Linnea: Automatic Generation of Efficient Linear Algebra Programs

Henrik Barthels

HPAC High Performance and Automatic Computing

RWTH AACHEN UNIVERSITY

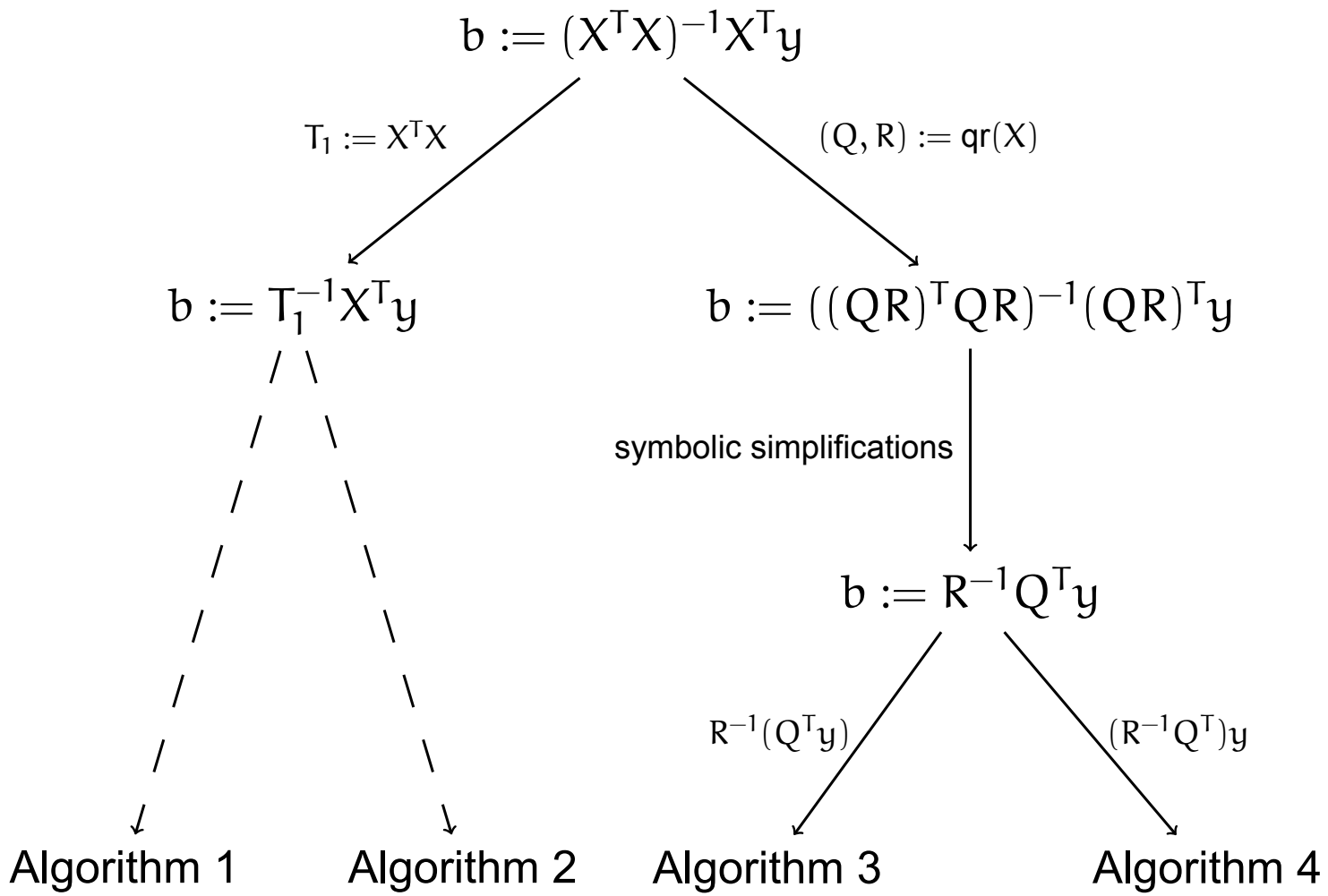
Introduction

- How to compute the following expressions?

$$b := (X^T X)^{-1} X^T y$$
$$b := (X^T M^{-1} X)^{-1} X^T M^{-1} y$$
$$x := W(A^T (AWA^T)^{-1} b - c)$$
$$x := \left(A^{-T} B^T B A^{-1} + R^T[\Lambda(Rz)]R\right)^{-1} A^{-T} B^T B A^{-1} y$$

- High-level languages are easy to use, but performance is usually suboptimal.
- BLAS and LAPACK can be fast, but require a lot of expertise.
- Goal: Simplicity **and** performance.

