Cl1ck: A code generator for linear algebra kernels

Diego Fabregat-Traver and Paolo Bientinesi

HPAC, RWTH Aachen
fabregat@aices.rwth-aachen.de

PL Lunch, September 12th, 2016
University of Texas at Austin

General idea

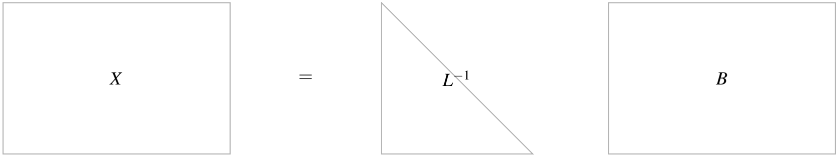
- From description of the target operation to algorithms and code
- DLA kernels: matrix products, factorizations, linear systems, ...
- Main focus of the talk: generation of algorithms
- Algorithms $\longrightarrow$ FLAME methodology (correct by construction)

Outline

1. FLAME: Notation and methodology
2. Cl1ck: Automating the FLAME methodology
3. Conclusions

Algorithm

Triangular system with multiple right-hand sides

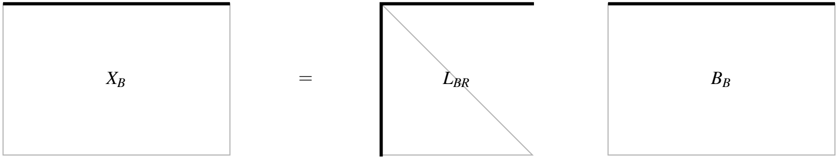


$$P_{\text{pre}}: \quad \{\, B = \hat{B} \quad \wedge \quad L = \hat{L} \quad \wedge \quad \mathit{LowerTriangular}(L) \,\}$$

$$P_{\text{post}}: \quad \{\, X := L^{-1}\hat{B} \quad \wedge \quad B := X \,\}$$

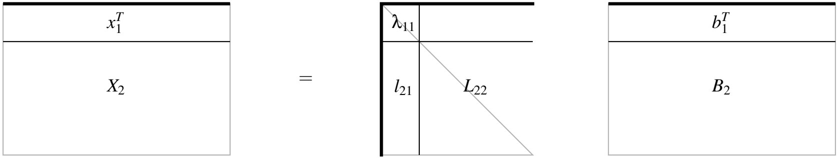
Algorithm

Triangular system with multiple right-hand sides



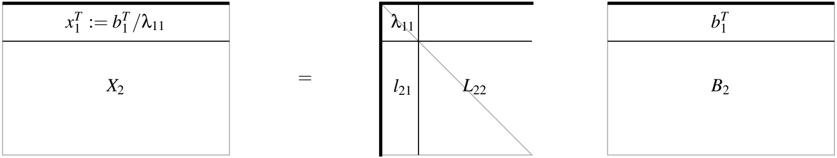
Algorithm

Triangular system with multiple right-hand sides



Algorithm

Triangular system with multiple right-hand sides



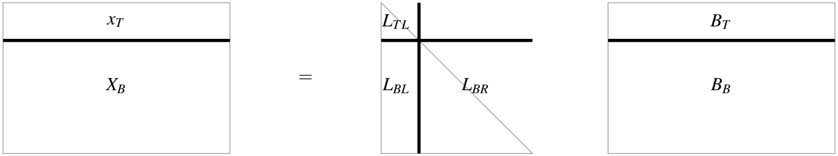
Algorithm

Triangular system with multiple right-hand sides

$$\left(\begin{array}{c} x_1^T := b_1^T / \lambda_{11} \\ \hline X_2 := X_2 - l_{21} x_1^T \end{array}\right) = \left(\begin{array}{c|c} \lambda_{11} & \\ \hline l_{21} & L_{22} \end{array}\right) \left(\begin{array}{c} b_1^T \\ \hline B_2 \end{array}\right)$$

Algorithm

Triangular system with multiple right-hand sides



Algorithm

Triangular system with multiple right-hand sides

$$\left(\begin{array}{c} X_0 \\ \hline x_1^T \\ \hline X_2 \end{array}\right) = \left(\begin{array}{c|c|c} L_{00} & & \\ \hline l_{10}^T & \lambda_{11} & \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right) \left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)$$

Algorithm

Triangular system with multiple right-hand sides

$$
\left(\begin{array}{c} X_0 \\ \hline x_1^T := b_1^T / \lambda_{11} \\ \hline X_2 \end{array}\right)
=
\left(\begin{array}{c|c|c} L_{00} & & \\ \hline l_{10}^T & \lambda_{11} & \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right)
\left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)
$$

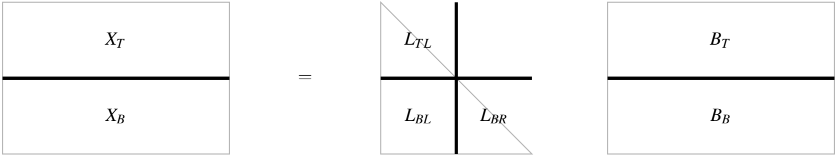
Algorithm

Triangular system with multiple right-hand sides

$$\left(\begin{array}{c} X_0 \\ \hline x_1^T := b_1^T / \lambda_{11} \\ \hline X_2 := X_2 - l_{21} x_1^T \end{array}\right) = \left(\begin{array}{c|c|c} L_{00} & & \\ \hline l_{10}^T & \lambda_{11} & \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right) \left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)$$

