# Automatic Generation of Loop-Invariants for Matrix Operations

**Diego Fabregat-Traver** and Paolo Bientinesi

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

Computer Algebra Systems and their Applications, CASA'2011
Santander, June 21st, 2011

# Automatic Generation of Algorithms

# Automatic Generation of Algorithms


$$
\begin{cases}
P_{\text{pre}} : \{\texttt{Unknown}(L) \wedge \texttt{LowTri}(L) \wedge \\
\qquad \texttt{UnitDiag}(L) \wedge \\
\qquad \texttt{Unknown}(U) \wedge \texttt{UppTri}(U) \wedge \\
\qquad \texttt{Known}(A) \wedge \exists\, \texttt{LU}(A)\} \\
\\
P_{\text{post}} : \{LU = A\}
\end{cases}
$$


$$
\begin{cases}
P_{\text{pre}} : \{\texttt{Unknown}(L) \wedge \texttt{LowTri}(L) \wedge \\
\qquad \texttt{UnitDiag}(L) \wedge \\
\qquad \texttt{Unknown}(U) \wedge \texttt{UppTri}(U) \wedge \\
\qquad \texttt{Known}(A) \wedge \exists\, \texttt{LU}(A)\} \\
\\
P_{\text{post}} : \{LU = A\}
\end{cases}
\longrightarrow
$$

**Partition** $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right), \ldots$
**where** $A_{TL}$ **is** $0 \times 0$

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

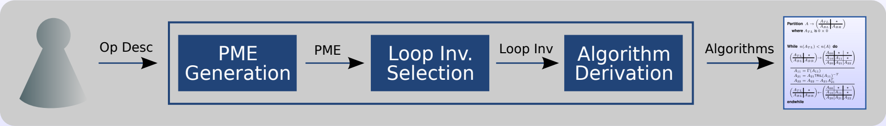
$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right), \ldots$$

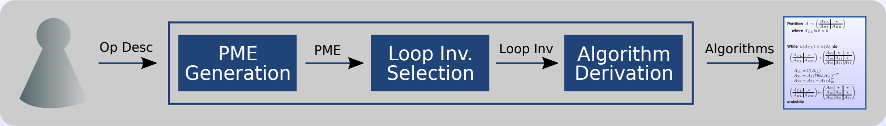
$$U_{01} = L_{00}^{-1} A_{01}$$
$$L_{10} = A_{10} U_{00}^{-1}$$
$$\{L_{11}, U_{11}\} = LU(A_{11} - L_{10} U_{01})$$

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

**endwhile**

# Automatic Generation of Algorithms






$$LU = A \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) = \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

# Automatic Generation of Algorithms


$$LU = A \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) = \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$$

Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

$$\Downarrow$$

Loop Invariant (1):

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & \\ \hline & \end{array}\right)$$

# Loop-Invariant Selection

## Steps


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

# Decomposition of the PME

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

**Tasks:**

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

# Decomposition of the PME

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

**Tasks:**

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```


Partitioned Matrix Expression:

$$\left( \begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array} \right)$$

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

# Decomposition of the PME

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$
2. $U_{TR} := L_{TL}^{-1} A_{TR}$

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$
2. $U_{TR} := L_{TL}^{-1} A_{TR}$

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

AI ces

# Decomposition of the PME

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$
2. $U_{TR} := L_{TL}^{-1} A_{TR}$
3. $L_{BL} := A_{BL} U_{TL}^{-1}$

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

# Decomposition of the PME

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$
2. $U_{TR} := L_{TL}^{-1} A_{TR}$
3. $L_{BL} := A_{BL} U_{TL}^{-1}$

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$
2. $U_{TR} := L_{TL}^{-1} A_{TR}$
3. $L_{BL} := A_{BL} U_{TL}^{-1}$

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```


### Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

### Tasks:

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$
2. $U_{TR} := L_{TL}^{-1} A_{TR}$
3. $L_{BL} := A_{BL} U_{TL}^{-1}$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR}$

### Patterns:

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

# Decomposition of the PME

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \color{red}{\{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR})} \end{array}\right)$$

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$
2. $U_{TR} := L_{TL}^{-1} A_{TR}$
3. $L_{BL} := A_{BL} U_{TL}^{-1}$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR}$

