A journey from scalar to tensor computations

A tale of efficiency and productivity

Paolo Bientinesi
Aachen Institute for Computational Engineering Science
RWTH Aachen University

Tensor Computation Workshop
September 14 & 15, 2017
Flatiron Institute, New York City

Deutsche Forschungsgemeinschaft DFG

HPAC High Performance and Automatic Computing

RWTH AACHEN UNIVERSITY

High-performance & Automatic Computing

- **Edoardo Di Napoli**
  Diego Traver-Fabregat Taxonomy of contractions: Can you GEMM?
  "Towards an Efficient Use of the BLAS Library for Multilinear Tensor Contractions", AMC 235, 2014

High-performance & Automatic Computing

- **Edoardo Di Napoli**
  Diego Traver-Fabregat Taxonomy of contractions: Can you GEMM?
  "Towards an Efficient Use of the BLAS Library for Multilinear Tensor Contractions", AMC 235, 2014

- **Elmar Peise** Performance prediction
  "On the Performance Prediction of BLAS-based Tensor Contractions", PMBS, SC'14

High-performance & Automatic Computing

- **Edoardo Di Napoli**
  Diego Traver-Fabregat
  Taxonomy of contractions: Can you GEMM?
  "Towards an Efficient Use of the BLAS Library for Multilinear Tensor Contractions", AMC 235, 2014

- **Elmar Peise**
  Performance prediction
  "On the Performance Prediction of BLAS-based Tensor Contractions", PMBS, SC'14

- **Yurii Aulchenko**
  Diego Traver-Fabregat
  Genome-wide association studies (GWAS)
  "Computing Petaflops over Terabytes of Data: The Case of Genome-Wide Association Studies", TOMS 40, 2014

High-performance & Automatic Computing

- **Edoardo Di Napoli**
  Diego Traver-Fabregat — Taxonomy of contractions: Can you GEMM?

  "Towards an Efficient Use of the BLAS Library for Multilinear Tensor Contractions", AMC 235, 2014

- **Elmar Peise** — Performance prediction

  "On the Performance Prediction of BLAS-based Tensor Contractions", PMBS, SC'14

- **Yurii Aulchenko**
  Diego Traver-Fabregat — Genome-wide association studies (GWAS)

  "Computing Petaflops over Terabytes of Data: The Case of Genome-Wide Association Studies", TOMS 40, 2014

- **Edoardo Di Napoli**
  Elmar Peise — Density Functional Theory: FLAPW methods

  "High-Performance Generation of the Hamiltonian and Overlap Matrices in FLAPW Methods", CPC 2017

High-performance & Automatic Computing

- **Edoardo Di Napoli**
  Diego Traver-Fabregat — Taxonomy of contractions: Can you GEMM?
  "Towards an Efficient Use of the BLAS Library for Multilinear Tensor Contractions", AMC 235, 2014

- **Elmar Peise** — Performance prediction
  "On the Performance Prediction of BLAS-based Tensor Contractions", PMBS, SC'14

- **Yurii Aulchenko**
  Diego Traver-Fabregat — Genome-wide association studies (GWAS)
  "Computing Petaflops over Terabytes of Data: The Case of Genome-Wide Association Studies", TOMS 40, 2014

- **Edoardo Di Napoli**
  Elmar Peise — Density Functional Theory: FLAPW methods
  "High-Performance Generation of the Hamiltonian and Overlap Matrices in FLAPW Methods", CPC 2017

- **Paul Springer** — High-performance kernels
  "TTC: A high-performance Compiler for Tensor Transpositions", TOMS 44, 2017
  "Design of a High-Performance GEMM-like Tensor-Tensor Multiplication", TOMS, 2017

History

Scalars

- 50s: Assembly code
  Building blocks == ISA

Scalars

- 50s: Assembly code
  Building blocks == ISA

- [1954]: FORTRAN (IBM, John Backus)
  "Specifications for the IBM Mathematical FORmula TRANslating system, FORTRAN"
  no more Assembly! $\rightarrow$ compiler

Scalars

- 50s: Assembly code
  Building blocks == ISA
- [1954]: FORTRAN (IBM, John Backus)
  "Specifications for the IBM Mathematical FORmula TRANslating system, FORTRAN"
  no more Assembly! $\rightarrow$ compiler $\rightarrow$ ACM Turing award (1977)

Scalars

- 50s: Assembly code
  Building blocks == ISA

- [1954]: FORTRAN (IBM, John Backus)
  "Specifications for the IBM Mathematical FORmula TRANslating system, FORTRAN"
  no more Assembly! $\rightarrow$ compiler $\rightarrow$ ACM Turing award (1977)

- However, NOT the solution to all problems

Matrices

- [70s, . . . , today]: Identification, standardization, optimization of building blocks

  Libraries: LINPACK, BLAS, LAPACK, FFTW, . . .