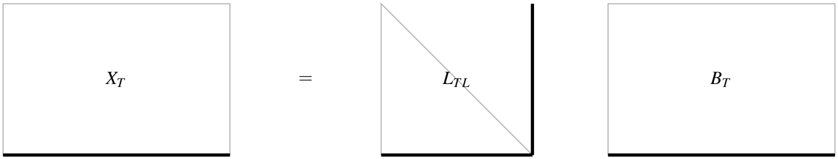
Algorithm

Triangular system with multiple right-hand sides



Algorithm

Triangular system with multiple right-hand sides



FLAME algorithm

Partition $L \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right), B \rightarrow \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right)$

where L_{TL} is 0×0, and B_T is $0 \times n$

while $\texttt{size}(B_T) < \texttt{size}(B)$ **do**

Repartition

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right), \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right) \rightarrow \left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)$$

where λ_{11} is 1×1, and b_1^T is $1 \times n$

$b_1^T := b_1^T / \lambda_{11}$

$B_2 := B_2 - l_{21} b_1^T$

Continue

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right), \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right) \leftarrow \left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)$$

endwhile

- Captures the mathematical notation
- Farewell to indices

Outline

1. FLAME: Notation and methodology
   - Methodology: Correct by construction
2. Cl1ck: Automating the FLAME methodology
3. Conclusions

FLAME methodology

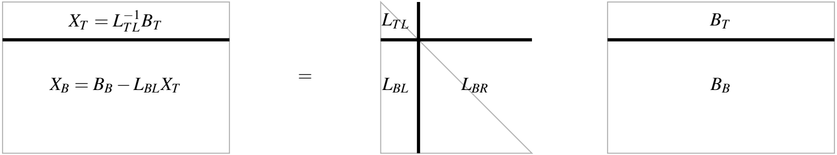
We need **a priori**:

- Precondition (P_{pre})
- Postcondition (P_{post})
- Loop invariant

Loop invariant expresses the state of the computation at certain points of loop:

- Before the first iteration
- At the beginning and end of each iteration
- After the last iteration

Loop invariant



Loop invariant

$$\left(\begin{array}{c} X_0 \\ \hline x_1^T \\ \hline X_2 \end{array}\right) = \left(\begin{array}{c|c|c} L_{00} & & \\ \hline l_{10}^T & \lambda_{11} & \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right) \left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)$$

Loop invariant

$$
\left(\begin{array}{c} X_0 \\ \hline x_1^T := b_1^T / \lambda_{11} \\ \hline X_2 := X_2 - l_{21} x_1^T \end{array}\right)
=
\left(\begin{array}{c|c|c} L_{00} & & \\ \hline l_{10}^T & \lambda_{11} & \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right)
\left(\begin{array}{c} B_0 \\ \hline b_1^T \\ \hline B_2 \end{array}\right)
$$

Loop invariant

$$\left(\begin{array}{c} X_T = L_{TL}^{-1} B_T \\ \hline X_B = B_B - L_{BL} X_T \end{array}\right) = \left(\begin{array}{c|c} L_{TL} & \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c} B_T \\ \hline B_B \end{array}\right)$$

Loop invariant

$$X_T := L_{TL}^{-1} B_T$$

$$X_B := B_B - L_{BL} X_T$$

The worksheet

Step	Skeleton
1a	$\{P_{\text{pre}}\}$
4	`Init`
2	$\{P_{\text{inv}}\}$
3	`While` G `do`
2	$\{P_{\text{inv}}\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{P_{\text{inv}}\}$
3	`end`
2,3	$\{P_{\text{inv}} \wedge \neg G\}$
1b	$\{P_{\text{post}}\}$

The worksheet

Step	Skeleton
1a	$\{P_{\text{pre}}\}$
4	`Init`
2	$\{P_{\text{inv}}\}$
3	`While` G `do`
2	$\{P_{\text{inv}}\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{P_{\text{inv}}\}$
3	`end`
2,3	$\{P_{\text{inv}} \wedge \neg G\}$
1b	$\{P_{\text{post}}\}$

Step	Skeleton
1a	$\{L = \hat{L} \wedge B = \hat{B} \wedge LowTri(L)\}$
4	`Init`
2	$\{P_{\text{inv}}\}$
3	`While` G `do`
2	$\{P_{\text{inv}}\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{P_{\text{inv}}\}$
3	`end`
2,3	$\{P_{\text{inv}} \wedge \neg G\}$
1b	$\{X := L^{-1}B \;\wedge\; B := X\}$

The worksheet

Step	Skeleton
1a	$\{P_{\text{pre}}\}$
4	`Init`
2	$\{P_{\text{inv}}\}$
3	`While` G `do`
2	$\{P_{\text{inv}}\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{P_{\text{inv}}\}$
3	`end`
2,3	$\{P_{\text{inv}} \wedge \neg G\}$
1b	$\{P_{\text{post}}\}$

