High-Performance Tensor Computations: Where Do We Stand?

Paolo Bientinesi
pauldj@cs.umu.se

March 1st, 2021
SIAM Conference on Computational Science and Engineering (via Zoom)

About me (and tensors)

- Taxonomy of contractions: Can you GEMM? — E. Di Napoli, D. Traver-Fabregat, P.B.

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

- Performance prediction — E. Peise, P.B.

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

- Density Functional Theory: FLAPW methods — E. Di Napoli, E. Peise, P.B.

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

- High-performance kernels — P. Springer, P.B.

  "TTC: A high-performance Compiler for Tensor Transpositions", ACM TOMS 44(2), 2017 — + J. Hammond
  "Design of a High-Performance GEMM-like Tensor-Tensor Multiplication", ACM TOMS 44(3), 2018
  "Spin Summations: A High-Performance Perspective", ACM TOMS 45(1), 2019 — + D. Matthews

- High-intensity kernels — C. Psarras, L. Karsson, P.B.

  "Concurrent Alternating Least Squares for multiple simultaneous Canonical Polyadic Decompositions", 2020

Matrices vs. Tensors

Historical overview

Linear Algebra Libraries: 1970s

"Basic Linear Algebra Subprograms for FORTRAN usage", ACM TOMS, 1979

BLAS-1

Linear Algebra Libraries: 1980s

BLAS-2: Mat-vec ops, ACM TOMS 1988. BLAS-3: mat-mat ops, ACM TOMS 1990

BLAS-1, BLAS-2, BLAS-3

Linear Algebra Libraries: 1990s

Solvers & eigensolvers, 1992

LAPACK

BLAS-1, BLAS-2, BLAS-3

Linear Algebra Libraries: 1990s

Distributed Memory, 1995, 1997

ScaLAPACK, PLAPACK, . . .

LAPACK

BLAS-1, BLAS-2, BLAS-3

Linear Algebra Libraries: 1990s

Dense & Sparse, 1997

PETSc, . . .

ScaLAPACK, PLAPACK, . . .

LAPACK

BLAS-1, BLAS-2, BLAS-3

Linear Algebra Libraries

and then more!

PETSc, Trilinos, . . .

ScaLAPACK, PLAPACK, Elemental, . . .

LAPACK, Plasma, SuperMatrix, Magma, . . .

BLAS-1, BLAS-2, BLAS-3, ATLAS, BTO-BLAS, BLIS, . . .

(Dense) Linear Algebra Libraries

Salient features

- Community effort. Standardized interface
- Careful organization: support routines, linear-systems, eigen-decompositions
- Clear layering: functionality, parallelism

(Dense) Linear Algebra Libraries

Salient features

- Community effort. Standardized interface
- Careful organization: support routines, linear-systems, eigen-decompositions
- Clear layering: functionality, parallelism
- Preferred outlet: ACM TOMS
- Performance & HW-driven development

(Dense) Linear Algebra Libraries

Salient features

- Community effort. Standardized interface
- Careful organization: support routines, linear-systems, eigen-decompositions
- Clear layering: functionality, parallelism
- Preferred outlet: ACM TOMS
- Performance & HW-driven development

But

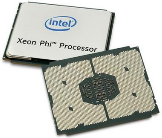
- Rigid interface
- Black-box nature
- Often sub-optimal at small scale

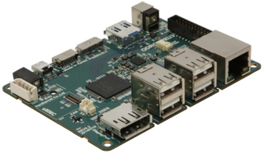
$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k (z_k - H x_k^b); \quad P_k^a := (I - K_K H)\, P_k^b$$

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

$$\Lambda := S(S^T AWAS)^{-1} S^T; \; \Theta := \Lambda AW; \; M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A) \Theta$$

$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y \qquad \ldots \qquad E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T$$





MUL ADD MOV

MOVAPD

VFMADDPD ...

$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k(z_k - H x_k^b); \quad P_k^a := (I - K_K H)\, P_k^b$$

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

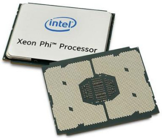
$$\Lambda := S(S^T AWAS)^{-1} S^T; \; \Theta := \Lambda AW; \; M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A)\Theta$$

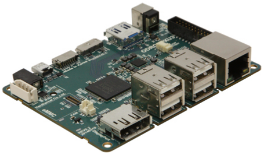
$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y \quad \ldots \quad E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T$$

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

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

... BLAS LAPACK ...





