

# LAMP: the Linear Algebra Mapping Problem

Henrik Barthels, Diego Fabregat, **Paolo Bientinesi**

Aachen Institute for Computational Engineering Science  
RWTH Aachen University

PASC17

June 26, 2017

Lugano, Switzerland

Deutsche  
Forschungsgemeinschaft

**DFG**



High Performance and  
Automatic Computing

**RWTH**AACHEN  
UNIVERSITY

$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y$$

exponential  
transient excision

$$q := u - U(P^T U)^{-1} P^T u$$

reduced basis  
methodology for  
parametric PDEs

$$\begin{cases} C_{\dagger} := P C P^T + Q \\ K := C_{\dagger} H^T (H C_{\dagger} H^T)^{-1} \end{cases}$$

probabilistic  
Nordsieck method  
for ODEs

$$E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T$$

L1-norm  
minimization on  
manifolds

$$\begin{cases} x_{k|k-1} = F x_{k-1|k-1} + B u \\ P_{k|k-1} = F P_{k-1|k-1} F^T + Q \\ x_{k|k} = x_{k|k-1} + P_{k|k-1} H^T \times (H P_{k|k-1} H^T + R)^{-1} (z_k - H x_{k|k-1}) \\ P_{k|k} = P_{k|k-1} - P_{k|k-1} H^T \times (H P_{k|k-1} H^T + R)^{-1} H P_{k|k-1} \end{cases}$$

Kalman filter

$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y$$

exponential  
transient excision

$$q := u - U(P^T U)^{-1} P^T u$$

reduced basis  
methodology for  
parametric PDEs

$$\begin{cases} C_{\dagger} := P C P^T + Q \\ K := C_{\dagger} H^T (H C_{\dagger} H^T)^{-1} \end{cases}$$

how to  
**EFFICIENTLY**  
compute these  
expressions?

probabilistic  
Nordsieck method  
for ODEs

$$E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T$$

L1-norm  
minimization on  
manifolds

$$\begin{cases} x_{k|k-1} = F x_{k-1|k-1} + B u \\ P_{k|k-1} = F P_{k-1|k-1} F^T + Q \\ x_{k|k} = x_{k|k-1} + P_{k|k-1} H^T \times (H P_{k|k-1} H^T + R)^{-1} (z_k - H x_{k|k-1}) \\ P_{k|k} = P_{k|k-1} - P_{k|k-1} H^T \times (H P_{k|k-1} H^T + R)^{-1} H P_{k|k-1} \end{cases}$$

Kalman filter

$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y$$

$$\begin{cases} C_{\dagger} := P C P^T + Q \\ K := C_{\dagger} H^T (H C_{\dagger} H^T)^{-1} \end{cases}$$

$$E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T \quad \dots$$



- MUL
- ADD
- MOV
- MOVAPD
- VFMADDPD
- ...

$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y$$

$$\begin{cases} C_{\dagger} := PCP^T + Q \\ K := C_{\dagger} H^T (H C_{\dagger} H^T)^{-1} \end{cases}$$

$$E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T \quad \dots$$

$$y := \alpha x + y$$

$$\{L, U\} := LU(A)$$

$$C := \alpha AB + \beta C$$

$$L := L^{-1}$$

$$C := AB^T + BA^T + C \quad \dots$$

LINPACK



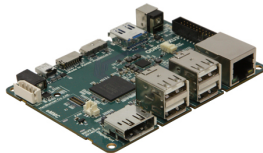
BLAS



LAPACK



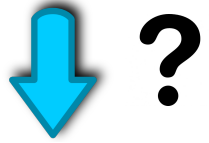
...



$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y$$

$$\begin{cases} C_{\dagger} := P C P^T + Q \\ K := C_{\dagger} H^T (H C_{\dagger} H^T)^{-1} \end{cases}$$