**Patterns:**

```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

# Decomposition of the PME

## An example: LU Factorization


Partitioned Matrix Expression:

$$\left(\begin{array}{c|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \{L_{BR}, U_{BR}\} = LU(A_{BR} - L_{BL} U_{TR}) \end{array}\right)$$

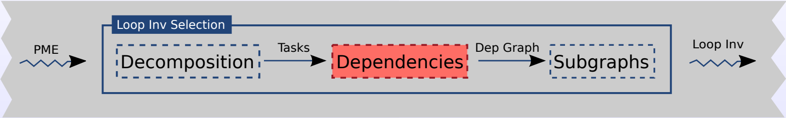
**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL})$
2. $U_{TR} := L_{TL}^{-1} A_{TR}$
3. $L_{BL} := A_{BL} U_{TL}^{-1}$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR}$
5. $\{L_{BR}, U_{BR}\} := LU(A_{BR})$

**Patterns:**

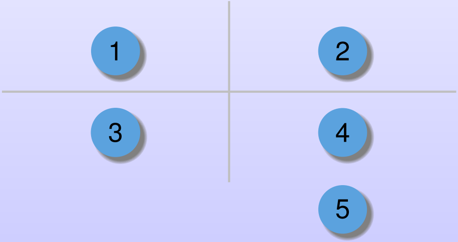
```
f_[A_?isSuboperandQ]
times[inv[A_?isTriangularQ], B_]
times[A_, inv[B_?isTriangularQ]]
plus[A_, minus[times[B_, C_]]
f_[A_?isComplexQ] → Decompose[A]
...
```

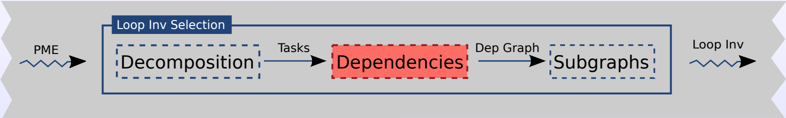
An example: LU Factorization




**Tasks:**

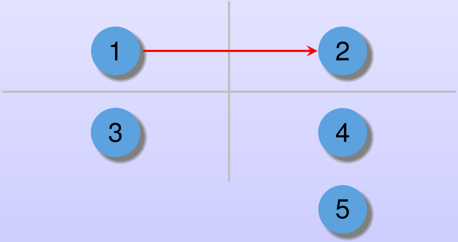
1. $\{L_{TL}, U_{TL}\} := LU(A_{TL});$
2. $U_{TR} := {L_{TL}}^{-1} A_{TR};$
3. $L_{BL} := A_{BL} {U_{TL}}^{-1};$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR};$
5. $\{L_{BR}, U_{BR}\} := LU(A_{BR}).$

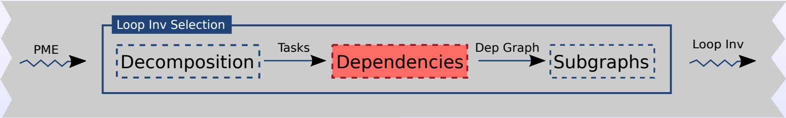





**Tasks:**

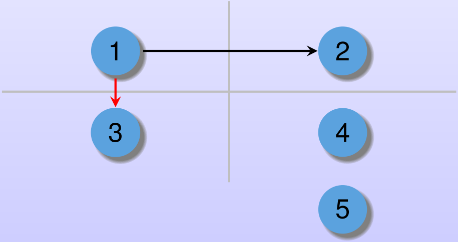
1. $\{L_{TL}, U_{TL}\} := LU(A_{TL});$
2. $U_{TR} := L_{TL}{}^{-1} A_{TR};$
3. $L_{BL} := A_{BL} U_{TL}{}^{-1};$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR};$
5. $\{L_{BR}, U_{BR}\} := LU(A_{BR}).$





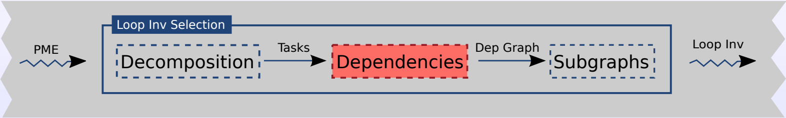

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL});$
2. $U_{TR} := L_{TL}{}^{-1} A_{TR};$
3. $L_{BL} := A_{BL} U_{TL}{}^{-1};$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR};$
5. $\{L_{BR}, U_{BR}\} := LU(A_{BR}).$



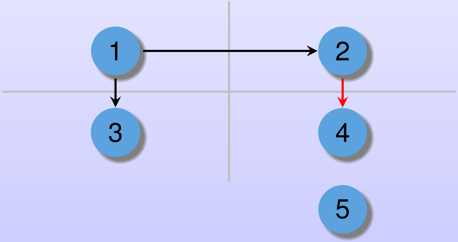
# Graph of Dependencies

## An example: LU Factorization

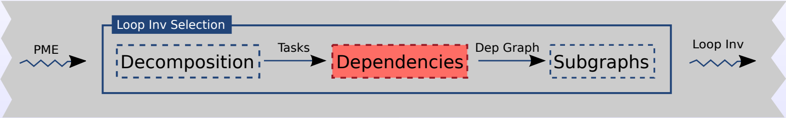



**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL});$
2. $U_{TR} := L_{TL}{}^{-1} A_{TR};$