MUL | ADD | MOV

MOVAPD

VFMADDPD ...

$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k(z_k - H x_k^b); \quad P_k^a := (I - K_K H)\, P_k^b$$

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

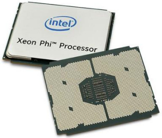
$$\Lambda := S(S^T AWAS)^{-1} S^T; \; \Theta := \Lambda AW; \; M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A) \Theta$$

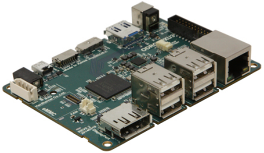
$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y \qquad \ldots \qquad E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T$$

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

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

... BLAS LAPACK ...





MUL ADD MOV

MOVAPD

VFMADDPD ...

$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k(z_k - H x_k^b); \quad P_k^a := (I - K_K H) P_k^b$$

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

$$\Lambda := S(S^T AWAS)^{-1} S^T; \; \Theta := \Lambda AW; \; M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A) \Theta$$

$$x := A(B^T B + A^T R^T \Lambda R A \quad \ldots$$

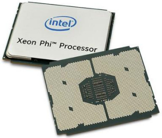
$$\ldots^{-1} U(I + U^T Q^{-1} U)^{-1} U^T$$

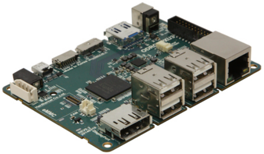
LINEAR ALGEBRA MAPPING PROBLEM (“LAMP”)

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

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

... BLAS LAPACK ...





MUL | ADD | MOV

MOVAPD

VFMADDPD ...

$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k(z_k - H x_k^b); \quad P_k^a := (I - K_K H)\, P_k^b$$

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

$$\Lambda := S(S^T AWAS)^{-1} S^T; \ \Theta := \Lambda AW; \ M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A) \Theta$$

$x := A(B^T B + A^T R^T \Lambda R A$ … $^{-1} U(I + U^T Q^{-1} U)^{-1} U^T$

LINEAR ALGEBRA MAPPING PROBLEM (“LAMP”)

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

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

… BLAS LAPACK …

C. Psarras, H. Barthels, *“The Linear Algebra Mapping Problem. Current state of linear algebra languages and libraries”.* [arXiv:1911.09421]

H. Barthels, C. Psarras, *“Linnea: Automatic Generation of Efficient Linear Algebra Programs”, ACM TOMS, 2021.* [arXiv:1912.12924]

Tensors

Tensor App #1 | Tensor App #2 | . . . | Tensor App #N

??? | ??? | · · · | ???

??? | ??? | · · · | ???

BLAS
LAPACK

???

. . .

???

MUL ADD MOV

MOVAPD

VFMADDPD . . .

but. . .

Comp. Physics

Comp. Chemistry

Data Science

Machine Learning

???	...	???
???		???

???	...	???
???		???

BLAS

LAPACK

???

. . .

???

MUL ADD MOV

MOVAPD

VFMADDPD . . .

Tensor computations

- (At least) Two separate worlds

[1]With notable differences.

Tensor computations

- (At least) Two separate worlds
  - Computational physics / chemistry

    Tensor = Multi-linear operator

    Contractions = Generalization of matrix-matrix product

[1]With notable differences.

Tensor computations

- (At least) Two separate worlds
  - Computational physics / chemistry

    Tensor = Multi-linear operator

    Contractions = Generalization of matrix-matrix product
  - Data science

    Tensor = Collection of data

    Decompositions = Generalization of matrix factorizations[1]

[1] With notable differences.

Tensor computations

- (At least) Two separate worlds
  - Computational physics / chemistry

    Tensor = Multi-linear operator

    Contractions = Generalization of matrix-matrix product
  - Data science

    Tensor = Collection of data

    Decompositions = Generalization of matrix factorizations[1]
- Terminology and notation vary (and conflict) even within one world
- Very few software efforts cut across the boundary

[1]With notable differences.