Step	Skeleton
1a	$\{L = \hat{L} \wedge B = \hat{B} \wedge LowTri(L)\}$
4	`Init`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	`While` G `do`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	`end`
2,3	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T \wedge \neg G\}$
1b	$\{X := L^{-1} B \wedge B := X\}$

The worksheet

Step	Skeleton
1a	$\{L = \hat{L} \wedge B = \hat{B} \wedge LowTri(L)\}$
4	`Init`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	`While` $X_T < X$ `do`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	`end`
2,3	$\{ X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T \wedge \neg (X_T < X) \}$
1b	$\{X := L^{-1} B \wedge B := X\}$

The worksheet

Step	Skeleton
1a	$\{L = \hat{L} \wedge B = \hat{B} \wedge LowTri(L)\}$
4	$X \rightarrow \left(\frac{X_T}{X_B}\right)$ `where` X_T `is` $0 \times n$`, ...`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	`While` $X_T < X$ `do`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	`end`
2,3	$\{\ X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\ \wedge \neg (X_T < X)\ \}$
1b	$\{X := L^{-1} B\ \wedge\ B := X\}$

The worksheet

Step	Skeleton
1a	$\{L = \hat{L} \wedge B = \hat{B} \wedge LowTri(L)\}$
4	$X \rightarrow \left(\frac{X_T}{X_B}\right)$ where X_T is $0 \times n$, ...
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	While $X_T < X$ do
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
5a	$\left(\frac{X_T}{X_B}\right) \rightarrow \left(\frac{X_0}{\frac{x_1^T}{X_2}}\right)$ where x_1^T is $1 \times n$, ...
6	$\{P_{\text{before}}\}$
8	Updates
7	$\{P_{\text{after}}\}$
5b	$\left(\frac{X_T}{X_B}\right) \leftarrow \left(\frac{\frac{X_0}{x_1^T}}{X_2}\right)$
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	end
2,3	$\{ X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T \wedge \neg (X_T < X) \}$
1b	$\{X := L^{-1} B \wedge B := X\}$

Step	Skeleton
1a	$\{L = \hat{L} \wedge B = \hat{B} \wedge LowTri(L)\}$
4	$X \rightarrow \left(\frac{X_T}{X_B}\right)$ where X_T is $0 \times n$, ...
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	While $X_T < X$ do
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
5a	$\left(\frac{X_T}{X_B}\right) \rightarrow \left(\begin{array}{c} X_0 \\ \hline x_1^T \\ \hline X_2 \end{array}\right)$ where x_1^T is $1 \times n$, ...
6	$\{X_0 = L_{00}^{-1} B_0 \wedge x_1^T = b_1^T - l_{10}^T X_0 \wedge X_2 = B_2 - L_{20} X_0\}$
8	Updates
7	$\{X_0 = L_{00}^{-1} B_0 \wedge x_1^T = (b_1^T - l_{10}^T X_0)/\lambda_{11} \wedge X_2 = B_2 - L_{20} X_0 - l_{21} x_1^T\}$
5b	$\left(\frac{X_T}{X_B}\right) \leftarrow \left(\begin{array}{c} X_0 \\ \hline x_1 \\ \hline X_2 \end{array}\right)$
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	end
2,3	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T \wedge X_T < X\}$
1b	$\{X := L^{-1} B \ \wedge \ B := X\}$

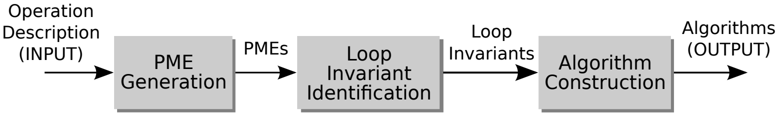
Step	Skeleton
1a	$\{L = \hat{L} \wedge B = \hat{B} \wedge LowTri(L)\}$
4	$X \rightarrow \left(\frac{X_T}{X_B}\right)$ where X_T is $0 \times n$, ...
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	While $X_T < X$ do
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
5a	$\left(\frac{X_T}{X_B}\right) \rightarrow \left(\begin{array}{c} X_0 \\ \hline x_1^T \\ \hline X_2 \end{array}\right)$ where x_1^T is $1 \times n$, ...
6	$\{X_0 = L_{00}^{-1} B_0 \wedge x_1^T = b_1^T - l_{10}^T X_0 \wedge X_2 = B_2 - L_{20} X_0\}$
8	$x_1^T := x_1^T / \lambda_{11} \quad \wedge \quad X_2 := X_2 - l_{21} x_1^T$
7	$\{X_0 = L_{00}^{-1} B_0 \wedge x_1^T = (b_1^T - l_{10}^T X_0)/\lambda_{11} \wedge X_2 = B_2 - L_{20} X_0 - l_{21} x_1^T\}$
5b	$\left(\frac{X_T}{X_B}\right) \leftarrow \left(\begin{array}{c} X_0 \\ \hline x_1 \\ \hline X_2 \end{array}\right)$
2	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T\}$
3	end
2,3	$\{X_T = L_{TL}^{-1} B_T \wedge X_B = B_B - L_{BL} X_T \wedge X_T < X\}$
1b	$\{X := L^{-1} B \ \wedge \ B := X\}$

Outline

1. FLAME: Notation and methodology
2. Cl1ck: Automating the FLAME methodology
3. Conclusions

3 Step generation of algorithms

- Cl1ck implements the FLAME methodology in 3 stages:



Input - Description of target operations

Operations are described by means of two predicates:

The *Precondition* (P_{pre}) and the *Postcondition* (P_{post}).