3. $L_{BL} := A_{BL} U_{TL}{}^{-1};$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR};$
5. $\{L_{BR}, U_{BR}\} := LU(A_{BR}).$

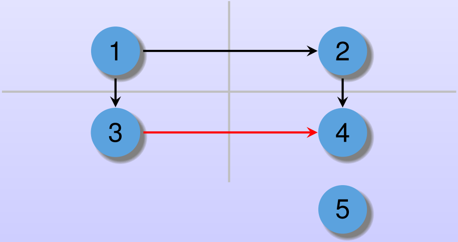


## An example: LU Factorization



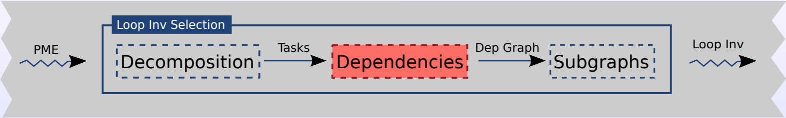

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL});$
2. $U_{TR} := {L_{TL}}^{-1} A_{TR};$
3. $L_{BL} := A_{BL} {U_{TL}}^{-1};$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR};$
5. $\{L_{BR}, U_{BR}\} := LU(A_{BR}).$



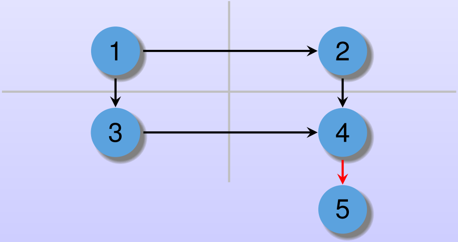
# Graph of Dependencies

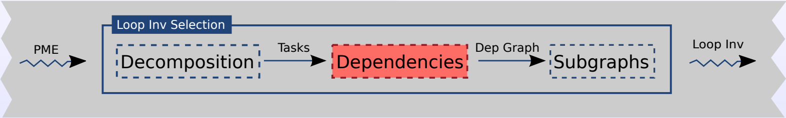
## An example: LU Factorization




**Tasks:**

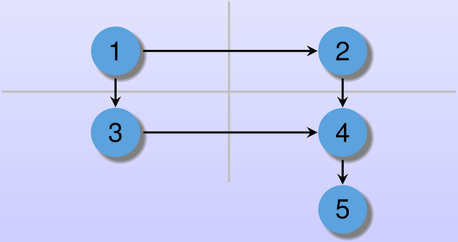
1. $\{L_{TL}, U_{TL}\} := LU(A_{TL});$
2. $U_{TR} := {L_{TL}}^{-1} A_{TR};$
3. $L_{BL} := A_{BL} {U_{TL}}^{-1};$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR};$
5. $\{L_{BR}, U_{BR}\} := LU(A_{BR}).$





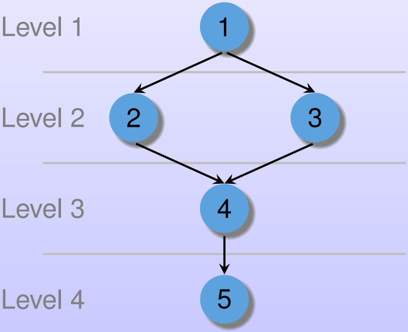

**Tasks:**

1. $\{L_{TL}, U_{TL}\} := LU(A_{TL});$
2. $U_{TR} := L_{TL}{}^{-1} A_{TR};$
3. $L_{BL} := A_{BL} U_{TL}{}^{-1};$
4. $A_{BR} := A_{BR} - L_{BL} U_{TR};$
5. $\{L_{BR}, U_{BR}\} := LU(A_{BR}).$



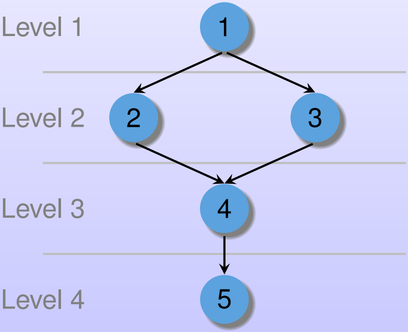
RWTH AACHEN UNIVERSITY

An example: LU Factorization

# Subgraphs Selection

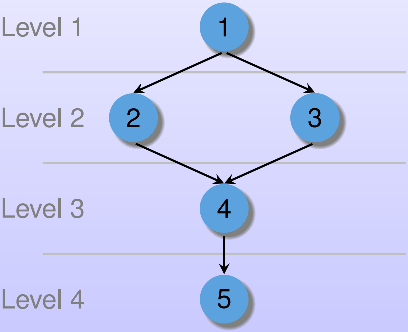
## An example: LU Factorization


- { }

# Subgraphs Selection

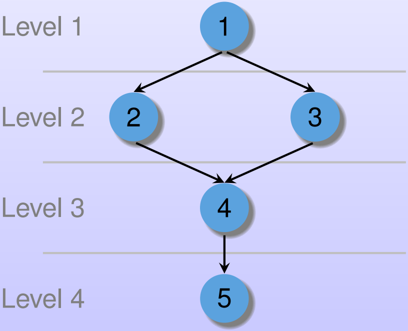
## An example: LU Factorization


- { }
- {1}

# Subgraphs Selection

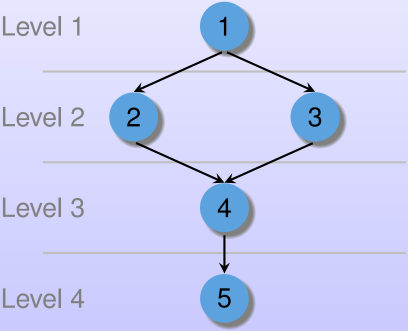
## An example: LU Factorization


- { }
- {1}
- {1, 2}
- {1, 3}
- {1, 2, 3}

# Subgraphs Selection

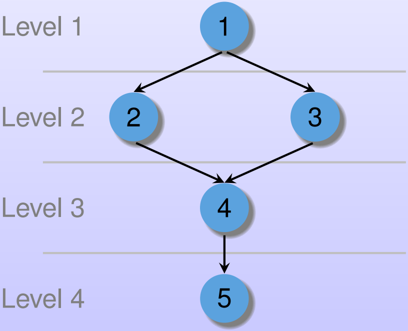
## An example: LU Factorization


- { }
- {1}
- {1, 2}
- {1, 3}
- {1, 2, 3}
- {1, 2, 3, 4}

# Subgraphs Selection

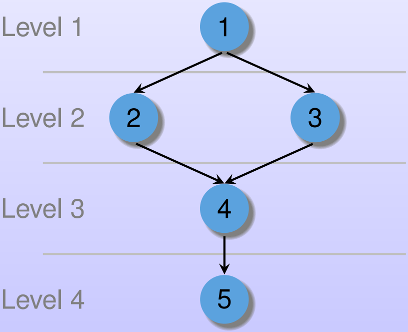
## An example: LU Factorization

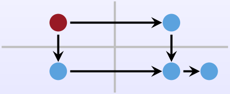

