A tale of efficiency and productivity.
From scalar to tensor computations.

Paolo Bientinesi
Aachen Institute for Computational Engineering Science
RWTH Aachen University

October 23, 2017
Umeå Universitet

The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

Linear mixed models

$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Lennard-Jones potential

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-2\hbar}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

Schrödinger eqn.

⋮

The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

LINEAR MIXED MODELS

$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

LENNARD-JONES POTENTIAL

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-2\hbar}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

SCHRÖDINGER EQN.

⋮

The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

Linear mixed models

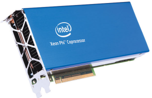
$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Lennard-Jones potential

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-2\hbar}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

Schrödinger eqn.

⋮



The world of scientific computing

$$\boldsymbol{y} = X\beta + Z\boldsymbol{u} + \epsilon$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

Linear mixed models

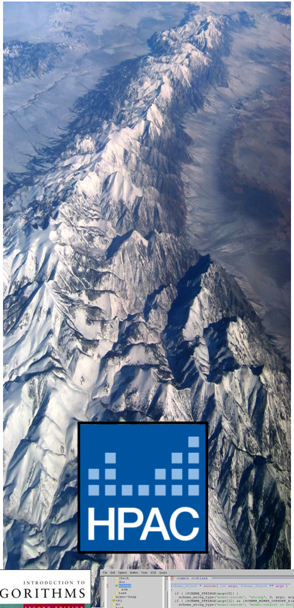
$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Lennard-Jones potential

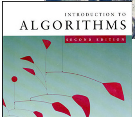
$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-2\hbar}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

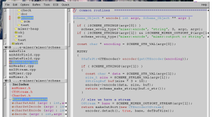
Schrödinger eqn.

⋮



HPAC





The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

LINEAR MIXED MODELS

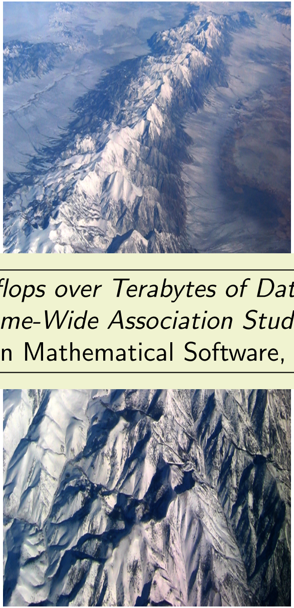
$$V_{LJ} = 4\varepsilon \left[\left(\frac{\sigma}{\ }\right)^{12} - \left(\frac{\sigma}{\ }\right)^{6} \right]$$

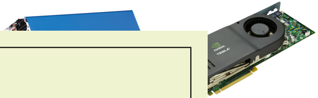
LENN

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\quad 2\mu \quad + V(\mathbf{r},t) \right] \Psi(\mathbf{r},t)$$

SCHRÖDINGER EQN.

⋮





Computing Petaflops over Terabytes of Data:
The Case of Genome-Wide Association Studies.
ACM Transactions on Mathematical Software, 2014.

The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

Linear mixed models

$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Lenn

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[-\frac{\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

Schrödinger eqn.

⋮

A Scalable, Linear-Time Dynamic Cutoff Algorithm for Molecular Dynamics.
International Supercomputing Conference, 2015.

The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

Linear mixed models

$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Lenn

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[-\frac{\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

Schrödinger eqn.

⋮

The Vectorization of the Tersoff Multi-Body Potential: An Exercise in Performance Portability.
Supercomputing (SC'16), 2016.

The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

LINEAR MIXED MODELS

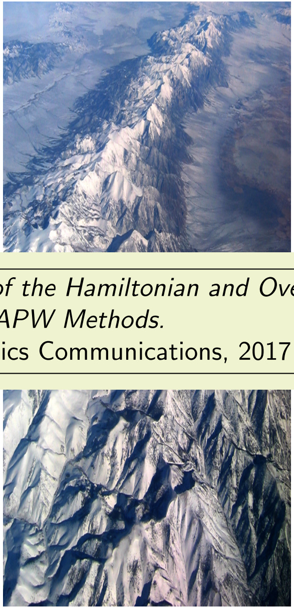
$$V_{LJ} = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

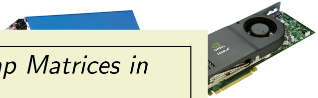
LENN

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

SCHRÖDINGER EQN.