Introduction



Input

```
n = 1500
m = 1000

Matrix M(n, n) <Input, SPD>
Matrix X(n, m) <Input, FullRank>
ColumnVector y(n) <Input>
ColumnVector b(m) <Output>

b = inv(X'*inv(M)*X)*X'*inv(M)*y
```

Instruction Set

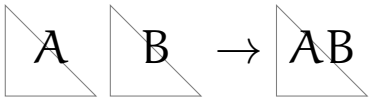
BLAS [DDC+90]

- $y \leftarrow Ax + y$
- $C \leftarrow AB + C$
- $B \leftarrow A^{-1}B$
- ...

LAPACK [AB+99]

- Matrix factorizations.
- Eigensolvers.
- Solvers for linear systems with specific properties.

Linear Algebra Knowledge

- Properties: `trmm` vs. `gemm`
- Inference of properties: 
- Simplifications: $A^T \rightarrow A$ if Symmetric(A)
- Rewriting expressions:

$$X := A^TA + A^TB + B^TA \quad \rightarrow \quad \begin{aligned} Y &:= B + A/2 \\ X &:= A^TY + Y^TA \end{aligned}$$

- Common subexpressions:

$$X := AB^{-T}C + B^{-1}A^T \quad \rightarrow \quad \begin{aligned} Z &:= AB^{-T} \\ X &:= ZC + Z^T \end{aligned}$$

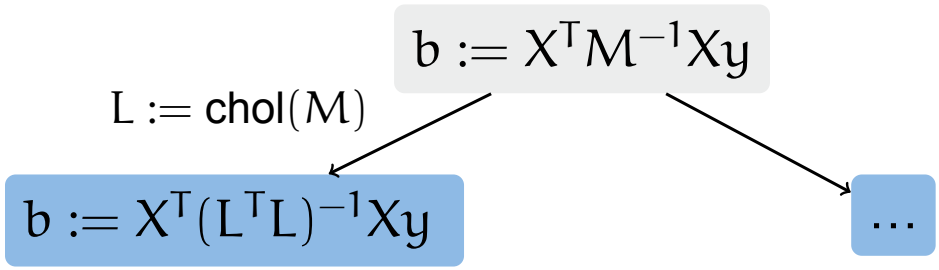
- Matrix chains:

$$\begin{aligned} (AB)c \quad & \mathcal{O}(n^3) \\ A(Bc) \quad & \mathcal{O}(n^2) \end{aligned}$$