- { }
- {1}
- {1, 2}
- {1, 3}
- {1, 2, 3}
- {1, 2, 3, 4}
- {1, 2, 3, 4, 5}

# Subgraphs Selection

## An example: LU Factorization


- { }
- {1}
- {1, 2}
- {1, 3}
- {1, 2, 3}
- {1, 2, 3, 4}
- {1, 2, 3, 4, 5}

| # | Subgraph | Loop-invariant |
|---|---|---|
| {1} |  | $\left(\begin{array}{c\|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & \neq \\ \hline \neq & \neq \end{array}\right)$ |

# Loop Invariants

## An example: LU Factorization

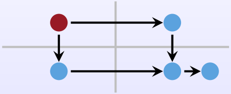
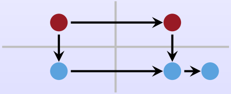
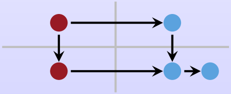
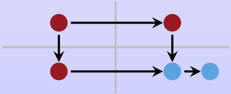
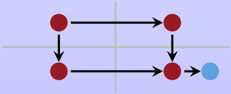
| # | Subgraph | Loop-invariant |
|---|---|---|
| {1} |  | $\left(\begin{array}{c\|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & \neq \\ \hline \neq & \neq \end{array}\right)$ |
| {1,2} |  | $\left(\begin{array}{c\|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline \neq & \neq \end{array}\right)$ |
| {1,3} |  | $\left(\begin{array}{c\|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & \neq \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \neq \end{array}\right)$ |
| {1,2,3} |  | $\left(\begin{array}{c\|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & \neq \end{array}\right)$ |
| {1,2,3,4} |  | $\left(\begin{array}{c\|c} \{L_{TL}, U_{TL}\} = LU(A_{TL}) & U_{TR} = L_{TL}^{-1} A_{TR} \\ \hline L_{BL} = A_{BL} U_{TL}^{-1} & A_{BR} = A_{BR} - L_{BL} U_{TR} \end{array}\right)$ |

Table: The five loop-invariants for the LU factorization.

**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 $0 \times 0$

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

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

**Variant 1**

$$\begin{aligned} U_{01} &= L_{00}^{-1} A_{01} \\ L_{10} &= A_{10} U_{00}^{-1} \\ A_{11} &= A_{11} - L_{10} U_{01} \\ \{L_{11}, U_{11}\} &= LU(A_{11}) \end{aligned}$$