⋮





High-Performance Generation of the Hamiltonian and Overlap Matrices in FLAPW Methods.
Computer Physics Communications, 2017.

The world of scientific computing

$$\mathbf{y} = X\boldsymbol{\beta} + Z\mathbf{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

Linear mixed models

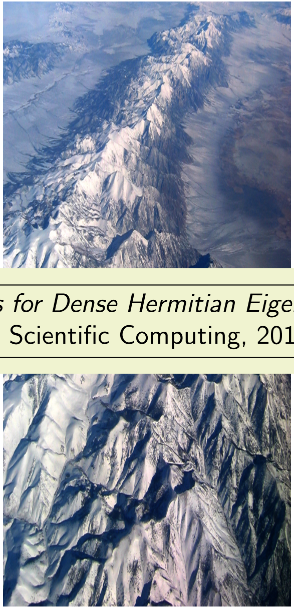
$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

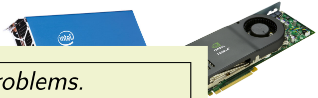
Lenn

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[-\frac{\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

Schrödinger eqn.

⋮





High-Performance Solvers for Dense Hermitian Eigenproblems.
SIAM Journal on Scientific Computing, 2013.

The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

Linear mixed models

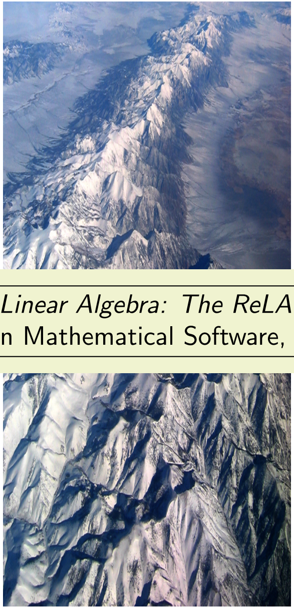
$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Lenn

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[-\frac{\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

Schrödinger eqn.

⋮



Recursive Algorithms for Dense Linear Algebra: The ReLAPACK Collection. ACM Transactions on Mathematical Software, 2017.

The world of scientific computing

$$\boldsymbol{y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon}$$

$$\min_x \|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2$$

Linear mixed models

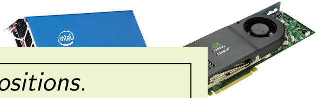
$$V_{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Lenn

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[-\frac{\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

Schrödinger eqn.

⋮



TTC: A High-Performance Compiler for Tensor Transpositions.
ACM Transactions on Mathematical Software, 2017.

A historical perspective: It used to be “easy”


IBM
360

Performance $\equiv$ Op.Count

Performance $\equiv$ Op.Count ... but productivity?

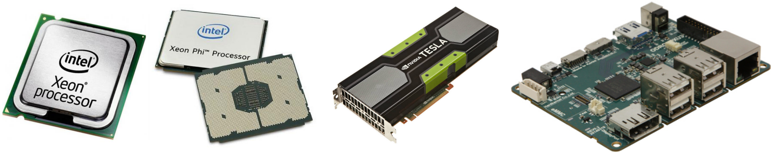
Nowadays

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

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

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

...



MUL ADD MOV

MOVAPD

VFMADDPD ...

Nowadays

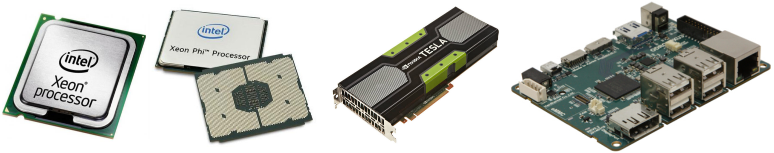
$$x := A(B^TB + A^TR^T\Lambda RA)^{-1}B^TBA^{-1}y$$

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

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

...

?



MUL ADD MOV

MOVAPD

VFMADDPD ...