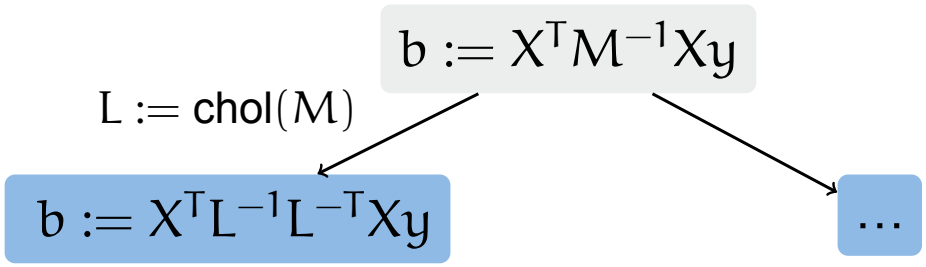
Derivation Graph

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

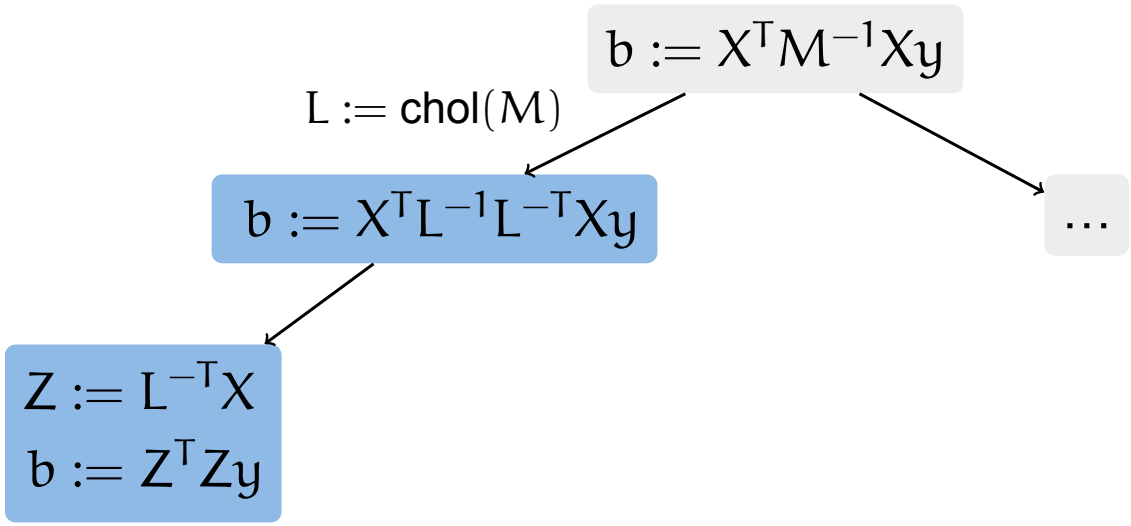
Derivation Graph



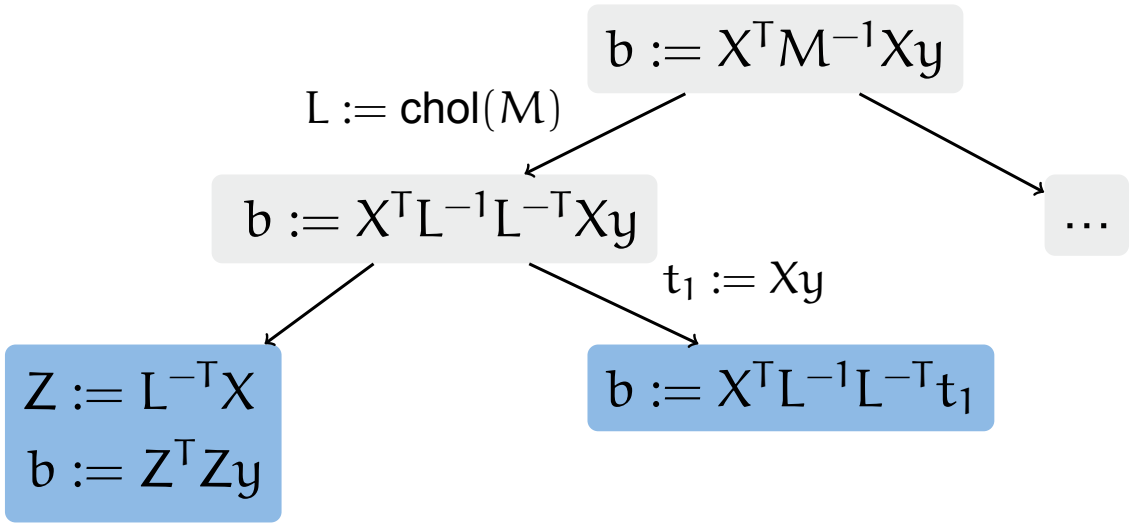
Derivation Graph



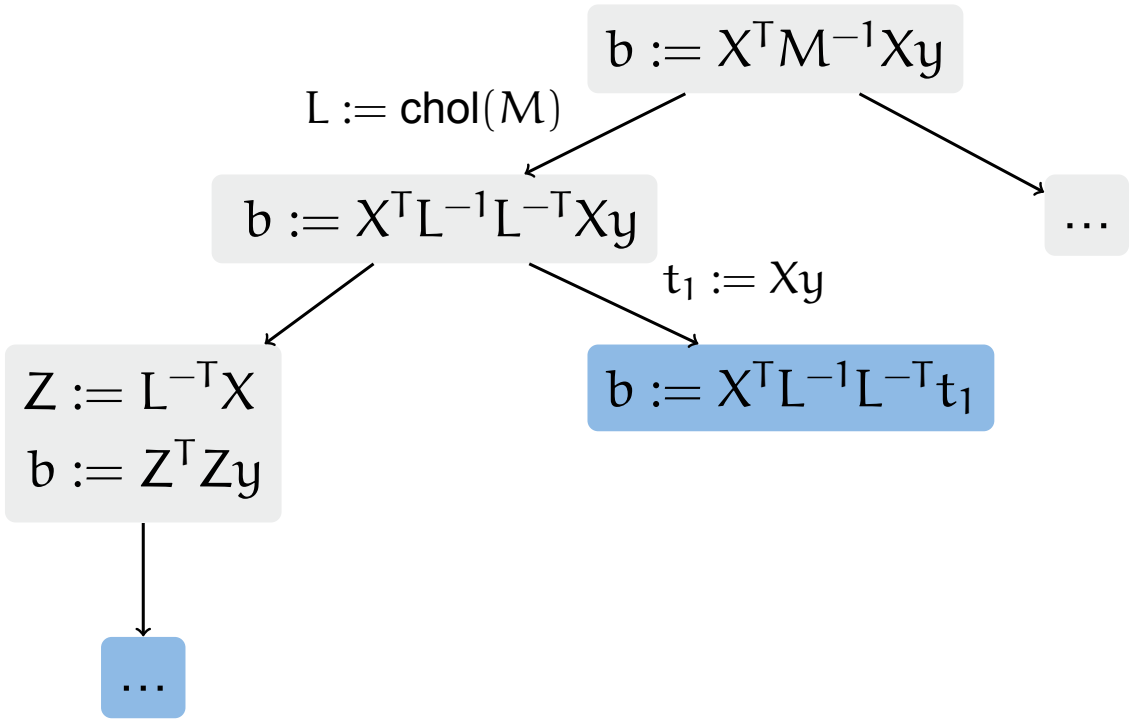
Derivation Graph



Derivation Graph