$$E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T \quad \dots$$

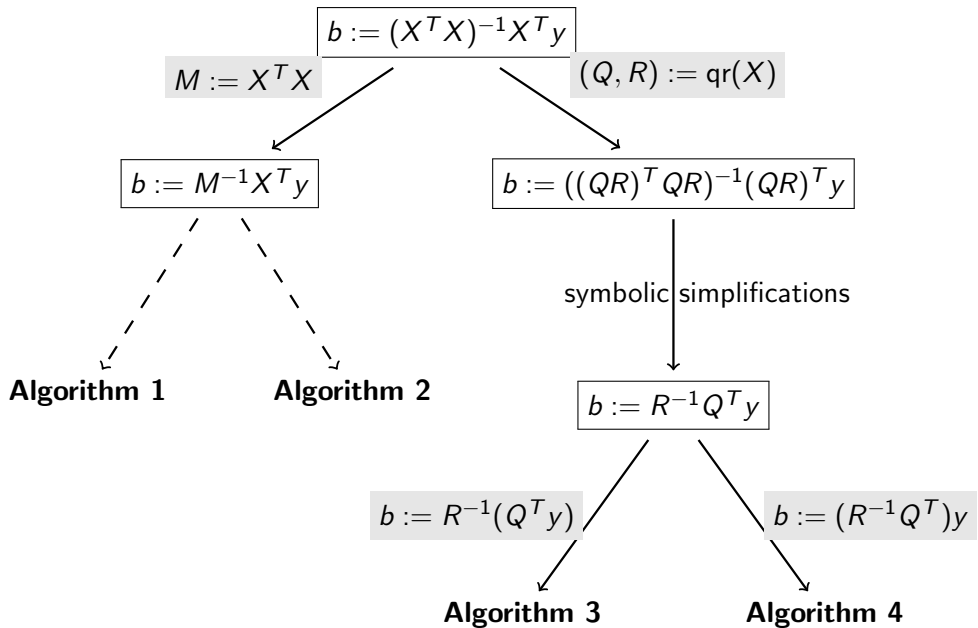


$$\begin{array}{ccc} y := \alpha x + y & \{L, U\} := LU(A) & C := \alpha AB + \beta C \\ L := L^{-1} & C := AB^T + BA^T + C & \dots \end{array}$$

LINPACK        BLAS        LAPACK        ...



$$b := (X^T X)^{-1} X^T y$$





# Linear Algebra Mapping Problem (LAMP)

# Linear Algebra Mapping Problem (LAMP)

- ▶  $\mathcal{E}$ : a list of assignments  $var_i := EXP_i$

## Linear Algebra Mapping Problem (LAMP)

- ▶  $\mathcal{E}$ : a list of assignments  $var_i := EXP_i$
- ▶  $\mathcal{K}$ : a list of available computational kernels (BLAS, LAPACK, ...)

# Linear Algebra Mapping Problem (LAMP)

- ▶  $\mathcal{E}$ : a list of assignments  $var_i := EXP_i$
- ▶  $\mathcal{K}$ : a list of available computational kernels (BLAS, LAPACK, ...)
- ▶  $\mathcal{M}$ : a metric (FLOPs, data movement, stability, time)

# Linear Algebra Mapping Problem (LAMP)

- ▶  $\mathcal{E}$ : a list of assignments  $var_i := EXP_i$
- ▶  $\mathcal{K}$ : a list of available computational kernels (BLAS, LAPACK, ...)
- ▶  $\mathcal{M}$ : a metric (FLOPs, data movement, stability, time)

## **LAMP:**

Find a decomposition of the expressions  $\mathcal{E}$  in terms of the kernels  $\mathcal{K}$ , optimal according to the metric  $\mathcal{M}$ .

# Linear Algebra Mapping Problem (LAMP)

- ▶  $\mathcal{E}$ : a list of assignments  $var_i := EXP_i$
- ▶  $\mathcal{K}$ : a list of available computational kernels (BLAS, LAPACK, ...)
- ▶  $\mathcal{M}$ : a metric (FLOPs, data movement, stability, time)

## **LAMP:**

Find a decomposition of the expressions  $\mathcal{E}$  in terms of the kernels  $\mathcal{K}$ , optimal according to the metric  $\mathcal{M}$ .

- ▶ Find a decomposition → easy
- ▶ Achieve optimality → NP complete

# LAMP is everywhere

## High-level languages

- ▶ Matlab
- ▶ R
- ▶ Julia
- ▶ Mathematica
- ▶ ...

# LAMP is everywhere

## High-level languages

- ▶ Matlab
- ▶ R
- ▶ Julia
- ▶ Mathematica
- ▶ ...

## Libraries

- ▶ Armadillo
- ▶ Blaze
- ▶ Blitz
- ▶ Eigen
- ▶ ...
- ▶ NumPy



# LAMP is everywhere

## High-level languages

- ▶ Matlab
- ▶ R
- ▶ Julia
- ▶ Mathematica
- ▶ ...