  Convenience, portability, separation of concerns

Matrices

- [70s, . . . , today]: Identification, standardization, optimization of building blocks

  Libraries: LINPACK, BLAS, LAPACK, FFTW, . . .

  Convenience, portability, separation of concerns

- [80s–early 90s]: Memory hierarchy

  Libraries $\rightarrow$ necessity

Matrices

- [70s, . . . , today]: Identification, standardization, optimization of building blocks

  Libraries: LINPACK, BLAS, LAPACK, FFTW, . . .

  Convenience, portability, separation of concerns

- [80s–early 90s]: Memory hierarchy

  Libraries $\rightarrow$ necessity

- **How to use them?** (not just optimal parenthesisation)

Applications

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

exponential transient excision

$$\forall i\ \forall j \quad b_{ij} := \left(X_i^T M_j^{-1} X_i\right)^{-1} X_i^T M_j^{1} y_j$$

GWAS

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

probabilistic Nordsieck method for ODEs

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

L1-norm minimization on manifolds

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

Kalman filter

$$x := A(B^TB + A^TR^T\Lambda RA)^{-1}B^TBA^{-1}y$$

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

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

...

$y := \alpha x + y$ | $LU = A$ | $\cdots$ | $C := \alpha AB + \beta C$

$X := A^{-1}B$ | $C := AB^T + BA^T + C$ | $X := L^{-1}ML^{-T}$ | $QR = A$

LINPACK **BLAS** **LAPACK** **...**

$$x := A(B^TB + A^TR^T\Lambda RA)^{-1}B^TBA^{-1}y$$

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

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

...

?