MS14 (today) & MS79 (tomorrow)

- (At least) Two separate worlds
  - Computational physics / chemistry

    Quantum Chemistry: Ed Valeev, Devin Matthews, Edgar Solomonik

    Quantum Physics: Pan Zhang, Lei Wang, Miles Stoudenmire
  - Data science

    Furong Huong, Hanie Sedghi, Vagelis Papalexakis
- Notation: Miles Stoudenmire
- Software: Edgar Solomonik

Tensors, presently

- No “Tensor BLAS” – no collections of building blocks
- No agreement on interface(s)
- Lack of reference implementations

Tensors, presently

- No "Tensor BLAS" – no collections of building blocks
- No agreement on interface(s)
- Lack of reference implementations
- No community effort!
  A jungle of independent libraries and packages, in a variety of languages
  Massive redundancy: replication of effort, low performance

Tensors, presently

- No “Tensor BLAS” – no collections of building blocks
- No agreement on interface(s)
- Lack of reference implementations
- No community effort!
  A jungle of independent libraries and packages, in a variety of languages
  Massive redundancy: replication of effort, low performance
- Application-driven development:
  Publications scattered among diffents fields
  Re-invention of the wheel

Matlab and R packages with support for CP decomposition (subset)

- **Tensor Toolbox** by Bader, Kolda, & others
  `https://www.tensortoolbox.org/`
- **Tensorlab** by Vervliet, Debals, Sorber, Van Barel, & De Lathauwer
  `https://www.tensorlab.net/index.html`
- **The N-way Toolbox** by Bro & Andersson
  `http://www.models.life.ku.dk/nwaytoolbox`
- **TensorBox** by Phan, Tichavsky, & Cichocki
  `https://github.com/phananhhuy/TensorBox`
- **Tensor Package** by Comon & others
  `http://www.gipsa-lab.fr/~pierre.comon/TensorPackage/tensorPackage.html`

- **multiway** by Helwig
  `https://cran.r-project.org/package=multiway`
- **ThreeWay** by Giordani, Kiers, & Del Ferraro
  `https://cran.r-project.org/package=ThreeWay`
- **rTensor** by Li, Bien, & Wells
  `https://cran.r-project.org/package=rTensor`

C/C++ packages with support for CP decomposition (subset)

- **Genten** by SANDIA (Phipps)
  `https://gitlab.com/tensors/genten`
- **SPLATT** by Smith & Karypis
  `https://github.com/ShadenSmith/splatt`
- **ParTI!** by Li, Ma, & Vuduc
  `https://github.com/hpcgarage/ParTI`
- **Cyclops** by Solomonik & others
  `https://github.com/cyclops-community`

And then there's Python, Fortran, ...

Representative operations – building blocks candidates

Data layout operations

- Reshape
- Permute / transpose
- Sort (sparse)
- Convert data layout
- Partition
- Distribute
- ...

Arithmetic operations

- Add, subtract, scale
- Inner product
- Norms
- Element-wise operations
- Tensor-times-vector (TTV)
- Tensor-times-matrix (TTM)
- MTTKRP
- Contractions
- ...

Decompositions

- CP (CANDECOMP/PARAFAC)
- Tucker
- INDSCAL
- PARAFAC2
- CANDELINC
- DEDICOM
- PARATUCK2
- ...

Representative operations – building blocks candidates

Data layout operations

- Reshape
- Permute / transpose
- Sort (sparse)
- Convert data layout
- Partition
- Distribute
- ...