From math to code

- Scalars
- Vectors & matrices
- Tensors

Scalars

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

Scalars

- [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

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

- **Pros**: A gigantic body of work on compilers

- **Cons**: Almost never the "right" level of abstraction
  Blocking/tiling

Vectors & Matrices

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

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

  Convenience, portability, separation of concerns

Vectors & Matrices

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

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

  Convenience, portability, separation of concerns

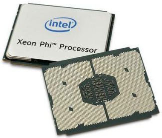
- [80s–early 90s]: Memory hierarchy $\rightarrow$ Op.Count $\not\equiv$ Performance

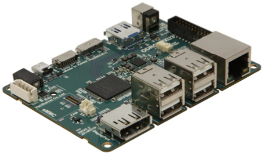
  Libraries $\rightarrow$ necessity

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

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

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





MUL ADD MOV

MOVAPD

VFMADDPD ...

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

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

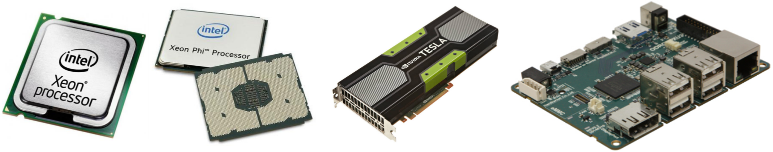
$$E := Q^{-1}U(I + U^TQ^{-1}U)^{-1}U^T$$

...

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

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

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



MUL ADD MOV

MOVAPD

VFMADDPD ...

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

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

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

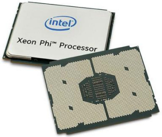
...

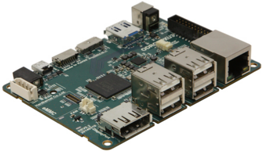
?

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

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

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





MUL ADD MOV

MOVAPD

VFMADDPD ...

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

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

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

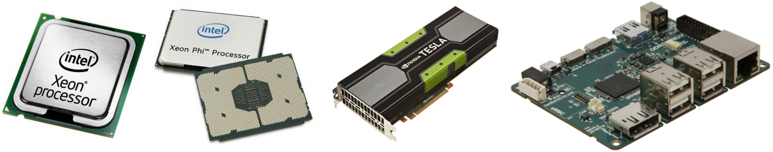
...

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

$X := A^{-1}B$ $C := A$ $^{-1}ML^{-T}$ $QR = A$

LINEAR ALGEBRA MAPPING PROBLEM (LAMP)

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



MUL   ADD   MOV

MOVAPD

VFMADDPD ...

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 $\mathcal{M}$.

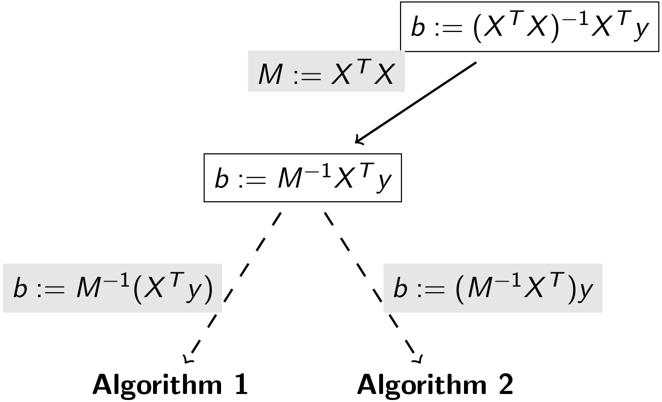
- Find a decomposition $\rightarrow$ easy
- Achieve optimality $\rightarrow$ NP complete

