)\ L_{TL} := \mathrm{CHOL} & \\ \hline 2)\ L_{BL} := \mathrm{TRSM} & 3)\ L_{BR} := \mathrm{CHOL(SYRK)} \end{array}\right)$$

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL} L_{TL}^{-T} & L_{BR} = \Gamma\left(A_{BR} - L_{BL} L_{BL}^{T}\right) \end{array}\right)$$

Dependencies:

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL} L_{TL}^{-T} & L_{BR} = \Gamma\left(A_{BR} - L_{BL} L_{BL}^{T}\right) \end{array}\right)$$

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL} L_{TL}^{-T} & L_{BR} = \Gamma\left(A_{BR} - L_{BL} L_{BL}^{T}\right) \end{array}\right)$$

Dependencies:

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL} L_{TL}^{-T} & L_{BR} = \Gamma\left(A_{BR} - L_{BL} {L_{BL}}^{T}\right) \end{array}\right)$$

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL}L_{TL}^{-T} & L_{BR} = \Gamma(A_{BR} - L_{BL}L_{BL}^{T}) \end{array}\right)$$

Loop Invariants:

- $\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL} & L_{BR} = A_{BR} \end{array}\right)$
- $\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL}L_{TL}^{-T} & L_{BR} = A_{BR} \end{array}\right)$
- $\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = A_{BL}L_{TL}^{-T} & L_{BR} = A_{BR} - L_{BL}L_{BL}^{T} \end{array}\right)$

Automation?

1)	f	$\longrightarrow$	PME	Keep Listening
2)	PME	$\longrightarrow$	Loop Invs	To Do
3)	Loop Inv	$\longrightarrow$	Algorithm	✓
4)	Algorithm	$\longrightarrow$	Code	✓
5)	Performance Prediction			In progress

As easy as it looks?

- Ambiguity
  - Notation & conventions
  - Inputs/ouputs
- Partitionings
  - Conformal partitionings
  - Non-uniqueness
  - Matrix properties
- Canonical form
  - Known/unknown quantities
- Extra knowledge

- Ambiguity
  - Notation & conventions
  - Inputs/ouputs

$$LL^T = A \quad \text{vs.} \quad A = LL^T$$

And what about

$$Z = XY, \quad XY = Z, \quad XY = Z, \quad XY = Z?$$

- Ambiguity
  - Notation & conventions
  - Inputs/ouputs

$$\underline{LL}^T = A \quad \text{vs.} \quad \underline{A} = LL^T$$

And what about

$$\underline{Z} = XY, \quad \underline{XY} = Z, \quad X\underline{Y} = Z, \quad \underline{X}Y = Z?$$

- Partitionings
  - Conformal partitionings
  - Non-uniqueness
  - Matrix properties

$$L\underline{X} = B$$

$$L\left(x_1|x_2|\cdots|x_n\right) = \left(b_1|b_2|\cdots|b_n\right)$$

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)\left(\begin{array}{c} X_T \\ \hline X_B \end{array}\right) = \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right)$$

- Partitionings
  - Conformal partitionings
  - Non-uniqueness
  - Matrix properties

$$L\underline{X} = B$$

$$L\left(x_1|x_2|\cdots|x_n\right) = \left(b_1|b_2|\cdots|b_n\right)$$

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)\left(\begin{array}{c} X_T \\ \hline X_B \end{array}\right) = \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right)$$

What about

$$AX + XB = C\ ?$$

- Canonical form
  - Known/unknown quantities

$$\left(\begin{array}{c|c} L_{TL}L_{TL}^T = A_{TL} & \\ \hline L_{BL}L_{TL}^T = A_{BL} & L_{BL}L_{BL}^T + L_{BR}L_{BR}^T = A_{BR} \end{array}\right)$$

$$\left(\begin{array}{c|c} L_{TL}L_{TL}^T = A_{TL} & \\ \hline L_{BL}L_{TL}^T = A_{BL} & L_{BR}L_{BR}^T = A_{BR} - L_{BL}L_{BL}^T \end{array}\right)$$

- Extra knowledge

$$\left(\begin{array}{c|c} L_{TL} = \Gamma(A_{TL}) & \\ \hline L_{BL} = \mathrm{TRSM} & L_{BR}L_{BR}^T = A_{BR} - L_{BL}L_{BL}^T \end{array}\right)$$

$$L_{BR} = \Gamma\left(A_{BR} - L_{BL}L_{BL}^T\right)$$

$$\mathrm{SPD}(A_{BR} - L_{BL}L_{BL}^T)?$$

Conclusions

- Abstraction $\Rightarrow$ Automation

Challenges

- High-level languages in HPC code (Python)
- Composing “2D” kernels to yield high-dim algs
- Modular performance modeling

- In collaboration with Diego Fabregat (AICES)
- Thanks to the FLAME team (UT Austin)
- Funding from DFG is gratefully acknowledged