Arithmetic operations

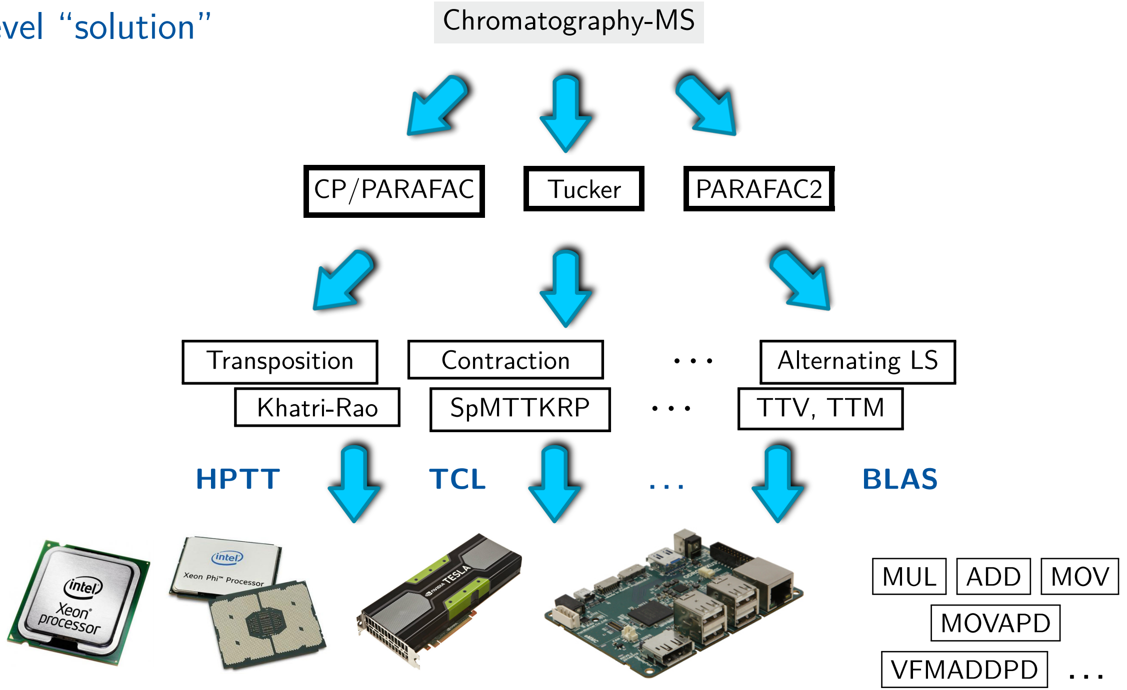
- Add, subtract, scale
- Inner product
- Norms
- Element-wise operations
- Tensor-times-vector (TTV)
- Tensor-times-matrix (TTM)
- MTTKRP
- Contractions
- ...

Decompositions

- CP (CANDECOMP/PARAFAC)
- Tucker
- INDSCAL
- PARAFAC2
- CANDELINC
- DEDICOM
- PARATUCK2
- ...

In setting up a library, where to draw the boundaries?

2-level "solution"



Algorithms for CP (PARAFAC) decomposition

Hence all the different libraries

- **Algebraic algorithms**
  - Generalized Rank Annihilation Method
  - Direct TriLinear Decomposition
  - The "algebraic algorithm" by Domanov and De Lathauwer
  - The "simpler algorithm" by Pimentel-Alarcón
  - ...
- **Alternating optimization algorithms**
  - Alternating Least Squares
  - Fast ALS
  - Hierarchical ALS
  - Regularized ALS
  - ...
- **All-at-once optimization algorithms**
  - Gradient descent
  - (Damped) Gauss–Newton
  - Nonlinear CG, GMRES
  - Quasi-Netwon (e.g., L-BFGS)
  - ...
- **Enhancements**
  - Line search
  - Compression
  - Randomization
  - Transient constraints
  - ...

Also . . .

Also . . . one vs. many instances

Coupled-Cluster methods

$$
\begin{aligned}
\tau_{ij}^{ab} &= t_{ij}^{ab} + \frac{1}{2} P_b^a P_j^i t_i^a t_j^b, \\
\tilde{F}_e^m &= f_e^m + \sum_{fn} v_{ef}^{mn} t_n^f, \\
\tilde{F}_e^a &= (1 - \delta_{ae}) f_e^a - \sum_m \tilde{F}_e^m t_m^a - \frac{1}{2} \sum_{mnf} v_{ef}^{mn} t_{mn}^{af} + \sum_{fn} v_{ef}^{an} t_n^f, \\
\tilde{F}_i^m &= (1 - \delta_{mi}) f_i^m + \sum_e \tilde{F}_e^m t_i^e + \frac{1}{2} \sum_{nef} v_{ef}^{mn} t_{in}^{ef} + \sum_{fn} v_{if}^{mn} t_n^f,
\end{aligned}
$$