$\ldots$

**Variant 5**

$$\begin{aligned} \{L_{11}, U_{11}\} &= LU(A_{11}) \\ U_{12} &= L_{11}^{-1} A_{12} \\ L_{21} &= A_{21} U_{11}^{-1} \\ A_{22} &= A_{22} - L_{21} U_{12} \end{aligned}$$

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

**endwhile**

* Variant 5 needs the assignment $\{L_{BR}, U_{BR}\} = A_{BR}$ before entering the loop.

# Conclusions

- The system automatically obtains loop-invariants from the PME by:
  - decomposing the operations in the PME into tasks,
  - building a graph of dependencies among tasks, and
  - generating all the possible subsets of the graph of dependencies.

# Conclusions

- The system automatically obtains loop-invariants from the PME by:
  - decomposing the operations in the PME into tasks,
  - building a graph of dependencies among tasks, and
  - generating all the possible subsets of the graph of dependencies.
- The approach is fairly general and applies to direct methods for:
  - matrix products,
  - solution of linear systems,
  - matrix factorizations,
  - equations arising in control theory such as Sylvester and Lyapunov.

# Conclusions

- The system automatically obtains loop-invariants from the PME by:
  - decomposing the operations in the PME into tasks,
  - building a graph of dependencies among tasks, and
  - generating all the possible subsets of the graph of dependencies.
- The approach is fairly general and applies to direct methods for:
  - matrix products,
  - solution of linear systems,
  - matrix factorizations,
  - equations arising in control theory such as Sylvester and Lyapunov.
- **One step closer to the automatic generation of algorithms.**

Thanks to:

- Dr. Edoardo Di Napoli
- Matthias Petschow
- Roman Iakymchuk
- AICES fellows

Financial support from the Deutsche Forschungsgemeinschaft (German Research Association) through grant GSC 111 is gratefully acknowledged.

Deutsche Forschungsgemeinschaft DFG