Example: Triangular Continuous-time Sylvester Equation

$$X := \Omega(L, U, C) \equiv \begin{cases} P_{\text{pre}} : \{\texttt{Input}(L) \wedge \texttt{Matrix}(L) \wedge \texttt{LowerTriangular}(L) \wedge \\ \qquad \texttt{Input}(U) \wedge \texttt{Matrix}(U) \wedge \texttt{UpperTriangular}(U) \wedge \\ \qquad \texttt{Input}(C) \wedge \texttt{Matrix}(C) \wedge \texttt{Output}(X) \wedge \texttt{Matrix}(X)\} \\ \\ P_{\text{post}} : \{LX + XU = C\}. \end{cases}$$

What's behind it

Built-in knowledge I

- Operators: $+$, $-$, $\times$, $^{-1}$, ^{T}
- Basic algebra:

$$\left(\frac{A_{TL}x_T + A_{TR}x_B}{A_{BL}x_T + A_{BR}x_B}\right) \longleftarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \times \left(\frac{x_T}{x_B}\right)$$

What's behind it

Built-in knowledge II

- Properties: triangular, symmetric, diagonal, ...
- Inheritance of properties:
  - $\mathcal{P}(L)_{2\times 2} \wedge LowerTriangular(L) \longrightarrow$
    $$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \wedge LowerTriangular(L_{TL}, L_{BR})$$
  - $\mathcal{P}(S)_{2\times 2} \wedge Symmetric(S) \longrightarrow$
    $$\left(\begin{array}{c|c} S_{TL} & S_{TR} \\ \hline S_{BL} & S_{BR} \end{array}\right) \wedge Symmetric(S_{TL}, S_{BR}) \wedge S_{BL} == S_{TR}^{T}$$

What's behind it

Built-in knowledge III

- Inference rules for each property:
  - $L = L_1 \times L_2 \wedge LowerTriangular(L_1, L_2) \rightarrow LowerTriangular(L)$
  - $L = L_1 + L_2 \wedge LowerTriangular(L_1, L_2) \rightarrow LowerTriangular(L)$
  - $S = S_1 + S_2 \wedge Symmetric(S_1, S_2) \rightarrow Symmetric(S)$
- Simplications:
  - $A^{-1} \times A \longrightarrow I$
  - $A^T \wedge Symmetric(A) \longrightarrow A$
  - $(A^T)^T \longrightarrow A$

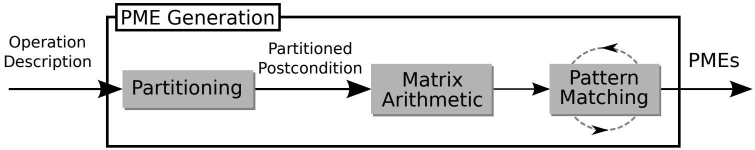
What's behind it

Learning

- How does a system recognize a recursive call to an operation that it is not initially known?
- CL1CK must incorporate a mechanism to extend its knowledge-base dynamically
- For each target operation, CL1CK creates a corresponding pattern:

$$LX + XU = C \wedge \mathit{Input}(L, U, C) \wedge \mathit{Output}(X) \wedge \\ \mathit{LowerTriangular}(L) \wedge \mathit{UpperTriangular}(U) \longrightarrow X = \Omega(L, U, C)$$

PME generation



PME generation

Partitionings

- We consider the partitionings 1×1, 1×2, 2×1 and 2×2
- Not every combination of partitionings is valid
- Constraints: Operators and properties

PME generation

Partitionings

- We consider the partitionings 1×1, 1×2, 2×1 and 2×2
- Not every combination of partitionings is valid
- Constraints: Operators and properties
- Naive approach: consider the $4^{\#operands}$ combinations

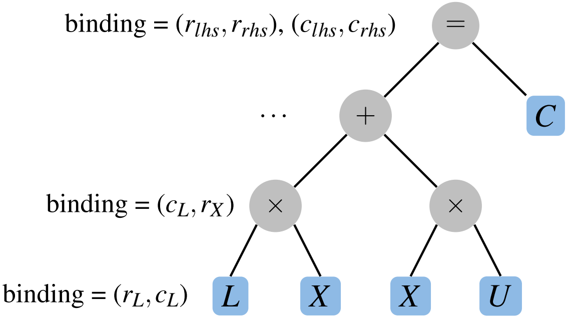
PME generation

Partitionings

- We consider the partitionings 1×1, 1×2, 2×1 and 2×2
- Not every combination of partitionings is valid
- Constraints: Operators and properties
- Naive approach: consider the $4^{\#operands}$ combinations
- CL1CK's approach: Bottom-up traversal of the expression tree (using an S-attributed grammar)