$y := \alpha x + y$ | $LU = A$ | $\cdots$ | $C := \alpha AB + \beta C$

$X := A^{-1}B$ | $C := AB^T + BA^T + C$ | $X := L^{-1}ML^{-T}$ | $QR = A$

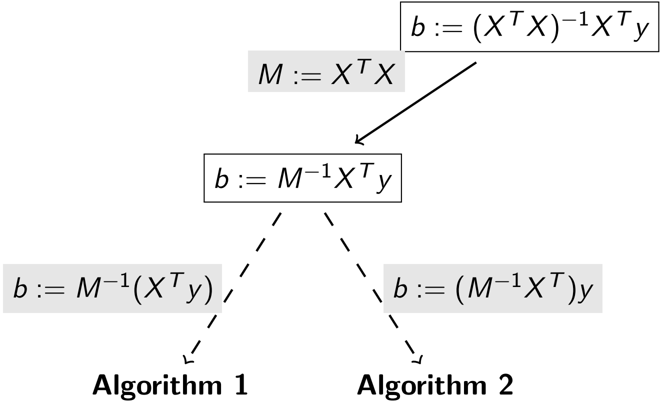
LINPACK **BLAS** **LAPACK** **...**

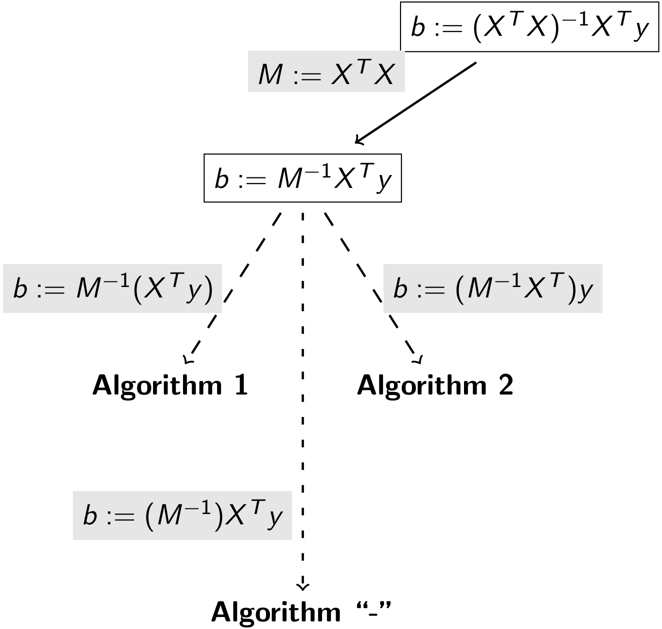
$$\boxed{b := (X^T X)^{-1} X^T y}$$

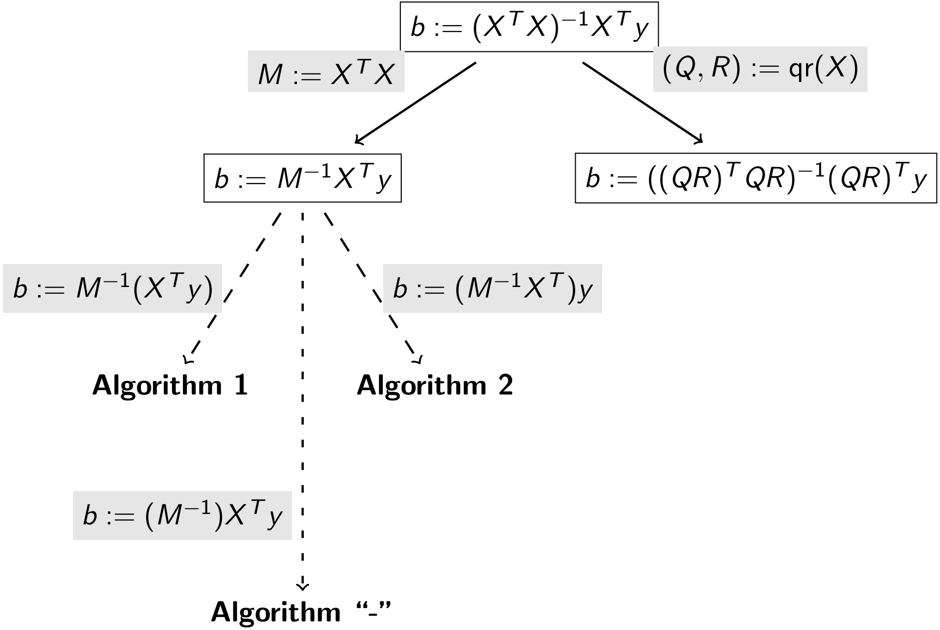
$$b := (X^T X)^{-1} X^T y$$

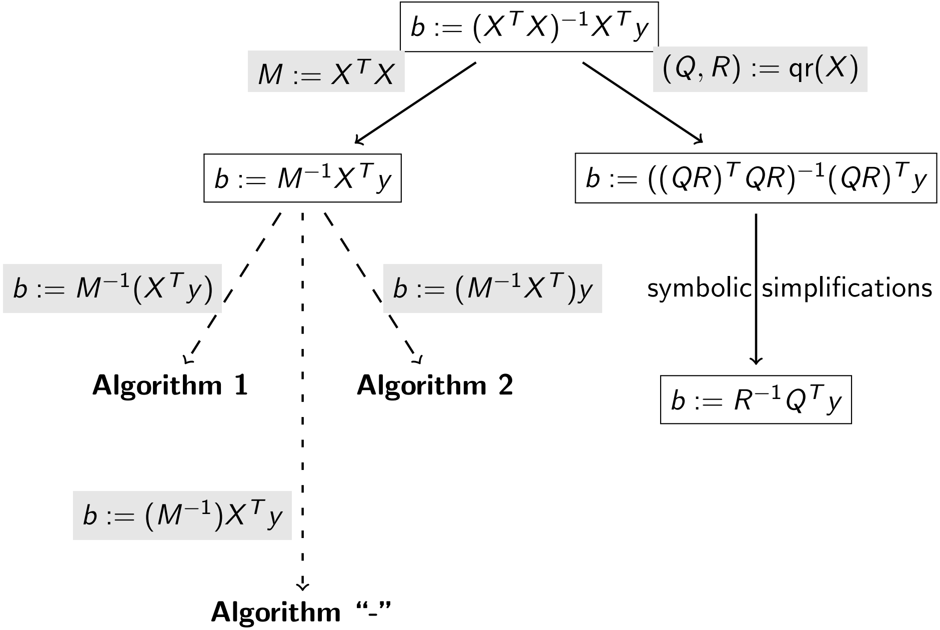
$$M := X^T X$$