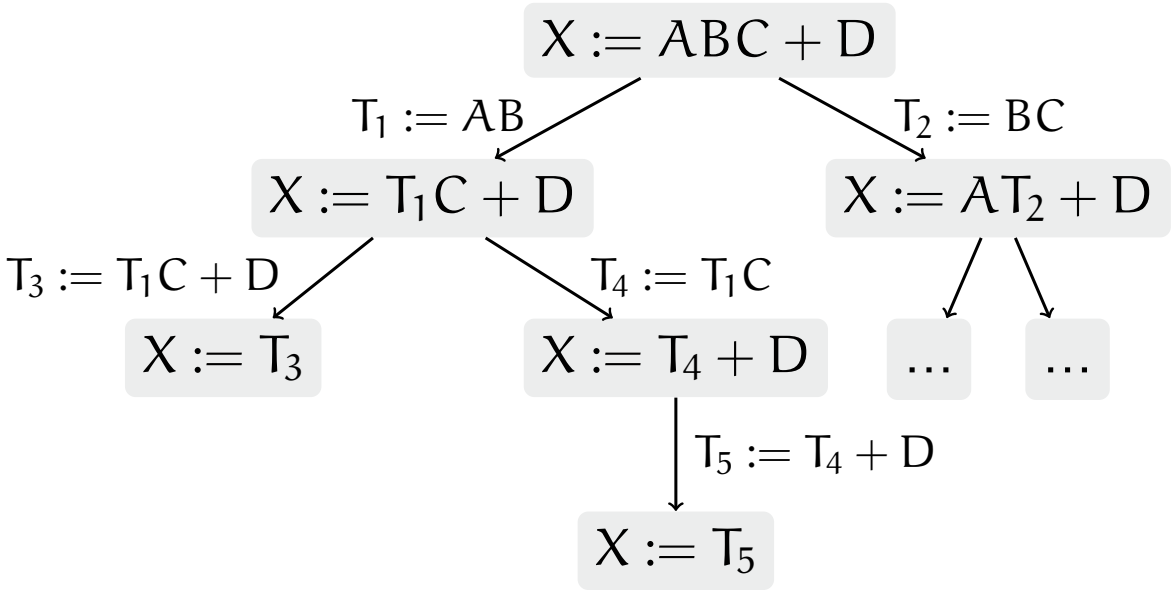


Derivation Graph

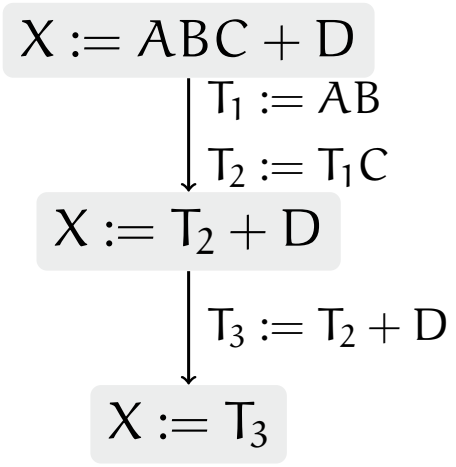


Derivation Graph

Exhaustive

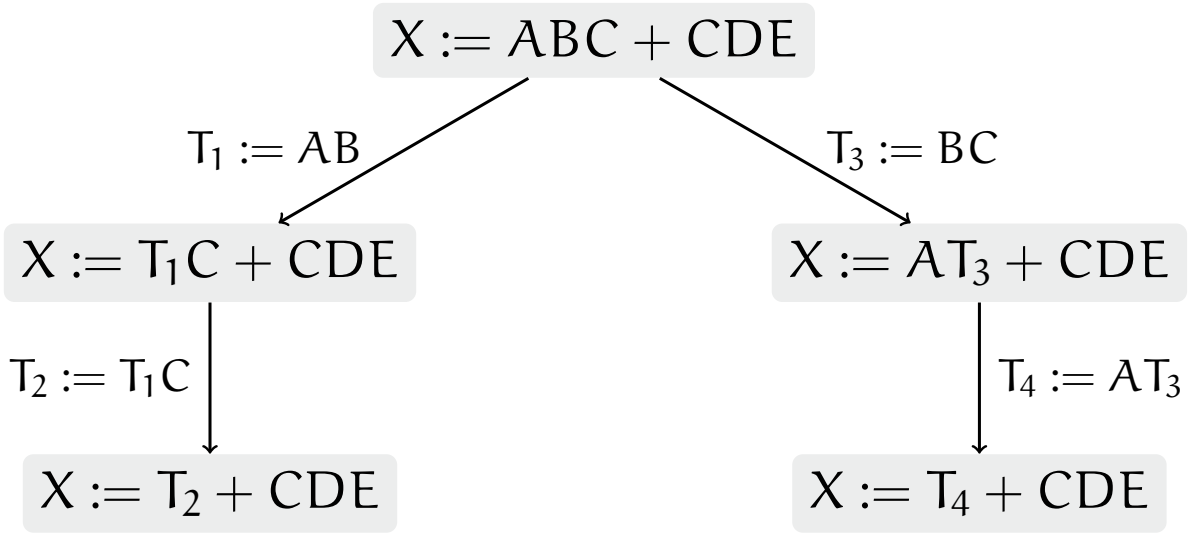


Constructive



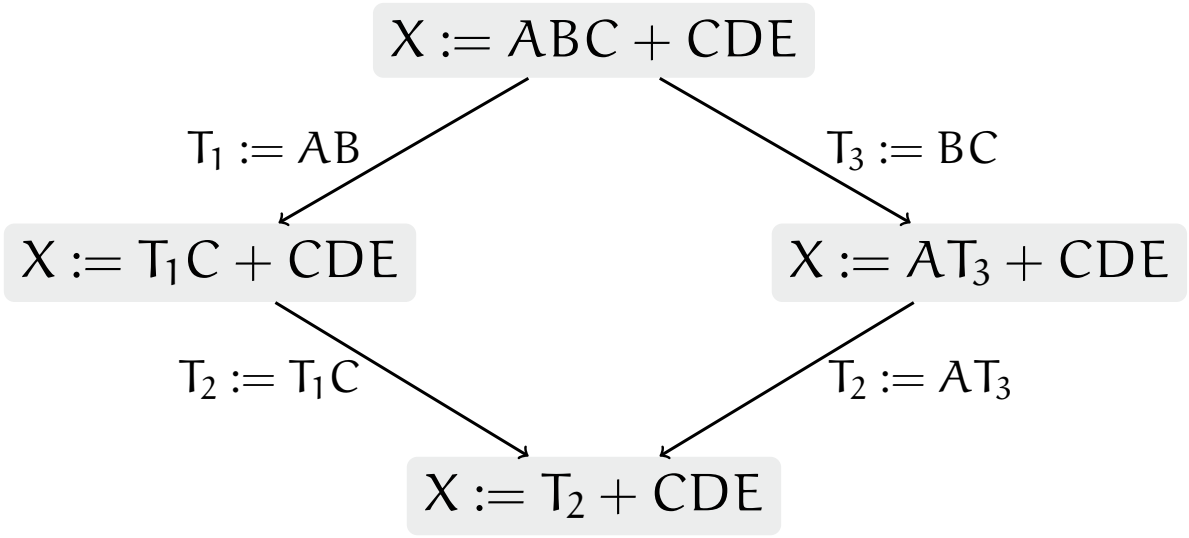
Derivation Graph

Reducing Redundancy



Derivation Graph