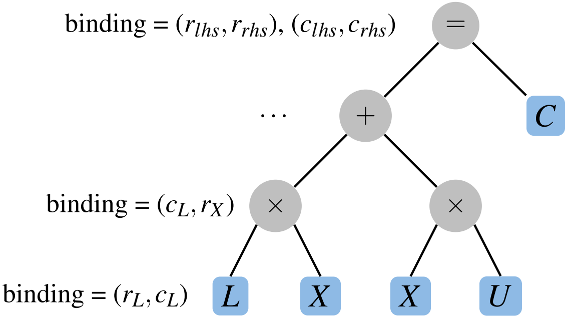
PME generation

Partitionings



PME generation

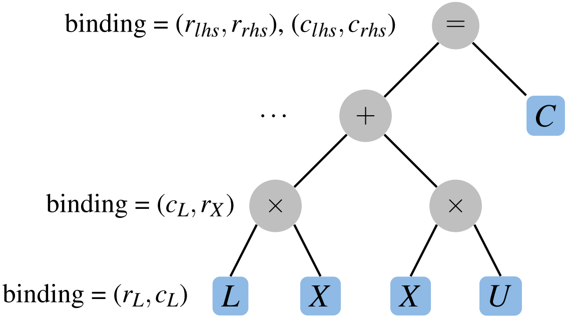
Partitionings



- $\mathcal{B} = \{\{r_L, c_L, r_X, r_C\}, \{r_U, c_U, c_X, r_X\}\}$

PME generation

Partitionings



- $\mathcal{B} = \{\{r_L, c_L, r_X, r_C\}, \{r_U, c_U, c_X, r_X\}\}$
- $g = |\mathcal{B}| \longrightarrow 2^g - 1$ valid combinations

PME generation

Partitionings

- $$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \times \left(\begin{array}{c} X_T \\ \hline X_B \end{array}\right) + \left(\begin{array}{c} X_T \\ \hline X_B \end{array}\right) \times (U) = \left(\begin{array}{c} C_T \\ \hline C_B \end{array}\right)$$
- $$(L) \times \left(\begin{array}{c|c} X_L & X_R \end{array}\right) + \left(\begin{array}{c|c} X_L & X_R \end{array}\right) \times \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) = \left(\begin{array}{c|c} C_L & C_R \end{array}\right)$$
- $$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \times \left(\begin{array}{c|c} X_{TL} & X_{TR} \\ \hline X_{BL} & X_{BR} \end{array}\right) + \left(\begin{array}{c|c} X_{TL} & X_{TR} \\ \hline X_{BL} & X_{BR} \end{array}\right) \times \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) = \left(\begin{array}{c|c} C_{TL} & C_{TR} \\ \hline C_{BL} & C_{BR} \end{array}\right)$$

PME generation

Find recursive definition

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \times \left(\frac{X_T}{X_B}\right) + \left(\frac{X_T}{X_B}\right) \times (U) = \left(\frac{C_T}{C_B}\right)$$

$$\downarrow$$

$$\left(\frac{L_{TL}X_T + X_T U = C_T}{L_{BL}X_T + L_{BR}X_B + X_B U = C_B}\right)$$

PME generation

Find recursive definition

$$\left(\frac{L_{TL}X_T + X_T U = C_T}{L_{BL}X_T + L_{BR}X_B + X_B U = C_B}\right)$$

$$\downarrow$$

$$\left(\frac{X_T = \Omega(L_{TL}, U, C_T)}{L_{BL}X_T + L_{BR}X_B + X_B U = C_B}\right)$$

PME generation

Find recursive definition

$$\left(\frac{X_T = \Omega(L_{TL}, U, C_T)}{L_{BL}X_T + L_{BR}X_B + X_BU = C_B}\right)$$

$$\downarrow$$

$$\left(\frac{X_T = \Omega(L_{TL}, U, C_T)}{L_{BR}X_B + X_BU = C_B - L_{BL}X_T}\right)$$

PME generation

Find recursive definition

$$\left(\frac{X_T = \Omega(L_{TL}, U, C_T)}{L_{BR}X_B + X_B U = C_B - L_{BL}X_T}\right)$$

$$\downarrow$$

$$\left(\frac{X_T = \Omega(L_{TL}, U, C_T)}{X_B = \Omega(L_{BR}, U, C_B - L_{BL}X_T)}\right)$$

PME generation

1 $$\left(\frac{X_T = \Omega(L_{TL}, U, C_T)}{X_B = \Omega(L_{BR}, U, C_B - L_{BL} \times X_T)}\right)$$

2 $$\left(\; X_L = \Omega(L, U_{TL}, C_L) \;\Big|\; X_R = \Omega(L, U_{BR}, C_R - X_L \times U_{TR}) \; \right)$$

3 $$\left(\begin{array}{c|r} X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) & X_{TR} := \Omega(L_{TL}, U_{BR}, C_{TR} - X_{TL}U_{TR}) \\ \hline X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) & \begin{array}{l} X_{BR} := \Omega(L_{BR}, U_{BR}, \\ \qquad C_{BR} - X_{BL}U_{TR} - L_{BL}X_{TR}) \end{array} \end{array}\right)$$