## Libraries

- ▶ Armadillo
- ▶ Blaze
- ▶ Blitz
- ▶ Eigen
- ▶ ...
- ▶ NumPy

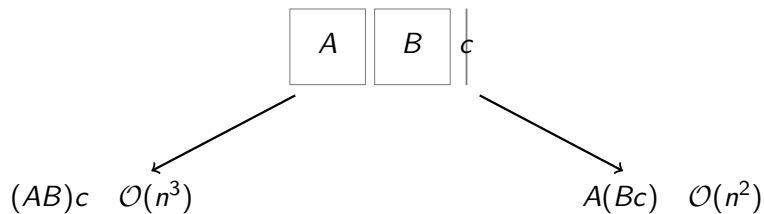
**human productivity vs. machine efficiency**

# Challenges and State of the Art

- ▶ Parenthesisation

# Challenges and State of the Art

- ▶ Parenthesisation



⇒ Matrix Chain Algorithm

# Challenges and State of the Art

- ▶ Parenthesisation

In practice: 
$$\left\{ \begin{array}{l} X := AB^T C^{-T} D + \dots \\ \text{LowerTriangular}(B) \\ \text{Symmetric}(C) \end{array} \right.$$

⇒ **Generalized** Matrix Chain Algorithm

# Challenges and State of the Art

- ▶ Metric: FLOPs vs. execution time

$$\operatorname{argmin}_{\mathcal{A}} (\text{FLOPs}(\mathcal{A})) \neq \operatorname{argmin}_{\mathcal{A}} (\text{time}(\mathcal{A}))$$

# Challenges and State of the Art

- ▶ Metric: FLOPs vs. execution time

$$\operatorname{argmin}_{\mathcal{A}} (\text{FLOPs}(\mathcal{A})) \neq \operatorname{argmin}_{\mathcal{A}} (\text{time}(\mathcal{A}))$$

⇒ Performance prediction

# Challenges and State of the Art

- ▶ Metric: FLOPs vs. execution time  
data moved, constraints on memory usage

$$\operatorname{argmin}_{\mathcal{A}} (\operatorname{data}(\mathcal{A})) \neq \operatorname{argmin}_{\mathcal{A}} (\operatorname{time}(\mathcal{A}))$$

# Challenges and State of the Art

- ▶ Multi-level metric: {stability, efficiency}

No explicit inversion!



# Challenges and State of the Art

- ▶ Multi-level metric: {stability, efficiency}

No explicit inversion!

$$X := AB^T C^{-T} D \quad \rightarrow \quad X := AB^T C^T \backslash D \quad \text{or} \quad X := AB^T / C^T \cdot D$$

$\Rightarrow$  Inversion  $\rightarrow$  Linear system  
better performance, better stability

# Challenges and State of the Art

- ▶ Multi-level metric: {stability, efficiency}

No explicit inversion!

$$X := AB^T C^{-T} D \quad \rightarrow \quad X := AB^T C^T \backslash D \quad \text{or} \quad X := AB^T / C^T \cdot D$$

However...

$$Y := A^{-1} B^{-1} \quad \text{inversion unavoidable}$$

$\Rightarrow$  Inversion  $\rightarrow$  Linear system  
better performance, better stability

# Challenges and State of the Art

▶ Linear algebra knowledge: identities, implications, theorems

- $((QR)^T QR)^{-1}(QR)^T y \rightarrow (R^T Q^T QR)^{-1} R^T Q^T y \rightarrow R^{-1} R^{-T} R^T Q^T y \rightarrow R^{-1} Q^T y$
- $\text{SPD}(A) \rightarrow \text{SPD}(A_{BR} - A_{BL} A_{TL}^{-1} A_{BL}^T)$  Schur complement

$\Rightarrow$  “Knowledge base” – expert system

# Challenges and State of the Art

- ▶ Inference of properties

$$E := Q^{-1}U(I + U^T Q^{-1}U)^{-1}U^T \quad \text{properties}(I + U^T Q^{-1}U) ?$$

$$\lambda(A, B) \wedge \begin{cases} \text{symm}(A) \\ \text{SPD}(B) \end{cases} \rightarrow \lambda(L^{-T}AL^{-1}) \quad \text{symmetric}(L^{-T}AL^{-1}) ?$$

⇒ Static analysis

## Challenges and State of the Art

- ▶ Common subexpressions

$$\begin{cases} X := AB^{-T}C \\ Y := B^{-1}A^T D \end{cases} \rightarrow \begin{cases} Z := AB^{-T} \\ X := ZC \\ Y := Z^T D \end{cases}$$

**Example:**  $w := AB^{-1}c$ ,  $\text{SPD}(B)$

**Example:**  $w := AB^{-1}c$ ,  $\text{SPD}(B)$

## Naive

```
w = A*inv(B)*c
```

## Recommended

```
w = A*(B\c)
```

## Generated

```
m10 = A; m11 = B; m12 = c;  
potrf!('L', m11)  
trsv!('L', 'N', 'N', m11, m12)  
trsv!('L', 'T', 'N', m11, m12)  
m13 = Array{Float64}(10)  
gemv!('N', 1.0, m10, m12, 0.0, m13)  
w = m13
```