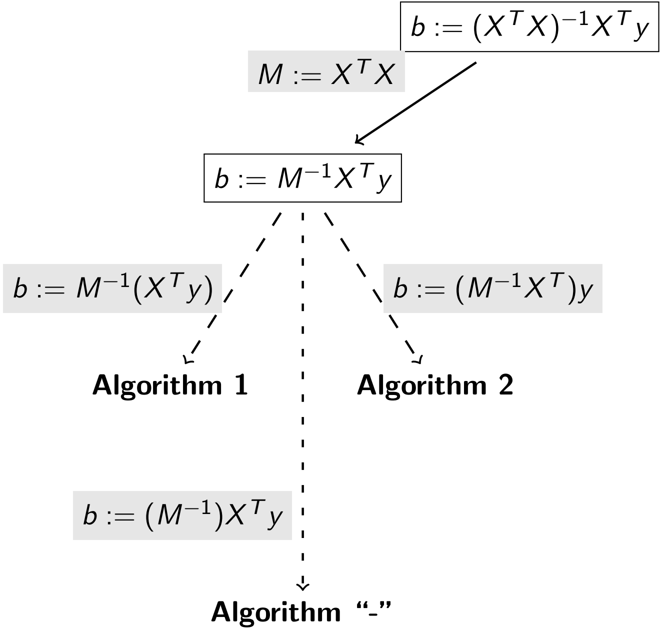
$$\boxed{b := (X^T X)^{-1} X^T y}$$

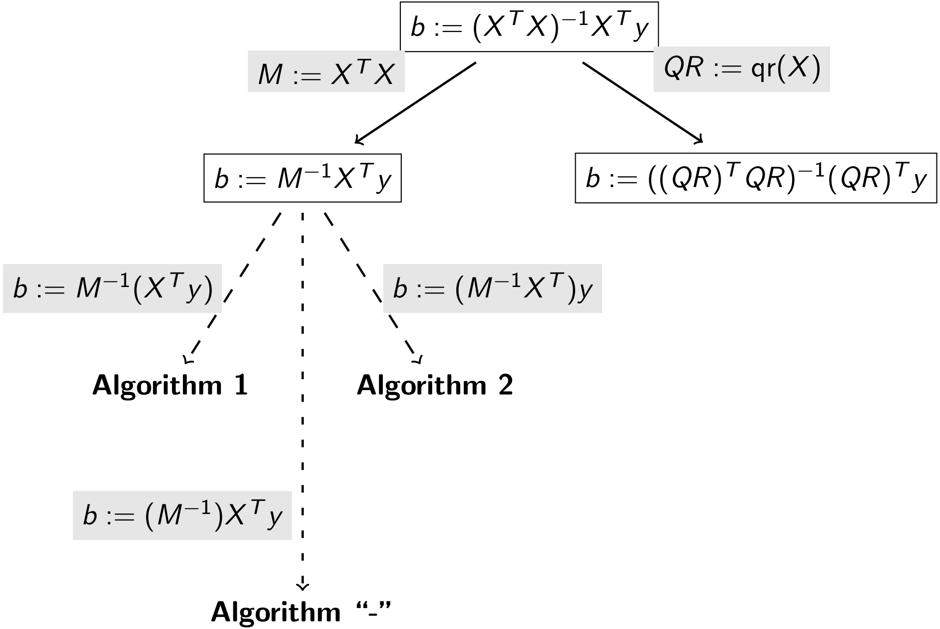
$$b := (X^T X)^{-1} X^T y$$

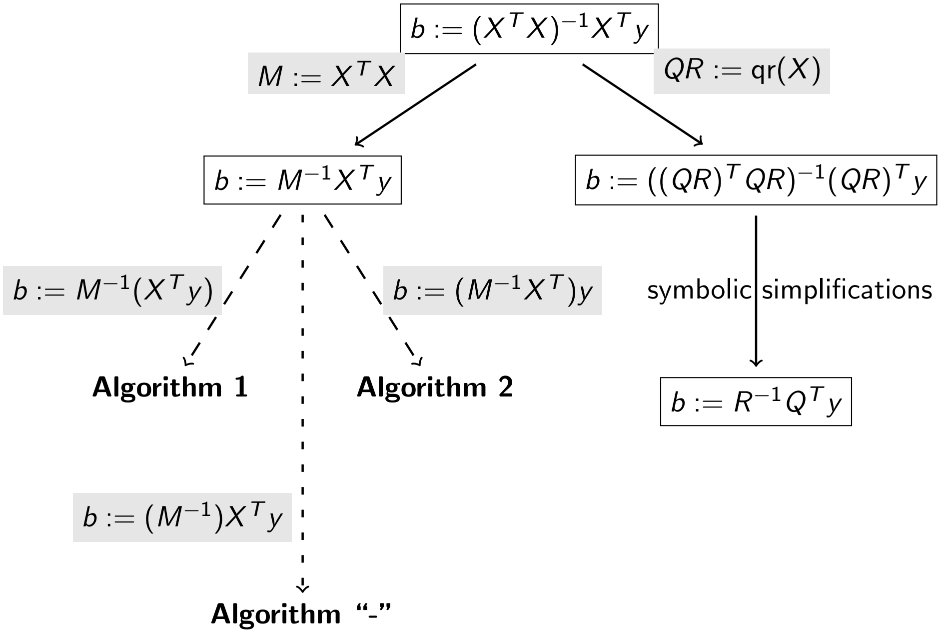
$$M := X^T X$$

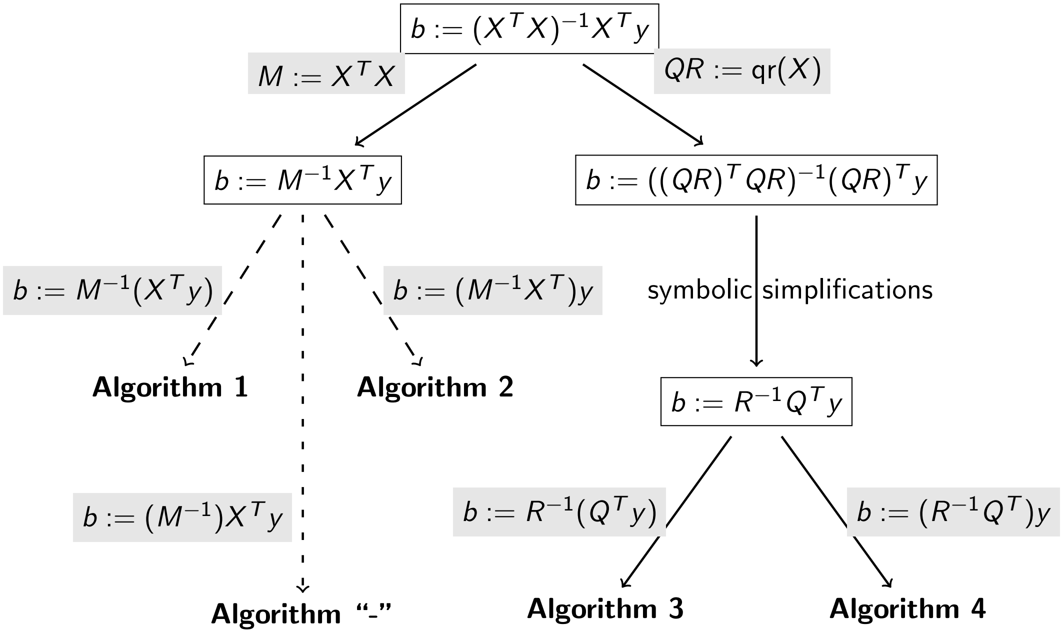
$$b := M^{-1} X^T y$$


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


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


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



$b := (X^T X)^{-1} X^T y$
$M := X^T X$
$QR := \mathrm{qr}(X)$
$b := M^{-1} X^T y$
$b := ((QR)^T QR)^{-1} (QR)^T y$
$b := M^{-1}(X^T y)$
$b := (M^{-1} X^T) y$
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

A well-known problem

High-level languages

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

Libraries

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

A well-known problem

High-level languages

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

Libraries

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

Productivity!

A well-known problem

High-level languages

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

Libraries

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

Productivity!

... but efficiency?

Example

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

Example

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

Naive $\leftarrow$ NEVER!!

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

Example

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

Naive $\leftarrow$ NEVER!!

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

Recommended

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

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

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

Naive $\leftarrow$ NEVER!!