$$
\begin{aligned}
\tilde{W}_{ei}^{mn} &= v_{ei}^{mn} + \sum_f v_{ef}^{mn} t_i^f, \\
\tilde{W}_{ij}^{mn} &= v_{ij}^{mn} + P_j^i \sum_e v_{ie}^{mn} t_j^e + \frac{1}{2} \sum_{ef} v_{ef}^{mn} \tau_{ij}^{ef}, \\
\tilde{W}_{ie}^{am} &= v_{ie}^{am} - \sum_n \tilde{W}_{ei}^{mn} t_n^a + \sum_f v_{ef}^{ma} t_i^f + \frac{1}{2} \sum_{nf} v_{ef}^{mn} t_{in}^{af}, \\
\tilde{W}_{ij}^{am} &= v_{ij}^{am} + P_j^i \sum_e v_{ie}^{am} t_j^e + \frac{1}{2} \sum_{ef} v_{ef}^{am} \tau_{ij}^{ef}, \\
z_i^a &= f_i^a - \sum_m \tilde{F}_i^m t_m^a + \sum_e f_e^a t_i^e + \sum_{em} v_{ei}^{ma} t_m^e + \sum_{em} v_{im}^{ae} \tilde{F}_e^m + \frac{1}{2} \sum_{efm} \\
z_{ij}^{ab} &= v_{ij}^{ab} + P_j^i \sum_e v_{ie}^{ab} t_j^e + P_b^a P_j^i \sum_{me} \tilde{W}_{ie}^{am} t_{mj}^{eb} - P_b^a \sum_m \tilde{W}_{ij}^{am} t_m^b + P
\end{aligned}
$$

credits to D. Matthews, E. Solomonik, J. Stanton, and J. Gauss

Also . . . one vs. many instances

Coupled-Cluster methods

$$
\begin{aligned}
\tau_{ij}^{ab} &= t_{ij}^{ab} + \frac{1}{2}P_b^a P_j^i t_i^a t_j^b, \\
\tilde{F}_e^m &= f_e^m + \sum_{fn} v_{ef}^{mn} t_n^f, \\
\tilde{F}_e^a &= (1-\delta_{ae})f_e^a - \sum_m \tilde{F}_e^m t_m^a - \frac{1}{2}\sum_{mnf} v_{ef}^{mn} t_{mn}^{af} + \sum_{fn} v_{ef}^{an} t_n^f, \\
\tilde{F}_i^m &= (1-\delta_{mi})f_i^m + \sum_e \tilde{F}_e^m t_i^e + \frac{1}{2}\sum_{nef} v_{ef}^{mn} t_{in}^{ef} + \sum_{fn} v_{if}^{mn} t_n^f, \\
\tilde{W}_{ei}^{mn} &= v_{ei}^{mn} + \sum_f v_{ef}^{mn} t_i^f, \\
\tilde{W}_{ij}^{mn} &= v_{ij}^{mn} + P_j^i \sum_e v_{ie}^{mn} t_j^e + \frac{1}{2}\sum_{ef} v_{ef}^{mn} \tau_{ij}^{ef}, \\
\tilde{W}_{ie}^{am} &= v_{ie}^{am} - \sum_n \tilde{W}_{ei}^{mn} t_n^a + \sum_f v_{ef}^{ma} t_i^f + \frac{1}{2}\sum_{nf} v_{ef}^{mn} t_{in}^{af}, \\
\tilde{W}_{ij}^{am} &= v_{ij}^{am} + P_j^i \sum_e v_{ie}^{am} t_j^e + \frac{1}{2}\sum_{ef} v_{ef}^{am} \tau_{ij}^{ef}, \\
z_i^a &= f_i^a - \sum_m \tilde{F}_i^m t_m^a + \sum_e f_e^a t_i^e + \sum_{em} v_{ei}^{ma} t_m^e + \sum_{em} v_{im}^{ae} \tilde{F}_e^m + \frac{1}{2}\sum_{efm} \\
z_{ij}^{ab} &= v_{ij}^{ab} + P_j^i \sum_e v_{ie}^{ab} t_j^e + P_b^a P_j^i \sum_{me} \tilde{W}_{ie}^{am} t_{mj}^{eb} - P_b^a \sum_m \tilde{W}_{ij}^{am} t_m^b + P
\end{aligned}
$$