# Experiments

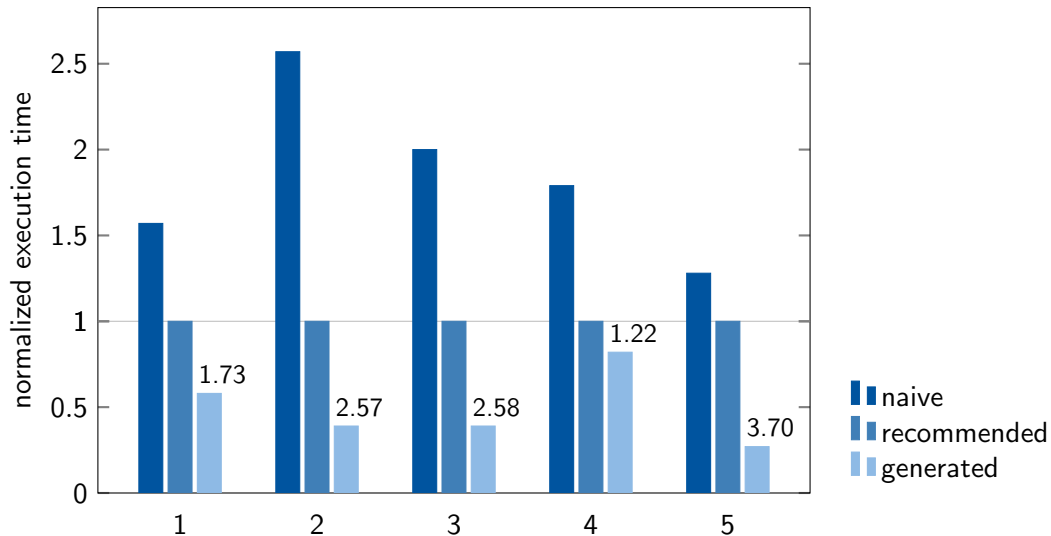
---

#	Example	
1	$b := (X^T X)^{-1} X^T y$	FullRank( $X$ )
2	$b := (X^T M^{-1} X)^{-1} X^T M^{-1} y$	SPD( $M$ ), FullRank( $X$ )
3	$W := A^{-1} B C D^{-T} E F$	LowTri( $A$ ), UppTri( $D, E$ )
4	$\begin{cases} X := A B^{-1} C \\ Y := D B^{-1} A^T \end{cases}$	SPD( $B$ )
5	$x := W(A^T(AWA^T)^{-1}b - c)$	FullRank( $A, W$ ) Diag( $W$ ), Pos( $W$ )

---



## Performance results



## Future Work

- ▶ **Linnea** as a compiler (off line) vs. **Linnea** as an interpreter (real time)
- ▶ Integration into languages and libraries
- ▶ Aforementioned challenges, and then some:  
sequences of operations, memory usage, tensors, . . .
- ▶ **YOU**: What instances of LAMP do you encounter?  
How do you solve them? Please let me know.

## (Initial) References

- ▶ **A Domain-specific Compiler for Linear Algebra Operations**,  
Diego Fabregat-Traver and Paolo Bientinesi  
Lecture Notes in Computer Science, Vol.7851, 2013.
- ▶ **Application-tailored Linear Algebra Algorithms: A Search-based Approach**,  
Diego Fabregat-Traver and Paolo Bientinesi,  
International Journal of High Performance Computing Applications, Vol.27(4), 2013.
- ▶ **The Matrix Chain Algorithm to Compile Linear Algebra Expressions**,  
Barthels and Paolo Bientinesi,  
DSLDI 2016, <https://arxiv.org/pdf/1611.05660>.

## (Initial) References

- ▶ **A Domain-specific Compiler for Linear Algebra Operations**,  
Diego Fabregat-Traver and Paolo Bientinesi  
Lecture Notes in Computer Science, Vol.7851, 2013.
- ▶ **Application-tailored Linear Algebra Algorithms: A Search-based Approach**,  
Diego Fabregat-Traver and Paolo Bientinesi,  
International Journal of High Performance Computing Applications, Vol.27(4), 2013.
- ▶ **The Matrix Chain Algorithm to Compile Linear Algebra Expressions**,  
Barthels and Paolo Bientinesi,  
DSLDI 2016, <https://arxiv.org/pdf/1611.05660>.

**Thank You!**