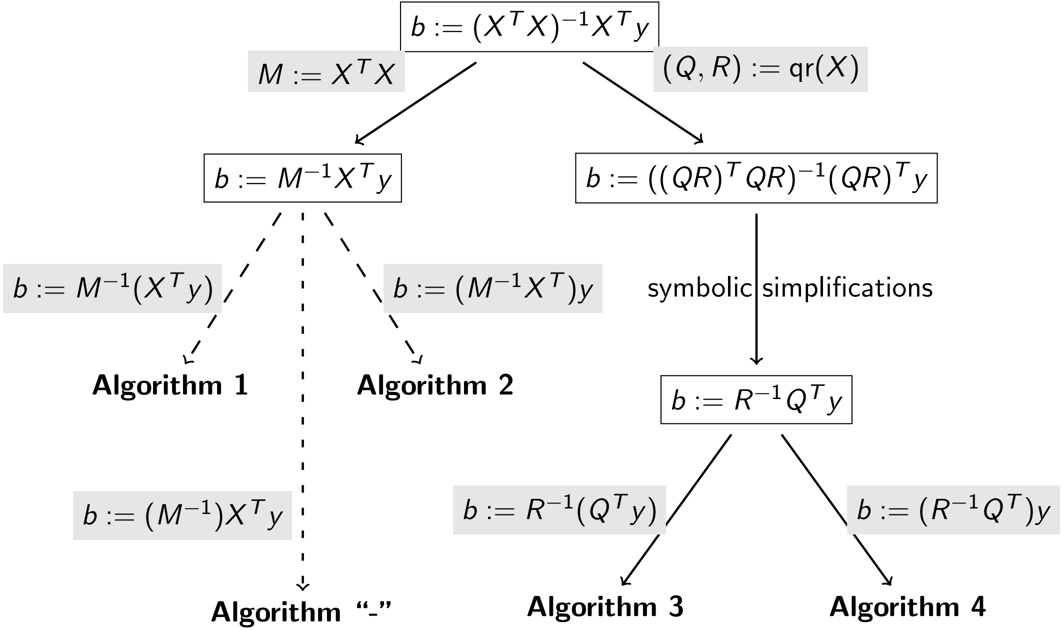
$$b := M^{-1} X^T y$$

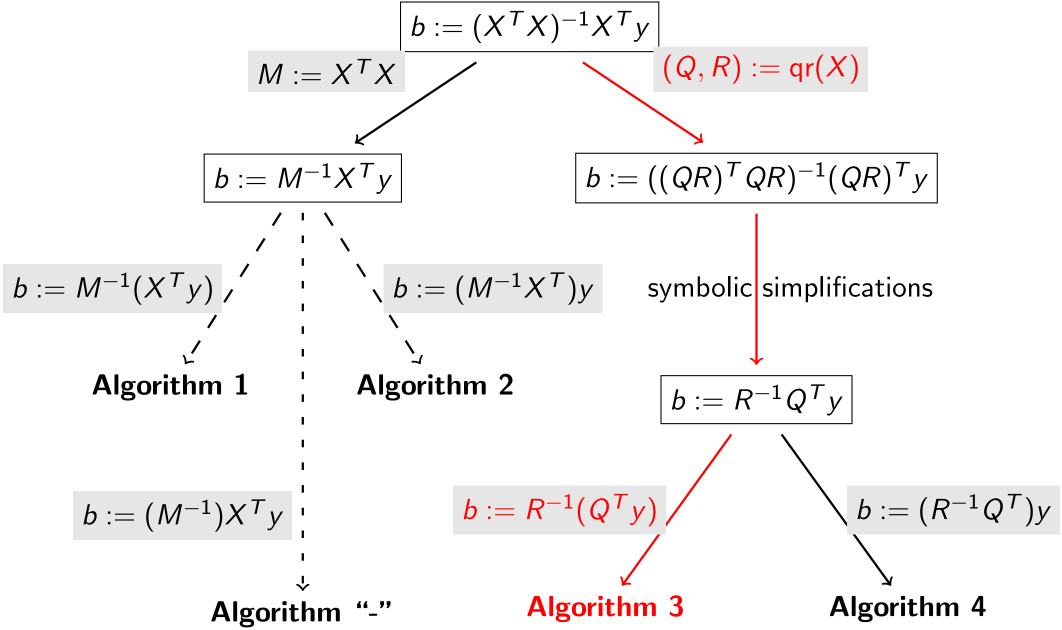

$b := (X^T X)^{-1} X^T y$
$M := X^T X$
$b := M^{-1} X^T y$
$b := M^{-1}(X^T y)$
$b := (M^{-1} X^T) y$
Algorithm 1
Algorithm 2

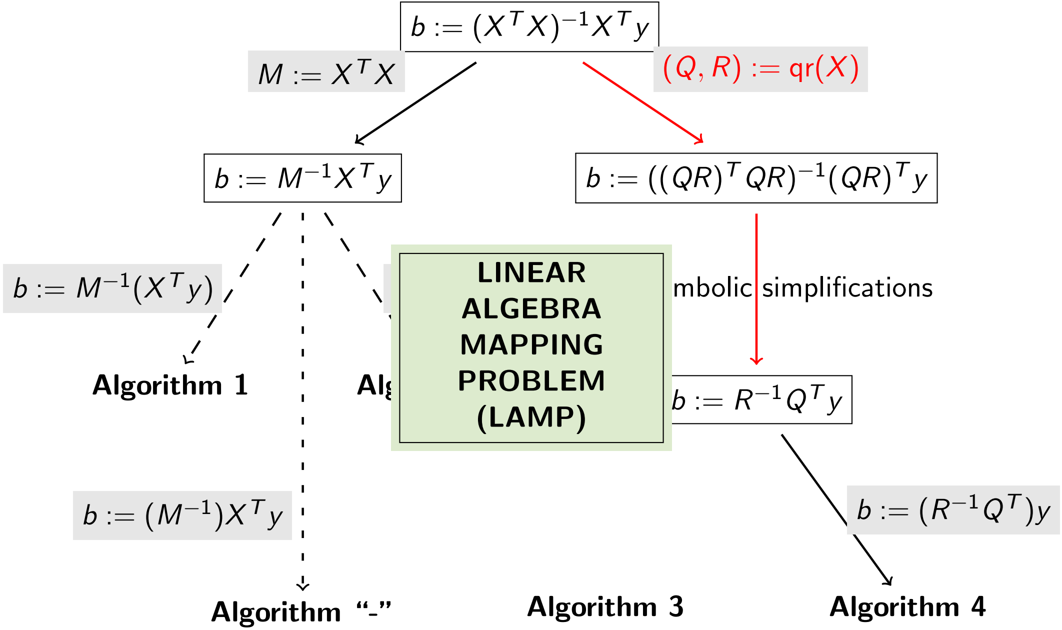

$b := (X^T X)^{-1} X^T y$
$M := X^T X$
$b := M^{-1} X^T y$
$b := M^{-1}(X^T y)$
$b := (M^{-1} X^T) y$
Algorithm 1
Algorithm 2
$b := (M^{-1}) X^T y$
Algorithm "-"


$b := (X^T X)^{-1} X^T y$
$M := X^T X$
$(Q, R) := \mathrm{qr}(X)$
$b := M^{-1} X^T y$
$b := ((QR)^T QR)^{-1} (QR)^T y$
$b := M^{-1}(X^T y)$
$b := (M^{-1} X^T) y$
Algorithm 1
Algorithm 2
$b := (M^{-1}) X^T y$
Algorithm “-”


$b := (X^TX)^{-1}X^Ty$
$M := X^TX$
$(Q, R) := \text{qr}(X)$
$b := M^{-1}X^Ty$
$b := ((QR)^TQR)^{-1}(QR)^Ty$
$b := M^{-1}(X^Ty)$
$b := (M^{-1}X^T)y$
symbolic simplifications
Algorithm 1
Algorithm 2
$b := R^{-1}Q^Ty$
$b := (M^{-1})X^Ty$
Algorithm "-"