PME generation

Triangular Sylvester Equation ($LX + XU = C$):

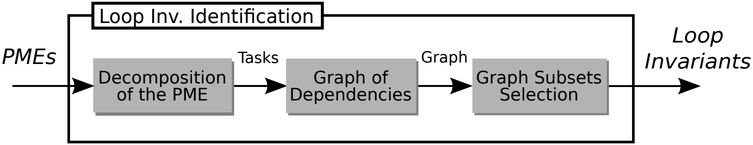
$$\left(\begin{array}{c|c} X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) & X_{TR} := \Omega(L_{TL}, U_{BR}, C_{TR} - X_{TL}U_{TR}) \\ \hline X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) & \begin{array}{l} X_{BR} := \Omega(L_{BR}, U_{BR}, \\ \qquad C_{BR} - X_{BL}U_{TR} - L_{BL}X_{TR}) \end{array} \end{array}\right)$$

Triangular Lyapunov Equation ($LX + XL^T = C$):

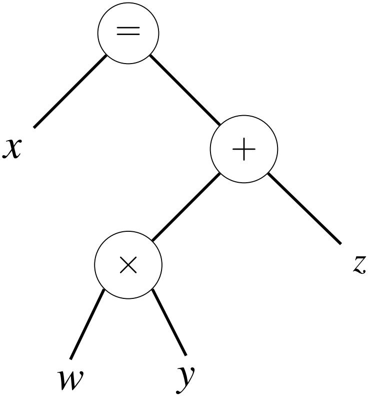
$$\left(\begin{array}{c|c} X_{TL} := \Lambda(L_{TL}, C_{TL}) & * \\ \hline X_{BL} := \Omega(L_{BR}, L_{TL}^T, C_{BL} - L_{BL}X_{TL}) & \begin{array}{l} X_{BR} := \Lambda(L_{BR}, \\ \qquad C_{BR} - X_{BL}L_{BL}^T - L_{BL}X_{BL}^T) \end{array} \end{array}\right)$$

Loop invariants

Loop invariants: subsets of the computation expressed in the PME



Tree tiling



ISA:

- load
- store
- fadd
- fmul
- …

What's behind it - IV

Building blocks

Completeness:

- $A + B$
- $A \times B$
- $-A$
- A^{-1}
- A^T
- $f(a_0, a_1, \ldots, a_n)$

What's behind it - IV

Building blocks

Completeness:

- $A + B$
- $A \times B$
- $-A$
- A^{-1}
- A^T
- $f(a_0, a_1, \ldots, a_n)$

Extended set:

- $A \times B + C$
- $A^T \times B + C$
- $A \times B^T + C$
- $A^T \times B^T$

- $L^{-1} \times B$
- $L^{-T} \times B$
- $B \times L^{-1}$
- $B \times U^{-1}$

Loop invariants

Decomposition of the PME

Example: 2×2 Sylvester.

$$\left(\begin{array}{c|c} X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) & X_{TR} := \Omega(L_{TL}, U_{BR}, C_{TR} - X_{TL}U_{TR}) \\ \hline X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) & \begin{array}{l} X_{BR} := \Omega(L_{BR}, U_{BR}, \\ \qquad C_{BR} - X_{BL}U_{TR} - L_{BL}X_{TR}) \end{array} \end{array}\right)$$

Loop invariants

Decomposition of the PME

Example: 2×2 Sylvester.

$$\left(\begin{array}{c|c} X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) & X_{TR} := \Omega(L_{TL}, U_{BR}, C_{TR} - X_{TL}U_{TR}) \\ \hline X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) & \begin{array}{l} X_{BR} := \Omega(L_{BR}, U_{BR}, \\ \qquad\qquad C_{BR} - X_{BL}U_{TR} - L_{BL}X_{TR}) \end{array} \end{array}\right)$$

1. $X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL})$

Loop invariants

Decomposition of the PME

Example: 2×2 Sylvester.

$$\left(\begin{array}{c|c} X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) & X_{TR} := \Omega(L_{TL}, U_{BR}, C_{TR} - X_{TL}U_{TR}) \\ \hline X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) & \begin{array}{l} X_{BR} := \Omega(L_{BR}, U_{BR}, \\ \qquad C_{BR} - X_{BL}U_{TR} - L_{BL}X_{TR}) \end{array} \end{array}\right)$$

1. $X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL})$
2. $T_1 := C_{TR} - X_{TL}U_{TR}$
3. $X_{TR} := \Omega(L_{TL}, U_{BR}, T_1)$

Loop invariants

Decomposition of the PME

Example: 2×2 Sylvester.

$$\left(\begin{array}{c|c} X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) & X_{TR} := \Omega(L_{TL}, U_{BR}, C_{TR} - X_{TL}U_{TR}) \\ \hline X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) & \begin{array}{l} X_{BR} := \Omega(L_{BR}, U_{BR}, \\ \qquad C_{BR} - X_{BL}U_{TR} - L_{BL}X_{TR}) \end{array} \end{array}\right)$$

1. $X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL})$
2. $T_1 := C_{TR} - X_{TL}U_{TR}$
3. $X_{TR} := \Omega(L_{TL}, U_{BR}, T_1)$
4. $T_2 := C_{BL} - L_{BL}X_{TL}$
5. $X_{BL} := \Omega(L_{BR}, U_{TL}, T_2)$

Loop invariants

Decomposition of the PME

Example: 2×2 Sylvester.