credits to D. Matthews, E. Solomonik, J. Stanton, and J. Gauss

Finite Element 3D diffusion operator

```
TE.BeginMultiKernelLaunch();
TE("T2_e_i1_i2_k3 = B_k3_i3 X_e_i1_i2_i3", T2, B, X);
TE("T1_e_i1_k2_k3 = B_k2_i2 T2_e_i1_i2_k3", T1, B, T2);
TE("U1_e_k1_k2_k3 = G_k1_i1 T1_e_i1_k2_k3", U1, G, T1);
TE("T1_e_i1_k2_k3 = G_k2_i2 T2_e_i1_i2_k3", T1, G, T2);
TE("U2_e_k1_k2_k3 = B_k1_i1 T1_e_i1_k2_k3", U2, B, T1);
TE("T2_e_i1_i2_k3 = G_k3_i3 X_e_i1_i2_i3", T2, G, X);
TE("T1_e_i1_k2_k3 = B_k2_i2 T2_e_i1_i2_k3", T1, B, T2);
TE("U3_e_k1_k2_k3 = B_k1_i1 T1_e_i1_k2_k3", U3, B, T1);
TE("Z_m_e_k1_k2_k3 = U_n_e_k1_k2_k3 D_e_m_n_k1_k2_k3", Z, U,
TE("T1_e_i3_k1_k2 = B_k3_i3 Z1_e_k1_k2_k3", T1, B, Z1);
TE("T2_e_i2_i3_k1 = B_k2_i2 T1_e_i3_k1_k2", T2, B, T1);
TE("Y_e_i1_i2_i3 = G_k1_i1 T2_e_i2_i3_k1", Y, G, T2);
TE("T1_e_i3_k1_k2 = B_k3_i3 Z2_e_k1_k2_k3", T1, B, Z2);
TE("T2_e_i2_i3_k1 = G_k2_i2 T1_e_i3_k1_k2", T2, G, T1);
TE("Y_e_i1_i2_i3 += B_k1_i1 T2_e_i2_i3_k1", Y, B, T2);
TE("T1_e_i3_k1_k2 = G_k3_i3 Z3_e_k1_k2_k3", T1, G, Z3);
TE("T2_e_i2_i3_k1 = B_k2_i2 T1_e_i3_k1_k2", T2, B, T1);
TE("Y_e_i1_i2_i3 += B_k1_i1 T2_e_i2_i3_k1", Y, B, T2);
TE.EndMultiKernelLaunch();
```

credits to A. Fisher – https://github.com/LLNL/acrotensor

Same in data science: Gas Chromatography

Workflow

...

4. **Fit model or rank $k \in [1, \dots, 15]$**, if needed, add non-negativity constraints
   Tensor decompositions: PARAFAC — PARAFAC2 — TUCKER

5. Determine whether or not one of the models is "right"
   - : Determine which of the components represent chemical information
   - : Start over; add/change constraints, change model

Same in data science: Gas Chromatography

Workflow

. . .

4. **Fit model or rank $k \in [1, \dots, 15]$**, if needed, add non-negativity constraints
   Tensor decompositions: PARAFAC — PARAFAC2 — TUCKER

5. Determine whether or not one of the models is “right”
   - : Determine which of the components represent chemical information
   - : Start over; add/change constraints, change model

Computation of each individual model: bandwidth bound!

Hence: *“Concurrent Alternating Least Squares for multiple simultaneous Canonical Polyadic Decompositions”, with C. Psarras, L. Larsson. (Submitted).*

Summary

	Matrices	**Tensors**
Driver	performance, HW	applications
Community effort	BLAST/LAPACK/...	group by group
Industry	wide support	not much
Standardization	interface, ...	"pointless"
Preferred outlet	ACM TOMS	—
Language support	plenty	language by language
Automation	plenty	TCE (2001), but then?

Summary

	Matrices	Tensors
Driver	performance, HW	applications
Community effort	BLAST/LAPACK/...	group by group
Industry	wide support	not much
Standardization	interface, . . .	"pointless"
Preferred outlet	ACM TOMS	—
Language support	plenty	language by language
Automation	plenty	TCE (2001), but then?

Still a long way to maturity! — Thank you for the attention.