$b := (X^T X)^{-1} X^T y$
$M := X^T X$
$(Q, R) := \mathrm{qr}(X)$
$b := M^{-1} X^T y$
$b := ((QR)^T QR)^{-1} (QR)^T y$
$b := M^{-1}(X^T y)$
$b := (M^{-1} X^T) y$
symbolic simplifications
Algorithm 1
Algorithm 2
$b := R^{-1} Q^T y$
$b := (M^{-1}) X^T y$
$b := R^{-1}(Q^T y)$
$b := (R^{-1} Q^T) y$
Algorithm "-"
Algorithm 3
Algorithm 4


$b := (X^TX)^{-1}X^Ty$
$M := X^TX$
$(Q, R) := \mathrm{qr}(X)$
$b := M^{-1}X^Ty$
$b := ((QR)^TQR)^{-1}(QR)^Ty$
$b := M^{-1}(X^Ty)$
$b := (M^{-1}X^T)y$
symbolic simplifications
Algorithm 1
Algorithm 2
$b := R^{-1}Q^Ty$
$b := (M^{-1})X^Ty$
$b := R^{-1}(Q^Ty)$
$b := (R^{-1}Q^T)y$
Algorithm "-"
Algorithm 3
Algorithm 4


$b := (X^TX)^{-1}X^Ty$
$M := X^TX$
$(Q, R) := \mathrm{qr}(X)$
$b := M^{-1}X^Ty$
$b := ((QR)^TQR)^{-1}(QR)^Ty$
$b := M^{-1}(X^Ty)$
mbolic simplifications
LINEAR
ALGEBRA
MAPPING
PROBLEM
(LAMP)
Algorithm 1
$b := R^{-1}Q^Ty$
$b := (M^{-1})X^Ty$
$b := (R^{-1}Q^T)y$
Algorithm "-"
Algorithm 3
Algorithm 4

Human productivity vs. machine efficiency

A well-known problem

High-level languages

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

Libraries

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

Example

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

Example

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

Naive $\leftarrow$ NEVER!!

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

Example

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

Naive $\leftarrow$ NEVER!!

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

Recommended

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

Example

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

Naive $\leftarrow$ NEVER!!

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

Recommended

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

Expert

```
L = Chol(B)
w = A * (L'/(L/c))
```

Example

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

Naive $\leftarrow$ NEVER!!

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

Recommended

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

Expert

```
L = Chol(B)
w = A * (L'/(L/c))
```

Generated – "Linnea" by H. Barthels

```
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
```

Tensors

- Building blocks?

Tensors

- Building blocks?
  - BLAS, LAPACK

$$(S)_{G',G} = \sum_{a} \sum_{L=(l,m)} \left(A_L^{a,G'}\right)^* A_L^{a,G} + \left(B_L^{a,G'}\right)^* B_L^{a,G} \, \|\dot{u}_{l,a}\|^2$$

$$\begin{aligned}(H)_{G',G} = \sum_{a} \sum_{L',L} & \left(A^*_{L',a,t'} \, T^{[AA]}_{L',L;a} \, A_{L,a,t}\right) + \left(A^*_{L',a,t'} \, T^{[AB]}_{L',L;a} \, B_{L,a,t}\right) \\ & + \left(B^*_{L',a,t'} \, T^{[BA]}_{L',L;a} \, A_{L,a,t}\right) + \left(B^*_{L',a,t'} \, T^{[BB]}_{L',L;a} \, B_{L,a,t}\right) .\end{aligned}$$

Generation of overlap and Hamiltonian matrices. With E. Di Napoli.