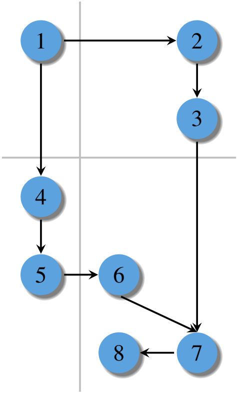
$$\left(\begin{array}{c|c} X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) & X_{TR} := \Omega(L_{TL}, U_{BR}, C_{TR} - X_{TL}U_{TR}) \\ \hline X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) & \begin{array}{l} X_{BR} := \Omega(L_{BR}, U_{BR}, \\ \qquad C_{BR} - X_{BL}U_{TR} - L_{BL}X_{TR}) \end{array} \end{array}\right)$$

1. $X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL})$
2. $T_1 := C_{TR} - X_{TL}U_{TR}$
3. $X_{TR} := \Omega(L_{TL}, U_{BR}, T_1)$
4. $T_2 := C_{BL} - L_{BL}X_{TL}$
5. $X_{BL} := \Omega(L_{BR}, U_{TL}, T_2)$
6. $T_3 := C_{BR} - X_{BL}U_{TR}$
7. $T_4 := T_3 - L_{BL}X_{TR}$
8. $X_{BR} := \Omega(L_{BR}, U_{BR}, T_4)$

Loop invariants

Graph of dependencies

1. $X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL})$
2. $T_1 := C_{TR} - X_{TL} U_{TR}$
3. $X_{TR} := \Omega(L_{TL}, U_{BR}, T_1)$
4. $T_2 := C_{BL} - L_{BL} X_{TL}$
5. $X_{BL} := \Omega(L_{BR}, U_{TL}, T_2)$
6. $T_3 := C_{BR} - X_{BL} U_{TR}$
7. $T_4 := T_3 - L_{BL} X_{TR}$
8. $X_{BR} := \Omega(L_{BR}, U_{BR}, T_4)$

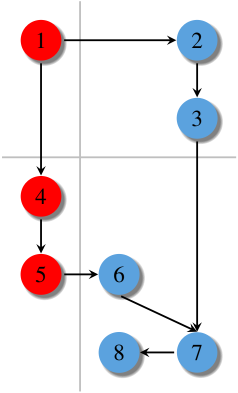


Loop invariants

Subgraph selection

1. $X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL})$
4. $T_2 := C_{BL} - L_{BL}X_{TL}$
5. $X_{BL} := \Omega(L_{BR}, B_{TL}, T_2)$

$$\left(\begin{array}{c|c} X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) & \neq \\ \hline X_{BL} := \Omega(L_{BR}, B_{TL}, C_{BL} - L_{BL}X_{TL}) & \neq \end{array}\right)$$



Generation of algorithms

Step	Skeleton
1a	$\{P_{\text{pre}}\}$
4	`Init`
2	$\{P_{\text{inv}}\}$
3	`While` G `do`
2	$\{P_{\text{inv}}\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{P_{\text{inv}}\}$
3	`end`
2,3	$\{P_{\text{inv}} \wedge \neg G\}$
1b	$\{P_{\text{post}}\}$

- We have:

$$P_{\text{pre}}: \quad \{\, \texttt{Input}(L, U, C) \wedge \texttt{Matrix}(L, U, C) \wedge \texttt{LowerTriangular}(L) \wedge \texttt{UpperTriangular}(U) \wedge \texttt{Output}(X) \wedge \texttt{Matrix}(X) \,\}$$

$$P_{\text{post}}: \quad \{LX + XU = C\}.$$

$$L_{\text{inv}}: \quad X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) \wedge X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL})$$

Generation of algorithms

Step	Skeleton
1a	$\{P_{\text{pre}}\}$
4	`Init`
2	$\{P_{\text{inv}}\}$
3	`While` G `do`
2	$\{P_{\text{inv}}\}$
5a	`Repartition`
6	$\{P_{\text{before}}\}$
8	`Updates`
7	$\{P_{\text{after}}\}$
5b	`Continue`
2	$\{P_{\text{inv}}\}$
3	`end`
2,3	$\{P_{\text{inv}} \wedge \neg G\}$
1b	$\{P_{\text{post}}\}$

- We can determine: $G \rightarrow X_{TL} < X \;\; (s.t. \neg G \rightarrow X_{TL} = X)$
- and thus:

$$X \rightarrow \left(\begin{array}{c|c} X_{TL} & X_{TR} \\ \hline X_{BL} & X_{BR} \end{array}\right) \quad \text{(Init)}$$

$$\rightarrow \left(\begin{array}{c|c|c} X_{00} & X_{01} & X_{02} \\ \hline X_{10} & X_{11} & X_{12} \\ \hline X_{20} & X_{21} & X_{22} \end{array}\right) \quad \text{(Repartition)}$$

$$\rightarrow \left(\begin{array}{c|c|c} X_{00} & X_{01} & X_{02} \\ \hline X_{10} & X_{11} & X_{12} \\ \hline X_{20} & X_{21} & X_{22} \end{array}\right) \quad \text{(Continue)}$$