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

Recommended

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

Expert

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

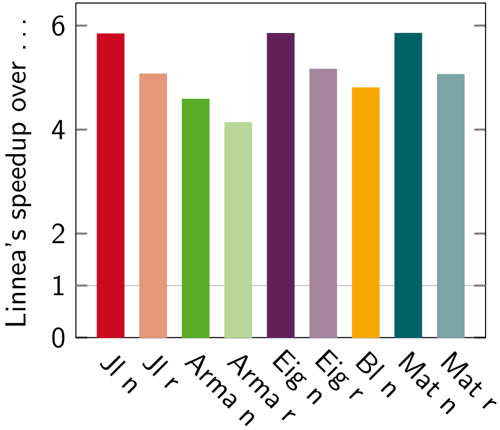
Generated – "Linnea" by H. Barthels

```
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
```

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	$x := W(A^T (A W A^T)^{-1} b - c)$	FullRank(A, W) Diag(W), Pos(W)
$\vdots$		

Linnea – Performance results



The Generalized Matrix Chain Algorithm, CGO 2018 (submitted).

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, CPC, 2017.

Tensors

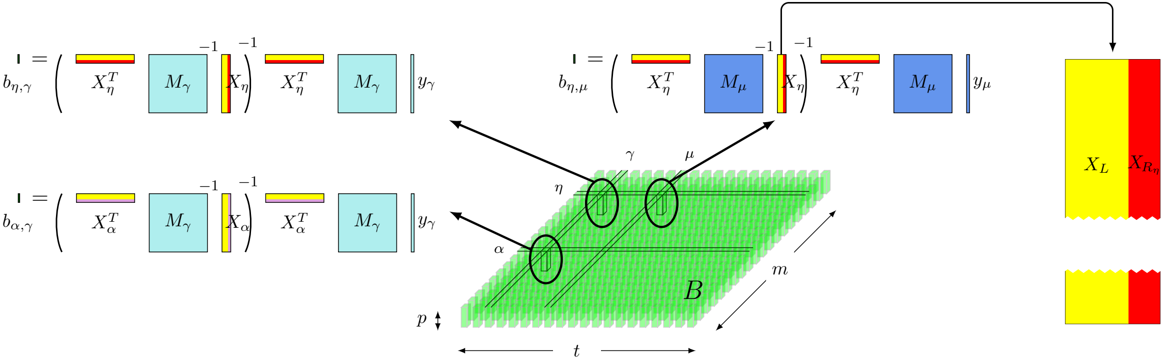
- Building blocks?
  - BLAS, LAPACK

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**: $4N_L^2 N_G$ FLOPs)
    add Y_a to Y_{HPD}
  failure:
    $X_a \coloneqq T_a^{[AA]} A_a$ (**zhemm**: $8N_L^2 N_G$ 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**: $8N_{A_{\neg\mathrm{HPD}}} N_L N_G^2$ FLOPs)
$H \mathrel{+}= Y_{\mathrm{HPD}}^H Y_{\mathrm{HPD}}$ (**zherk**: $4N_{A_{\mathrm{HPD}}} N_L N_G^2$ FLOPs)

10x more flops. Speedups: 1.5–2.5x.

Tensors

- Building blocks?
  - BLAS, LAPACK



Computing Petaflops over Terabytes of Data: The Case of Genome-Wide Association Studies, ACM TOMS 2017.

Tensors

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

Coupled-Cluster methods (CCS, CCSD, . . .)

TTC: A High-Performance Compiler for Tensor Transpositions, ACM TOMS 2017.

Design of a High-Performance GEMM-like Tensor-Tensor Multiplication, (accepted) ACM TOMS 2017.

Spin Summations: A High-Performance Perspective, (submitted) ACM TOMS 2017.

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.

Tensors

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

Tensors

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

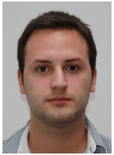
Tensors

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

Tensors

- Building blocks?
  - BLAS, LAPACK
  - Contractions, transpositions, . . .
  - ???
- Do we have a unifying language/formalism?
- Are we ready to fix interfaces & standards?
- For once, shall we focus on **performance *AND* productivity**?
- Thank you, and thanks to the team!


E. Di Napoli

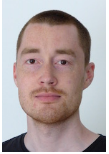

D. Fabregat


M. Petshow


E. Peise


P. Springer


M. Höhnerbach


H. Barthels