Tensors

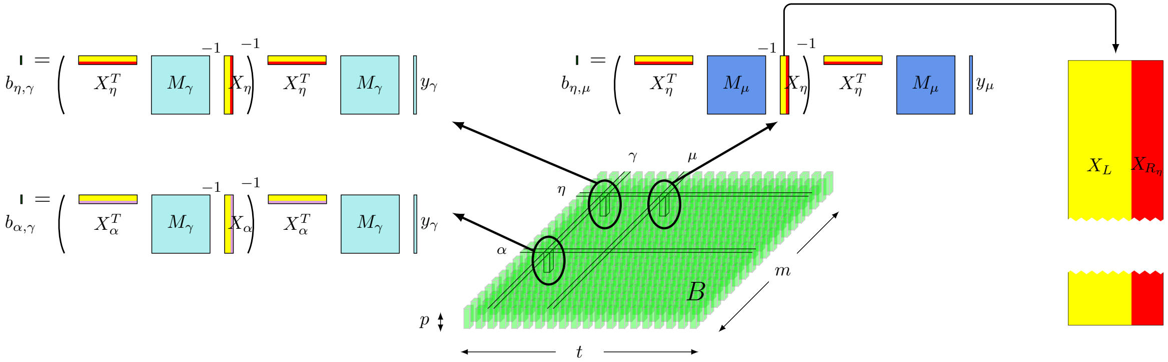
- Building blocks?
  - BLAS, LAPACK

for $i \mathrel{+}= 1, \ldots, N_A$:
  try:
    $C_a \coloneqq \mathrm{Chol}(T_a^{[AA]})$ (**zpotrf**: $\frac{4}{3} N_L^3 + O(N_L^2)$ FLOPs)
  success:
    $Y_a \coloneqq C_a^H A_a$ (**ztrmm**: $4 N_L^2 N_G$ FLOPs)
    add Y_a to Y_{HPD}
  failure:
    $X_a \coloneqq T_a^{[AA]} A_a$ (**zhemm**: $8 N_L^2 N_G$ FLOPs)
    add X_a to $X_{\neg\mathrm{HPD}}$
    add A_a to $A_{\neg\mathrm{HPD}}$
$H \mathrel{+}= A_{\neg\mathrm{HPD}}^H X_{\neg\mathrm{HPD}}$ (**zgemm**: $8 N_{A_{\neg\mathrm{HPD}}} N_L N_G^2$ FLOPs)
$H \mathrel{+}= Y_{\mathrm{HPD}}^H Y_{\mathrm{HPD}}$ (**zherk**: $4 N_{A_{\mathrm{HPD}}} N_L N_G^2$ FLOPs)

10x more flops. Speedups: 1.5–2.5x. With E. Di Napoli.

Tensors

- Building blocks?
  - BLAS, LAPACK



Tensors

- Building blocks?
  - BLAS, LAPACK
  - Contractions, transpositions, . . .

CCS, CCSD, . . .

Tensors

- Building blocks?
  - BLAS, LAPACK
  - Contractions, transpositions, ...
  - ???

$$TPP_{\alpha_1,n_1,\alpha'_1,n'_1,s_1,s_2,s'_1,s'_2} =$$

$$\frac{1}{\beta} \sum_{s_3,s_4,s'_3,s'_4} \sum_{n=-N_{int}}^{N_{int}-1} \sum_{\alpha,\beta}^{N_p} PP^{n,s'_3,s'_4}_{\alpha_1,n_1,\alpha,s_1,s_2} X^{n,s'_3,s'_4}_{\alpha,\beta,s_3,s_4} PP^{n,s'_3,s'_4}_{\beta,\alpha'_1,n'_1,s_1,s_2}$$

Quantum Field Theory, Single Impurity Anderson Model. With E. Di Napoli.

Tensors

- Building blocks?
  - BLAS, LAPACK
  - Contractions, transpositions, . . .
  - ???
- Do we have a unifying language/formalism? Tensor Networks?

Tensors

- Building blocks?
  - BLAS, LAPACK
  - Contractions, transpositions, . . .
  - ???
- Do we have a unifying language/formalism?   Tensor Networks?
- Are we ready to fix interfaces & standards?

Tensors

- Building blocks?
  - BLAS, LAPACK
  - Contractions, transpositions, ...
  - ???
- Do we have a unifying language/formalism? Tensor Networks?
- Are we ready to fix interfaces & standards?
- For once, shall we focus on **performance *AND* productivity**?

Experiments – Linnea

#	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	$\begin{cases} X := AB^{-1}C \\ Y := DB^{-1}A^T \end{cases}$	SPD(B)
5	$x := W(A^T(AWA^T)^{-1}b - c)$	FullRank(A, W) Diag(W), Pos(W)
⋮		

Performance results