Generation of algorithms

$$L_{\text{inv}}: \quad X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) \;\wedge\; X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL})$$

$\Downarrow_{Repartition}$

$$L_{\text{inv}}: \quad (X_{00}) := \Omega((L_{00}), (U_{00}), (C_{00})) \;\wedge$$

$$\left(\begin{array}{c} X_{10} \\ \hline X_{20} \end{array}\right) := \Omega(\left(\begin{array}{c|c} L_{11} & 0 \\ \hline L_{21} & L_{22} \end{array}\right), (U_{00}), \left(\begin{array}{c} C_{10} \\ \hline C_{20} \end{array}\right) - \left(\begin{array}{c} L_{10} \\ \hline L_{20} \end{array}\right) X_{00})$$

Generation of algorithms

$$L_{\text{inv}}: \quad \begin{aligned} &X_{TL} := \Omega(L_{TL}, U_{TL}, C_{TL}) \ \wedge \\ &X_{BL} := \Omega(L_{BR}, U_{TL}, C_{BL} - L_{BL}X_{TL}) \end{aligned}$$

$\Downarrow$*Repartition*

$$L_{\text{inv}}: \quad (X_{00}) := \Omega((L_{00}), (U_{00}), (C_{00})) \ \wedge$$

$$\left(\begin{array}{c} X_{10} \\ \hline X_{20} \end{array}\right) := \Omega(\left(\begin{array}{c|c} L_{11} & 0 \\ \hline L_{21} & L_{22} \end{array}\right), (U_{00}), \left(\begin{array}{c} C_{10} \\ \hline C_{20} \end{array}\right) - \left(\begin{array}{c} L_{10} \\ \hline L_{20} \end{array}\right) X_{00})$$

$\Downarrow$*Flatten*

$$L_{\text{inv}}: \quad \begin{aligned} &X_{00} := \Omega(L_{00}, U_{00}, C_{00}) \ \wedge \\ &X_{10} := \Omega(L_{11}, U_{00}, C_{10} - L_{10}X_{00}) \ \wedge \\ &X_{20} := \Omega(L_{22}, U_{00}, C_{20} - L_{20}X_{00} - L_{21}X_{10}) \end{aligned}$$

Generation of algorithms

$$
\begin{aligned}
P_{\text{before}}: \quad & X_{00} := \Omega(L_{00}, U_{00}, C_{00}) \;\wedge \\
& X_{10} := \Omega(L_{11}, U_{00}, C_{10} - L_{10}X_{00}) \;\wedge \\
& X_{20} := \Omega(L_{22}, U_{00}, C_{11} - L_{20}X_{00} - L_{21}X_{10})
\end{aligned}
$$

$$
\begin{aligned}
P_{\text{after}}: \quad & X_{00} := \Omega(L_{00}, U_{00}, C_{00}) \;\wedge \\
& X_{10} := \Omega(L_{11}, U_{00}, C_{10} - L_{10}X_{00}) \;\wedge \\
& X_{20} := \Omega(L_{22}, U_{00}, C_{20} - L_{20}X_{00} - L_{21}X_{10}) \\
& X_{01} := \Omega(L_{00}, U_{11}, C_{01} - X_{00}U_{01}) \;\wedge \\
& X_{11} := \Omega(L_{11}, U_{11}, C_{11} - L_{10}X_{01} - X_{10}U_{01}) \;\wedge \\
& X_{21} := \Omega(L_{22}, U_{11}, C_{21} - L_{20}X_{01} - L_{21}X_{11} - X_{20}U_{01})
\end{aligned}
$$

Examples of code

- Too long. See file m10.c

Outline

Conclusions

- FLAME
  - Notation close to “blackboard” algorithm
  - Systematic methodology to generate correct algorithms
- Cl1ck
  - Automates the generation in 3 stages
  - From description to (PMEs to loop invariants to) algorithms
  - Requires
    - Linear algebra built-in knowledge
    - Learning of operations and PMEs
- Other domains?

Thank you for your attention!

FLAME:

- The science of deriving dense linear algebra algorithms. *Paolo Bientinesi et al*. ACM TOMS.
- Representing Dense Linear Algebra Algorithms: A Farewell to Indices. *Paolo Bientinesi and Robert van de Geijn*. FLAWN #17

CL1CK

- Knowledge-Based Automatic Generation of Partitioned Matrix Expressions. *Diego Fabregat-Traver and Paolo Bientinesi*. CASC 2011.
- Automatic Generation of Loop-Invariants for Matrix Operations. *Diego Fabregat-Traver and Paolo Bientinesi*. ICCSA 2011.
- Article with the complete FLAME + CL1CK to appear soon.

DFG

Financial support from DFG through Grant BI 1533/2-1 is gratefully acknowledged