Reducing Redundancy



Results

Example: $w := AB^{-1}c$

Naive

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

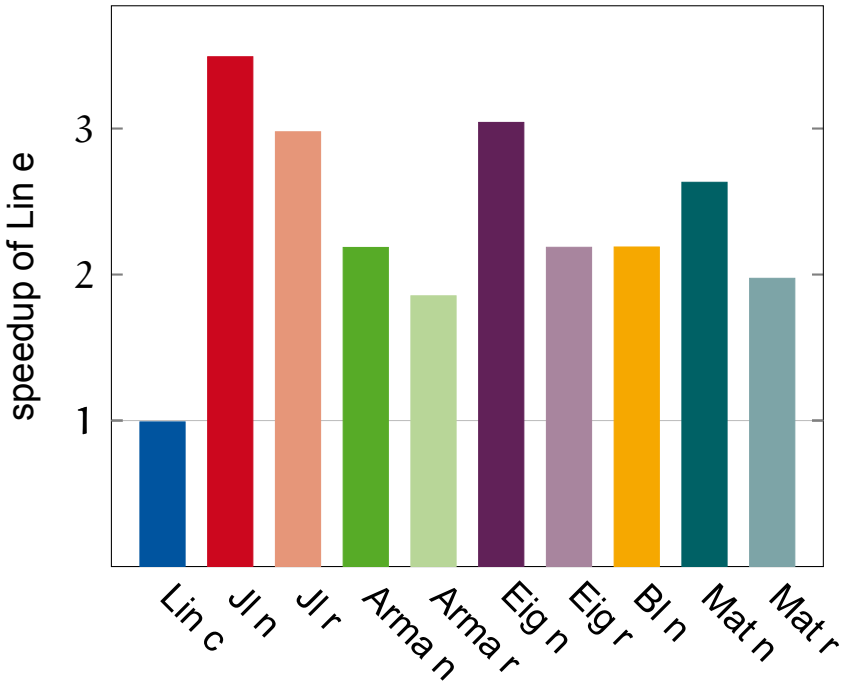
Recommended

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

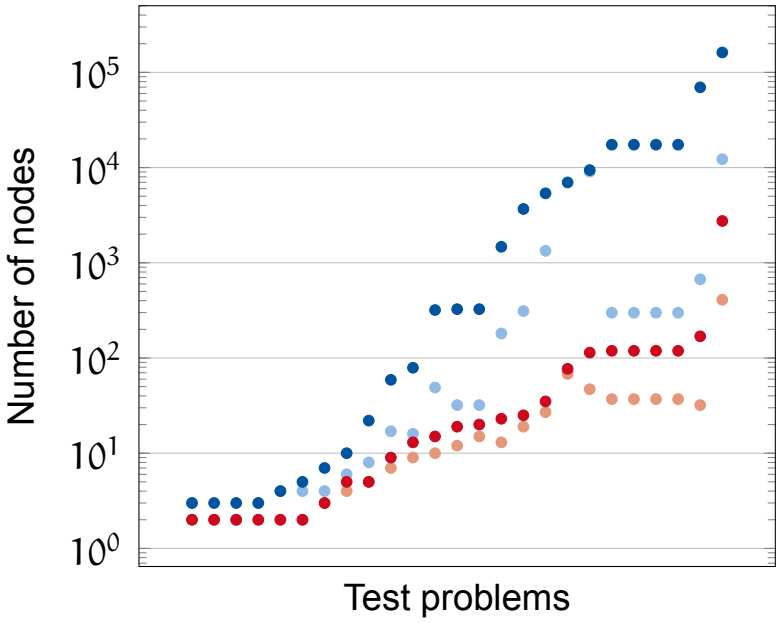
Generated

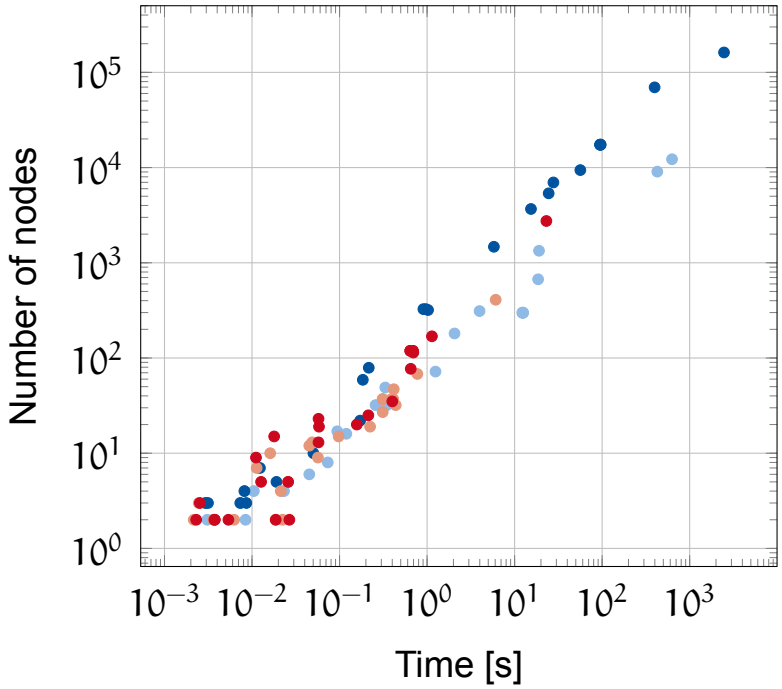
```
ml0 = A; ml1 = B; ml2 = c;
potrf!('L', ml1)
trsv!('L', 'N', 'N', ml1, ml2)
trsv!('L', 'T', 'N', ml1, ml2)
ml3 = Array{Float64}(10)
gemv!('N', 1.0, ml0, ml2, 0.0, ml3)
w = ml3
```

Results


speedup of Lin e
3
2
1
Lin c
Jl n
Jl r
Arma n
Arma r
Eig n
Eig r
Bl n
Mat n
Mat r

Results





exhaustive, merging
exhaustive, no merging
constructive, merging
constructive, no merging

References

[AB+99] Edward Anderson, Zhaojun Bai, et al. *LAPACK Users' guide*, volume 9. SIAM, 1999.

[DDC+90] Jack J. Dongarra, Jeremy Du Croz, et al. A set of Level 3 Basic Linear Algebra Subprograms. *ACM TOMS*, 16(1):1–17, 1990.